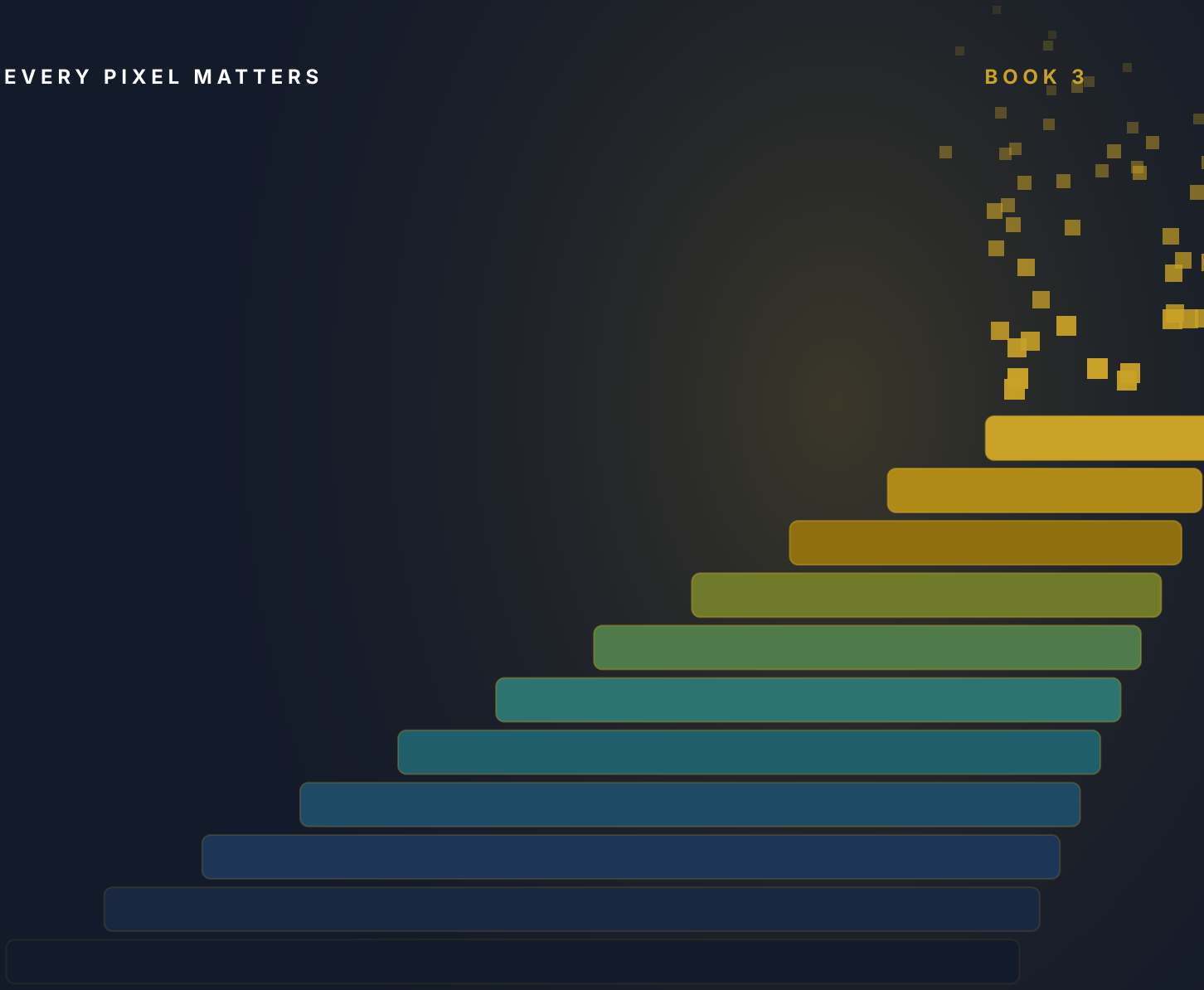
EVERY PIXEL MATTERS

BOOK 3



FROM SILICON TO SCREEN

From Silicon to Screen

Who Controls Your Systems, and What Trust Costs

Every system your people trust can be changed by more hands than any team can name: chip makers, firmware signers, kernel and driver vendors, identity providers, management consoles, package maintainers and AI agents. This book walks the stack from silicon to screen on Windows, macOS, iOS, Linux and Android, shows who controls each layer, how fast each can be fixed and how each recovers, and gives a CTO the Trust Ledger: a plain way to decide what to own, rent, pin or isolate, judged by security, cost, visibility and business value.

Juan Cardena

Enterprise AI & Agentic Systems Architect · blog.jcardena.com

CONTENTS

BEFORE YOU START

Reader's Brief: Who Controls Your Systems

PROLOGUE

One Power Button, One Approval

PART I · THE MAP

01 Who Controls It: Authorities, Clocks and the Trust Ledger

02 Five Platforms, Five Trust Models

PART II · BELOW THE OS

03 Silicon and the Computers Inside the Computer

04 Firmware and Boot: Keys with End Dates

05 Privileged Code: Kernels, Hypervisors, Drivers and the Morning of 19 July 2024

PART III · THE CONTROL PLANES

06 Identity and Keys: Who Can Sign, Who Can Log In

07 The Management Plane: Everyone Who Can Push

08 The Software Supply Chain: Signed Is Not Safe

09 When Trust Must Be Withdrawn: Stop, Revoke, Roll Back, Recover

PART IV · THE CHAIN AND THE TAIL

10 Zero-Click: From the Screen to the Silicon

11 The Long Tail: End of Support, Edge Appliances and Security Debt

PART V · MONEY, RECORD AND DECISIONS

12 Who Pays for Security: The Market

13 The Record: Spyware, Sabotage and the Outage

14 The Cost of Trust: Control, Concentration and a Day Down

15 Own, Rent, Pin, Isolate: The Decision and the Contract

PART VI · AHEAD

16 AI on Both Sides of the Stack

17 The Next Trust Migration: Memory Safety, Attestation and Post-Quantum Keys

CLOSING

Epilogue: The Last Authority Is You

APPENDICES

A Glossary

B The Trust Ledger, Blank

C The Five-OS Trust Reference Table

D Measuring the Patch and Recovery Clocks

F Case File Index

G The Visibility Map

H Annotated Contract Clauses

E Reading List and Sources

BEFORE YOU START

Reader's Brief: Who Controls Your Systems

Who can change each layer of your systems? How fast does a fix reach you? What does it cost when it fails? This book asks those three questions about every layer between the chip and the screen, on Windows, macOS, iOS, Linux and Android. This brief tells you who the book is for, introduces the three companies that come back in every chapter, shows the one map the book uses, and gives you a reading path for your role. It takes ten minutes.

Who this book is for

It is written for the **CTO, VP Engineering or chief architect** at a company of 50 to 1,000 engineers, and, in this book, equally for the **CISO**. Companies of that size rent almost every layer they run: the processor and its microcode, the firmware, the operating system, the security sensor in the kernel, the identity provider, the management console. They are large enough that one bad change in one of those layers reaches thousands of devices before anyone at the company has approved it.

You do not need to write kernel code. Technical terms are explained where they first appear in the text. You do need to answer for four things about each layer: its **security**, its **cost**, how much of it you can actually **see**, and the **business value** it protects. Book 1 of this series, *From Bit to Pixel*, walked the whole stack and asked what to own and what to rent. Book 2, *From Code to Glass*, stayed on the layer that draws the screen. This book goes back down the stack with one new question per layer: who controls it. You can read it without the other two.

Two readers will get less from it. A team that writes firmware or kernels already knows the mechanics, though the chapters on recovery, money and contracts may still be new. A startup with one cloud app and no device fleet should read the prologue, Chapter One and Chapter Fifteen, and come back when it has devices, regulators or customers' desktops to answer for.

The idea in one paragraph

Security is usually sold as a set of controls. This book treats it as a set of **authorities**: the parties that can write, build, sign, publish, push or configure the code in each layer of your systems, plus anyone who can change it by exploiting it. For each layer it records four facts. Who can change it. How fast a fix reaches 90% of *your* devices, not the vendor's. Who can

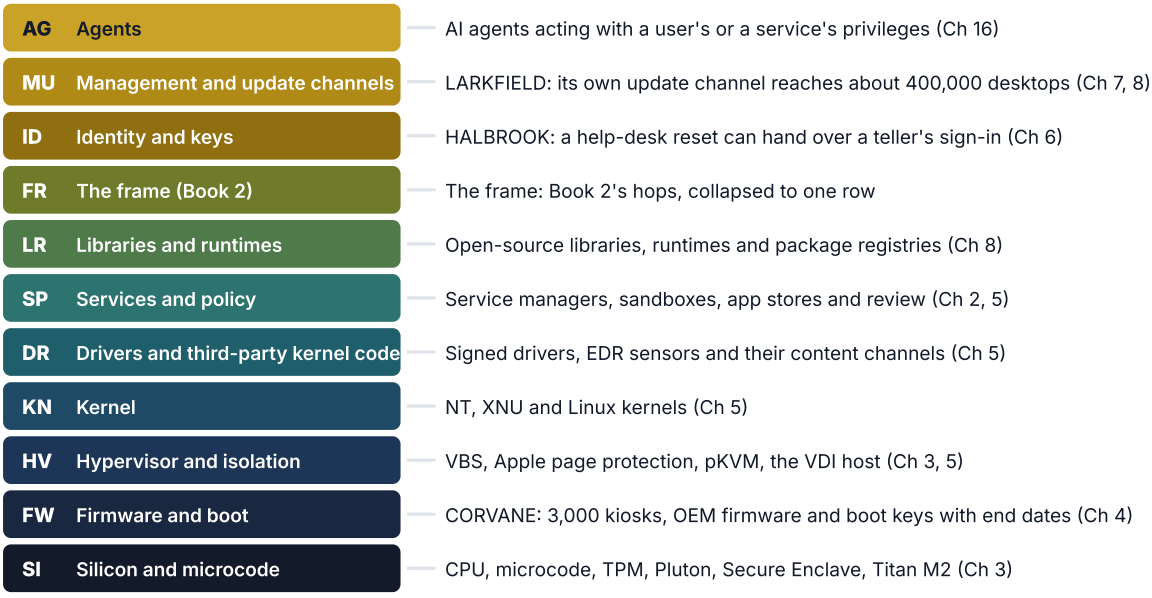
see it go wrong. How fast you can stop the change, revoke what must be revoked and recover. The record is a one-page worksheet called the **Trust Ledger**, and Chapter One shows you how to fill it in. Every decision in the book is one of two choices, **Own** or **Rent**, with two modifiers: **Pin** a version until a stated date, or **Isolate** the layer from something named.

The master stack

Figure B.1 is the master stack, the one map the book uses; later figures, such as Figure 1.3, reuse its rows. It has eleven rows, from silicon at the bottom to AI agents at the top, and each row has a two-letter code that the rest of the book uses. Each chapter opens with a thin strip of the same rows and raises the ones it covers. A twelfth row, EX, holds what sits outside the device but on the recovery path, such as VPN appliances and the cloud host; it appears only where a chapter needs it.

FIGURE B.1

The master stack: eleven rows from silicon to agents, and where each company's riskiest authority sits



A company name marks that company's riskiest row. Inside the chapters, red marks only a row that a case attacked or broke.

Every chapter opens one or more of these rows; the three company names mark where each running company's riskiest authority sits.

Author's diagram. Row codes from the book's Trust Ledger (Chapter One); the three companies are invented composites.

Three companies, three ways to lose

Advice about "trust" is easy to agree with and hard to use, so every chapter tests its ideas on the same three companies. All three are invented. Each is a composite of a common kind of business, and they are the same three companies as in Book 2, now seen from below the screen. Each chapter takes one of them as its lead.

HALBROOK BANK, THE REGULATED COMPANY

Halbrook is a listed US regional bank with 2,500 branch PCs on Windows, advisors on laptops and on virtual desktops in a rented cloud, and a mobile app on iOS and Android. It owns a small subsidiary in Ireland. Its questions are about identity and concentration. Who can reset a teller's sign-in? How many outside parties can push code into a branch desktop without a Halbrook person approving it? What happens when the one security sensor on every desk fails at once? Its clocks are set by regulators: the US banking agencies' rule to notify the regulator within 36 hours of determining that a "notification incident" has occurred (a computer-security incident that has materially disrupted or degraded its core operations or a key business line, or is reasonably likely to), the SEC's Form 8-K Item 1.05 four business days after it determines an incident is material, and, for the Irish subsidiary only, the EU's Digital Operational Resilience Act (DORA), in application since 17 January 2025.

CORVANE HOTELS, THE DEVICE FLEET OPERATOR

Corvane is a hotel group headquartered in the Netherlands, with 300 properties in the EU and the US, about 3,000 self check-in kiosks on Windows IoT, Android tablets at the front desks and signage on Linux. Its machines live for years in lobbies with no engineer nearby. Its questions are about time. Which firmware and boot keys are on each kiosk, and when do they stop getting fixes? What does a kiosk that will not start cost in site visits? Can a property recover with no network? Its main clock is the EU's General Data Protection Regulation (GDPR), Article 33: report a personal-data breach to the supervisory authority within 72 hours of becoming aware of it, where feasible, unless the breach is unlikely to result in a risk to people's rights and freedoms. Its payment kiosks also fall under the card industry's PCI DSS rules wherever they store, process or transmit card data, or can affect its security.

LARKFIELD SOFTWARE, THE VENDOR

Larkfield sells a desktop app with about 400,000 installs on Windows, macOS and Linux, plus a companion on iOS and Android. It runs its own update channel and holds its own code-signing key, which makes it one of the authorities in its customers' ledgers. Its questions are the vendor's side of trust. Who can sign as Larkfield? Which open-source packages does it ship? How fast can it stop a bad release? Its clock is the EU Cyber Resilience Act: as a manufacturer selling in the EU it must report actively exploited vulnerabilities from 11 September 2026, and the rest of the regulation applies from 11 December 2027.

The value map

Twelve questions, one per company per lens, each with the chapter that answers it. Find your company's row and start there.

Company	Security	Cost	Visibility	Value
Halbrook	Who can reset a teller's sign-in, and how is that path attacked? **Chapter Six**	What does one day with every branch desk down cost, and who pays? **Chapter Fourteen**	How many authorities can push code to a branch desktop without us? **Chapter One**	Which rows do we own, rent, pin or isolate, and what goes in the contract? **Chapter Fifteen**
Corvane	Who controls the code that runs before Windows on 3,000 kiosks? **Chapter Four**	What does a kiosk that will not start cost in site visits, multiplied across 300 properties? **Chapter Fourteen**	Which kiosks have the 2023 boot certificates, and how would we know? **Chapter Four**	How fast can a property recover with no engineer on site? **Chapter Nine**
Larkfield	What if our signing key or build system is the way in? **Chapter Eight**	How long does the Cyber Resilience Act oblige us to keep fixing what we ship? **Chapter Eleven**	Who else can push to our customers' desktops besides us? **Chapter Seven**	Can we stop a bad release before it reaches 400,000 desktops? **Chapter Nine**

Reading paths

The book reads front to back: Part I gives the map and the worksheet, Part II goes below the operating system, Part III covers the control planes that can change everything at once, Part IV follows one attack chain and the long tail, Part V covers money, the record and the decision, and Part VI looks ahead. If you have one evening, follow your role.

Your role	Read first	Then	What you will decide differently
CTO or VP Engineering	Prologue, Chapters One and Two	Seven, Nine, Fourteen, Fifteen	Treat security as a map of authorities with clocks, not a tool budget
CISO or security architect	Chapters One, Four, Five	Six, Seven, Ten, Sixteen	Put firmware, boot keys, kernel-level vendors, help desks and consoles in scope, each with an owner
Chief architect	Chapter Two	Three, Five, Eight, Seventeen	Choose platforms by who holds the keys and what you may change
End-user computing or fleet owner	Chapters One, Four, Nine	Seven, Eleven, Appendix D	Measure days to 90% of the fleet and the time to recover, from your own telemetry
CFO, procurement or risk	Prologue, Chapter Fourteen	Twelve, Thirteen, Fifteen, Appendix H	Price control against a day down; write staged rollout and recovery support into contracts
Board or audit	This brief, the prologue, Chapter Fifteen	Thirteen, Fourteen	Ask which authorities can change our systems without us knowing, and how fast we recover
AI platform lead	Chapters One, Seven	Sixteen, Seventeen	Treat every agent as an authority with its own row in the ledger

How the book handles facts

Each date, number, version and name in the chapters was checked, before the sources were frozen on 18 November 2026, against a primary source: a vendor advisory or document, a standard, a court record, a government report, an original research paper, or the source code itself. Where a number is a party's own estimate, the book says whose. Where a fact has not been published, the book marks it **np** (not published) and draws it as a gap, never as a measured bar. Companies and governments are named in connection with attacks only as courts, governments, Citizen Lab or the vendor's own advisory name them. The three running companies are invented. Opinions are labeled "Our view", with a date, because a book about clocks should say when its own clock stopped. Each chapter's sources are listed in the Reading List and Sources appendix; errors found after the freeze will be corrected there and in the next edition.

HOW TO USE THIS BOOK ON MONDAY

Security. Pick the one system whose failure would hurt most, such as Halbrook's teller desktop, Corvane's kiosk or Larkfield's update channel.

Cost. Write down what one day without it costs. A rough number is enough to rank the rows.

Visibility. List every party that can push code into it. Mark the ones you would not see doing so.

Value. Read Chapter One with that list beside you, and fill in the first Trust Ledger row before Chapter Two.

Monday rule: one system, one ledger, one named owner per row. The rest of the book fills in the rows.

PROLOGUE

One Power Button, One Approval



At 07:58 on a Tuesday, a teller at a Halbrook Bank branch presses the power button on her desktop. At 08:02 she approves a customer's payment. Between those two moments the machine runs code written, built, signed and shipped by seven parties outside the bank, and at least five update channels could have changed that code while the branch was dark. Halbrook wrote one small piece of it. This prologue walks those four minutes in order and names, for each step, who could have changed it, and who would have noticed.

The desktop is a business PC running Windows 11 Enterprise, managed with Microsoft Intune (Microsoft's cloud device-management service), encrypted with BitLocker, and protected by a third-party endpoint detection and response (EDR) sensor. Nothing about it is unusual: the authorities on this desk are the ones found on many desks in regulated companies. The section headings use the two-letter row codes of the Reader's Brief, from SI (silicon) to MU (management and update channels).

07:58, rows SI and FW: before Windows exists

Power reaches the processor, which starts fetching instructions from the firmware chip at a fixed address. How the processor carries out its instructions can be adjusted after manufacture by **microcode**, a layer of low-level code the chip maker writes. Microcode updates reach this PC by two routes: inside the PC maker's firmware, or as an operating-system update that Windows loads early in each boot. Microsoft has shipped Intel microcode through Windows Update since at least 2018, documented in its support articles such as KB4093836. Chapter Three explains why a fleet should record which route actually applied the running revision.

Next comes the **firmware**, the code that starts the hardware and finds the operating system. On a PC it follows the UEFI specification. It is usually written by an independent BIOS vendor, customized and signed by the PC maker, and updated through the PC maker's own tool or through Windows Update. Many business PCs also check the firmware's own signature in hardware before it runs; Chapter Four names those mechanisms.

The firmware then enforces **Secure Boot**. It will start only boot code whose signature matches its allowed list, the **DB**, and does not appear on its forbidden list, the **DBX**. Two other key stores decide who may change those lists: the **Platform Key** (PK), normally held by the PC maker, and the **Key Exchange Key** list (KEK), which holds Microsoft's key. Microsoft's documentation lets PC makers hold the PK themselves or use a Microsoft-managed one. As the boot proceeds, the firmware measures each boot component and extends its hash into registers of the **Trusted Platform Module** (TPM), the platform configuration registers (PCRs), while an event log outside the TPM records what was measured, so that a trusted server can later check what booted. Microsoft calls this Measured Boot.

Who can change it: the chip maker (microcode), the BIOS vendor and PC maker (firmware), the PC maker and Microsoft (key stores). **Who would notice:** the TPM's measurements, if anyone reads them: an attestation service, such as the device-health checks in Intune's compliance policies, or Keylime on Linux. Halbrook's own staff have not touched any of it.

07:58 to 07:59, rows FW, HV, KN and DR: Windows starts

The firmware hands over to the **Windows Boot Manager**, signed by Microsoft under a certificate called Windows Production PCA 2011. That certificate reached the end of its validity on 19 October 2026, and Microsoft's 2023 replacements have to be in the firmware's lists before new boot-level protections can reach the machine. Microsoft states that devices without the new certificates "will continue to start and operate normally". Chapter Four tells that story; the desk still boots.

On capable hardware, Windows then starts its own hypervisor and uses it to run **virtualization-based security** (VBS): a small, isolated region beside the kernel that holds secrets and checks kernel code. Its best-known feature, memory integrity, is on by default on many new Windows 11 devices that meet Microsoft's requirements. Then the **kernel** loads, written, built and signed by Microsoft.

The first third-party driver to start on this desk is normally the EDR sensor's. Windows provides a class of **Early Launch Antimalware** (ELAM) drivers, which Microsoft documents as loading "before all non-Microsoft boot drivers", so that security software can judge the drivers that follow; the sensor goes first only if its vendor registers it as one. On a new installation of Windows 10 version 1607 or later with Secure Boot on, every new kernel driver must carry Microsoft's signature; Chapter Five lists the exceptions. That signature proves who submitted the driver. It does not fix what the driver will read: the sensor's behavior is shaped by **content**, rule and configuration files that the EDR vendor sends on its own schedule, separately from the driver itself.

Who can change it: Microsoft (boot manager, hypervisor, kernel) and the EDR vendor (driver and content). **Who would notice:** Microsoft's attestation and the EDR vendor's console. Halbrook sees the console; it does not see the content before it lands.

08:00, row ID: the teller signs in

The teller unlocks the desktop with **Windows Hello for Business**. Microsoft describes it as two factors: "the user's private key protected by the device's security module" and a PIN that "never leaves the device". The key is tied to Halbrook's identity provider, a cloud directory the bank rents. Before the bank's systems trust the session, Intune reports whether the device is compliant with Halbrook's policies, and the identity provider issues the tokens that the teller's apps present for the rest of the day.

Two paths around that sign-in matter more than the sign-in itself. One is the **help desk**: when a teller forgets her PIN or loses her phone, someone resets her authentication methods, and whoever can talk that person into a reset can become the teller. The other is the **BitLocker recovery key**, the 48-digit password that unlocks the disk when the TPM refuses to. Microsoft's documentation notes that access to recovery passwords stored in its directory "can also be delegated to the help desk", and Halbrook has delegated it. Its help desk therefore holds the key to two rows: the person, and the disk. Chapter Six follows the first; Chapter Nine follows the second.

Who can change it: Halbrook's identity team, Halbrook's help desk, and the identity provider. **Who would notice:** the identity provider's sign-in logs, if someone reviews resets.

08:01 to 08:02, rows LR, FR and SP: the approval

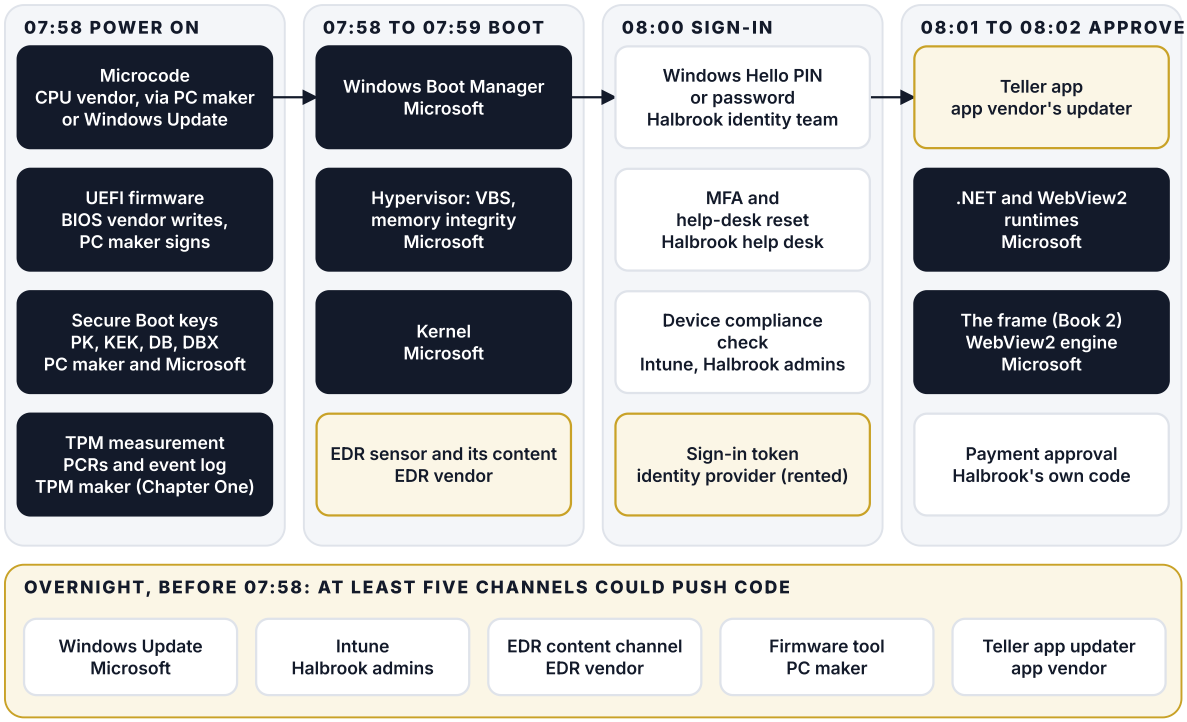
The teller opens the payment app. It is built by a software vendor on Microsoft's .NET runtime and shows its approval screen through Microsoft's Edge WebView2 runtime (WebView2), which Book 2 followed hop by hop from the tap to the glass. The vendor's own updater keeps the app current. The approval rules and the button's meaning are Halbrook's own code, the one part of these four minutes the bank wrote.

Around the app sit the **services and policy** of the operating system: the Defender platform, the sandbox WebView2 places around web content, the Group Policy or Intune settings Halbrook pushes. The frame itself, from the page to the pixels, is one row in this book, FR, and Book 2 is its home.

Who can change it: the app vendor, Microsoft (runtimes and platform) and Halbrook (policy and the approval logic). **Who would notice:** Halbrook's app logs, for the part Halbrook wrote.

FIGURE P.1

Four minutes at a branch desk, and every authority that could have changed them



Between the power button and the approval, code from seven parties outside the bank and three Halbrook teams runs, and at least five channels can change that code overnight.

Author's diagram of a typical Windows 11 business PC, based on Microsoft's Secure Boot, measured boot, ELAM and VBS documentation (Chapter One sources). Times are illustrative; the order is the documented boot order.

Overnight, row MU: at least five channels that did not ask

The four minutes started long before 07:58. While the branch was dark, at least five channels could push code or content to this desk; Chapter One finds more.

- **Windows Update**, through the rings Halbrook configures: monthly security updates, released on the second Tuesday of each month since Microsoft moved to a monthly cycle in October 2003.
- **Intune**, run by Halbrook's endpoint team: apps, scripts and policy, as often as an administrator presses a button.
- **The EDR vendor's content channel**: new detection content, on the vendor's schedule, usually without a Halbrook change ticket.
- **The PC maker's firmware and driver tool**: firmware and driver packages, on the tool's own schedule, which is how Halbrook has left it.

- **The payment app's updater**: new versions from the app vendor.

Halbrook can delay or stage the first two. It could stage the fourth through the PC maker's management settings, but has not. The third and the fifth are, in Halbrook's current contracts, the vendor's to schedule. Each is a privileged channel into every branch desk at once, and each is an authority in its own right. Figure P.1 shows all of them on one page.

The same desk on 19 July 2024

On Friday 19 July 2024 at 04:09 UTC, shortly after midnight on the US East Coast, one of those overnight channels published a content file to Windows hosts running the CrowdStrike Falcon sensor, and Windows machines that were online and received it before the vendor reverted it at 05:27 UTC were exposed to a crash; many crashed, and many again on every restart; Microsoft estimated about 8.5 million devices. Figure P.2 traces that morning on the rows of this desk. The full case has two homes, the engineering in Chapter Five and the record in Chapter Thirteen; here it is only a trace on this desk's rows.

FIGURE P.2

The same desk on 19 July 2024: a content file stopped it before the power button



Numbered steps in order. Red: the failure path. Branch times assume US Eastern time (UTC minus 4).

One content update from one rented channel crashed the kernel, and the way back ran through the identity row: the disk-encryption recovery key.

CrowdStrike, technical details (20 July 2024) and root cause analysis (6 August 2024); Microsoft, recovery guidance and blog (20 July 2024). Full case in Chapter Five; the record in Chapter Thirteen.

What matters here is the shape of the path, not the vendor. The failure entered through the management row, ran through the drivers row into the kernel, and stopped the machine below the point where any remote tool could reach it. The way back ran through the identity row: on encrypted machines, a technician needed the BitLocker recovery key before the bad file could be removed, which Microsoft's recovery guidance (KB5042421) describes. The authority that caused the failure could not undo it on a machine that would not start. The authority that could undo it, by hand, was the customer.

From the trace to the map

Put the four minutes and the overnight hours together and you have the shape of this book. Every layer between the chip and the approval has **authorities**: parties that write it, build it, sign it, publish it, push it or configure it, and anyone who can change it by exploiting it. Every layer has a **patch clock**: how long a fix takes to reach most of your own devices, not the vendor's. Every layer has a **recovery clock**: how long it takes to stop a bad change and get the fleet back. And for every layer, someone can or cannot **see** what happened.

Figure B.1, in the Reader's Brief, draws the stack those authorities sit on. Chapter One turns it into a worksheet, the Trust Ledger, and fills in its first row for this desk. Chapter Two draws the same map five times, for Windows, macOS, iOS, Linux and Android, because the authorities are not the same on each, and neither is what the customer is allowed to change.

> *Halbrook wrote one piece of the code that ran in those four minutes. It answers for all of it.*

SECURITY LENS

Who can change it? On one branch desk: the chip maker, the BIOS vendor, the PC maker, Microsoft, the EDR vendor, the identity provider, the app vendor, and three Halbrook teams: ten parties in these four minutes. Chapter One's full inventory of the desk adds the TPM maker, a printer driver maker and a desk-side support contractor, folds the BIOS vendor into the PC maker, whose firmware image carries its code, and counts twelve: nine outside the bank, three inside. Anyone who exploits any of them joins the list.

How fast can it be fixed? At the speed of the slowest authority that has to act, then at the speed of Halbrook's own rollout. Nobody outside Halbrook can measure the second clock for it.

What does AI change here? Our view (as of November 2026): AI makes finding and weaponizing flaws in each of these layers cheaper, and makes help-desk impersonation more convincing. It does not add a single new channel to the desk; it makes the existing ones busier.

BUSINESS IMPLICATIONS

Security. The least watched part of Halbrook's day is not the approval; it is the overnight window when at least five channels can push without asking. Corvane's kiosks have the same window, with no teller to notice a blank screen at 07:58.

Cost. Every outside authority on the desk is also a contract or a license. Most of those contracts say little about how changes are staged, how they are stopped, or who pays when a change stops the desk.

Visibility. Halbrook can see its own app, its own policy and its own console. Most of what runs before 08:00 reports only to the party that wrote it. For Larkfield, the vendor, the mirror image applies: its updater is one of the overnight channels on someone else's desk.

Value. Customers notice the approval at 08:02, and the branch that cannot open at 09:00. Recovery time, not the number of controls, is what they feel.

Monday rule: list every channel that can change your most important desk overnight, and mark which ones you can pause.

CHAPTER ONE

Who Controls It: Authorities, Clocks and the Trust Ledger



Halbrook Bank's audit committee asks its CISO the question boards have asked for twenty years: "Are we secure?" She answers with a different question and a number. The question is: who, outside this room, can change the software on our 2,500 branch desktops tonight without anyone here approving it? The number, from an inventory her team took in a week, is four parties with no change ticket at all, out of twelve that can change the desktop in some way. Nobody on the committee had asked for that number before. This chapter shows how to produce it, row by row, and turns it into a one-page worksheet, the Trust Ledger. The decision it changes: you stop buying security as a list of controls and start managing it as a list of authorities, each with a clock.

Trust is a list of authorities, not a perimeter

The older picture of security draws a wall around the company: trusted inside, untrusted outside. The teller's four minutes in the prologue show why that picture fails on a modern device. The code that runs on the desk was written, built and signed outside the wall, and much of it arrives through channels that start outside the wall and end inside the kernel. The useful question is not "is it inside?" but "who can change it?"

This book calls each party that can change a layer an **authority**. "Change" covers six different powers, and they are often held by different parties for the same component (Figure 1.1).

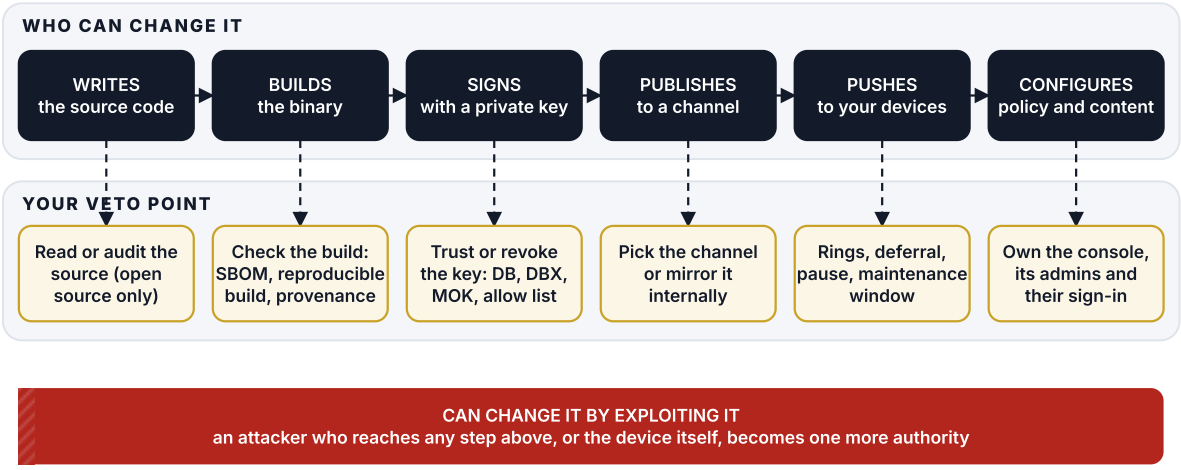
- **Writes:** the people who author the source code, such as an open-source maintainer or a vendor's engineers.
- **Builds:** whoever turns source into the binary you run, on a build system you rarely see.
- **Signs:** whoever holds the private key whose signature your device accepts. A signature proves who released the code, not that the code is good.

- **Publishes:** whoever runs the channel the code is offered through: an update server, a store, a repository.
- **Pushes:** whoever decides when the code reaches your devices: a vendor's automatic update, your own management console, a managed service provider.
- **Configures:** whoever sets the policy and content that change the code's behavior without changing the code, such as a security sensor's detection rules.

There is a seventh authority on every row, and nobody grants it: **anyone who can change the component by exploiting it**. An attacker who steals a signing key becomes a signer; one who takes over a management console becomes a pusher. The six granted authorities decide how many doors there are. The seventh decides which door gets tried.

FIGURE 1.1

Six ways to hold authority over a component, and the veto point a customer keeps at each



Count every party at every step; the customer's power is the veto under each one, and a check is a veto only where policy enforces it.

Author's diagram. The split of authority into writing, building, signing, publishing, deploying and configuring follows the book's Trust Ledger method (Chapter One).

Under each authority sits a **veto point**, the place where the customer can say no. You can audit open source, check a vendor's build provenance and its software bill of materials (SBOM, the list of components in a build), refuse or revoke a key, mirror a channel internally, stage a push in rings, and own the console and the people who sign in to it. Our view (as of November 2026): many companies use only the last two. Larkfield Software sees

the same picture from the other side: for its 400,000 customer desktops, Larkfield *is* the writer, builder, signer, publisher and pusher of one row, and every one of its customers has a veto point it may or may not use.

Four facts per row

Once you have named the authorities for a layer, four facts decide how much risk that layer carries. They are the spine of every chapter.

Fact	The question	Why it decides risk
Who can change it	Which authorities can write, build, sign, publish, push or configure it, and how far one action reaches	Each authority is a way in, and one action by a pusher can reach every device at once
How fast it is fixed	How many days from the first public sign of a flaw to the fix running on 90% of *your* devices	Exposure lasts until your fleet is fixed, not until the vendor ships
Who can see it	What telemetry, logs or attestation report on this layer, and to whom	A layer nobody watches fails silently, and its recovery starts late
How fast you stop and recover	Whether you can pause the channel, revoke the key or file, roll back, and how long until 90% of devices work again	Customers feel the outage, not the root cause

These four facts describe a row. The four lenses of the Reader's Brief (security, cost, visibility, value) judge the decision you take on it: the facts are the evidence, the lenses the verdict.

Every fact gets an **evidence tag**, because a ledger full of confident guesses is worse than one that admits what it does not know. **O** means observed on your own fleet. **D** means documented by the vendor. **C** means promised in a contract. **I** means inferred. **U** means unknown. A row tagged U is not a failure of the exercise; it is the exercise's main finding.

Two clocks: patch and recovery

The **patch clock** has four marks (Figure 1.2). **T0** is the earliest of three moments: the vendor learns of the flaw, the flaw becomes public, or it is known to be exploited. **T1** is the day a fix is available for *this device class*, which may be weeks after the first fix for another

class. **T2** and **T3** are the days the fix is running on 50% and 90% of your own devices. After T3 comes the **tail**: the devices still exposed, counted, with a reason for each group.

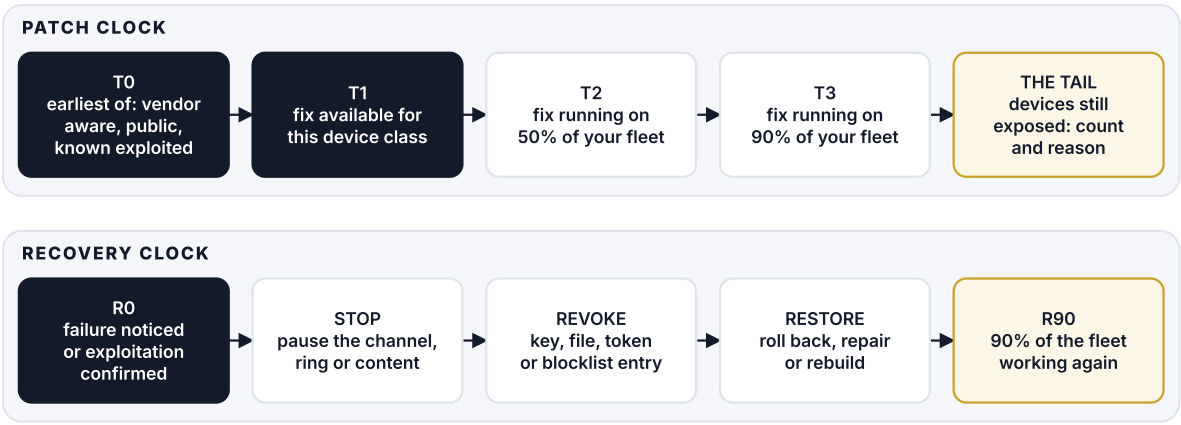
T0 to T1 is mostly the vendor's. T1 to T3 is mostly yours, though vendors still gate parts of it with staged offers and holds, and only your own telemetry can measure it: Windows Update or Intune compliance reports, Apple MDM inventory, Android management reports, package timestamps on Linux. Vendor adoption curves describe the vendor's whole customer base, not your fleet. Appendix D gives the method.

The number that protects you is T3, not the average. An attacker may need only one exposed, reachable device, and the tail is where the devices that nobody restarts, nobody owns or nobody can reach collect: the kiosk in a closed lobby, the laptop in a drawer, the virtual machine template. The NIST guide to enterprise patch management, SP 800-40 Revision 4 (April 2022), frames patching as preventive maintenance, a cost of doing business to be planned and resourced, not an emergency. The same framing applies to the tail: it needs an owner and a budget, not a hope.

Regulators have started to set the clock for you. When the US Cybersecurity and Infrastructure Security Agency (CISA) created its Known Exploited Vulnerabilities (KEV) catalog with Binding Operational Directive 22-01 on 3 November 2021, it gave federal civilian agencies two weeks to fix newer catalog entries and six months for older ones. On 10 June 2026, CISA revoked that directive and replaced it with BOD 26-04, which sets the deadline per vulnerability from four facts: whether the asset is publicly exposed, whether the flaw is in the KEV catalog, whether exploitation can be automated, and how much control an attacker gains. The shortest deadline is three days. The directive binds only federal agencies, but its logic, exposure and exploitation decide the clock, is the logic of this book.

FIGURE 1.2

Two clocks per row: how fast a fix reaches your fleet, and how fast your fleet comes back



T0 to T1 is mostly the vendor's clock; T1 to T3 and most of the recovery clock are yours, and only your own devices can measure them.

Author's diagram. Patch management as planned preventive maintenance follows NIST SP 800-40 Rev. 4 (April 2022).

The **recovery clock** runs on the same page. **R0** is the moment a failure or an exploitation is noticed. Then come three actions, usually in this order and sometimes in parallel: **stop** the channel, ring or content that caused it; **revoke** whatever must no longer be trusted, a key, a file, a token, a blocklist entry; **restore** by rolling back, repairing or rebuilding. **R90** is the moment 90% of the fleet works again. The morning of 19 July 2024, told in full in Chapters Five and Thirteen, is the case that made the recovery clock visible: the stop took the vendor 78 minutes, and restore, for many companies, took days of hands on keyboards. Chapter Nine is about this clock alone.

The Trust Ledger

The **Trust Ledger** records the authorities, dependencies, evidence and exit paths for one critical system on one A4 page. It is not a list of things the company believes are trustworthy. Book 1's Pixel Bill of Materials answered what is on the box; Book 2's Frame Ledger answered how the screen gets drawn. The Trust Ledger answers who can change each layer, how fast a fix reaches you, who can see it fail, and how you stop and recover. It sits beside registers auditors already ask for: Halbrook's Irish subsidiary must keep DORA's register of information "in relation to all contractual arrangements on the use of ICT services provided by ICT third-party service providers" (Article 28(3)), and the ledger's rented rows name the

same providers, adding what the register does not hold: each one's clocks and exit path. In the same way it supplements, and does not replace, the supplier controls auditors already test, such as the supply-chain category (GV.SC) of the NIST Cybersecurity Framework 2.0.

It has three parts. The **header block**, written once per system, holds the facts that apply to every row: the system and the business service it supports; the **platform profile** (exact operating system release, hardware class and management state, fixed in Chapter Two); the fleet denominator, meaning how many devices are in scope; the business value at stake and the cost of one day down; the regulator clocks that apply, with what starts each; the loss the ledger protects against; the assessment date and next review; and the accountable executive.

The **rows** are the eleven layers of the master stack, plus an optional twelfth.

Code	Row	Examples	Home chapter
SI	Silicon and microcode	CPU, GPU, NPU, microcode, TPM, Pluton, Secure Enclave, Titan M2, management engines	Three
FW	Firmware and boot	UEFI and its key stores (PK, KEK, DB, DBX), iBoot, Android Verified Boot, shim, BMC, radios	Four
HV	Hypervisor and isolation	VBS and memory integrity, Apple's page protection, pKVM, the cloud host under a virtual desktop	Three, Five
KN	Kernel	Windows NT, XNU, Linux	Five
DR	Drivers and third-party kernel code	Signed drivers, EDR sensors and their content, kernel modules, eBPF programs	Five

SP	Services and policy	Service managers, SELinux, AppArmor, privacy consent, app sandboxes, stores and review	Two, Five
LR	Libraries and runtimes	System libraries, open-source dependencies, language runtimes, package registries	Eight
FR	The frame	Book 2's eleven hops from source to glass, collapsed to one row	Book 2
ID	Identity and keys	Device keys, sign-in, the help-desk reset path, token-signing and code-signing keys, the identity provider	Six
MU	Management and update channels	OS update, PC maker tools, MDM, remote management, EDR console and content, app updaters	Seven
AG	Agents	Any AI agent or automation acting with a user's or a service's privileges	Sixteen
EX	External dependencies (optional)	VPN and firewall appliances, the cloud host, DNS, key escrow	Eleven

The **eight core columns** are the same on every row:

1. **Component:** the real component and version, its update policy (evergreen, pinned, tied to the OS, bundled), and how many devices run it.
2. **Who can change it:** the named authorities, split by power where they differ: writes, builds, signs, publishes, pushes, configures. A vendor's content channel counts.

3. **Exposure and blast radius:** which untrusted inputs reach it (network, files, messages, USB, a help-desk call, a console login) or "local admin only"; what privilege it runs with; how many devices or tenants one action reaches.
4. **Patch clock:** T0, T1, T2, T3 and the tail for the last one to three critical cases, measured on your fleet.
5. **Stop, revoke, recover:** the pause switch, the revocation path, the rollback, the date of the last drill, and the measured time to R90.
6. **Who can see it:** telemetry, attestation or logs, or "nothing", with its evidence tag.
7. **Support end:** the date the owner stops fixing it, the source of that date, and the lead time to replace it.
8. **Decision and owner:** Own or Rent, any Pin or Isolate modifier, a one-line reason, a named person and the next review date.

Six optional columns can be added when a chapter or an audit needs them: the root of trust the row verifies against, the yearly cost of control, who pays on failure, an AI exposure flag, a link to the Pixel Bill of Materials or Frame Ledger, and a contract clause reference.

One fill rule keeps the ledger honest: **any cell you cannot fill in twenty minutes gets "np" (not published), "nc" (not collected) or "U" (unknown)**. All three mean unknown; the first two also say why: the vendor has not published the fact, or you have not collected it. Those marks are the finding. A row with three of them is a row where the company is trusting an authority it cannot measure.

Own or Rent, then Pin or Isolate

Every row ends in a decision on two separate axes. The first axis is who holds the authority. **Own** means your company holds it: you sign, you push, or you configure, and you answer for the clock. **Rent** means another party holds it under a contract or a license, and you answer for choosing them and for watching them. Almost everything below the operating system is rented; owning a row usually means owning its policy and its schedule, not its code.

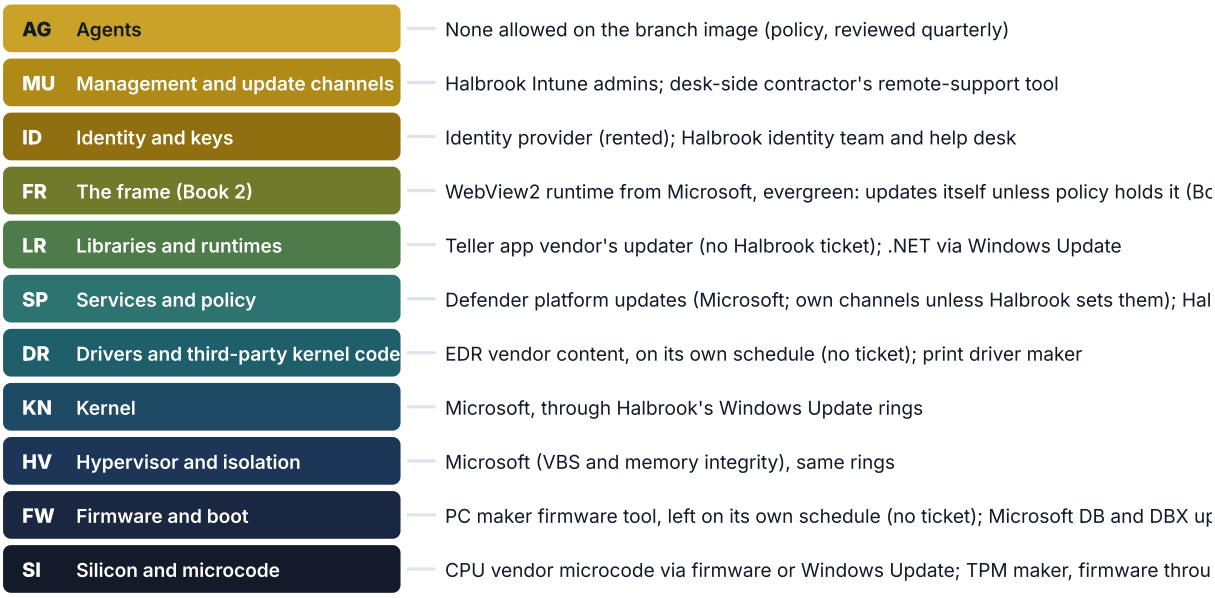
The second axis has two modifiers. **Pin** holds a version still: a firmware release on a kiosk model, a kernel line on a server image. A pin without an end date is debt, so a pin must carry one. **Isolate** limits what a row can reach or what can reach it: an EDR's content staged by property group, a kiosk with no AI agents on its image. An isolation that does not say what it isolates *from* is a slogan, so it must name its partner. Chapter Fifteen turns these choices into a flowchart and contract clauses.

Counting the authorities on one desk

The CISO's number in this chapter's opening came from a simple inventory: for each row of the branch desktop, list every party that can push a change into it, and mark the ones that can do so without a Halbrook change ticket. Figure 1.3 shows the result. Twelve parties can change the desktop. Three of them are Halbrook teams. Four outside parties need no Halbrook ticket at all: the EDR vendor's content channel, the teller app vendor's updater, the PC maker's firmware tool, which Halbrook has left on its own schedule, and the TPM maker, whose firmware arrives through that tool. The count is by party, not by channel, and one rule decides what a party is: a maker counts separately when its code is a separate component that reports its own version on the device, as TPM firmware does, and folds into another party when its code is built into that party's image, as the BIOS vendor's code is built into the PC maker's firmware. Microsoft counts as ticketed because most of its changes pass through Halbrook's Windows Update rings, but two of its channels, Defender platform updates and the evergreen WebView2 runtime, follow Microsoft's own schedule unless Halbrook sets update policy for them. Chapter Seven counts channels. Counted by row instead, one party stands out: Microsoft is named on seven of the eleven rows of Figure 1.3, and sits on two more as the signer of drivers and the operator of Intune. That is the concentration figure Halbrook's board asked for.

FIGURE 1.3

One Halbrook branch desktop: twelve parties can change it, four of them without a Halbrook change ticket



Invented inventory. Twelve parties, counted by party: nine outside the bank, three Halbrook teams (identity team, help desk, endpoint admin:

The count is not the attack surface, but every party on this list is a row with a clock, a channel and a contract.

Author's inventory for an invented company (Halbrook Bank), built on Microsoft's documented update paths for Windows 11 Enterprise. Counts are illustrative; Chapter Seven gives the inventory method.

The count is not the attack surface. A party with no ticket may be well run, and a party behind a ticket may be careless. But every party on the list is a row with a clock, a channel and a contract, and the four without a ticket are the ones whose changes Halbrook learns about after they land.

Who holds the keys, by platform

The same desk on another operating system has different authorities. Chapter Two draws each platform in full; this summary shows only the differences that change a decision in this chapter.

Question	Windows	macOS	iOS	Linux	Android
Who holds the boot keys	PC maker (Platform Key) and Microsoft (KEK, DB)	Apple, in the chip	Apple, in the chip	Microsoft's UEFI CA signs shim; the distribution's key in shim; the owner may add keys	The device maker
Who signs the kernel	Microsoft	Apple	Apple	The distribution, or you	Google certifies the Generic Kernel Image; the device maker adds modules and signs the image
Third-party code in the kernel	Yes, Microsoft-signed	Only after the owner lowers startup security; most tools run in user space	None	Yes, signed modules when Secure Boot is on	Chip and device-maker modules
Who ships the OS fix	Microsoft	Apple	Apple	The distribution	Google publishes; each device maker ships
What the customer can delay	Rings and deferrals	MDM deferral, 1 to 90 days	MDM deferral, 1 to 90 days	Everything: you run the repositories	Postpone an update up to 30 days, or freeze updates for periods up to 90 days (managed devices)
Firmware and silicon	BIOS vendor and PC maker; microcode via firmware or Windows Update	Apple only	Apple only	PC maker; through LVFS only where the maker takes part	Chip maker and device maker

HOW IT GOT THIS WAY

The idea that a system's security rests on a small, checkable core is older than the PC. In February 1970 a Defense Science Board task force chaired by Willis Ware delivered *Security Controls for Computer Systems* (RAND R-609). It was classified, and stayed so until the Defense Advanced Research Projects Agency declassified it on 10 October 1975. It treated a resource-sharing computer as something whose hardware, software and procedures all had to be controlled together. In October 1972 James P. Anderson's *Computer Security Technology Planning Study* (ESD-TR-73-51) for the US Air Force gave that core a name, the **reference monitor**, and three requirements: the mechanism "must be tamper proof", "must always be invoked", and "must be small enough to be subject to analysis and tests, the completeness of which can be assured".

Hold today's desk up against those three requirements. The mechanism that decides what runs on it is spread across a chip maker's microcode, a firmware vendor's code, two parties' boot keys, an operating-system vendor's kernel and hypervisor, a security vendor's driver and its daily content, an identity provider and a help desk. Each part may meet Anderson's test on its own. The whole cannot be analyzed by any one party, because no one party can see all of it. The Trust Ledger is a way to analyze it piece by piece, by the authority that holds each piece.

What hasn't changed: in September 1975 Jerome Saltzer and Michael Schroeder published eight design principles in the *Proceedings of the IEEE*. Two of them are still the test this book applies to every row. **Least privilege:** "every program and every user of the system should operate using the least set of privileges necessary to complete the job". **Fail-safe defaults:** "base access decisions on permission rather than exclusion". A content channel that can change kernel behavior on every desk at once, with no staging, strains the first; a forbidden list that must be updated before an old boot manager stops being trusted is the second, inverted. Text checked against the paper, 18 November 2026.

THE UNSEEN ANGLE

Security inventories count assets: devices, applications, accounts. An asset inventory does not record **authorities**: the parties that can change those assets without you. Count them for one system, and mark which ones you would see acting. The number is not on any dashboard, because each vendor's console shows only its own channel. It is also the number a board can understand: "four outside parties can change every branch desk tonight without a ticket" is a sentence, where "2,500 endpoints with an average patch age of 11 days" is a statistic. Source: the method of this chapter; the update paths it counts are documented by each vendor, as listed in this chapter's sources.

> *Security is not a wall around your systems. It is a list of everyone who can change them, and how fast each of them can be stopped.*

These rows in the Trust Ledger

The worked example is Halbrook Bank's branch teller desktop. The header block comes first, then three rows. Clock cells marked "to measure" are placeholders until Halbrook's own telemetry fills them; the first pass exists to find them.

Header field	Halbrook branch teller desktop
System and service	Teller desktop: payments, cash, account servicing in 2,500 branch seats
Platform profile	Windows 11 Enterprise, current feature release, on business PCs with TPM 2.0, Secure Boot and VBS; Intune-managed; BitLocker on (Chapter Two)
Fleet denominator	2,500 desktops
Value and cost of a day down	Branch revenue and fees for one day, plus overtime and customer remediation (Chapter Fourteen models it)
Regulator clocks	Banking agencies: 36 hours after determining a notification incident; SEC Form 8-K Item 1.05: four business days after determining materiality; DORA for the Irish subsidiary only
Loss protected against	Branches unable to serve customers for more than one business day; fraudulent payment approval
Assessed / next review	November 2026 / quarterly
Accountable executive	Chief Operating Officer, with the CISO

Component	Who can change it	Exposure and blast radius	Patch clock	Stop, revoke, recover	Who can see it	Support end	Decision and owner
MU Windows Update, Intune, EDR content, PC maker tool, teller app updater	Microsoft; Halbrook endpoint team; EDR vendor; PC maker; app vendor	Each channel reaches all 2,500 desks; console sign-in is the way in	Windows: T1 Patch Tuesday, T3 to measure (O); others nc	Rings pause Windows Update; EDR content pause untested; app updater: none	O for Windows and Intune; D for EDR; U for app updater	Per product lifecycle; app vendor: U	**Own** the ring schedule across all five channels; **Isolate** content by branch region. Owner: head of end-user computing; review quarterly
DR EDR sensor driver and content	EDR vendor writes, builds, publishes and pushes; Microsoft signs the driver	Kernel privilege; one content push reaches every desk	Per vendor; nc	Vendor revert; Halbrook hand recovery needs BitLocker keys; last drill: none	D (vendor console)	Per contract	**Rent**; **Isolate**: staged content by ring, written into the contract. Owner: CISO
ID Teller sign-in, help-desk reset, BitLocker recovery passwords	Identity provider; Halbrook identity team; Halbrook help desk	A phone call to the help desk; reset reaches one teller, recovery-password access reaches every disk	Provider-side fixes: np; reset policy: Halbrook's	Revoke sessions and methods; rotate recovery passwords after use	O (sign-in and reset logs), if reviewed	n/a	**Rent** the provider; **Own** the reset procedure and who may read recovery passwords. Owner: head of identity

SECURITY LENS

Who can change it? For any row: the six granted authorities (writes, builds, signs, publishes, pushes, configures) plus whoever exploits any of them. On Halbrook's desk, twelve parties in all, four of them without a Halbrook ticket.

How fast can it be fixed? T0 to T1 is the slowest authority's clock; T1 to T3 is yours. You inherit the slowest clock on the row and then add your own. Only your telemetry measures the second part.

What does AI change here? Our view (as of November 2026): AI-assisted vulnerability discovery and exploit writing move T0 earlier and shrink the time between a public fix and a working attack; they do not shorten T1 to T3 unless the customer automates its own rollout. The gap this chapter measures is the gap AI widens.

BUSINESS IMPLICATIONS

Security. Halbrook's exposure on any row lasts until T3 on its own fleet, whatever the vendor's release date says. Corvane's kiosks, spread over 300 properties with no engineer on site, will have a longer tail than Halbrook's branches, and the tail is where Corvane's exposure lasts longest.

Cost. Every outside authority is a contract or a license, and a channel to watch. Larkfield's 400,000 desktops make Larkfield an authority in thousands of other companies' ledgers, and the Cyber Resilience Act now gives its customers' regulators a way to ask about its clock.

Visibility. Many rows below the kernel report little that a customer can use. Mark them U instead of guessing; a ledger with honest U marks shows where to spend on telemetry.

Value. Recovery time is what customers notice. A row with a tested stop and a measured R90 is worth more to the business than a row with one more control.

Monday rule: for your most critical system, list every authority that can change it, mark the ones that need no ticket from you, and put a measured T3 on one row.

OUR VIEW, AS OF NOVEMBER 2026

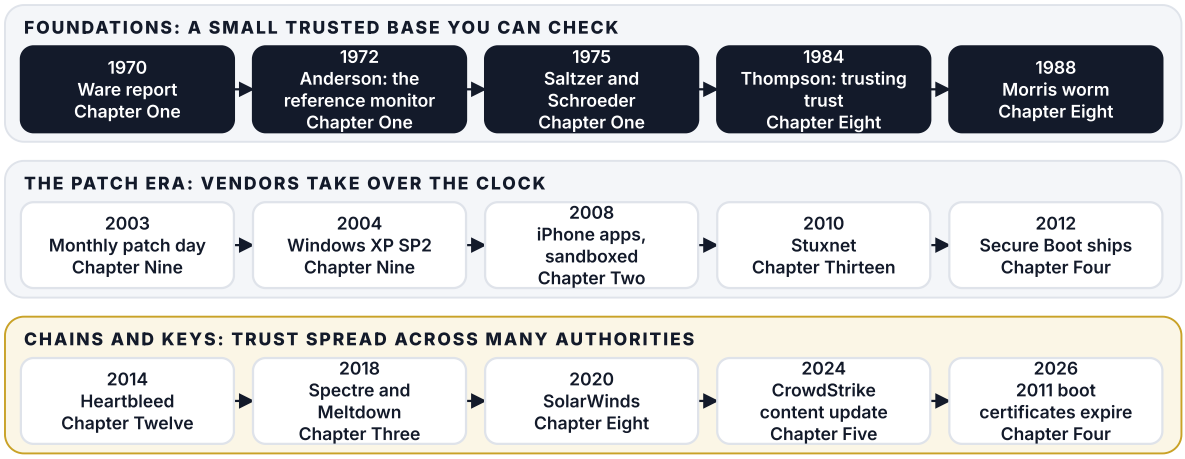
Regulators are moving from "patch everything within N days" to clocks set by exposure and exploitation, as CISA's BOD 26-04 did in June 2026. We expect auditors to follow, and to ask for measured T3 and R90 per critical system within a few years. Companies that keep a Trust Ledger will be able to answer from their own data; companies that do not will answer with vendor release dates, which measure someone else's fleet.

The history spine, 1970 to 2026

The rest of the book tells history where it explains a decision, in the "How it got this way" box of the chapter that needs it. This spine puts the fifteen dates in one place. Read it as the story of one shift: from a small trusted base that one team could check, to a trusted base spread across a dozen authorities, each with its own clock.

FIGURE 1.4

The history spine, 1970 to 2026: from one checkable trusted base to a dozen authorities with clocks



Fifteen dated points; each points to the chapter whose history box tells it.

The primary document for each point is listed in the sources for Chapter One.

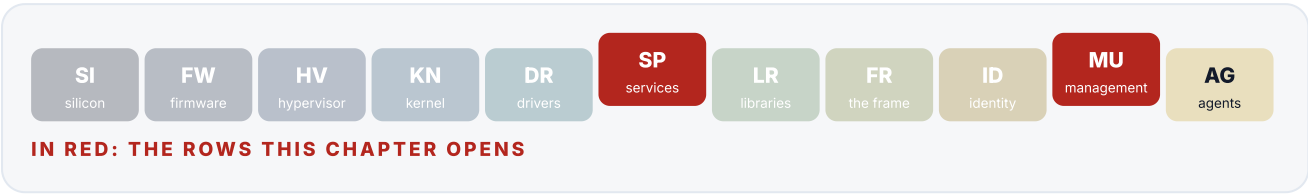
Date	Event	Why it matters to a Trust Ledger	Told in
Feb 1970	Ware report, *Security Controls for Computer Systems* (RAND R-609); declassified 10 Oct 1975	Security as a property of the whole system: hardware, software and procedure	Chapter One
Oct 1972	Anderson, *Computer Security Technology Planning Study* (ESD-TR-73-51)	The reference monitor: tamper proof, always invoked, small enough to verify	Chapter One
Sept 1975	Saltzer and Schroeder, "The Protection of Information in Computer Systems"	Least privilege and fail-safe defaults: still the test for every row	Chapter One
Aug 1984	Thompson, "Reflections on Trusting Trust" (*Communications of the ACM*)	"You can't trust code that you did not totally create yourself": the build as an authority	Chapter Eight

2 Nov 1988	The Morris worm	One program spreads across the internet in a night: a recovery clock measured across many owners at once	Chapter Eight
Oct 2003	Microsoft moves to a monthly security release	The vendor's patch clock becomes a calendar that customers plan around	Chapter Nine
6 Aug 2004	Windows XP Service Pack 2 released to manufacturing	Security defaults shipped in a service pack: the vendor changes the platform under the customer	Chapter Nine
10 Jul 2008	Apple's App Store opens; every third-party iPhone app sandboxed	One vendor holds the store, the review and the sandbox	Chapter Two
Jul 2010	Stuxnet becomes public; Microsoft Security Advisory 2286198 (16 Jul 2010)	Signed drivers with stolen certificates: a signature proves who, not what	Chapter Thirteen
26 Oct 2012	Windows 8 goes on sale; certified PCs ship with UEFI Secure Boot on	Boot keys become a row, with holders and, later, end dates	Chapter Four
7 Apr 2014	Heartbleed (CVE-2014-0160), OpenSSL advisory	A library used across the internet, in a flaw in one small feature: who pays to maintain it	Chapter Twelve
3 Jan 2018	Spectre and Meltdown made public, ahead of the 9 Jan coordinated date	A flaw in the chip, fixed by microcode and software from several parties	Chapter Three

13 Dec 2020	SolarWinds SUNBURST disclosed; CISA Emergency Directive 21-01	A signed update from a trusted build system as the way in	Chapter Eight
19 Jul 2024	CrowdStrike content update; Microsoft estimates about 8.5 million Windows devices affected	One content channel stops machines below the reach of remote tools	Chapter Five
Jun to Oct 2026	Microsoft's 2011 Secure Boot certificates expire: KEK CA 24 Jun, UEFI CA 27 Jun, Windows Production PCA 19 Oct	Keys with end dates: devices still boot, but new boot-level protections need the 2023 certificates	Chapter Four

CHAPTER TWO

Five Platforms, Five Trust Models



In September 2026, the platforms under Halbrook Bank changed three times without Halbrook doing anything. On 14 September Apple released iOS 27, the release its advisors' supervised iPhones would be offered, and macOS 27. On 29 September Microsoft made Windows 11 version 26H2, the next release for its branch desktops, generally available. On 30 September Google began requiring that apps installed on certified Android devices in Brazil, Indonesia, Singapore and Thailand come from verified developers, a rule that will reach Halbrook's own customer app when the global rollout planned for 2027 arrives. Each change came from a different authority, through a different channel, under a different set of rules about what Halbrook was allowed to delay or refuse. This chapter draws those rules, one platform at a time, in the same grammar. The decision it changes: you stop comparing platforms by features and start comparing them by trust model, meaning who holds the keys, who pushes changes, and what the customer is allowed to change.

First, say exactly which platform you mean

"Windows" and "Android" are not configurations. A Pixel phone and a low-cost tablet both run Android, and they have different authorities on almost every row. A Linux server under a support contract and a community Linux laptop share a kernel and little else. So every Trust Ledger in this book names a **platform profile** in its header: the exact release, the hardware class and the management state. The profiles below are fixed for the whole book, as of November 2026.

Platform	Profile used in this book	Why this profile
Windows	Windows 11 Enterprise, version 24H2 or later (25H2 and 26H2 install as enablement packages on the same base), on a current business PC with TPM 2.0, Secure Boot and VBS available; Windows 11 IoT Enterprise LTSC 2024 for kiosks	The enterprise desktop and the long-life device
macOS	macOS 27 on a Mac with Apple silicon, enrolled in mobile device management (MDM)	The current release on the current hardware family
iOS	iOS 27 on a supported iPhone; supervised when company-owned	The phone a bank's advisors and customers carry
Linux	Two profiles: **enterprise** (Red Hat Enterprise Linux 10 or Ubuntu 24.04 LTS, Secure Boot on through shim) and **community** (for example Debian or Fedora, no support contract)	The contract changes the clock more than the kernel does
Android	Android 17 on three profiles: **Pixel** (Google is the maker), **large-maker flagship**, **low-cost device**	The maker decides the clock

Each trust map in this chapter has the same parts. On the left, **who holds the keys**. In the middle, **the chain** from apps at the top to the boot chain at the bottom, with dark boxes for layers the platform vendor controls, gold for third parties, white for the customer. On the right, **who pushes changes**. Underneath, **firmware and silicon**, drawn as a dependency, not a sixth platform. At the bottom, in gold, **the customer's column**: what you may change. The maps are full-page; in the web edition each opens full screen.

Windows: the open PC and its many signers

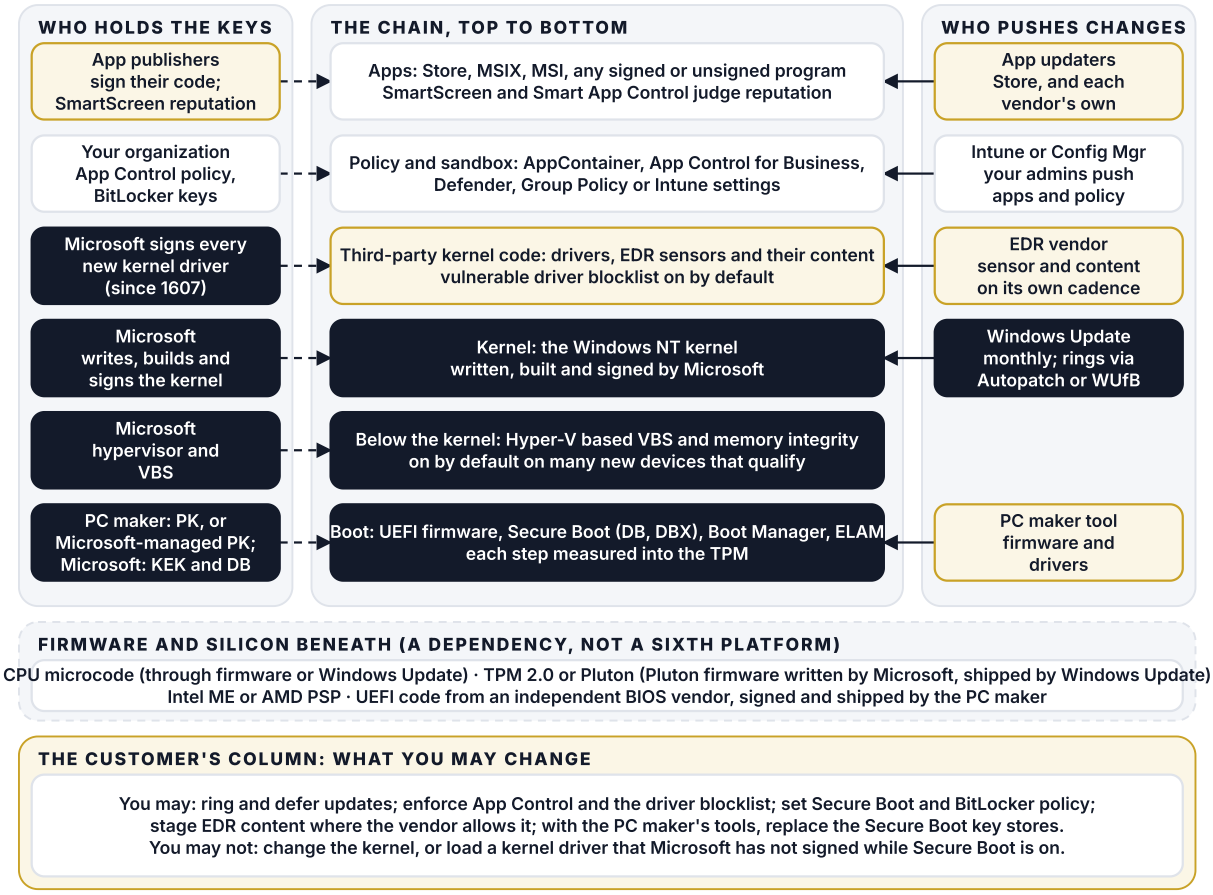
Windows inherits the open PC: many hardware makers, many driver writers, many software publishers, one operating-system vendor in the middle. Its trust model reflects that (Figure 2.1). At the bottom, the PC maker normally holds the **Platform Key**, and Microsoft's key sits in the **Key Exchange Key** list. Microsoft's own guidance lets a maker either manage the PK or use a Microsoft-managed PK, which Microsoft calls "the preferred PK for the Windows ecosystem". The allowed list, DB, carries the Windows Production PCA 2011 certificate that signs Windows' boot manager and the Microsoft UEFI CA 2011 that signs third-party boot code such as Linux's shim and the option ROMs on add-in cards. On Secured-core PCs, Microsoft's specification says the third-party UEFI CA is "not trusted by default". The 2023 successors, Windows UEFI CA 2023, Microsoft UEFI CA 2023, Microsoft Option ROM UEFI CA 2023 and Microsoft Corporation KEK 2K CA 2023, are the subject of Chapter Four.

Above the boot chain, Microsoft holds the kernel and, on capable hardware, the hypervisor that runs virtualization-based security. The door into the kernel for everyone else is Microsoft's signature: "Starting with Windows 10, version 1607, Windows will not load any new kernel-mode drivers which are not signed by the Dev Portal". The exceptions Microsoft lists are narrow: machines upgraded from earlier releases, machines with Secure Boot off, and drivers signed with certificates issued before 29 July 2015. Since the Windows 11 2022 update, Microsoft's vulnerable driver blocklist is on by default for all devices. That signature is a statement about who submitted a driver. It is not a statement about the content the driver will later read, which is why third-party security sensors are drawn in gold.

Windows usually carries many push channels, because the open PC has many authorities: Windows Update, the PC maker's firmware and driver tool, the EDR vendor, the customer's own Intune or Configuration Manager, and every application's updater. The customer's power is correspondingly large: rings and deferrals for Windows Update, App Control for Business to decide what may run, BitLocker and Secure Boot policy, and, with the PC maker's tools, the right to replace the key stores entirely.

FIGURE 2.1

Windows: an open chain with many signers, and Microsoft's signature at the kernel door



On Windows the PC maker, Microsoft and third-party driver and security vendors each hold a link, and the customer owns the policy that ties them together.

Author's diagram based on Microsoft Learn: Secure Boot key creation and management guidance; kernel-mode code signing policy; memory integrity; Microsoft recommended driver block rules; Pluton; KB5062710 (all checked 18 November 2026).

macOS and iOS: one vendor from the chip up

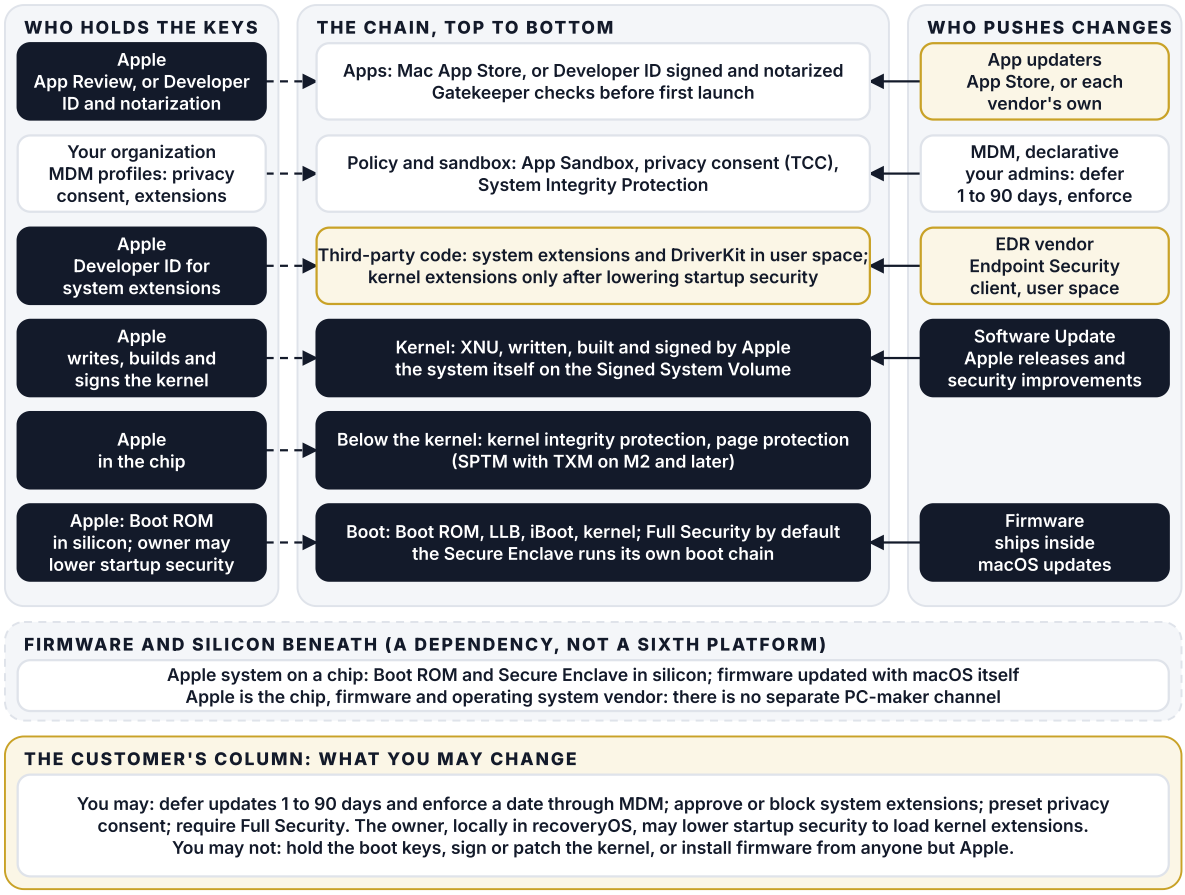
Apple designs the chip and writes the firmware and the operating system, and it runs the main store. On a Mac with Apple silicon, the chain of trust starts in silicon: the Apple Platform Security guide describes the Boot ROM as "the first step in the chain of trust", followed by the Low-Level Bootloader (LLB) and iBoot, which load the kernel. The Secure Enclave runs its own boot chain from its own Boot ROM. The operating system itself lives on the **Signed System Volume**, introduced in macOS 11, and the guide documents kernel integrity protection and, on chips from M2 and A15 onward, a Secure Page Table Monitor

that works with a Trusted Execution Monitor to protect the kernel's page tables. On iPhone and iPad it "replaces the PPL", the page protection layer of earlier chips, which Apple never used in macOS.

The Mac keeps one door the iPhone does not: the owner can lower **startup security**. Full Security is the default; third-party kernel extensions load only after the owner selects Reduced Security in recoveryOS and approves them. Since macOS 10.15 Catalina in 2019, Apple has moved third-party drivers and security tools to **system extensions** and DriverKit, which run "in user space rather than at the kernel level". By default, Gatekeeper opens software from outside the Mac App Store only if it is signed by a registered developer and notarized by Apple; a user or an MDM policy can make exceptions (Figure 2.2).

FIGURE 2.2

macOS on Apple silicon: one vendor from the boot ROM to the kernel, third parties moved to user space



Apple holds every key below the apps; the customer's power is MDM policy and the owner's local choice to lower startup security.

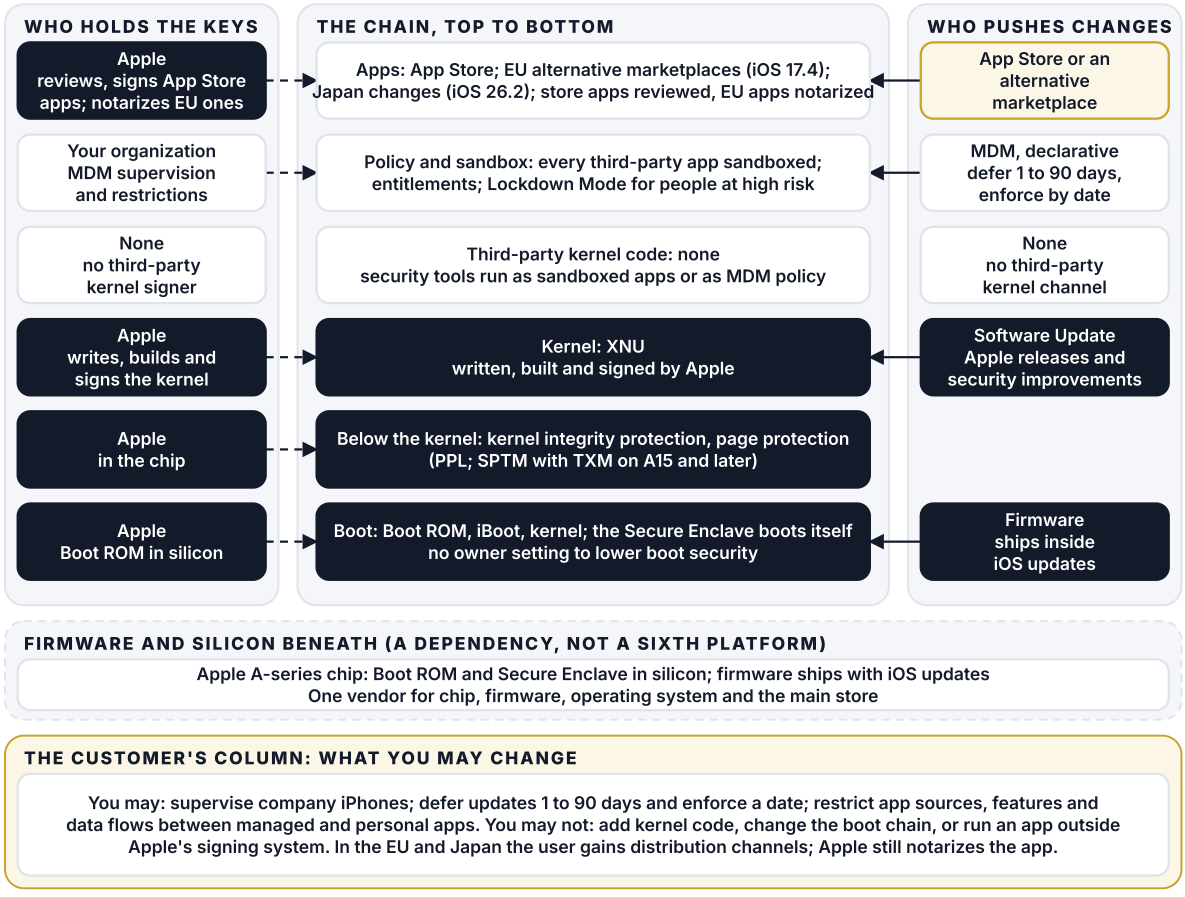
Author's diagram based on the Apple Platform Security guide (boot process for a Mac with Apple silicon; startup security; system extensions; kernel integrity protection; Signed System Volume) and Apple Platform Deployment, Software Update declarative configuration (checked 18 November 2026).

The iPhone closes the remaining doors (Figure 2.3). There is no third-party kernel code and no owner setting to lower boot security. "All third-party apps are sandboxed", in the words of Apple's guide. Since iOS 17.4 in March 2024, users in the European Union can install apps from alternative marketplaces under the EU's Digital Markets Act, and since iOS 26.2 Apple has made further changes in Japan under that country's Mobile Software Competition Act. Those changes add distribution channels; they do not add signers. For apps in the EU, Apple states that its notarization "applies to all apps, regardless of their distribution channel".

Both platforms push security fixes through Apple's Software Update, and since iOS 26.1, iPadOS 26.1 and macOS 26.1 also through **Background Security Improvements**, small security fixes delivered between releases. The first shipped on 17 March 2026. The customer's power on both is MDM policy: Apple's declarative software-update settings let an organization defer updates by 1 to 90 days and enforce a deadline; supervised iPhones accept tighter restrictions; Lockdown Mode, available since iOS 16 and macOS Ventura, narrows the attack surface for people at high risk.

FIGURE 2.3

iOS: the shortest chain, no third-party kernel code, and almost every key in one hand



Apple reviews and signs App Store apps and notarizes EU alternative distribution; the customer's power is supervision, deferral and restriction.

Author's diagram based on the Apple Platform Security guide (boot process; app security; kernel integrity protection), Apple Developer, DMA and apps in the EU, Apple Newsroom (Japan, 17 December 2025) and Apple Platform Deployment (checked 18 November 2026).

Linux: the community, the distribution and the owner

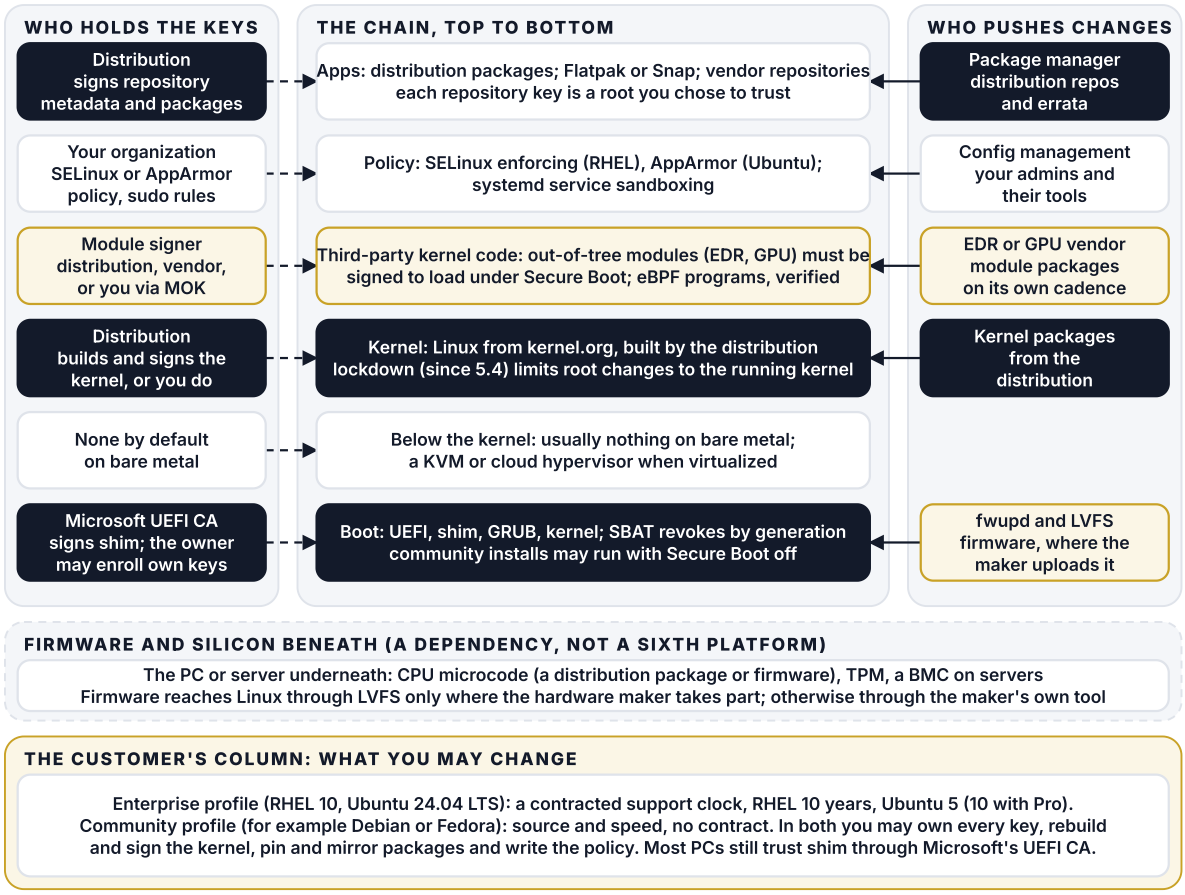
Linux has no single platform vendor. The kernel comes from kernel.org; a **distribution** builds, signs and supports it with thousands of packages; the owner can replace any of it. On a PC with Secure Boot, the first Linux code to run is **shim**, which the shim project describes as "a first-stage UEFI bootloader" that checks the next stage "against a built-in certificate", the distribution's. Shim itself is trusted because Microsoft's UEFI CA signed it. The owner can enroll additional keys as **Machine Owner Keys** (MOK). Revocation moved from lists of hashes to generation numbers with **SBAT**, first shipped in shim 15.3 in March 2021 after the BootHole flaw showed how slow hash revocation was (Chapter Four).

The kernel's **lockdown** feature, merged in Linux 5.4 in November 2019, limits how far even the root user can change the running kernel. On the enterprise profiles in this book, Red Hat Enterprise Linux and Ubuntu, third-party kernel modules, from EDR sensors to GPU drivers, must be signed to load when Secure Boot is on; that is the distribution's policy, not a rule of UEFI itself. Policy is the distribution's choice: Red Hat Enterprise Linux runs SELinux enforcing by default on new installs, while Ubuntu loads AppArmor by default. Both confine what a program may do once it runs; deciding which programs may run at all is a separate tool, such as fapolicyd, Red Hat's application allowlisting framework, which by default trusts what the RPM package manager installed.

The two Linux profiles differ more in their clocks than in their code (Figure 2.4). The **enterprise profile** rents a support clock: Red Hat documents a ten-year Full and Maintenance Support lifecycle for each major release, and Canonical gives Ubuntu 24.04 LTS standard security maintenance to May 2029, expanded maintenance with Ubuntu Pro to May 2034, and a Legacy add-on beyond that. The **community profile** has source and speed, and no one under contract. In both, the owner may hold every key, from the Platform Key to the module-signing key, and that is the real distinction: on Linux, you can own the rows other platforms only let you configure, and then you own their clocks too.

FIGURE 2.4

Linux: two profiles, one rule: you may hold every key, and then you hold every clock



The enterprise profile rents a support clock from the distribution; the community profile trades the contract for source and speed; both can be owned down to the boot keys.

Author's diagram based on the shim README and SBAT.md (rhboot/shim), Microsoft Secure Boot key guidance, Linux kernel lockdown (5.4), Red Hat and Ubuntu lifecycle and security documentation, and fwupd.org (checked 18 November 2026).

Android: open source, shipped by many makers

Android's code is open, its kernel is Linux, and its biggest authority on any one device is usually neither Google nor the customer. It is the **device maker**, which holds the boot keys, builds the image, adds the chip maker's drivers and decides when updates ship (Figure 2.5). **Android Verified Boot**, part of the platform since Android 8.0, checks each partition from "a hardware-protected root of trust to the bootloader" onward; an unlocked device shows the orange boot state. Since Android 12, devices launching with kernel 5.10 or later use Google's **Generic Kernel Image**, which moves chip and board support "out of the core kernel into loadable vendor modules". Google's certified monthly GKI releases carry a

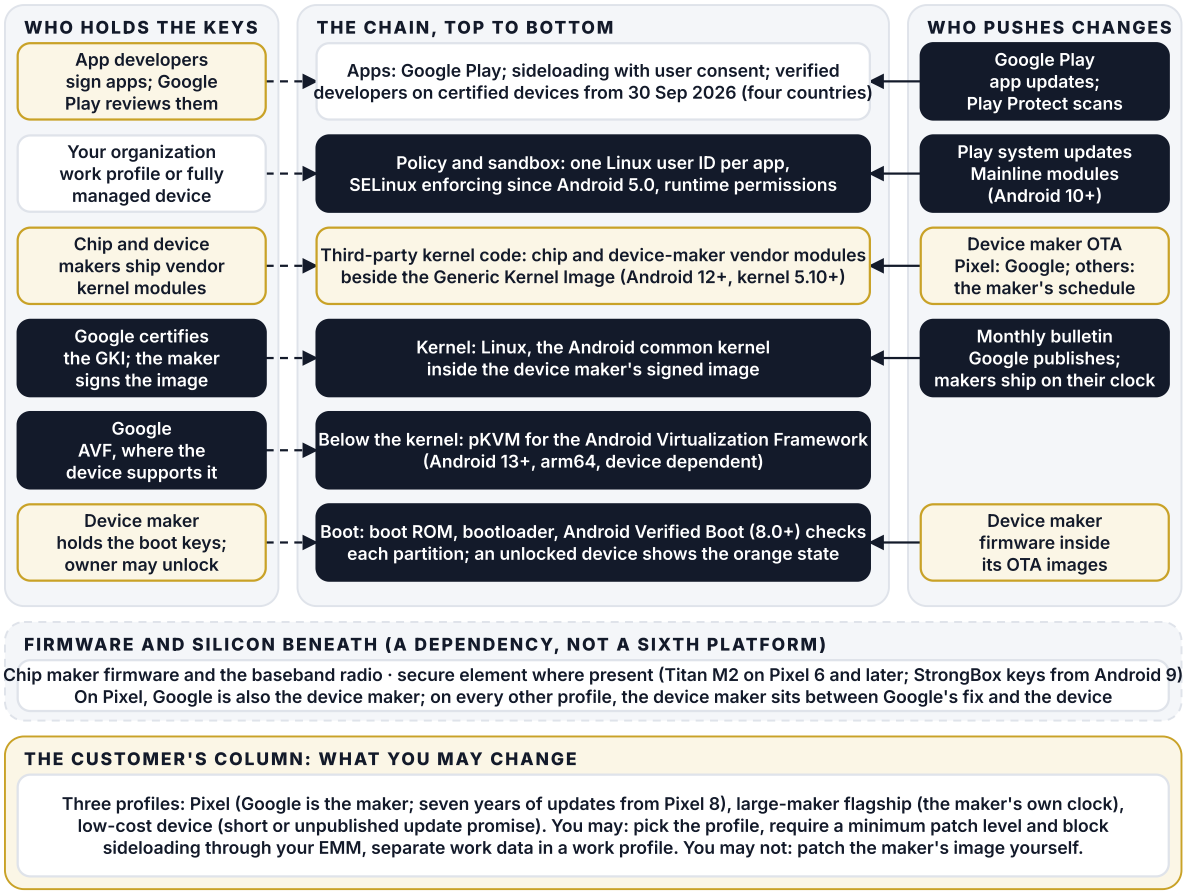
"Google inserted certificate" attesting that they were built from a known source baseline; the maker adds its modules and signs the device image under its own Verified Boot key. Where the hardware supports it, the Android Virtualization Framework, introduced in Android 13, runs a protected hypervisor, pKVM, below the kernel.

Above the kernel, Android isolates apps with one Linux user ID per app, "a kernel-level Application Sandbox", and SELinux has been fully enforcing since Android 5.0. Since Android 10, Google can update core system components directly as **Mainline** modules through Google Play system updates. Google publishes a security bulletin each month and notifies its Android partners of the issues "at least a month before publishing". From there, each maker runs its own clock.

So the Android profile you buy decides the clock you get. On a **Pixel**, Google is the maker, and since the Pixel 8 in October 2023 has promised seven years of OS and security updates. A **large-maker flagship** runs on that maker's published promise. A **low-cost device** may have a short promise or none published. The customer's power is the profile choice plus management through an enterprise mobility management (EMM) console: a work profile or a fully managed device, a minimum patch level for access, a block on sideloading, and an update policy that can postpone a system update once for up to 30 days or freeze updates for periods of up to 90 days. Android's own documentation notes that the 30-day postponement does not hold back monthly security patches, and that freeze periods do. Since 30 September 2026, apps installed on certified devices in the four first countries must come from developers verified with Google, with a global rollout planned for 2027; unregistered apps can still be installed through the Android Debug Bridge or an "advanced flow".

FIGURE 2.5

Android: three profiles, one platform, and the device maker standing between Google's fix and your device



Google writes the fix and publishes the bulletin, but the device maker signs the image and sets the clock, so the profile you buy decides the clock you get.

Author's diagram based on source.android.com (Verified Boot, app sandbox, SELinux, Generic Kernel Image, Android Virtualization Framework, Mainline, security bulletins), Android Developers (developer verification, Keystore) and Google's Pixel 8 announcement (checked 18 November 2026).

The customer's column

Most platform comparisons score features. This one scores what you are allowed to change, because that decides which rows you can own at all.

The customer may	Windows	macOS	iOS	Linux	Android
Hold or replace the boot keys	Yes, with the PC maker's tools	Not Apple's root; at Permissive Security the owner can boot a locally signed kernel	No	Yes: MOK, or your own PK	Only by unlocking, which changes the boot state
Revoke boot code	DBX updates (Microsoft), or your own	Apple only	Apple only	SBAT and DBX; your own keys	Device maker only
Run third-party kernel code	Yes, Microsoft-signed; blocklist on	Only at Reduced Security, set by the owner locally	No	Yes, signed modules	Maker's choice
Decide what apps may run	App Control for Business	MDM and Gatekeeper policy	MDM restrictions; Apple reviews and signs App Store apps, and EU alternative distribution is notarized	Your repositories; fapolicyd allowlisting on RHEL	EMM policy; sideloading block on managed devices
Delay OS updates	Rings and deferrals	1 to 90 days, with a deadline	1 to 90 days, with a deadline	Indefinitely: you run the repositories	One 30-day postponement per update, or freeze periods up to 90 days
Stage third-party content	Where the EDR vendor allows	Where the vendor allows	Not applicable	Where the vendor allows	Not applicable
Patch the platform yourself	No	No	No	Yes	No
Firmware and silicon	Choose PC maker and model; stage its firmware tool	Take Apple's firmware with macOS	Take Apple's firmware with iOS	Use LVFS where the maker takes part	Choose the maker; its images carry the firmware

HOW IT GOT THIS WAY

Each trust model is the shadow of a business model. The **open PC** let any maker build hardware and any company write drivers and software; Windows had to trust many signers, and the customer received the most configuration power, along with the most authorities to watch. **Apple** integrated chip, firmware, operating system and store, so it could close doors the PC had to leave open. The App Store opened on 10 July 2008 with every third-party app sandboxed, and since 2019 Apple has moved most third-party drivers and security tools on the Mac out of the kernel into user space. **Linux** grew as a community kernel shipped by distributions, so trust follows the distribution's keys and the owner's choices, and support became a product sold separately from the code. **Android** took the open-source road through many device makers, so the platform vendor writes the fix and a separate company signs and ships it; Generic Kernel Images and Mainline modules are Google's attempts to shorten that second step.

What hasn't changed: the more keys one vendor holds, the less the customer can change, and the fewer parties a fix must pass through on its way to a device. That is a statement about path length, not about speed, and it is measured per case. On 17 March 2026 Apple shipped its first Background Security Improvement, a WebKit fix delivered between releases; Google, by its own rule, tells Android makers of each month's issues at least a month before the bulletin, and when each maker's fix reaches its devices is that maker's clock, often not published. Checked 18 November 2026.

THE UNSEEN ANGLE

The trust model is part of what you buy, and it is rarely in the purchase decision. A tablet's price is on the quote; its update promise, the profile that decides its patch clock and support end, is on a support page that may not exist for a low-cost model. Before buying any device class, write its platform profile and fill the customer's column for it. If a cell reads "the maker decides" and the maker has published nothing, the ledger entry is "np", and the price is not the whole cost. One way to get the promise in writing is Google's Android Enterprise Recommended program: makers of listed knowledge-worker devices commit to 90-day security updates and must publish "the end date when the device is guaranteed to receive security updates". Source: the customer's column above; Android Developers, SystemUpdatePolicy; Apple Platform Deployment, Software Update declarative configuration; Google, Android Enterprise Recommended devices.

> *Compare platforms by what you are allowed to change, not by what they can do. The first decides what you can own.*

These rows in the Trust Ledger

This chapter adds no new rows; it fills the platform-profile field of each company's header. The worked example is Corvane Hotels' mixed fleet, where one management row splits into three because each platform has its own authorities and clock.

Company	Platform profiles in the header
Halbrook Bank	Branch desktops: Windows 11 Enterprise, Intune-managed. Advisor phones: iOS 27, supervised. Customer app: iOS and Android, unmanaged (the customer's device). Advisor desktops: virtual desktops on a rented cloud hypervisor (HV row, Chapter Three)
Corvane Hotels	Kiosks: Windows 11 IoT Enterprise LTSC 2024. Front-desk tablets: Android, low-cost profile, fully managed. Signage: Linux, enterprise profile (Ubuntu 24.04 LTS)
Larkfield Software	Its customers' desktops: all five profiles, not managed by Larkfield. Its own build machines: Linux, enterprise profile (Chapter Eight)

Component	Who can change it	Exposure and blast radius	Patch clock	Stop, revoke, recover	Who can see it	Support end	Decision and owner
MU kiosks: Windows Update for IoT Enterprise LTSC 2024; PC maker tool; EDR content	Microsoft; kiosk maker; EDR vendor; Corvane endpoint team	About 3,000 kiosks at 300 properties	T1 Patch Tuesday; T3 to measure (O)	Rings by property group; field visit if a kiosk will not start	O (Intune reports)	Microsoft: 10 Oct 2034	**Rent** the OS; **Own** the rings; **Pin** the LTSC release to 2034. Owner: head of property IT
MU front-desk tablets: maker OTA, Play system updates, EMM	Device maker; Google; Corvane via EMM	One tablet per desk; customer data on screen	Bulletin monthly; maker's T1: np	Postpone 30 days; freeze up to 90; no rollback	O (EMM patch level)	Maker's promise: np	**Rent**; replace with a profile that publishes an update promise at next refresh. Owner: head of property IT
MU signage: Ubuntu archive, Landscape or config tool, fwupd	Canonical; Corvane; hardware maker through LVFS where it takes part	Public screens in every lobby	Ubuntu notices; T3 to measure	Corvane pins and mirrors packages; rebuild from image	O (package versions)	Standard May 2029; Pro May 2034	**Own** the mirror and schedule; **Isolate** signage from the kiosk and payment networks. Owner: signage lead

SECURITY LENS

Who can change it? On Windows, many signers and many channels, with the most configuration left to you. On macOS and iOS, Apple below the apps. On Linux, the distribution, or you. On Android, Google, the chip maker and, above all, the device maker.

How fast can it be fixed? The number of parties between the fix and the device sets the path; the customer's own rollout sets the rest. A mixed fleet runs at the speed of its slowest profile, which for Corvane is the low-cost tablet whose maker publishes no clock.

What does AI change here? Our view (as of November 2026): AI-assisted analysis of each platform's monthly fixes makes the time between a published fix and a working attack shorter on every platform. That penalizes long paths from fix to device, and it makes the Android maker's step and the customer's own deferral settings the two places to measure first.

BUSINESS IMPLICATIONS

Security. Control and the patch path trade against each other. Halbrook gets the most control on Windows and the shortest path on iOS; it needs both in its ledger, row by row, rather than a single platform preference.

Cost. A mixed fleet pays for five trust models: Corvane pays for Windows rings, an EMM for Android, a mirror for Linux, and three different support calendars, and it runs at the slowest one's clock. Larkfield pays for all five on its customers' desktops without managing any of them.

Visibility. Each platform exposes different telemetry: Windows event tracing and Intune reports, Apple's MDM inventory and Endpoint Security framework, Android's EMM patch level, Linux package databases and audit logs. Appendix G maps them per row.

Value. The trust model is part of the product. A cheaper tablet with no published update promise carries a cost the quote does not show, and a platform that lets you own the keys is worth paying for only if you will staff the rows you then own.

Monday rule: write the exact platform profile for every device class you run, and fill in the customer's column for each before the next purchase.

OUR VIEW, AS OF NOVEMBER 2026

The five trust models are converging at the top and diverging at the bottom. Apps on every platform are moving toward verified developers and sandboxes: Android's developer verification in 2026 is the latest step. Below the kernel, the gap between platforms where one vendor holds every key and platforms where the customer may hold them will remain, because it follows the business models in the history box. We expect regulators to keep widening app distribution on phones without widening who signs the kernel.

PART II · BELOW THE OS

CHAPTER THREE

Silicon and the Computers Inside the Computer



On 3 January 2018, the day Google's Project Zero published the flaws that became known as Spectre and Meltdown, Microsoft told Azure customers that "the majority of Azure infrastructure has already been updated" and that it would begin "automatically rebooting the remaining impacted VMs starting at 3:30pm PST" that afternoon. The self-service maintenance window "has now ended". Picture Halbrook Bank's advisors on cloud desktops that day: the fix for a flaw in the processor under their sessions was applied by someone else, on someone else's clock, and Halbrook's part was a maintenance notice and, once the self-service window had closed, a reboot it did not schedule. This chapter is about that layer, and it moves silicon from a purchase made once to a Trust Ledger row with owners, a patch clock and a record of which fix is actually running.

Seven months under embargo, one week of patches

Begin with the dates. Project Zero's post says the team "reported this issue to Intel, AMD and ARM on 2017-06-01". It named three variants: bounds check bypass (CVE-2017-5753) and branch target injection (CVE-2017-5715), together called **Spectre**, and rogue data cache load (CVE-2017-5754), called **Meltdown**. The idea behind all three is **speculative execution**: to stay fast, a processor guesses which instructions come next and runs them before it knows the guess is right. Wrong guesses are thrown away, but they leave traces in shared caches, and timing those traces can leak memory the program was never allowed to read. A **side channel** is exactly that: a leak through timing or another physical effect rather than through a broken permission check.

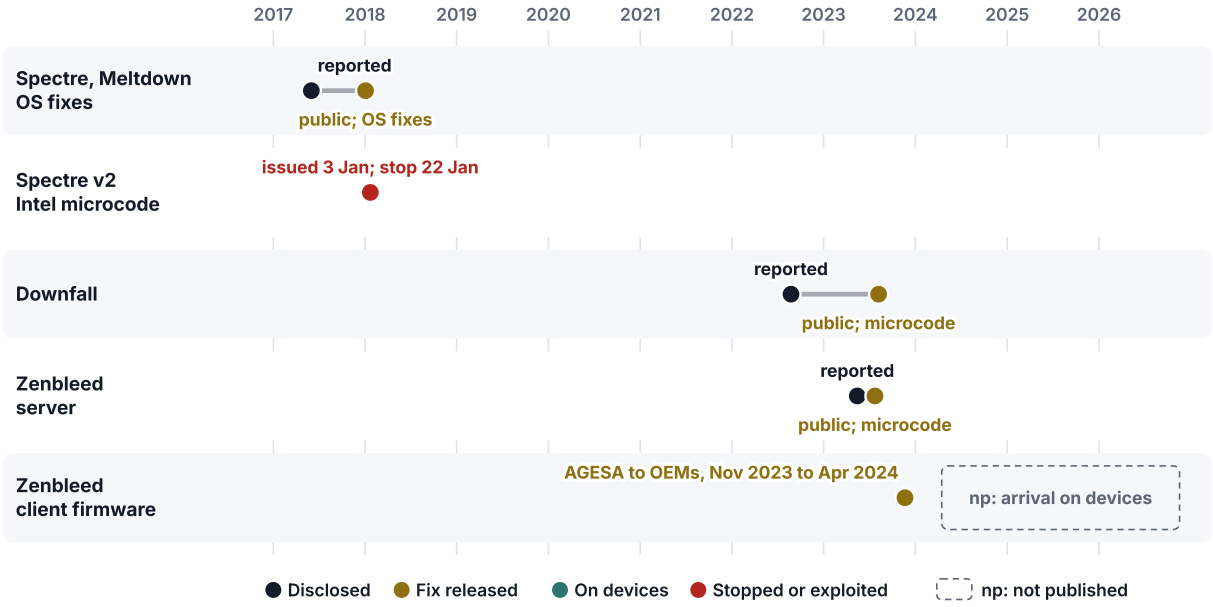
The coordinated plan was to publish on 9 January 2018. Google's security blog explains why that did not happen: "We are posting before an originally coordinated disclosure date of January 9, 2018 because of existing public reports and growing speculation in the press and security research community about the issue." So the public clock started on 3 January.

Several fixes were already in place by then. Apple says it "has already released mitigations in iOS 11.2, macOS 10.13.2, and tvOS 11.2 to help defend against Meltdown"; Apple's security release list dates macOS High Sierra 10.13.2 to 6 December 2017, four weeks before anyone outside the embargo knew why. Linux 4.14.11, released on 2 January 2018, carried kernel page table isolation (KPTI). Microsoft published advisory ADV180002 on 3 January with Windows updates. Apple's Spectre mitigations followed on 8 January in iOS 11.2.2 and a macOS 10.13.2 Supplemental Update.

The hardware half of the fix took longer and went backwards once. Intel's advisory INTEL-SA-00088, released 3 January, included "updated Intel microprocessor microcode to our customers and partners". Three weeks later, Microsoft's KB4078130 records, Intel reported that this microcode could cause "higher than expected reboots and other unpredictable system behavior", and "on January 22, Intel recommended that customers stop deploying the current microcode version". Microsoft then shipped an out-of-band update that turned off only its own CVE-2017-5715 mitigation, so that machines that had taken the faulty microcode would stop rebooting.

FIGURE 3.1

From report to running fix: three CPU flaws, three clocks



Spectre and Meltdown sat under embargo for seven months and Downfall for almost a year; Zenbleed's client firmware reached OEMs up to nine months after disclosure, and its arrival on devices was not published.

Source: Google Project Zero and Google Security Blog (3 Jan 2018); Apple support 101886 and security releases; MSRC ADV180002; Linux 4.14.11 changelog; Microsoft KB4078130; downfall.page and INTEL-SA-00828; AMD-SB-7008; lock.cmpxchg8b.com/zenbleed. Checked 18 Nov 2026.

Two lessons come out of that month and recur through this book. The first is that the operating system fix and the silicon fix are different rows with different owners: Microsoft, Apple and the Linux maintainers could ship kernel changes in days, while the microcode had to pass from the CPU vendor through OEMs before it could run on a device. The second is that a fix can be withdrawn. A fleet that recorded only "patched" on 10 January would have recorded the wrong thing on 23 January. Was any of this exploited? The researchers' own answer, on the Meltdown and Spectre site, is: "We don't know." Book 1, Chapter 4 named Spectre in one line; this chapter is where the clock gets measured.

Downfall and Zenbleed: the same shape, five years later

Two later flaws show that 2018 was not a one-off. **Downfall** (CVE-2022-40982) abuses the vector `Gather` instruction on many Intel processors to read data left in an internal register file by other programs on the same core. Its discoverer, Daniel Moghimi, writes that he "reported this vulnerability to Intel August 24, 2022"; Intel published INTEL-SA-00828 on 8 August 2023, almost a year later. The fix is a microcode update, and Intel states its cost plainly: "Although the performance impact to most workloads is minimal, specific workloads may show performance impacts of up to 50%." Intel adds that the microcode "provides an MSR interface that allows software to opt-out of the mitigation", and on Linux the boot parameter `gather_data_sampling=off` does so. In other words, the owner of the workload can switch the fix off for speed, and somebody has to decide whether they may.

Zenbleed (CVE-2023-20593) leaks register contents across processes on AMD "Zen 2" processors. Tavis Ormandy of Google reported it to AMD on 15 May 2023, and AMD published bulletin AMD-SB-7008 on 24 July 2023. The bulletin's own table is the most useful patch clock in this chapter, because it lists dates per product line. Server microcode for second-generation EPYC "Rome" processors is dated 6 June 2023, before disclosure. The firmware packages that client machines need (AMD's AGESA, which OEMs build into their BIOS) were released to OEMs much later: 21 November 2023 for one Threadripper line, 7 February to 20 March 2024 for desktop parts, 29 February 2024 for "Renoir" mobile chips, and 15 April 2024 for Ryzen Embedded V2000. The bulletin was revised five times after first publication, mostly to move those dates. After AGESA, each OEM must build, test and publish its own BIOS, and when that reached any given laptop or kiosk is not published anywhere in one place.

For a fleet owner, Zenbleed's lesson is the gap between the first two kinds of owner. A data centre running Linux could load AMD's server microcode through the OS the week the bulletin appeared. A fleet of business laptops waited for AGESA, then for its OEM, then for its own BIOS rollout. Same flaw, same vendor, clocks nine months apart. Two details sharpen the gap. The discoverer documented a stopgap anyone could apply at once: setting a "chicken bit", a bit in a model-specific register (MSR) that switches off the faulty

optimization, at some performance cost. And on Windows the microcode updates Microsoft documents (KB4093836) are Intel's; for Zenbleed, AMD's client fix travelled through AGESA and the OEM's BIOS.

Microcode has two owners

Microcode is the processor's own internal program, which the CPU vendor can patch after the chip ships. The patch is not burned in; it is loaded each time the processor starts. It can come from two places, and the difference matters more than most fleet reports show.

- **From firmware.** The BIOS or UEFI firmware carries a microcode patch and applies it early in boot. The OEM decides when a new one ships, as part of a BIOS update.
- **From the operating system.** The Linux kernel documentation describes an early loader that reads microcode from the initial RAM disk (initrd) "very early during boot", and notes that the loader saves the patch so it is reapplied "when CPUs resume from a sleep state". Its stated use cases include "updating the microcode on platforms beyond the OEM End-Of-Life support". Microsoft also ships Intel microcode updates through Windows Update, Windows Server Update Services (WSUS) and the Update Catalog (its KB4093836 lists them). The OS vendor decides when these ship.

Who may sign microcode is itself an authority, and in 2025 it was bypassed. In **EntrySign** (CVE-2024-56161), Google's researchers showed that AMD's microcode signature check used an insecure hash, so that an attacker with local administrator rights could load microcode of their own on Zen 1 to Zen 4 processors; Zen 5 was added to the advisory on 7 April 2025. Google reported it to AMD on 25 September 2024 and published on 3 February 2025. The fix could not take the fast route: Ubuntu's advisory states that an OEM BIOS update is required and that the OS microcode package cannot resolve it alone. Google also warned that the flaw threatened confidential virtual machines (AMD's Secure Encrypted Virtualization, SEV-SNP); tenants could confirm the fix from the firmware version values in their attestation reports.

Whether a fix arrives at all is the vendor's call too. **SinkClose** (CVE-2023-31315) let code with kernel (ring 0) access change the processor's System Management Mode configuration on many AMD chips. AMD's bulletin AMD-SB-7014, first published on 9 August 2024, was revised on 14 August to give Ryzen 3000 desktop chips a mitigation target, on 19 August to say that fix was available, and on 30 October to add mitigations for older parts, including Ryzen 2000 desktop chips. For a fleet, the bulletin's product list is part of the support end date.

The operating system's loader normally applies its patch only if it is newer than the revision already loaded, so the revision actually running depends on the BIOS version, the OS update state and, on Linux, whether the initrd was rebuilt. Two kiosks with the same model number can run different microcode. Some protections care about the path too: for Intel Software Guard Extensions (SGX), its secure-enclave feature, Intel says the Downfall fix attests as up to date only when the patch is loaded from platform flash, "for example, with an updated BIOS".

The kernel's own developers have started to make this visible. A 2025 Linux change (commit `4e2c719782a8`, "x86/cpu: Help users notice when running old Intel microcode") added a file at `/sys/devices/system/cpu/vulnerabilities/old_microcode` that reports when the loaded microcode is older than the newest release the kernel knows. Its commit message is blunt: "it is as important to keep your microcode up to date as it is to keep your kernel up to date." The practical rule for the Trust Ledger follows: the SI row records the **effective microcode revision per device**, read from the running system (on Linux, `/proc/cpuinfo` and that file), not the BIOS version from the purchase record.

When software cannot fix the silicon

Microcode cannot fix everything. Some flaws live in parts of the chip that no update reaches, and then the mitigation is software working around the hardware, or a newer chip.

GoFetch (March 2024) targets the data memory-dependent prefetcher (DMP) in Apple's M-series chips, a unit that fetches memory it predicts a program will need and can be made to treat secret data as addresses. The researchers disclosed it to Apple "on December 5, 2023 (107 days before public release)", and their answer to "Can the DMP be disabled?" is: "Yes, but only on some processors. We observe that the DIT bit set on m3 CPUs effectively disables the DMP. This is not the case for the m1 and m2." DIT (data-independent timing) is a processor mode that cryptographic code can turn on. On the older chips, the fix falls to the authors of each cryptographic library, at a performance cost they choose. The flaw is the same; who carries the fix depends on which chip generation you bought.

Rowhammer is the oldest example in this chapter. Kim and colleagues, from Carnegie Mellon University and Intel Labs, showed at ISCA 2014 that "activating the same row in DRAM corrupts data in nearby rows"; they induced errors "in most DRAM modules (110 out of 129) from three major DRAM manufacturers". Memory makers have added in-chip defences since. In September 2025 the COMSEC group at ETH Zurich published **Phoenix** (CVE-2025-6202), reporting bit flips on all 15 DDR5 modules from SK Hynix that it tested and a privilege-escalation exploit "in default settings on a PC". The researchers were told of a

BIOS update for AMD client machines but "could not independently verify if this update adequately addresses Phoenix". Eleven years on, the fix is still partial, and it still arrives through firmware.

Thunderspy (2020) is the physical-access case. Björn Ruytenberg found seven flaws in Thunderbolt security and wrote that they "cannot be fixed in software" and "will require a silicon redesign". Thunderbolt, USB4 and PCIe ports can let a device read and write the computer's memory directly (DMA, direct memory access). The defence is the **IOMMU**, a unit that limits which memory each device may touch. Microsoft's Kernel DMA Protection uses it to block external peripherals "unless the drivers for these peripherals support memory isolation", and Microsoft notes that the IOMMU must be enabled in firmware for the protection to work. Whether it is on is a firmware setting on each device, which brings us to the other computers in the box.

Roots of trust, and the computers inside the computer

A **hardware root of trust** is the hardware, and the code fixed in it, that the rest of the system's security starts from: it may verify the next boot stage, record measurements, protect keys or support recovery. On the platforms in this book it is usually a small security processor, separate from the main CPU or built into the same chip. Each platform in this book has one, and each answers "who writes its code?" differently.

- **TPM 2.0** is the Trusted Computing Group's standard, built as a separate chip or as firmware inside the processor. Windows 11 lists TPM 2.0 as a minimum requirement. Its firmware comes from the chip or platform maker, through OEM updates.
- **Microsoft Pluton** is "a hardware security processor designed by Microsoft and built into the system-on-chip (SoC) by silicon partners". The notable line is the update path: "Microsoft authors and maintains Pluton firmware, with updates delivered through Windows Update." On boot, Pluton loads firmware from the motherboard's flash; during Windows startup, it can load a newer one from Windows Update. One root of trust, two delivery channels, as with microcode.
- **Apple's Secure Enclave** is "a dedicated secure subsystem integrated into Apple" SoCs. It has its own boot ROM, which Apple describes as "immutable code that establishes the hardware root of trust for the Secure Enclave", and that ROM checks the signature of the Apple-signed sepOS before running it.
- **Google's Titan M2**, introduced with the Pixel 6, is a discrete security chip "fully designed and developed by Google" on an in-house RISC-V processor, working with Trusty, Google's open-source trusted execution environment (TEE).

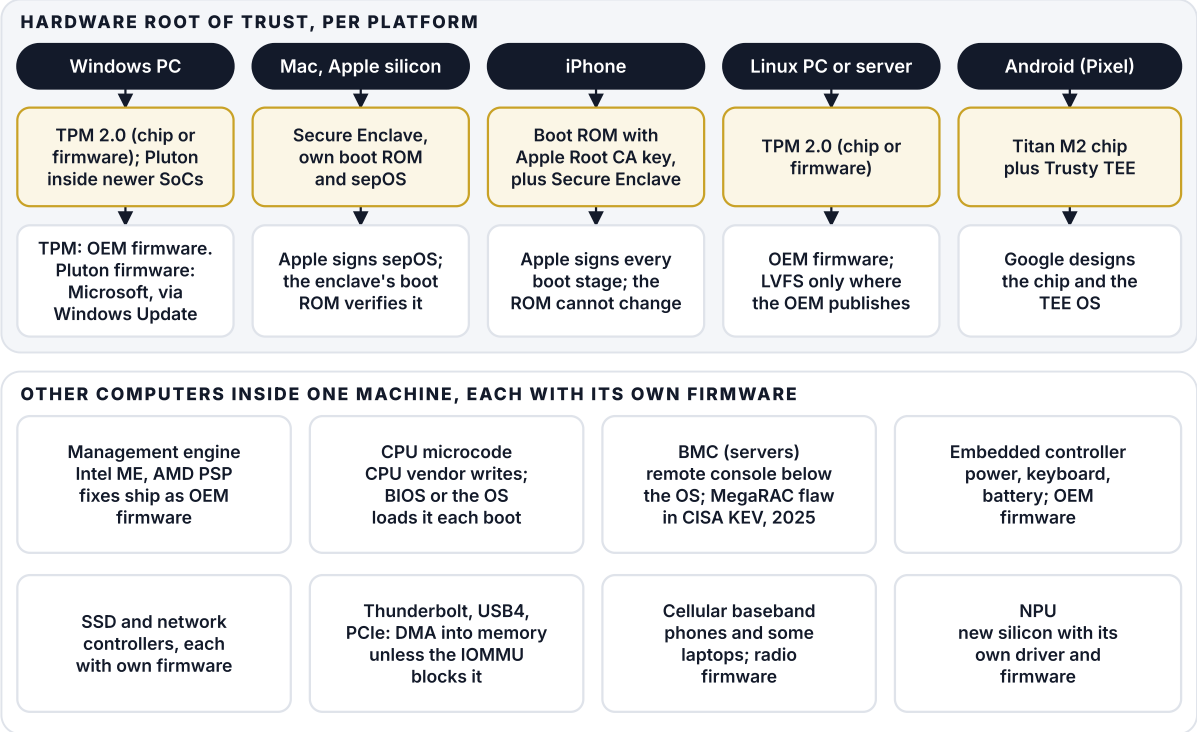
- **Caliptra**, hosted by the CHIPS Alliance, is open-source "IP and firmware for an integrated Root of Trust block" aimed at "datacenter-class SoCs like CPUs, GPUs, DPUs, TPUs". It matters to an enterprise mostly through its cloud provider, which is where the HV row below comes in.

A root of trust protects keys. It does not make the rest of the machine one computer. A business PC also contains several other processors that run their own firmware, below and beside the OS:

- **Management engines.** Intel's Management Engine (ME) and AMD's Platform Security Processor (PSP) run before and alongside the main CPU. Intel's advisory INTEL-SA-00086 (20 November 2017) covered flaws in ME versions 6.x to 11.x and told users to "check with your system manufacturer for updated firmware". The fix owner was the OEM, not Intel and not the OS.
- **Baseboard management controllers (BMCs)** on servers give remote console access below the OS. CISA added CVE-2024-54085, an authentication bypass in AMI's MegaRAC SPx BMC firmware, to its Known Exploited Vulnerabilities catalog on 25 June 2025.
- **Embedded controllers, storage controllers and network controllers**, each with firmware from the OEM or a component supplier.
- **Cellular basebands** in phones and some laptops, which run the radio stack.
- **Neural processing units (NPUs)**, the newest silicon. Microsoft defines Copilot+ PCs by an NPU that "can perform more than 40 trillion operations per second (TOPS)". An NPU brings its own driver and firmware, and so its own row.

FIGURE 3.2

Each platform's main root of trust, and several other computers in the same box



Each platform anchors trust in a small security processor, but the same machine also runs a management engine, microcode, device controllers and DMA-capable ports, each with its own firmware and its own update owner.

Author's diagram based on Microsoft Learn (Pluton, Kernel DMA Protection), Apple Platform Security Guide (Secure Enclave, boot process), Google Security Blog (Pixel 6, Oct 2021), TCG TPM 2.0, Intel INTEL-SA-00086, CISA KEV (CVE-2024-54085), checked 18 Nov 2026.

One kiosk can hold five or more processors with separately updated firmware, from at least three authorities, and a fleet report shows only the ones someone chose to collect. Chapter Four follows the firmware that boots the main CPU; Chapter Ten shows a real attack chain that ended at undocumented chip registers; this chapter's job is to put every one of these computers on the inventory.

Side by side: silicon across five platforms

Only the differences that change a decision are listed. Platform profiles are those fixed in Chapter Two.

Decision point	Windows	macOS	iOS	Linux	Android
Root of trust	TPM 2.0 (required for Windows 11); Pluton on supported AMD, Intel, Qualcomm and NVIDIA chips	Secure Enclave with its own boot ROM	Boot ROM plus Secure Enclave	TPM 2.0 where the hardware has one	Varies by OEM; Pixel: Titan M2 and Trusty TEE
Who writes root-of-trust firmware	TPM: chip or platform maker via OEM; Pluton: Microsoft via Windows Update	Apple	Apple	Chip or platform maker via OEM	OEM or SoC vendor; Google on Pixel
Microcode or CPU fix path	OEM BIOS, or Windows Update	Apple; chip and OS from one vendor	Apple	OEM BIOS, or the kernel's early loader from the distribution's package	SoC vendor to OEM to the device image
Can you see the running revision?	Not in most fleet reports; must be collected	Through the OS build	Through the OS build	`/proc/cpuinfo`; `old_microcode` flag for Intel	No separately loaded microcode; the security patch level is the maker's claim, not a measurement
Firmware and silicon	ME or PSP, EC, SSD, NIC firmware from OEM; Kernel DMA Protection needs IOMMU on in firmware	Apple: one vendor for chip, firmware and OS		Same OEM components as Windows; LVFS where the OEM publishes	Baseband and SoC firmware from the SoC vendor through the OEM

The pattern: where one company makes both the chip and the OS, the silicon fix has one owner and arrives with the OS update. Where they are different companies, the fix crosses at least two owners, and the fleet has to read which one won.

HOW IT GOT THIS WAY

Caches, speculation and prefetchers were built to make processors fast, and for decades the threat model for a chip was a crash, not an attacker sharing the core. The Meltdown researchers wrote that it affects "every Intel processor which implements out-of-order execution", which they describe as "effectively every processor since 1995 (except Intel Itanium and Intel Atom before 2013)". Downfall's discoverer says the affected processors "have been around since 2014". Rowhammer's 2014 paper notes that the industry had been aware of the problem "since at least 2012, which is when a number of patent applications were filed by Intel". Each time, the design choice predated the attack by years, and the fix had to be retrofitted through microcode, firmware and operating systems that were never designed to coordinate.

What hasn't changed: shared microarchitectural state between programs is still where these flaws live, and the list keeps growing. Measured in the Linux kernel's source (`torvalds/linux`, `drivers/base/cpu.c`): the `/sys/devices/system/cpu/vulnerabilities` folder was added on 7 January 2018 by commit `87590ce6e373` with 3 entries (Meltdown and two Spectre variants); on `master` as fetched on 18 November 2026 (latest change to that file: commit `620824516557`, 28 September 2026) it lists 19.

THE UNSEEN ANGLE

A fix can be installed and switched off. Intel's Downfall microcode lets software opt out, and on Linux one boot parameter does it. The kernel reports each flaw's state in `/sys/devices/system/cpu/vulnerabilities` as "Not affected", "Vulnerable" or "Mitigation: ..." followed by the mitigation in effect. So a kiosk can carry the right microcode revision and still be exposed by a boot setting someone chose for speed, and a fleet report that lists only versions will call it fixed. Collect the mitigation state as well as the revision, and record who may change the boot parameters. Sources: Linux kernel documentation, ABI file `sysfs-devices-system-cpu`; Intel, Gather Data Sampling guidance.

> *A chip is not bought once. It is a row with a vendor, two delivery channels and a revision you have to read from the running machine.*

These rows in the Trust Ledger

The worked example is Corvane Hotels' check-in kiosk (about 3,000 Windows IoT Enterprise kiosks across 300 properties), plus one row for Halbrook Bank's advisors on cloud VDI. Cells marked "to measure" are placeholders until Corvane collects them; the marks are the finding.

Component	Who can change it	Exposure and blast radius	Patch clock	Stop, revoke, recover	Who can see it	Support end	Decision and owner
SI Kiosk x86 CPU and microcode; two OEM models, A and B (Chapter Four)	CPU vendor writes; OEM ships in BIOS; Microsoft ships via Windows Update	Local code only; one flaw reaches all 3,000 kiosks	Last case to measure: Downfall (T0 public 8 Aug 2023); T2, T3 nc	Microcode is reloaded each boot; a bad revision is replaced by shipping an older or newer one (the January 2018 pattern)	U: effective revision not collected	OEM hardware support date, from the purchase contract	**RENT**, **PIN** the hardware model to contract end; owner: end-user computing lead; review quarterly
SI TPM 2.0 (firmware TPM) used by BitLocker	Chip vendor writes; OEM ships	Local and physical; all kiosks	Per OEM advisory; T1 to T3 nc	OEM firmware update; BitLocker recovery keys escrowed and tested (Chapter Nine)	D: TPM version via OS tools	OEM firmware support end	**RENT**; own the recovery-key escrow; owner: security architect
SI Management engine, EC, SSD and NIC firmware	Intel or AMD, component makers; all shipped by OEM	Below the OS, independent of it; all kiosks of the model	Per OEM BIOS cycle; np for most components	OEM firmware only	U for most components	OEM firmware support end	**RENT**; **ISOLATE**: disable remote management features not used; owner: end-user computing lead
SI Thunderbolt and USB-C ports in the kiosk housing	OEM firmware sets IOMMU; Corvane sets Kernel DMA Protection policy	Physical access; one kiosk at a time	No patch for pre-2019 designs (Thunderspy)	Disable or block the port	D: Kernel DMA Protection state in Windows Security or msinfo32	n/a	**OWN** the policy: protection on, ports covered; owner: property IT
HV Halbrook advisor VDI: cloud provider's host CPU, microcode and hypervisor	Cloud provider only	Shared hosts; every advisor session on the affected hosts	Provider's clock; in Jan 2018 Azure had most hosts updated before disclosure	Provider-scheduled reboots of Halbrook's VMs	C: provider maintenance notices; on confidential VMs, firmware versions in attestation reports; I otherwise	Provider's hardware lifecycle, not published per host	**RENT**; write host-patch notice and maintenance windows into the contract; owner: VDI service owner

SECURITY LENS

Who can change it? The CPU vendor writes microcode; an OEM or an OS vendor delivers it; the workload owner can sometimes switch a mitigation off (Intel's Downfall opt-out). Root-of-trust firmware has its own author: Microsoft for Pluton, Apple for the Secure Enclave, Google for Titan M2, the chip or platform maker for most TPMs. On cloud desktops, only the provider can change the host.

How fast can it be fixed? The OS half of Spectre and Meltdown reached Windows, Linux and macOS within days of 3 January 2018. The silicon half slipped: Intel told customers to stop deploying its January microcode after three weeks, and Zenbleed's client firmware reached OEMs between four and nine months after disclosure, with device arrival not published. Some silicon fixes never come; GoFetch on M1 and M2 is mitigated in software or not at all.

What does AI change here? Our view (as of November 2026): automated analysis can shorten the search for new microarchitectural leaks, but the fix still runs through the same slow chain of microcode, AGESA or BIOS, OEM release and fleet rollout. That chain is the part to measure. Ordinary endpoint telemetry rarely gives reliable evidence that a side channel was used, so prevention and patch clocks carry more weight on this row than detection. NPUs add a new SI component with its own driver and firmware; give it a row before it gets a workload.

BUSINESS IMPLICATIONS

Security. Corvane's kiosks inherit the slowest of three clocks: CPU vendor, OEM BIOS and Windows Update. Halbrook's advisors inherit their cloud provider's clock below the hypervisor, which they cannot see and should not try to run.

Cost. Mitigations can cost performance, and the measured range is wide: Intel says most workloads see minimal impact from the Downfall fix and specific workloads up to 50%. Larkfield Software, whose desktop app runs on 400,000 customer machines, cannot choose its customers' microcode, so it should benchmark its own workload with mitigations on and publish the result rather than advise turning them off.

Visibility. The common blind spot is the effective microcode revision and the firmware of the non-CPU processors. Corvane can collect the first from each kiosk with a script; for the second it depends on the OEM's update tooling and advisories.

Value. Cloud isolation claims rest on this row. When Halbrook buys VDI, part of what it buys is the provider's ability to patch silicon flaws across its hosts before disclosure, as Azure did for most hosts in January 2018. That is worth a contract clause.

Monday rule: add "effective microcode revision" and "firmware owner per processor" to the hardware inventory for your most critical device class, and fill both from the running devices, not the purchase order.

OUR VIEW, AS OF NOVEMBER 2026

Speculative-execution and memory-disturbance flaws will keep arriving: the kernel's vulnerability list grew from 3 to 19 entries in under nine years, and Rowhammer reached DDR5 in 2025. Root-of-trust firmware is moving toward the OS vendor's update channel (Pluton through Windows Update) and toward open designs (Caliptra), which should shorten its clock on the devices that have it. For a fleet bought today, the decision that pays is to choose hardware whose firmware and microcode you can update and read centrally, and to record the end of the OEM's firmware support on the purchase order; Chapter Eleven turns that date into a calendar.

CHAPTER FOUR

Firmware and Boot: Keys with End Dates



On 24 June 2026 the Microsoft Corporation KEK CA 2011 certificate expired. It is the key Microsoft uses to sign updates to the lists of boot code that PCs trust and distrust. Corvane Hotels' 3,000 check-in kiosks started the next morning as usual, which is what Microsoft had said would happen: a device without the newer certificates "will still start and operate normally". The difference was invisible at the front desk. Kiosks that had received the 2023 replacement could keep receiving new boot-level protections; kiosks that had not "will no longer be able to receive new security protections for the early boot process". Nobody at Corvane had chosen that date. It was set in 2011, in a certificate shipped inside firmware Corvane bought from an OEM. The decision it changes: the Secure Boot policy, its keys and their end dates become something the company owns and tracks per device model, even though the firmware itself stays rented.

Three certificates, three jobs, three end dates

UEFI Secure Boot is the check PC firmware makes before it runs each piece of boot code: is this code signed by a key I trust, and is it on the list of code I must refuse? The UEFI specification keeps that trust in four stores. The **Platform Key (PK)** belongs to the platform owner, normally the OEM, and controls changes to the next store. **Key Exchange Keys (KEK)** may update the two lists. The **db** holds the certificates and hashes of code that may run; the **dbx**, the forbidden signature database, holds what must not run. If an image's signature "is found in the forbidden database, the image will not be started".

Microsoft's KB5062710 lists three 2011 certificates that expire in 2026, and they do three different jobs.

- **Microsoft Corporation KEK CA 2011**, in the KEK store, "signs updates to DB and DBX". Expires **24 June 2026**. Successor: Microsoft Corporation KEK 2K CA 2023.

- **Microsoft UEFI CA 2011**, in the db, signs "third-party boot loaders and EFI applications" and "third-party option ROMs" (the firmware on add-in cards such as graphics and network adapters). Expires **27 June 2026**. It has two successors, because Microsoft now separates the jobs: Microsoft UEFI CA 2023 for loaders and Microsoft Option ROM UEFI CA 2023 for option ROMs.
- **Microsoft Windows Production PCA 2011**, in the db, is "used for signing the Windows boot loader". Expires **19 October 2026**. Successor: Windows UEFI CA 2023.

FIGURE 4.1

Three 2011 certificates, three jobs, three end dates



The 2026 expiry was not one event: each 2011 certificate signed something different, each has its own 2023 successor, and none of the three expiries stops a device from starting.

Source: Microsoft KB5062710 (updated 18 May 2026) and KB5079373 (10 Feb 2026); Red Hat, Expiration of Secure Boot signing certificates in 2026. Checked 18 Nov 2026.

Expiry does not stop a device booting. Microsoft's KB5079373 (February 2026) is explicit: "If your device reaches the expiration date without the new certificates, it will still start and operate normally." What stops is new protection: "New Secure Boot and Boot Manager protections cannot be applied," and "some third-party components that rely on Microsoft Secure Boot trust may fail to update." Red Hat says the same for Linux: "a shim signed by this key will continue to boot as long as the public certificate is not removed from or revoked by the system." Red Hat has shipped shims signed under both the 2011 and 2023 certificates, so that "as long as either of the Microsoft certificates is present in the firmware db, the machine will boot." The expiry is a slow loss of the ability to be fixed, not a cliff. Book 1, Chapter 4 mentioned the expiry in one line; here is the mechanism.

Three details turn this from a Microsoft task into a fleet programme.

The OEM holds a key in the chain. Adding a new KEK requires a signature from the Platform Key, which the OEM controls. Microsoft's guidance to manufacturers says "all OEMs must create, sign, and submit updates for the new Microsoft Corporation KEK CA 2023 to Microsoft", so that Microsoft can push it to devices already in the field. KB5079373 adds that "some devices may require an OEM firmware update to apply the new certificates correctly", and that such updates may be available only for devices still in support.

Firmware keeps two copies. Dell's Secure Boot transition FAQ (version 4, 24 May 2026) separates the "Active database", which Windows Update can change, from the "Default database", which is "updated using a BIOS FLASH". Dell states it updates platforms whose end of service life falls after 31 December 2025, and not those whose service life ended before 1 January 2026; for older machines, "the BIOS on these older computers may not support or retain them."

Microsoft rolls it out by confidence class. Microsoft's IT guidance (KB5062713) sorts device classes into confidence levels; "High Confidence" means "Microsoft has validated this device class is safe for updates", while other classes stay under observation. A fleet of one OEM model may sit in a different class from the same model with another firmware version.

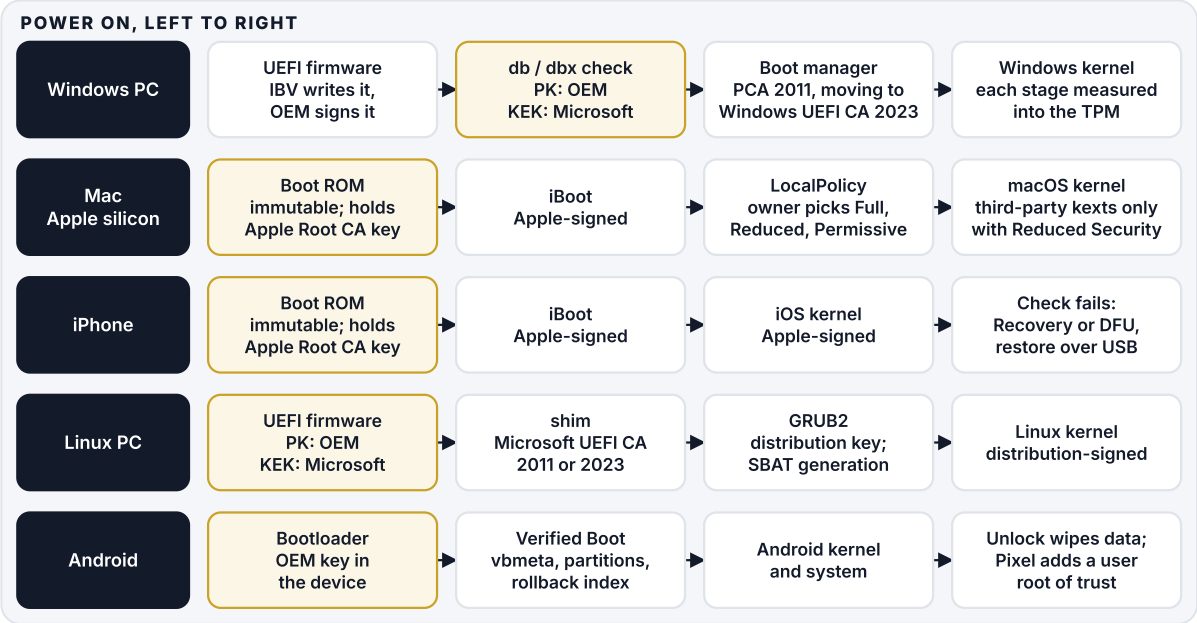
What happened to fleets after June 2026? As of this book's source freeze, no primary report from Microsoft, an OEM or a named fleet describes failures caused by the expiry itself; that is recorded here as not published, not as "nothing happened".

Who holds the keys: five boot chains

Every platform in this book checks each boot stage against a key it already holds. What differs is how many parties hold those keys, and so how many must act before a change reaches a device. Chapter Two drew the full trust maps; Figure 4.2 isolates the boot chain.

FIGURE 4.2

Five boot chains, and who holds the key at each step



Every platform checks each boot stage against a key it already trusts, but only the PC chains split that trust across an OEM, Microsoft and a distribution, which is how a 2011 certificate came to sit in machines that no single party can update alone.

Author's diagram based on the UEFI 2.10 specification, ch. 32; Microsoft Learn, Secure Boot key creation and management guidance; Microsoft KB5062710; Apple Platform Security Guide (boot process; startup disk security policy); Android Verified Boot documentation; rhboot/shim SBAT.md. Checked 18 Nov 2026.

- **Windows PCs.** An **independent BIOS vendor (IBV)** writes most of the UEFI firmware; the OEM customizes and signs it and enrolls the PK; Microsoft's certificates sit in the KEK and db. Since Windows 8 (on sale from 26 October 2012), Microsoft has required x86 PCs certified for Windows to ship with Secure Boot on by default, to trust Microsoft's certificate, and to let the user turn Secure Boot off.
- **Macs with Apple silicon.** The boot ROM is "immutable code" holding the "Apple Root certificate authority (CA) public key". Apple signs every stage, but the owner keeps a real choice: a per-installation **LocalPolicy** set to Full, Reduced or Permissive Security from recoveryOS. Changes that significantly weaken security require entering recoveryOS with the power button, so malware cannot make them remotely. Third-party kernel extensions need Reduced Security.
- **iPhone.** Same Apple chain, no owner setting. If iBoot or an earlier stage fails its check, the device enters Recovery or DFU (device firmware update) mode and must be restored over USB.

- **Linux PCs.** Distributions boot through **shim**, a small loader signed under Microsoft's third-party UEFI CA, which then checks GRUB and the kernel against the distribution's own key. A Linux fleet on PCs therefore depends on Microsoft's third-party CA, which is why the June 2026 expiry was a Linux event too.
- **Android.** Android Verified Boot (AVB) checks partitions against the OEM's key and stores rollback-protection data so that attackers cannot roll a device back "to a vulnerable version of Android". Unlocking the bootloader wipes user data: per Android's documentation, "all state transitions wipe the data partitions". Pixel devices allow a user-settable root of trust "in addition to the built-in root of trust".

On many business PCs the chain starts one step earlier, before the UEFI code runs. **Intel Boot Guard**, introduced with Intel's 4th-generation Core platforms, verifies the firmware's initial boot block against an OEM key whose public half, in a UEFI Forum paper's words, is "programmed into field programmable fuses during the manufacturing process. These fuses cannot be updated once written." AMD's counterpart, Platform Secure Boot, also binds the processor to the OEM's firmware signing key with one-time fuses, as AMD explained to the trade press (ServeTheHome). The fuse makes the OEM's private key the real root of the PC, and that key cannot be rotated in the field. In May 2023, after a ransomware group published data stolen from MSI, Binarly reported that it held firmware image-signing keys for 57 MSI products and Intel Boot Guard keys for more than a hundred; Intel said the Boot Guard keys are generated by the OEM and are not Intel's. On those boards a malicious firmware image signed with the leaked key can pass the hardware check.

The supplier chain on a PC is long: IBV writes, OEM builds and signs, an integrator may customize the image, and the customer configures. Each link can ship a fix late or not at all, which the next two cases show.

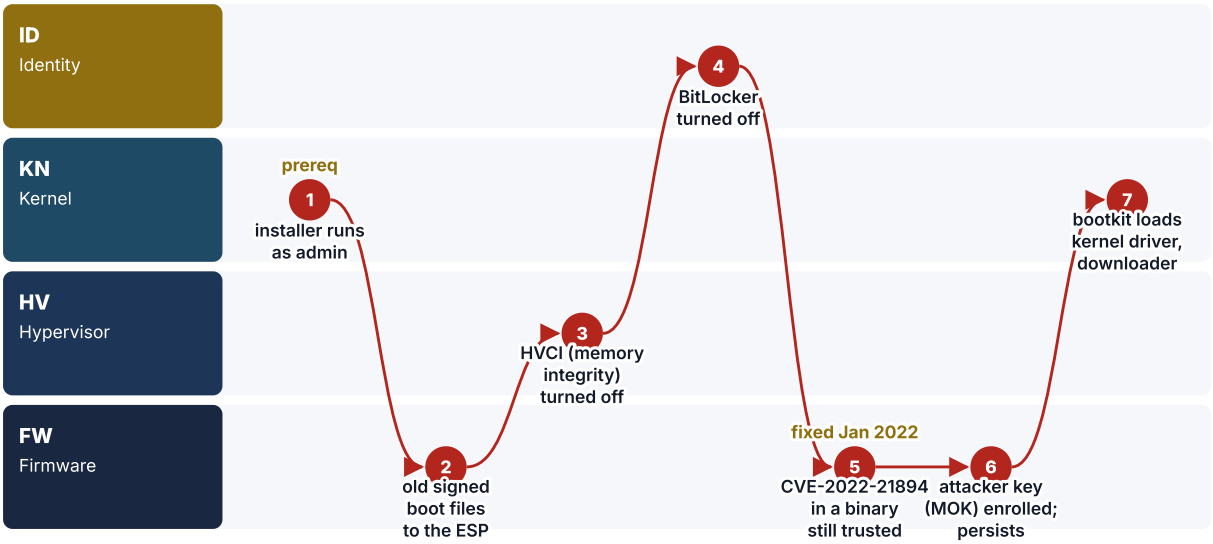
BlackLotus: patching is not revoking

On 1 March 2023 ESET published the first public analysis of **BlackLotus**, "the first in-the-wild UEFI bootkit bypassing UEFI Secure Boot on fully updated UEFI systems". A **bootkit** is malware that runs during boot, before the operating system and its defences. ESET linked it to a bootkit "being sold on hacking forums for $5,000 since at least October 2022"; that is the advertised price ESET reported, not a record of sales. It had a precedent one layer down: in September 2018 ESET described **LoJax**, the "first UEFI rootkit found in the wild", which wrote itself into the firmware's flash and survived reinstalling Windows and replacing the disk. LoJax was not properly signed, and ESET noted that Secure Boot was the first defence that could have stopped it. BlackLotus was built to get past Secure Boot.

The mechanism is the lesson. BlackLotus exploited CVE-2022-21894, a Secure Boot bypass that Microsoft fixed on 11 January 2022. In ESET's words: "Although the vulnerability was fixed in Microsoft's January 2022 update, its exploitation is still possible as the affected, validly signed binaries have still not been added to the UEFI revocation list." BlackLotus simply brought its own copies of legitimate but vulnerable binaries to the machine, as ESET puts it. The firmware checked their signatures, found Microsoft's, and ran them. The patch had fixed the boot manager installed on Windows; it had not told firmware to stop trusting the old one.

FIGURE 4.3

BlackLotus: a fixed bug, still trusted, below every control on the desk



Numbered steps in order. Red: the path the attack or failure took.

BlackLotus needed administrator access once, then used an old Microsoft-signed boot manager that firmware still trusted, more than a year after the bug in it was fixed, to enroll its own key and run before Windows.

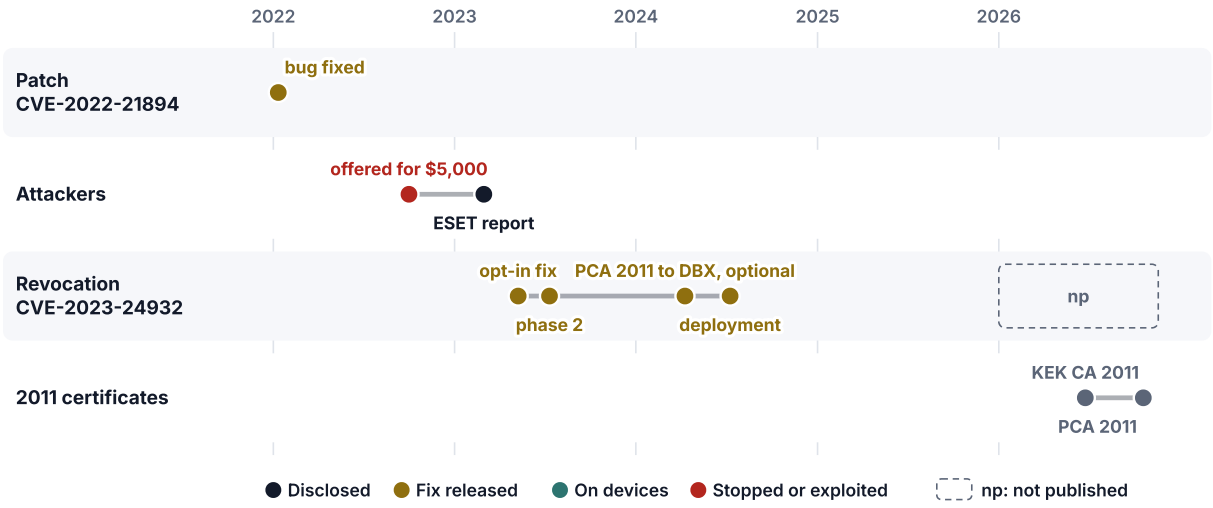
Source: ESET Research, BlackLotus UEFI bootkit: Myth confirmed (1 Mar 2023); MSRC CVE-2022-21894 (11 Jan 2022) and CVE-2023-24932 (9 May 2023); Microsoft KB5025885. Checked 18 Nov 2026.

ESET describes the sequence: the installer, which needs administrator rights, deploys files to the EFI System Partition (ESP), disables HVCI (memory integrity, Chapter Five) and BitLocker, and reboots; the old boot manager is exploited; the attackers' Machine Owner Key is enrolled "to achieve persistence even on systems with UEFI Secure Boot enabled". Microsoft's own note is that the malware "requires physical or administrative access to the device". The precondition is real, and so is the result: a foothold below every control the desk runs.

Revocation turned out to be a multi-year programme. Microsoft tracked the bypass as CVE-2023-24932 (9 May 2023) and staged it in KB5025885: opt-in mitigations in May and July 2023; an evaluation phase from 9 April 2024 that added the Windows UEFI CA 2023 certificate to the db, a boot manager signed by it, and the option to put Windows Production PCA 2011 into the dbx; and a deployment phase from July 2024, whose mitigations "are not enabled by default". The final enforcement phase "will not begin before January 2026", with at least six months' notice. In the version of KB5025885 updated on 30 September 2026, its date was still "to be announced".

FIGURE 4.4

Patched in 2022, still not revoked by default in late 2026



The bug BlackLotus used was fixed in January 2022, but revoking the vulnerable boot managers has run as an opt-in programme since May 2023, and Microsoft had not announced the enforcement date when this book's sources were frozen.

Source: MSRC Security Update Guide (CVE-2022-21894, CVE-2023-24932); Microsoft KB5025885 (change log to 30 Sept 2026); ESET Research (1 Mar 2023); Microsoft KB5062710. Checked 18 Nov 2026.

Why so slow? Because revocation breaks things the patch did not. KB5025885 warns that after the mitigations "you might be unable to start or recover your device from existing media", that "older external boot media might not start", and that "even reformatting of the disk will not remove the revocations if they have already been applied." Revoking the old certificate protects the device and, in the same step, invalidates every old recovery USB stick, network-boot (PXE) image and install medium in the building. Chapter Nine turns that trade-off into a decision tree. For this chapter, the rule is simpler: a fix that ships as a patch is not finished until the vulnerable signed code is also revoked, and revocation needs its own plan, test and clock.

Revocation can be fast when little depends on the revoked code. On 14 January 2025 ESET published CVE-2024-7344, a flaw in a recovery application signed under Microsoft's third-party UEFI CA 2011 that let it load unsigned code at boot, and Microsoft released the fix, revoking the vulnerable binaries, the same day. An application few fleets need at boot can be revoked in one monthly update; the Windows boot manager, on every recovery stick in the building, cannot.

LogoFAIL and PKfail: when the firmware itself is the flaw

BlackLotus abused trusted code. The next two cases are flaws inside the firmware, where the dbx cannot help at all.

LogoFAIL. On 29 November 2023 Binarly announced flaws in the image parsers that UEFI firmware uses to draw the manufacturer's logo at boot; the full details were presented at Black Hat Europe and published on 6 December 2023, when CERT/CC issued VU#811862, "Image files in UEFI can be abused to modify boot behavior". The affected parsers came from the IBVs: CERT/CC lists American Megatrends (AMI), Insyde and Phoenix among the affected vendors. Binarly describes attackers storing malicious logo images "either on the EFI System Partition (ESP) or inside unsigned sections of a firmware update to bypass Secure Boot". The parser runs before any signature check on the OS, and the code doing the parsing is the firmware itself, so there is no signed binary to put in the dbx. The fix is a firmware update from each OEM, built on the IBV's corrected code. Phoenix's statement to CERT/CC shows the chain in one sentence: "We have provided updates to our customers and they are producing firmware updates."

PKfail. On 25 July 2024 Binarly reported that many devices shipped with test Platform Keys never meant for production, often with certificates whose issuer contains "DO NOT TRUST" or "DO NOT SHIP". Binarly's scan found that "more than 10% of firmware images in our dataset use an untrusted Platform Key", falling to 8% for firmware from the past four years, and that the first affected firmware was released "back in May 2012, while the latest was released in June 2024". Binarly also reported finding the private part of one such key in a public leak. Whoever holds a PK's private half can enroll a new KEK and, through it, change the db, so the whole Secure Boot chain on those devices was only as strong as a test key's secrecy. CERT/CC's note VU#455367 (30 August 2024) recommends a firmware update from the PC vendor, links a tool to replace the PK, and records Intel's statement that its affected products are no longer supported. For a device past its support date, the fix may be a hardware replacement.

BootHole and the size of the revocation list

The precedent for all of this is **BootHole** (CVE-2020-10713), published by Eclypsium on 29 July 2020: a buffer overflow in how GRUB2 parsed its configuration file, `grub.cfg`, which "is a text file and typically is not signed". Any shim and GRUB pair that loaded an external config was exposed, on Linux and on Windows machines that trusted Microsoft's third-party CA. Microsoft published guidance the same day. Eclypsium warned that "until all affected versions are added to the dbx revocation list, an attacker would be able to use a vulnerable version of shim and GRUB2 to attack the system", and recorded that, "due to the risk of bricking systems or otherwise breaking operational or recovery workflows", the dbx updates would first be offered for manual application.

BootHole also exposed a hard limit. The shim project's design note on SBAT records that the BootHole revocation added "3 certificates and 150 image hashes" to the dbx on x64, consuming "10kB of the 32kB, or roughly one third, of revocation storage typically available on UEFI platforms". Revoking by hash does not scale when a component is signed by many vendors in many versions. The answer, **SBAT** (Secure Boot Advanced Targeting), revokes by a generation number in the binary's metadata, so a new revocation edits an entry instead of adding hashes. Microsoft's Secure Version Number for its boot manager, added to KB5025885 in 2024, follows the same idea. Both are firmware-era fixes for a storage limit set in firmware.

How firmware actually reaches a device

Firmware updates reach PCs in three main ways: the OEM's own update tool or download page; Windows Update, for OEMs that publish firmware there as UEFI capsule updates; and on Linux, the Linux Vendor Firmware Service (LVFS) with the `fwupd` client, a portal "used by all major Linux distributions". Coverage on LVFS is per vendor and uneven; on 18 November 2026 its vendor list showed Dell "responsible for 9667 firmware files" and Lenovo's ThinkPad team for 4203; coverage for other OEMs and models varies, so check each model. For a Corvane kiosk, the honest answer to "who can update this firmware, and how?" is often "the OEM's tool, on a technician's visit".

Two things follow for operations. First, measured boot records each stage into the TPM, and changing Secure Boot keys changes those measurements; Red Hat warns that "PCR7 will change after the db update" (PCR7 is the TPM register that records the Secure Boot configuration), which breaks TPM-sealed disk unlocking and attestation baselines until they are re-sealed. On Windows, plan BitLocker recovery keys before the key update, not after. Second, measurement helps only if someone reads it. Attestation, the remote check of those measurements, is consumed in Chapter Nine (recovery) and Chapter Seventeen (its limits).

Side by side: boot and firmware across five platforms

Decision point	Windows	macOS	iOS	Linux	Android
Who holds the boot keys	OEM (PK), Microsoft (KEK, db entries)	Apple (boot ROM key); owner sets LocalPolicy	Apple	OEM (PK), Microsoft (KEK, third-party CA), distribution (shim's key)	OEM; Pixel adds a user root of trust
What the customer can change	Secure Boot on or off; custom PK and db on many PCs; apply DBX and 2023 certificates on Microsoft's schedule	Full, Reduced or Permissive Security, from recoveryOS	Nothing	Same as Windows, plus Machine Owner Keys in shim	Lock or unlock (wipes data) on devices that allow it
Revocation path	dbx and boot manager SVN; staged by Microsoft (KB5025885)	Apple signing; per-device personalized signatures in Full Security	Apple signing	dbx and SBAT generations	AVB rollback index
2026 certificate exposure	KEK, UEFI CA and PCA 2011 all expire	Not affected	Not affected	UEFI CA 2011 (shim) and KEK 2011	Not affected
Firmware and silicon	IBV writes, OEM ships; Windows Update or OEM tool	Apple: one vendor for chip, firmware and OS		OEM; LVFS where the OEM publishes	SoC vendor and OEM, inside the device image

The 2026 calendar belonged to the PC platforms. Apple and Android avoided it by holding their chains end to end, which is the same property that leaves their customers fewer settings to change.

HOW IT GOT THIS WAY

BIOS was trusted because it lived in ROM and did little. Its successor, UEFI, is based on Intel's EFI 1.10 specification and is now maintained by the UEFI Forum; it gave pre-boot firmware drivers, a file system, network stacks and image parsers, which is to say an attack surface. Secure Boot added a key hierarchy on top, and from 2012 Microsoft required PCs certified for Windows 8 to ship with it on. To make one Windows and one shim boot on hardware from hundreds of makers, the industry agreed to trust a small set of Microsoft certificates in almost every PC. That made PC boot security workable and created a single calendar of end dates that nobody downstream chose.

What hasn't changed: firmware still parses untrusted images before the OS loads. Binarly frames LogoFAIL against Rafal Wojtczuk and Alexander Tereshkin's Black Hat USA 2009 talk, "the exploitation of a BMP parser bug in the UEFI reference code", 14 years earlier. Measured in today's reference code (`tianocore/edk2` , `MdeModulePkg/Library/BaseBmpSupportLib/BmpSupportLib.c`): the library was added on 22 December 2017 (commit `4123d6da7095`), its last change was a formatting pass on 5 December 2021 (commit `1436aea4d570`), and on the repository's head as fetched on 18 November 2026 its header still says that `TranslateBmpToGopBlt()` "receives untrusted input and performs basic validation."

THE UNSEEN ANGLE

Certificates are a calendar set years ago by someone else; 2026 is the first time most CTOs met theirs. The hidden part is that the calendar lives in two places on each PC. Dell's FAQ distinguishes the Active database, which Windows Update can change, from the Default database, which only a BIOS flash updates. Our reading (an inference from Dell's FAQ, not a Dell statement): a kiosk that received the 2023 certificates through Windows Update but never got the BIOS update can return to the 2011-only set if a technician resets Secure Boot keys to factory defaults during a repair. Track "2023 certificates in the Active db" and "in the Default db" as two columns per model, and add "never reset Secure Boot keys to defaults" to the field-repair checklist until both are yes. Source: Dell, "Secure Boot Transition FAQ", article 000390990, version 4 (24 May 2026).

> *A patch fixes the code you have. A revocation stops trusting the code an attacker can bring. They are two jobs with two clocks.*

These rows in the Trust Ledger

Corvane Hotels' check-in kiosk again: about 3,000 kiosks across 300 properties, two OEM models, Windows IoT Enterprise LTSC, plus the Linux signage players in each lobby. Placeholders stay marked until measured.

Component	Who can change it	Exposure and blast radius	Patch clock	Stop, revoke, recover	Who can see it	Support end	Decision and owner
FW Kiosk UEFI firmware (IBV code, OEM build), models A and B	IBV writes; OEM builds, signs and ships; Corvane installs	ESP and logo path (LogoFAIL class); physical USB; all kiosks of a model	LogoFAIL: T0 6 Dec 2023; T1 per OEM; T2, T3 nc	OEM firmware only; DBX does not fix parsers; rollback per OEM	D: firmware version by inventory	OEM firmware support end per model, from the contract	**RENT**; **OWN** a firmware ring (5 properties first); owner: end-user computing lead
FW Secure Boot keys: PK, KEK, db (2023 certificate status)	OEM (PK); Microsoft (KEK, db) via Windows Update; Corvane (setting)	Every kiosk; a missing KEK 2023 blocks all future Microsoft db and dbx updates	Model A: Active db 2023 yes, Default db to measure; model B: High Confidence class to check	Re-apply via Windows Update or BIOS flash; never reset to defaults in the field	D: Microsoft's PowerShell check and event log, per kiosk	2011 certificates expired (KEK 24 June 2026, PCA 19 Oct 2026); track the 2023 successors per model	**OWN** the policy and the per-model status; owner: security architect; review monthly until both columns are yes
FW Boot manager revocation (CVE-2023-24932)	Microsoft stages; Corvane opts in before enforcement	Admin or physical access needed; all kiosks	T1 9 May 2023; enforcement np; Corvane T3 nc	Revocation cannot be undone by reformatting; old recovery media stop working	D: event IDs in KB5025885	n/a	**OWN**: apply to a pilot ring after new recovery media are built; owner: end-user computing lead
FW Lobby signage, Linux on x86 with shim and GRUB	Distribution (shim, GRUB); Microsoft (third-party CA); OEM (PK)	Physical; all 300 lobbies	Dual-signed shim: per distribution; T3 nc	SBAT generations; dbx	U: no fleet report of db contents	Distribution release support end	**RENT**; **PIN** to a distribution release with a dual-signed shim until its end of support; owner: signage vendor manager

SECURITY LENS

Who can change it? On a PC, at least three parties: the IBV writes the firmware, the OEM signs it and holds the PK, Microsoft holds the KEK and the certificates most code is signed under. A Linux distribution adds a fourth through shim. Apple holds its whole chain and lets the Mac owner pick a security level. Android OEMs hold theirs, and Pixel owners may add their own root of trust.

How fast can it be fixed? A boot manager bug can be patched in a monthly update, as CVE-2022-21894 was on 11 January 2022. Revoking the vulnerable signed versions took Microsoft from May 2023 to a still-unannounced enforcement date. Parser flaws inside firmware, such as LogoFAIL, are fixed only when each OEM ships a new firmware build and each customer installs it, and a test PK on a device past support may never be fixed.

What does AI change here? Our view (as of November 2026): firmware binaries are a good target for automated analysis, because the same IBV code appears across many OEMs, as LogoFAIL and PKfail both showed through large scans. That speeds discovery on the attacking side and the defending side alike, and does nothing for the slowest step, OEM release and field installation. Keep the firmware ring and the per-model certificate status current, because those are what decide how long a newly found flaw stays open on your fleet.

BUSINESS IMPLICATIONS

Security. Firmware sits below every control Corvane bought; BlackLotus turned off HVCI and BitLocker on its way down. Halbrook Bank's 2,500 branch PCs carry the same 2011 certificates, and its branch PCs are where a missed KEK update would quietly stop future boot fixes.

Cost. A firmware fix can mean a field visit. If a kiosk model needs a BIOS flash that cannot be pushed remotely, Corvane's cost is one technician visit per property, multiplied across 300 properties; price it before the next certificate or revocation deadline, not during it. Rebuilding recovery and install media before revocation is a separate, cheaper line.

Visibility. The ESP and firmware are outside most EDR views. Corvane can see firmware versions and, per kiosk, whether the 2023 certificates are applied; it cannot see firmware parser flaws until an OEM advisory names them.

Value. A certificate you did not track becomes a programme you did not budget. Larkfield Software, which ships desktop software rather than boot code, still has a stake: customers whose Linux machines fail to accept a new shim will open tickets with every vendor on the box.

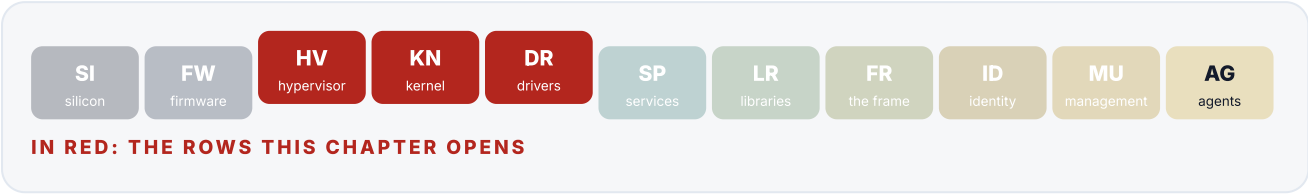
Monday rule: for each device model you own, write down who holds its PK, whether the 2023 certificates are in its Active and Default databases, its firmware support end date, and whether your recovery media still boot after revocation.

OUR VIEW, AS OF NOVEMBER 2026

The 2026 expiry was a rehearsal. Microsoft's enforcement of the BlackLotus revocation is still ahead, the 2023 certificates have end dates of their own, and the post-quantum key migration in Chapter Seventeen will move the same stores again. Fleets that built a per-model certificate inventory and a firmware ring this year will run those programmes as routine; fleets that relied on "it still boots" will meet the next deadline the same way they met this one. Buy hardware whose firmware the OEM publishes through a central channel (Windows Update or LVFS) and whose firmware support end is written in the contract.

CHAPTER FIVE

Privileged Code: Kernels, Hypervisors, Drivers and the Morning of 19 July 2024



At 04:09 UTC on Friday 19 July 2024, CrowdStrike published a routine content update for its Falcon sensor to Windows hosts running sensor version 7.11 and above. At Halbrook Bank it was nine minutes past midnight in the eastern branches. Nobody at the bank approved the update, and nobody needed to: the bank had bought a security product whose vendor could change what its kernel driver did, on every desk, without a release. Seventy-eight minutes later the vendor reverted the file. In that window Windows machines that were online and received it were exposed to a crash; many crashed, and many could not finish booting. This chapter tells that morning as a technical case, then asks the question behind it for every row below the screen: whose code runs with kernel privilege on your devices, what the vendors have moved below or out of the kernel, and who can update it. The decision it changes is narrow and expensive to skip: **a vendor's content channel into privileged code gets the same deployment discipline as code**, with rings, a pause and a tested rollback, written into the contract.

The morning: seventy-eight minutes and a long tail

Start with the clock, because it is the part a buyer can change. CrowdStrike's Preliminary Post Incident Review (PIR), published on 24 July 2024, gives the window: a Rapid Response Content update "was published to Windows hosts running sensor version 7.11 and above" at 04:09 UTC and "triggered crashes (BSOD) on systems that were online between 04:09 and 05:27 UTC". Mac and Linux hosts were not affected. Windows hosts that were offline in that window, or did not connect during it, were not affected either.

Two features made the window decisive. First, the update was not a new sensor. It was **content**: configuration data the sensor reads to decide what to watch and what to block. Second, the sensor's driver starts during boot, as security drivers commonly do, so that it can

see malware that starts before user programs. A machine that crashed and restarted loaded the same driver and the same file and could crash again. The vendor's revert at 05:27 UTC stopped new victims. It did not help a machine that could not stay up long enough to download the fixed file.

On 20 July Microsoft's David Weston wrote: "We currently estimate that CrowdStrike's update affected 8.5 million Windows devices, or less than one percent of all Windows machines." That figure is Microsoft's estimate, and this book uses it as such. It was not a cyberattack. No intruder was involved; a trusted vendor's normal update path delivered a defect to privileged code. The record of who paid, and the lawsuits that followed, is in Chapter Thirteen.

What the root cause analysis says

CrowdStrike's External Technical Root Cause Analysis (RCA), dated 6 August 2024, is unusually specific, and the specifics are the lesson. Its vocabulary needs four definitions.

- **Sensor Content** ships inside a sensor release. It includes **Template Types**: predefined fields, compiled into the sensor, that detection engineers can match against.
- **Rapid Response Content** is delivered from the cloud between sensor releases, in **Channel Files**. Each Channel File carries **Template Instances**: regular-expression criteria for one Template Type.
- The **Content Interpreter**, part of the sensor's C++ code, tests live system data against those criteria.
- The **Content Validator** checks new Channel Files against the Template Type definitions before they ship.

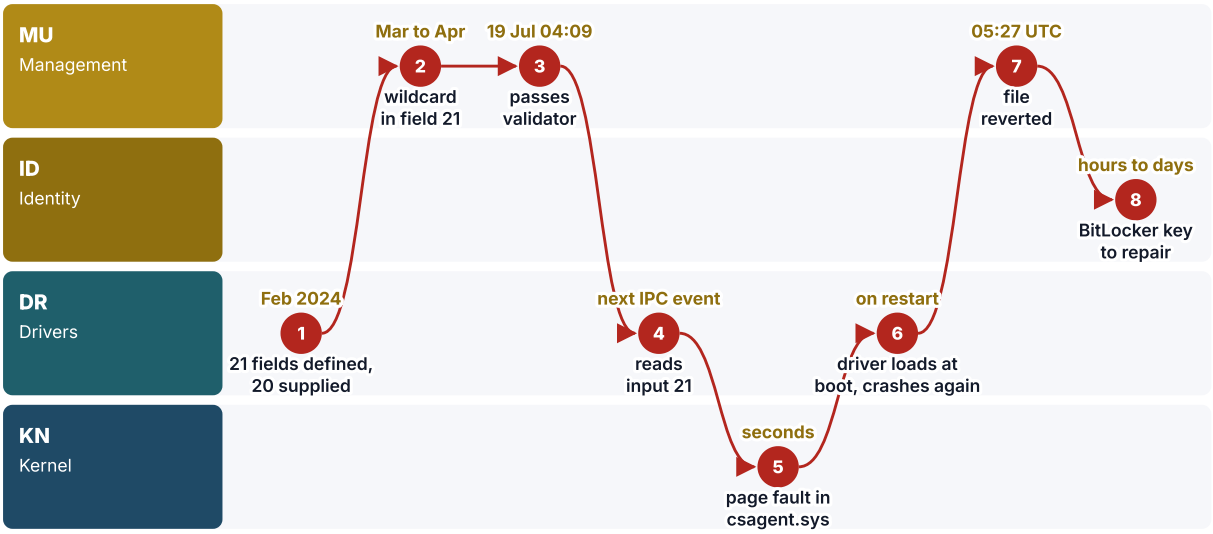
With sensor 7.11 in February 2024, CrowdStrike added a Template Type for Windows interprocess communication (IPC), such as named pipes. Its instances travel in Channel File 291. The RCA states the flaw in one sentence: "The new IPC Template Type defined 21 input parameter fields, but the integration code that invoked the Content Interpreter with Channel File 291's Template Instances supplied only 20 input values to match against." Testing did not catch it, in part because the test instances and the first IPC instances deployed in the field used a wildcard in the 21st field, so the interpreter never had to read it.

On 19 July two new IPC Template Instances were deployed. One used a non-wildcard criterion for the 21st field. The Content Validator passed it, because it "based its assessment on the expectation that the IPC Template Type would be provided with 21 inputs." At the next IPC notification, the interpreter tried to read a 21st value from a 20-entry array. The RCA calls this "a latent out-of-bounds read issue in the Content Interpreter". The crash dump it publishes shows the fault in `csagent.sys`, CrowdStrike's file system filter driver: a

page fault on an address that held no valid memory, and a bugcheck. Microsoft's own analysis of crash reports, published on 27 July, says its observations "confirm CrowdStrike's analysis that this was a read-out-of-bounds memory safety error".

FIGURE 5.1

Channel File 291 on the Ledger: a latent sensor flaw, a content push, a kernel crash



Numbered steps in order. Red: the failure path. Steps 1 and 2 are the latent flaw; steps 3 to 6 took minutes.

The mismatch shipped in February 2024 and waited five months; the first content update to use the 21st input field turned a latent out-of-bounds read into a kernel crash.

CrowdStrike, External Technical Root Cause Analysis, Channel File 291 (6 Aug 2024) and Preliminary Post Incident Review (24 Jul 2024); Microsoft Intune recovery tool post (20 Jul 2024, updated 21 Jul).

Two sentences in the RCA deserve a place in any contract review. The first: "Rapid Response Content is configuration data; it is not code or a kernel driver." That is true, and it is why content moves faster than code. The second is the list of fixes: compile-time checks of input counts, a runtime bounds check in the interpreter, wider test coverage, new Content Validator checks, and "Template Instances should have staged deployment", with customers given control over "where and when Rapid Response Content updates are deployed." Content is not code, but on that morning it changed what privileged code did as much as a code release could. That is the reason it needs the same deployment discipline.

One control many customers relied on did not apply. Sensor update policies let a customer run the latest sensor or one or two versions older (N-1, N-2), but they govern sensor releases. Rapid Response Content was delivered separately, to every sensor from 7.11 up, so a fleet pinned at N-1 or N-2 received Channel File 291 like any other. CrowdStrike's content update

policies now let a customer assign each content category to an early-access or general-availability ring, or pause it, though system-critical content cannot be paused. Microsoft moved the requirement up a level: under version 3.0 of the Microsoft Virus Initiative (MVI), its program for security vendors, announced in 2025, "security product updates must be gradual, leverage deployment rings and leverage monitoring". Microsoft's own Defender Antivirus has the same two-part shape, product updates on one schedule and security intelligence on another, and Microsoft documents gradual-rollout channels for its monthly platform and engine updates.

The recovery clock decided the cost

The patch clock for this case is short and odd: for machines that stayed up, T0 and T1 are both 19 July, because the fix (the reverted file) existed 78 minutes after the fault. For machines already in a crash loop, the fix was a manual procedure, packaged as a tool by Microsoft the next day. The clock that decided the cost was the **recovery clock**: how long until each crashed machine was working again.

Recovery was often a person at a keyboard. On 20 July Microsoft published a recovery tool that builds boot media; the first release had one path, and the update of 21 July added a second. The path through the Windows Preinstallation Environment (WinPE) repairs the disk directly and does not need local admin rights, but "you may need to manually enter the BitLocker recovery key". The Safe Mode path can avoid the key on devices that use TPM-only BitLocker protectors, but needs a local administrator account, and devices with a TPM plus PIN need the PIN or the key. Either way, a team needed boot media, physical or remote console access, and a working way to look up recovery keys for thousands of machines at once. Chapter Nine draws that dependency graph.

CrowdStrike reported progress as sensor connectivity: more than 97% of Windows sensors online as of 24 July (its update of 25 July), and "~99% of Windows sensors are online compared to before the content update" by 29 July, with a normal weekly variance of about 1%. That is a measure of sensors reporting in, not of tellers serving customers or flights leaving. Use it as the vendor's curve, labeled, and measure your own.

Microsoft's structural answer came in stages. At Ignite on 19 November 2024 it announced the Windows Resiliency Initiative, including **Quick Machine Recovery**, which lets IT "execute targeted fixes from Windows Update on PCs" that cannot boot. Microsoft announced it generally available on Windows 11, version 24H2, in July 2025 (update KB5062660). It is on by default for Windows 11 Home and off by default on Pro, Education and Enterprise. A bank that has not turned it on and tested it does not have it.

Who runs in the kernel, and what sits below it

The **kernel** is the part of the operating system that runs with full hardware privilege: memory, devices, every process. Kernel privilege is the step many local exploits try to reach, because from there they can switch off the controls above. Not every attack ends there, and some of what follows is about layers vendors have placed under or beside the kernel so that kernel privilege is no longer the last word.

Windows uses its hypervisor to create Virtualization-based Security (VBS): an isolated environment that, in Microsoft's words, "assumes the kernel can be compromised". Memory integrity, also called Hypervisor-protected Code Integrity (HVCI), moves the decision about which kernel code may run into that environment. Microsoft says it is "enabled by default on many new Windows 11 devices that meet hardware and compatibility requirements", and that actual protection "depends on device eligibility, driver compatibility, policy, rollout assignment, and restart." Incompatible drivers can block it.

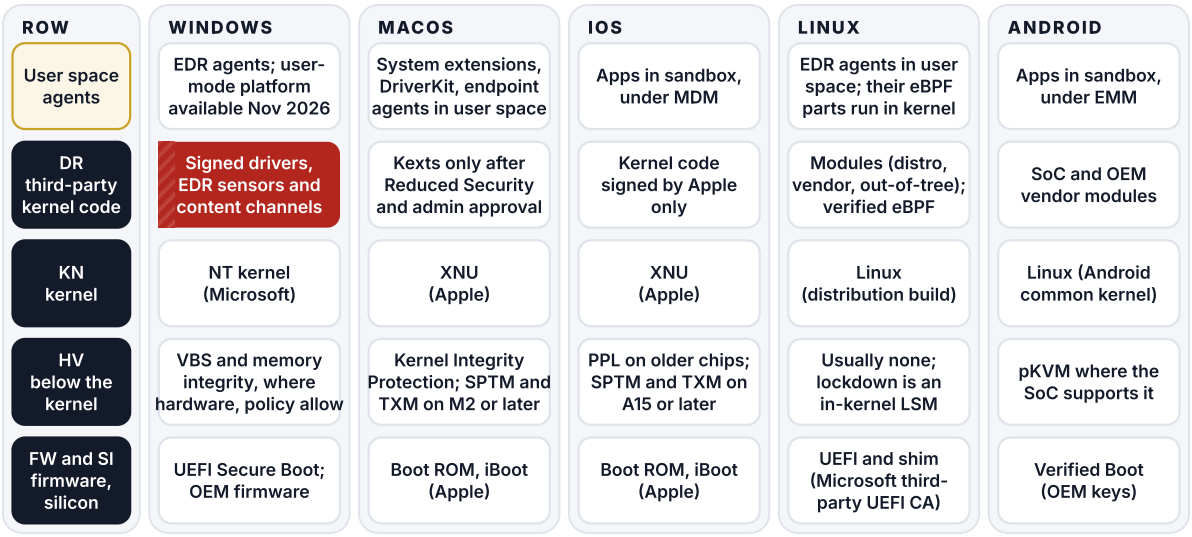
Apple builds the protections into its chips. The Apple Platform Security Guide describes Kernel Integrity Protection, enabled once the kernel finishes initializing, and on recent chips a **Secure Page Table Monitor** (SPTM) and **Trusted Execution Monitor** (TXM) that sit beside the kernel and police its page tables and code signing. SPTM "is supported in the A15 or later and M2 or later SOCs" and replaces the older Page Protection Layer (PPL), which Apple used in iOS, iPadOS, visionOS and watchOS but not in macOS. The protection a Mac or iPhone gets therefore depends on its chip generation.

Android devices can run **pKVM**, a KVM-based hypervisor that "isolates pVMs and Android into mutually distrusted execution environments". The host Android kernel cannot read a protected virtual machine's memory unless the owner shares it. Google is explicit that system-on-chip vendors "must satisfy a new set of requirements" to support it, so it is device dependent.

Linux usually runs on bare metal with nothing under it. Its defenses are inside the kernel: Linux Security Modules such as SELinux, AppArmor and Landlock, and since Linux 5.4 (24 November 2019) the lockdown module, "intended to strengthen the boundary between UID 0 (root) and the kernel", which a distribution can turn on.

FIGURE 5.2

Privileged code per OS: who runs in the kernel, and what sits below it where enabled



Windows is the platform where third-party security sensors commonly run in the kernel; a layer below the kernel exists on each platform only where the hardware, the chip generation or the policy turns it on.

Author's diagram based on Microsoft Learn (memory integrity; driver signing policy), Windows IT Pro Blog (17 Nov 2026), Apple Platform Security Guide and Apple Platform Deployment (system and kernel extensions), Android Open Source Project (pKVM), kernel.org (LSM and BPF docs) and the Linux 5.4 release notes.

Decision point	Windows	macOS	iOS	Linux	Android
Layer below the kernel	VBS and memory integrity, where hardware, drivers and policy allow	Kernel Integrity Protection; SPTM and TXM on M2 or later	PPL on older chips; SPTM and TXM on A15 or later	Usually none; defenses are in-kernel LSMs and lockdown	pKVM where the SoC vendor supports it
Third-party kernel code	Signed drivers; EDR sensors, which commonly use kernel drivers, and their content	Kexts only after lowering to Reduced Security with admin approval	Kernel code signed by Apple only	Distribution, vendor and out-of-tree modules; eBPF programs	SoC and OEM vendor modules
Where an EDR sensor runs	Kernel driver today; user-mode platform generally available on Windows 11 from 17 Nov 2026, vendor dates vary	System extension in user space	No kernel sensor; apps and MDM	Kernel module or eBPF	No kernel sensor; apps and EMM
Who can stop a bad push	The vendor; you, only if the contract and console give you content rings	The vendor; you, through MDM approval of extensions	Apple	Your distribution and you, through package pinning	Google and the OEM
Firmware and silicon	Memory integrity works best on CPUs with Mode-Based Execution Control (MBEC, Intel) or Guest Mode Execute Trap (GMET, AMD); older CPUs emulate them at a performance cost	Protection set by chip generation, not by a setting		Lockdown depends on Secure Boot via shim	pKVM depends on the SoC

Signed is identity, not quality: drivers, BYOVD and the blocklist

Windows decides which third-party kernel code may load by signature. Microsoft's driver signing policy states: "Starting with Windows 10, version 1607, Windows will not load any new kernel-mode drivers which are not signed by the Dev Portal." Before that, 64-bit Windows from Vista onward required an Authenticode signature cross-signed by Microsoft. Exceptions remain for upgraded machines, machines with Secure Boot off, and drivers signed with certificates issued before 29 July 2015. A signature says who submitted the driver. It says nothing about whether the driver is safe to run.

Two precedents show the limits. In 2010, Stuxnet's drivers carried valid signatures. Symantec's W32.Stuxnet Dossier (version 1.4, February 2011) records a driver "digitally signed with a compromised Realtek certificate that was revoked on July 16, 2010 by

Verisign", and a later driver signed with a certificate from JMicron, revoked on 22 July 2010. Revocation came after the drivers were in the wild. Chapter Thirteen tells the rest of that case.

The second precedent needs no stolen key. In **bring your own vulnerable driver** (BYOVD), an attacker with admin rights installs a legitimately signed driver that has a known flaw, then uses the flaw to act in the kernel. ESET reported on 30 September 2022 that in autumn 2021 attackers targeting an aerospace company employee in the Netherlands and a political journalist in Belgium used "the first recorded abuse of the CVE-2021-21551 vulnerability", in the Dell driver `dbutil_2_3.sys`, to disable "the monitoring of all security solutions on compromised machines".

Microsoft's answer is the **vulnerable driver blocklist**. Its documentation says that "since the Windows 11 2022 update, the vulnerable driver blocklist is enabled by default for all devices", and that it is also enforced when memory integrity, Smart App Control or S mode is on. Microsoft says the list is updated quarterly and delivered through the monthly Windows updates. Microsoft also says its downloadable list "usually contains a more complete set of known vulnerable drivers than the version in the OS", because it holds back some blocks to avoid breaking devices. The blocklist is a revocation path with a deliberate lag, and the lag is the vendor's compatibility judgment, not yours.

Moving code out of the kernel

Every platform is moving third-party code away from full kernel privilege, at a different speed.

Apple went first. From macOS 10.15 (2019) developers could build system extensions that "run in user space rather than at the kernel level", using frameworks including DriverKit for USB and human interface drivers. Apple's deployment guide names endpoint security tools among the agents that no longer need kernel extensions. On a Mac with Apple silicon, a kernel extension now needs the user or MDM to lower the startup security to Reduced Security and approve it.

Linux offers eBPF (a name that began as "extended Berkeley Packet Filter"), small programs loaded into the kernel after a verifier checks them. The verifier rejects invalid control flow and any loop it cannot prove will end (bounded loops have been accepted since Linux 5.3), then, in the kernel documentation's words, "descends all possible paths" to check memory safety. That makes eBPF more constrained than a kernel module, not risk free: the verifier and the just-in-time compiler are themselves kernel code with their own bug history. Red Hat's knowledge base records kernel panics on Red Hat Enterprise Linux 9.4 kernels caused by the Falcon sensor: "eBPF program causes kernel panic on kernels 5.14.0-410+". Linux hosts were spared on 19 July 2024; they were not spared the class of failure.

Memory safety moves into the kernel itself. Linux 6.1, released on 11 December 2022, "includes initial support for the Rust programming language": the infrastructure for writing kernel code in Rust, not a conversion of existing C. Google wrote in November 2025 that "Android's 6.12 Linux kernel is our first kernel with Rust support enabled and our first production Rust driver". Rust removes classes of memory-safety bugs in its safe code; unsafe blocks and calls into C can reintroduce them, and it does not remove other vulnerabilities. Figure 5.3 shows Google's numbers for Android as a whole.

FIGURE 5.3

Android: memory-safety bugs as a share of Android vulnerabilities

Year	Memory-safety share	What Google says about the number
2019	76%	Share of Android vulnerabilities caused by memory-safety issues
2024	24%	Data for 2024 extrapolated to the full year from the September 2024 bulletin
2025	Below 20%	First year under 20%; published about two months before year end and described as very likely close to final

Google reports the share fell as new code moved to memory-safe languages; it is a share of reported vulnerabilities, not a measure of risk.

Google Security Blog, Eliminating Memory Safety Vulnerabilities at the Source (25 Sept 2024); Google, Rust in Android: move fast and fix things (13 Nov 2025).

A note on counting. In February 2024 the Linux kernel became its own CVE Numbering Authority. Its process documentation says the team is "overly cautious and assign CVE numbers to any bugfix that they identify." The jump in Linux kernel CVE counts since then is a change in process, not a collapse in quality, and Linux CVE totals should not be read as a ranking against other kernels.

Windows is the last to move, and moved on 17 November 2026. In November 2024 Microsoft said it was building capabilities so that "security products, like anti-virus solutions, can run in user mode just as apps do", with a private preview for partners planned

for July 2025. On 17 November 2026 Microsoft announced that the "Windows endpoint security platform is reaching general availability on Windows 11." Its user-mode APIs cover monitoring and blocking for file systems, processes, the registry and the Object Manager. Microsoft's partner table lists availability by vendor from Q4 2026 to the first half of 2027, CrowdStrike Falcon in Q1 2027. General availability of the platform is not the same as your sensor running on it.

HOW IT GOT THIS WAY

The PC industry grew on an open driver model: any hardware maker could ship code that ran in the kernel, and that openness is why a printer, a scanner or a card reader from almost anyone works on Windows. Driver signing, made mandatory for 64-bit Windows from Vista, added an identity check, and from Windows 10 version 1607 that identity had to come through Microsoft's own portal. It never checked quality beyond test suites. Security vendors went into the kernel for the reasons Microsoft lists in its July 2024 post: visibility into system-wide activity, the chance to load before malware, and tamper resistance, with the trade-off stated plainly: "Kernel drivers provide the above properties at the cost of resilience." For most of two decades the operating system offered no equivalent door in user space. Apple closed the door on its own platform in 2019; Microsoft opened its user-mode door on 17 November 2026.

What hasn't changed: a signed driver is trusted until someone revokes or blocks it, and the community catalogue of drivers known to be abused is still growing. Measured on the public LOLDrivers repository (`magicsword-io/LOLDrivers`, file `loldrivers.io/content/api/drivers.json`, commit `39c70ca`, 27 October 2026): 713 entries, 588 classed as vulnerable drivers and 125 as malicious, covering 2,082 known vulnerable driver samples. The `dbutil_2_3.sys` entry was added on 9 January 2023, more than a year after ESET's first recorded abuse. Checked 18 November 2026.

THE UNSEEN ANGLE

Most fleet tools inventory software versions. Few can tell you which *content* version each sensor is running, when it arrived, or who approved it, because that morning nobody outside the vendor approved it. CrowdStrike's own RCA lists the remedy as "staged deployment" with customer control over "where and when" content lands, which means the control exists only if the customer configures it. Ask for the content version as a fleet metric, set content rings in the console, and test the pause. Then put the pause, and who may press it, in the contract.

> *You can rent a vendor's kernel code. You cannot rent away the morning it fails; the recovery clock is always yours.*

These rows in the Trust Ledger

Halbrook Bank, branch teller desktop (Windows 11 Enterprise, about 2,500 PCs). Times are targets until measured.

Component	Who can change it	Exposure and blast radius	Patch clock	Stop, revoke, recover	Who can see it	Support end	Decision and owner
DR: EDR sensor driver, sensor version pinned N-1	EDR vendor writes and builds; Microsoft signs via its portal; Halbrook sets the sensor update policy	Kernel; one release reaches all 2,500 PCs; the N-1 pin does not cover content	Vendor release to 90% of fleet: nc, target 14 days via sensor rings	Sensor policy rollback in console; last drill: none	D (vendor console); O once fleet report built	Per vendor's sensor support policy	**Rent**, **Pin** N-1 with an end date per release; end-user computing (EUC) lead; review quarterly
DR: EDR content channel (rapid detection content)	EDR vendor, continuously ; Halbrook only through content ring settings	Changes kernel behavior on every PC without a release	T0 to T1 in minutes by design; T3 nc	Content pause and rings: configured? U. Recovery: boot media, BitLocker keys; last drill: none	U (content version not in fleet reports)	n/a	**Rent**, **Isolate**: content staged canary, 5% of branches, then all; CISO; contract clause by next renewal
DR: third-party drivers (card readers, printers, signature pads)	Five hardware vendors; Microsoft signs; blocklist by Microsoft	Kernel; local admin to load; some reachable over USB	Vendor driver to 90%: nc	Vulnerable driver blocklist on; App Control audit policy	D (Code Integrity event log)	Per hardware vendor, often unpublished	**Own** the allowlist; **Isolate** by device class; EUC lead
KN and HV: Windows kernel, VBS and memory integrity	Microsoft writes and ships; Halbrook configures VBS, memory integrity, Quick Machine Recovery	Every PC; memory integrity blocked on PCs with incompatible drivers	Monthly update to 90%: target 10 days; hotpatch months, where enrolled, need no restart	Quick Machine Recovery: off by default on Enterprise, enable and test; WinRE path documented	O (Win32_DeviceGuard via fleet query)	Windows 11 release lifecycle dates	**Rent** the code, **Own** the policy: memory integrity on for 100% of eligible PCs; EUC lead

SECURITY LENS

Who can change it? On Windows, Microsoft and every vendor whose driver you install, plus every security vendor with a content channel into its driver. On macOS, Apple, and third parties only in user space unless you lower startup security. On iOS, Apple alone. On Linux, your distribution, your hardware vendors and anyone allowed to load modules or eBPF programs. On Android, Google, the SoC vendor and the device maker.

How fast can it be fixed? Content moved from fault to revert in 78 minutes on 19 July 2024; machines took days, because the recovery clock was hands, keys and boot media. Driver revocation through the blocklist runs on a quarterly list with deliberate holds. Measure both clocks for your own fleet.

What does AI change here? Our view (as of November 2026): AI-assisted fuzzing and code review are well suited to bugs like a 20-versus-21 field mismatch, and to finding flaws in signed drivers that attackers can bring with them. Both sides get faster. The defender's gain only counts if it reaches a release, and the attacker's gain on a driver lasts until the blocklist catches up.

BUSINESS IMPLICATIONS

Security. Halbrook's sensor is the best visibility it has and one of the few authorities able to stop every branch at once. Corvane runs a sensor on 3,000 kiosks across 300 properties; for Larkfield, the question is whether its own app ships kernel components at all, and the answer should be no.

Cost. One push stopped millions of devices in one morning, by Microsoft's estimate above; Chapter Fourteen prices a day down. For Corvane, a kiosk that needs a person at the keyboard is a field visit per property, so a BitLocker key lookup that works at 3 a.m. is worth more than a faster sensor.

Visibility. You cannot see a vendor's content push before it lands unless you configure rings. Add content version and arrival time to the fleet dashboard, and memory integrity status per PC.

Value. Recovery throughput decided the cost of 19 July, not detection quality. Halbrook's customers noticed branches, not sensors.

Monday rule: list every vendor that can change kernel behavior on your devices without your release process, and for each one write the date you last tested its pause.

OUR VIEW, AS OF NOVEMBER 2026

Windows endpoint sensors will move to the user-mode platform over 2027 on the dates their vendors publish, and some kernel components will remain. Until a vendor's product is generally available on that platform for your Windows build, plan as if the sensor is in the kernel. Ask each vendor three questions in writing: which components still run in kernel mode, how content is staged by default, and how you pause it.

CHAPTER SIX

Identity and Keys: Who Can Sign, Who Can Log In



On Thursday 15 June 2023 the security operations center of the US State Department saw something odd in access to its email. The next day a custom alert rule it called "Big Yellow Taxi" fired on a log named MailItemsAccessed, and on Friday 16 June its team contacted Microsoft. The intruders had exploited nothing on the State Department's side. They had presented valid-looking sign-in tokens, signed with a Microsoft key created in 2016, and Exchange Online had accepted them. The State Department could see it only because it paid for a license tier that included that log. In the year around it, the same lesson came from other directions: an intruder who can make your systems accept a session does not need an exploit. This chapter is about the two kinds of authority that decide that: **who can sign** code or tokens your machines will accept, and **who can log in**, including the person at your help desk who can reset a sign-in. The decision it changes: every signing key and every reset path gets a row in the ID line of the Trust Ledger, with a named holder, a rotation date and a tested way to withdraw it.

Two trusts that are easy to confuse

Identity mixes two different questions, and incidents happen in the gap between them.

- **Signing trust.** Which keys can make a machine accept something: a code-signing key that makes Windows or macOS run your installer, a token-signing key that makes a cloud service accept a session, a firmware or boot key (Chapter Four). Whoever holds the private key holds the authority. The holder is often a vendor, not you.
- **Sign-in trust.** Who can prove to be a user or a workload: passwords, multi-factor authentication (MFA), passkeys, device keys, and the recovery paths behind them, including a help-desk agent who can reset a password or move an MFA method to a new

phone.

Both meet in a **token**: a signed statement that says "this session belongs to this user, with these rights, until this time". A token lets its holder use the authority it carries, as long as the service's validation accepts it. That is why a stolen session token can be worth more than a password, and why a validation bug can turn a key for one service into a key for another.

The company's **identity provider** (IdP), such as Microsoft Entra ID or Okta, is where the two trusts meet for most business systems. It issues the tokens, holds the policies and runs the reset flows. For a modern company it is a control plane in the same sense as the management consoles of Chapter Seven: one privileged admin action there can reach every application behind it. It also signs in non-human identities, such as service accounts, OAuth applications (apps granted delegated access to data) and workload tokens, none of which can answer an MFA prompt.

Storm-0558: a key, a missing check, a paid log

The case is told from two primary records: Microsoft's own posts from the Microsoft Security Response Center (MSRC), and the report of the US Cyber Safety Review Board (CSRB), dated 20 March 2024 and released by the Department of Homeland Security on 2 April 2024. Microsoft names the actor Storm-0558; the Board describes it as "assessed to be affiliated with the People's Republic of China in pursuit of espionage objectives".

What happened. Microsoft wrote on 11 July 2023 that the actor had gained access "beginning on May 15, 2023" to email accounts at about 25 organizations by forging authentication tokens with "an acquired Microsoft account (MSA) consumer signing key". The CSRB's final count: the Exchange Online mailboxes "of 22 organizations and over 500 individuals around the world", 503 personal accounts in Microsoft's report, including those of the US Commerce Secretary and the US Ambassador to China.

Why the key worked. A key out of Microsoft's hands was the first failure. Two more had to line up before it could open enterprise mail.

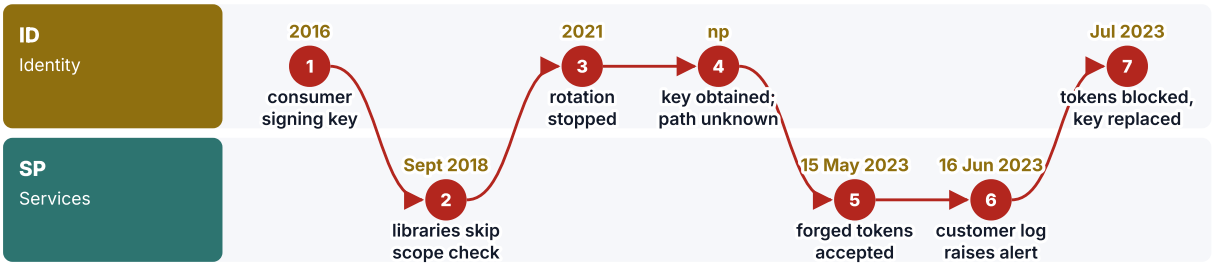
- **An old key that should have been retired.** The tokens were signed "by a key Microsoft had created in 2016". The Board found that the consumer MSA system rotated keys manually and infrequently, and that Microsoft "stopped the rotation entirely in 2021 following a major cloud outage linked to the manual rotation process". The 2016 key "was originally intended to be retired in March 2021".
- **A validation gap.** Microsoft's September 2023 post explains that it "introduced a common key metadata publishing endpoint in September 2018". Shared libraries checked that a token's signature was valid, but not whether the signing key was in scope

for that kind of account. Mail developers assumed the libraries did that check. As a result "the mail system would accept a request for enterprise email using a security token signed with the consumer key."

How the key left. This part is not settled, and the book does not settle it. Microsoft's 6 September 2023 post set out a hypothesis: a crash of the consumer signing system in April 2021 produced a dump containing the key, which was moved to a debugging environment where a compromised engineer account could reach it. On 12 March 2024 Microsoft added a correction: "we have not found a crash dump containing the impacted key material." Its leading hypothesis is still that operational errors let key material leave the secure signing environment.

FIGURE 6.1

Storm-0558 on the Ledger: an old consumer key plus a validation gap



Numbered steps in order. Red: the path the attack took. np: how and when the key left is not published.

Neither the 2016 key nor the missing scope check was enough alone; together they let forged tokens open enterprise mailboxes, and a customer's paid log found it.

Microsoft MSRC (11 Jul 2023; 6 Sept 2023, updated 12 Mar 2024); Cyber Safety Review Board, Review of the Summer 2023 Microsoft Exchange Online Intrusion (20 Mar 2024).

Who saw it. Not the key holder. The Board writes that Microsoft failed "to detect the compromise of its cryptographic crown jewels on its own, relying instead on a customer". That customer could see the intrusion because it had bought Microsoft's top government license tier (G5), which included the premium audit log. The Board's summary of the whole case is one line: "Storm-0558 was able to succeed because of a cascade of security failures at Microsoft." Microsoft blocked tokens signed with the key and replaced it; Chapter Nine returns to the revocation. In September 2024 Microsoft said it had moved Entra ID and Microsoft Account token-signing keys "to hardware-based security modules (HSMs) and virtualization-based security in Windows, with automatic rotation".

One coda: on 20 January 2025 the Department of Homeland Security ended all memberships on its advisory committees, the CSRB included (reported by Nextgov and others). The report remains published.

Logging in instead of breaking in

Storm-0558 needed a stolen key. Three other cases from 2022 and 2023 needed less: a working sign-in path and, in two of them, a person willing to approve it.

Uber, September 2022. Uber's own security update, revised through 19 September 2022, says an attacker compromised the account of an external contractor whose password "was likely purchased on the dark web" after malware infected the contractor's personal device. The attacker then triggered repeated two-factor approval requests. "Eventually, however, the contractor accepted one." Uber says it believes the attacker is "affiliated with a hacking group called Lapsus$". This is the pattern known as **MFA fatigue** or push bombing: if a second factor is a tap on "Approve", a flood of prompts can eventually produce one tap.

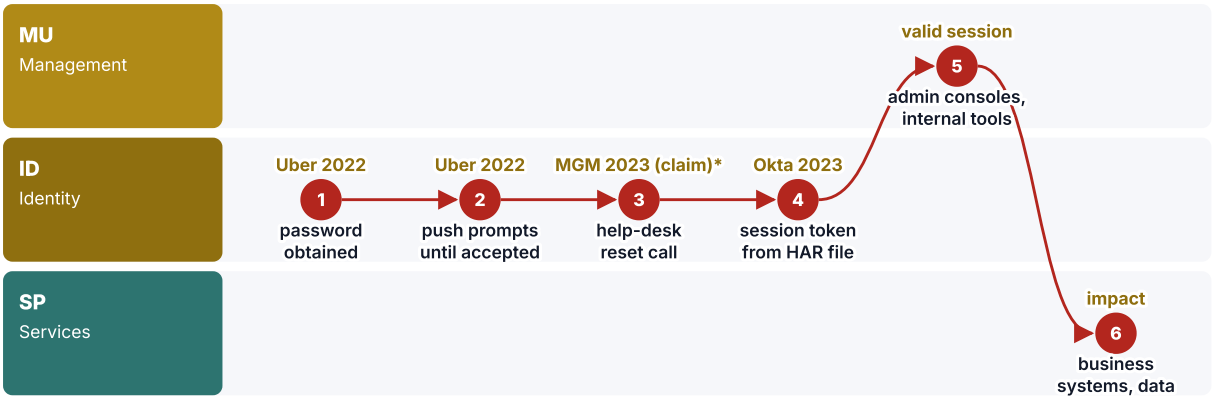
MGM Resorts, September 2023. MGM's Form 8-K of 5 October 2023 estimates "a negative impact from the cyber security issue in September of approximately $100 million" to Adjusted Property EBITDAR (earnings before interest, taxes, depreciation, amortization and rent) for its Las Vegas Strip and regional operations, plus less than $10 million in one-time expenses, and says it expects insurance to cover the impact. The filing does not say how the intruders got in. The widely repeated account, that they found an employee on LinkedIn and called the IT help desk, comes from the attackers' side, relayed by a research group and reported by the press on 14 September 2023. MGM has not confirmed it. The technique itself is on the government record: the FBI and CISA advisory AA23-320A (16 November 2023, revised 29 July 2025) describes actors who "conduct spearphising [sic] calls to convince IT help desk personnel to reset passwords and/or transfer MFA tokens".

Okta, October 2023. Okta reported on 20 October 2023 "adversarial activity that leveraged access to a stolen credential" in its support case system. Its root cause post of 3 November gives the window, "from September 28, 2023 to October 17, 2023", and the reach: files of "134 Okta customers, or less than 1% of Okta customers". Some files were HAR files, browser recordings that customers upload for troubleshooting, and they can contain session tokens. The attacker used them to "hijack the legitimate Okta sessions of 5 customers". The stolen credential was a service account stored in the support system, exposed because "an employee had signed-in to their personal Google profile on the Chrome browser of their Okta-managed laptop", where "the username and password of the service account had been saved". On 29 November Okta widened the count: the actor had also downloaded a report with "the names and email addresses of all Okta customer support system users", which

touched every customer outside its separate FedRAMP High and DoD IL4 support system. Okta's fixes included tying admin sessions to network location, so a token replayed from elsewhere forces a new sign-in.

FIGURE 6.2

Logging in instead of breaking in: three cases, one pattern



A pattern, not one intrusion. *MGM: the attackers' claim, not confirmed by MGM.

In each case the intruder used a working sign-in path (a push approval, a help-desk reset or a stolen session) rather than an exploit.

Uber security update (15 to 19 Sept 2022); MGM Resorts Form 8-K (5 Oct 2023) and press reports of the attackers' claims (labeled); Okta root cause (3 Nov 2023); FBI and CISA advisory AA23-320A (16 Nov 2023, revised 29 Jul 2025).

The record since then repeats the pattern. In January 2024 Microsoft found that Midnight Blizzard had password-sprayed "a legacy, non-production test tenant account" without MFA, then used a legacy test OAuth application with "elevated access to the Microsoft corporate environment". In June 2024 Mandiant reported about 165 organizations notified after an actor signed in to Snowflake customer instances, which did not require MFA, with credentials stolen by infostealer malware. In April 2025 Marks & Spencer was hit; on 21 May it estimated a cost of about £300 million to its 2025/26 operating profit before mitigation, and its chairman told a House of Commons committee on 8 July that entry came through "sophisticated impersonation" involving a third party.

In the published accounts, none of these intruders needed a software flaw in the IdP. Each used a path the company had built on purpose: a push approval, a reset by phone, a support upload. The advisory's own mitigation is phishing-resistant MFA: "Implement FIDO/WebAuthn authentication or Public Key Infrastructure (PKI)-based MFA", which it says is "not susceptible to push bombing or SIM swap attacks." The same property defeats **adversary-in-the-middle** phishing, in which a proxy page relays a real sign-in and keeps

the session cookie; Microsoft reported in July 2022 that one such campaign had targeted more than 10,000 organizations, and that the stolen cookie let attackers in "even if the target's MFA is enabled". A FIDO credential is bound to the real site, so the proxy has no sign-in to relay; a session stolen later from the device itself is a separate problem. The counterexample is on record: in August 2022 the SMS phishing wave that reached Twilio also texted at least 76 Cloudflare employees, three entered their passwords, and the attackers "could not get past the hard key requirement" (Cloudflare, 9 August 2022). For push prompts, Microsoft's answer is number matching, now "enabled for all Authenticator push notifications": the user types a number shown on the sign-in screen, so a blind tap approves nothing. The advisory has no equally simple fix for the help desk, because the help desk exists to let people back in.

Keys in hardware, across five platforms

For high-assurance accounts, the strongest commonly deployed sign-in credentials are device-bound keys held in hardware. A **passkey**, in the FIDO Alliance's definition, is "an authentication credential based on FIDO standards" (WebAuthn between the site and the browser, CTAP between the device and an external authenticator such as a security key) that the user unlocks the way they unlock the device. Each passkey is bound to one site or app, so it cannot be typed into a fake page. Passkeys come in two kinds, and the difference matters for the help desk: **synced** passkeys are copied across a user's devices by a provider's cloud with end-to-end encryption, and **device-bound** passkeys, including those on security keys, never leave one device. A synced passkey depends on the security of the Apple or Google account that syncs it, so its recovery path is the help-desk problem in another form.

Decision point	Windows	macOS	iOS	Linux	Android
Where device keys live	TPM 2.0 (a Windows 11 hardware requirement); Windows Hello keys in software if no TPM and no policy forbids it	Secure Enclave on Apple silicon	Secure Enclave	TPM where present; depends on distribution and tooling	TEE-backed Keystore; StrongBox secure element on some devices with Android 9 or later
Passkeys	Windows Hello as a FIDO2 authenticator; security keys	Platform passkeys, synced or on security keys		Security keys, or a phone through the cross-device flow	Platform passkeys, synced or on security keys
What the company must set	Policy to require hardware-generated Hello keys	MDM policy on which passkey providers and accounts are allowed		Your own enrollment and pluggable authentication module (PAM) configuration	Device model list with StrongBox where you need it
Firmware and silicon	TPM firmware is an OEM and chip vendor row (Chapter Three)	Secure Enclave firmware ships with Apple updates		As Windows on the same hardware	StrongBox requires its own CPU, secure storage and true random-number generator

Two documented details change fleet decisions. Microsoft's Windows Hello for Business documentation says keys are generated "in hardware (TPM 1.2 or 2.0) or software, based on the configured policy setting", and that "to guarantee that keys are generated in hardware, you must configure a policy setting." Android's documentation says StrongBox is available on devices "running Android 9 (API level 28) or higher" that include it, and requires "its own CPU", secure storage and a true random-number generator. In both cases the strongest protection is available but not automatic.

The signing key you own

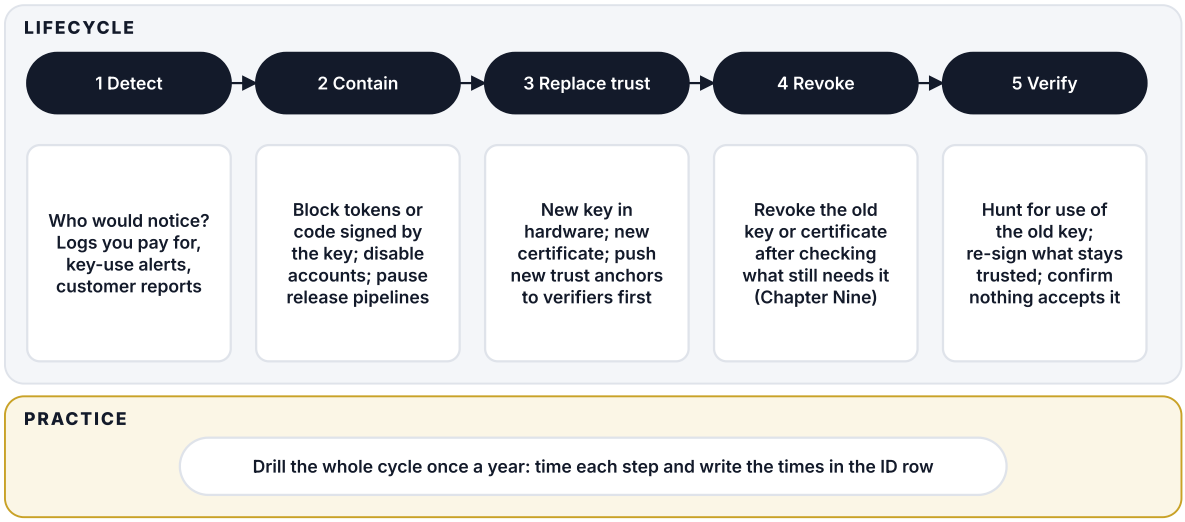
Larkfield Software ships an Electron-style app to about 400,000 desktops, signed with its own code-signing certificate and updated through its own channel. That key is among the most valuable things Larkfield owns, and its customers never see it: anyone who holds it can ship code that customer machines will run as Larkfield's.

The industry has already moved the floor. The CA/Browser Forum's baseline requirements for publicly trusted code-signing certificates, with a compliance date of 1 June 2023, say CAs "SHALL ensure that the Subscriber's Private Key is generated, stored, and used in a suitable Hardware Crypto Module". A key in a file on a build server is no longer an option for a new public code-signing certificate, though a private CA can still issue one to a software key. That fixes storage. It does not answer who can ask the hardware to sign, from which pipeline, with what approval and what log.

Storm-0558 is the cautionary case for any key holder, Larkfield included: a key due to retire in 2021 was still valid in 2023 because rotation had stopped, and a verifier accepted it outside its scope. A key inventory is therefore a discovery exercise, not a list you already have. For each key, record who holds it, where it is stored, what accepts it, when it was last rotated, and how long it would take to stop every verifier from trusting it. Figure 6.3 is the lifecycle to drill; Chapter Eight follows signed code that was still malicious, and Chapter Nine follows revocation.

FIGURE 6.3

Key-compromise lifecycle: five steps, each with a clock



A leaked signing key is not dealt with when a new one exists; it is dealt with when nothing anywhere still accepts the old one, and only a drill tells you how long that takes.

Author's diagram based on the Storm-0558 record (MSRC 2023 to 2024; CSRB, March 2024) and the CA/Browser Forum code-signing requirements (hardware key storage from 1 June 2023).

HOW IT GOT THIS WAY

For most of the 2000s, identity at work lived on the domain-joined machine: an Active Directory account, a Kerberos ticket, a password and a badge. As work moved to cloud services, the sign-in moved to a cloud identity provider and the session became a signed token any internet service could check. Key hygiene and help-desk verification did not move as fast. The CSRB found that Microsoft's own consumer identity system "had no process for automated signing key rotation" in its early form, and the attacks of 2022 and 2023 used help desks and push approvals built for convenience in an era of passwords.

What hasn't changed: a person on a phone can still be the weakest authority in the chain. The FBI and CISA advisory AA23-320A, as revised on 29 July 2025, still lists calls to help-desk staff to reset passwords or transfer MFA among the techniques in use. Checked 18 November 2026.

THE UNSEEN ANGLE

Identity is now a layer of the operating system in all but name. Ask who holds each of the company's signing keys and the answer is usually a search, not a list. Ask how long a caller with an executive's birth date needs to get that MFA reset, and the answer is a guess. The CSRB report makes a second point that is easy to miss: the State Department found the intrusion because it paid for a log. Visibility in the ID row is often a price-list item, and the list moves only after a public failure: on 19 July 2023 Microsoft said it would give standard-tier customers detailed mail-access logs and raise default retention from 90 to 180 days, and the mail-access log reached the standard tier in public preview in May 2024. Check which sign-in and mailbox logs your license tier includes, and whether they are switched on, before you need them.

> *Intruders often log in rather than break in. The help desk that lets your people back in is an authentication path, and it belongs in the Ledger.*

These rows in the Trust Ledger

Halbrook Bank (sign-in for 2,500 branch staff and advisors) and Larkfield Software (its code-signing key). Times are targets until measured. Row codes, evidence tags (O, D, C, I, U) and the marks np and nc are as defined in Chapter One.

Component	Who can change it	Exposure and blast radius	Patch clock	Stop, revoke, recover	Who can see it	Support end	Decision and owner
ID: Halbrook IdP tenant and its token-signing keys	IdP vendor holds signing keys and code; Halbrook admins hold policy	Every app behind single sign-on; one admin action reaches all staff	Vendor fixes: np; Halbrook policy changes: same day	Session revocation tested? U. Break-glass accounts in hardware keys; last drill: nc	D (vendor logs); O only for log tiers Halbrook pays for	Contract term	**Rent**; **Own** the policy and the log tier; CISO; review quarterly
ID: staff sign-in, passkeys on TPM 2.0 and phones	Microsoft (Windows Hello), Apple and Google (phones); Halbrook policy	Phishing pages, push prompts	Policy rollout to 90%: target 60 days	Remove a passkey and revoke sessions in minutes; measured: nc	O (IdP sign-in logs)	Device support dates	**Own**: hardware-backed passkeys required, push approvals off; identity lead
ID: help-desk reset path	Help-desk staff and outsourced service desk	A phone call; reaches any account, including admins	Procedure change: days	Callback to a number on file, manager approval for admins, 24-hour hold on new MFA for privileged roles; last test: none	I (ticket notes only)	n/a	**Own**, **Isolate**: admin resets only in person or by video with a second approver; service desk manager
ID: Larkfield code-signing key	Larkfield release engineering; the CA that issues the certificate	Signs installers and updates for about 400,000 desktops	Rotation: not since issue (np)	Key in a hardware module; revoke and re-issue drill: none; time to re-sign and ship: nc	D (HSM audit log), if reviewed	Certificate expiry date	**Own**: signing only from the release pipeline with two approvals; new key at each certificate renewal; head of release, CISO approves

SECURITY LENS

Who can change it? Your IdP vendor holds the token-signing keys; your admins hold policy; your help desk can reset any user; your release team controls the code-signing key in its hardware module, and your CA can revoke the certificate. Where hardware is present and policy requires it, the device key is held by hardware you rent: TPM, Secure Enclave or StrongBox; on Linux it depends on your configuration.

How fast can it be fixed? Policy fixes in the IdP take effect in minutes. Replacing trust in a key takes as long as the slowest verifier that still accepts the old one: the 2016 key stayed valid for more than two years after its planned retirement.

What does AI change here? Our view (as of November 2026): voice and text generation make a convincing help-desk caller cheaper to produce, so a reset path that relies on a person recognizing a voice or a story is weaker than it was. Phishing-resistant credentials and callbacks to known numbers do not depend on that judgment.

BUSINESS IMPLICATIONS

Security. Halbrook's riskiest authority is not its firewall but its service desk. Where Corvane's front-desk staff share a tablet account across shifts, one stolen session becomes many; each person needs their own sign-in. Larkfield's code-signing key can make 400,000 desktops run anything.

Cost. The hardware for strong keys is already paid for: TPM 2.0 is a Windows 11 hardware requirement, and Secure Enclaves are in the Apple devices Halbrook issues. The rollout is not free (the Ledger above budgets 60 days), but recovery after a sign-in compromise costs more: MGM's filing put one month's impact at about $100 million to a single earnings measure, and Marks & Spencer estimated about £300 million.

Visibility. A forged token or a reset account looks like a valid sign-in. Pay for the logs that would show it, as the State Department did, and alert on MFA method changes for privileged accounts.

Value. Staff and customers experience identity as "can I get in", so the speed of a legitimate reset is a value line, not only a risk line.

Monday rule: ask your help desk to reset a test executive's MFA from an unknown number, and write down how long it took and who approved it.

OUR VIEW, AS OF NOVEMBER 2026

Passwords will keep shrinking as a sign-in factor, and the attack will move to the recovery paths that remain: help desks, account recovery and session tokens. A company that moves staff to passkeys should fund the reset procedure next, because that is where an intruder will go.

CHAPTER SEVEN

The Management Plane: Everyone Who Can Push



On 11 March 2026 Stryker Corporation, a US medical technology company, identified a cybersecurity incident that, in the words of its Form 8-K filed the same day, "resulted in a global disruption to the Company's Microsoft environment". The next day it reported that order processing, manufacturing and shipping were disrupted. On 18 March the US Cybersecurity and Infrastructure Security Agency (CISA), without naming the victim, published an alert about "malicious cyber activity targeting endpoint management systems of U.S. organizations" that "misuses legitimate endpoint management software", and asked every organization to require a second administrator's approval before a device wipe. Nothing in that alert is about a flaw in a device. It is about the console that manages the devices. This chapter changes one decision: which consoles you treat as production authorities over your fleet, and how many people must agree before one of them acts on every device at once.

Everyone who can push

The **management plane** is every system that can change many of your devices without anyone touching them: install or remove software, run a script, change a policy, update firmware, lock or erase a device. Most companies run several, bought at different times by different teams. The common ones:

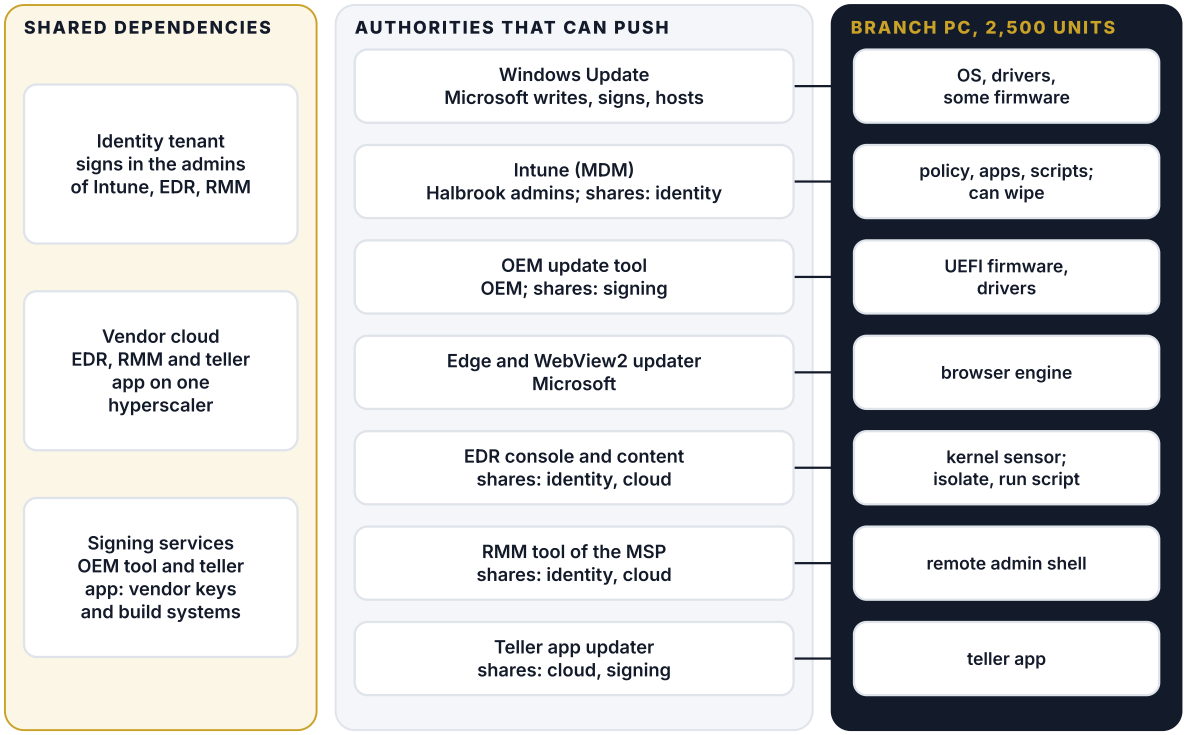
- **MDM or UEM.** Mobile device management, now usually sold as unified endpoint management: Microsoft Intune, the Apple MDM protocol and the products built on it, Android Enterprise through an enterprise mobility management (EMM) product. They set policy, deploy apps and scripts, and can wipe.
- **RMM and remote-support tools.** Remote monitoring and management software gives a technician, often at a managed service provider (MSP), a remote shell, a script runner and a file transfer on every enrolled machine.

- **Security consoles.** The EDR console and its content channel (Chapter Five) can isolate, kill, quarantine and run scripts on every sensor.
- **Update channels.** The OS vendor's update service, the OEM's firmware and driver tool, the browser updater, each application's own updater, the package repositories on Linux.
- **The systems behind them.** The identity provider that signs in the consoles' administrators, the CI/CD pipelines that build what they push, the code-signing services, the certificate authorities, the cloud tenants that host the consoles.

Chapter One gave the procedure for counting update authorities. Run it on one Halbrook Bank branch PC and the count is seven channels that can change the device (Figure 7.1; the Microsoft Store is turned off by policy on that image). The count is the easy part. The harder part is the left column of the figure: the shared dependencies behind the channels. Three consoles that sign in through the same identity tenant are, for an attacker who owns that tenant, one console. Two vendors whose update services sit in the same cloud region can fail together. Counting channels without their dependencies overstates how independent your authorities are.

FIGURE 7.1

Seven authorities can push to one Halbrook branch PC, and three shared dependencies sit behind them



Counting push channels is not enough: when several channels share one identity tenant, one cloud or one signing service, a single failure behind them reaches every channel at once.

Author's diagram. Halbrook Bank is an invented running company; the inventory is illustrative, built with the update-authority procedure from Chapter 1.

The plane also reaches into the trusted computing base, the set of components a device must trust to be secure. Some of that base now lives in vendors' clouds. Google's Play Integrity API, which many Android banking apps use to decide whether a device is genuine, returns its verdicts from Google's services, and Google's own documentation tells developers to "plan in advance" for "the unlikely event of a large-scale Play Integrity API outage". An attestation service, an OEM update content delivery network or an EDR vendor's cloud is part of your management plane even if no one in your company can log in to it.

Kaseya VSA: the console as the delivery system

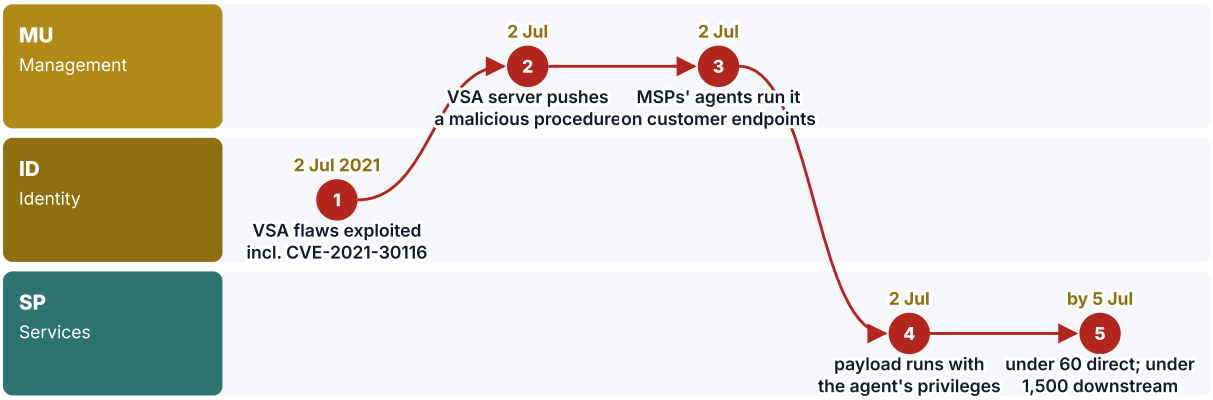
On Friday 2 July 2021 Kaseya's incident team learned of an attack on VSA, its remote monitoring and management product, used mostly by MSPs to run their customers' computers. Kaseya told on-premises customers to shut down their VSA servers immediately and took its own software-as-a-service (SaaS) servers offline as a precaution. By 5 July

Kaseya reported that fewer than 60 of its customers, all running VSA on-premises, had been directly compromised, and that the impact reached "fewer than 1,500 downstream businesses". Those are the customers of the MSPs, many of whom had never heard of Kaseya.

The mechanism matters more than the count (Figure 7.2). The attackers did not break into 1,500 businesses. They got past the login of internet-reachable VSA servers and then used the product's normal function: send a procedure to every managed agent, and the agents run it with the privileges they were given to install software. One action on one server reached every endpoint each affected MSP managed. On 4 July CISA and the FBI asked affected MSPs to require multi-factor authentication on every account they controlled, to restrict RMM traffic to known address pairs, and to put RMM administrative interfaces behind a VPN or firewall.

FIGURE 7.2

Kaseya VSA, 2 July 2021: one management server, three tiers of victims



Numbered steps in order. Red: the path the attack took. Counts are Kaseya's.

The attackers did not break into each business; they used the management server's own job of pushing software to its agents.

Kaseya incident notices (2 to 26 July 2021); DIVD case DIVD-2021-00011; CISA-FBI guidance (4 July 2021).

The clock is the second lesson. The Dutch Institute for Vulnerability Disclosure (DIVD) had reported a set of VSA vulnerabilities to Kaseya on 6 April 2021. Kaseya fixed several of them in April and May. Its case file entry for 26 June 2021 reads: "Kaseya released 9.5.7 on SaaS Resolving CVE-2021-30116 and CVE-2021-30119"; the first is the credential-leak flaw, and the fix was **for SaaS** only. The on-premises servers, run by MSPs on their own machines,

did not yet have that fix when the attack came six days later. Kaseya published the on-premises release notes on 11 July, and CISA's notice of 12 July says Kaseya released VSA 9.5.7a for on-premises customers.

Clock point	Date (2021)	Source
T0: vendor aware (DIVD report)	6 April	DIVD case DIVD-2021-00011
T1 for SaaS: 9.5.7 fixes CVE-2021-30116	26 June	DIVD case file
Exploited; on-premises servers told to shut down	2 July	Kaseya notice
T1 for on-premises: 9.5.7a	11 July	Kaseya notice; CISA (12 July)
Universal decryptor obtained / announced	21 / 22 July	Kaseya notice
T2, T3 on MSP servers	np	No public fleet figure

The same shape returned in February 2024. ConnectWise published a critical bulletin for its ScreenConnect remote-access product on 19 February, after an independent researcher reported the flaws on 13 February. One of them, CVE-2024-1709, let an attacker with network access to the management interface create an administrator account. ConnectWise patched its cloud instances itself, "within 48 hours" by its account; on-premises partners had to update by hand, and ConnectWise began suspending unpatched on-premises servers. CISA added the CVE to its Known Exploited Vulnerabilities catalog on 22 February. In both cases the vendor's own cloud copies were fixed first, and the copies customers ran themselves waited for the customers.

The update server can be the door too. On 23 October 2025 Microsoft released an out-of-band fix for CVE-2025-59287 in Windows Server Update Services (WSUS), the server many companies use to pass Microsoft's updates on to their own PCs; NVD's entry says it "allows an unauthorized attacker to execute code over a network", and CISA noted that "a prior update did not fully mitigate" it before adding it to the KEV catalog on 24 October. Whoever takes over that server sits on the push path to every PC behind it.

Stryker: the console's own commands

The 2026 case is different in one way that matters: the public record describes no product flaw at all. Stryker's filings describe a disruption to its Microsoft environment, then, on 23 March, a corporate network disruption "including but not limited to" that environment, and

a letter from Palo Alto Networks Unit 42 describing a compromise "involving impacts to Stryker's Entra ID environment, servers, and workstations". Early on Stryker said it had "no indication of ransomware or malware"; later it said the attacker "used a malicious file to run commands" to hide its activity. On 9 April Stryker amended its 8-K under Item 1.05 to say the incident had a material impact on its operations and on its first-quarter 2026 results. Press reports at the time, citing people close to the investigation, said the attackers used the device-management console's own remote-wipe command on a very large number of devices; Stryker's filings do not say how devices were affected, and this book does not repeat the reported counts.

What the record does show is CISA's response. Its alert of 18 March asks for three things, all about the console's administrators and none about the endpoints: least-privilege roles that limit both what an administrator can do and which devices they can act on; phishing-resistant MFA and conditional access for privileged console actions; and **multi-admin approval**, a rule that a second administrator must approve sensitive changes such as device wipes and changes to applications, scripts, roles and configurations. Microsoft's documentation for Intune Multi Admin Approval (updated 30 July 2026) lists the protected areas: app deployments, compliance and configuration policies, the wipe, retire and delete device actions, role-based access control, Windows scripts, and device categories. It notes that "an administrator can't approve their own requests", that even a Global Administrator's change needs a different approver, and that the rule also intercepts calls made by automation through the Graph API.

Two details on the same page are the ones a fleet owner should read twice. First, the rule is off until someone configures it, and it needs at least two administrator accounts. Second, an access policy can exclude named enterprise applications, and Microsoft warns that "each exclusion creates a gap in your approval workflow". A two-person rule with an exempt automation account is a one-person rule for whoever holds that account.

When a government closes the channel: Kaspersky

Governments are authorities over the management plane too. On 20 June 2024 the US Commerce Department's Bureau of Industry and Security (BIS) announced a Final Determination on Kaspersky Lab, Inc. under its information and communications technology rules. From 20 July 2024 Kaspersky could not enter new agreements with US persons. From 12:00 AM EDT on 29 September 2024 it was barred from "providing any anti-virus signature updates and codebase updates" in the United States. BIS's stated reasons were about the update channel itself: the company's "broad access to, and administrative privileges over, customer information", and the "capability or opportunity to

install malicious software and withhold critical updates". The same release said users who kept the software "will not face legal penalties" but assume "all the cybersecurity and associated risks of doing so". BIS also added three Kaspersky entities to the Entity List.

The determination did not touch a single device. It ended the vendor's right to update them, which for an antivirus product means its protection decays from a known date (Figure 7.3). What happened next shows how much a security vendor's channel can do. Kaspersky told TechCrunch (reported 26 September 2024, secondary) that a migration of its US consumer customers to another vendor's product had started at the beginning of September and that on Windows it "was done automatically", so users "would not experience a gap in protection"; users who had not registered an email address said they had not known it was coming. The last thing the channel did before it closed was to remove its own product and install someone else's.

FIGURE 7.3

Kaspersky in the US, 2024: a government closed the update channel, and the vendor's last push replaced its own product

Date (2024)	Who acted	What happened	What it did to installed devices
20 June	US Commerce Department (BIS)	Final Determination announced; three Kaspersky entities added to the Entity List	Nothing yet; users told they face no legal penalty for keeping the software
20 July	BIS rule takes effect	No new agreements with US persons	None; existing installations keep updating
Early September	Kaspersky (its statement)	Migration of US consumer customers to another vendor's product begins	Product replacement planned through Kaspersky's own channel
Week of 19 September	Kaspersky's update channel (press reports)	On Windows the migration ran automatically; some users said they had no warning	Kaspersky software removed and another vendor's product installed
29 September	BIS rule takes effect	Signature and codebase updates barred in the US	Any remaining installation stops receiving updates

The government's acts and the vendor's acts are separate rows: the Final Determination ended the right to update; the replacement of installed software was the vendor's own use of its channel.

Source: BIS press release and Kaspersky prohibition page (20 June 2024); Kaspersky statement as reported by TechCrunch (26 September 2024), secondary.

For a CTO the lesson is not about one vendor or one country. Any authority with a push channel can be stopped, sold, sanctioned or ordered, and its last push may not be the one you expected. Export controls and lawful orders belong in the "who can change it" column next to the vendor.

Five platforms, five management channels

Each platform gives the management plane a different shape. The table keeps only the differences that change a decision; Chapter Two has the full platform profiles.

Decision point	Windows	macOS	iOS	Linux	Android
Built-in management channel	The OS's MDM interface (used by Intune and other UEM products); Group Policy and Configuration Manager on domain PCs	Apple's MDM protocol; supervision through Automated Device Enrollment	Apple's MDM protocol; supervision through Automated Device Enrollment or Apple Configurator	No OS-vendor MDM; configuration tools and the distribution's repositories	Android Enterprise through an EMM: fully managed device or work profile
Erase from the console	Yes (Intune wipe action)	Yes (Erase Device command; Apple silicon or T2 conditions apply)	Yes (Erase Device command)	Only what your own agent can do	Fully managed: whole device; work profile: the work profile
Two-person rule for mass actions	Intune Multi Admin Approval, if configured	Depends on the MDM product, not on Apple	Depends on the MDM product	Depends on your tooling	Depends on the EMM product
Other authorities that push to the same device	Windows Update, OEM tool, Microsoft Store, app updaters, RMM, EDR	Apple software update, app updaters, EDR	Apple updates, App Store	Distribution and vendor repositories, Snap or Flatpak stores, EDR	OEM updates, Google Play system updates, Play Store
Firmware and silicon	OEM tools and Windows Update can deliver UEFI updates	Firmware ships inside Apple's OS updates	Firmware ships inside Apple's OS updates	fwupd and the Linux Vendor Firmware Service (LVFS) where the OEM publishes; otherwise the OEM tool	The OEM's over-the-air (OTA) image

Apple's model concentrates authority: one vendor builds the OS, signs it, ships the firmware with it and defines the management protocol, and the customer chooses the MDM product. Linux spreads it: the distribution, each third-party repository and your own configuration tool are separate authorities, and the "erase" button exists only if you built it. Windows and Android sit between, with the OS vendor, the OEM and a management product of the customer's choice all holding push rights.

Running the inventory on a Corvane kiosk

Corvane Hotels has about 3,000 Windows IoT check-in kiosks across 300 properties. Its central IT team runs Intune and the EDR console. Many properties also contract a local MSP, which installed its own RMM agent for remote support. Run the Chapter One procedure on one kiosk image and write down three things for each channel: who holds the authority, what one action can reach, and what stops it.

- **Windows Update**, scheduled through Intune rings: Microsoft writes and signs; Corvane's admins set the rings. One ring change reaches every kiosk in that ring.
- **Intune**: any Corvane account with the Intune Administrator role can wipe every kiosk. Until Multi Admin Approval is configured, one stolen account is enough.
- **The EDR console and its content channel**: two authorities in one, the vendor's content pushes and Corvane's console actions (Chapter Five for the content).
- **The property MSPs' RMM tools**: up to 300 separate consoles, each run by a company Corvane audits once a year at best, each able to run a script as an administrator on that property's kiosks.
- **The kiosk app's own updater**: Corvane's integrator builds, signs and pushes it.

The count is five channel types and, once each MSP is counted separately, a few hundred consoles. Two channel types, the MSPs' RMM tools and the kiosk app's updater, have no two-person rule at all; Intune has one that nobody has configured; and Corvane cannot see the administrators of the RMM consoles. It goes into the Ledger below as written.

HOW IT GOT THIS WAY

Central management was built to save technicians' time, and it did. Microsoft's Systems Management Server, the ancestor of Configuration Manager, officially launched on 7 November 1994, so that one team could inventory and update thousands of Windows PCs without visiting them. On 8 April 2010 Apple previewed iPhone OS 4 with a Mobile Device Management service that third-party servers could use to "wirelessly configure, query and even wipe or lock managed iPhones". RMM tools gave MSPs the same reach over many small customers at once. Each step moved authority from the person at the device to whoever holds the console, and from many consoles to fewer. The time saved was real; the concentration was rarely priced.

What hasn't changed: whoever controls the console controls the fleet. In July 2021 the delivery mechanism of the Kaseya attack was the RMM product's own procedure push; in March 2026 CISA's answer to an attack on a company's Microsoft environment was to ask for a second approver on the console's own wipe command. Same lesson, 56 months apart (Kaseya notices, 2 to 26 July 2021; CISA alert, 18 March 2026).

THE UNSEEN ANGLE

Look at what CISA asked for after the 2026 attack: least privilege in the console, and phishing-resistant MFA and conditional access in the identity provider "to block unauthorized access to privileged Intune actions". The console's protection is the identity tenant's protection. The same tenant usually signs in the EDR console, the RMM portal and the cloud subscriptions. So a more useful number in a management-plane inventory than the count of consoles is the count of **independent** ones: those that share no identity tenant, no administrator, no cloud account and no signing service with the others. Many companies will find the honest answer is one. Source: CISA, "CISA Urges Endpoint Management System Hardening After Cyberattack Against US Organization" (18 March 2026).

> *Every console that can act on the whole fleet raises two questions: how many people can use it, and whether one of them can act alone.*

These rows in the Trust Ledger

Corvane Hotels, check-in kiosk on Windows IoT Enterprise, about 3,000 devices. Clock cells are placeholders until measured from Corvane's own telemetry (Appendix D). Row codes and the tags O, D, C, I, U, np and nc are as defined in Chapter One.

Component	Who can change it	Exposure and blast radius	Patch clock	Stop, revoke, recover	Who can see it	Support end	Decision and owner
MU: Intune tenant (policy, apps, scripts, wipe)	Corvane admins; signs in through Corvane's identity tenant	Console login from the internet; one action reaches all 3,000 kiosks	n/a (service); console changes immediate	Multi Admin Approval not configured; no drill	D: Intune audit log, if exported	Contract term	**Rent** the service, **Own** the approval rule: two approvers for wipe, scripts, apps, roles; end-user computing lead; review quarterly
MU: Windows Update rings	Microsoft writes and signs; Corvane sets rings in Intune	Every kiosk per ring	T1 Patch Tuesday; T2, T3 nc	Ring pause per ring; rollback per update; drill np	O: Intune update reports	Per Windows IoT LTSC lifecycle (Chapter Eleven)	**Rent**; **Own** the ring schedule; end-user computing lead
MU: property MSPs' RMM agents	Up to 300 MSPs and their technicians	Remote shell as administrator on that property's kiosks; vendor flaws hit every MSP at once	Per RMM vendor; on-premises servers lag cloud ones	Remove agent; no central kill	U: Corvane cannot see MSP console logins	Per MSP contract	**Isolate**: no RMM agent on kiosks; MSP support through Corvane's own remote tool with session approval; CISO; by Q2
MU and DR: EDR console actions	Corvane security team; vendor content channel (Chapter Five)	Isolate, kill, run script on every kiosk	Per vendor	Console role review; content staging per property group	D: vendor console	Per contract	**Rent**; **Own** console roles and staging; security operations lead
MU: kiosk app updater	Corvane's integrator builds, signs, pushes	All kiosks per release	Integrator's release train	Kill switch in app; rollback to previous build	O: app telemetry	Integrator contract	**Own** the release gate: staged by property group; product owner

SECURITY LENS

Who can change it? Every administrator of every console that manages the device, every vendor that runs one of those consoles as a service, every MSP technician with an RMM login, and, as the Kaspersky determination shows, a government that can close a vendor's channel. Write them down by name and count the independent ones.

How fast can it be fixed? A vendor fixes the copy of its console it runs in its own cloud first: ConnectWise reported mitigating its cloud instances within 48 hours in 2024, and Kaseya's SaaS fix preceded the on-premises one by 15 days in 2021. A console you host yourself is on your clock. A console you never configured to need two approvers cannot be fixed by any patch.

What does AI change here? Microsoft's Multi Admin Approval page now shows suggestions from a "Change Review Agent" next to Windows PowerShell script requests: an AI agent is already advising the human who approves a fleet-wide change. Our view (as of November 2026): that agent cannot approve anything, but it shapes the decision of the person who can, so it belongs in the AG row. Keep the human decision, log what the agent recommended, and never let an agent be the second approver.

BUSINESS IMPLICATIONS

Security. Corvane's riskiest authority is not a kiosk flaw; it is a few hundred RMM consoles it cannot see. Halbrook's seven channels in Figure 7.1 share one identity tenant, so its Chapter Six work on admin accounts is also its management-plane defence. Larkfield is itself a push channel on 400,000 customer desktops, and its customers' security teams should be asking it the questions in this chapter.

Cost. A second approver costs minutes per change. A wipe of every device in a tenant costs a rebuild of every device and the business days in between; Stryker reported a material impact on a quarter. Removing RMM agents from Corvane's kiosks costs a support process change, not a license.

Visibility. Export every console's audit log to a place its own administrators cannot delete, and alert on mass actions: more than a set number of wipes, retires or script deployments in an hour. Ask each MSP and SaaS vendor for its console's admin list and login logs in the contract.

Value. Central management is why a team of ten can run 3,000 kiosks. The goal is not fewer consoles at any price; it is fewer independent ways to act on every device at once, each with a brake.

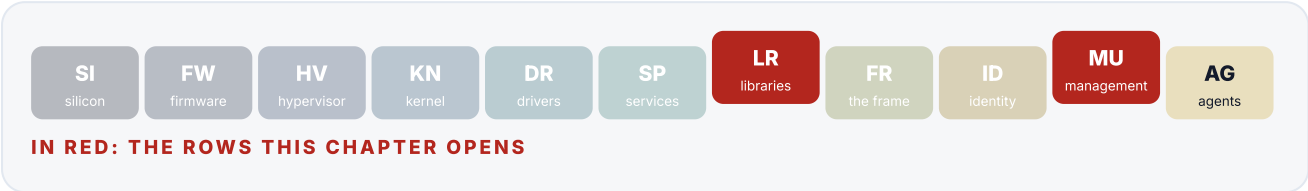
Monday rule: list every console that can wipe, script or install on more than 100 of your devices, then turn on a two-person rule for mass actions in each one that supports it, and write down the ones that cannot.

OUR VIEW, AS OF NOVEMBER 2026

Two-person approval for destructive and fleet-wide actions will become a default expectation in endpoint-management contracts and audits, the way MFA for administrators did. The harder work is the exemptions: automation accounts, break-glass accounts and vendor support access. Expect auditors to ask for that exemption list, and expect the attacks to move to it. Companies that also remove standing console rights, activating administrator roles only for a task and a time, will have fewer accounts to defend.

CHAPTER EIGHT

The Software Supply Chain: Signed Is Not Safe



At 08:51 Pacific time on 29 March 2024, a software engineer named Andres Freund posted to the oss-security mailing list under the subject "backdoor in upstream xz/liblzma leading to ssh server compromise". He had noticed that SSH logins on his Debian test machines were using too much CPU. He measured it: a failed login took 0.299 seconds before a library update and 0.807 seconds after. Chasing half a second, he found that the release tarballs of xz-utils 5.6.0 and 5.6.1, a compression library present in most Linux systems, carried code built to compromise the SSH server. The releases were made by a maintainer of the project, under that maintainer's legitimate release authority. No signature failed. No control designed to catch it did. This chapter changes one decision: what you verify about software before you let it in, given that a valid signature tells you who shipped it and nothing about whether you should run it.

Author, builder, signer, publisher

Software reaches a device through a chain of authorities. An **author** writes the code. A **builder** turns source into a binary or a package. A **signer** applies a key that tells the device who vouches for the binary. A **publisher** puts it in a registry, a repository or an update channel, and a **pusher** (Chapter Seven) delivers it. In open source the author and the publisher may be one volunteer; in a commercial product each step may be a different team or a different company.

The person who writes the code is often not the most important authority in that chain. Whoever controls the build can add code that no author wrote. Whoever holds the signing key can make any binary look like the vendor's. A **signature** answers the question "who" (this binary was approved by the holder of this key) and nothing about the question "should". Three of this chapter's four cases used valid trust: a real build system, a real signing certificate, a real maintainer account, a real release process. The fourth, Log4Shell,

shows how far one ordinary flaw travels once trusted code is everywhere. Chapter Thirteen's Stuxnet drivers and Chapter Five's CrowdStrike content file were trusted for the same reason: they arrived through a channel the device was built to accept.

The industry's answers are **SBOMs** (software bills of materials: lists of the components inside a product) and **provenance** (signed records of where and how a binary was built), formalized in frameworks such as SLSA (Supply-chain Levels for Software Artifacts) and tools such as Sigstore, open tooling and services with which a developer or build pipeline signs a release and records the signature in a public log. They are worth having, and the cases below show their limits. SLSA's version 1.0 specification, published 19 April 2023, says plainly that it "does not address source threats", which include "an authorized individual introducing an unauthorized change", in other words an insider. Provenance proves origin, not intent.

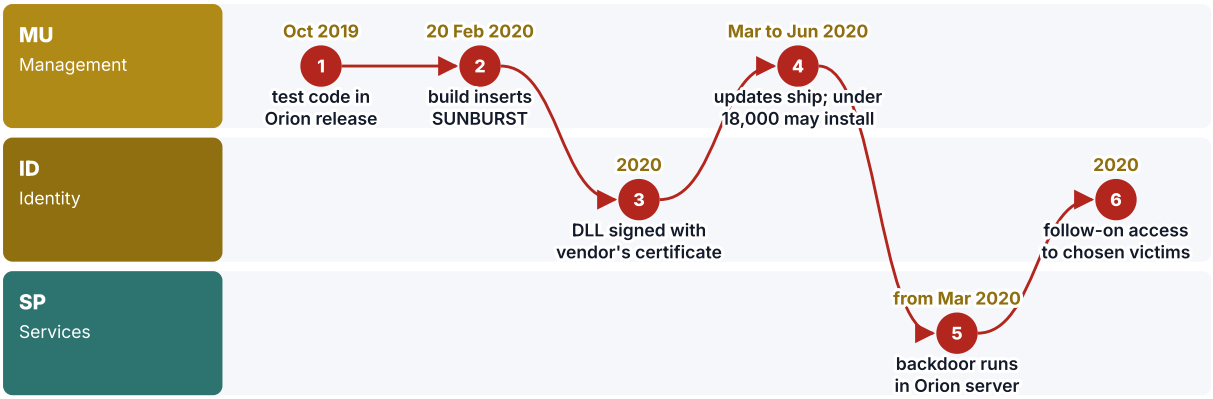
SolarWinds: the build system as the attacker's workstation

On 13 December 2020 CISA issued Emergency Directive 21-01, ordering federal agencies to disconnect or power down SolarWinds Orion products, network monitoring software used across government and industry. The next day SolarWinds filed a Form 8-K. It said a vulnerability had been inserted into Orion updates "released between March and June 2020", that it "was introduced as a result of a compromise of the Orion software build system and was not present in the source code repository", and that "the actual number of customers that may have had an installation of the Orion products that contained this vulnerability" was "fewer than 18,000", out of about 33,000 active Orion maintenance customers it had notified on 13 December. Orion accounted for about 45% of SolarWinds' revenue in the first nine months of 2020.

SolarWinds' own investigation, published 11 January 2021, gave the build-side timeline. It "begins in September 2019"; the October 2019 Orion release contained modifications that tested the attackers' ability to insert code; the injection inserted the backdoor, named SUNBURST, into builds "starting on February 20, 2020"; and the attackers removed it from SolarWinds' environment in June 2020. CISA's advisory AA20-352A recorded that the malicious library "was then signed by the legitimate SolarWinds code signing certificate". The flow on the Ledger is short (Figure 8.1): a build system, a key, an update channel, and the customer's own trusted monitoring server as the foothold. From there, CISA recorded, the intruders took "the SAML signing certificate" of victims' identity systems and forged sign-in tokens: Chapter Six's token problem, reached through a supply chain.

FIGURE 8.1

SolarWinds SUNBURST: the backdoor entered at build time and left with a valid signature



Numbered steps in order. Red: the path the attack or failure took.

The source repository was clean; the build system added the code, the vendor's key signed it, and the vendor's update channel delivered it.

SolarWinds Form 8-K (14 Dec 2020) and blog (11 Jan 2021); CISA AA20-352A (Dec 2020).

On 15 April 2021 the US Treasury and the UK's National Cyber Security Centre named Russia's Foreign Intelligence Service (SVR) as responsible. The US Securities and Exchange Commission sued SolarWinds and its chief information security officer in 2023; most claims were dismissed in July 2024, and on 20 November 2025 the parties filed a joint stipulation to dismiss the rest with prejudice (SEC Litigation Release 26423). The legal record closed with no finding against the company. The technical record is the lesson: code review of the repository would have found nothing, because the code was never in the repository.

xz-utils: two years to become the builder

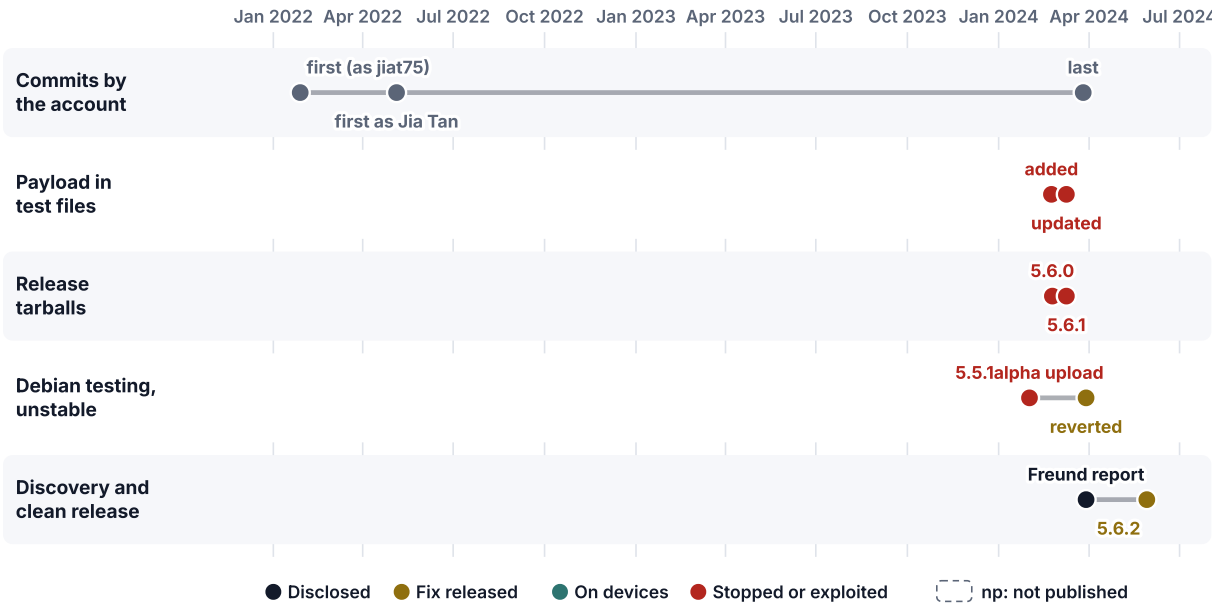
The xz case moved the attack one step earlier, from the vendor's build system to the open-source maintainer's chair. Freund's post (29 March 2024) found that the backdoor was in the 5.6.0 and 5.6.1 release tarballs, using two binary test files committed to the repository, with the code that activated it present in the tarballs but not in the git repository. Freund noted that sshd does not use liblzma directly; several distributions patch OpenSSH to notify systemd, and libsystemd depends on liblzma. Red Hat assigned CVE-2024-3094.

The repository itself records the timeline (Figure 8.2). Measured on 18 November 2026 with `git log` on github.com/tukaani-project/xz (head `f54febc540`, author dates): the account's earliest commit, authored as "jiat75", is dated 28 January 2022 and was committed by the original maintainer on 7 February; the first authored as "Jia Tan" is dated

5 May 2022 and the last 26 March 2024; of the 979 commits reachable from the v5.6.1 tag and committed since 1 June 2022, 450 carry that author name. The test files that carried the payload were added on 23 February 2024 ("Tests: Add a few test files.", `cf44e4b7`) and updated on 9 March ("Tests: Update two test files.", `6e636819`). The release commits for 5.6.0 and 5.6.1 are dated 24 February and 9 March 2024. The project's own page says the 5.6.x tarballs "were created and signed by Jia Tan". The backdoor was also gated by a key: a public analysis of the code found that it acted only on a payload "signed with the attacker's ED448 key". Here a signature was the attacker's trigger, not the defender's check. This book records the account's actions as the repository and the project describe them; it does not attribute them to any person or state.

FIGURE 8.2

xz-utils: about two years of commits, five weeks of exposure, one engineer's measurement



The backdoor reached Debian's and Fedora's development branches through ordinary release tarballs and was found by one engineer chasing a half-second delay, not by a control built to catch it; Debian counts its packages as affected from a pre-release upload of 1 February.

Freund, oss-security (29 Mar 2024); Debian security announcement (29 Mar 2024); Red Hat (29 Mar 2024); tukaani.org/xz-backdoor; git log of github.com/tukaani-project/xz at f54febc540 (measured 18 Nov 2026).

Exposure was narrow because distribution practice worked. Freund wrote that the versions "have not yet widely been integrated by linux distributions", mostly in pre-release versions. Debian's announcement of 29 March said "no Debian stable versions are known to be affected"; the compromised packages were in testing, unstable and experimental, from an

upload dated 1 February 2024, and Debian reverted to upstream 5.4.5 code, versioned `5.6.1+really5.4.5-1`. Red Hat said Fedora Rawhide users might have received 5.6.0 or 5.6.1 and Fedora 40's beta contained affected libraries, and that "No versions of Red Hat Enterprise Linux (RHEL) are affected". The project published clean releases on 29 May 2024.

The clock is the uncomfortable part. Measured from the account's first commit in the repository, the attacker's clock ran at least 25 months before the backdoor shipped. The defenders' clock, from Freund's post to Debian's revert, ran the same day. What stood between the two was not a control. It was one engineer's benchmark and the fact that stable distributions move slowly. Slow release trains, often treated as a cost, were in this case one of the protections.

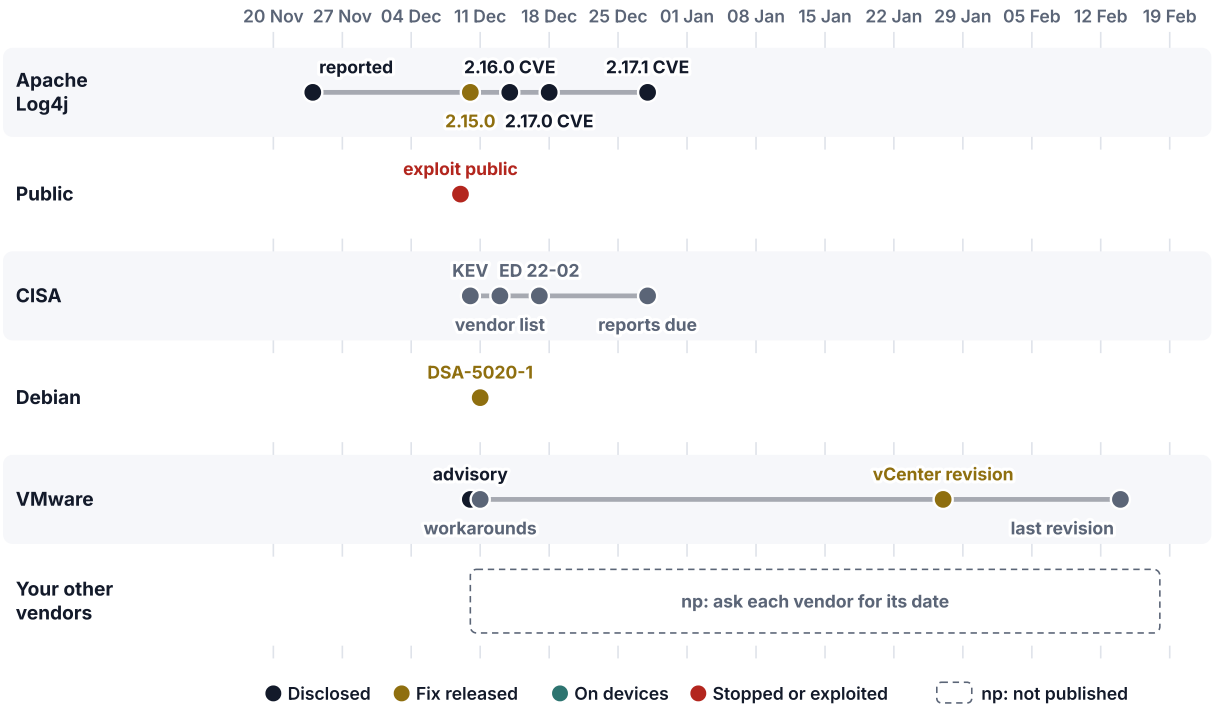
Log4Shell: the library inside everything

Log4j is a logging library for Java, maintained by the Apache Software Foundation and embedded in thousands of products. The US Cyber Safety Review Board's report (11 July 2022) gives the dates. On 24 November 2021 a security engineer from the Alibaba Cloud Security team reported a flaw: a string written to a log could make the library look up a remote address through JNDI, the Java naming interface, and load code from it. Apache committed a fix on 5 December, tagged a release candidate on 6 December, and, after the flaw was discussed publicly on 9 December, made version 2.15.0 public on 10 December. CVE-2021-44228 scored 10.0, the highest CVSS score. Further fixes followed, with CVEs published on 14, 18 and 28 December (fixed in 2.16.0, 2.17.0 and 2.17.1, each released on or just before its CVE date). CISA added the flaw to its Known Exploited Vulnerabilities catalog on 10 December, published a community list of affected products on 13 December, and issued Emergency Directive 22-02 on 17 December, requiring federal agencies to report affected applications by 28 December.

Apache's lane in Figure 8.3 is 34 days from report to the last December fix. That was not the clock that decided a company's exposure. The question for every CTO that week was "which of my vendors bundle it, and when will each of them ship?" Debian shipped its package update on 11 December. VMware's advisory, VMSA-2021-0028, was first published on 10 December and revised 13 more times, the last on 14 February 2022, as fixes and workarounds reached its many products. For most commercial products no fix date was ever published in a form a customer could compare. The Board found that one federal cabinet department reported spending 33,000 hours on its Log4j response, and called Log4j an "endemic vulnerability" whose vulnerable instances "will remain in systems for many years to come, perhaps a decade or longer".

FIGURE 8.3

Log4Shell: Apache's lane ends in December; your vendors' lanes run on their own clocks



Apache's dates are short and public; the lane that decides your exposure is each vendor's, and many of those dates were never published.

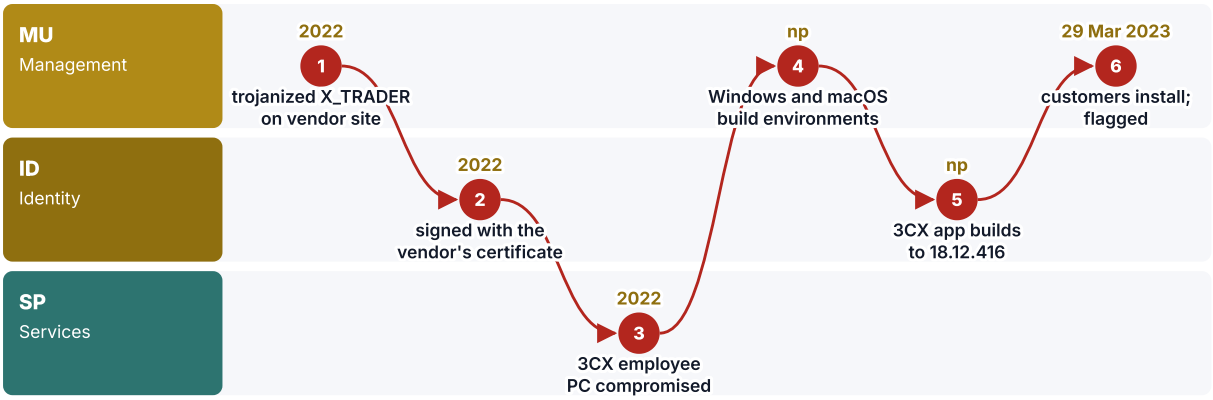
CSRB, Review of the December 2021 Log4j Event (11 July 2022); CISA ED 22-02; Debian DSA-5020-1; VMSA-2021-0028 revision history.

3CX: one signed supply chain inside another

On 29 March 2023 CrowdStrike reported malicious activity from the legitimate, signed 3CX desktop app, a business phone client on Windows and macOS. Mandiant's investigation (20 April 2023) found that 3CX's network had first been compromised through a trojanized installer of X_TRADER, trading software from Trading Technologies that had reportedly been discontinued in 2020 but was still downloadable from Trading Technologies' website in 2022. The installer was signed with a Trading Technologies certificate set to expire in October 2022. From there the attacker reached 3CX's Windows and macOS build environments, and 3CX's own builds, up to 18.12.416, carried a downloader to customers (Figure 8.4). Mandiant called it the first time it had seen one software supply chain attack lead to another, and tracks the activity as a cluster it calls UNC4736. Trading Technologies said it could not verify Mandiant's assertions.

FIGURE 8.4

3CX, 2022 to 2023: one signed supply chain inside another



Numbered steps in order. Red: the path the attack or failure took.

A signed installer from one software vendor became the way into a second vendor's build systems, whose signed app then reached its own customers.

Mandiant, 3CX Software Supply Chain Compromise Initiated by a Prior Software Supply Chain Compromise (20 Apr 2023); CrowdStrike blog (29 Mar 2023).

Chapter Thirteen tells NotPetya (June 2017), which Cisco Talos traced to the update system of a Ukrainian accounting package; it belongs to the same family: the update channel of trusted software as the delivery route.

The registry and Larkfield's own pipeline

Larkfield Software ships an Electron-style desktop app to about 400,000 desktops. Its build pulls hundreds of open-source packages from the npm registry, runs on hosted CI runners, signs the result with Larkfield's code-signing key and pushes it through Larkfield's own update channel. Every one of those steps is an authority over its customers' desktops.

On 23 September 2025 CISA warned of a self-replicating worm in the npm ecosystem, named "Shai-Hulud", that had compromised over 500 packages. After gaining access it scanned for credentials, including GitHub personal access tokens and cloud API keys, published them, then authenticated to npm as the compromised developer, injected code into that developer's other packages and published new versions. CISA's advice reads like a supply-chain checklist: pin npm dependencies to known safe releases produced before 16 September 2025, rotate all developer credentials, require phishing-resistant MFA on developer accounts, and audit GitHub apps, webhooks and secrets. A second wave, found on 24 November 2025, backdoored at least 796 packages through install-time scripts, by Datadog's count. The build pipeline is a target too: in March 2025 the widely used GitHub Action tj-actions/changed-files was compromised (CVE-2025-30066), and runs between 12

and 15 March, CISA said, could expose secrets such as access keys, GitHub tokens and npm tokens in build logs. GitHub's response for npm, announced on 22 September 2025, the day before CISA's alert, was to deprecate long-lived classic tokens, limit token lifetimes and push publishers to trusted publishing from CI and to FIDO-based 2FA.

For Larkfield that is a measurable design choice. A lockfile that pins exact versions, a build that installs from that lockfile only, a delay before new upstream versions are accepted, and build runners that hold no long-lived publishing tokens turn a registry incident into a review task instead of a release to 400,000 machines. Provenance attestations on Larkfield's own releases let its customers check that what they received was built by Larkfield's pipeline from a named commit. None of that proves the commit was benign, which is why the code-signing key stays in a hardware security module with a human approval per release (Chapter Six).

Decision point	Windows	macOS	iOS	Linux	Android
Who must vouch for an app before it runs	By default, without an application-control policy, desktop apps need no signature to run; Authenticode signature and SmartScreen reputation shape warnings; drivers need Microsoft's signature (Chapter Five)	Developer ID signature and Apple notarization for apps outside the App Store, checked by Gatekeeper	Apple signs App Store apps; alternative distribution in the EU still passes Apple notarization	The distribution signs its repository; each added repository brings its own key	The developer signs the APK; with Play App Signing, Google holds the app signing key for Play delivery
Who can withdraw trust quickly	Microsoft (Defender, SmartScreen, certificate revocation); the vendor's own channel	Apple (certificate and notarization revocation); the vendor's channel	Apple	The distribution (pull or revert a package, as Debian did with xz); the vendor's repository	Google, for apps distributed through Play; the developer, through its own update channel
Where bundled libraries hide	Inside each installer and app folder	Inside each app bundle	Inside each app, statically linked	Shared system packages, plus whatever containers, Snaps, Flatpaks and vendor tarballs carry	Inside each APK

The table's last row is the Log4Shell lesson in platform form. Where a distribution ships one shared copy of a library, one package update fixes every program that uses it, once each program restarts; containers, Snaps, Flatpaks and vendored copies carry their own. Where each app carries its own copy, as on every row except the Linux system packages, each vendor must ship separately, and your fleet is only as patched as its slowest vendor.

HOW IT GOT THIS WAY

In his Turing Award lecture, "Reflections on Trusting Trust" (Communications of the ACM, vol. 27, no. 8, August 1984), Ken Thompson described a compiler modified to insert a backdoor into the login program, and to reinsert itself whenever the compiler was rebuilt from clean source. His conclusion: "You can't trust code that you did not totally create yourself." Four years later, on 2 November 1988, the Morris worm spread from a computer at MIT and, by the FBI's account, hit an estimated 6,000 of the roughly 60,000 computers then on the internet within 24 hours: the first internet-scale reminder that shared code meant shared flaws. Shared code then made software cheap, from Unix libraries to the npm registry, and the trust placed in each builder and maintainer was never priced. SolarWinds was Thompson's compiler with a build server in its place; xz was Thompson's argument with a maintainer account in its place.

What hasn't changed: you still trust whoever built the binary, and a feature added for convenience stays in the code for years. Measured on 18 November 2026 in github.com/apache/logging-log4j2 (2.x branch, head `572cd83040`): the JNDI lookup class `log4j-core/.../lookup/JndiLookup.java` was added on 18 July 2013 (commit `f1a0cac60f`, "LOG4J2-313 - Add JNDILookup"), was present at the 2.0 release tag (12 July 2014), was still present at 2.14.1 (6 March 2021), the last release before Log4Shell, and is still in the source tree today. The December 2021 fixes disabled JNDI by default (commit `c362aff473`, "LOG4J2-3208 - Disable JNDI by default", 11 December 2021, in 2.16.0) and removed message lookups; they did not remove the file. Eight years passed between the convenience and the CVE.

THE UNSEEN ANGLE

Two of this chapter's cases put the malicious code where source review cannot see it. SolarWinds told the SEC the backdoor "was not present in the source code repository of the Orion products"; Freund found the xz activation code in the release tarballs, not in git. So "we review the source" and "it's open source, anyone can read it" are both answers to the wrong question. The authority that matters is the build: who controls the machine that turned source into the file you installed, and can you check that the file matches the source? Your vendor's build system is part of your Trust Ledger whether or not the vendor lets you see it. Ask for it: reproducible or attested builds, the build platform, who can change it, and how the signing key is approved per release. Sources: SolarWinds Form 8-K (14 December 2020); Freund, oss-security (29 March 2024).

> *A signature tells you who shipped it. Nothing in the signature tells you whether you should run it.*

These rows in the Trust Ledger

Larkfield Software, desktop app on about 400,000 customer desktops (Windows, macOS, Linux). Clock and count cells are placeholders until measured from Larkfield's build and update telemetry (Appendix D). Row codes and tags are as defined in Chapter One.

Component	Who can change it	Exposure and blast radius	Patch clock	Stop, revoke, recover	Who can see it	Support end	Decision and owner
LR: npm dependency tree of the app (direct and transitive; count nc)	Each package's maintainers and anyone holding their tokens; the registry	Build machines at install time; every customer desktop after release	Upstream fix, then Larkfield release, then customer update; last cases nc	Lockfile pin; revert to previous lock; no drill yet	D: SBOM per release; O: dependency-scan alerts	Per package; many have none	**Own** the gate: exact pins, a hold period for new versions, review of install scripts; engineering lead
LR: bundled Chromium and Node (Electron)	Electron project; Larkfield ships	All desktops; renderer reachable from content	Electron release, then Larkfield release (Book 2, Chapter 11)	Staged rollout; rollback to previous build	O: version telemetry	About 24 weeks per Electron major	**Own**, **Pin** to a supported major with an end date; engineering lead
MU: CI/CD build pipeline (hosted runners)	Larkfield engineers with pipeline write access; the CI vendor	Every release; one change reaches all customers	CI vendor's advisories; Larkfield's own fixes immediate	Rotate pipeline secrets; rebuild from a known-good commit; drill np	D: pipeline audit log; provenance per build	CI vendor contract	**Rent** the runners, **Own** the provenance: signed build attestations, no long-lived publish tokens on runners; head of platform
ID: code-signing keys (Windows Authenticode certificate; Apple Developer ID)	Holders of the HSM approval role (Chapter Six)	Any binary signed with it is accepted as Larkfield's on 400,000 desktops	New key at each certificate renewal, on the CA's and Apple's schedules	Revoke and re-sign with a new certificate; customer impact untested	D: HSM signing log	Certificate expiry date	**Own**: two-person approval per release signature; head of release, CISO approves
MU: Larkfield update channel	Larkfield release managers	All 400,000 desktops	Larkfield's own clock: release to 90% of active installs nc	Server-side halt of a rollout; previous build available	O: update telemetry	n/a	**Own**: staged rings, a halt switch tested each quarter; release manager

SECURITY LENS

Who can change it? Everyone with write access to a dependency you install, everyone who can change your build system or your vendors' build systems, and everyone who can approve a signature with your key. For Larkfield's customers, that list includes Larkfield's CI vendor and hundreds of npm maintainers they have never heard of.

How fast can it be fixed? The library's clock is the short one: Apache's first Log4Shell fix was public 16 days after the report, and Debian reverted xz the day it was disclosed. The long clock is the bundling vendors': VMware's advisory was still being revised two months later, and most vendors published no comparable date at all. Measure your vendors' clocks the way you measure your own.

What does AI change here? Our view (as of November 2026): AI coding assistants make it cheaper to add a dependency and cheaper to write a plausible patch, so the volume of change arriving at maintainers and at your lockfile goes up while the number of reviewers does not. Treat an AI-suggested dependency like any other new authority: pin it, review its install scripts, and record why it is there. Agents that run in CI with publish tokens are authorities in the AG row and need the same limits as the runners.

BUSINESS IMPLICATIONS

Security. Larkfield is the supply chain for 400,000 desktops; its signing key is its customers' risk. Halbrook should ask every desktop vendor the Log4Shell question before the next incident: which open-source components do you bundle, and what is your time from upstream fix to your release? Corvane's kiosk integrator should answer the same question in writing.

Cost. Recall the federal department that spent 33,000 hours on Log4j alone. An SBOM per product and an inventory that can answer "where is library X?" in an hour cost a small fraction of those hours.

Visibility. You see a package arrive; you do not see who built it. Ask for build provenance, store SBOMs from every vendor, and ask for VEX documents (Vulnerability Exploitability eXchange, minimum elements set out by CISA in April 2023), in which a vendor states whether a listed flaw affects its product. Keep a list of vendors who will provide none of these; that list is a finding.

Value. Shared code is why Larkfield can ship one app on three operating systems with a small team. The answer is not to stop using it but to price the builders you trust: maintainers you depend on, a hold period on new versions, and funding or staff time for the few components you cannot do without.

Monday rule: for your most widely deployed product, name the machine that builds the release, everyone who can change it, and everyone who can approve its signature; if you cannot name them, that is the first row of your Ledger.

OUR VIEW, AS OF NOVEMBER 2026

Signed provenance for releases will become a routine procurement ask, the way SBOMs did, and it will help most with SolarWinds-style build tampering. It will not stop the next xz, because that attack was made by a trusted author through a legitimate build. The defences that worked in 2024 were slow stable releases, an engineer who measured something odd, and distributions able to revert in hours. Keep all three, and fund the maintainers whose work your products cannot ship without.

CHAPTER NINE

When Trust Must Be Withdrawn: Stop, Revoke, Roll Back, Recover



On 6 April 2022 NIST published the fourth revision of its patching guide, SP 800-40, and gave it a subtitle that changes the conversation: *Preventive Maintenance for Technology*. NIST "frames patching as a critical component of preventive maintenance". Maintenance has a second half that most patch programs never write down: what you do when the thing you just installed, or the key you have trusted for years, turns out to be the problem. Every earlier chapter named an authority that can change your systems. This one asks the reverse question: when trusted code, a key or a channel must no longer be trusted, how do you stop it, how do you withdraw it, and how fast do you get back to work? The decision it changes is simple to state. Next to every patch clock in the Trust Ledger, a team writes a **recovery clock**, measured in a drill, and treats a channel it cannot stop as a liability.

Deployment is a curve, not a date

"We patched in March" is a date. What protects you is a curve: how many of your devices run the fix 24 hours, 7 days and 30 days after the vendor released it, and who is left over. Chapter One defined the patch clock: T0 when the flaw is known, T1 when a fix exists for your device class, T2 when 50% of your fleet runs it, T3 at 90%. The curve between T1 and T3 is where your exposure lives, and its shape is set by decisions you own.

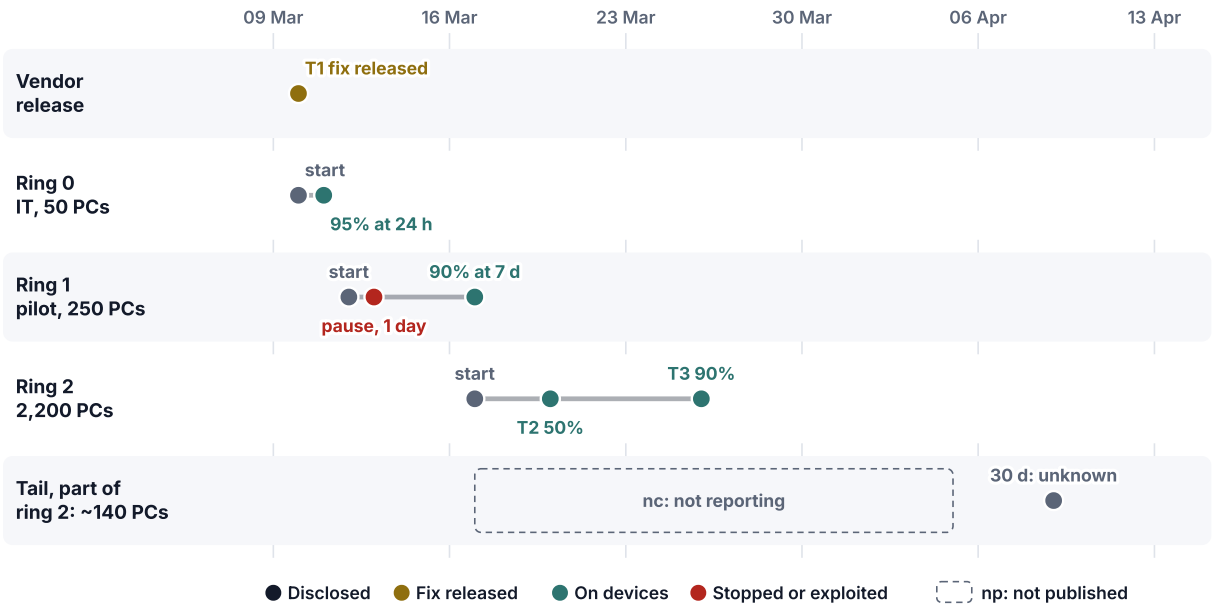
Five numbers describe the curve, and a fleet report should carry all five:

- **Eligible population:** the devices that should receive this fix. Devices on an unsupported release are not eligible; they belong in Chapter Eleven, not in the denominator, but their count goes beside the percentage.
- **Installed at 24 hours, 7 days and 30 days**, as a share of the eligible population.

- **Waiting for a restart:** installed but not running. Chapter Five and Book 2 (Chapter 11) showed that a fix staged on disk does not protect code already running until that process or the machine restarts, unless it is hotpatched.
- **Failed installs**, with the error, so a pattern shows up as a pattern.
- **Not reporting:** devices that stopped sending telemetry. They are not patched or unpatched; they are unknown, and the unknown share is the tail.

FIGURE 9.1

Deployment is a curve, not a date: who runs the fix at 24 hours, 7 days and 30 days, and who is still in the tail



An illustrative month for Halbrook's 2,500 branch PCs: three rings, one tested pause, and a tail that reports nothing, which is where the exposure lives.

Author's diagram. Halbrook figures are invented and illustrative; the method (T0 to T3, tail) is in Chapter 1 and Appendix D; maintenance groups follow NIST SP 800-40 Rev. 4 (April 2022).

Figure 9.1 is the curve for one illustrative month at Halbrook Bank. It uses **rings**: groups of devices that receive a change in order, so a fault shows up on 50 IT machines before it reaches 2,500 branch PCs. NIST calls the same idea maintenance groups. The tail of about 140 PCs is the part of ring 2 that stopped reporting. Rings are not the opposite of speed. In the figure the first ring has the fix within 24 hours, and the last ring reaches 90% on day 16. A flat deployment to every device on day one is faster only on the day it works.

The vendors' own controls set the edges of the curve. Windows Update client policies (formerly Windows Update for Business) let an administrator defer quality updates "for a period of up to 30 days" and pause them "for a period of up to 35 days". Hotpatch, generally available for Windows 11 Enterprise 24H2 on Intel and AMD devices from 2 April 2025, removes most restarts: Microsoft plans "eight hotpatch months, and four baseline months" a year, with a restart only in the baseline month of each quarter. Apple's device management lets an organization set "a custom deferral period from 1 to 90 days" on supervised devices. Android splits the platform: since Android 10, Mainline modules can reach devices "from the Android Google Play system update feature", outside the device maker's full image, while the rest of the image still travels on each maker's clock. On Linux, kernel livepatching works by "allowing for function calls to be redirected; thus, fixing critical functions without a system reboot", where a distribution offers it.

Each of these is a lever on the same curve. Deferral moves T1 later on purpose, hotpatch and livepatch shrink the restart gap, and Mainline gives part of Android a faster channel. None of them shrinks the tail; that takes inventory, and compensating controls for what inventory finds.

Stop: every channel needs a brake you have tested

Chapter Seven counted the channels that can push code into one endpoint: OS update, OEM tools, MDM, RMM, the EDR console and its content channel, app updaters, package managers. For each of them the Ledger's fifth column asks one plain question: if a push goes wrong at 04:00, who can stop it, with what control, and how long does that take?

The vendors' answers differ, and the differences are worth writing down before the morning you need them.

- **Pause before it spreads.** Windows quality updates can be paused for up to 35 days by policy. A Play Console staged rollout can be halted, and Google says that then "no additional users will receive the app version"; devices that already have it keep it. Larkfield's own updater needs the same switch, and it needs it on the server side, because a client that has already crashed cannot ask whether it should stop.
- **Withdraw what shipped.** Apple's Background Security Improvements, supported from iOS, iPadOS and macOS 26.1 and first shipped on 17 March 2026 (a WebKit fix, CVE-2026-20643), can be removed, but only all at once: users "can also elect to remove all Background Security Improvements currently applied on their device", which returns the device to its base update. Apple can also remove one itself. Its predecessor showed why that matters: on 10 July 2023 Apple shipped the Rapid Security Response iOS 16.5.1 (a), a WebKit fix for a flaw that may have been actively exploited, and on 12 July replaced it with 16.5.1 (c) "to fix an issue that prevents some websites from displaying properly".

- **Fall back automatically.** Android's A/B system updates write the new image to a second slot. AOSP states that if an update "fails to boot, the device will reboot back into the old partition and remains usable". That is a brake that works even when nobody is awake, but only for the OS image, not for an app or a driver.
- **Content channels.** The case that made this column famous was a content update, not a code update. On 19 July 2024 CrowdStrike reverted the faulty Channel File 291 at 05:27 UTC, 78 minutes after release; the full case is in Chapter Five and the record in Chapter Thirteen. Afterwards CrowdStrike said it would "implement a staggered deployment strategy for Rapid Response Content", starting with a canary deployment, and give customers "granular selection of when and where these updates are deployed". Its root cause analysis, published in August 2024, reported that new capabilities "have been implemented and deployed to our cloud that allow customers to control how Rapid Response Content is deployed". Whether a customer has set them is a configuration question, which is why the Ledger records it per vendor.

A brake that has never been pulled is a hope. The fifth column carries the date of the last drill and the measured time from "stop" to "no further devices received it".

Revoke: withdrawing trust without stranding recovery

Revocation is the act of telling every verifier that something it used to accept is no longer acceptable: a boot manager added to the Secure Boot forbidden list, a signing key retired, a certificate marked revoked, a token invalidated. Revocation can be the strongest stop, because it reaches copies you cannot find, but only where a verifier receives the revocation and enforces it; offline devices, stale caches and clients that skip the check keep accepting the old item. That strength is also its cost: a revocation aimed at a key or a file hash does not distinguish between the attacker's copy and your own recovery media signed the same way.

Microsoft's handling of the BlackLotus bootkit is the clearest public record of that trade (the attack itself is told in Chapter Four). Its support article KB5025885, for CVE-2023-24932, staged the revocation of vulnerable Windows boot managers over years: an initial deployment on 9 May 2023, a second deployment on 11 July 2023, an evaluation phase from 9 April 2024, and an enforcement phase that Microsoft said "will not begin before January 2026" and for which it promised at least six months of warning. As of its change log of 30 September 2026, no enforcement date had been announced. The article explains the slowness in two sentences: once the mitigation is applied, "it cannot be reverted if you continue to use Secure Boot on that device", and "even reformatting of the disk will not remove the revocations". Older boot media built without the updates may not start. The staging exists so that a fleet can refresh its recovery media, images and network boot servers before the old boot manager stops being trusted.

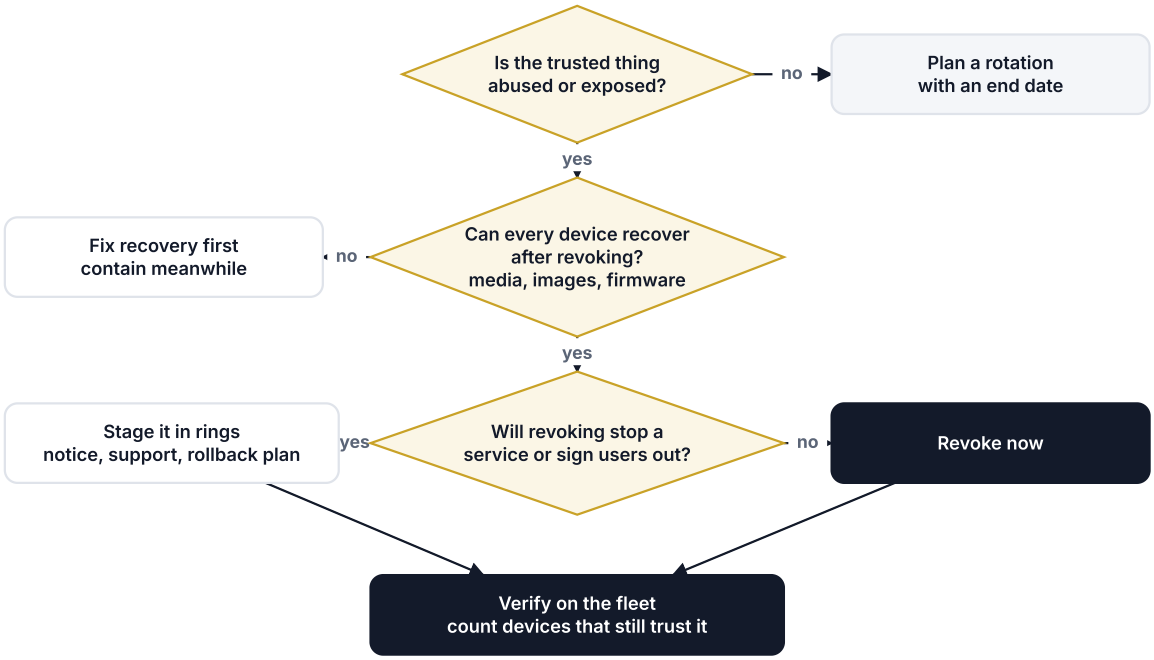
Rollback cuts both ways. At Black Hat USA in August 2024, SafeBreach researcher Alon Leviev showed Windows Downdate, which took over the Windows Update process to put older, vulnerable system files back on a patched machine that would still "appear fully up-to-date". Microsoft assigned CVE-2024-21302 and CVE-2024-38202. Its answer for the first, guidance first published on 13 August 2024 (KB5042562), is again a revocation, used as an anti-rollback control: a Microsoft-signed policy that blocks the old files, after which "the device cannot be reverted" while Secure Boot stays on, so recovery media and the recovery environment must be updated first. The same trade as BlackLotus, two years later.

Identity revocations have the same shape, with users instead of boot media on the other side. After the Storm-0558 intrusion (Chapter Six) Microsoft stated that it "revoked all MSA signing [keys] which were valid at the time of the incident, including the actor-acquired MSA key". Removing a token-signing key from the published key set makes tokens signed with it fail validation once each verifier refreshes its copy of the keys; that is the point, and it also signs out every user and service that holds one. When CircleCI disclosed a security incident on 4 January 2023, it told customers to "immediately rotate any and all secrets stored in CircleCI", and it rotated every GitHub OAuth token on its customers' behalf by 7 January. Each customer then had to find which of its own pipelines broke. The root cause, published on 12 January, was malware on an engineer's laptop that stole "a valid, 2FA-backed SSO session" (Chapter Six's session theft).

Web certificates show the limit and the industry's answer. A revoked TLS certificate stops being trusted only where clients check revocation, so the CA/Browser Forum chose shorter lives instead: ballot SC-081, approved on 11 April 2025, cuts the maximum life of a public TLS certificate from 398 days, in steps from March 2026, to 47 days by March 2029. A certificate that expires in weeks needs less revoking, and much more automated renewal. Detection runs the other way: Certificate Transparency logs, public append-only records of issued certificates, are how a company learns that a certificate it never asked for was issued in its name.

FIGURE 9.2

Revoke or keep: revocation is only safe when recovery still works afterwards



Ask the recovery question before the revocation question: a revocation that strands old media or signs everyone out must be staged like any other change.

Author's diagram based on Microsoft KB5025885 (staged Secure Boot revocations for CVE-2023-24932) and the key-compromise lifecycle in Chapter Six.

Figure 9.2 turns these cases into the order of questions. The recovery question comes before the revocation question. If revoking would leave some devices unable to recover, fix recovery first and contain the threat in the meantime, unless the active threat is worse than the outage, which is the incident commander's call; if revoking would stop a service or sign everyone out, stage it like any other change. In both cases, finish by counting the devices that still trust the old item, because a revocation that reached 90% of the fleet left 10% trusting the attacker.

Cross-signing is the gentler form of the same move: a new key or certificate is trusted alongside the old one for a period, so devices can move before the old one is withdrawn. Chapter Four shows that transition for the 2011 Secure Boot certificates. The Ledger records the period's end date, because a transition without an end date is a second root of trust that nobody planned to keep.

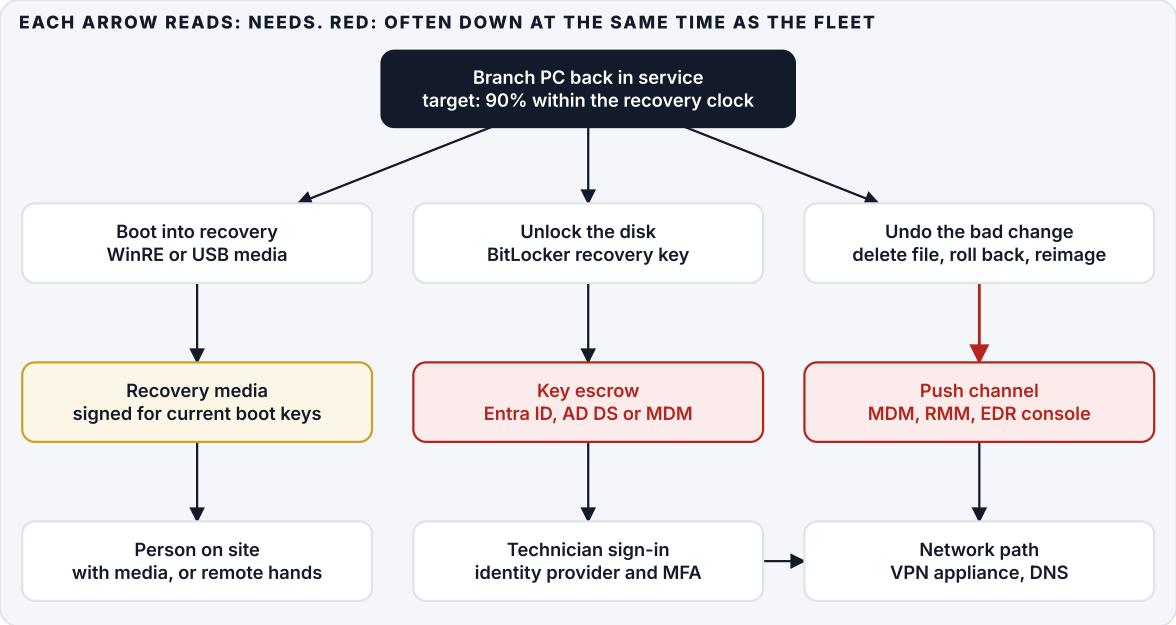
Recover: the recovery clock and what it depends on

The **recovery clock** runs from the moment a service fails to the moment 90% of it works again for the people who use it. It is not the time to the fix. The CrowdStrike file was reverted in 78 minutes; CrowdStrike reported that about 99% of Windows sensors were online again as of 29 July 2024, ten days later, and that number counts sensors connected, not businesses restored. The gap between those two numbers is recovery, and it was spent mostly on dependencies.

Microsoft's manual fix for affected machines (KB5042421) warned: "You may be asked to enter your BitLocker recovery key." That one sentence is the whole lesson of Figure 9.3. Recovering one device needed a recovery boot, the disk key, and a way to remove one file. Where escrow policy is set, the disk key is stored in Microsoft Entra ID for Entra-joined devices or in Active Directory Domain Services for domain-joined ones; a Mac's FileVault personal recovery key can be escrowed with the device management service. Reading the key needs a technician who can sign in, which needs the identity provider and MFA, which needs a network path through the VPN appliance and DNS. Removing the file needs either a push channel, which cannot reach a machine that does not boot, or a person with media at the device. Microsoft released a recovery tool (KB5042429) on 20 July 2024 that builds bootable media with two repair options; it still needed someone to plug the media in.

FIGURE 9.3

Recovery has its own supply chain: what must work before one device can be restored



Each arrow reads 'needs'; the red boxes are the dependencies that a fleet-wide boot failure can take down at the same moment as the devices.

Author's diagram based on Microsoft's BitLocker recovery and Windows Recovery Environment documentation and the July 2024 recovery guidance (Chapter 5).

Draw your own version of Figure 9.3 for each critical system, then mark in red every box that runs on, or depends on, the fleet it is meant to recover. Those are the dependencies that fail together. Four questions find most of them:

- **Does recovery need the identity provider?** If the escrow store, the technician's sign-in or the MFA app depends on the failed fleet, keep a break-glass path: a sealed list of recovery keys for the most critical sites, held offline, split between two custodians, audited, and regenerated after use.
- **Does recovery need the management plane?** If the MDM, RMM or EDR console is the only way to push a fix, recovery from a failure of that console, or a compromise of it (Chapter Seven), needs a second path.
- **Does recovery need the network?** A branch or a hotel property that loses its VPN appliance loses its route to the escrow store and the update service at the same time.
- **Does recovery need a person on site?** Then the recovery clock is a staffing number: devices per technician per hour, times sites, plus travel.

That last question is where the cost of a bad push lives. Corvane has about 3,000 kiosks across 300 properties. If one change leaves every kiosk needing a hands-on repair, and a trained person fixes about six kiosks an hour once on site, the work is roughly 500 technician-hours before anyone travels; those are planning numbers to replace with a drill, not measurements. The cheapest recovery hour is the one prepared in advance: a golden image that can be laid down on a kiosk without the network, recovery media that are already signed for the current boot keys, and a property-level procedure that lets a duty manager check guests in on paper while the kiosks come back.

The vendors' answer: recovery built into the platform

Platform vendors have started to treat recovery as a product feature. Microsoft's **Quick Machine Recovery** became generally available on Windows 11 version 24H2 with update KB5062660 in July 2025. After repeated boot failures, the device starts in the Windows Recovery Environment, connects to the network, and, in Microsoft's words, uses "a secure and connected Windows Recovery Environment (WinRE) to scan Windows Update for remediation options", applies one and restarts. Its defaults matter for the Ledger: Microsoft's current documentation says cloud remediation is on for Home and unmanaged Pro devices and "disabled by default on enterprise-managed systems". An enterprise must decide to turn it on, and it then depends on the network and on Windows Update being reachable from recovery, which puts it back on Figure 9.3.

Apple's equivalents are recoveryOS on the Mac and the removable Background Security Improvements described above; Android's are the A/B fallback and Mainline. Linux distributions offer livepatch, package downgrades and, on image-based systems, a rollback to the previous image; which of these exists depends on the distribution, so the platform profile in Chapter Two decides the row.

Our view (as of November 2026): built-in recovery shortens the clock for the common case, a bad update that the vendor can fix centrally. It does not help when the failure is in your own escrow store, identity provider or console, and those are the cases a Trust Ledger exists for.

Withdrawal step	Windows	macOS	iOS	Linux	Android
Stage and pause OS updates	Defer quality updates up to 30 days, pause up to 35; rings in Intune or Autopatch	MDM deferral 1 to 90 days on supervised devices	MDM deferral 1 to 90 days on supervised devices	Your own repository mirror and timing	Device maker's OTA plus EMM policy; Play system updates for Mainline
Fix without a full OS update	Hotpatch (Enterprise 24H2): 8 months a year without restart	Background Security Improvements (26.1 on)	Background Security Improvements (26.1 on)	Kernel livepatch where the distribution offers it	Mainline modules through Google Play system updates
Roll back a shipped change	Uninstall the update; QMR remediation from WinRE where enabled	Remove all Background Security Improvements at once	Remove all Background Security Improvements at once	Package downgrade; image rollback on image-based systems	A/B fallback to the old slot if the new one fails to boot
Disk key escrow for recovery	BitLocker key in Entra ID or AD DS	FileVault personal recovery key escrowed to MDM	Not applicable to the user (hardware-bound)	Depends on distribution and tooling	Not applicable to the user (hardware-bound)
Firmware and silicon	DBX (the Secure Boot forbidden list) and boot manager revocations staged by KB5025885; cannot be reverted with Secure Boot on	Apple-signed boot chain; recovery through recoveryOS	Apple-signed boot chain	shim and SBAT (Secure Boot Advanced Targeting) revocations via the distribution and firmware DBX	Verified Boot with rollback protection set by the device maker

The iOS and Android key rows say "not applicable to the user" because the platform holds the storage keys in hardware and recovery runs through the vendor's restore path, not a key a company escrows; Chapter Two describes each profile.

HOW IT GOT THIS WAY

Automated updating was built for speed first and control later. Windows Update shipped with Windows 98, generally available on 25 June 1998, as "a centralized Web-based resource site". On 1 October 2003 Microsoft announced it would move to a monthly security bulletin cadence, and the first monthly release followed that month: the day the industry calls Patch Tuesday. Windows XP Service Pack 2 reached manufacturers on 6 August 2004 and was delivered through Automatic Updates, which Microsoft urged every user to turn on. Each step moved more of the fleet onto the vendor's clock. The controls to slow, pause and withdraw a change, rings, deferrals and staged rollouts, came afterwards, as products and policies layered on top of a channel designed to push.

What hasn't changed: the monthly rhythm set in October 2003 still drives the Windows fleet. Microsoft's hotpatch documentation for Windows 11 in 2025 still plans the year as twelve monthly releases, "eight hotpatch months, and four baseline months", twenty-two years after the first Update Tuesday (Microsoft Learn, Windows Autopatch hotpatch page, checked 18 November 2026). What has changed is only what happens between the monthly release and a running fix.

THE UNSEEN ANGLE

Secure but unrecoverable is a real state. KB5025885 says in plain words that a revocation cannot be reverted while Secure Boot stays on, and that older boot media may not start afterwards. A fleet that applies it before refreshing its recovery media has made itself more secure and less recoverable on the same day. The pattern is general: the recovery path usually depends on something the incident has just broken, whether that is the disk key in an escrow store on the failed servers, the console that cannot reach a machine that does not boot, or the network path through an appliance at the property. No vendor dashboard shows that dependency, because it crosses vendors. Only your own drawing of Figure 9.3 does.

> *A control you cannot stop is a liability, and a revocation you cannot recover from is an outage you scheduled yourself.*

These rows in the Trust Ledger

Halbrook Bank, branch desktop (2,500 Windows PCs, BitLocker on, Entra-joined). Only the columns this chapter changes are filled in detail; the times are drill targets until Halbrook measures them. Row codes and tags are as defined in Chapter One.

Component	Who can change it	Exposure and blast radius	Patch clock	Stop, revoke, recover	Who can see it	Support end	Decision and owner
MU Windows quality updates, three rings	Microsoft pushes; Halbrook configures rings and deferrals	All 2,500 PCs; ring 0 is 50 IT PCs	T3 target day 16 (as illustrated in Figure 9.1); tail nc	Pause up to 35 days, tested monthly; uninstall path documented; recovery 90% target 8 hours	O (Intune reports), plus a not-reporting list	Per OS release (Chapter Eleven)	**Rent**; **Own** the rings and the pause; end-user computing lead; review quarterly
DR EDR sensor and its content channel	EDR vendor writes and pushes content; Halbrook sets sensor policy	Kernel; one push reaches every PC	Per vendor; content several times a day	Content staging per ring where the contract allows; last drill date; BitLocker keys reachable offline for 25 critical branches	D (vendor console), C if contracted	Per contract	**Rent**; **Isolate**: content staged by ring; CISO; review at renewal
ID BitLocker key escrow in Entra ID	Microsoft runs it; Halbrook admins read keys	Every encrypted PC depends on it for recovery	na	Break-glass export of keys for critical sites, sealed and audited; read test in each drill	O (audit logs of key reads)	na	**Rent**; **Own** the break-glass copy; identity lead; review semi-annually
FW Boot manager revocations (KB5025885)	Microsoft signs; Halbrook decides when to apply	All PCs, plus recovery media and network boot	Evaluation phase since April 2024; enforcement not announced	Refresh media and images first; apply by ring; not reversible with Secure Boot on	D (event logs), O once collected	2011 certificates: Chapter Four	**Own** the order of steps; end-user computing lead; review at each Microsoft phase change

SECURITY LENS

Who can change it? An authority that can push can usually pause its own future pushes; withdrawing what already shipped is a separate power that many lack. Whether you can stop or withdraw depends on controls the vendor exposes and you have configured: deferrals, pauses, staged content, removable security updates. Revocation is held by whoever holds the root: Microsoft for the Windows boot chain, your identity team for token-signing keys, your vendor for its own pipeline.

How fast can it be fixed? Two clocks, not one. The patch clock in this chapter's figure reaches 90% on day 16 by design. The recovery clock depends on the slowest dependency in Figure 9.3: the CrowdStrike file was reverted in 78 minutes, and the vendor's sensor-connectivity figure reached about 99% ten days later.

What does AI change here? Little in the physical part of recovery. An agent allowed to pause channels or rotate keys is itself an authority and needs a row (Chapter Sixteen).

BUSINESS IMPLICATIONS

Security. Halbrook's exposure is the tail in Figure 9.1, not the median. Corvane's is a revocation applied before its 300 properties have fresh recovery media. Larkfield's is an updater with no server-side brake: a bad release keeps spreading to 400,000 desktops until someone ships the next one.

Cost. Recovery labor and field visits dominate the bill of a bad push. For Corvane, every hour of hands-on repair per property is multiplied by 300; a golden image and pre-signed media are cheap by comparison. For Halbrook, the cost is branch hours with tellers but no working desktop.

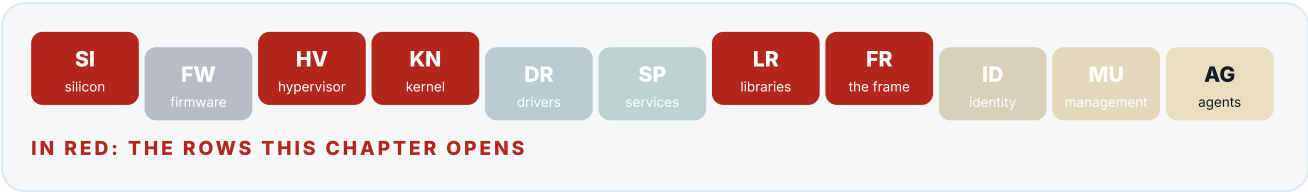
Visibility. A fleet report can say 90% while the most exposed 10% is untouched: devices that stopped reporting, sit in a closet, or never restart. Require the not-reporting count in every patch report, and require the recovery clock in every drill report.

Value. Customers notice recovery time, not patch time. For a hotel at check-in and a bank at opening hour, time to recover 90% may matter more than time to patch 90%, and it is the number a board can understand.

Monday rule: for each push channel, name the person who can stop it, pull that brake in a drill this quarter, and write down the time to stop and the time to recover 90%.

CHAPTER TEN

Zero-Click: From the Screen to the Silicon



On 1 June 2023 Kaspersky published a report about its own phones. While "monitoring the network traffic of our own corporate Wi-Fi network dedicated for mobile devices", its researchers had found iPhones talking to servers they should not know. The infection arrived as an iMessage attachment, and "without any user interaction, the message triggers a vulnerability that leads to code execution"; the message and the exploit were then deleted. The oldest traces went back to 2019. Kaspersky called it Operation Triangulation. Six months later, on 27 December 2023, it published the whole chain: four zero-days, a font instruction at the front and undocumented registers of the chip in the middle. The decision this chapter changes is how you protect the few people in your company who are worth a chain like that, and which platform features you require on their devices.

Where Book 2 stopped

A **zero-click** attack needs no action from the victim: the device parses a message or attachment on its own, and the parser is the target. Book 2 (Chapter 14) told the parser side of three cases: FORCEDENTRY, where Citizen Lab found in September 2021 that an image decoder in iMessage was the first contact; BLASTPASS in 2023; and the libwebp flaw that needed a separate fix from every product that bundled the library. It also described Apple's two platform answers, the BlastDoor parsing service in iOS 14 and Lockdown Mode in iOS 16. This chapter does not repeat the decoders. It starts at the moment the first parser has been beaten and asks what is left: which boundaries still stand between that first foothold and full control of the device, who owns each one, and which of them the attackers had to break.

The short answer from the public record is that the parser is the front door, not the house. Every documented chain below needed at least one more privilege or isolation bypass after the first foothold, and each bypass crossed a boundary someone else owns. That is why the

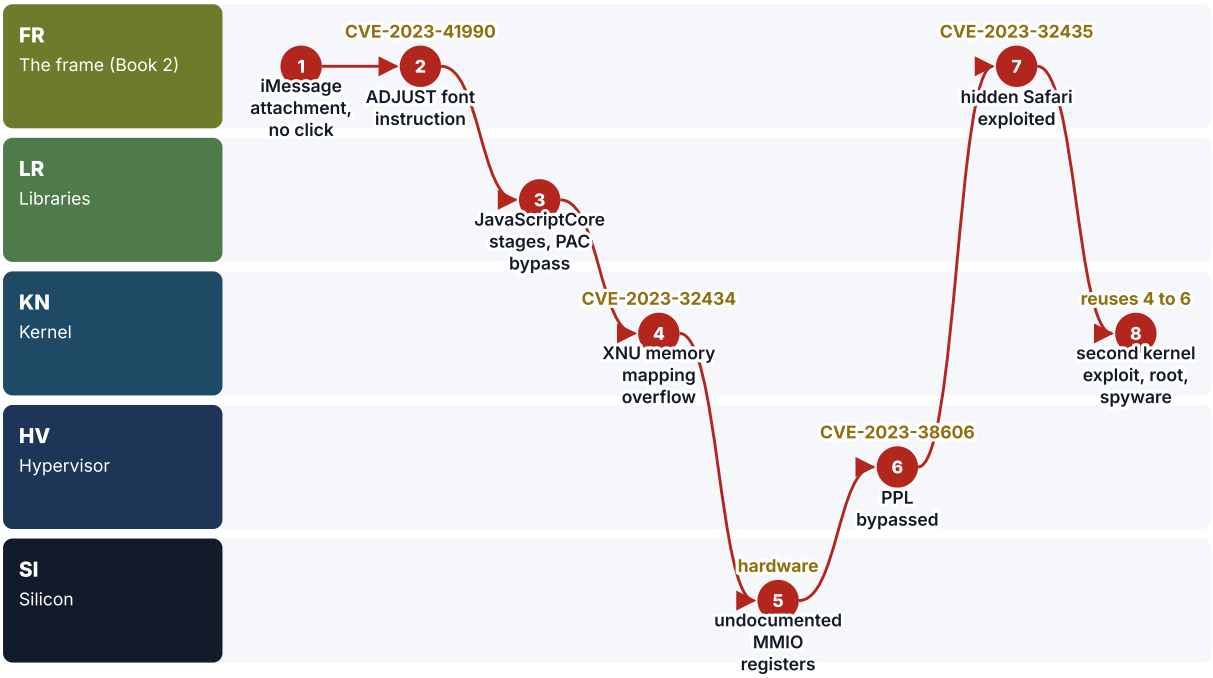
Trust Ledger treats a zero-click chain as a walk across rows, not as one bug.

Operation Triangulation, stage by stage

Kaspersky's researchers Boris Larin, Leonid Bezvershenko and Georgy Kucherin presented the chain at the 37th Chaos Communication Congress (37C3) on 27 December 2023, in a talk titled "Operation Triangulation: What You Get When Attack iPhones of Researchers", and Larin published the written version the same day. The chain "used four zero-days and was designed to work on iOS versions up to iOS 16.2". Figure 10.1 draws it on the Ledger rows; the text below stays at the level of Kaspersky's own description.

FIGURE 10.1

Operation Triangulation: one message crossed five Ledger rows, from a font instruction to undocumented chip registers and back to the kernel



Numbered steps in Kaspersky's order. Red: the path the attack took. Mechanism level only.

The chain as Kaspersky published it on 27 December 2023: four zero-days, no click, and two trips through the kernel, with each step owned by a different part of the platform. PPL sits on the isolation (HV) row because it is a protection layer beside the kernel, not a hypervisor.

Source: Kaspersky Securelist, "Operation Triangulation: The last (hardware) mystery" (B. Larin, 27 December 2023) and 37C3 talk; Apple security notes for iOS 16.3, 16.4, 16.5.1, 16.6 and 15.7.7, 15.7.8.

- **The frame (FR).** The attachment exploited "the remote code execution vulnerability CVE-2023-41990 in the undocumented, Apple-only ADJUST TrueType font instruction". In Kaspersky's words, "this instruction had existed since the early nineties before a patch removed it."
- **Libraries and runtimes (LR).** Return and jump oriented programming, then "multiple stages written in the NSExpression/NSPredicate query language, patching the JavaScriptCore library environment" so that a large JavaScript exploit could run. It "included a Pointer Authentication Code (PAC) bypass for exploitation of recent models"; PAC has the CPU sign and check pointers such as return addresses, so a forged pointer faults.
- **Kernel (KN).** The exploit used "the integer overflow vulnerability CVE-2023-32434 in XNU's memory mapping syscalls" "to obtain read/write access to the entire physical memory of the device at user level".
- **Silicon (SI) and isolation (HV).** Kernel read and write is not the end on a recent iPhone. Apple's **Page Protection Layer** (PPL) sits beside the kernel and, in Apple's words, "is designed to prevent user space code from being modified after code signature verification is complete", even by a compromised kernel. The exploit "uses hardware memory-mapped I/O (MMIO) registers to bypass the Page Protection Layer (PPL)". Apple mitigated this as CVE-2023-38606.
- **Back through the frame and the kernel.** With all of that, the exploit cleaned up, opened Safari invisibly and sent it to a web page. A script checked the victim; then "the Safari exploit uses CVE-2023-32435 to execute a shellcode", and the shellcode ran "another kernel exploit", which "uses the same vulnerabilities: CVE-2023-32434 and CVE-2023-38606". It obtained root and loaded the spyware.

The silicon step is the one that changes the Ledger. Kaspersky found that the registers were "unknown hardware registers of the chip unused by the firmware", that writing to them required a hash computed with "a custom algorithm" and "a predefined sbox table", and that the attackers had it. Kaspersky offered a guess, labeled as one: "Our guess is that this unknown hardware feature was most likely intended to be used for debugging or testing purposes by Apple engineers or the factory, or that it was included by mistake." Apple's iOS 16.6 note describes the fix only as "An app may be able to modify sensitive kernel state."

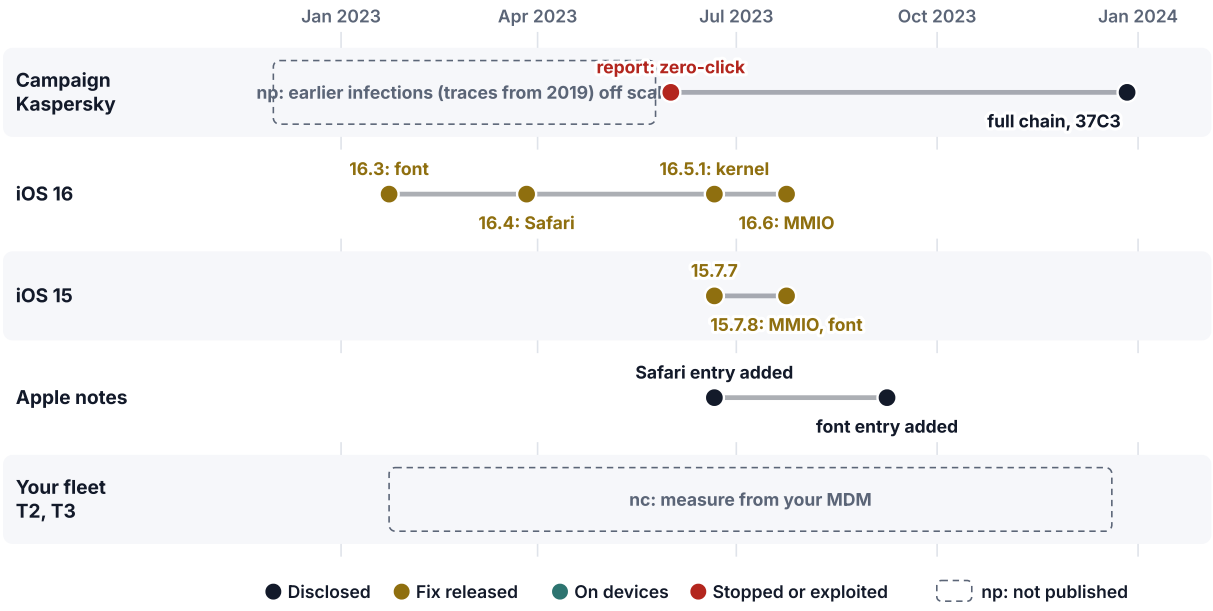
Kaspersky did not name an actor in its reports, and this book does not either.

The clocks: a silent fix and four advisories

The patch timeline in Figure 10.2 is the most instructive part of the case for a CTO, because it shows what the customer could and could not see.

FIGURE 10.2

Triangulation's clocks: the font flaw was fixed in January 2023, and Apple's notes said so in September



Apple's iOS 16.3 fixed the font flaw in January 2023, credited to Apple, but its notes said so only from 8 September 2023; the hardware bypass was fixed in iOS 16.6 on 24 July 2023.

Source: Apple security notes HT213606 (iOS 16.3), HT213676 (iOS 16.4), HT213814 (iOS 16.5.1), HT213811 (iOS 15.7.7), HT213841 (iOS 16.6), HT213842 (iOS 15.7.8); Kaspersky Securelist (1 June and 27 December 2023).

- **iOS 16.3, 23 January 2023.** Apple fixed CVE-2023-41990 in FontParser, "Processing a font file may lead to arbitrary code execution", credited to Apple. The entry was added to the iOS 16.3 notes on 8 September 2023. On the day it shipped, a customer reading the notes could not have known it existed.
- **iOS 16.4, 27 March 2023.** Fixed CVE-2023-32435 in WebKit, the Safari stage. That entry was added on 21 June 2023, credited to Kaspersky's researchers.
- **iOS 16.5.1 and iOS 15.7.7, 21 June 2023.** Fixed the kernel flaw CVE-2023-32434. The note says: "Apple is aware of a report that this issue may have been actively exploited against versions of iOS released before iOS 15.7."
- **iOS 16.6 and iOS 15.7.8, 24 July 2023.** Fixed the MMIO bypass, CVE-2023-38606; iOS 15.7.8 also carried the font fix for the older line.

Three visibility facts follow, and each belongs in the "Who can see it" column. First, a **silent fix** is real: the first fix in this chain shipped months before its notes described it, so a team that patches only when an advisory looks urgent would have waited for no reason. Second,

the "may have been actively exploited" sentence is the strongest signal Apple prints, and it arrived after the attack was public. Third, the attack worked on iOS 16.2 and earlier. A fleet that ran 16.3 or later had closed the first door before anyone outside Apple knew it was a door. The fastest defense in this case was not threat intelligence; it was a short patch clock on every device.

Three comparison chains

FORCEDENTRY's escape (2021). Book 2 told how FORCEDENTRY's first stage beat the image decoder. Project Zero's Ian Beer and Samuel Groß published the second half on 31 March 2022. The exploit escaped from the sandboxed `IMTranscoderAgent` process to "arbitrary code execution in the context of CommCenter", a more privileged process, and it was "notable for using only logic bugs": it abused `NSPredicate` and `NSExpression`, ordinary system features for evaluating queries, rather than corrupting memory. Project Zero did not name a CVE for the escape, and this book does not assign one. The lesson for the Ledger is that the sandbox around a parser is a boundary owned by the OS, and the OS's own trusted services are on the far side of it.

Trident (2016). On 24 August 2016 Citizen Lab reported a chain used against a human rights defender. It was **one-click**, not zero-click: "The one-click vector involves sending the target a normal SMS text message with a link to a malicious website." Apple fixed its three flaws, CVE-2016-4655, CVE-2016-4656 and CVE-2016-4657, in iOS 9.3.5 the next day, crediting Citizen Lab and Lookout. Trident is the landmark documented in-the-wild iOS chain of its time: a browser bug, then the kernel. Seven years later Triangulation needed no click, added a hardware step, and went through the kernel twice. Some chains skip the message parser altogether: on 16 March 2023 Project Zero reported eighteen flaws in Samsung's Exynos modems, four of which allowed "Internet-to-baseband remote code execution" by an attacker who needed only to "know the victim's phone number". These cases suggest the direction of travel is down the stack.

WhatsApp and ImageIO (2025). On 20 August 2025 Apple shipped iOS 18.6.2 to fix CVE-2025-43300, an out-of-bounds write in its ImageIO framework that "may have been exploited in an extremely sophisticated attack against specific targeted individuals". WhatsApp's August 2025 advisory assessed that its own flaw, CVE-2025-55177, in the handling of linked-device synchronization messages, "in combination with an OS-level vulnerability on Apple platforms (CVE-2025-43300), may have been exploited". One chain, two vendors' advisories, each fixing its own row.

What removed a path: the architectural answers

Patches close one chain. Architecture changes what the next chain must contain. Four vendor answers are on the public record, and each maps to a Ledger row.

- **Isolate the first parser (FR).** Project Zero described BlastDoor on 28 January 2021 as "a new, tightly sandboxed 'BlastDoor' service which is now responsible for almost all parsing of untrusted data in iMessages". It does not stop a chain; it forces the chain to add an escape, as FORCEDENTRY then did.
- **Shrink the surface for people at risk (FR, LR).** Lockdown Mode, announced on 6 July 2022, is a setting each user turns on: Apple says it "is not a configurable option for Mobile Device Management", and a device cannot enroll in MDM while it is on, so enroll first. Apple's support page says: "Most message attachment types are blocked, other than certain images, video, and audio", and certain complex web technologies are limited; Apple's announcement named just-in-time JavaScript compilation among them. Triangulation entered through an attachment and used JavaScriptCore and Safari; this book does not claim Lockdown Mode would have stopped it, because Kaspersky did not test that. The one public data point is Citizen Lab's of 18 April 2023: it had "not seen PWNYOURHOME successfully used against any devices on which Lockdown Mode is enabled". Android 16's Advanced Protection mode is Google's comparable switch for at-risk users.
- **Tag memory in hardware (SI, KN).** Arm's Memory Tagging Extension (MTE), defined in the Armv8.5 architecture, gives memory and pointers small tags that the CPU checks on each access, so many use-after-free and overflow bugs fault instead of giving control. Project Zero wrote in November 2023 that with the Pixel 8 "there's finally a production handset that allows you to enable MTE", as a developer option in Android 14; Android 16's Advanced Protection turns MTE on for supported apps. Apple announced **Memory Integrity Enforcement** on 9 September 2025: "always-on memory-safety protection covering key attack surfaces", including "the kernel and over 70 userland processes", built on an Enhanced Memory Tagging Extension, in the iPhone 17 models and iPhone Air.
- **Remove the instruction (FR).** The smallest answer was a deletion: a 1990s font instruction that, in Kaspersky's words, existed "before a patch removed it".

Our view (as of November 2026): hardware memory tagging is the most important of these for the rows below the frame, because it raises the cost of the kernel and runtime steps that every documented chain in this chapter needed. It is new, it depends on the exact chip, and the platform profile in Chapter Two must say which devices have it on.

Zero-click question	Windows	macOS	iOS	Linux	Android
Who parses messages before you click	Each messaging and mail app, and the OS preview handlers it calls	Apple's Messages and its parsing services, plus each third-party app	Apple's Messages through BlastDoor (iOS 14 on), plus each app	Each app and the libraries the distribution ships	The SMS and RCS app, each messaging app, and the platform media stack
One switch for at-risk users	No single named mode; set app and attachment policy per app	Lockdown Mode	Lockdown Mode	No single mode; distribution and app policy	Advanced Protection (Android 16)
Hardware memory tagging	Not offered as a platform default on the PCs in this book's profiles	Apple's September 2025 MIE announcement covered iPhone models, not Macs; check per chip	Memory Integrity Enforcement on iPhone 17 models and iPhone Air	Depends on the CPU and distribution build	MTE on supported devices (Pixel 8 on, developer option; Advanced Protection in Android 16)
What the advisory tells you	Security Update Guide marks exploited CVEs	Apple notes; exploited flaws marked "may have been actively exploited", sometimes added later	Same as macOS; Triangulation's first fix was listed months after release	Distribution advisories, per package	Monthly bulletin; exploited issues marked as possibly "under limited, targeted exploitation"
Firmware and silicon	Chip and firmware issues fixed by OEM firmware, microcode or OS mitigations (Chapter Three)	Fixed with OS updates on Apple silicon	Triangulation's chip-register bypass was mitigated by an OS update, iOS 16.6	Microcode and kernel mitigations from the distribution	Device maker and chip vendor, through the monthly bulletin

HOW IT GOT THIS WAY

Messaging apps parse rich media automatically because users want previews, stickers, fonts and links that render before they tap. Every one of those conveniences is code that runs on untrusted input with no click. The industry's answers followed the attacks: Trident in 2016 was answered with iOS 9.3.5 the next day, FORCEDENTRY's decoder bug with a sandboxed parser that was already in place and an escape that went around it, Triangulation with fixes in four releases and, two years later, hardware memory tagging on Apple's newest phones. Each answer moved the next attack one row lower.

What hasn't changed: fonts are still programs. The TrueType instruction set that Book 2 (Chapter 3) traced to 1991 is still interpreted by every TrueType-capable rasterizer. In FreeType's interpreter, `src/truetype/ttinterp.c`, opcodes `0x8F` and `0x90` are labeled `/* ADJUST */` and sent to `Ins_UNKNOWN`, that is, rejected as unknown; the label dates from commit `49d474f6f` (11 August 2016) and the rejection path from `d03a67a12` (11 January 2015). Apple's own interpreter ran the same-named instruction until the 2023 patch. Measured 18 November 2026 with `git blame` on the FreeType repository at `d7339729` (17 November 2026). One open-source engine had declined an instruction that another vendor kept for three decades.

THE UNSEEN ANGLE

The chain that began with one font instruction ended at chip registers that the firmware did not use and that no public document described. No customer could have put those registers in a risk register, because only the chip's designer knew they existed. What this case adds to Book 2's picture is a boundary that no customer knew existed. Kaspersky found the chain not with a phone-based product but by watching its own network traffic. For a company with a handful of high-value phones, one visible signal can be on the network, not on the device: an MDM inventory reports version and compliance, not this kind of traffic. Phones roam onto cellular networks, so the network is one source of evidence, not a detector. The other source is the vendor: since 2021 Apple has sent threat notifications, by lock-screen alert and email, to users it believes were "individually targeted by mercenary spyware attacks", in over 150 countries so far. Decide in advance who at the company such a user calls.

> *The parser is the front door. Triangulation shows that the house also has rooms its owner never listed.*

These rows in the Trust Ledger

Halbrook Bank, advisor and executive iPhones (about 120 supervised devices, the people a targeted chain would choose). The FR row is expanded because this is where the chain begins. Row codes and tags are as defined in Chapter One.

Component	Who can change it	Exposure and blast radius	Patch clock	Stop, revoke, recover	Who can see it	Support end	Decision and owner
FR Messages, attachment and font parsing (BlastDoor, FontParser, WebKit)	Apple writes, signs and ships; each user turns on Lockdown Mode; Halbrook sets app policy	Any sender who knows the number or account; one device per message, no click	Triangulation: T1 iOS 16.3 (23 Jan 2023) for the first flaw; Halbrook T3 target 7 days	Lockdown Mode on for the 120, turned on by each user after MDM enrollment; update enforcement by MDM with no user deferral	D (Apple notes, sometimes later); network monitoring O	Apple does not publish a date (Chapter Eleven)	**Rent**; **Isolate**: Lockdown Mode for advisors and executives; CISO; review each iOS release
KN XNU kernel	Apple only	Reachable after any user-space foothold	Triangulation: iOS 16.5.1 (21 Jun 2023)	Restore and re-enroll is the recovery; no customer control below that	U on device; D from Apple	As FR	**Rent**; **Own** the minimum iOS version; mobile lead
SI Apple A-series chip; PPL and MMIO registers	Apple designs; fixes ship as OS mitigations	Reachable only after kernel compromise; every device of that chip generation	Triangulation: iOS 16.6 (24 Jul 2023)	Behavior mitigated in software (iOS 16.6); the silicon changes only at refresh, so move to chips with memory tagging on a plan	U	Hardware support per Apple, unpublished	**Rent**; buy iPhone 17 class devices for the 120 at next refresh; procurement; review annually
MU iOS updates through MDM	Apple publishes; Halbrook enforces	All 120 devices	Measure T2 and T3 from MDM reports (nc today)	Enforced update with a deadline; noncompliant devices lose bank app access	O (MDM)	na	**Own** the deadline; mobile lead; review monthly

SECURITY LENS

Who can change it? For this chain, only Apple could ship fixes for the five rows it crossed; the customer controlled how fast the fix ran on its devices, whether its users had Lockdown Mode on, and who could reach those phones. On Android the same rows are split between Google, the device maker and the chip vendor, so the Ledger needs more names.

How fast can it be fixed? The vendor's fixes arrived across six months and four releases, and the first one before any public report. Your clock is the gap between a release and your 90% mark, and for a small group of high-value devices it can be days, because you can enforce updates without asking.

What does AI change here? Our view (as of November 2026): automated analysis helps attackers search parsers for bugs as much as it helps defenders spot odd traffic. Neither changes the architecture point: hardware tagging and isolated parsers raise the cost of every step after the first.

BUSINESS IMPLICATIONS

Security. Chains cross several owners in seconds. Halbrook's advisors and executives are the targets, not the branch tellers, so the strongest controls go on about 120 phones, not 2,500 desktops. Corvane's general managers carry the same risk on a smaller scale; Larkfield's code-signing key holders are its high-value people.

Cost. Chains like this are advertised by brokers at prices discussed in Chapter Twelve, which is why they are spent on few targets. Defense for those few is cheap by comparison: Lockdown Mode costs some convenience, and an earlier refresh of 120 phones to hardware with memory tagging is a small line in a bank's budget.

Visibility. A fully patched phone can look healthy to MDM while it talks to a command server. Kaspersky found Triangulation on its own Wi-Fi. Where law and staff agreements allow, send high-value devices through a monitored network path and keep the logs; DNS and flow records show where a phone talks, not what it says.

Value. The people a chain targets hold the decisions, the client relationships and the keys. Protecting them is protecting the value, and it is one of the few places in this book where a strong control costs little.

Monday rule: list the people in your company worth a zero-click chain, have them turn on Lockdown Mode or Advanced Protection, and enforce updates on their devices within seven days of release.

OUR VIEW, AS OF NOVEMBER 2026

We expect the next documented chains to spend more of their effort below the kernel, where Triangulation found an undocumented door, and less in the parser, which isolation and memory-safe rewrites are shrinking, though attackers will keep choosing the cheapest chain that works. Customers cannot audit chip registers. What they can do is buy devices whose platform turns on hardware memory tagging, keep high-value phones within days of the current release, and watch the network.

CHAPTER ELEVEN

The Long Tail: End of Support, Edge Appliances and Security Debt



On 5 February 2026 the US Cybersecurity and Infrastructure Security Agency issued Binding Operational Directive 26-02, "Mitigating Risk From End-of-Support Edge Devices". It told federal civilian agencies to inventory every device on CISA's end-of-support edge device list within three months, to decommission those devices within twelve, to remove every end-of-support edge device within eighteen, and to "establish a process for continuous discovery of all edge devices". A government had concluded that firewalls and VPN gateways nobody will fix any more are a risk worth a deadline. The decision this chapter changes is plain: every row of the Trust Ledger gets a support-end date, with its source, and every purchase order for a device that will live longer than its software states who fixes it and until when.

Unsupported is a date, not a feeling

End of support is the date after which the owner of a component no longer commits to fix it. It does not mean the component stops working, and it does not mean it can never be patched. It means the ordinary promise is over: no scheduled security updates, no obligation to respond. Microsoft's lifecycle policy puts it carefully: "Older products may not meet today's more demanding security requirements. Microsoft may be unable to provide security updates for older products."

Three things blur that line in practice, and the Ledger should record each separately.

- **Paid extensions.** Windows 10 reached end of support on 14 October 2025. Microsoft sells Extended Security Updates to organizations at "$61 USD per device for Year One", and "the price doubles every consecutive year, for a maximum of three years". It brings "critical and important security updates" only. That is a bridge, priced to end.

- **Exceptional fixes.** Some vendors patch an old line when a flaw is exploited. Apple shipped iOS 16.7.11 on 31 March 2025 with fixes for CVE-2025-24200 and CVE-2025-43200 (the second listed only from June 2025), noting that one "may have been exploited in an extremely sophisticated attack against specific targeted individuals", and it was still releasing iOS 15.8.8 and 16.7.16 on 11 May 2026. Useful, but not a commitment you can plan on.
- **Different clocks inside one device.** A kiosk's OS can be supported while its firmware, its payment driver or its boot certificates are not. Chapter Four showed that the 2011 Secure Boot certificates expired between June and October 2026 on devices whose operating system was fully supported. Those devices "will continue to start and operate normally", Microsoft says, but can no longer receive new protections for the early boot process.

So "unsupported" in the Ledger means one thing: no ordinary vendor commitment to fix it after a stated date. The cell holds that date, its source and the lead time to replace the component, because the replacement is usually the only lasting fix.

The support calendar, per platform

Each platform publishes its promise differently, and some do not publish one at all.

Windows. Microsoft publishes exact dates per release on its lifecycle pages. The long-term servicing channel (LTSC) editions used in kiosks and other fixed-function devices run much longer than the general releases: Windows 10 IoT Enterprise LTSC 2021 to 13 January 2032, Windows 10 IoT Enterprise LTSC 2019 to 9 January 2029, Windows 11 IoT Enterprise LTSC 2024 from 1 October 2024 to 10 October 2034. The non-IoT editions are shorter: Windows 10 Enterprise LTSC 2021 ends on 12 January 2027 (with paid ESU offered to 8 January 2030), Windows 11 Enterprise LTSC 2024 on 9 October 2029. A kiosk that looks like "Windows 10" may therefore be supported until 2032 or have lost support in October 2025; the edition decides, and the Ledger records it.

Apple. Apple does not publish an end date for iOS, iPadOS or macOS support. Where UK law requires it, Apple publishes a floor: its statement for a 13-inch iPad Pro gives a "Minimum 5 Years from the first supply date". Its deployment guide says that "not all known security issues are addressed in previous versions", and its security releases page shows what actually happened: older lines receive selected fixes for a time, then stop. Beyond that floor, any support-end date for an Apple device in your Ledger is an inference from history. Label it **I**, and keep the exploited-bug fixes for old lines separate from the support for the current major release.

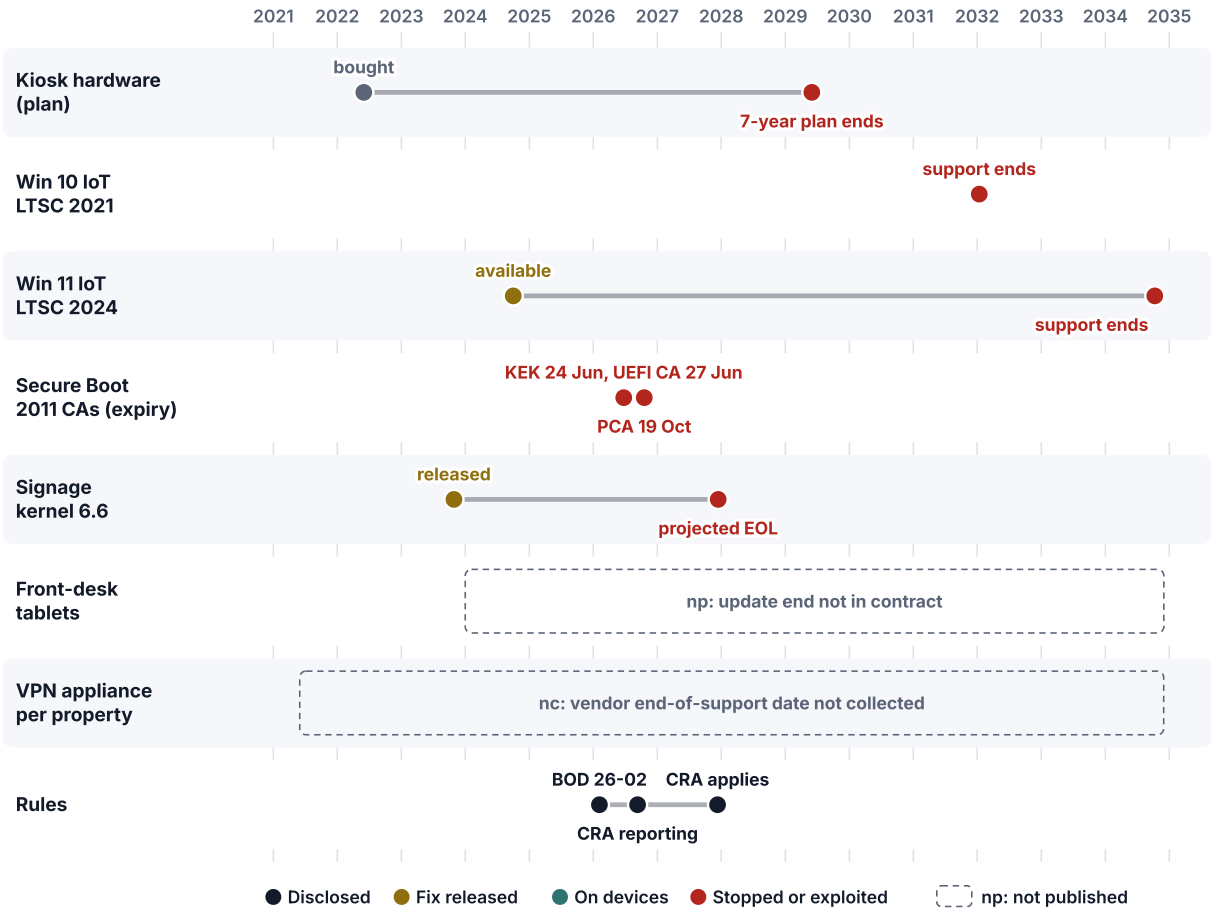
Android. The promise is the device maker's, model by model. A sample of three published policies: Google committed in October 2023 to "seven years of software support for the Pixel 8 and Pixel 8 Pro", counted "from when the device first became available on the Google Store in the US"; Samsung announced for the Galaxy S24 in January 2024 "seven generations of OS upgrades and seven years of security updates"; Fairphone commits to "eight years of updates and software support" for the Fairphone 5, "until 2031". Three makers are a sample, not the market. Nor is the monthly bulletin a steady clock: Google's July 2025 Android bulletin said "There are no Android security patches", and September's listed more than 100 CVEs. For a tablet with no published promise, the cell says **np**, and that is the finding.

Linux. The kernel community publishes projected end-of-life dates for its longterm kernels, and says each "usually starts with only a 2-year projected EOL that can be extended further" and that the dates "are not set in stone". On 18 November 2026 the page listed 6.18 and 6.12 to December 2028, 6.6 and 6.1 to December 2027, and 5.15 and 5.10 to December 2026. Most companies do not run kernel.org kernels directly; they run a distribution or a device maker's build with its own lifecycle. The Ledger cell names whose date it is.

The law. The UK's Product Security and Telecommunications Infrastructure (PSTI) regulations, enforced from 29 April 2024, make makers of consumer connectable products state a defined support period. The EU's ecodesign regulation for smartphones and tablets, applying from 20 June 2025, requires operating-system updates for at least five years after a model's last unit is placed on the market. For products sold in the EU, the Cyber Resilience Act now makes the support period a duty. Article 13(8) requires manufacturers to set a support period that reflects expected use, and "the support period shall be at least five years" unless the product is expected to be in use for less. The regulation applies from 11 December 2027, with the reporting duties of Article 14 from 11 September 2026. For Larkfield, a manufacturer under the Act, the support period for versions it places on the EU market from December 2027 becomes a published commitment, not a release-notes habit.

FIGURE 11.1

Corvane's kiosk calendar: every row has its own end date, and the ones nobody published are the risk



Published support ends and certificate expiry dates for the software on Corvane's check-in kiosks and property estate, set against the hardware plan; dashed boxes are dates no vendor has given Corvane. KEK: key exchange key; PCA: production certificate authority.

Source: Microsoft Lifecycle pages for Windows 10 IoT Enterprise LTSC 2021 and Windows 11 IoT Enterprise LTSC 2024; Microsoft KB5062710; kernel.org releases page; Regulation (EU) 2024/2847 Art. 71; CISA BOD 26-02. Hardware plan, tablets and VPN lanes are Corvane's (invented, illustrative).

Figure 11.1 puts Corvane's kiosk estate on one calendar, and it shows a mismatch. The kiosks were bought on a seven-year plan, the OS edition runs to 2032, the boot certificates expired in 2026 (no new boot-level protections after that), the signage kernel's projected end is December 2027, and two lanes have no date at all. The tail of a fleet is not one date; it is the earliest date in the row nobody wrote down.

Support question	Windows	macOS	iOS	Linux	Android
Is an end date published?	Yes, per edition, on Microsoft lifecycle pages	No; inferred from history (tag I)	No end date; a UK PSTI minimum per model; end inferred from history (tag I)	Kernel: projected per longterm release; distributions publish their own	Per device maker and model; absent for many devices
Long-life option for fixed-function devices	LTSC and IoT LTSC editions (to 2032 and 2034 for current IoT editions)	None separate from the current releases	None separate from the current releases	Longterm kernels and long-life distribution releases	Depends on the device maker; check the written promise
Paid bridge after the end	ESU: $61 per device in year one, doubling, up to three years (Windows 10)	None offered	None offered	From distribution vendors, by contract	None from the platform
Fixes for older versions	Only through ESU	Selected exploited fixes for older releases, no commitment	Selected exploited fixes for older lines, no commitment (iOS 15.8.8, 16.7.16 in May 2026)	Whatever the distribution backports	Whatever the device maker ships
Firmware and silicon	OEM firmware support end, rarely published per model; Secure Boot certificates (Chapter Four)	Tied to Apple's OS support for the Mac model	Tied to Apple's OS support for the model	Vendor firmware via LVFS or OEM tools, where offered	Chip vendor support for the board decides how long the device maker can patch

Edge appliances: outside the screen, on the path

The **EX** row of the Trust Ledger holds what sits outside the screen but on the attack path: VPN gateways, firewalls, remote-access appliances, load balancers. They do not draw a pixel. They do decide who reaches the systems that do, and they are often the first thing an intruder touches.

FIGURE 11.2

Edge appliances on the attack path: what the annual reports and one directive say

Source	Data period	What it says about edge devices
Mandiant M-Trends 2024	2023 investigations	Exploits were the initial vector in 38% of intrusions; "the first and third most targeted vulnerabilities were related to edge devices"
Mandiant M-Trends 2025	2024 investigations	Exploits were the initial vector in 33%; "the most frequently exploited vulnerabilities affected security devices" at the network edge
Verizon DBIR 2025	Incidents analyzed for the 2025 report	Edge devices and VPNs were 22% of vulnerability exploitation targets, up from 3%; about 54% fully remediated, median 32 days
GTIG zero-day review 2024	2024	75 zero-days exploited; 44% in enterprise products; 20 in security and networking products
GTIG zero-day review 2025	2025	90 zero-days exploited; 43 (48%) in enterprise software and appliances; 21 in security and networking products
CISA BOD 26-02	Issued 5 Feb 2026	US federal agencies: inventory end-of-support edge devices in 3 months, decommission listed ones in 12, all in 18

Five independent sources over three years point at the same row, the appliance at the edge of the network; each uses its own population and method, so the figures are not a trend and should not

be combined.

Source: Mandiant M-Trends 2024 and 2025; Verizon 2025 Data Breach Investigations Report, executive summary; Google Threat Intelligence Group, zero-day reviews of 2024 (29 April 2025) and 2025 (5 March 2026); CISA BOD 26-02 (5 February 2026). Each figure is the publisher's own, with its own method.

Figure 11.2 collects what the annual reports say, each in its own words and with its own method. Mandiant's M-Trends 2024 found that in its 2023 investigations "the first and third most targeted vulnerabilities were related to edge devices"; M-Trends 2025 found for 2024 that "the most frequently exploited vulnerabilities affected security devices" at the network edge. Verizon's 2025 Data Breach Investigations Report found edge devices and VPNs were 22% of the targets of vulnerability exploitation, more than seven times the 3% of the year before, and that only about 54% of those vulnerabilities were fully remediated, at a median of 32 days. Google's Threat Intelligence Group counted 20 security and networking products among the 75 zero-days exploited in 2024, and 21 among 90 in 2025.

The government record is more direct. CISA added the Citrix NetScaler flaw known as Citrix Bleed, CVE-2023-4966, to its Known Exploited Vulnerabilities (KEV) catalog on 18 October 2023, and Fortinet's FortiOS SSL VPN flaw CVE-2024-21762 on 9 February 2024. For Ivanti Connect Secure and Policy Secure (CVE-2023-46805 and CVE-2024-21887, both added to KEV on 10 January 2024) it issued Emergency Directive 24-01 on 19 January 2024, then Supplemental Direction on 31 January ordering agencies to "disconnect all instances" by 2 February. For Cisco firewalls, Emergency Directive 25-03 of 25 September 2025 went further for old hardware: agencies had to "permanently disconnect these devices on or before September 30, 2025" where the models had reached end of support. That is the long tail and the edge in one sentence: an exploited appliance that the vendor no longer fixes has no patch path, only an exit. Barracuda reached the same point in June 2023 for its Email Security Gateway (CVE-2023-2868): "Compromised ESG appliances must be immediately replaced regardless of patch version level."

For Corvane, the edge is 300 VPN appliances, one per property, bought with the network contract and managed by an integrator. They are on the recovery path in Chapter Nine's Figure 9.3 and on the attack path here. Nobody at Corvane has their end-of-support dates. That is the lane marked **nc** in Figure 11.1.

n-days and the tail

A **zero-day** is exploited before a fix exists; an **n-day** is exploited after the fix is published, against devices that have not applied it. The tail of a fleet, and everything past its support end, lives in n-day territory: the fix exists somewhere, but not on that device.

Mandiant's analysis of 2023, published by Google Cloud on 16 October 2024, counted 138 vulnerabilities exploited that year, 97 as zero-days and 41 as n-days, and an average time to exploit of 5 days. The zero-day count is the headline; the n-day count is the long tail's bill. For an unsupported device, every later flaw that also affects the old code is a permanent n-day: the fix may exist for supported versions, and attackers can study it to find the bug in the old one.

Our view (as of November 2026): for most companies the larger, steadier risk sits in the tail, not at the zero-day front. Zero-days are spent on chosen targets (Chapter Ten). The n-day against an appliance two years past its support end is open to anyone who reads the advisory.

Paying down security debt

Security debt is the work a team owes because a component outlived its support or was never updated: the refresh not budgeted, the appliance not replaced, the old edition kept because one app needs it. Like financial debt it has a due date and it charges interest, in exposure and, through ESU and emergency projects, in money.

- **Write the date before you buy.** A purchase order for a device that will run five to ten years states the OS edition, the firmware support term, the device maker's update promise and the appliance vendor's end-of-support policy. Where there is none, the Ledger records np at purchase, not at the incident.
- **Price the bridge.** ESU at $61 per device in year one, doubling each year, is a known cost. For 3,000 kiosks left on a general Windows 10 release instead of an LTSC edition, that would be about $183,000 in the first year if every kiosk qualified, before the doubling: simple arithmetic on Microsoft's published price, and a reason to choose the IoT LTSC edition at purchase.
- **Pin with an end date, isolate what is past it.** A component kept past support is a **Pin** in the Ledger, and a Pin needs an end date. While it lasts, **Isolate** it from what it could reach: no inbound management from the internet, its own network segment, no shared credentials.
- **Plan the refresh as one budget year.** A fleet bought together reaches its end together. Spread purchases, or plan the year when they all fall due.

HOW IT GOT THIS WAY

Devices outlive software promises. A kiosk, a branch PC or a firewall is bought on hardware life, depreciated over years, and kept as long as it powers on. The software inside it is bought on a support promise that was usually written by the vendor, not by the buyer, and rarely appears on the purchase order. When Windows editions, phone models and appliances had short lives, the mismatch was small. Long-life editions, seven-year phone promises and regulations such as the CRA are the industry's slow answer: they make the date longer and more visible, not unnecessary.

What hasn't changed: for fixed-lifecycle products, the date is set at launch. Microsoft's Fixed Lifecycle Policy still promises "a defined support and servicing Lifecycle timeline at the time of product launch" and "a minimum of five years Mainstream Support" (Microsoft Learn, Fixed Lifecycle Policy, checked 18 November 2026; its current revision took effect on 13 April 2010). Windows 11 IoT Enterprise LTSC 2024, launched on 1 October 2024, carries an end date of 10 October 2034 from its first day. The buyer has had ten years' notice. What usually goes missing is the copy of that date in the buyer's own records.

THE UNSEEN ANGLE

Security debt has a due date, and most fleets cannot say what it is. Endpoint tools report OS versions; they rarely report the firmware's support term, the tablet maker's update promise or the VPN appliance's end-of-support date, because those live in vendor pages and contracts, not on the device. That is why CISA's directive starts with an inventory and with "continuous discovery of all edge devices": the government's first step was to find the devices, not to patch them. A Trust Ledger row with no support-end date is the first audit finding, before any vulnerability scan.

> *Fleets are bought on hardware life and abandoned on software life. The gap between the two is the long tail.*

These rows in the Trust Ledger

Corvane Hotels, check-in kiosk (about 3,000 kiosks at 300 properties) and the property edge. Column 7 is filled for every row; the other columns are summarized. Row codes and tags are as defined in Chapter One.

Component	Who can change it	Exposure and blast radius	Patch clock	Stop, revoke, recover	Who can see it	Support end	Decision and owner
KN Windows 10 IoT Enterprise LTSC 2021 on kiosks	Microsoft writes and ships; Corvane sets rings	Every kiosk; guest-facing	Monthly; measured per ring (Chapter Nine)	Golden image per kiosk model	O (fleet reports)	13 Jan 2032 (Microsoft lifecycle page)	**Rent**; **Pin** to this edition until the kiosk refresh (hardware plan ends 2029); new kiosks on LTSC 2024 (to 2034); end-user computing lead
FW Kiosk OEM firmware and Secure Boot certificates	Independent BIOS vendor writes, OEM signs	Every kiosk	OEM releases, irregular	OEM firmware update only; 2023 certificate status per device (Chapter Four)	D	OEM firmware support: np. 2011 certificates expired June to October 2026	**Rent**; demand a written firmware support term at next purchase; procurement
DR Payment-terminal driver	Terminal vendor	Kernel on every kiosk; payment data	Per vendor	Rollback by image only	D	Per contract: nc	**Rent**; add support end and fix SLA to the contract; payments lead
MU Front-desk Android tablets	Device maker and Google	Staff devices at 300 properties	Monthly bulletin, then maker's image	EMM wipe and re-enroll	O (EMM)	np: no update end in the contract	**Rent**; buy only models with a published update end date; IT procurement
EX VPN appliance, one per property	Appliance vendor; integrator configures	Internet-facing; reaches the property network and the recovery path	Vendor advisories; KEV entries for this class	Replace or disconnect; no tested failover	U	nc: vendor end-of-support date not collected	**Rent**; collect dates within 30 days, **Isolate** management interfaces, replace any past end of support; network lead

SECURITY LENS

Who can change it? After the support end, nobody is obliged to answer, short of a paid extension or third-party support. The vendor may fix it as an exception; you cannot make it. Your only authority left is to isolate it or replace it.

How fast can it be fixed? Past the support end, the patch clock has no T1. The clock that matters becomes the replacement lead time: procurement, imaging and site visits, which for 300 properties is months, not days.

What does AI change here? Our view (as of November 2026): AI-assisted analysis makes it cheaper to compare a fixed version with an old one and find the flaw the fix closed, which shortens the time from a published fix to a working n-day. That makes unsupported components more exposed every year they stay, not less.

BUSINESS IMPLICATIONS

Security. Unsupported means no ordinary vendor commitment to fix, not "unpatchable", and the difference matters for planning. Corvane's largest gap is its edge: 300 appliances with no recorded support end. Halbrook's is its branch PCs on whatever edition each refresh brought; Larkfield's is the opposite side, its own promise of support to 400,000 desktops under the CRA.

Cost. Extended support is priced to push upgrades: $61 per device for year one, doubling. A forced refresh lands in one budget year unless purchases were spread. Larkfield's cost is the engineering to keep old releases patched for the support period it publishes.

Visibility. Few fleets track support ends per row. Add the column to the asset inventory, with the source of each date and an evidence tag; an empty cell is the finding.

Value. The long tail is where a planned refresh beats an emergency one. A board can understand a calendar with dates and costs; it cannot act on "legacy risk".

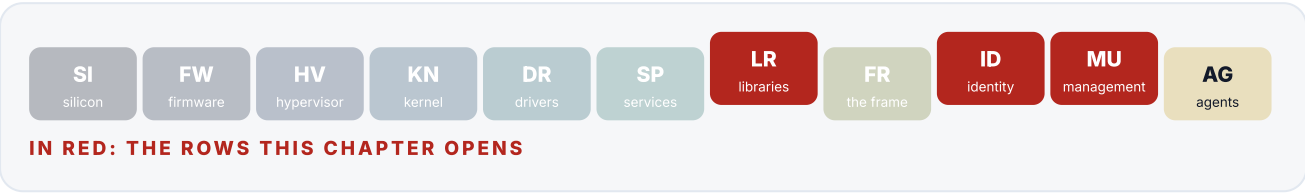
Monday rule: pick one critical system, fill column 7 for every row with a date and its source, and list every edge appliance with its end-of-support date; any cell you cannot fill is next quarter's first task.

OUR VIEW, AS OF NOVEMBER 2026

Published support periods will keep getting longer and more binding, through phone makers' seven-year promises, long-life OS editions and the CRA's five-year floor from December 2027. They will not remove the long tail, because devices will still outlive their dates and fleets will still lose track of appliances. The companies that do well will be the ones that hold the date in their own Ledger and act on it a year early.

CHAPTER TWELVE

Who Pays for Security: The Market



On 10 October 2025 Apple said it would double the top award in its security bounty to $2 million for an exploit chain that reaches the goals of a mercenary spyware attack, with bonuses that could take one payout above $5 million. An exploit broker's price list, published in April 2024, asks between $5 million and $7 million for a zero-click full chain against iOS. Both numbers are about the same target, a full chain against a phone, though they buy different things, and neither is the price of your security. They are two signals from a market that decides how many people look for flaws in the layers you rent, who learns about a flaw first, and who pays when it is used. This chapter follows that money. The decision it changes is small and concrete: for each row of your Trust Ledger, add the optional column "who pays on failure", and fill it before you sign the next renewal.

Five parties, one stack

Every row of the Trust Ledger (Chapter 1) has an authority that can change it. Behind every authority is a budget. Five groups take part in paying for, or doing, the security work on the layers below your screen, each for a different reason.

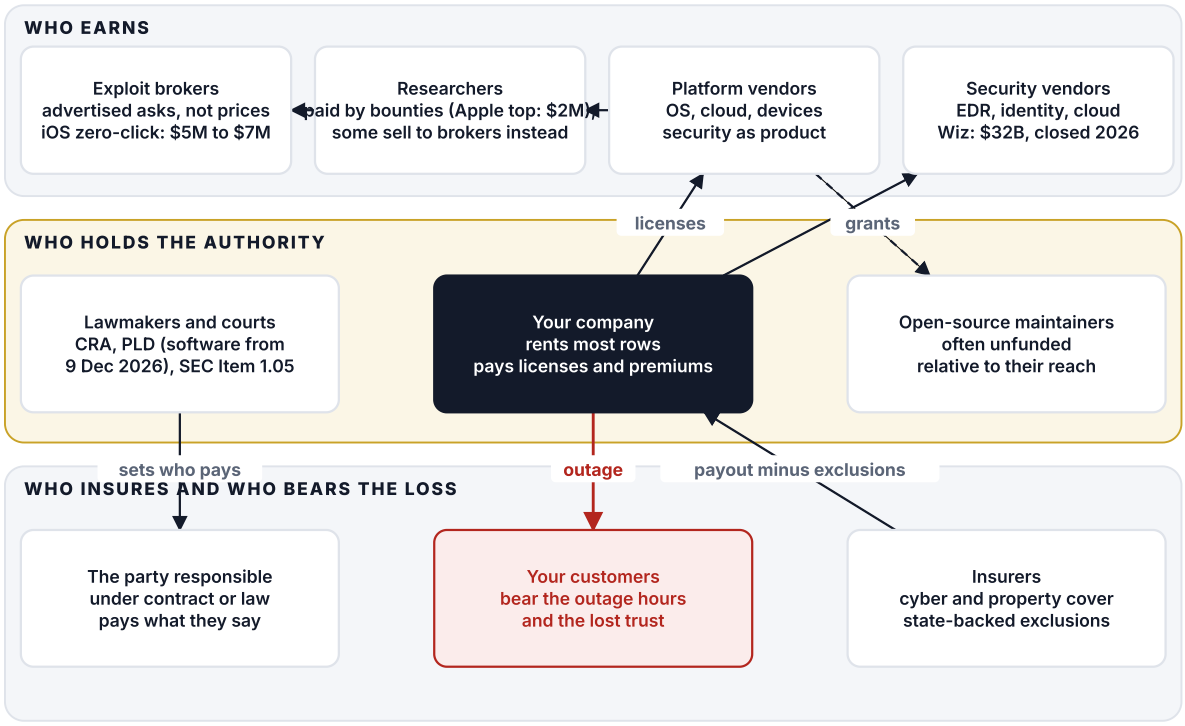
- **Platform vendors** pay because security is part of what they sell and part of what keeps customers on the platform. Microsoft launched its Secure Future Initiative on 2 November 2023 with engineering goals for software development, identity protection and faster vulnerability response. The money comes from licenses, subscriptions and hardware margins.
- **Security vendors** sell controls you add to the stack: endpoint detection and response (EDR) sensors, identity services, cloud posture tools. Their scale shows in one dated fact: Google announced its $32 billion deal for Wiz in March 2025 and completed it on 11 March 2026.

- **Researchers** are paid by bounty programs to report flaws to the vendor, or by brokers to sell them to someone else.
- **Open-source maintainers** write much of the code in the libraries and runtimes row (LR), and many are paid by nobody for that work.
- **You** pay twice: once in licenses and premiums, and again, in the part of the loss nobody else covers, when a row fails.

Insurers stand beside you and pay part of some losses, under exclusions you agreed to. Lawmakers and courts decide, slowly, how much of the loss can be moved back to the authority that caused it. Figure 12.1 puts these flows on one page.

FIGURE 12.1

Who earns, who holds the authority, who insures and who bears the loss



Money flows toward the authorities that change your systems; the loss from a failed change flows the other way, to you and your customers, unless a contract, a policy or a law moves it back.

Author's diagram based on Apple Security Research (10 Oct 2025), Google VRP 2025 review (2026), Crowdfense program page (2025), Google press release (11 Mar 2026), Regulation (EU) 2024/2847, Directive (EU) 2024/2853, SEC final rule (26 Jul 2023). Amounts are dated; broker figures are advertised asks.

The figure has one asymmetry worth naming before the details. Most of the money flows toward the authorities that can change your systems. The loss from a failed change flows away from them, to you and your customers, unless a contract, an insurance policy or a law moves it back. The rest of this chapter is about those three ways back.

Bounties and brokers: two numbers that are not one price

A **bug bounty** is a published reward a vendor pays for a vulnerability reported to it, so it can fix the flaw. The large programs now publish what they pay, and the numbers are dated:

- **Apple** announced on 10 October 2025 that, from November 2025, its top award would be $2 million for exploit chains that reach mercenary-spyware goals, and that bonuses for bypassing Lockdown Mode or finding bugs in beta software could take a payout "in excess of $5 million". Apple said it had paid more than $35 million to more than 800 researchers since 2020.
- **Google** said it paid more than $17 million in 2025 to more than 700 researchers across its programs, a record for the program and more than 40% above 2024.
- **Microsoft** said on 5 August 2025 that its bounty programs had paid $17 million in the past year to 344 researchers from 59 countries.

An **exploit broker** buys working exploits and sells them on, usually to clients it does not name. Crowdfense, a broker that publishes its program, updated its list in April 2024 with asks such as "iOS Zero Click Full Chain (e.g. iMessage): from 5 to 7 M USD" and "Android Zero Click Full Chain (e.g Whatsapp, RCS): 5 M USD", and says its acquisition budget has grown to $30 million. Its page does not name its clients.

It is tempting to set the two lists side by side and say that the defenders are outbid. Do not. The numbers measure different things. A bounty pays for a report: the vendor gets the bug and can fix it, though the fix still has to reach every device that runs the code. A broker's figure is an advertised ask for an **exclusive, working, chained capability**, often with a promise to keep it working after updates, and a broker's buyer pays for the chance that the flaw stays unfixed. A bounty maximum is the top of a range most reports never reach; a broker's number is marketing, published by a buyer that wants sellers. Neither is a market price in the sense an economist would accept, because the trades on the broker side are secret. What both numbers do show, dated and sourced, is direction: the published price of a zero-click chain against a phone has been rising on both sides, and a working chain against a phone is worth millions to someone.

For a CTO, the practical lesson is narrower than the headlines. The bounty of the vendor that owns a row is one signal of that row's intake: it pays for flaws to arrive at the vendor instead of at a broker. It says nothing about triage, engineering or how fast the fix reaches your fleet.

Ask your platform and security vendors whether they run a public bounty, what it covers, and whether the component you rent from them is in scope. A row whose owner runs no bounty, publishes no advisories and has no security contact is a row where you are more likely to learn about the flaw from someone else.

Who pays for the fix	Windows	macOS	iOS	Linux	Android
Who funds the OS fix	Microsoft, from licenses and subscriptions	Apple, from hardware and services	Apple, from hardware and services	The distribution: a paid enterprise vendor, or volunteers for community builds	Google for Android itself and its own devices; each device maker for its builds
Public bounty for the OS	Microsoft bounty programs	Apple Security Bounty	Apple Security Bounty (top award $2 million from November 2025)	Per project; no single owner pays for the whole stack	Google's reward programs for Android and Google devices
Who pays for fixes after the support end	You, where Microsoft sells extended updates for a release (Chapter 11)	No paid extension; replace the Mac	No paid extension; replace the device	You: a paid enterprise distribution whose vendor backports fixes, or your own backports	The device maker's promise, or nobody (Chapter 11)
Firmware and silicon	Chip vendors and device makers fund their own fixes; the fix reaches you only through an OEM firmware update or an OS update (Chapters 3 and 4), so the OEM's support budget is part of your patch clock.				

The unpaid rows: open source after Heartbleed

In this book's cases the libraries row is where the market is thinnest. On 7 April 2014 the OpenSSL project published the advisory for CVE-2014-0160, the flaw known as Heartbleed, in the library that encrypted a large share of the web's traffic. Days later Steve Marquess, then running the OpenSSL Software Foundation, wrote that the foundation typically received about $2,000 a year in donations. The code was everywhere; the money was not. Log4Shell in December 2021 (Chapter 8) repeated the pattern in a Java logging library kept by volunteers.

The response came as industry funds, each a few years apart:

- **Core Infrastructure Initiative**, announced by the Linux Foundation on 24 April 2014, with founding companies pledging money over three years to fund projects on the critical path of the internet, OpenSSL first.
- **Open Source Security Foundation (OpenSSF)**, launched by the Linux Foundation on 3 August 2020, which folded the Core Infrastructure Initiative into a wider cross-industry body.

- **Alpha-Omega**, launched under OpenSSF on 1 February 2022 with an initial $5 million from Microsoft and Google, to fund security work in the most critical projects and to scan thousands more.
- **The Sovereign Tech Fund**, set up by the German government in 2022 and now run as the Sovereign Tech Agency, which a European Commission case study (June 2025) credited with more than EUR 24.6 million for more than 60 open-source projects.

These funds are real and they have changed specific projects. They are also small next to the reach of the code. Chapter 8 tells the xz-utils case of 2024, where a long campaign to take over a compression library ran through a project kept alive by volunteers; this chapter adds only the money question. When your Ledger's LR row lists a dependency, the honest entry in "who pays on failure" is often "nobody we have a contract with". The EU laws below keep it that way on purpose: the Product Liability Directive does not apply to free and open-source software developed or supplied outside a commercial activity, and the Cyber Resilience Act gives non-profit "open-source software stewards" (Article 24) lighter duties than manufacturers. The maker that ships the package inside a commercial product, such as Larkfield, carries the duty for it. That is not a reason to avoid open source. It is a reason to fund, or at least to watch, the few components your business cannot run without.

Insurance follows attribution

Cyber insurance is the one contract written for the purpose of paying for a failed row, whoever caused it; SLA credits and indemnities pay only for their own vendor's failures. The NotPetya cases of 2017 show what that contract is worth when the failure has a state behind it. The full case is in Chapter 13; here is what the insurance record says.

Merck. NotPetya hit Merck in June 2017 through a Ukrainian accounting program. Merck claimed under its property insurance, and its insurers relied on an exclusion for "hostile/warlike action". On 1 May 2023 the Appellate Division of the New Jersey Superior Court (Merck & Co. v. ACE American Insurance Co., No. A-1879-21) held that the exclusion did not cover a cyberattack on a nonmilitary company: the wording, written for armed conflict, did not reach this event. The parties settled in January 2024, before the New Jersey Supreme Court ruled, so the decision stands without a final word from the state's highest court.

Mondelez. Mondelez sued Zurich American in Illinois state court after Zurich invoked a war exclusion to deny a claim of over $100 million. The parties settled in late 2022, during the trial, on terms they did not disclose. No court ruled on whether NotPetya was an act of war under that policy.

The market did not wait for the courts. In August 2022 Lloyd's of London told its managing agents (Market Bulletin Y5381) that standalone cyber-attack policies written from 31 March 2023 must, at a minimum, exclude losses from war and from state-backed cyber attacks that significantly impair a state's ability to function or its security capabilities, be clear about systems outside the affected state, and set out a robust basis for how the parties will attribute an attack to a state. Model clauses from the Lloyd's Market Association lean primarily on attribution by the government of the state where the affected computer system is located.

Put those facts together and a new dependency appears on your Ledger. Whether a loss is covered can turn on a government's public attribution. For NotPetya, the UK and US governments attributed the attack on 15 February 2018, almost eight months after the event. A claim filed in the weeks after an attack may be paid, disputed or denied depending on a statement nobody at your company controls, made on a clock nobody at your company sets. The insurance row, in other words, has a patch clock of its own.

Liability moves: from disclaimers to duties

For most of the history of commercial software, the license carried the liability decision. The vendor disclaimed warranties and capped damages, the customer accepted, and the cost of a defect stayed where it landed. Three moves since 2023 have started to change that, at different speeds.

The United States: a proposal. The National Cybersecurity Strategy of 2 March 2023 listed, under its third pillar, an objective to "shift liability for insecure software products and services". It proposed working with Congress on legislation that would prevent makers "with market power from fully disclaiming liability by contract" and would offer a safe harbor to vendors that build and maintain software securely. It needed legislation, and no such law had passed by the source freeze of this book. Executive Order 14306 of 6 June 2025 then amended the two earlier cyber orders, 13694 and 14144, and struck several of the secure-software requirements that the January 2025 order (14144) had placed on federal contractors through the acquisition rules. The 2023 strategy was not revoked; the liability shift it proposed remains a proposal.

The European Union: duties with dates. Two laws move cost toward the maker of software sold in the EU:

- The **Cyber Resilience Act**, Regulation (EU) 2024/2847, entered into force on 10 December 2024. Its reporting duties (Article 14) apply from 11 September 2026: a manufacturer must give an early warning within 24 hours of becoming aware of an actively exploited vulnerability in its product. Most of the remaining obligations apply

from 11 December 2027, including security requirements and a support period, which Article 13(8) says "shall be at least five years", or the expected use time where a product is expected to be in use for less.

- The revised **Product Liability Directive**, Directive (EU) 2024/2853, adopted on 23 October 2024, names software expressly as a product. A claimant must still show a defect, damage the directive covers (death, personal injury, damage to property, or destroyed or corrupted data not used for professional purposes) and causation, under national law that transposes it. It applies to products placed on the market or put into service after 9 December 2026, weeks after this book's source freeze.

Vendors: commitments. The large vendors made public promises that are not law but are written down: Microsoft's Secure Future Initiative (2 November 2023) is one; Chapter 5 covers Microsoft's later work with security vendors on kernel access. A promise written in a blog post is evidence of intent. A promise written in your contract is evidence you can use, and the contract's liability cap, its carve-outs and any indemnity are where most "who pays" questions are settled today (Chapter 15).

None of this moves money to you on its own. What it changes is what a vendor must do, by when, and what a court may later ask about. For Larkfield Software, the ISV in this book that sells a desktop app in the EU, it changes more than that: Larkfield now carries reporting duties by law and, for products placed on the market after 9 December 2026, can be the party that pays when a defect causes damage the directive covers. For a vendor whose installs are mostly in businesses that reach is narrow: the directive excludes property used only for professional purposes and data used for professional purposes.

HOW IT GOT THIS WAY

The licenses that shipped with commercial software from the 1980s onward disclaimed warranties and capped damages, and buyers had little choice but to accept them. The 2023 US strategy states the result plainly: software makers could use their market position to disclaim liability by contract, which weakened the incentive to test before release and to build securely. Patching became a free service the vendor offered on its own schedule, not a duty with a clock. The first bounty programs at the large platform vendors and the industry funds after Heartbleed were voluntary answers to the same gap: they paid for security where no contract or law required it. The EU acts of 2024 are the first broad attempt in this book's sources to turn those voluntary answers into duties with dates.

What hasn't changed: in April 2014 the library that protected much of the web's traffic received about $2,000 a year in donations, by its foundation's own account; ten years later, in March 2024, the xz case (Chapter 8) again ran through a widely used component maintained by volunteers. The pattern holds in those two cases, which is enough to put a funding question on every LR row.

THE UNSEEN ANGLE

Your cyber insurance policy has a Trust Ledger of its own, and few companies draw it. Take the exclusions (war, state-backed attack, infrastructure failure, unpatched known vulnerabilities, contractual liability) and map each one to the rows it can void. Depending on its wording, a state-backed exclusion can reach any row an intelligence service might use, including your MU and DR rows, the ones that matter most in Chapter 13. Where a policy has an "unpatched vulnerability" clause (wordings vary), it turns your patch clock (T3 and the tail) into a coverage condition. Do this before the broker meeting, not after the claim. Sources: Lloyd's Market Bulletin Y5381 (August 2022) for the state-backed requirement and its attribution basis; Merck v. ACE (N.J. App. Div., 1 May 2023) for what an old war clause did and did not cover.

> *A bounty buys the bug for the vendor. A broker's ask is for the bug to stay alive. Your insurance may depend on a government's statement eight months later.*

These rows in the Trust Ledger

The worked example is Larkfield Software's desktop app: about 400,000 customer desktops on Windows, macOS and Linux, built on an Electron-style runtime, updated through Larkfield's own channel and signed with Larkfield's own code-signing key. The optional column "who pays on failure" is filled; the figures in it are the model's, for an invented company.

Row and component	Who can change it	Exposure and blast radius	Patch clock	Stop, revoke, recover	Who can see it	Support end	Decision and owner	Who pays on failure
LR: bundled Chromium-based runtime, pinned per release	Upstream writes; runtime project builds; Larkfield ships	Every document and web page the app renders; all 400,000 desktops	T1 = Larkfield release after the runtime fix; T3 measured from update telemetry	Update channel pause; rollback to previous signed build	D (vendor advisories); O (Larkfield update telemetry)	Runtime support per major release (Book 2, Chapter 11)	**Own** the upgrade train; **Pin** one major with an end date; owner: head of desktop engineering	Larkfield, toward customers; nobody toward Larkfield. From 11 Sept 2026, Larkfield also carries the CRA reporting duty
LR: open-source packages from a public registry, about 1,100 in the lock file	Hundreds of maintainers write and publish; the registry hosts	Build machines and every shipped build	np for most maintainers; Larkfield's own clock from advisory to rebuild	Lock files, a private mirror, rebuild from a known-good lock	I (scanner output); U for unmaintained packages	np for most packages	**Rent**; **Isolate** builds from live registry; fund the five packages the product cannot run without	Larkfield; no contract with any maintainer
ID: code-signing key for Windows and macOS builds	Two release engineers can sign; a hardware security module holds the key	A stolen key can sign malware that every customer's OS will trust as Larkfield	Revocation by the certificate issuer; clock np until drilled	Revoke and re-issue; re-sign current builds; last drill: none	O (signing logs in the HSM)	Certificate expiry date	**Own**; owner: CISO; drill revocation this quarter	Larkfield, and every customer that trusted the signature
MU: Larkfield's update channel and server	Release engineering pushes; the cloud host runs it	One approved push reaches every desktop that checks in	Larkfield's own: hours from build to 50% of desktops	Rings: 1%, 10%, 50%, 100%; kill switch on the server	O (check-in telemetry)	n/a	**Own**; rings and kill switch tested monthly	Larkfield, through SLA credits (Chapter 14); for products placed on the market after 9 Dec 2026, possible EU product liability, subject to defect, covered damage (mainly non-professional users' property and data) and causation

SECURITY LENS

Who can change it? In the money view, whoever is paid to look. A vendor with a large bounty and a staffed response team hears about flaws in its rows from people it pays; a component with no owner hears about them from an advisory, or from an incident. Brokers pay others to make sure some flaws reach a buyer instead of the owner.

How fast can it be fixed? Funding sets capacity, not speed. A rich vendor can still ship slowly; an unfunded project can still fix in hours. What funding changes is whether there is anyone on duty when the report arrives, and whether the fix reaches downstream copies. Measure the clock per row (Appendix D); do not infer it from a budget.

What does AI change here? Our view (as of November 2026): AI-assisted tools lower the cost of finding some classes of bugs for both sides, which puts pressure on bounty budgets and on unfunded maintainers at the same time. Chapter 16 has the dated evidence, including one project that closed its bounty over AI-generated reports.

BUSINESS IMPLICATIONS

Security. For Larkfield Software, the vendor side of trust: its signing key and update channel are now rows other companies rent, and the EU makes it report actively exploited flaws in its product within 24 hours from 11 September 2026. Halbrook Bank's EDR and identity vendors are the authorities on its riskiest rows; their bounty scope and advisory practice belong in the vendor review.

Cost. Corvane Hotels' cyber policy renews each spring. Before renewal, its risk team maps the state-backed exclusion and the "unpatched vulnerability" clause to the kiosk Ledger's MU, DR and FW rows, and asks the broker in writing what attribution standard the policy uses.

Visibility. Advertised broker asks and bounty maxima are the most visible public price signals in this market, and both are published by interested parties. Use them as dated direction, never as a valuation of your exposure.

Value. Liability law is beginning to price defects into the product. A vendor that can show its patch clock, its support period and its reporting process is cheaper to rent, because it is less likely to cause the loss in the first place.

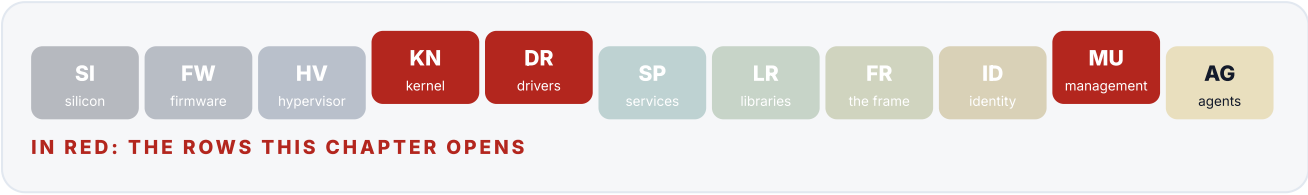
Monday rule: add "who pays on failure" to the five riskiest rows of one critical Ledger, with the contract clause or policy section that says so; where the answer is "nobody", write that down.

OUR VIEW, AS OF NOVEMBER 2026

The EU's two acts will move cost toward software makers, but slowly and unevenly: the CRA's main duties apply from December 2027, and the new liability rules apply only to products placed on the market after 9 December 2026. In the United States, absent legislation, liability will keep being set by contracts and by litigation such as the cases in Chapter 13. Expect vendors to answer with longer published support periods and more formal reporting, and to resist uncapped liability in contracts; Chapter 15 shows which clauses to ask for anyway.

CHAPTER THIRTEEN

The Record: Spyware, Sabotage and the Outage



On 15 February 2018 the British government said that the Russian military was "almost certainly responsible" for the NotPetya attack of June 2017, and hours later the White House said the same. The attack had reached the world through the update server of a Ukrainian accounting program. Seven years after the attack, on 19 July 2024, a security vendor's routine content update stopped Windows machines across airlines, hospitals and banks, with no attacker anywhere in the story. Put the two events side by side on the Trust Ledger and they share one pattern: a trusted update channel, privileged code on every endpoint that took the push, and machines that would not start. This chapter tells what the stack below the screen has actually been used for, by whom, according to the record. The decision it changes: Halbrook Bank stops writing separate plans for "a cyberattack" and "a vendor outage" on its update and driver rows, and writes one.

The rule for this chapter

This is the book's most sensitive material, so it is held to the narrowest sources. Technical facts come from the published analyses of vendors and researchers, cited by name. Attackers, and anyone accused of an attack, are named only as a court, a government or Citizen Lab names them, or as a vendor names itself in its own advisory. Damage figures carry the name of whoever said them. Where the record names nobody, this book names nobody. Each case told in full here is told only here; the technical mechanisms live elsewhere and get one sentence and a pointer: the Channel File 291 mechanism in Chapter 5, the zero-click chains in Chapter 10, the image decoders behind them in Book 2, Chapter 14, and the build-system compromises (SolarWinds in 2020, xz in 2024) in Chapter 8.

Four cases follow: Stuxnet, sabotage of industrial controllers through signed Windows drivers; Pegasus and the mercenary spyware market, documented by researchers, a government and a US court; NotPetya, a destructive attack through a trusted update

channel; and Channel File 291, the outage that used the same kind of channel by accident.

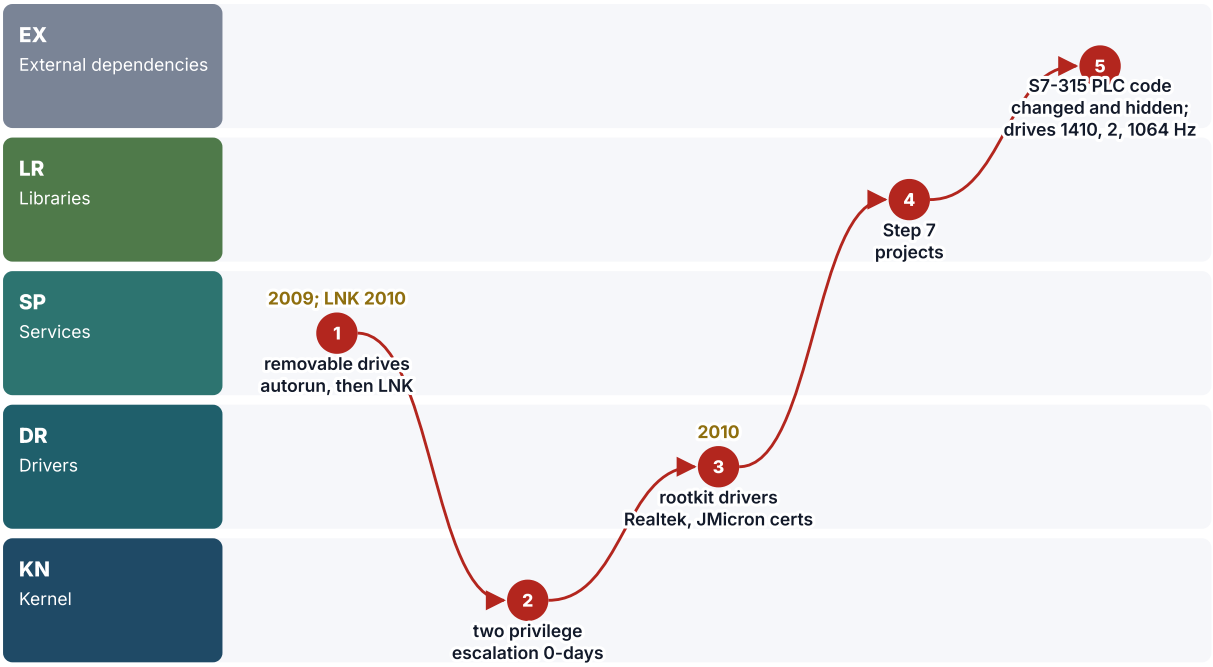
Stuxnet: sabotage through signed drivers

On 17 June 2010 the antivirus company VirusBlokAda reported a worm that spread through a flaw in the way Windows handled shortcut (`.lnk`) files. Symantec's analysis, published as the W32.Stuxnet Dossier (version 1.4, February 2011), is the detailed vendor account this section relies on, and it is the source for everything below.

- **Spread.** Stuxnet copied itself through removable drives, local networks and Siemens Step 7 project files, and used four unpatched Windows vulnerabilities, two to spread and two to raise privilege. The earliest sample the dossier records dates from June 2009 and spread through autorun files; the shortcut flaw appears in the variant of March 2010. Microsoft fixed the shortcut flaw out of band on 2 August 2010 (MS10-046) and the print spooler flaw Stuxnet used to spread on 14 September 2010 (MS10-061).
- **Trust.** Its kernel drivers were signed with a valid certificate issued to Realtek Semiconductor, whose private key the attackers held, used from 25 January 2010. Verisign revoked that certificate on 16 July 2010; the next day a new driver appeared, signed with a certificate from JMicron Technology, which Verisign revoked on 22 July. Chapter 5 uses these drivers as the signing precedent: a signature proves who signed, not what the code does.
- **Target.** Stuxnet carried attack code for Siemens S7-315 and S7-417 controllers (the dossier found the S7-417 code incomplete); the S7-315 sequence looked for frequency converters from two named manufacturers, then changed the drive frequency, in one sequence from 1410 Hz to 2 Hz to 1064 Hz, against a normal range the dossier gives as 807 to 1210 Hz. It hid the changed code from the operators, which the dossier calls "the first ever PLC rootkit".
- **Reach.** By 29 September 2010 Symantec counted about 100,000 infected hosts, about 60% of them in Iran, and judged that the target was an industrial control system "likely in Iran".

FIGURE 13.1

Stuxnet: from removable drives to the frequency drives



Numbered steps in order. Red: the path the attack or failure took.

Stuxnet crossed every trusted layer between a Windows engineering station and a motor, and drivers signed with two companies' compromised certificates let its kernel code load as trusted.

Symantec, W32.Stuxnet Dossier, version 1.4 (February 2011).

The dossier names no author, and no court or government record used in this book does either, so this book names none. What the record does establish is enough for a CTO. Stuxnet crossed every trust boundary between an office PC and a motor, and at the kernel step it did not need to break signing: it carried stolen keys. The observed interval from the first known signed driver to revocation was about six months, and the attackers had a second key ready the day after the first was revoked.

Pegasus and the mercenary spyware market: the record

"Mercenary spyware" is the term Apple and researchers use for surveillance tools that companies build and sell to governments. The public record on one vendor, NSO Group, and its Pegasus product is the most complete of any, because it comes from three independent kinds of source: forensic research by Citizen Lab at the University of Toronto, actions by the US government, and two civil lawsuits in a US federal court.

The research record. Citizen Lab's reports set the timeline. On 24 August 2016 it described a one-click chain against an iPhone, found after a targeted human rights defender sent it the suspicious links he had received; Citizen Lab had reported it to Apple on 15 August, and Apple shipped iOS 9.3.5 on 25 August. On 18 September 2018 Citizen Lab reported Pegasus infrastructure in operations it traced to 45 countries, noting that the list could include countries that were not NSO customers. On 13 September 2021 it published the zero-click chain it called FORCEDENTRY, six days after sending Apple the sample; Apple fixed it the same day in iOS 14.8. On 7 September 2023 it published BLASTPASS, a zero-click chain that delivered Pegasus to a staff member of a civil society group in Washington, DC. Citizen Lab said it disclosed the chain to Apple immediately on finding it the week before, and Apple fixed it in iOS 16.6.1 on the day of publication. Citizen Lab said, and Apple confirmed to it, that Lockdown Mode blocked that attack. How these chains worked below the screen is Chapter 10's subject.

The government record. On 4 November 2021 the US Department of Commerce added NSO Group and Candiru to the Entity List, citing information that they "developed and supplied spyware to foreign governments" that used it to target, among others, government officials, journalists and businesspeople (Federal Register, 86 FR 60759). The listing restricts US exports to them; it is not a criminal finding. On 27 March 2023 Executive Order 14093 barred US federal agencies from operational use of commercial spyware that poses counterintelligence or security risks to the United States. The market did not stop with one vendor: on 5 March 2024 the US Treasury sanctioned members of the Intellexa Consortium, which it described as developing and selling the Predator spyware.

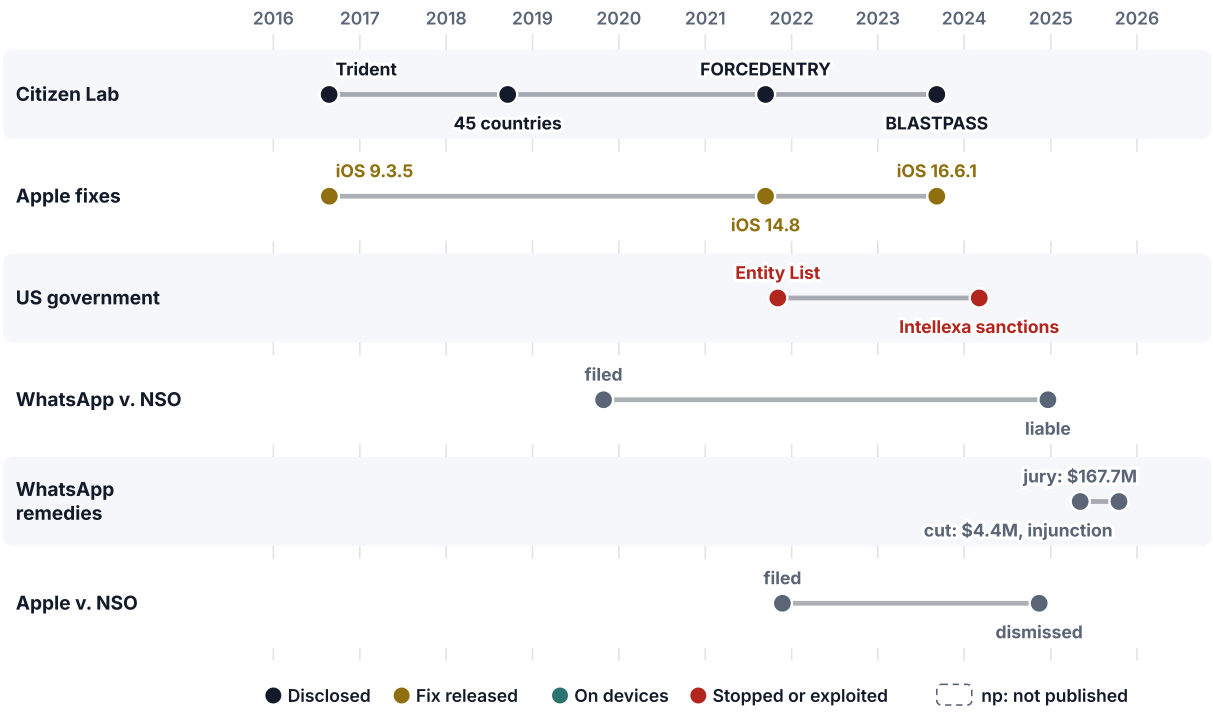
The court record. WhatsApp and its parent sued NSO Group on 29 October 2019 in the Northern District of California (No. 4:19-cv-07123), alleging that NSO used WhatsApp's servers to send malware to about 1,400 devices in 2019. The case gives a complete civil record, from liability to damages, against a spyware vendor in a US court:

- 20 December 2024: Judge Phyllis Hamilton granted summary judgment that NSO was liable under the federal Computer Fraud and Abuse Act and California's computer fraud law, and for breach of WhatsApp's terms, leaving only damages for trial.
- 6 May 2025: a jury awarded $444,719 in compensatory damages and $167,254,000 in punitive damages.
- 17 October 2025: the judge granted a permanent injunction barring NSO from breaking into or attempting to break into WhatsApp's systems, and cut the punitive award to about $4 million, nine times the compensatory award, for a total of about $4.4 million.
- NSO appealed to the Ninth Circuit (No. 25-7380), which denied its request to stay the injunction on 28 January 2026; opening and answering briefs were filed in February and March 2026, and the appeal was pending at this book's source freeze.

Apple sued NSO on 23 November 2021 (No. 3:21-cv-09078). In September 2024 Apple asked the court to dismiss its own case, telling the court that discovery could put vital security information at risk and that the spyware industry had changed; the court dismissed it without prejudice in November 2024.

FIGURE 13.2

The Pegasus record, 2016 to 2026: reports, fixes, sanctions, suits and verdicts



Apple's fixes came within days of Citizen Lab's private reports and on the day of each public report; governments acted years later; the jury award came five and a half years after the suit was filed and was cut by the judge five months later. The 2026 appeal is in the text.

Citizen Lab reports (2016 to 2023); Apple security notes; Federal Register 86 FR 60759 (4 Nov 2021); US Treasury (5 Mar 2024); N.D. Cal. 4:19-cv-07123 and 3:21-cv-09078; 9th Cir. 25-7380.

Read Figure 13.2 for its clocks, not its drama. From Citizen Lab's private report to Apple's fix took about ten days in 2016 and six days in 2021; the public report and the fix came out on the same day, because the two were coordinated. When a credible report reached the platform owner, the fix followed within days. The government clock was years. The court clock, from filing to a verdict on damages, was five and a half years, and it is still running on appeal. For a company whose executives, lawyers or engineers could be targets, the only clock it controls is the first one: how fast its own devices take the platform's fix, and whether its highest-risk users run the platform's protective mode.

NotPetya: the full case

27 June 2017. Machines in Ukraine and then around the world began to stop. The US-CERT alert of 1 July 2017 (TA17-181A) set out the path. The delivery mechanism was M.E.Doc, a Ukrainian tax accounting program, and the alert says the attackers "used a backdoor to compromise M.E. Doc's development environment as far back as April 14, 2017": the program's own update mechanism carried the payload to customers. Once inside a network, NotPetya stole Windows credentials with a modified version of the Mimikatz tool, used them to run itself on other machines through PsExec and WMI, and tried the SMBv1 exploits EternalBlue and EternalRomance against unpatched systems; Microsoft had fixed those flaws in MS17-010 on 14 March 2017, 105 days earlier. With administrator rights it modified the master boot record "to enable encryption of the master file table (MFT)", the index of every file on the disk, making machines unusable. The alert said it "behaves more like destructive malware rather than ransomware": there was no evidence that the key could be recovered, so paying would not bring the data back.

Attribution. On 15 February 2018 the UK Foreign Office minister said the government judged the Russian government responsible, based on the National Cyber Security Centre's assessment that the Russian military was "almost certainly responsible". The same day, the White House press secretary's statement said that "the Russian military launched the most destructive and costly cyber-attack in history" in June 2017; the superlative is the White House's. On 19 October 2020 the US Department of Justice announced an indictment, returned by a grand jury in the Western District of Pennsylvania, of six officers of Unit 74455 of the Russian military intelligence directorate (GRU), covering NotPetya among other attacks. The indictment is an accusation, not a conviction.

The bill, with its speakers.

- The Justice Department's announcement said the defendants' malware attacks caused nearly one billion US dollars in losses to the three victims named in the indictment alone: hospitals and medical facilities of the Heritage Valley Health System in Pennsylvania, FedEx's subsidiary TNT Express, and a large US pharmaceutical manufacturer.
- Wired (Andy Greenberg, 22 August 2018) reported "more than $10 billion in total damages", according to a White House assessment confirmed to Wired by Tom Bossert, then the President's homeland security adviser. That figure is an estimate by the government that attributed the attack.
- The same Wired article reported company figures: Maersk's chairman put its cost at $250 million to $300 million; Merck told shareholders it lost $870 million; FedEx took a $400 million blow; Mondelez a $188 million hit. Each is the company's own number.

The insurance aftermath, in which Merck's and Mondelez's insurers invoked war exclusions, is told in Chapter 12.

Channel File 291: the outage as record

The mechanism is Chapter 5's: on 19 July 2024 at 04:09 UTC CrowdStrike released a content configuration update for its Windows sensor that triggered an out-of-bounds memory read in the sensor and crashed Windows; it reverted the file at 05:27 UTC, but machines that had loaded it kept crashing on boot. Microsoft estimated that about 8.5 million Windows devices were affected, "less than one percent of all Windows machines". This section is the record of what followed.

No attacker. CrowdStrike's senior vice president for counter adversary operations, Adam Meyers, told a House Homeland Security subcommittee on 24 September 2024 that "we let our customers down", and that the incident was not a cyberattack. CrowdStrike's own review says the file reached Windows hosts running sensor version 7.11 or later that were online between 04:09 and 05:27 UTC. Its preliminary review had committed to staggered deployment of this content, starting with a canary group, and to giving customers more control over when it arrives.

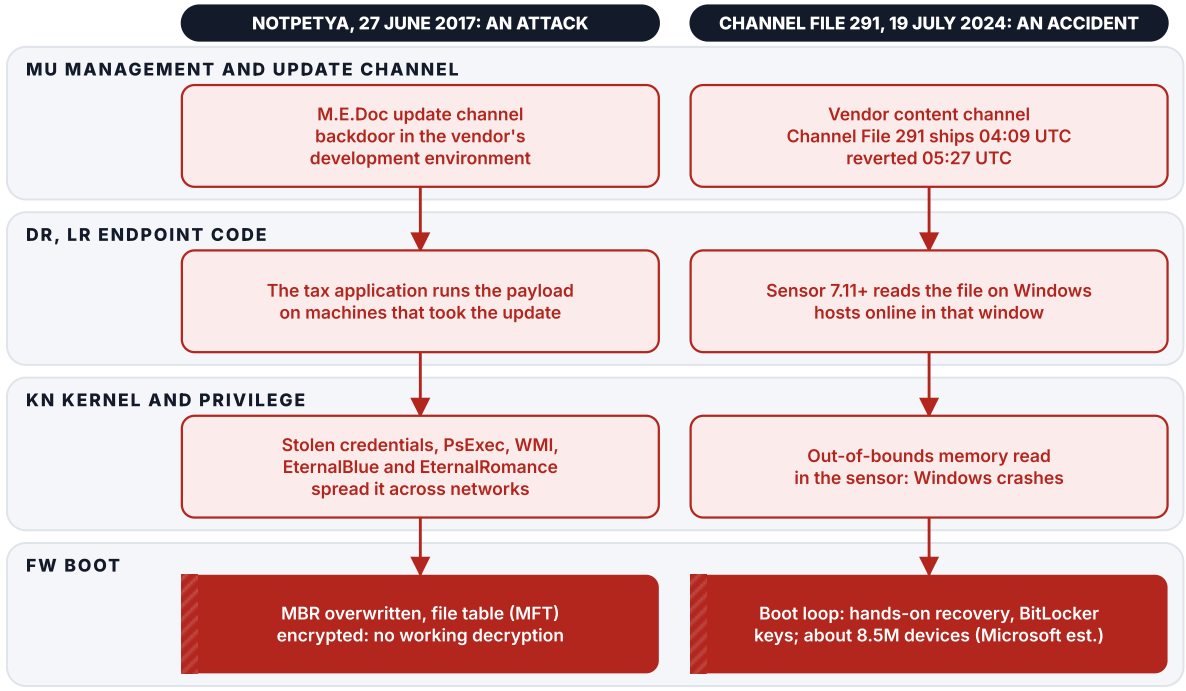
The suits. The largest named claim came from Delta Air Lines. Delta's own SEC filing in August 2024 estimated a revenue hit of about $380 million and about $170 million in costs from the disruption, which cancelled about 7,000 flights. CrowdStrike's outside counsel wrote to Delta that month that any liability was contractually capped at an amount in the single-digit millions. Delta sued CrowdStrike in the Superior Court of Fulton County, Georgia, on 25 October 2024; the case later moved to the court's Metro Atlanta Business Case Division. On 16 May 2025 the court granted CrowdStrike's motion to dismiss in part and denied it in part; reporting on the order says claims including gross negligence and computer trespass went forward. In its quarterly report for the period ended 30 April 2026, CrowdStrike said discovery was ongoing and that it could not estimate the amount of any loss. Other suits ended differently: passenger class actions against CrowdStrike were dismissed on 18 June 2025, and the Fifth Circuit affirmed on 20 May 2026. The same quarterly report still lists a putative securities class action against the company.

What Channel File 291 adds to the record is a fact no attack could prove so cleanly: a trusted channel with kernel reach can produce the effects of a destructive attack with no intent at all. The cost figures and the concentration question belong to Chapter 14.

The overlay: same rows, different intent

FIGURE 13.3

NotPetya and Channel File 291: one pattern on the Trust Ledger rows



An attack in 2017 and an accident in 2024 shared a pattern: one trusted update channel, code that ran on every endpoint that took it, and machines that would not start. NotPetya also spread across networks; Channel File 291 did not.

US-CERT alert TA17-181A (1 Jul 2017, rev. 15 Feb 2018); CrowdStrike PIR (24 Jul 2024) and RCA (6 Aug 2024); Microsoft (20 Jul 2024).

Figure 13.3 draws NotPetya and Channel File 291 on the same Ledger rows. Both began on the MU row, in an update channel the customer had chosen to trust. Both reached code that runs on every endpoint that checked in. Both ended with privileged code leaving machines that would not start, one by design and one by accident. The paths in between differ: NotPetya then stole credentials and moved across networks, which Channel File 291 never did. The difference that matters most sits in two cells: intent, and recoverability. With no evidence that NotPetya's key could be recovered, its victims' machines came back by reimaging and restoring from backup; Channel File 291 could be fixed on each machine, but often only by hands on the keyboard and a BitLocker recovery key.

How each was discovered:

- **Stuxnet:** by an antivirus vendor that reported a sample in June 2010, then by months of public analysis.
- **Pegasus:** by a targeted person who forwarded suspicious links to researchers in 2016, and by forensic work on phones afterwards.
- **NotPetya:** by its own damage, within hours, and then by incident responders tracing it to the M.E.Doc update.
- **Channel File 291:** by the vendor itself, which reverted the file 78 minutes after release; the damage was already on the machines that had loaded it.

Record point	Windows	macOS	iOS	Linux	Android
Cases in this chapter	Stuxnet (2010), NotPetya (2017), Channel File 291 (2024)	None of the four cases here	Pegasus chains in Citizen Lab reports, 2016 to 2023	None of the four cases here	None of the four cases here
Who could push to a large share of the fleet in one action	Microsoft, the EDR vendor's content channel, any app updater running with admin rights (the M.E.Doc pattern), the management console (Chapter 7)	Apple, the MDM, app updaters	Apple; the MDM on managed devices	Distribution repositories, configuration management, security agents	Google services, the device maker, the EMM
Built-in mode for targeted users	No equivalent named in this record	Lockdown Mode	Lockdown Mode (blocked BLASTPASS, per Citizen Lab and Apple)	No equivalent named in this record	Advanced Protection (Android 16, announced May 2025); device support varies (Chapter 10)
Firmware and silicon	None of the four cases changed firmware on the PC. NotPetya overwrote the boot record and encrypted the file table on disk; Channel File 291 left machines in a boot loop; Stuxnet changed the code on industrial controllers, which are computers of their own outside the Ledger's PC rows (the EX row).				

HOW IT GOT THIS WAY

Update channels were built to solve the problem of the 2000s: machines that never got patched. Automatic updates, vendor-run content feeds and admin-rights updaters turned patching from a project into a background service, and that was the right call; most of this book's patch clocks depend on it. The cost was concentration of trust. The more devices a channel reaches and the more privilege its payload gets, the more a single push is worth to an attacker, or costs in an accident. M.E.Doc in 2017 and Channel File 291 in 2024 sit at the two ends of that trade, seven years apart.

What hasn't changed: the four cases in this record used trusted paths, not new kinds of magic: stolen signing keys (2010), a vendor's update server (2017), a platform's own message and image handling (2016 to 2023), and a vendor's content channel (2024). Checked against the sources above on 18 November 2026.

THE UNSEEN ANGLE

An outage and an attack share one Trust Ledger, and most companies keep two plans for them: a security incident plan owned by the CISO and a business continuity plan owned by operations. On the MU and DR rows those plans describe the same failure. Whether the push came from an attacker or a bad release, the first hour asks the same questions: which channel, which devices took it, how do we stop the next push, and how do we restore machines that will not boot. The difference arrives later, in the notification clocks (Chapter 14) and in the insurance file (Chapter 12), where attribution can decide what is covered. Sources: US-CERT TA17-181A (2017); CrowdStrike preliminary post incident review (24 July 2024).

> *Each case in this record ran through a path its victims trusted. In 2017 and in 2024, one push through that path was enough to stop machines.*

These rows in the Trust Ledger

Halbrook Bank's branch desktop, with the DR and MU rows replayed for both events: what each cell would have said on 26 June 2017 for an M.E.Doc-style updater, and on 18 July 2024 for an EDR content channel. The entries are the book's model for an invented bank. As in the rest of Part V, the sensor's own code is a DR row and the content channel that feeds it is an MU row.

Row and component	Who can change it	Exposure and blast radius	Patch clock	Stop, revoke, recover	Who can see it	Support end	Decision and owner
DR: EDR sensor, kernel mode, 2,500 branch PCs and VDI hosts	EDR vendor writes and signs; vendor console pushes sensor versions on Halbrook's rings	Kernel; one sensor release reaches the ring it is assigned to	Sensor versions: Halbrook's rings, T3 in 7 days	Pin sensor version N-1; rollback tested	D (vendor console); O (Halbrook crash telemetry)	Per contract	**Rent**; **Pin** sensor N-1 with an end date; owner: endpoint lead
MU: EDR content channel (the 2024 row)	EDR vendor pushes content; Halbrook had no veto on 18 July 2024	One push reached every connected sensor at once	Minutes; the vendor's clock, not Halbrook's	No customer pause in 2024; recovery by hand with BitLocker keys; drill: none	U before the push; O after (machines down)	n/a	**Rent**; **Isolate**: contract for staged content and a customer pause; owner: CISO
MU: third-party app updater running as admin (the 2017 row)	The app vendor's build and update server; anyone who compromises them	Every PC with the app; admin rights; network spread after that	The app vendor's own; np	Block the updater at the firewall; no rollback	U (no inventory of admin updaters in 2017)	np	**Isolate**: no admin-rights updaters outside Halbrook's own MU channel; owner: endpoint lead
KN and FW: recovery path for a machine that will not boot	Halbrook's service desk, with BitLocker recovery keys from the directory	All branch PCs in one event	Recovery clock, not patch clock: target 90% of branch PCs in 8 hours	Key escrow, boot media at each branch region; last drill date: to set	O (key retrieval logs)	n/a	**Own**; owner: head of end-user computing; drill twice a year

SECURITY LENS

Who can change it? In every case here, someone the victim trusted, or someone holding that party's keys: a hardware maker's signing certificate, an accounting vendor's update server, a messaging platform's servers, a security vendor's content channel.

How fast can it be fixed? In the documented Pegasus chains, the platform fix came six to ten days after Citizen Lab's private report. Microsoft's Stuxnet fixes came about seven and thirteen weeks after the worm was first reported, and MS17-010 was 105 days old when NotPetya struck. For NotPetya and Channel File 291, the fix was not the problem; recovery was.

What does AI change here? Our view (as of November 2026): nothing in this record depended on AI. What AI may change is the cost of the reconnaissance and the reverse engineering that each case required, on both sides; Chapter 16 sets out what the published evidence supports.

BUSINESS IMPLICATIONS

Security. Halbrook Bank writes one runbook for "a push we did not approve reached the fleet", used for both attack and outage, with the stop and recovery steps first and the attribution question last.

Cost. No "biggest bill" belongs in a board paper. Corvane Hotels' risk report cites each NotPetya and Channel File 291 figure with its speaker (the Justice Department, the White House via Wired, Delta, Microsoft), because each is an estimate or a claim made by an interested party.

Visibility. Larkfield Software is, for its 400,000 customers, what M.E.Doc was in 2017: a vendor with an update channel. It publishes how its channel is protected and how a customer can pause it, before a customer asks.

Value. Trust lost lasts longer than the outage. The court records here run for years after the event; the customers' memory of the day runs longer.

Monday rule: list every channel that can push code or content into the kernel or with admin rights on your fleet, and for each one write who can stop the next push and how long it takes.

CHAPTER FOURTEEN

The Cost of Trust: Control, Concentration and a Day Down



At Halbrook Bank's March board meeting the audit committee asks one question about the Trust Ledger: "What does a day without these rows cost us, and how much of that day can one vendor take away?" The CISO has the security budget by tool. The CFO has revenue by channel. Nobody has the two on one page. This chapter builds that page. It prices each row twice, once for what it costs to control and once for what a day without it costs, and it adds a number few board packs show: the share of revenue one authority can interrupt with one approved action. The decision it changes is where the next dollar goes: to the control that removes the most priced loss per dollar, not to the row with the loudest vendor.

Two price tags per row

Every Ledger row has a **cost of control**: what you pay, every year, to hold the decision you wrote in its last column. It also has a **cost of failure**: what one bad day on that row costs, whoever caused it. Most budgets show the first and guess the second.

The cost of control has five parts, and only the first shows up as a line item:

- **Licenses and subscriptions:** the EDR, the identity platform, the management console, extended support.
- **People:** the hours to run rings, read telemetry, test recovery and answer the auditor.
- **Performance:** the CPU, memory and battery a security agent takes from every device, paid in hardware refresh.
- **Compatibility:** the apps that break when a control is turned on, and the exceptions that follow.
- **Delay:** the hours a ring holds a fix back from 90% of the fleet. This is the cheapest control on most rows and the one most often priced at zero.

The table below is the model for Halbrook Bank. The figures are invented for an invented bank; the structure is the point.

Row and control	Yearly cost of control (model)	What it buys	What it costs in time
ID: phishing-resistant sign-in for staff, help-desk reset with callback and manager approval	$420,000 (keys, platform tier, 1.5 people)	Closes the "log in instead of break in" path of Chapter 6	About 20 minutes more per reset; resets fall after the first quarter
DR: EDR on 2,500 branch PCs and VDI hosts, sensor pinned at N-1	$310,000 licenses; 0.5 person	Detection and response on the riskiest endpoints	Sensor fixes reach the fleet one release later
MU: three Intune rings (5%, 25%, all) for OS and app updates	$60,000 (0.4 person for ring health)	A bad update stops at 5% of devices, if telemetry catches it before the next ring	Up to 72 hours added to T3 for a normal fix; hours for an emergency fix
FW: firmware updates by model, Secure Boot certificate status tracked	$45,000 (0.3 person)	Boot-level protections keep arriving (Chapter 4)	A firmware ring per model, two weeks each
Visibility: device health attestation, crash telemetry, 13 months of sign-in and console logs	$180,000 (storage and tooling)	The O tags in the "who can see it" column	None at run time; analyst hours when used

Read the last column first. The MU rings cost $60,000 a year and up to three days of delay on routine fixes. When telemetry catches a bad update before the next ring, they turn a fleet-wide failure into a 5% failure. In this model the ring buys more risk reduction per dollar than any other line, and it is the control most easily removed when a patch-compliance metric looks bad.

A day down, priced honestly

The cost of failure starts with one distinction that most estimates skip: revenue that is **deferred** and revenue that is **lost**. A payment that waits a day still arrives; a guest who walks to the hotel next door does not come back to check in. A loss estimate counts revenue that is lost and money actually paid out, including fines, legal fees and regulatory

remediation when they arise; deferred revenue stays out, except for what the delay itself costs (financing, penalties), which counts as paid. Figure 14.1 applies that to one day for each of the book's three invented companies.

FIGURE 14.1

One day down, three companies: what is deferred, what is lost, what recovery costs

Line (first day)	Halbrook Bank (2,500 branch PCs)	Corvane Hotels (3,000 kiosks, 300 properties)	Larkfield Software (400,000 customer desktops)
Revenue at stake that day	$2.7M (branch channel: 38% of $1.8B over 252 days)	EUR 6.6M (1/365 of EUR 2.4B)	$0.95M (1/252 of $240M)
Deferred (comes back later)	about $2.3M (85%): payments and loan work done in the next days	none: guests still stay and pay (room revenue not affected)	most of it: subscriptions already paid
Permanently lost	about $0.4M (15%): waived fees, lost applications	about EUR 0.07M a day: walk-ins, late check-outs	np until renewal: churn shows months later
Recovery labor (one-time)	$0.16M: 2,500 PCs x 45 min x $85 an hour	EUR 0.14M: 300 field visits x EUR 450, about EUR 0.045M a day over 3 days	$0.30M: 12,000 tickets x $25
Overtime, contractors, travel	$0.25M	EUR 0.12M a day front-desk overtime	$0.05M
Customer remediation	$0.30M goodwill and fee waivers	EUR 0.05M a day vouchers	$1.0M SLA credits (5% of a month)
Loss estimate (lost + paid)	about $1.1M	about EUR 0.29M a day while visits run; about EUR 0.86M for 3 days	about $1.35M, plus churn not yet known
What decides the number	hours until branches open with working PCs	field visits a day the team can run	whether Larkfield's channel has a pause and rings

The same kind of event costs each company differently: the bank mostly defers branch revenue, the hotel group mostly pays for hands and travel, the software vendor pays in credits, support and trust. The loss estimate counts the lost and paid lines; what a delay itself costs counts as paid.

Author's model for the book's invented companies; assumptions stated in Chapter 14. Not data about any real company.

The model's assumptions, so you can replace them with your own:

- **Halbrook Bank:** $1.8 billion annual revenue over 252 business days, 38% of it through the branch channel; all 2,500 branch PCs down for one business day; 85% of that day's branch revenue deferred, the rest lost to waived fees and abandoned applications; 45 minutes of hands-on work per PC at $85 an hour fully loaded.
- **Corvane Hotels:** EUR 2.4 billion over 365 days; all 3,000 check-in kiosks down; rooms still sold, so almost nothing deferred or lost from room revenue; the cost is front-desk overtime each day and one field visit per property at EUR 450, paid once per property, about 100 visits a day for three days.
- **Larkfield Software:** $240 million a year in subscriptions; a bad release through its own channel crashes the app on desktops for a day; a support ticket rate of 3% at $25 a ticket; SLA credits of 5% of a month's fees; churn unknown until renewals.

Three lessons fall out of the model, and they hold well beyond it. First, for Halbrook the branch revenue at stake that day ($2.7 million) is not the loss (about $1.1 million); reporting the first as the second overstates the case and loses the board's trust the first time someone checks. Second, for Corvane the loss per day is small and the days are many: what decides the cost is recovery throughput, how many properties a team can visit in a day, not the revenue line. Third, for Larkfield the largest line is a contract it wrote (the SLA credit), and the most important one is the one nobody can fill yet: customers who leave at renewal.

Published losses from real events follow the same rule, and they must carry their speaker. Delta Air Lines estimated, in its own SEC filing of August 2024, about $380 million of lost revenue and about $170 million of costs from the July 2024 disruption, and in litigation it claims more from CrowdStrike; those are Delta's figures, contested in court (Chapter 13). The insurer analytics firm Parametrix estimated on 24 July 2024 that the outage cost US Fortune 500 companies, excluding Microsoft, $5.4 billion in direct losses, with insured losses of $540 million to $1.08 billion; those are a model's estimates, published days after the event. The CrowdStrike case itself is told in Chapter 5 (mechanism) and Chapter 13 (record).

Do not add the same loss twice

A Trust Ledger lists rows; failures do not respect them. When a kernel-mode sensor crashes Windows (DR), the identity sign-in on those machines fails too (ID), the teller app does not start (LR), and the management channel that could push a fix cannot reach a machine that will not boot (MU). If each row's owner writes "a day down: $1.1 million" in the row's cost of failure, the Ledger now says one event costs $4.4 million.

The fix is to price **scenarios**, not rows. Write each loss scenario once in the Ledger's header block ("one push stops every branch PC", "the identity provider is down for a business day", "a firmware update fails on one PC model"), price it once, and list the rows it crosses. Then rank rows by how many priced scenarios they appear in and how much each scenario costs. That ranking, not the sum of the rows, is what a budget should follow.

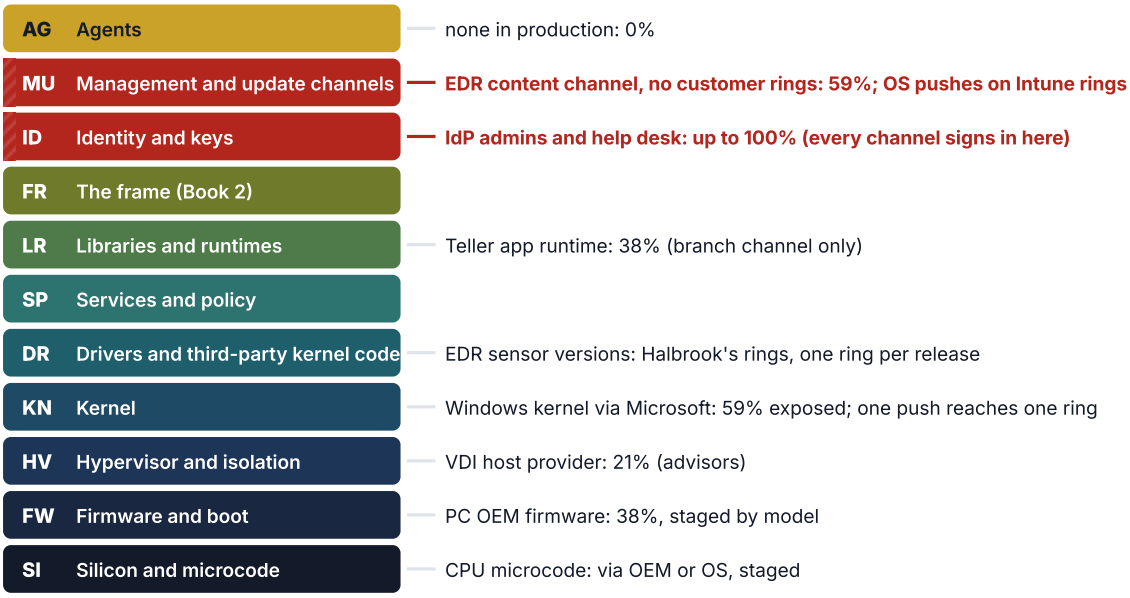
Correlation also works across companies. A failure in an authority that many firms rent arrives everywhere at once: your recovery contractors, your BitLocker key service, your airline and your bank's other branches are all busy on the same morning. Recovery estimates that assume outside help is available at normal rates on the worst day are optimistic by construction.

Concentration: one authority, one action, one share of revenue

The number this chapter asks every board to see is **concentration**: for each authority on the Ledger, the share of revenue it can interrupt with one approved action. "Approved" matters. The question is not what an attacker could do with that authority's keys (Chapter 6) but what the authority itself can do through its normal channel on a normal day: push a release, push content, change a policy, revoke a certificate.

FIGURE 14.2

Concentration at Halbrook Bank: the share of revenue one approved action can interrupt



Red: one approved action reaches more than half of revenue. Model figures for an invented company.

Two authorities can each stop more than half of revenue with one approved action; the others are bounded by rings, if telemetry catches the failure and promotion stops, or by the one channel they serve.

Author's model for the invented Halbrook Bank: branch and teller 38% of revenue, mobile 41%, advisors on VDI 21%. Shares are of revenue interrupted, not revenue lost.

On Halbrook's model, two authorities stand out. The identity provider and its administrators can stop sign-in for every channel, branch, mobile and advisor, with one policy change: up to 100% of revenue. The EDR vendor's content channel, without customer rings, reaches every branch PC and every VDI host at once: 59%. Everything else is bounded, either by Halbrook's own rings (the Windows update channel through Intune), provided telemetry catches a failure before the next ring, or by the single channel the authority serves (the VDI host provider, the teller app runtime).

Concentration has a partner number: **recovery throughput**, the devices or properties a team can bring back per hour when the authority's action has to be undone by hand. In Halbrook's model, keys sit in the directory and 165 people (branch staff guided by the service desk) each spend 45 minutes per PC: about 220 PCs an hour if nothing else queues, so 90% of the 2,500 branch PCs are back in about 10 hours and all of them in about 11. Those are lower bounds, not forecasts. They miss the 8-hour target in Chapter 13's Ledger; meeting it takes at least 211 people working without a pause, so more in practice, or a faster key path.

Corvane's throughput is set by field visits, about 100 properties a day. For scale from the record, Maersk's chairman said in January 2018 that the company reinstalled about 45,000 PCs and 4,000 servers in ten days after NotPetya. Concentration tells you how much one action can stop; throughput tells you how long it stays stopped. Divide the affected fleet by the throughput to get the hours down, price those hours with the scenario, and you have the number worth a contract negotiation.

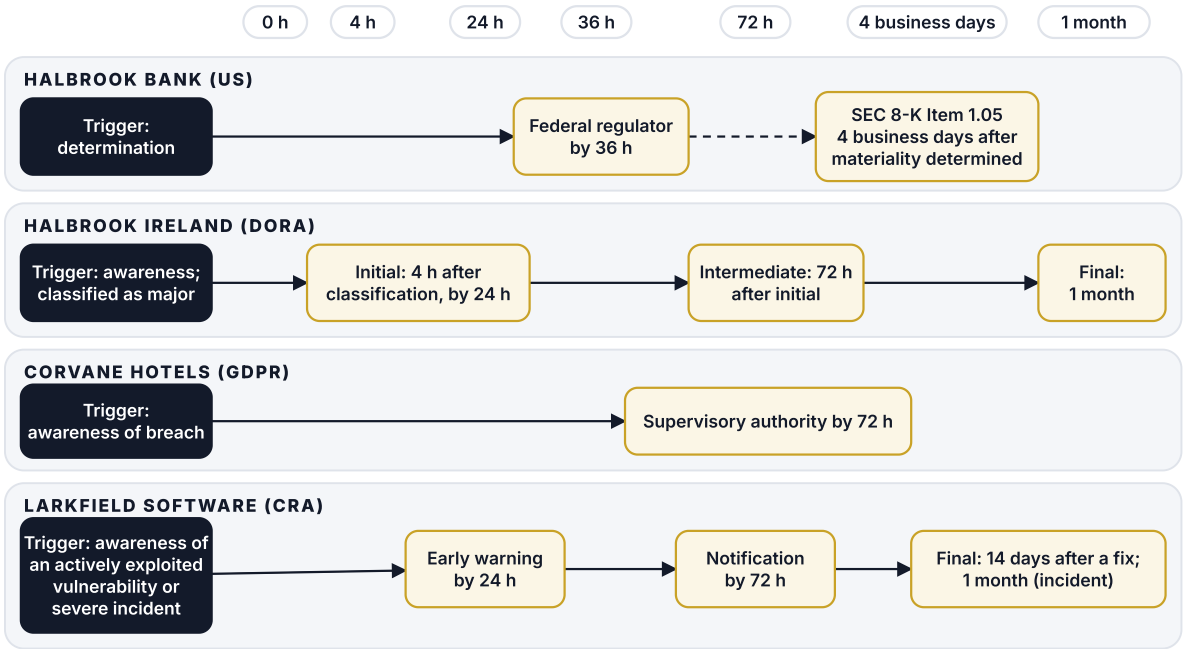
Two controls cut concentration without changing vendors: rings that the customer controls, and a pause the customer can press. Both are contract terms as much as technical settings, and Chapter 15 gives the clause text.

Clocks with triggers

A day down also starts regulatory clocks, and each one starts at a different event. The trigger matters more than the number of hours: a team that starts the right clock at the wrong moment will be late with the earliest notice. Figure 14.3 sets the clocks that apply to the book's three companies.

FIGURE 14.3

Regulatory clocks per company, and what starts each one



Each clock starts at a different trigger (awareness, a determination, a classification); a team that does not know which event starts which clock will miss the earliest one.

86 FR 66424 (2021) and OCC Bulletin 2021-55; SEC final rule (26 Jul 2023); Delegated Regulation (EU) 2025/301 Art. 5; Regulation (EU) 2016/679 Art. 33; Regulation (EU) 2024/2847 Art. 14. Scale not linear.

- **Halbrook Bank, US banking regulators.** Under the computer-security incident notification rule issued by the OCC, the Federal Reserve and the FDIC in November 2021 (in effect from April 2022), a bank must notify its primary federal regulator as soon as possible and no later than 36 hours after it determines that a notification incident has occurred. A bank service provider performing covered services must notify its bank customers as soon as possible when it determines that an incident has materially disrupted or degraded those services, or is reasonably likely to, for four or more hours; that clock applies to Halbrook's EDR or core-banking vendor only if it is such a provider.
- **Halbrook Bank, SEC.** Under the rules the SEC adopted on 26 July 2023, a listed company files a Form 8-K under Item 1.05 generally within four business days after it determines that a cybersecurity incident is material, and it must make that determination without unreasonable delay. The clock starts at the determination, not at discovery.
- **Halbrook's Irish subsidiary, DORA.** The Digital Operational Resilience Act, Regulation (EU) 2022/2554, applies from 17 January 2025. Under its reporting rules (Delegated Regulation (EU) 2025/301, Article 5), an initial notification of a major ICT-related incident is due within four hours of classifying it as major and no later than 24 hours after becoming aware of it; an intermediate report within 72 hours of the initial notification; a final report within one month of the latest intermediate report.
- **Corvane Hotels, GDPR.** Where a day down is also a personal data breach, Article 33 of Regulation (EU) 2016/679 requires notice to the supervisory authority without undue delay and, where feasible, within 72 hours of becoming aware of it. An outage with no data exposure starts no GDPR clock; an attack through the same channel may.
- **Larkfield Software, Cyber Resilience Act.** From 11 September 2026, Article 14 of Regulation (EU) 2024/2847 requires a manufacturer to send an early warning within 24 hours of becoming aware of an actively exploited vulnerability or a severe incident affecting its product, a fuller notification within 72 hours, and a final report within 14 days of a fix being available (for a vulnerability) or within a month (for an incident).

The same morning can start some of these clocks and not others. A vendor outage at Halbrook with no data loss may never become "material" or a "notification incident"; the same symptoms caused by an attacker may start more of them, depending on scope, data exposure and materiality. That is why the first-hour question in Chapter 13's runbook, "which channel, which devices", is also the question that decides which clocks are running.

Visibility is a cost line

The "who can see it" column of the Ledger is not free. Every O tag, a fact observed on your own fleet, is paid for in telemetry, storage and the people who read it. Every U tag, unknown, is paid for later, in the hours spent finding out during an incident. Three visibility costs

repay themselves on the rows this chapter prices:

- **Crash and boot telemetry** per device, so that the first report of a bad push comes from your own dashboard, not from the branch manager's phone call.
- **Console and policy logs** for every authority on the MU and ID rows, kept long enough for the slowest clock you must answer (DORA's final report comes a month or more after the event).
- **Recovery key retrieval logs**, which double as the measurement of recovery throughput: the rate at which keys were retrieved is the rate at which machines came back.

What decides recovery cost	Windows	macOS	iOS	Linux	Android
Machine will not start after a bad push	Recovery environment plus the BitLocker recovery key per device; Quick Machine Recovery on Windows 11 24H2 where enabled (off by default on Pro and Enterprise)	recoveryOS plus the FileVault recovery or institutional key	Restore from a computer or reset, then MDM re-enrollment	Rescue boot plus the disk encryption passphrase or escrowed key, from your own tooling	Factory reset or reflash, then EMM re-enrollment
What makes it a field visit	No remote console below the OS; keys not escrowed	No admin on site with a second Mac	Devices not supervised	No out-of-band console	Devices not enrolled in zero-touch or equivalent
Firmware and silicon	A failed firmware update can need the OEM's recovery procedure or a board swap on any platform; that is a field visit or a shipment, priced per device, and it belongs in the FW row's cost of failure (Chapter 4).				

HOW IT GOT THIS WAY

For most of the PC era, the cost of a day down was a private number. No regulator asked for it on a clock, and few contracts priced it. That changed in a short span. The US banking agencies' 36-hour rule was issued in November 2021 and took effect in April 2022; the SEC's four-business-day rule was adopted in July 2023 and took effect for most filers in December 2023; DORA was adopted in 2022 and applies from January 2025; the Cyber Resilience Act's reporting duty applies from September 2026. Each one forces a company to decide, fast and in writing, whether an incident crosses a threshold, and what it cost or might cost is part of that judgment. The Trust Ledger's header block, with its value at stake, cost of a day down and regulator clocks, exists because those decisions now have deadlines.

What hasn't changed: in the cases in this book, most of the cost of an incident was paid by the customer, not by the authority that caused it. In August 2024 CrowdStrike's counsel wrote that its liability to Delta was contractually capped in the single-digit millions; Delta's own filing put its revenue loss and costs at about $550 million, and Parametrix estimated $5.4 billion of direct losses for the Fortune 500 excluding Microsoft. Each figure is its speaker's. The gap between them is the point.

THE UNSEEN ANGLE

A ring is the cheapest control on most rows: a few hours or days of delay, priced at zero in most budgets because it shows up only as a slightly worse patch-compliance number. Put the delay on the same page as the concentration it removes. For Halbrook's model, three Intune rings cost $60,000 a year and up to 72 hours on routine fixes, and, if telemetry catches the failure before promotion, they cap a bad Windows update at 5% of devices instead of 59% of revenue. The same logic, applied to the EDR content channel, is the clause CrowdStrike's preliminary review described after July 2024: staged deployment and customer control over timing. Source: CrowdStrike preliminary post incident review (24 July 2024); the ring figures are the book's model.

> *The headline revenue of a lost day is not the loss. The loss is what never comes back, plus what you paid to get the rest back.*

These rows in the Trust Ledger

This chapter fills the header block, once per system, for all three companies. The money figures are the book's model for invented companies; the clocks are the real rules that would apply to them.

Header field	Halbrook Bank: branch teller desktop	Corvane Hotels: check-in kiosk	Larkfield Software: desktop app and its update channel
System and business service	Teller desktop for payments and account service in 2,500 branch positions	Self check-in and payment at 300 properties	The product itself, on about 400,000 customer desktops
Platform profile	Windows 11 Enterprise, managed by Intune, BitLocker on	Windows IoT Enterprise LTSC, kiosk mode, OEM firmware	Windows, macOS and Linux; Electron-style runtime; Larkfield's own updater
Fleet denominator	2,500 PCs plus VDI hosts	About 3,000 kiosks	About 400,000 desktops (customer-owned)
Value at stake; cost of a day down	38% of revenue through branches; about $1.1M lost and paid per day down	Check-in and payment; about EUR 0.29M a day, EUR 0.86M for a three-day event	All revenue at renewal; about $1.35M per bad-release day, plus churn
Concentration	IdP up to 100%; EDR content 59% (Figure 14.2)	EDR content and kiosk app pipeline: 100% of kiosks each	Larkfield's own channel: 100% of its customers' installs
Regulator clocks and their trigger	36 h after determining a notification incident; 8-K four business days after determining materiality; DORA for the Irish subsidiary from awareness and classification	GDPR 72 h from awareness of a personal data breach; PCI DSS duties for payment kiosks	CRA 24 h, 72 h, 14 days from awareness of an actively exploited vulnerability or severe incident
Loss scenario	One push stops every branch PC for a business day	A bad update stops kiosks at every property; fixes need a visit	A bad release crashes the app for a day; or the channel is used to push malware
Assessment date; next review	March; September	April; October	Every major release; at least quarterly

Accountable executive	Chief operating officer	Chief information officer	Chief technology officer

SECURITY LENS

Who can change it? Concentration is the money form of this question: an authority that can change many rows on many devices with one action is an authority whose bad day is your bad day, whatever the cause.

How fast can it be fixed? For cost, the clock that matters is often the recovery clock, not the patch clock. A fix that exists in minutes but needs a technician at each device costs as much as the devices take to reach.

What does AI change here? Our view (as of November 2026): AI can speed up the analyst work behind the first hour (which channel, which devices) and the drafting of notifications against the clocks. It does not speed up a field visit or a BitLocker key typed by hand, which is where most of the day-down cost in this model sits.

BUSINESS IMPLICATIONS

Security. Halbrook Bank adds one line to its board report: the two authorities that can each interrupt more than half of revenue with one approved action, and what has been done this quarter to bound each one.

Cost. Corvane Hotels prices its day down by field visits, not by revenue. Its cheapest saving is not a new tool but escrowed recovery keys and an out-of-band console on each property's kiosks, which turn a visit into a remote fix.

Visibility. Larkfield Software's update telemetry is the only evidence that its rings and kill switch work. It keeps the check-in logs long enough to answer a CRA final report, and it publishes its own recovery throughput to customers who ask.

Value. A board that sees concentration and recovery throughput next to the cost of control can trade them: a ring that costs $60,000 a year against a scenario that costs $1.1 million a day.

Monday rule: for one critical system, write the header block, price one loss scenario as lost plus paid (not as the day's revenue), and name the authority with the highest share of revenue it can interrupt in one action.

OUR VIEW, AS OF NOVEMBER 2026

Financial supervisors already ask about concentration. DORA requires a financial entity to assess ICT concentration risk before it contracts for critical functions (Articles 28 and 29), sets mandatory contract terms such as audit rights and exit strategies (Article 30), and on 18 November 2025 the European supervisory authorities designated the first critical ICT third-party providers for direct oversight. The UK's critical third parties regime, in force from 1 January 2025, does the same for providers HM Treasury designates. Halbrook's Irish subsidiary answers this question already. Expect it to spread from finance to other regulated sectors in the form this chapter uses: which single provider can stop which share of your critical services, and how fast you recover. Companies that already keep that number in their Ledgers will answer in a day; the rest will spend a quarter building it.

CHAPTER FIFTEEN

Own, Rent, Pin, Isolate: The Decision and the Contract



On 6 August 2024, CrowdStrike published its root cause analysis of the 19 July outage (told in full in Chapter Five and on the record in Chapter Thirteen). One mitigation in it reads like a contract clause that nobody had written: "Customers can choose where and when Rapid Response Content updates are deployed." A control a vendor adds after a failure is often one a customer could have asked for before it. This chapter turns fourteen chapters of cases into one decision per Trust Ledger row, and into the clauses that make the decision hold when the next push goes wrong.

Two axes, not four boxes

Books 1 and 2 used four words for the last column of their worksheets. Book 3 keeps the words and changes their shape, because the cases in this book showed that they answer different questions. The Trust Ledger's decision has one axis and two modifiers.

- **Own or Rent** is the axis. It answers one question: *who holds the authority to change this row?* You **own** a row when your people write, sign or push the change and can do it on your own clock. You **rent** it when someone else holds that authority and you rely on their clock. Many rows split: you rent the code and own the policy on top of it, such as the rings, the settings and the schedule. The Ledgers below write such a row as "Rent; Own the policy"; where the split hides a risk, break the row in two.
- **Pin** is a modifier. It freezes a version or a configuration on purpose. A pin must carry an **end date**, the date the frozen thing stops being fixed or must be reviewed. A pin without an end date is unpatched debt with no repayment plan.
- **Isolate** is a modifier. It contains a row you cannot change fast enough. An isolation must **name what the row is isolated from** (the network, the payment path, other tenants, the agent) and how you **prove** it holds: a test, a log, an attestation.

A fourth question sits beside the decision and is easy to forget: **lifecycle**. For each row you retain it, replace it or retire it. A Corvane kiosk image on an operating system with a published support end is a retain decision with a replace date already written on it. A row with no support end at all is a finding (Chapter Eleven).

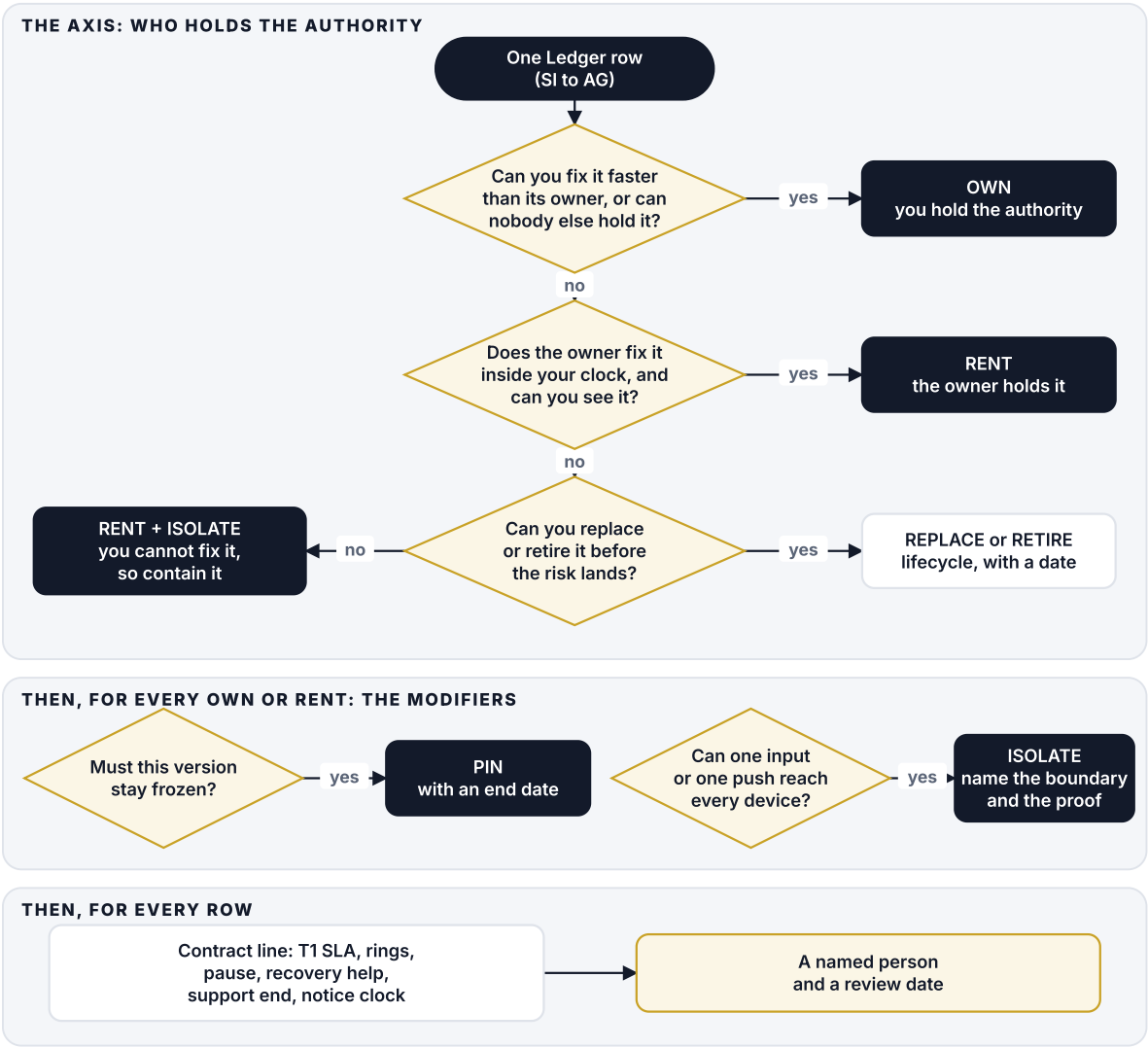
Written this way, a Ledger cell reads like a sentence a board can check: "Rent, Pin to 31 March 2027, Isolate from the payment network, proven by the quarterly segmentation test; owner: head of end-user computing; review: January 2027."

The flowchart, one row at a time

Figure 15.1 is the decision as a flowchart. It runs once per row, not once per vendor. One vendor can hold several rows, and the right answer is often different for each: Halbrook rents its EDR sensor's kernel code (DR) but owns the ring schedule for the content channel that feeds it (MU). Throughout the book the sensor's code is a DR row and its content channel an MU row.

FIGURE 15.1

One decision per row: an axis (Own or Rent), two modifiers (Pin, Isolate), then a contract line and a name



Run the flowchart once per Trust Ledger row, not once per vendor; a Pin must carry an end date and an Isolate must name its boundary and its proof.

Author's diagram based on the book's case files; NIST SP 800-161 Rev. 1 (C-SCRM); Regulation (EU) 2024/2847 Art. 13(8).

The first question is the hard one: **can you change this row and fix it faster than its current owner, or is it something nobody else can hold, such as your own signing key?** For most rows below the kernel the honest answer is no. A bank does not write microcode, and a hotel group does not sign UEFI images. For its own application, its own sign-in policy and its own update rings, the answer is often yes.

If you cannot own it, ask whether the owner fixes it **inside your clock** and whether you can **see** them do it. "Inside your clock" means the owner's T1 (fix available for your device class) plus your own T1 to T3 (fix on 90% of your fleet) fits the exposure. An internet-facing VPN appliance with a published T1 of weeks fails that test against an exploitation clock that Chapter Sixteen measures in days. "See" means the evidence tags from Chapter One: observed on your fleet, documented by the vendor, or contracted. A row that only reaches "inferred" or "unknown" is still rented, but without evidence, and that is a finding.

If the owner is too slow or invisible, the next question is lifecycle: can you **replace or retire** the row before the risk lands? If you can, that is the decision. If you cannot, you **isolate** it. Isolation is often a control you can hold yourself on a row you do not own. Speed is not the only test at each step: capability, liability, staffing and exit cost weigh on the answer too, and the flowchart is a starting point for that conversation, not a substitute for it.

Every Own and every Rent then passes two modifier questions. Must this version stay frozen for a period (a regulated build, a kiosk image, a certified payment terminal)? Then **Pin**, with an end date. Can untrusted input or one approved push reach every device in scope? Then **Isolate**, with a named boundary and a proof. Finally every row, whatever its letters, gets a contract line and a named person.

When "own" is the wrong answer

The series has repeated one rule since Book 1: owning more is not automatically safer. Every row you own is a row you must patch, staff and recover, on a clock you now answer for. The cases in this book make the rule concrete.

- **Firmware and silicon (SI, FW).** Most companies should not own these rows. The fix for a parser flaw in firmware comes from the firmware vendor through the device maker (Chapter Four); a revocation list does not repair it. What a company can own is the **policy** on top: Secure Boot on, a firmware update ring, the status of the 2023 Secure Boot certificates tracked per device. Own the policy, rent the code.
- **Kernels and drivers (KN, DR).** Writing kernel code to avoid a vendor's kernel code moves the risk; it does not remove it. Halbrook could build its own endpoint sensor. It would then own a kernel driver on 2,500 branch PCs with no larger engineering team than the one it has today. Rent the sensor, own the content rings and the recovery drill.
- **Identity providers (ID).** Running your own identity provider makes you the authority over token-signing keys. Chapter Six showed what one signing key and one validation flaw can do at a hyperscaler with large security teams. Own the help-desk reset procedure and the conditional-access policy; rent the token service unless you have a team that can rotate keys on the hour.

Where owning is right, it is usually a thin layer: your own application, your own update channel, your own rings, your own keys for your own code. Larkfield Software is the clearest case. It must own its code-signing key and its update pipeline, because no one else can. The CRA, from the other side, makes the same point: a manufacturer is responsible for the vulnerabilities of its product "including its components" for the support period (Regulation (EU) 2024/2847, Article 13(8)).

Decision lever	Windows	macOS	iOS	Linux	Android
How long a customer can hold an update (Pin, short)	Quality updates deferred up to 30 days and paused up to 35 days by policy; feature updates deferred up to 365 days	Apple documents release dates for devices on a 90-day deferral	Same 90-day deferral as macOS, set through device management	As long as the team chooses; no vendor limit on a self-managed host	Device owner can postpone once for 30 days, or freeze up to 90 days with a 60-day gap; makers may exempt security updates
How long a version can be held (Pin, long)	Long-term servicing channel with a fixed end: Windows 11 IoT Enterprise LTSC 2024 supported to October 2034	Tied to Apple's support for the hardware and OS version (Chapter Eleven)	Tied to Apple's support for the device	The distribution's long-term release, on its published schedule	The device maker's stated update period
Who holds the stop for third-party privileged code	The vendor's content channel; customer rings only where the contract or console provides them	System extensions approved by policy; Apple holds the platform	Apple only; no third-party kernel code	The team, per module and eBPF program	The device maker and Google
Firmware and silicon	Rent from the OEM; own Secure Boot policy and the firmware ring	Rent from Apple; firmware ships with the OS	Rent from Apple	Rent from the OEM, often through LVFS; own the ring	Rent from the maker and chip vendor

Sources for the table: Microsoft Learn, "Configure Windows Update client policies"; Apple Platform Deployment, "Manage software updates"; Android Developers, "Manage system updates"; Microsoft Lifecycle, Windows 11 IoT Enterprise LTSC 2024. All checked 18 November 2026.

The contract: seven clauses that carry the decision

For a rented row, a decision that is not in the contract or in a setting you control is a preference. The seven clauses below come from the cases in this book. They are written as asks a buyer can put in a request for proposal. Appendix H carries the full annotated text.

Clause	What to ask for	Why (the case behind it)	Rows
1. Patch SLA to T1	Days from the vendor becoming aware of an exploited flaw to a fix available for your device class, by severity; the vendor reports its actual T1 for the last three critical cases	Edge appliances and firmware fixes arrive on the slowest clocks in the book (Chapters Four, Eleven)	FW, DR, LR, EX
2. Staged rollout and customer rings for content	Every update and every content or configuration push reaches a canary group first; the customer can choose where and when content lands	CrowdStrike's 2024 mitigation list added staged deployment of template instances and customer control over Rapid Response Content (Chapter Five)	DR, MU
3. Pause and kill switch	A documented way for the customer to pause a channel within one hour, and for the vendor to stop a bad push globally, both tested once a year	A stop that has never been drilled is not a stop (Chapter Nine)	MU, DR, AG
4. Recovery support	Vendor tools, people and a recovery path that works when the device cannot boot or the console is down; a stated recovery target	Recovery from a boot loop was often hands on keyboard, device by device (Chapters Five, Nine)	KN, DR, FW
5. Support end date	A printed end date for the product, the firmware and any component the vendor bundles; notice of change at least one replacement cycle ahead	The CRA sets a support period of at least five years, or the expected use time where that is shorter, for products placed on the EU market (Article 13(8))	All

6. Firmware support	Who builds, signs and ships firmware fixes, through which channel, for how long after the hardware is sold	Fixes for firmware flaws come from the firmware supply chain, not from a revocation list (Chapter Four)	SI, FW
7. Notification clocks	Hours from the vendor's awareness of an exploited vulnerability or a breach of its own systems to notice to you, plus logs and artifacts you can investigate with	Your own regulator clocks start from your awareness (Chapter Fourteen); CISA's Secure by Design Pledge (May 2024) asks vendors to give customers evidence of intrusions	ID, MU, LR

The seven carry the decision; they are not the whole contract. Standard supplier terms belong beside them: a software bill of materials (SBOM) with each release, with exploitability statements (VEX) where a listed flaw does not apply (Chapter 8); audit rights or independent assurance reports; notice of sub-processors; liability caps with carve-outs, and indemnity; and exit and termination assistance. For Halbrook's Irish subsidiary these are not optional: DORA Article 30 requires terms such as audit and access rights and exit strategies in contracts for ICT services that support critical functions. Clause 2 is also no longer only a customer's ask: in November 2024 Microsoft said its work with endpoint security vendors through the Microsoft Virus Initiative would focus on safe deployment practices for security product updates. CISA and the FBI's Secure by Demand Guide (6 August 2024) gives buyers a published list of questions to put to software makers, and it fits beside these clauses. Contracts and regulatory summaries vary by jurisdiction; this chapter is not legal advice.

Two of these clauses already exist in law for one of the three companies. As a manufacturer selling into the EU, Larkfield must send an early warning of an actively exploited vulnerability in its product "within 24 hours" of becoming aware of it, and a fuller notification within 72 hours (CRA Article 14(2)), from 11 September 2026. What the law asks of Larkfield as a supplier is a fair model for what Halbrook and Corvane can ask of theirs.

For the method behind the clauses, NIST SP 800-161 Revision 1 (May 2022, updated November 2024) is the reference for cybersecurity supply chain risk management, the practice of "identifying, assessing, and mitigating cybersecurity risks throughout the supply chain". It gives the vocabulary procurement and audit already use. The Trust Ledger is the one-page view that points to it.

What vendors usually refuse. This paragraph is the author's experience from negotiations, not a published fact. Vendors rarely accept a hard T1 for flaws they did not write (third-party components inside their product), uncapped liability for an outage, or a customer veto on security content during an active campaign. They more often accept staged rollout, customer-defined rings for non-urgent content, a named recovery contact, release notes for content, and a printed support end. Ask for the first group anyway; it tells you how the vendor thinks about the risk it puts on you. Settle on the second group in writing.

Exit and retirement

The last part of every decision is the way out. Three exits recur in this book's cases.

- **Supplier failure or removal.** A supplier can fail, be sold, or be barred. Chapter Seven told how a government stopped one vendor's update channel by a dated order. For each rented row, write the replacement lead time next to the support end. If replacement takes longer than the notice you would get, the row needs an isolation plan now.
- **Key destruction.** When you retire a signing key or a device identity, destroy it and record that you did. Larkfield's old code-signing certificates and Halbrook's retired device certificates belong in the same register as their live ones, with a revocation date and a destruction date.
- **Device resale and disposal.** A kiosk or a laptop that leaves the fleet still carries keys, enrolment records and sometimes firmware passwords. Retirement means removing it from management, revoking its credentials and wiping or destroying storage, in that order, with a record per device.

The kiosk fleet shows how dates drive the exit. Microsoft's lifecycle page lists Windows 10 IoT Enterprise LTSC 2021 support to January 2032 and Windows 11 IoT Enterprise LTSC 2024 to October 2034. A Corvane kiosk bought in 2026 with a seven-year depreciation schedule fits inside the 2024 channel and not inside the 2021 one. That one line in the Ledger decides which image the next purchase order specifies.

HOW IT GOT THIS WAY

Enterprise software contracts grew up around licenses, uptime and support response times, not around who may change the product on a customer's machine. Automatic updates turned that around. Each vendor that gained the right to push code also gained a seat at the customer's change board, usually without a clause saying so. Supply chain guidance followed: NIST's C-SCRM practice (SP 800-161, revised in May 2022), CISA's Secure by Design principles and the voluntary pledge of May 2024, and in the EU a binding law, the Cyber Resilience Act, in force since 10 December 2024 and applying in full from 11 December 2027. Each moved some of the duty back to the party that holds the authority. None of them writes your Ledger for you.

What hasn't changed: a fixed-lifecycle product still arrives with its end date printed on the day it ships, and the customer still has to put that date in the plan. Windows 10 2015 LTSB started on 29 July 2015 and its extended support ended in October 2025; Windows 10 IoT Enterprise LTSC 2021 started on 16 November 2021 and ends in January 2032. Both are about ten years, both were published at launch. Checked on Microsoft's lifecycle pages, 18 November 2026.

THE UNSEEN ANGLE

Isolate is often the least used decision in a Ledger. Teams spend their effort on the rows they can patch and leave the rows they cannot patch to the vendor. But when you can neither change a row nor wait for its owner, containment is a control you hold yourself: a network segment, a separate desktop for the payment screen, no AI agent on the kiosk image. A content ring that holds a push for an hour is staging, not isolation, and belongs in the MU row. Isolation costs little, it is testable, and it works whether or not the vendor's fix arrives. The catch is proof. An isolation that nobody tests decays into a diagram. Put the test and its last date in the Ledger cell. Source: the Ledger method, Chapter One; the segmentation evidence tags, Appendix G.

> *Rent the code, own the policy, and write the stop into the contract before the next push needs it.*

These rows in the Trust Ledger

Figures 15.2 to 15.4 are the three companies' filled Ledgers, eight rows each, with the eight core columns. They summarize decisions taken across the book; the chapter that carries each case is named in the cell. Values marked "to measure" are placeholders a real team fills from its own telemetry (Appendix D). The companies are invented; the clocks, dates and

mechanisms are real. Key to the cells: T1 is the day a fix is available for your device class, T3 the day it runs on 90% of your devices; O observed on your fleet, D documented by the vendor, C contracted, I inferred, U unknown; np not published, nc not collected; ESP is the EFI system partition, LVFS the Linux Vendor Firmware Service. Ring sizes differ by channel (5%, 25%, all for Intune; 1%, 10%, 100% for an app pipeline; four steps for Larkfield): each is a choice per channel, not a standard.

FIGURE 15.2

Halbrook Bank: the filled Ledger puts its decisions on identity and the five push channels

Row and component	Who can change it	Exposure and blast radius	Patch clock	Stop, revoke, recover	Who can see it	Support end	Decision and owner
FW OEM UEFI and Secure Boot keys on 2,500 branch PCs	Firmware vendor writes; OEM signs and ships; Microsoft holds the UEFI CA	Boot path and ESP; all 2,500 PCs	OEM firmware ring: to measure	OEM firmware update; 2023 CA status tracked per PC (Ch 4)	D: boot attestation	OEM firmware support end	**Rent**; **Own** the policy (Secure Boot on, firmware ring). Head of end-user computing
HV Cloud VDI host for advisors	Cloud provider	Tenant boundary; every advisor session	Provider's clock: np to the customer	Provider failover; region drill once a year	C: provider reports	Per contract	**Rent**; **Isolate**: no payment approval inside VDI. CTO
DR EDR sensor driver	EDR vendor writes and signs; sensor versions on Halbrook's rings	Kernel; 2,500 PCs	Sensor: per release, ring by ring	Pin N-1; rollback tested; boot recovery drill with recovery keys ready (Ch 5, 9)	D: vendor console. O: own crash telemetry	Per contract	**Rent** the sensor; **Pin** N-1 with an end date. CISO
ID Identity provider, MFA, help-desk reset	IdP vendor; IdP admins; help-desk staff	A phone call to the help desk; every account	Policy change: hours	Session and token revocation; reset only after a callback to a directory number and manager approval, or in person (Ch 6, 16)	O: sign-in and reset logs	Service; per contract	**Rent** the token service; **Own** the reset path and access policy. CISO
MU Windows Update, OEM tool, EDR console and content channel, MDM, teller-app pipeline	Five authorities	Each reaches all 2,500 PCs; EDR content in one push	Per channel: to measure	Pause per channel; Intune rings 5%, 25%, all; teller app 1%, 10%, 100%; EDR content: 50-PC canary, one-hour hold, tested pause	O for MDM and Windows Update; U for the OEM tool	n/a	**Own** the ring schedule across all five, including EDR content, where each vendor's console and contract provide rings. Head of end-user computing
LR WebView2 runtime and open-source libraries in the teller app	Microsoft; upstream maintainers; Halbrook's own team	Teller-screen content	Microsoft cadence plus Halbrook's restart window (Book 2, Ch 11)	Daily restart window; rebuild the app	O: running version reported	Microsoft lifecycle	**Rent**; **Own** the restart window and the app's dependency list. VP Engineering

Row and component	Who can change it	Exposure and blast radius	Patch clock	Stop, revoke, recover	Who can see it	Support end	Decision and owner
AG Advisor-assist agent	Model vendor; Halbrook's AI platform team configures	Reads client files, drafts messages; advisors only	Model updates on the vendor's clock	Revoke the agent identity; switch off tool access in minutes (Ch 16)	O: agent action log	Per contract	**Rent** the model; **Own** the permissions; **Isolate** from payments and any irreversible action. AI platform lead
EX Branch VPN and firewall appliances	Appliance vendor	Internet-facing; every branch	Vendor T1: to measure	Config backup; replacement plan (Ch 11)	D: vendor advisories	Vendor support end date	**Rent**; **Isolate** the management interface from the internet; replace by support end. Head of network

Halbrook rents most rows and owns the thin layer that decides when change reaches its branches: rings, the help-desk reset path, the restart window and agent permissions.

Author's worked example (invented company) based on this book's case files; values marked to measure come from the bank's own telemetry (Appendix D).

FIGURE 15.3

Corvane Hotels: a long-life fleet decides by support ends, staging and field visits

Row and component	Who can change it	Exposure and blast radius	Patch clock	Stop, revoke, recover	Who can see it	Support end	Decision and owner
FW OEM UEFI and Secure Boot keys on about 3,000 kiosks	Firmware vendor writes; OEM signs and ships	ESP and image-parsing path; all kiosks	To measure	OEM firmware update only (a revocation list does not fix parsers); 2023 CA status tracked (Ch 4)	D: attestation where enabled	OEM firmware support end	**Rent**; **Own** the policy: Secure Boot on, firmware ring by property. Head of property IT
KN Windows 11 IoT Enterprise LTSC 2024	Microsoft	Local kiosk; all kiosks	Monthly quality update; rings by property: to measure	Roll back the quality update; reimage from a golden image at the property	O: MDM compliance	October 2034 (Microsoft lifecycle)	**Rent**; **Pin** to LTSC 2024 until hardware refresh, at the latest 2034. Head of property IT
DR EDR sensor and payment-terminal driver	Two vendors write and sign	Kernel; every kiosk	Per vendor	Sensor and driver versions staged by property group; field-visit recovery priced per kiosk (Ch 14)	D: vendor consoles	Per contract	**Rent**; **Isolate** by staging: three canary properties first. CISO
MU Windows Update, OEM tool, EDR console and content channels, RMM, kiosk app pipeline	Five authorities	Each reaches 3,000 kiosks	Per channel: to measure	Kill switch for the kiosk app; pause per channel; EDR content to three canary properties first	Partial: O for MDM; U for RMM	n/a	**Own** the ring schedule across all five, where each vendor provides rings. Head of property IT
LR WebView2 in the check-in app	Microsoft	Guest input on the screen	Microsoft cadence; live after the nightly restart (Book 2)	Nightly restart window	O: running version	Microsoft lifecycle	**Rent**; **Own** the nightly restart. Kiosk product owner
SP Android front-desk tablets	Device maker and Google	Staff use; 300 properties	Maker cadence: to measure	Freeze periods (at most 90 days) only around peak season	O: patch level in device management	Maker's stated update period	**Rent**; replace before the update period ends. Head of property IT
AG Agents on kiosks	None allowed	n/a	n/a	n/a	O: image audit	n/a	**Isolate**: no agents on the kiosk image. CISO

Row and component	Who can change it	Exposure and blast radius	Patch clock	Stop, revoke, recover	Who can see it	Support end	Decision and owner
EX Property VPN appliances	Appliance vendor	Internet-facing; 300 properties	Vendor T1: to measure	Offline recovery kit per property (Ch 9)	D: advisories	Vendor support end	**Rent**; **Isolate** the management plane; replace at support end. Head of network

Corvane pins its kiosk operating system to a printed support end, stages every privileged push by property, and keeps agents off the kiosk image entirely.

Author's worked example (invented company); support dates from Microsoft Lifecycle (checked 18 November 2026); Android freeze limits from Android Developers, Manage system updates.

FIGURE 15.4

Larkfield Software: the vendor side owns its key, its channel and its pipeline

Row and component	Who can change it	Exposure and blast radius	Patch clock	Stop, revoke, recover	Who can see it	Support end	Decision and owner
ID Code-signing key for 400,000 desktops	Larkfield release engineering, in a hardware security module	Signs every update; one misuse reaches every customer	Key rotation drill: to measure	Revoke and re-sign; dual control; retired keys destroyed and recorded	O: HSM audit log	n/a	**Own**; **Isolate** signing from the build system. VP Engineering
MU Larkfield's own update channel	Larkfield	Pushes to 400,000 desktops	T1 to T3 from update telemetry: to measure	Staged rollout 1%, 10%, 50%, 100%; server-side stop; rollback build	O: update telemetry	n/a	**Own**. Head of release
MU Build system and CI	Larkfield; CI vendor	One compromise reaches every build (Ch 8)	n/a	Rebuild from clean sources; compare builds	O: build logs	Per contract	**Own** the pipeline; **Isolate** builds from developer laptops. VP Engineering
LR Bundled Chromium through Electron	Chromium and Electron upstream; Larkfield ships	Web content, files and messages the app opens	Electron supports the latest three majors (Book 2, Ch 11)	Rebuild and ship through its own channel	O: version telemetry	Per Electron major	**Own** the upgrade train; **Pin** only to a supported major, with its end date. Head of release
LR Open-source dependencies and the package registry	Maintainers; the registry	Build time; every build	To measure	Lockfiles; provenance checks; revert to a known build	D and O: software bill of materials	Per package	**Rent**; **Pin** through lockfiles with a monthly review. VP Engineering
KN, DR Kernel code	None shipped by Larkfield	n/a	n/a	n/a	n/a	n/a	**Own** the choice: no kernel driver, by design. CTO
SP OS signing and notarization services	Apple; Microsoft	Distribution and first launch	n/a	Platform revocation	D	n/a	**Rent**. Head of release
AG AI coding assistants	Model vendor; Larkfield configures	Write code into the repository	Model updates on the vendor's clock	Revoke tokens; no merge without human review	O: review and merge logs	Per contract	**Rent** the model; **Own** the review gate. VP Engineering

Larkfield is the authority its customers rent, so it owns the rows that can reach 400,000 desktops in one action and writes their stops into its own release process.

Author's worked example (invented company); Electron support policy as cited in Book 2, Chapter 11; CRA duties from Regulation (EU) 2024/2847.

SECURITY LENS

Who can change it? The decision column turns that question into a commitment. Own means your people hold the change. Rent means a named authority does, under a contract with a stop and a clock. Decisions are per row, not per vendor: one supplier can be rented on its code and controlled by you on its rings.

How fast can it be fixed? A rented row is only as fast as its owner's T1 plus your T1 to T3. Put the first in the contract and measure the second. A pin stops the clock on purpose; the end date restarts it.

What does AI change here? Our view (as of November 2026): faster discovery and exploitation (Chapter Sixteen) push more rows from "Rent" to "Rent and Isolate", because the owner's clock no longer fits the exposure. AI agents also become a row to decide on their own (AG): most companies should rent the model, own the permissions and isolate the agent from irreversible actions.

BUSINESS IMPLICATIONS

Security. Halbrook's decisions concentrate on ID and MU, because one help-desk reset or one console login reaches every branch. Corvane's concentrate on FW and lifecycle, because 3,000 kiosks in 300 properties make every boot failure a field visit. Larkfield's concentrate on its own key and pipeline, because it is the authority for 400,000 desktops.

Cost. Pin is a loan with interest. Every month a pin runs past its end date adds the cost of the eventual jump plus the exposure in between. Isolation is cheap by comparison: Corvane's per-property network segment and kiosk image without agents cost less per year than one week of field visits after a bad push.

Visibility. Isolate needs proof it holds. Each isolation cell names a test and its last run date. A row tagged "U" in the visibility column cannot be rented with confidence; it is the first row to fix.

Value. The filled Ledger becomes the board report. Halbrook's audit committee no longer asks "are we secure?" but "which three rows have no tested stop, and when will they?" That question has an owner and a date.

Monday rule: for the critical system you picked in Chapter One, write one decision per row (Own or Rent, plus any Pin with an end date and any Isolate with a named boundary and its proof) and put the matching clause into the next renewal of each rented row.

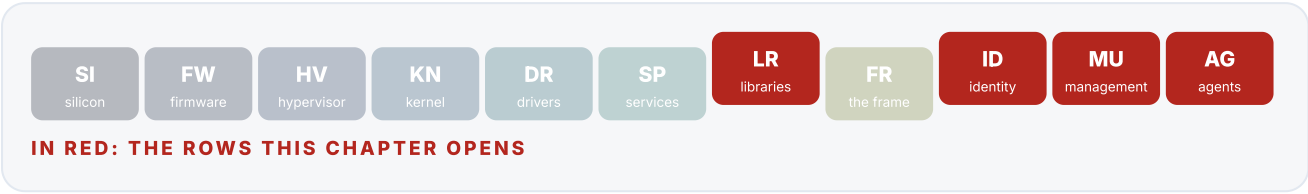
OUR VIEW, AS OF NOVEMBER 2026

Customer-controlled rings for security content are likely to become a standard contract term for endpoint and management products, because one vendor added them after a public failure and buyers now know they can ask. Support-end dates printed at sale will spread from regulated EU markets to global contracts as the CRA's support-period duty becomes normal practice for manufacturers. The decision that will grow fastest in Ledgers is Isolate, applied to AI agents and to edge appliances whose owners' clocks no longer fit the exploitation clock.

PART VI · AHEAD

CHAPTER SIXTEEN

AI on Both Sides of the Stack



In mid-September 2025, Anthropic detected an espionage operation running through its own coding tool. Its report, published on 13 November 2025, says a group it assesses "with high confidence" was Chinese and state-sponsored used Claude Code as an orchestrator against roughly 30 targets, with AI executing "80-90% of tactical operations independently", and that its investigation "validated a handful of successful intrusions". In the same year, AI systems on the defender side were finding real flaws in widely used open-source code and drafting patches in under an hour. This chapter asks what changes, row by row, when both sides use the same kinds of tools, and it changes one decision: agents get their own row in the Trust Ledger, with an owner, permissions and a stop.

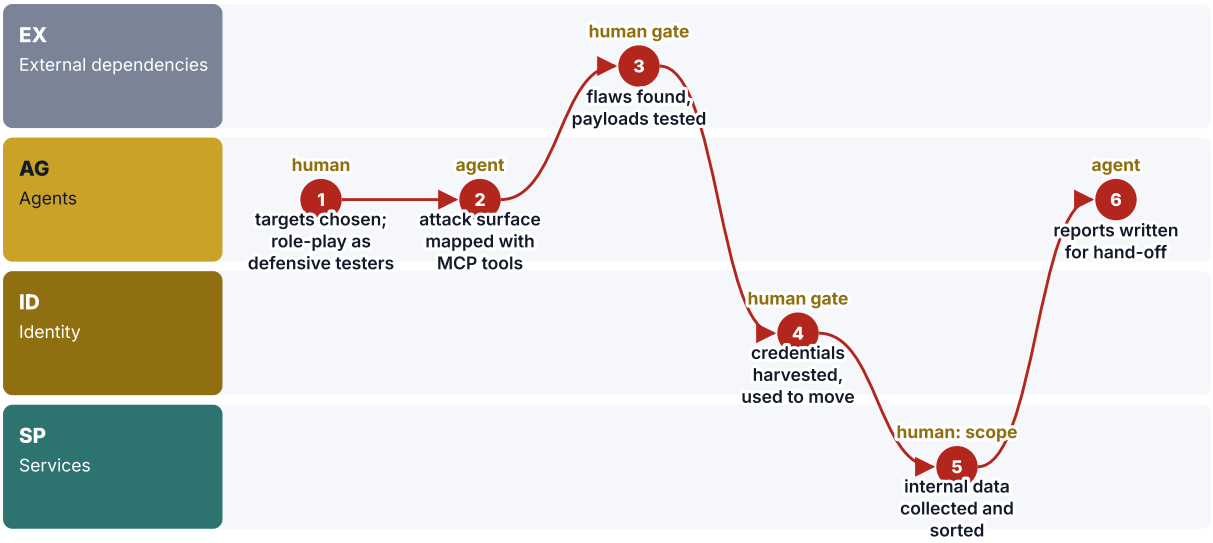
An operation that ran on an agent

Book 1, Chapter 13 argued that AI moves the seams of a system without removing them. Anthropic calls GTG-1002, its designation for the group, "the first documented case of a large-scale cyberattack executed without substantial human intervention". That is the vendor's characterization of its own incident. The account below is the report's own; no independent investigation has been published, and outside researchers quoted in the press questioned the report because it published no indicators of compromise.

The operators built a framework that used Claude Code and Model Context Protocol (MCP) tools, an open standard Anthropic introduced on 25 November 2024 for connecting AI assistants to tools and data. The framework split the attack into small tasks, such as scanning, credential checks and data extraction, "each of which appeared legitimate when evaluated in isolation". The operators told the model they worked for a legitimate security firm doing defensive testing. In the report's words, that role-play let them operate long enough to launch the campaign before "the sustained nature of the attack triggered detection".

FIGURE 16.1

GTG-1002 as Anthropic reports it: agents did the volume, people kept the irreversible gates



Numbered phases in the report's order. Gold labels mark who decided each step.

Six phases from Anthropic's report (November 2025), placed on the Ledger rows they touched; the report gives no CVEs and names no victims.

Source: Anthropic, "Disrupting the first reported AI-orchestrated cyber espionage campaign", full report, November 2025 (updated 17 November 2025). Attribution and figures are Anthropic's.

Three details change how a CTO reads the case.

- **Humans kept the gates.** The report places human decisions at target selection, at the move from reconnaissance to exploitation, at the use of harvested credentials, and at the final scope of data taken. The agent did the volume; people made the irreversible choices. The same pattern is the right one for defenders, and Halbrook's agent row below copies it.
- **The tempo was the signal.** Anthropic describes "thousands of requests" at peak, "multiple operations per second". A human operator does not work at that rate. Rate and pattern are what a defender can see when the content of each request looks routine.
- **The agent was unreliable.** The report says the model "frequently overstated findings and occasionally fabricated data", claiming credentials that did not work. Anthropic calls this "an obstacle to fully autonomous cyberattacks". It is also an obstacle to fully autonomous defense.

Anthropic banned the accounts over ten days, notified affected organizations and worked with authorities. In Ledger terms, the model provider was an **authority** on the attack path: it could see the activity and it could stop it, for its own model only. That is the shape of every rented AI row: the vendor holds a stop you do not. Google's Threat Intelligence Group gave a second vendor's account on 5 November 2025: for the first time it had found malware families that call a language model while they run, one of them (PROMPTFLUX) still in testing, and it disabled the accounts involved.

The clock that moved: time to exploit

The thesis of this book has a second half: discovery and exploitation are getting faster than the time a fix takes to reach your fleet, and AI may speed them further. The best public series on the first half is Mandiant's time-to-exploit analysis. **Time to exploit** (TTE) is Mandiant's "average time taken to exploit a vulnerability before or after a patch is released".

FIGURE 16.2

Time to exploit fell from about two months to days, then below zero



Mandiant's average time to exploit by period, with the 2025 estimate shown apart because it comes from a different publication and is an estimate.

Source: Mandiant (Charrier, Weiner), "How Low Can You Go? An Analysis of 2023 Time-to-Exploit Trends", 15 October 2024; Mandiant, M-Trends 2026 (blog, 23 March 2026). Method notes in the text.

Mandiant reported an average TTE of 63 days for 2018 to 2019, 44 days for 2020 to the start of 2021, and 32 days for 2021 to 2022. For 2023 it reported "an average of just five days" across 138 vulnerabilities it tracked as exploited in the wild, 97 of them zero-days and 41 n-days, flaws first exploited after their fix was public (Casey Charrier and Robert Weiner, 15

October 2024). Its M-Trends 2026 report (23 March 2026) put the mean time to exploit for 2025 at "an estimated -7 days, meaning exploitation is routinely occurring before a patch is even released".

Read the method before the trend. The 2023 average excludes 15 outliers. A year with more zero-days pulls the average down, because a zero-day is exploited before the patch by definition. The M-Trends figure is an estimate from a different publication. None of these sources attributes the fall to AI, and this book does not either: the trend began before capable models existed, and AI can only be said to have **accelerated** work that was already getting faster.

The n-day numbers are the ones a patch owner should pin to the wall. Of the 41 vulnerabilities exploited in 2023 after their fix was public, Mandiant found 12% exploited within one day, 29% within one week and 56% within one month. A fleet that took 30 days to reach 90% (T3) would have been behind on more than half of the 41 n-days Mandiant saw exploited that year.

The defender's machines: finding and fixing

AI on the defender side moved from claims to dated results in about a year.

- **1 November 2024.** Big Sleep, a joint effort of Google Project Zero and Google DeepMind, reported "an exploitable stack buffer underflow" in SQLite. The developers fixed it the same day, before it reached an official release. The team wrote that it believed this was "the first public example" of an AI agent finding a previously unknown exploitable memory-safety issue in widely used software.
- **15 July 2025.** Google said Big Sleep, working from Google Threat Intelligence leads, found CVE-2025-6965 in SQLite, a flaw "known only to threat actors and was at risk of being exploited", and described this as the first time it believed an AI agent had been used "to directly foil efforts to exploit a vulnerability in the wild". That is Google's claim; it has not published the details.
- **8 August 2025.** At the DARPA AI Cyber Challenge (AIxCC) final, the seven finalist systems found 54 of the 63 synthetic vulnerabilities in the final's challenges and patched 43 (DARPA corrected an earlier total of 70). They also found 18 real vulnerabilities in open-source code and provided 11 patches for real vulnerabilities. DARPA reported an average of 45 minutes per patch and about $152 per competition task. Team Atlanta won the $4 million first prize; Trail of Bits and Theori placed second and third. DARPA said all seven systems would be released as open source.
- **6 October 2025.** Google DeepMind introduced CodeMender, an agent that proposes and rewrites code fixes, and said it had already upstreamed 72 security fixes to open-source projects, each reviewed by people before submission.

The same tools strain the people who receive reports. In June 2025 XBOW, an autonomous testing system, reached the top of HackerOne's US leaderboard after about 90 days of submissions, by the company's own account. In January 2026 Daniel Stenberg, curl's lead maintainer, ended the project's bug bounty from the end of that month, to remove the incentive for made-up reports, many of them AI-generated. Bounty programs and volunteer maintainers now triage at machine volume, which is the pressure Chapter 12 points to.

These results can lower the cost of T0 and T1 for code someone can change. They do nothing for T2 and T3 on your devices. A fix found in 45 minutes still waits for the maintainer to accept it, the vendor to ship it, your rings to promote it and your devices to restart. AI triage cuts the cost of a fix most for the companies that own their release path: Larkfield, which can merge, sign and push a fix to 400,000 desktops on its own clock, gains the most. A company that rents every row gains a faster advisory and the same wait.

This is where the unseen angle of this chapter comes from: if discovery and exploitation take days, much of the risk sits in the tail. The rows with the slowest T3, such as firmware, edge appliances and kiosks that restart once a week, are where you lose.

Agents as an authority: the AG row

An **agent** in this book is any AI system or automation that acts with a user's or a service's privileges: it reads, writes, sends, clicks or calls tools. It belongs in the Ledger for the same reason an RMM console does. It can change your systems, at machine speed, on someone's behalf. Apple has let apps expose actions to assistants through App Intents since 2022; in 2025 Windows began to give the agent itself a separate identity. The table shows what each platform documents, dated.

Agent question	Windows	macOS	iOS	Linux	Android
How apps expose actions to agents	Agent connectors in the experimental agent workspace (support page updated 5 December 2025)	App Intents (macOS 13 and later), for Apple Intelligence, Siri, Spotlight and Shortcuts	App Intents (iOS 16 and later)	No platform framework; agents run as ordinary processes and call tools directly	AppFunctions (Android 16 and later; experimental, limited access)
What the agent runs as	A separate agent account in a separate Windows session, created only after an administrator turns the setting on (off by default)	The app that hosts it, under the user's permissions	The app that hosts it, inside its sandbox	The user or service account that starts it	A caller holding the `EXECUTE_APP_FUNCTIONS` permission
What it can reach by default	On request, the six known folders (Documents, Downloads, Desktop, Music, Pictures, Videos), with consent per agent in later preview builds	What the user grants the host app	Only what each app's intents expose	Everything that account can reach	Only the functions each app declares
Vendor's stated risk	Microsoft names cross-prompt injection: content in documents or UI "can override agent instructions"	Apple's documentation describes the framework, not a risk statement		n/a	Google advises exposing "only safe, non-destructive tasks"

Sources for the table: Microsoft Support, "Experimental Agentic Features" (updated 5 December 2025); Apple Developer, App Intents framework metadata; Android Developers, AppFunctions overview and agent skill guidance. All checked 18 November 2026; preview features change often.

The pattern across vendors is the same as the pattern in GTG-1002: a separate identity for the agent, a narrow list of things it may touch, and a person at the irreversible step. What is not the same is maturity. Windows labels its framework experimental; Android calls AppFunctions experimental with limited access. A Ledger entry for an agent should cite the platform build it was checked on.

Prompt injection: an input problem without a patch

Most rows in this book have an input that untrusted parties can reach. For an agent, that input is any text the model reads: a web page, an email, a document, a ticket, a tool's output. **Prompt injection** is text in that input written to change what the agent does. OWASP lists

it first in its Top 10 for LLM Applications (LLM01:2025), next to "Excessive Agency" (LLM06:2025), and on 9 December 2025 published a separate Top 10 for Agentic Applications.

No general model-level fix has been demonstrated. A particular injection path or tool defect can be patched; the underlying problem cannot yet. A model that follows instructions in text cannot reliably tell the operator's instructions from an attacker's when both arrive as text. Microsoft's own agent page says so in plain words. The controls that reduce the impact are the old ones, applied to a new caller, and none of them is complete: least privilege for the agent's identity, isolation from the systems where a mistake cannot be undone, a human approval for irreversible actions, and a log of what the agent did that someone reads.

New silicon, new runtimes, attested compute

AI also adds components to rows that already exist. On-device model runtimes sit in LR, beside the other runtimes a fleet must patch. Their accelerators and drivers sit in SI and DR, with their own vendors and update channels. Model files and their tokenizers are inputs that a runtime parses, which makes them part of the exposure column, and a model downloaded from a public hub is a supply chain input like any package (Chapter 8).

When the work leaves the device, the question becomes who can see it on the server. Apple's answer, published on 10 June 2024, is Private Cloud Compute (PCC). Apple says devices will send data only to PCC nodes "running authorized and verifiable software images", checking that a node's measurements "match a software release in the public transparency log". That is attestation used as a trust gate. Chapter Seventeen explains what attestation proves and what it does not.

HOW IT GOT THIS WAY

Machines that find and fix flaws on their own are older than today's models. On 4 August 2016 DARPA ran the final of its Cyber Grand Challenge, in which fully automated systems found, exploited and patched flaws in software written for the contest; ForAllSecure's Mayhem won. Fuzzing at industrial scale followed, then the language models of the 2020s, which added reading code and writing patches in plain source. AIxCC in 2025 was the same contest moved to real open-source projects and the tools of its time. Agents with access to a user's files and accounts are the newer step: they turn a tool into an authority.

What hasn't changed: least privilege and a human approval for irreversible actions. Saltzer and Schroeder wrote in 1975 that "every program and every user of the system should operate using the least set of privileges necessary to complete the job". In November 2025, the GTG-1002 operators themselves kept people at the irreversible steps, and in December 2025 Microsoft's agent page described separate agent accounts with limited folder access. The principle is fifty years old; the new caller is the agent.

THE UNSEEN ANGLE

AI makes the tail matter more. When Mandiant's estimated time to exploit is negative and AI agents find bugs in minutes, the advantage goes to whoever closes the last 10% of devices fastest, not to whoever learns about the flaw first. The rows that lose are the ones nobody reports on: kiosks that reboot once a week, appliances patched by hand, firmware that needs a field visit. Look at your slowest T3, not your average. Sources: Mandiant TTE 2023 (15 October 2024); M-Trends 2026 (23 March 2026); Chapter Eleven's long-tail list.

> *AI can shorten discovery, and in documented cases attack work, on the rows it reaches. Your release path, your rings and your recovery shorten the fix.*

These rows in the Trust Ledger

Halbrook Bank gives its advisor-assist agent a row of its own and adds an AI exposure flag (optional column) to the rows that AI changes most.

Component	Who can change it	Exposure and blast radius	Patch clock	Stop, revoke, recover	Who can see it	Support end	Decision and owner
AG Advisor-assist agent on cloud VDI; reads client files, drafts messages and meeting notes	Model vendor (model updates); Halbrook AI platform team (tools, permissions); advisors (prompts)	Any document or email it reads can carry injected text; reaches every advisor's files it is granted	Model: vendor's clock, np. Tool permissions: Halbrook, hours	Agent identity revoked in one step; tool access switched off per tool; drill twice a year	O: Halbrook's agent action log; C: vendor abuse reports	Per contract	**Rent** the model; **Own** identity and permissions; **Isolate**: no payment, transfer or send without an advisor's approval. AI platform lead
ID Help-desk reset path (AI exposure flag: yes)	Help-desk staff; IdP admins	A convincing voice or message, now cheaper to produce; every account	Procedure change: days	Callback to a directory number plus manager approval, or in person, before any reset; never video alone (a deepfaked video call cost Arup about $25 million in 2024); session revocation	O: reset logs	n/a	**Own** the procedure. CISO
EX Branch VPN appliances (AI exposure flag: yes)	Appliance vendor	Internet-facing; exploitation clock measured in days or less	Vendor T1 plus Halbrook rollout: to measure	Management interface off the internet; replacement plan	D: advisories	Vendor support end	**Rent**; **Isolate**; shorten T3 to days. Head of network
MU Teller-app release pipeline (AI exposure flag: yes, on the defender side)	Halbrook engineering	Pushes to 2,500 PCs	AI-assisted triage and patch drafting; human review before merge	Rings 1%, 10%, 100%; rollback build	O: pipeline logs	n/a	**Own**; AI may draft a fix, never merge one. VP Engineering

SECURITY LENS

Who can change it? A new authority: the agent, acting with someone's privileges, and behind it the model vendor, the tool authors and anyone whose text the agent reads. Name all of them in the AG row.

How fast can it be fixed? Discovery and patch drafting now take minutes for code you own. Exploitation of public flaws takes days or less on Mandiant's numbers. Your T2 and T3 did not change by themselves. The gap between the two clocks is the risk.

What does AI change here? This chapter is the answer. In short: it can shorten discovery and attack work on the rows it reaches, it adds an AG row with its own permissions and stop, and it rewards the companies that can ship a fix on their own clock.

BUSINESS IMPLICATIONS

Security. Halbrook treats its agent as a privileged identity, whatever app hosts it: separate account, narrow tools, approvals at the irreversible step. Corvane keeps agents off its kiosk image entirely. Larkfield lets AI draft fixes for its own code and keeps a person at the merge.

Cost. AI triage saves analyst time for everyone, but it cuts the cost of a fix most where you can ship it. Larkfield's cost per fix falls; Halbrook's and Corvane's fall only on the rows they own, which is why the cheaper investment for them is a shorter T3, not a better scanner.

Visibility. Agent actions need their own audit trail, kept with the same retention as admin logs. Rate and pattern are among the signals, as GTG-1002 showed; a log that records only the final action misses it.

Value. AI-found bugs reward companies with a short T1 to T3. Larkfield can turn that into a sales point: a published median days from report to fix on customers' desktops.

Monday rule: give every agent in your company a row in the Trust Ledger with its own identity, its tool list, the irreversible actions it may not take without a person, and a tested way to switch it off.

OUR VIEW, AS OF NOVEMBER 2026

The rows that matter most over the next three years are the slow ones. AI-assisted discovery will keep producing more reported flaws, and exploitation will keep starting before or near the patch. T3 will decide more of the outcome than it did, so patch telemetry and rings should get more of the budget that scanners got before. Agents will become a standard Ledger row with its own identity class in the operating systems, and prompt injection will be handled the way the industry handled untrusted input before: by limiting what the reader is allowed to do, not by hoping it reads safely.

CHAPTER SEVENTEEN

The Next Trust Migration: Memory Safety, Attestation and Post-Quantum Keys



On 27 October 2026, NIST published the final version of its report on the move to post-quantum cryptography, NIST IR 8547. Its tables give the same two dates for every quantum-vulnerable signature and key-establishment scheme in its standards, at 112-bit security and above: "Deprecated after 2030", "Disallowed after 2035". Most companies will read that as a cryptography problem for later. This chapter reads it as a Trust Ledger problem for now, together with two other migrations already under way: memory safety and attestation. Each changes who can be trusted on which row, and each rewards the fleets that know their keys, their code and their evidence. The decision this chapter changes is the inventory: what you list today decides how much of the next migration you can do on your own clock.

Three migrations, one question

The book has asked one question of every layer: who can change it, how fast, and how do you know. The three migrations change the answer in different places.

- **Memory safety** changes the KN, DR, LR and SI rows. Code written in memory-safe languages, and hardware that checks memory accesses, remove or sharply reduce whole classes of bugs that attackers have used for decades, in the code they cover; unsafe blocks, foreign code and tagging's probabilistic checks leave gaps. They do not remove bugs.
- **Attestation** changes the "who can see it" column on every row it reaches. It can turn a vendor's claim about what is running into signed evidence a verifier can check, and it makes that verifier a new authority. It proves measurements, not good behaviour.
- **Post-quantum cryptography** changes the ID and MU rows most: every key that signs code, firmware, tokens and certificates, and every channel that carries updates. It also reaches VPNs, HSMs and any stored data that must stay secret for years. It is a migration window with dated targets, not one cliff.

None of the three is a product to buy. Each is a change in what you must ask of the authorities you already rent, and in what you must own yourself: the list.

Memory safety: classes, not bugs

A **memory-safety** bug is code reading or writing memory it should not: past the end of a buffer, after the memory was freed, through a pointer of the wrong type. On 16 July 2019 Microsoft's Security Response Center wrote that "~70% of the vulnerabilities Microsoft assigns a CVE each year continue to be memory safety issues". On 13 November 2025 Google's Android team reported that memory-safety vulnerabilities fell "below 20% of total vulnerabilities for the first time" in 2025, after years of writing new platform code in Rust. It estimated about 0.2 vulnerabilities per million lines of its Rust code, from about five million lines and a single potential flaw caught before release, against "closer to 1,000" memory-safety vulnerabilities per million lines in its historical C and C++ data, and said Android's 6.12 Linux kernel carries its first production Rust driver.

Read the Android result carefully. The share fell because new code was written in Rust while old C and C++ code aged and was fixed. It is a statement about a codebase and its new code, not about a language making software correct. Logic bugs, configuration mistakes, stolen keys and every identity case in Chapter Six are untouched. The right wording for a board paper is the one Google uses for its own numbers: fewer memory-safety vulnerabilities in the code written in a memory-safe language.

On 24 June 2025 NSA and CISA published a joint information sheet, "Memory Safe Languages: Reducing Vulnerabilities in Modern Software Development", asking software makers to adopt memory-safe languages and publish roadmaps. CISA and the FBI's Product Security Bad Practices guidance (version 2, January 2025) is more specific: new product lines in memory-safe languages, and for existing products a published memory safety roadmap "by the end of 2025". For a buyer that turns into a contract question for each rented row: what share of the code that parses untrusted input is memory safe, and what is the plan for the rest?

Hardware is the second front. Memory tagging gives each piece of memory and each pointer a small tag and lets the processor stop an access whose tags do not match. Android added platform support for Arm's Memory Tagging Extension (MTE) in Android 13, and the first listed devices are Pixel 8 and later, which shipped with Android 14; it is off by default for apps and offered as a developer option on some phones. Apple went further on 9 September 2025 with Memory Integrity Enforcement (MIE) in all iPhone 17 models and iPhone Air. Apple describes it as "always-on memory safety protection", built on Enhanced MTE (a specification Apple says it worked on with Arm, released in 2022) in synchronous mode, and covering the kernel and "over 70 userland processes". Apple's stated aim is to

make mercenary spyware chains "immensely more expensive and difficult to develop and maintain". The research line underneath is CHERI, a Cambridge and SRI International project funded by DARPA since 2010 that adds hardware capabilities to processors; Arm's Morello prototype boards began shipping to researchers in January 2022.

Memory-safety lever	Windows	macOS	iOS	Linux	Android
Vendor's own published measure	About 70% of Microsoft's CVEs per year were memory safety (2019)	Apple publishes design goals (MIE, 2025), not a share		No single owner; per subsystem	Below 20% of vulnerabilities in 2025 (Google, November 2025)
Memory-safe code in the platform	No published share (np)	No published share (np)		Rust support in the kernel (Chapter Five); drivers by choice	New platform code largely in Rust; first production Rust kernel driver in Android's 6.12 kernel
Firmware and silicon: hardware memory tagging	Depends on the CPU; not a platform default	Not stated for Macs in Apple's MIE post	MIE on A19 and A19 Pro (iPhone 17, iPhone Air)	Supported on Arm hardware with MTE, by configuration	MTE on select devices (Pixel 8 and later), opt-in

Sources for the table: MSRC (16 July 2019); Apple Security Research (9 September 2025); Google (13 November 2025); Android Developers, "Arm Memory Tagging Extension". Checked 18 November 2026.

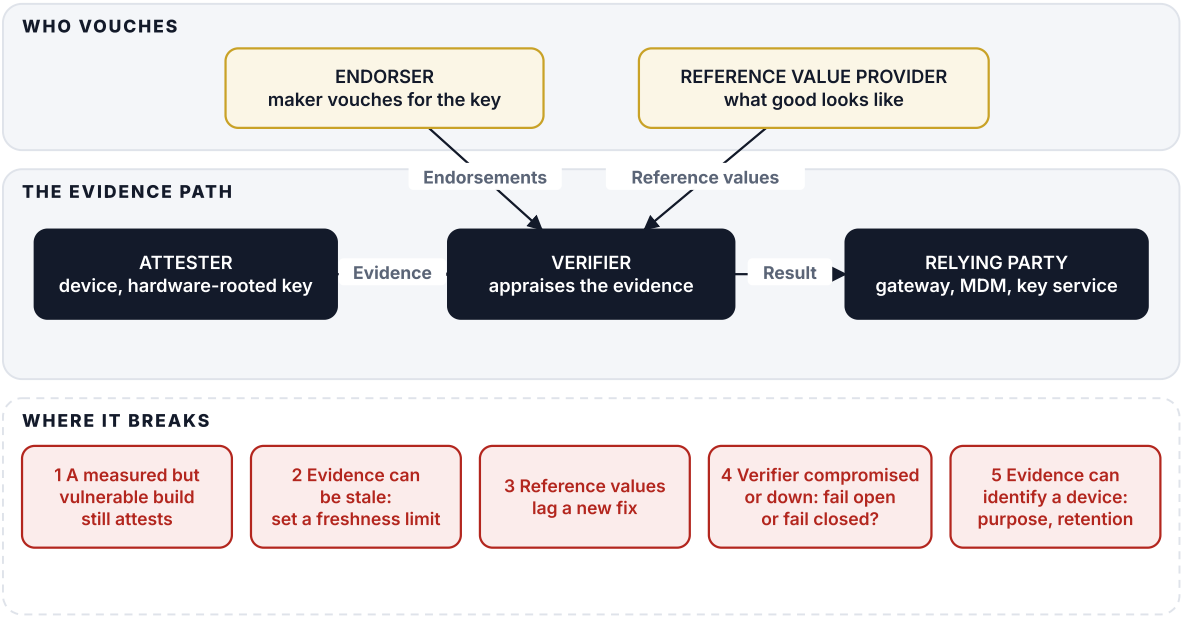
Attestation and its limits

Remote attestation is a device or service proving to someone else what software it is running. The IETF's architecture for it, RFC 9334 (January 2023), names the parts. An **Attester** (the device) produces **Evidence**, "a set of Claims" about itself, usually measurements of what booted, signed by a key rooted in hardware. A **Verifier** checks that Evidence against **Reference Values** (what good looks like, usually from the maker) and **Endorsements** (the maker vouching for the hardware key), and produces an **Attestation Result**. A **Relying Party** (your access gateway, your MDM, a key service) acts on that result.

Chapter Sixteen showed attestation as a trust gate: Apple's Private Cloud Compute devices send data only to nodes whose measurements match a release in a public transparency log. Used this way it is powerful. It also has limits that a CTO should write into the Ledger next to each use.

FIGURE 17.1

Attestation is a chain of parties, and each link has its own way to fail



RFC 9334's roles, with the five places a CTO should write a limit into the Ledger: measured is not benign, freshness, reference values, the verifier itself, and privacy.

Author's diagram based on IETF RFC 9334, Remote ATtestation procedureS (RATS) Architecture (January 2023).

- **It proves measurements, not benign software.** A signed, measured, vulnerable kernel produces valid evidence, and passes if the verifier's policy still accepts that measurement. Attestation tells you *which* build is running; the patch clock tells you whether that build is safe.
- **Freshness.** Evidence describes a moment. RFC 9334 describes nonces, signed timestamps and epoch identifiers to bound its age, and notes that "there is, however, always a race condition possible". Decide how old a result may be before you act on it.
- **Reference values.** A verifier is only as current as its list of good measurements. When a vendor ships a fix, the reference values must move too, or the patched devices fail and the old ones pass.
- **The verifier is an authority.** Whoever runs the verifier decides what "trustworthy" means for your fleet. A compromised or careless verifier passes anything. Put it in the Ledger as its own row, with an owner.
- **Privacy.** Evidence can identify a device. Corvane's guest kiosks and Halbrook's employees' phones need a stated purpose and retention for what is collected.

- **When the attestation service is down.** Decide in advance whether access fails open (business continues, trust unchecked) or fails closed (trust holds, business stops). Most teams discover their answer during the outage. Write it down and drill it, as Chapter Nine asks for every stop.

Post-quantum keys: a window, not a cliff

A large enough quantum computer would break the public-key schemes in use today, RSA and elliptic-curve cryptography, which sign code, firmware, certificates and tokens and set up most encrypted connections. Nobody has published such a machine. The migration starts anyway, for two reasons. Data recorded today under quantum-vulnerable key exchange can be decrypted later if such a machine arrives while the data still matters, and keys that sign firmware and software live as long as the devices that trust them.

The standards and the dates are now published.

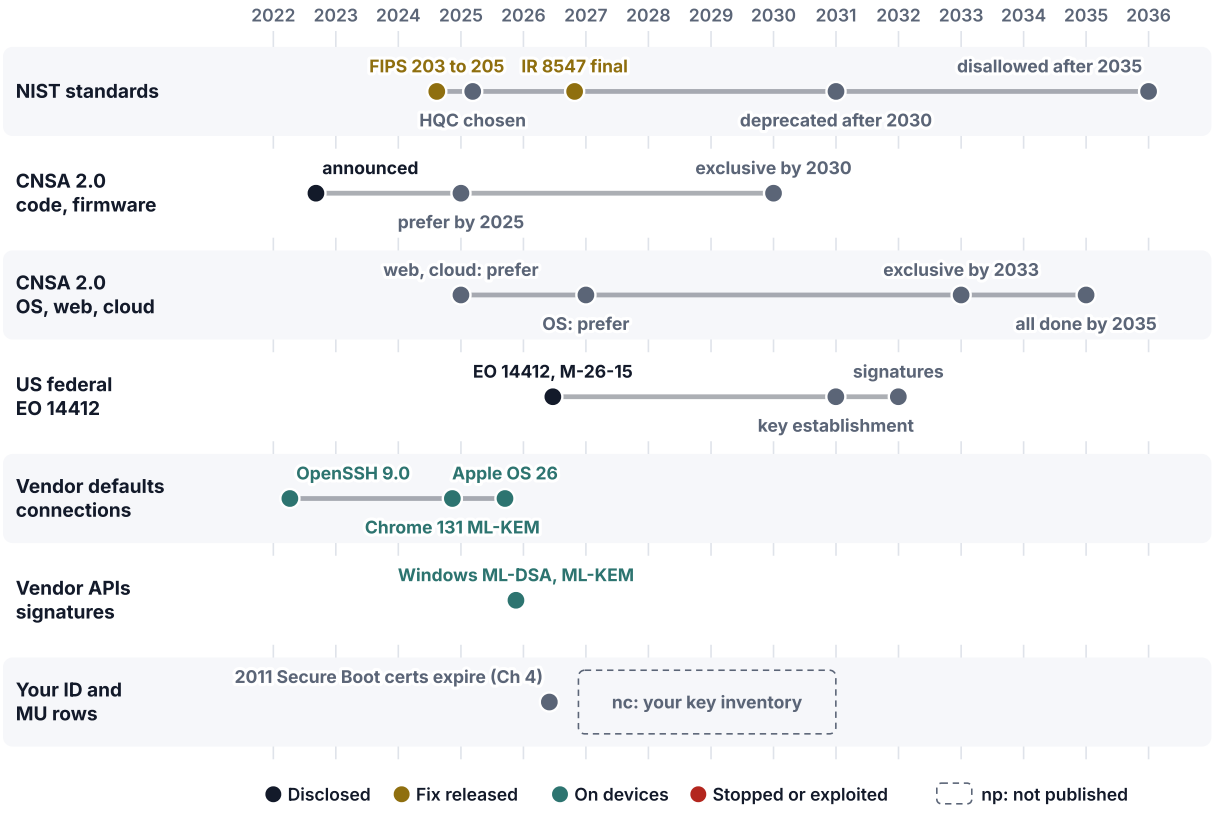
- **13 August 2024.** NIST published FIPS 203 (ML-KEM, key establishment, derived from CRYSTALS-Kyber), FIPS 204 (ML-DSA, signatures, from CRYSTALS-Dilithium) and FIPS 205 (SLH-DSA, hash-based signatures, from SPHINCS+). It said a further signature standard derived from FALCON was in development.
- **11 March 2025.** NIST selected HQC as a backup key-establishment algorithm "based on a different math approach than ML-KEM", with a final standard expected in 2027.
- **22 and 24 June 2026.** Executive Order 14412 and OMB Memorandum M-26-15 directed US federal agencies to move their most sensitive systems early. NIST summarizes the targets as high-value assets and high-impact systems migrating "by 2030 for key establishment and 2031 for digital signatures"; NIST IR 8547 cites M-26-15's deadline for key establishment as 31 December 2030.
- **27 October 2026.** NIST IR 8547 (final): quantum-vulnerable signature and key-establishment schemes at 112-bit security and above are deprecated after 2030 and disallowed after 2035 under NIST standards.

For US national security systems, NSA's CNSA 2.0 advisory (September 2022) sets dates per category. Software and firmware signing: "support and prefer CNSA 2.0 by 2025, and exclusively use CNSA 2.0 by 2030". Web browsers, servers and cloud services: prefer by 2025, exclusive by 2033. Traditional networking equipment such as VPNs and routers: prefer by 2026, exclusive by 2030. Operating systems: prefer by 2027, exclusive by 2033. NSA expects the whole transition complete by 2035. Halbrook, Corvane and Larkfield are not national security systems, but their vendors sell to them, so these dates set vendor roadmaps. Two civilian calendars point the same way. The EU's coordinated roadmap (NIS Cooperation Group, 23 June 2025) asks Member States to start by the end of 2026, move

high-risk systems by 2030 and as many others as feasible by 2035; it reaches Corvane's EU properties and Halbrook's Irish subsidiary. The UK's NCSC (20 March 2025) sets discovery and a plan by 2028, priority migration by 2031 and completion by 2035.

FIGURE 17.2

Post-quantum migration on the ID and MU rows: dated targets across a decade, not one cliff



Standards, government targets and vendor defaults are published; the lane that is not is your own inventory of keys and signatures. Year targets are drawn at the start of the year named.

Source: NIST CSRC (13 August 2024); NIST news (11 March 2025; 27 October 2026); NIST IR 8547 (October 2026); NSA CNSA 2.0 advisory (September 2022); OpenSSH release notes; Chrome 131 release notes; Apple support 122756; Microsoft Security blog (18 November 2025).

Vendors have already moved the defaults on connection keys, where hybrid schemes combine a classical and a post-quantum algorithm. OpenSSH 9.0 (8 April 2022) made a hybrid exchange its default; OpenSSH 10.0 (9 April 2025) switched the default to `mlkem768x25519-sha256`. Chrome 131 (12 November 2024) switched its hybrid TLS key exchange from Kyber to ML-KEM. Apple's iOS 26 (released 15 September 2025) and macOS Tahoe 26 advertise a hybrid ML-KEM exchange in TLS 1.3 by default. On 18 November 2025 Microsoft made ML-KEM and ML-DSA generally available in Windows 11 (24H2 and 25H2)

and Windows Server 2025 through its cryptography libraries. The connection layer is moving on vendors' clocks. The signature layer, where firmware, code, certificates and tokens live, is the slow part, and it is the part you need your own inventory of.

That is why **cryptographic agility**, the ability to change an algorithm or a key without rebuilding the system, is the real project. It has three steps, all on the ID and MU rows: list every key and certificate, with its algorithm, owner and expiry; find which ones are hard-coded in firmware, apps and appliances; and make the replacement path a tested process. Kernel-mode security products moving out of the kernel is the other migration in this period; Chapter Five covers it.

HOW IT GOT THIS WAY

Each of these migrations took longer than its first announcement suggested. Hardware capabilities for memory safety were a research topic for over a decade: CHERI work has been funded by DARPA since 2010, Arm's Morello boards reached researchers in January 2022, the Enhanced MTE specification came in 2022, and Apple shipped always-on tagging in a phone in 2025. Microsoft put a number on memory-safety bugs in 2019, and Google reported its own share falling below 20% six years later. In cryptography, NSA published CNSA 2.0 two years before NIST finished the standards it relies on. The pattern is the same each time: standards first, defaults in a few large platforms next, the long tail of firmware and appliances last.

What hasn't changed: the code that draws most web pages is still overwhelmingly C and C++. In Chromium's source tree at commit `16c0007869ad` (17 November 2026), excluding `third_party` and vendored directories, the author counted 327 Rust files (`.rs`) against 110,380 C and C++ files (`.c` , `.cc` , `.cpp` , `.cxx` , `.h` , `.hpp`), using `git ls-files` . File counts are a rough signal, not a share of code size or of the code attackers reach. Memory-safe code is arriving at the edges of large codebases, not replacing their centres. Measured 18 November 2026.

THE UNSEEN ANGLE

The fleets that struggled with the 2026 Secure Boot certificate move (Chapter Four) are the preview of the post-quantum key migration. That move asked one question, "which devices trust which signing certificate, and can we update it?", for one set of keys on one platform, with dates known years ahead. The post-quantum move asks the same question for every key that signs firmware, code, certificates and tokens, across every platform and appliance. A team that cannot yet report its 2023 CA status per device will not be able to report its signature algorithms per device either. Fix the first report and you have built the second. Sources: Chapter Four's certificate calendar; NIST IR 8547 (October 2026).

> *Memory safety removes classes, not bugs. Attestation proves what runs, not that it is good. Post-quantum is a window, not a cliff.*

These rows in the Trust Ledger

Corvane Hotels adds the three migrations to the rows of its kiosk fleet and property network. The optional "root of trust" column is folded into the component cell.

Component	Who can change it	Exposure and blast radius	Patch clock	Stop, revoke, recover	Who can see it	Support end	Decision and owner
FW Kiosk firmware signing keys and Secure Boot certificates (root: OEM and Microsoft keys)	OEM and firmware vendor sign; Microsoft holds the UEFI CA	Every kiosk's boot path; about 3,000 kiosks	2023 CA move: to measure. Post-quantum firmware signatures: vendor roadmap, np	Certificate update through Windows and OEM firmware; field visit as fallback	D: attestation where enabled. U: signature algorithm per device	OEM firmware support end	**Rent**; **Own** the per-device report of keys and algorithms. Head of property IT
ID Kiosk device certificates and back-end TLS (root: Corvane's private CA)	Corvane's PKI team	Every kiosk-to-backend connection	Certificate renewal: to measure	Revoke and reissue; algorithm change tested in one property	O: PKI inventory	Per certificate	**Own**; inventory every key with algorithm and expiry by 2027. CISO
EX Property VPN appliances	Appliance vendor	Internet-facing; 300 properties	Post-quantum key exchange: vendor roadmap, np	Replacement at support end	D: vendor statements	Vendor support end	**Rent**; buy only models with a dated post-quantum roadmap. Head of network
ID Attestation verifier for kiosk health (root: TPM endorsement keys)	MDM vendor runs the verifier; Corvane sets the policy	Decides which kiosks reach payment services	Reference values move with each OS update	Fail-closed for payments, fail-open for check-in; drilled yearly	O: attestation results	Per contract	**Rent** the verifier; **Own** the fail mode and freshness limit. CISO
SI Front-desk tablets: hardware memory tagging	Chip and device makers	Staff tablets; 300 properties	n/a	Replacement cycle	D: maker specifications	Maker's stated update period	**Rent**; prefer devices with memory tagging at next refresh. Head of property IT

SECURITY LENS

Who can change it? The same authorities as today, with new duties: chip makers decide memory tagging, OS vendors decide memory-safe code and connection defaults, verifier operators decide what "trusted" means, and every key owner decides when an algorithm changes. You own the inventory.

How fast can it be fixed? These are migrations measured in years, with published dates: 2030 and 2035 for NIST deprecation and disallowance, 2030 to 2033 for most CNSA 2.0 categories. The fast part (connection key exchange) is already moving on vendors' clocks. The slow part (signing keys in firmware and appliances) moves at the speed of your hardware refresh.

What does AI change here? Our view (as of November 2026): AI-assisted bug finding (Chapter Sixteen) makes memory-safety bugs cheaper to find in old C and C++ code, which raises the value of the classes that memory-safe code and tagging remove. It also helps a team build its key inventory from code and configuration, if someone checks the result.

BUSINESS IMPLICATIONS

Security. Larkfield can ask of its own code, and of every library it bundles, what share of the input-parsing code is memory safe. Halbrook prefers memory-tagged phones for its staff at the next refresh. Corvane's priority is the inventory of firmware keys.

Cost. Post-quantum migration is a key and certificate inventory project before it is a cryptography project. The 2026 CA move is the rehearsal. Budget the inventory now; the replacement is spread over hardware refreshes Corvane already plans.

Visibility. Attestation turns "trust me" into evidence only if someone checks it, keeps the reference values current, and decides what happens when the verifier is down.

Value. Verifiable claims about what ships become a sales feature. Larkfield can offer customers a signed statement of what it ships (build provenance, signing algorithm, memory-safe share) and the date of its post-quantum signing move, before its customers' procurement teams ask.

Monday rule: start a register of every signing key and certificate your critical system depends on, with algorithm, owner, expiry and whether it can be changed without new hardware; the 2026 CA report is its first page.

OUR VIEW, AS OF NOVEMBER 2026

The three migrations will not finish in five years, and a CTO should not try to time them. The hedge is the same for all three: own the inventory, rent from authorities that publish dated roadmaps, and isolate what cannot move. Memory tagging will spread from flagship phones to other device classes as chips gain it, and buyers will start asking for it in specifications. Attestation will become a common gate for access to sensitive services, which makes the verifier one of the most important rented rows in the Ledger. Post-quantum connection keys will be the default in major browsers, operating systems and remote-access tools well before 2030, with appliances and embedded devices later; post-quantum firmware and code signatures will lag by a hardware generation, and the fleets with an inventory will be the ones that finish first.

CLOSING

EPILOGUE

The Last Authority Is You



This book started at a power button and followed every authority that could change a teller's desk in the four minutes before a payment was approved: the chip and its microcode, the firmware and its keys, the hypervisor, the kernel, the drivers and their content, the identity provider and its help desk, the five channels that push code overnight, the runtimes, the frame, and the agents. Most of those authorities do careful work. None of them answers to the bank for the whole stack. One party is left that answers for all of it: the company that runs the system, deciding row by row whom it trusts, how fast it expects a fix, and how it gets back up.

What the stack looks like now

A stack used to look like layers of technology. Read through the Trust Ledger, it looks like a list of people and organizations, each with a key, a channel and a clock. Some sign. Some push. Some can stop a push and some cannot. Some publish their clocks and some publish nothing. The cases in this book were not failures of one layer. They were moments when an authority on one row could change many rows at once, and nobody downstream could see it coming or stop it in time.

Seeing the stack this way changes the questions a leadership team asks. "Are we secure?" becomes "which authorities can change this system without us knowing, and how many devices does one of their actions reach?" "Are we patched?" becomes "how many days from a vendor's fix to 90% of our fleet, and which devices are in the last 10%?" "Do we have a backup?" becomes "how long, measured in a drill, from a bad push to a working branch?"

> *You do not have to own the stack. You have to know who does, how fast they can fix it, and how you recover when they cannot.*

The Monday list

None of these five steps needs a new platform. Each needs a named person and a date. Together they take one critical system from "trusted" to "accounted for".

Step	What it takes	Who decides
1. Fill one Trust Ledger	Pick the system whose failure costs the most in a day: a payment approval, a check-in, a release pipeline. Write the header block (Chapter One), then one line per row from SI to AG. Any cell you cannot fill in 20 minutes gets "np", "nc" or "U". Those marks are the first findings.	Chief architect, with the system's business owner
2. List every authority that can push to it	Operating system updates, firmware tools, the EDR console and its content channel, the management console, remote-support tools, app updaters, package managers and agents. Count them. For each, write who holds the account and how a push is stopped (Chapter Seven).	Head of end-user computing, with the CISO
3. Put three numbers on every row that has them	T3, the measured days to reach 90% of the fleet with a fix, from your own telemetry (Appendix D); the support end, with its source; and, for each priced scenario the row appears in (Chapter Fourteen), the recovery time, measured in a drill, not estimated (Chapter Nine).	Head of end-user computing
4. Drill one stop	Choose the channel that reaches the most devices with one action. Pause it, on a test ring, and time how long the pause takes to hold. Then drill one recovery of a device that cannot boot. Write both times into the Ledger.	CISO, with operations

5. Write one decision per row, and its clause	Own or Rent; Pin only with an end date; Isolate only with a named boundary and a test. Put the matching clause into the next renewal of every rented row (Chapter Fifteen, Appendix H).	CTO, with procurement

The fifth step is the one that lasts. A Ledger without decisions is an inventory. A Ledger with decisions, owners and review dates is a control, and it becomes the page a board reads instead of a colour on a heat map.

Where the three companies landed

Halbrook Bank filled its first Trust Ledger for the branch payment-approval desk. It found five authorities that could push code to every branch PC overnight and could drill a stop for only two of them. It now owns the ring schedule across all five, after its EDR vendor's contract and console added customer rings, with a 50-PC canary and a one-hour hold for EDR content. It moved its help-desk reset path to a callback to a directory number with manager approval, or an in-person check, gave its advisor-assist agent a separate identity with no power to send or pay without an advisor's approval, and reports one number to its audit committee every quarter: the slowest T3 among its critical rows.

Corvane Hotels rebuilt its kiosk plan around dates. Its kiosks are pinned to an operating system channel with a published support end, and its next purchase order specifies the image that fits the depreciation schedule. Every privileged push now lands first in three canary properties. It keeps agents off the kiosk image entirely, keeps an offline recovery kit at each property, and has started the register of firmware keys and certificates that the 2026 certificate move showed it was missing. It now prices a bad push in field visits, not in tickets.

Larkfield Software accepted that it is an authority for 400,000 desktops. It owns its signing key in a hardware module with dual control, its update channel with staged rollout and a server-side stop, and its build pipeline, isolated from developer laptops. It lets AI draft fixes for its own code and keeps a person at every merge. It meets its 24-hour duty to report actively exploited vulnerabilities under the Cyber Resilience Act, and it publishes a figure its customers can hold it to: the median days from a vulnerability report to the fix on their desktops.

None of the three owns the whole stack. Each now knows who does, and has written down what it will do when one of them fails.

The last authority

Book 1 of this series ended at the human eye. Book 2 ended at the last pixel a system controls. This book ends at the last authority a system answers to. Vendors, maintainers, regulators and agents will keep changing the layers under your screens, most of the time for the better. What they cannot do is decide, for your company, which of them to trust with which row, how long to wait for them, and what to do when they are wrong. That decision does not come with any product. It stays with you.

> **ONE STEP FURTHER: THE JC10 RUNTIME**
>
> For one row of my own stack, I made the decision this book asks most companies not to make lightly: I own it. I build and run my own runtime under the jc10 name. I built it by asking it the questions this book asks of every layer: who can change it, how fast a fix reaches it, and how it recovers when something goes wrong. Owning it means I answer those questions myself, every day, on my own clock.
>
> A later book in this series goes deeper into how it is built. This book stays with the decision. You can follow the work at https://jc10.ai/.

APPENDICES

APPENDIX A

Glossary

These are the technical terms this book uses, in plain words and in alphabetical order. Each entry is a short definition and names the place where the term first appears; that chapter, or the one it points to, tells the full story with dates and sources. Where a definition depends on a version or a date, it is as of November 2026.

A to C

Term	Meaning	First used
A/B system updates	An Android update method that writes the new system image to a second slot, so a device whose new image fails to boot falls back to the old one.	Chapter Nine
Advanced Protection	Android 16's single switch for people at high risk, which among other things turns on hardware memory tagging for supported apps.	Chapter Ten
Agent	Any AI system or automation that acts with a user's or a service's privileges (reads, writes, sends, clicks or calls tools); it has its own Trust Ledger row, AG.	Chapter One
AGESA	AMD's firmware package for client processors, which PC makers build into their BIOS; a silicon fix delivered in AGESA reaches a device only after its maker ships a new BIOS.	Chapter Three
App Control for Business	The Windows feature that lets an organization decide which applications and drivers may run.	Chapter Two
App Intents	Apple's framework through which apps expose actions to Apple Intelligence, Siri, Spotlight and Shortcuts.	Chapter Sixteen
AppFunctions	Android's experimental framework (Android 16 and later, limited access) through which apps declare functions an agent may call.	Chapter Sixteen

Attestation	A device or service proving to a verifier, with signed evidence and a fresh challenge (a nonce) so an old proof cannot be replayed, what state it is in; on the devices in this book the evidence is rooted in hardware, and it proves measurements, not good behavior.	Chapter One
Attester, Verifier, Relying Party	The three roles in the IETF's attestation architecture (RFC 9334): the device that produces evidence, the service that checks it against reference values, and the system that acts on the result.	Chapter Seventeen
auid	The Linux audit ID of the user who first logged in, kept through su and sudo, so audit records name the person behind a process.	Appendix G
Authority	Any party that can change a layer of your systems: by writing, building, signing, publishing, pushing or configuring it, or by exploiting it.	Chapter One
Automated Device Enrollment (ADE)	Apple's service that enrolls company-owned devices in the company's MDM at first start and makes them supervised.	Chapter Seven
Background Security Improvements	Small Apple security fixes delivered between releases (from iOS, iPadOS and macOS 26.1); a user can remove them only all at once.	Chapter Two
Baseband	The processor and firmware that run the cellular radio in phones and some laptops.	Chapter Three
Baseboard management controller (BMC)	A separate processor on a server that gives remote console access below the operating system.	Chapter One
Binding Operational Directive (BOD)	An order from CISA that binds US federal civilian agencies, such as BOD 26-04 on patch deadlines and BOD 26-02 on end-of-support edge devices.	Chapter One
BitLocker recovery key	The 48-digit password that unlocks a Windows disk when the TPM will not; recovery from a boot failure often needs it.	Prologue
Blast radius	How many devices, users or tenants one action on a row can reach.	Chapter One
BlastDoor	A tightly sandboxed iOS service, from iOS 14, that parses almost all untrusted data arriving in iMessage.	Chapter Ten
Boot ROM	Code fixed in the chip at manufacture that starts the chain of trust on Apple devices; it cannot be updated.	Chapter Two
Bootkit	Malware that runs during boot, before the operating system and its defenses start.	Chapter Four

Break-glass path	An emergency way in, such as a sealed offline list of recovery keys or a hardware-key admin account, kept for when the normal path has failed.	Chapter Six
Bring your own vulnerable driver (BYOVD)	An attack in which someone with admin rights installs a legitimately signed driver with a known flaw and uses the flaw to act in the kernel.	Chapter Five
Bug bounty	A published reward for a vulnerability reported so that it can be fixed, paid by the vendor or by a platform or fund that runs the program.	Chapter Twelve
Caliptra	An open-source design, hosted by the CHIPS Alliance, for a root-of-trust block inside data-center chips.	Chapter Three
Canary deployment	Sending a change first to a small group of devices, so that a fault shows there before it reaches the rest.	Chapter Five
CERT/CC	The CERT Coordination Center at Carnegie Mellon University, which coordinates vulnerability disclosure and publishes vulnerability notes numbered VU#.	Chapter Four
Certificate authority (CA)	An entity whose certificate others trust to vouch for keys; Microsoft's UEFI CA, for example, signs third-party boot code.	Chapter Two
Channel File	The file in which CrowdStrike delivers Rapid Response Content to its sensor; Channel File 291 carried the 19 July 2024 fault.	Chapter Five
CHERI	A research project of Cambridge and SRI International, funded by DARPA since 2010, that adds hardware capabilities to processors to enforce memory safety.	Chapter Seventeen
CI/CD pipeline	The automated system that builds, tests and publishes software; whoever can change it can change what ships.	Chapter Seven
CISA	The US Cybersecurity and Infrastructure Security Agency, which runs the KEV catalog and issues directives to federal agencies.	Chapter One
CNSA 2.0	NSA's suite and timetable for moving US national security systems to post-quantum algorithms, by category, with the transition expected complete by 2035.	Chapter Seventeen
Code-signing key	The private key whose signature makes an operating system accept a program as coming from its publisher.	Chapter Six

Component-based servicing (CBS)	The Windows servicing stack that installs packages such as cumulative updates; `Get-HotFix` lists only what it installed.	Appendix D
Concentration	For one authority, the share of revenue it can interrupt with one approved action through its normal channel.	Chapter Fourteen
Content (security content)	Rules and configuration data a security sensor reads to decide what to watch and block; it can change privileged behavior without a new release.	Prologue
Content Interpreter, Content Validator	In CrowdStrike's sensor, the code that tests live data against content criteria, and the check that validates new content before it ships.	Chapter Five
Cross-signing, trust overlap	In PKI, one certificate authority signing another's certificate, as Microsoft cross-signed kernel driver signatures (Chapter Five); Chapter Nine also uses the word for the overlap in which a new key is trusted alongside the old one, so devices can move before the old one is withdrawn.	Chapter Five
Cryptographic agility	The ability to change an algorithm or a key without rebuilding the system.	Chapter Seventeen
CSAF	Common Security Advisory Framework: an OASIS standard for machine-readable security advisories, which Red Hat and others publish.	Appendix D
CVE	Common Vulnerabilities and Exposures: the public identifier given to one vulnerability, such as CVE-2014-0160.	Chapter One
CVE Numbering Authority (CNA)	An organization allowed to assign CVE identifiers; the Linux kernel became its own CNA in February 2024.	Chapter Five
CVSS	The Common Vulnerability Scoring System, which rates severity from 0 to 10.	Chapter Eight
Cyber Resilience Act (CRA)	Regulation (EU) 2024/2847: security duties for makers of products with digital elements sold in the EU, with reporting from 11 September 2026 and most obligations from 11 December 2027.	Reader's Brief
Cyber Safety Review Board (CSRB)	A US board that reviewed major incidents, including Log4j (2022) and Storm-0558 (2024); its members were dismissed in January 2025.	Chapter Six

D to K

Term	Meaning	First used
Data-independent timing (DIT)	A processor mode that cryptographic code can turn on so that timing does not depend on secret data; on M3 chips it disables the prefetcher GoFetch abused.	Chapter Three
db (allowed signature database)	The UEFI store of certificates and hashes of boot code that may run.	Prologue
dbx (forbidden signature database)	The UEFI store of signatures and hashes of boot code that must not run; adding an entry revokes it.	Prologue
Declarative device management (DDM)	Apple's MDM model in which the device applies declarations on its own and reports status items back to the server.	Appendix D
Deferral	A policy that holds an update back for a set number of days after release.	Chapter One
DFU mode	The iPhone's device firmware update mode, used to restore a device over USB when its boot chain fails a check.	Chapter Four
Digital Operational Resilience Act (DORA)	Regulation (EU) 2022/2554, in application from 17 January 2025, with incident reporting clocks for EU financial entities.	Reader's Brief
Direct memory access (DMA)	A device reading and writing the computer's memory directly, without the CPU, as Thunderbolt, USB4 and PCIe devices can.	Chapter Three
dm-verity	A Linux and Android kernel feature that checks each block of a read-only partition against a signed hash tree as it is read.	Appendix G
DriverKit, system extensions	Apple's frameworks, from macOS 10.15, that let drivers and security tools run in user space instead of the kernel.	Chapter Two
Early Launch Antimalware (ELAM)	A Windows driver class started before other boot drivers, so security software can judge the drivers that follow.	Prologue
eBPF	Small programs loaded into the Linux kernel after a verifier checks them; more constrained than a kernel module, not risk free.	Chapter One
Edge appliance	A device at the network boundary, such as a VPN gateway or firewall, that decides who reaches internal systems; one kind of EX row (external dependencies).	Chapter Eleven
EFI System Partition (ESP)	The disk partition from which UEFI firmware loads boot code; BlackLotus and LogoFAIL both used it.	Chapter Four

Emergency Directive (ED)	A CISA order to US federal civilian agencies to act on an urgent threat by a set date, such as ED 21-01 on SolarWinds.	Chapter One
End of support	The date after which a component's owner no longer commits to fix it; the component may keep working.	Chapter One
Endpoint detection and response (EDR)	A security product with a sensor on each device and a console that can isolate, kill and run scripts; on Windows the sensor commonly uses a kernel driver.	Prologue
Endpoint Security framework	Apple's user-space interface through which security tools watch and block system events on macOS.	Chapter Two
Enterprise mobility management (EMM)	A product that manages Android devices through Android Enterprise, as a fully managed device or a work profile.	Chapter Two
Entity List	A US Commerce Department list that restricts exports to named organizations; it is not a criminal finding.	Chapter Seven
Entra ID	Microsoft's cloud identity provider, formerly Azure Active Directory, which signs users in, holds BitLocker recovery keys and keeps sign-in and audit logs.	Chapter Six
Evidence tags (O, D, C, I, U)	The marks on Ledger evidence, used most in column 6: observed on your fleet, documented by the vendor, contracted, inferred or unknown.	Chapter One
Exploit broker	A company that buys working exploits and sells them on, usually to clients it does not name.	Chapter Twelve
Exposure	Which untrusted inputs reach a component (network, files, messages, USB, a help-desk call, a console login), and with what privilege.	Chapter One
Extended Security Updates (ESU)	Microsoft's paid security updates after end of support; for Windows 10, $61 per device in year one, doubling each year, for up to three years.	Chapter Eleven
Fail open, fail closed	What access does when a check (such as an attestation service) is down: allow and continue unchecked, or deny and stop.	Chapter Seventeen
FIDO, WebAuthn, CTAP	The standards behind passkeys: FIDO is the alliance, WebAuthn the browser interface, CTAP the protocol between devices.	Chapter Six
FileVault	macOS disk encryption; its personal recovery key can be escrowed with the device management service.	Chapter Nine

FIPS 203, 204, 205	NIST's first post-quantum standards (13 August 2024): ML-KEM for key establishment, ML-DSA and SLH-DSA for signatures.	Chapter Seventeen
Firmware	The code that starts the hardware and finds the operating system, and the code inside other chips on the board.	Prologue
Form 8-K	The report a US public company files with the SEC to disclose a material event, including a material cybersecurity incident.	Reader's Brief
Frame Ledger	Book 2's worksheet for how one critical screen is drawn, hop 0 to hop 10; in this book it collapses to the FR row.	Chapter One
fwupd and LVFS	The Linux firmware update client and the Linux Vendor Firmware Service it draws from; coverage depends on each device maker taking part.	Chapter One
Gatekeeper, notarization	On macOS, the check that software from outside the App Store is signed by a registered developer and has been checked (notarized) by Apple.	Chapter Two
Generally available (GA)	Released for production use, as opposed to a preview.	Chapter Two
Generic Kernel Image (GKI)	Google's common Android kernel, from Android 12 with kernel 5.10, with chip and board support moved into loadable vendor modules.	Chapter Two
Golden image	A known-good system image that can be laid down on a device to rebuild it.	Chapter Nine
HAR file	A browser recording of web traffic, uploaded for troubleshooting, that can contain session tokens.	Chapter Six
Hardware security module (HSM)	A tamper-resistant device that stores private keys and signs with them, so the key never leaves it.	Chapter Six
Help-desk reset path	The procedure by which support staff reset a password or move an MFA method; whoever can talk them into it can become the user.	Prologue
Hotpatch	Windows updates applied without a restart; Microsoft plans eight hotpatch months and four baseline months a year.	Chapter Nine
Hybrid key exchange	A connection key exchange that combines a classical and a post-quantum algorithm, such as mlkem768×25519.	Chapter Seventeen
Hypervisor	Software that runs below operating systems and isolates them from each other; Windows uses one for VBS.	Prologue

iBoot, Low-Level Bootloader (LLB)	The Apple boot stages that follow the Boot ROM and load the kernel.	Chapter Two
Identity provider (IdP)	The service, such as Microsoft Entra ID or Okta, that signs users in, holds policies and issues tokens for business systems.	Prologue
Independent BIOS vendor (IBV)	A company, such as AMI, Insyde or Phoenix, that writes most of the UEFI firmware a PC maker customizes and signs.	Prologue
Integrity Measurement Architecture (IMA)	A Linux kernel subsystem that measures files as they are opened or run and can extend those measurements into the TPM.	Appendix G
Intune	Microsoft's cloud device-management service (MDM and UEM) for Windows, Apple and Android devices.	Prologue
IOMMU, Kernel DMA Protection	The IOMMU limits which memory each device may touch; Kernel DMA Protection is the Windows feature that uses it against external peripherals.	Chapter Three
Isolate	A Ledger modifier: contain a row by naming what it is isolated from and how you prove the isolation holds.	Reader's Brief
KB article	Microsoft's Knowledge Base number for an update or support topic, such as KB5025885.	Prologue
Kernel	The part of the operating system that runs with full hardware privilege over memory, devices and every process.	Prologue
Kernel Integrity Protection (KIP)	Apple's hardware-enforced protection that locks the kernel's code in memory once the kernel finishes initializing, so it cannot be changed.	Chapter Five
Kernel lockdown	A Linux feature, merged in 5.4 (November 2019), that limits how far even root can change the running kernel.	Chapter Two
Kernel page table isolation (KPTI)	The Linux kernel's Meltdown mitigation, in Linux 4.14.11 (2 January 2018).	Chapter Three
Key Exchange Key (KEK)	The UEFI key store whose keys may update db and dbx; Microsoft's KEK CA 2011 expired on 24 June 2026.	Prologue
Known Exploited Vulnerabilities (KEV) catalog	CISA's list of vulnerabilities known to be exploited in the wild, created in November 2021.	Chapter One

L to P

Term	Meaning	First used

Least privilege	Saltzer and Schroeder's 1975 principle that every program and user should run with the least privilege the job needs.	Chapter One
Linux Security Module (LSM)	A kernel framework for access-control policies such as SELinux, AppArmor and Landlock.	Chapter Five
Livepatch	Fixing critical Linux kernel functions by redirecting calls, without a reboot, where a distribution offers it.	Chapter Nine
LocalPolicy (Full, Reduced, Permissive Security)	The per-installation boot policy on a Mac with Apple silicon, set from recoveryOS; third-party kernel extensions need Reduced Security.	Chapter Two
Lockdown Mode	An Apple setting, from iOS 16 and macOS Ventura, that narrows the attack surface for people at high risk.	Chapter Two
Lockfile	A file that pins the exact versions of every dependency a build installs.	Chapter Eight
Long-term servicing channel (LTSC)	Windows editions for fixed-function devices with long fixed support dates, such as Windows 11 IoT Enterprise LTSC 2024, supported to 10 October 2034.	Chapter Two
Machine Owner Key (MOK)	A key the owner of a Linux PC enrolls in shim so that code signed with it will boot.	Chapter Two
Mainline modules	Core Android components, from Android 10, that Google updates directly through Google Play system updates.	Chapter Two
Managed service provider (MSP)	A company that runs other companies' computers, often through an RMM tool.	Chapter Seven
Management engine	A separate processor beside the main CPU, such as Intel's ME or AMD's PSP, that runs its own firmware before and alongside the OS.	Chapter One
Measured boot	Recording a measurement of each boot stage in the TPM, so that later software or a remote verifier can ask what booted.	Prologue
Memory integrity (HVCI)	The Windows feature that moves the decision about which kernel code may run into the isolated VBS environment.	Prologue
Memory Integrity Enforcement (MIE)	Apple's always-on hardware memory tagging, announced 9 September 2025 for iPhone 17 models and iPhone Air.	Chapter Ten

Memory safety	Code never reading or writing memory it should not (past a buffer, after it was freed, through a wrong pointer type); memory-safe languages remove these bug classes in their safe code, though unsafe blocks and calls into C or C++ can bring them back.	Chapter Five
Memory Tagging Extension (MTE)	An Arm feature that tags memory and pointers so the CPU faults on mismatched accesses.	Chapter Ten
Mercenary spyware	Surveillance tools that companies build and sell to governments.	Chapter Twelve
MFA fatigue (push bombing)	Flooding a user with sign-in approval prompts until one is accepted.	Chapter Six
Microcode	The processor's own internal program, patched by the CPU vendor after the chip ships and loaded at each start from firmware or the OS.	Prologue
Mobile device management (MDM), unified endpoint management (UEM)	Products that set policy, deploy apps and scripts and can wipe devices remotely.	Chapter One
MMIO registers	Hardware registers reached through memory addresses; Operation Triangulation used undocumented ones to bypass Apple's page protection.	Chapter Ten
Model Context Protocol (MCP)	An open standard, introduced by Anthropic on 25 November 2024, for connecting AI assistants to tools and data.	Chapter Sixteen
Multi Admin Approval	An Intune rule that requires a second administrator to approve sensitive changes such as wipes, scripts and role changes.	Chapter Seven
Multi-factor authentication (MFA)	Signing in with more than one kind of proof, such as a password and a phone approval.	Chapter Six
n-day	A vulnerability exploited after its fix is published, against devices that have not applied it.	Chapter Eleven
np, nc, na	The Ledger's marks for "not published", "not collected" and "not applicable"; they are findings, not gaps to hide.	Chapter One
NPU	A neural processing unit for AI workloads, with its own driver and firmware and so its own row.	Chapter Three
OEM	Original equipment manufacturer: the company that builds and brands a device and ships its firmware, such as a PC or phone maker.	Chapter Three
Option ROM	Firmware on an add-in card, such as a graphics or network adapter, that UEFI runs at boot.	Chapter Two

Out-of-bounds read	Code reading past the end of a buffer; the Channel File 291 crash was one.	Chapter Five
Own, Rent	The Ledger's decision axis: your people hold the authority to change the row, or another party does under a contract.	Reader's Brief
Page Protection Layer (PPL), SPTM, TXM	Apple mechanisms beside the kernel that police page tables and code signing; SPTM with TXM replaces PPL on A15 and M2 or later.	Chapter Two
Passkey	A FIDO sign-in credential bound to one site or app and unlocked like the device; synced passkeys are copied across a user's devices, device-bound ones never leave one device.	Chapter Six
Patch clock (T0, T1, T2, T3, tail)	T0 the earliest of vendor aware, public or known exploited; T1 fix available for this device class; T2 and T3 fix running on 50% and 90% of your fleet; the tail is what remains.	Chapter One
Patch Tuesday	Microsoft's monthly security release, on the second Tuesday of each month since October 2003.	Chapter One
Pin	A Ledger modifier: freeze a version or configuration on purpose, always with an end date.	Reader's Brief
Pixel Bill of Materials (PBOM)	Book 1's list of what is on the box, one row per component from the processor to the lit pixel.	Chapter One
pKVM	Android's protected KVM-based hypervisor, part of the Android Virtualization Framework (Android 13), where the chip supports it.	Chapter Two
Platform Configuration Register (PCR)	A TPM register that accumulates boot measurements; PCR7 changes when Secure Boot keys change.	Chapter Four
Platform Key (PK)	The top UEFI key, normally held by the PC maker, that controls changes to the KEK store.	Prologue
Platform profile	The exact OS release, hardware class and management state a Ledger covers.	Chapter One
Play Integrity API	Google's service that tells Android apps whether a device and app look genuine, with verdicts returned from Google's servers.	Chapter Seven
Pluton	A security processor designed by Microsoft and built into chips by partners; Microsoft writes its firmware and ships it through Windows Update.	Chapter One
Post-quantum cryptography	Public-key algorithms designed to resist a large quantum computer, which would break RSA and elliptic-curve schemes.	Chapter Four

Prompt injection	Text in an agent's input written to change what the agent does.	Chapter Sixteen
Provenance	A signed record of where and how a binary was built.	Chapter Eight

Q to Z

Term	Meaning	First used
Quick Machine Recovery	A Windows 11 feature (24H2, July 2025) that lets a device that cannot boot fetch a fix from Windows Update through WinRE; off by default on Enterprise.	Chapter Five
Rapid Response Content	CrowdStrike detection content delivered from the cloud between sensor releases.	Chapter Five
Rapid Security Response	Apple's earlier form of between-release security fixes, replaced by Background Security Improvements.	Chapter Nine
RCS	Rich Communication Services: the carrier messaging standard that succeeds SMS.	Chapter Ten
Recovery clock (R0, R90)	R0 is when a failure is noticed; then stop, revoke, restore; R90 is when 90% of the fleet works again.	Chapter One
Reference monitor	Anderson's 1972 name for the core that decides access: tamper proof, always invoked, small enough to verify.	Chapter One
Remote monitoring and management (RMM)	Software that gives a technician a remote shell, script runner and file transfer on every enrolled machine.	Chapter Seven
Revocation	Telling every verifier that something it used to accept (a boot manager, a key, a certificate, a token) is no longer acceptable.	Chapter One
Rings	Groups of devices that receive a change in order, so a fault shows on a few before it reaches all.	Chapter One
Root of trust (hardware)	The component everything above it relies on because nothing below it checks it, such as immutable boot code or a security processor (a TPM, Pluton, the Secure Enclave or Titan M2); it measures, stores, reports or verifies what runs.	Chapter One
Rowhammer	Corrupting memory by repeatedly activating the same DRAM row so nearby rows flip bits.	Chapter Three
Rust	A memory-safe programming language now used in parts of the Linux and Android kernels.	Chapter Five

SBAT	Secure Boot Advanced Targeting: revocation of Linux boot components by generation number instead of by hash, first in shim 15.3 (March 2021).	Chapter Two
SBOM	Software bill of materials: a list of the components inside a product.	Chapter Eight
Secure Boot	The UEFI check that runs only boot code whose signature or hash is allowed in db and not listed in dbx.	Prologue
Secure Enclave	Apple's dedicated security subsystem with its own boot ROM, integrated into its chips.	Chapter One
Secure version number (SVN)	A counter carried in signed boot code; once the minimum is raised, older versions with a lower number are refused, which blocks rollback to flawed code.	Chapter Four
Security debt	The work owed because a component outlived its support or was never updated.	Chapter Eleven
SELinux, AppArmor	Linux mandatory access-control policies; Red Hat Enterprise Linux enforces SELinux by default, Ubuntu loads AppArmor.	Chapter One
shim	A small first-stage Linux boot loader signed under Microsoft's third-party UEFI CA, which checks the next stage against the distribution's key.	Chapter One
Side channel	A leak through timing or another physical effect rather than through a broken permission check.	Chapter Three
Signed System Volume	The cryptographically sealed macOS system volume, from macOS 11.	Chapter Two
SLSA	A framework for build provenance; version 1.0 (April 2023) states it does not address source threats such as an insider's change.	Chapter Eight
Software supply chain	The chain of authors, builders, signers and publishers through which software reaches a device.	Chapter Eight
Speculative execution	A processor running guessed instructions before it knows the guess is right; wrong guesses can leave traces in caches.	Chapter Three
Staged rollout	Releasing a change to growing shares of devices, with a chance to halt between stages.	Chapter Five
StrongBox	An Android Keystore backed by a separate secure element with its own CPU, storage and random-number generator, on some devices with Android 9 or later.	Chapter Six
Supervised device	A company-owned Apple device enrolled so that tighter MDM restrictions apply.	Chapter Two

Support period	Under the CRA, the period a manufacturer must handle vulnerabilities; at least five years unless expected use is shorter.	Chapter Eleven
Template Type, Template Instance	In CrowdStrike's sensor, predefined fields compiled into the sensor, and the criteria for one type delivered in a Channel File.	Chapter Five
Time to exploit (TTE)	Mandiant's average time taken to exploit a vulnerability before or after a patch is released.	Chapter Sixteen
Titan M2	Google's discrete security chip in Pixel phones from the Pixel 6, working with the Trusty TEE.	Chapter One
Token	A credential an app presents to prove a sign-in or a grant of rights, until it expires: often a signed statement (such as a JWT), sometimes an opaque string that only its issuer can check.	Chapter Six
Trust Ledger	This book's one-page record of the authorities, dependencies, evidence and exit paths for one critical system (Appendix B).	Reader's Brief
Trusted computing base	The set of components a device must trust to be secure.	Chapter Seven
Trusted execution environment (TEE)	An isolated area of a processor for sensitive code and keys, such as Trusty on Pixel.	Chapter Three
Trusted Platform Module (TPM)	A Trusted Computing Group standard security chip, or firmware, that holds keys and records boot measurements; TPM 2.0 is required for Windows 11.	Prologue
UEFI	The firmware specification PCs follow, maintained by the UEFI Forum, with its key stores PK, KEK, db and dbx.	Prologue
Update build revision (UBR)	The number after the build in a Windows version (for example 26100.UBR), raised by each cumulative update.	Appendix D
Verified Boot (AVB)	Android's check of each partition against the device maker's key, with rollback protection.	Chapter One
Veto point	The place where the customer can say no to an authority's change: audit, provenance check, key refusal, mirror, rings, console control.	Chapter One
VEX	Vulnerability Exploitability eXchange: a statement, delivered with an SBOM, that says whether a listed flaw actually affects the product.	Chapter Eight
Virtual desktop (VDI)	A desktop that runs on a server or cloud host and is shown on the user's device.	Chapter Three

Virtualization-based security (VBS)	An isolated environment Windows builds with its hypervisor, designed on the assumption that the kernel can be compromised.	Prologue
Vulnerable driver blocklist	Microsoft's list of signed drivers Windows refuses to load, on by default since the Windows 11 2022 update.	Chapter Two
War exclusion	An insurance clause excluding losses from hostile or warlike action, or from state-backed cyber attacks.	Chapter Twelve
Windows endpoint security platform	Microsoft's user-mode interfaces for security products, generally available on Windows 11 from 17 November 2026.	Chapter Five
Windows Hello for Business	Windows sign-in with a private key protected by the device's security module plus a PIN or biometric.	Prologue
WinPE, WinRE	The Windows Preinstallation and Recovery Environments, small Windows systems used to repair a machine that will not start.	Chapter Five
Zero-click	An attack that needs no action from the victim: the device parses a message on its own, and the parser is the target.	Chapter Ten
Zero-day	A vulnerability exploited before a fix exists.	Chapter Ten

APPENDIX B

The Trust Ledger, Blank

This is the worksheet the book fills chapter by chapter. Copy it, one per critical system. The header block is written once; the twelve rows and eight core columns follow. Eight columns do not fit across an A4 page with room to write, so the core sheet is printed in two halves with the same rows; in a spreadsheet, keep them as one sheet. Write the system name, the assessment date and the sheet version at the top of each half, so printed halves cannot be mixed up. Chapter One explains every part, Chapters Fourteen and Fifteen show filled examples, and one filled row closes this appendix.

Header: one per system

Field	Fill in
System and business service	
Platform profile (exact OS release, hardware class, management state; Chapter Two)	
Fleet denominators: devices eligible for fixes, and all devices in scope, including unsupported and not reporting (Appendix D)	
Business value at stake; cost of one day down (lost plus paid, not the day's revenue; Chapter Fourteen)	
Regulator clocks that apply, and what starts each one	
Loss scenario this Ledger protects against (priced once, here, not per row)	
Assessment date; next review	
Accountable executive	

Core sheet, first half: columns 1 to 4

Rows run from silicon (SI) to agents (AG), with EX for what sits outside the device but on the attack or recovery path. A row that does not apply gets "na" and a reason, not a blank.

Row	1 Component	2 Who can change it	3 Exposure and blast radius	4 Patch clock (T0 to T3, tail)
SI Silicon and microcode				
FW Firmware and boot				
HV Hypervisor and isolation				

KN Kernel				
DR Drivers and third-party kernel code				
SP Services and policy				
LR Libraries and runtimes				
FR The frame (Book 2)				
ID Identity and keys				
MU Management and update channels				
AG Agents				
EX External dependencies (add when an outside service can stop, authorize or recover the system)				

Core sheet, second half: columns 5 to 8

Row	5 Stop, revoke, recover	6 Who can see it (O, D, C, I, U)	7 Support end	8 Decision and owner
SI Silicon and microcode				
FW Firmware and boot				
HV Hypervisor and isolation				
KN Kernel				
DR Drivers and third-party kernel code				
SP Services and policy				
LR Libraries and runtimes				
FR The frame (Book 2)				
ID Identity and keys				
MU Management and update channels				
AG Agents				

EX External dependencies (add when an outside service can stop, authorize or recover the system)	

What goes in each column

Column	What goes in it
1 Component	The real component and version, its update policy (evergreen, pinned, tied to the OS, bundled) and how many devices run it.
2 Who can change it	The named authorities, split by power where they differ: writes, builds, signs, publishes, pushes, configures. A vendor's content channel counts.
3 Exposure and blast radius	Which untrusted inputs reach it (network, files, messages, USB, a help-desk call, a console login) or "local admin only"; its privilege; how many devices or tenants one action reaches.
4 Patch clock	T0 (earliest of vendor aware, public, known exploited), T1 (fix available for this device class), T2 and T3 (fix running on 50% and 90% of your fleet), and the tail with a count and a reason, for the last one to three critical cases. Measured on your fleet (Appendix D).
5 Stop, revoke, recover	The pause switch, the revocation path (dbx, blocklist, key rotation, token invalidation), the rollback, the date of the last drill and the measured time to R90. Add what the stop or revocation itself breaks (devices that will no longer boot, users signed out, an old version that brings a fixed flaw back) and what "works again" means for this service: a user signed in and a business transaction completed, not only a device checking in.
6 Who can see it	Telemetry, attestation or logs, or "nothing", each with an evidence tag: O observed on your fleet, D documented by the vendor, C contracted, I inferred, U unknown (Appendix G).
7 Support end	The date the owner stops fixing it, the source of that date, and the lead time to replace it (Chapter Eleven).

8 Decision and owner	Own or Rent, plus Pin (with an end date) and Isolate (from what, and the proof it holds); a one-line reason; a named person; the next review date (Chapter Fifteen).

Optional columns

Add them when a chapter or an audit needs them: root of trust (what the row verifies against), yearly cost of control, who pays on failure, AI exposure flag, link to the Pixel Bill of Materials or Frame Ledger row (Appendix C), and contract clause reference (Appendix H).

Rules for filling it

- **Twenty minutes per cell on the first pass.** A timebox, not an evidence standard: any cell you cannot fill in twenty minutes gets "np" (not published), "nc" (not collected) or "U" (unknown) for now, and goes on the list for a deeper look where the row carries high value. Those marks are the finding.
- **Clocks come from your telemetry,** not from the vendor's release date or its adoption curve (Appendix D).
- **Price scenarios, not rows.** Write each loss once in the header and list the rows it crosses, so one event is not counted several times (Chapter Fourteen).
- **Every Pin has an end date; every Isolate names its boundary and its test.**
- **One owner per row,** a person, not a team, with a review date.

One filled row

This is the SI row of Corvane Hotels' kiosk Ledger, as Chapter Three fills it, set out column by column so the density and the marks show. Note how much of it is a mark rather than a number: that is a normal first Ledger.

Column	Corvane kiosk, SI row (Chapter Three)
1 Component	Kiosk x86 CPU and microcode; one OEM model in 2 BIOS versions
2 Who can change it	CPU vendor writes; OEM ships in BIOS; Microsoft ships via Windows Update
3 Exposure and blast radius	Local code only; one flaw reaches all 3,000 kiosks
4 Patch clock	Last case to measure: Downfall (T0 public 8 Aug 2023); T2, T3 nc
5 Stop, revoke, recover	Microcode is reloaded each boot; a bad revision is replaced by shipping an older or newer one (the January 2018 pattern)
6 Who can see it	U: effective revision not collected
7 Support end	OEM hardware support date, from the purchase contract
8 Decision and owner	Rent, Pin the hardware model to contract end; owner: end-user computing lead; review quarterly

APPENDIX C

The Five-OS Trust Reference Table

Every row of the Trust Ledger on Windows, macOS, iOS, Linux and Android, collected from the chapters' side-by-side tables. Each cell says what its chapter says, for the platform profiles fixed in Chapter Two; the chapter tells the story and gives the sources. Use it to fill column 2 (who can change it) and column 5 (stop, revoke, recover) of a first Ledger. Facts are as of November 2026.

How to read it

- **Platform profiles.** Windows 11 Enterprise 24H2 or later (Windows 11 IoT Enterprise LTSC 2024 for kiosks); macOS 27 on Apple silicon, MDM-enrolled; iOS 27, supervised when company-owned; Linux enterprise (RHEL 10 or Ubuntu 24.04 LTS, Secure Boot through shim) and community; Android 17 on Pixel, large-maker flagship and low-cost device.
- **"Maker" on Android** means the device maker, which holds the boot keys and ships the image. On Linux, "you" means the owner may hold every key and so owns the clock too.
- **Capability or default.** A cell names what the platform offers for the profile. Where a control works only once someone turns it on, the cell says so ("where enabled", "if configured", "by policy"); preview features are marked experimental. Record in your Ledger which state you have observed.
- **Where a cell differs from its chapter's table,** this appendix carries the later correction, with its source in Appendix E.

Below the operating system (SI, FW, HV)

Row	Windows	macOS	iOS	Linux	Android
SI Root of trust	TPM 2.0 (required for Windows 11); Pluton on supported chips, firmware by Microsoft through Windows Update	Secure Enclave with its own boot ROM	Boot ROM plus Secure Enclave	TPM 2.0 where the hardware has one	Varies by maker; Pixel: Titan M2 and Trusty TEE
SI Microcode path	OEM BIOS, or Windows Update; the newer one wins at boot	Apple, with the OS	Apple, with the OS	OEM BIOS, or the kernel's early loader from the distribution's package	No field-loaded microcode as on x86; CPU and SoC fixes ride in the maker's firmware image (FW row)
SI Running revision visible?	Not in most fleet reports; collect it	Through the OS build	Through the OS build	`/proc/cpuinfo`; `old_microcode` flag for Intel	na as microcode; the security patch level is the maker's claim of what the image contains
FW Who holds the boot keys	PC maker (PK); Microsoft (KEK, db entries)	Apple (boot ROM key); owner sets LocalPolicy	Apple	PC maker (PK); Microsoft (KEK, third-party CA); distribution (shim's key); owner may add MOKs	Maker; Pixel adds a user-settable root of trust
FW What the customer can change	Secure Boot on or off; custom PK and db on many PCs; apply dbx and 2023 certificates on Microsoft's schedule	Full, Reduced or Permissive Security, from recoveryOS	No boot policy choice; a DFU restore reinstalls Apple-signed software only	As Windows, plus Machine Owner Keys in shim	Lock or unlock (wipes data) where allowed
FW Revocation path	dbx and boot manager SVN, staged by Microsoft (KB5025885); not reversible with Secure Boot on	Apple signing; personalized signatures in Full Security	Apple signing	dbx and SBAT generations	AVB rollback index
FW 2026 certificate exposure	KEK, UEFI CA and Windows Production PCA 2011 all expired (24 Jun, 27 Jun, 19 Oct 2026)	Not affected	Not affected	UEFI CA 2011 (shim) and KEK 2011	Not affected
FW How firmware arrives	OEM tool or Windows Update (UEFI capsules)	Inside Apple's OS updates	Inside Apple's OS updates	fwupd and LVFS where the OEM publishes; otherwise the OEM tool	The maker's over-the-air image

HV Layer below the kernel	VBS and memory integrity, where hardware, drivers and policy allow	Kernel Integrity Protection; SPTM and TXM on M2 or later	PPL on older chips; SPTM and TXM on A15 or later	Usually none; defenses are in-kernel LSMs and lockdown	pKVM where the chip vendor supports it

The operating system (KN, DR, SP)

Row	Windows	macOS	iOS	Linux	Android
KN Who signs the kernel	Microsoft	Apple	Apple	The distribution, or you	Google releases the Generic Kernel Image with a Google-inserted certificate; the maker adds its vendor modules and signs the device image, and its Verified Boot key roots the chain
KN Who ships the OS fix	Microsoft	Apple	Apple	The distribution, or you	Google publishes a monthly bulletin; each maker ships
DR Third-party kernel code	Microsoft-signed drivers; vulnerable driver blocklist on by default; EDR sensors and their content	Kexts only after Reduced Security with admin approval; tools in user space	None	Signed modules when Secure Boot is on; out-of-tree modules; eBPF programs	Chip vendor and maker modules
DR Where an EDR sensor runs	Kernel driver today; user-mode platform GA on Windows 11 from 17 Nov 2026, vendor dates vary	System extension in user space	No kernel sensor; apps and MDM	Kernel module or eBPF	No kernel sensor; apps and EMM
DR Who can stop a bad push	The vendor; you, only if contract and console give you content rings	The vendor; you, through MDM approval of extensions	Apple	Your distribution and you, through package pinning	Google and the maker
SP Decide what apps may run	App Control for Business	MDM and Gatekeeper policy	MDM restrictions; Apple reviews and signs App Store apps, and EU alternative distribution is notarized	Your repositories and policy; fapolicyd allowlisting on RHEL. SELinux (RHEL) and AppArmor (Ubuntu) confine what a program may do once it runs	EMM policy; sideloading block on managed devices
SP Who must vouch for an app	No signature needed to run; Authenticode and SmartScreen shape warnings	Developer ID signature and notarization, checked by Gatekeeper	Apple signs App Store apps; EU alternative distribution still notarized	The distribution signs its repository; each added repository brings its own key	The developer signs; with Play App Signing, Google holds the key for Play delivery

Above the kernel (LR, FR, ID)

Row	Windows	macOS	iOS	Linux	Android
LR Where bundled libraries hide	Inside each installer and app folder	Inside each app bundle	Inside each app, statically linked	Shared system packages, plus containers, Snaps, Flatpaks and vendor tarballs	Inside each APK
LR Memory-safety share of vulnerabilities (vendor's own figure)	About 70% of Microsoft's CVEs per year (2019)	Not published; Apple publishes design goals (MIE, 2025)		No single owner; per subsystem	Below 20% of Android vulnerabilities in 2025 (Google)
LR Memory-safe code and hardware tagging	Share of memory-safe code not published (np)	Share not published (np)	Share not published (np); MIE on iPhone 17 models and iPhone Air	Rust support in the kernel since 6.1; drivers by choice	New platform code largely in Rust; first production Rust kernel driver in the 6.12 kernel
FR Who parses messages before you click	Each messaging and mail app, and the preview handlers it calls	Messages and its parsing services, plus each app	Messages through BlastDoor (iOS 14 on), plus each app	Each app and the libraries the distribution ships	The SMS and RCS app, each messaging app, the media stack
FR One switch for at-risk users	No single named mode	Lockdown Mode	Lockdown Mode	No single mode	Advanced Protection (Android 16)
ID Where device keys live	TPM 2.0; Windows Hello for Business prefers the TPM and falls back to software keys only when no usable TPM exists and the "Use a hardware security device" policy is off	Secure Enclave	Secure Enclave	TPM where present; depends on distribution and tooling	TEE-backed Keystore; StrongBox on some devices with Android 9 or later
ID Disk key escrow for recovery	BitLocker key in Entra ID or AD DS	FileVault personal recovery key escrowed to MDM	Hardware-bound; not a key the company escrows	No built-in escrow: store a LUKS recovery key yourself; Clevis with Tang servers or a TPM2 binding (systemd-cryptenroll) unlocks without one	Hardware-bound; not a key the company escrows

Control planes and the future rows (MU, AG)

Row	Windows	macOS	iOS	Linux	Android
MU Built-in management channel	The OS's MDM interface (Intune and other UEM); Group Policy and Configuration Manager	Apple's MDM protocol; supervision through Automated Device Enrollment	Apple's MDM protocol; supervision through ADE or Apple Configurator	No MDM protocol in the OS; vendor fleet tools are optional (Red Hat Satellite and Insights, Canonical Landscape), plus configuration tools and repositories	Android Enterprise through an EMM
MU Delay OS updates	Defer quality updates up to 30 days, pause up to 35; feature updates up to 365 days	1 to 90 days, with a deadline	1 to 90 days, with a deadline	As long as you choose: you run the repositories	Postpone once for 30 days, or freeze up to 90 days
MU Fix without a full OS update	Hotpatch (Enterprise 24H2): eight months a year without restart	Background Security Improvements (26.1 on)	Background Security Improvements (26.1 on)	Kernel livepatch where the distribution offers it	Mainline modules through Google Play system updates
MU Roll back a shipped change	Uninstall the update; Quick Machine Recovery from WinRE where enabled	Remove all Background Security Improvements at once	Remove all Background Security Improvements at once	Package downgrade; image rollback on image-based systems	A/B fallback if the new slot fails to boot
MU Two-person rule for mass actions	Intune Multi Admin Approval, if configured	Depends on the MDM product	Depends on the MDM product	Depends on your tooling	Depends on the EMM product
MU Support end published?	Yes, per edition; LTSC and IoT LTSC run longest	No; inferred from history (tag I)	No; inferred from history (tag I)	Kernel: projected per longterm release; distributions publish their own	Per maker and model; absent for many devices
AG What an agent runs as	A separate agent account in its own session, after an admin turns the setting on (experimental)	The host app, under the user's permissions (App Intents)	The host app, inside its sandbox (App Intents)	The user or service account that starts it	A caller holding EXECUTE_APP_FUNCTIONS (AppFunctions, experimental)
AG What it can reach by default	On request, the six known folders, with consent per agent	What the user grants the host app	Only what each app's intents expose	Everything that account can reach	Only the functions each app declares

Sources: the side-by-side tables of Chapters One to Seventeen and their source notes in Appendix E. Preview features (AG rows) change often; record the build you checked.

Row mapping across the series

Book 1's Pixel Bill of Materials lists what is on the box by layer; Book 2's Frame Ledger follows one screen from hop 0 (source) to hop 10 (glass and readers), plus an input row and a capture row. The Trust Ledger keeps the same stack and asks who controls each part. Link rows; do not copy them.

Trust Ledger row	Book 1: Pixel Bill of Materials layer	Book 2: Frame Ledger hops
SI Silicon and microcode	Silicon and boot (Book 1, Chapter 1)	None directly; the GPU and display hardware under hops 6 (raster) and 9 (scanout)
FW Firmware and boot	Silicon and boot (Book 1, Chapter 1)	None
HV Hypervisor and isolation	No separate layer; Silicon and boot and The kernel (Book 1, Chapters 1 and 2)	None
KN Kernel	The kernel (Book 1, Chapter 2)	None directly; kernel display drivers under hop 9
DR Drivers and third-party kernel code	Drivers, storage and network (Book 1, Chapter 3)	Hop 9 (display driver and scanout)
SP Services and policy	Security, identity and the trusted pixel (Book 1, Chapter 4)	The "process and sandbox" column of every hop
LR Libraries and runtimes	Runtimes and languages; Toolchain and distribution (Book 1, Chapters 5 and 8)	The libraries inside hops 1 to 6 (parsers, shaping, font and image libraries, the 2D graphics library)
FR The frame	Graphics and text; Browser engine; Pixel (Book 1, Chapters 6 and 7)	Hops 0 to 10, plus the input and capture rows
ID Identity and keys	Security, identity and the trusted pixel (Book 1, Chapter 4)	None directly; the input row records who touches the glass
MU Management and update channels	Toolchain and distribution (Book 1, Chapter 8)	The "who patches, channel, typical delay" column of every hop

AG Agents	AI moves the seams (Book 1, Chapter 13)	Hop 10 (glass and readers: the accessibility tree agents read)
EX External dependencies	None	None

APPENDIX D

Measuring the Patch and Recovery Clocks

Column 4 and column 5 of the Trust Ledger hold two clocks that only your own telemetry can measure. This appendix gives the definitions from Chapter One, a recipe per platform that a fleet team can run this week, and the rules for marking what nobody published. Report names, tables and fields are as Microsoft, Apple, Jamf, Google and the Linux projects document them in November 2026; check them against the current documentation before you build on them.

The definitions

Mark	Definition	Who owns it
T0	The earliest of three moments: the vendor learns of the flaw, the flaw becomes public, or it is known to be exploited. Record the three separately, each with its source; where one is not public, mark it np and take T0 from the earliest known one, saying so.	Nobody; it is a fact to record
T1	The day a fix is available for this device class, which may be weeks after the first fix for another class.	The vendor (and, for firmware and Android, the device maker)
T2	The day the fix is running on 50% of your eligible devices.	You
T3	The day the fix is running on 90% of your eligible devices.	You
Tail	The devices still exposed after T3, counted, with a reason for each group: not reporting, failed install, waiting for restart, unsupported.	You
R0	The moment a failure or an exploitation is noticed.	You
Stop, revoke, restore	The three recovery actions in order: stop the channel, ring or content; revoke what must no longer be trusted; restore by rollback, repair or rebuild.	You and the channel's owner
R90	The moment 90% of the affected fleet works again for the people who use it, reported to the day.	You

Three rules from Chapters One and Nine apply to every recipe. **Two denominators:** T2 and T3 are shares of the eligible population, the devices that should receive this fix; report beside them the exposure across all devices in scope, so unsupported devices (a Chapter Eleven finding), devices offline by design such as seasonal kiosks, and devices not reporting stay visible in the tail instead of dropping out of the count. **Installed is not running:** count a fix only after the restart or process restart that activates it. **Not reporting is unknown,** not patched and not unpatched; it is the tail.

Read each date from the device's own timestamps, not from the day a report showed it: cloud reports refresh on their own schedules, stated in their documentation, and a curve read off a stale report runs late.

Five numbers per fix

For each of the last one to three critical cases on a row, record: the eligible population; the share running the fix at 24 hours, 7 days and 30 days after T1; the share installed but waiting for a restart; failed installs with the error; and devices not reporting. T2 and T3 are read off that curve (Figure 9.1).

Windows: Intune and Windows Update for Business reports

- **T1.** Microsoft's "Windows 11 release information" page lists each update with its availability date, build and KB article. For monthly updates, T1 is normally the second Tuesday.
- **Per-device state.** Windows Update for Business reports store data in your Azure Monitor Log Analytics workspace. Table `UCClientUpdateStatus` has one record per device per update, with `TargetKBNumber`, `UpdateReleaseTime`, `UpdateInstalledTime`, `RestartRequiredTime` and the current `ClientSubstate`. Table `UCClient` has one record per device, with the running `OSBuild`, `OSSecurityUpdateStatus` and `LastWUScanTime`.
- **Recipe.** Take the eligible devices from `UCClient` and keep a daily snapshot of each device's running `OSBuild`, because the tables hold current state and an install time, not the day the fix started running. For one KB, take each device's `UpdateInstalledTime` from `UCClientUpdateStatus`; where `RestartRequiredTime` is set and the running `OSBuild` has not moved, count the device as waiting for restart. A device counts as running the fix from the first snapshot whose build is at or above the fixed one; the day that count crosses 50% and 90% of eligible devices is T2 and T3. Devices whose `LastWUScanTime` is older than your reporting window are the not-reporting share.

- **Without Log Analytics.** The Intune admin center's Windows Update Distribution Report (Reports, Windows Updates) shows devices on each quality update or later, with release date and last check-in, exported as CSV. Microsoft notes the report is cached for three days, so generate it again before you read a date off it.

Windows: WSUS and the device itself

- **WSUS.** The console's Update Status Summary and Computer Status Summary reports show what each client reported at its last contact. For a curve, query the WSUS database (`SUSDB`) through its public views, such as `PUBLIC_VIEWS.vComputerTarget` and `vUpdateInstallationInfoBasic`, whose state values distinguish installed, installed pending reboot, failed and not installed. Microsoft's documentation for these views dates from the WSUS 3.0 era; test your queries. Microsoft deprecated WSUS in September 2024: no new features, but it still publishes updates through it.
- **On the device.** The running build revision is the `UBR` value under `HKLM\SOFTWARE\Microsoft\Windows NT\CurrentVersion`, read with `CurrentBuildNumber`; each cumulative update raises it. `Get-HotFix` returns an `InstalledOn` date, but only for updates installed through component-based servicing.

On one Windows device, in PowerShell: the running build revision, and the newest update that component-based servicing recorded.

```
(Get-ItemProperty 'HKLM:\SOFTWARE\Microsoft\Windows
NT\CurrentVersion').UBR
(Get-HotFix | Sort-Object -Property InstalledOn)[-1]
```

macOS and iOS: MDM, declarative status and Jamf

- **T1.** Apple's security releases page lists each release with its date. Apple does not disclose issues until a fix is generally available, and Chapter Ten showed that an entry can be added months after release. When that happens, the note's revision date is the public date, not the vendor's awareness: record vendor awareness as np, and measure only the intervals whose two ends are known.
- **Per-device state.** The MDM `DeviceInformation` query returns `OSVersion`, `BuildVersion` and `SupplementalBuildVersion`, which shows an installed Background Security Improvement. Declarative device management reports the same through status items such as `device.operating-system.build-version`, `device.operating-system.supplemental.build-version`, `softwareupdate.install-state` and `softwareupdate.failure-reason`.

- **Jamf Pro.** `GET /api/v4/computers-inventory` with the GENERAL and OPERATING_SYSTEM sections returns `operatingSystem.build`, `operatingSystem.supplementalBuildVersion`, `general.reportDate` and `general.lastContact` per Mac (the v1 endpoint was deprecated on 30 June 2025); `GET /api/v2/mobile-devices/detail` filters iPhones and iPads by `osBuild` and `osSupplementalBuildVersion`.
- **Recipe.** Pull inventory daily and keep each snapshot. A device counts as running the fix from the first snapshot where its build (or supplemental build) is at or above the fixed one; on Apple platforms the build changes only after the restart, so installed and running coincide. Devices whose last contact is older than your window are the not-reporting share.

Android: the Android Management API or your EMM

- **T1 is two dates.** Google's bulletin overview lists each monthly bulletin with its publication date and patch levels: that is Google's T1. Each device maker ships on its own clock, and makers and chip vendors publish their own bulletins. T1 for your device class is the maker's release for your model; if the maker publishes no date, mark it np.
- **Per-device state.** The Android Management API's device resource (`enterprises.devices.list`) carries `softwareInfo.securityPatchLevel`, `softwareInfo.androidBuildNumber`, `softwareInfo.systemUpdateInfo.updateReceivedTime` and `lastStatusReportTime`. Software information is filled only when the device's policy enables software reporting; most EMM products expose the same patch level in their own reports.
- **Recipe.** Snapshot daily. Count a device as fixed when its security patch level is at or after the bulletin's level; that is the maker's claim about the image, good compliance evidence but not proof of each fix, and a maker can publish its own exceptions. Read T2 and T3 from the snapshots. The gap between Google's bulletin date and `updateReceivedTime` on your devices is everything after Google: the chip vendor, the maker, any carrier and your own EMM deferral together. Split it only where each party publishes its date.

Linux: package timestamps

- **T1.** The distribution's advisory: Ubuntu Security Notices, Red Hat security advisories (RHSA, also published as machine-readable CSAF files), or the Debian security announcements.

- **Install time.** On RPM systems, `rpm -qa --last` lists packages newest first, and the query format `%{INSTALLTIME:date}` gives one package's install time. On Debian and Ubuntu, `/var/log/dpkg.log` records each install and upgrade with date, time, package and version, and `/var/log/apt/history.log` records each apt run.
- **Running, not installed.** Compare `uname -r` with the newest installed kernel. On Red Hat family systems, `needs-restarting -r` exits with 1 when a reboot is required; on Ubuntu, the file `/var/run/reboot-required` is a hint that one is.

On one Linux host, in a shell: one package's install time on RPM systems, its upgrades on Debian and Ubuntu, and whether a reboot is still needed.

```
rpm -q --qf '%{NAME} %{VERSION}-%{RELEASE} %{INSTALLTIME:date}\n'
openssl
zgrep -h " upgrade " /var/log/dpkg.log* | grep " openssl"
uname -r; needs-restarting -r; echo $?
```

Collect the output from every host through your configuration tool, keyed by host and date, and build the same curve as on the other platforms.

Below the operating system: firmware, Secure Boot and livepatch

The SI, FW and DR rows have clocks too, and the OS build does not show them. Collect these per device, keyed by host and date, and build the same curve.

- **Firmware version.** Windows: `SMBIOSBIOSVersion` from the `Win32_BIOS` class. Linux: `dmidecode -s bios-version`, and `fwupdmgr get-devices` for every device fwupd manages. Apple and Android: inside the OS build.
- **Running microcode.** Linux: the `microcode` field in `/proc/cpuinfo`. Windows has no documented fleet report for it; collect it with your inventory or EDR tool, or mark it nc (Chapter Three).
- **Secure Boot certificates.** Windows: the `UEFICA2023Status` value under `HKLM\SYSTEM\CurrentControlSet\Control\SecureBoot\Servicing`, and events 1801 (the new certificates are not yet applied) and 1808 (they are in the firmware), as Microsoft's guidance for IT professionals documents (Chapter Four).
- **Kernel livepatch.** `kpatch list` on Red Hat systems and `canonical-livepatch status` on Ubuntu show which patches are loaded in the running kernel.
- **EDR sensor and content version.** Each vendor's console shows them in its own way; if yours does not export them per device, mark the cell nc and ask for it in the contract (Appendix H).

The recovery clock

- **R0** is the first alert or ticket that named the failure; take it from the incident record, not from memory.
- **Stop** is the time from R0 until no further device received the change: the pause, the halted rollout or the vendor's revert. Measure it in a drill for every push channel (Chapter Nine).
- **Restore throughput.** Where recovery needs a disk key, the key reads are the measurement. Microsoft Entra ID's audit log records each "Read BitLocker key" activity (category KeyManagement). Pair each read with the device's next check-in (`LastWUScanTime`, `lastStatusReportTime` or Jamf's `general.lastContact`) to get time per device; that pairing is this book's method, not a vendor feature.
- **R90** is the day 90% of the affected devices work again for their users. Define "works" per service before the incident: the device boots, a user signs in, the critical application starts, and a test transaction completes. A check-in is supporting evidence, not proof. A vendor's sensor-connectivity figure is the vendor's curve, labeled as such (Chapter Five); measure your own.

How to mark what is missing

Mark	Use it when	Example
np	Not published: the authority holds the date but has not made it public.	A tablet maker's release date for the monthly patch; the BlackLotus enforcement date in KB5025885
nc	Not collected: the data exists in your reach but you do not gather it yet.	Effective microcode revision per kiosk; EDR content version per device
na	Not applicable to this row, with a one-line reason.	No agents on the kiosk image
U	Unknown: nobody can say, including the vendor.	Who can see a firmware parser flaw before an advisory

In every timeline in this book, an np interval is drawn as a labeled gap, never as a bar, because a bar implies a measured duration. Use the same rule in your own charts. The fill rule from Chapter One holds here: a cell you cannot fill in twenty minutes on the first pass gets one of these marks (Appendix B).

APPENDIX F

Case File Index

Every case the book tells, in the order it first appears, with the Trust Ledger rows it crossed, the identifiers the chapter names, the key dates and one source: the advisory, record or original research write-up the chapter relied on. The chapter listed first is the case's narrative home; the others give it a sentence and a pointer. Each chapter's full source list is in Appendix E. Where a chapter names no CVE, this index does not add one.

How to use it

- **Find your rows.** Look in the second column for the rows on your Ledger that carry U or np marks, and read the cases that crossed them.
- **Find your clocks.** The key dates give T0 and T1 where they are public; T2 and T3 are yours to measure (Appendix D).
- **Abbreviations.** ED is a CISA Emergency Directive, KEV its Known Exploited Vulnerabilities catalog, VU# a CERT/CC vulnerability note (Appendix A).
- **Damage figures are not here.** Each one belongs to its speaker and stays in its chapter.

The index

Case	Rows crossed	Identifiers	Key dates	Chapter	Source
Spectre and Meltdown	SI, HV, KN	CVE-2017-5753, CVE-2017-5715, CVE-2017-5754	Reported 1 Jun 2017; public 3 Jan 2018 (planned 9 Jan); Intel told customers to stop deploying its microcode 22 Jan 2018	Three	`https://googleprojectzero.blogspot.com/2018/01/reading-privileged-memory-with-side.html`
Downfall	SI	CVE-2022-40982	Reported 24 Aug 2022; INTEL-SA-00828 8 Aug 2023	Three	`https://www.intel.com/content/www/us/en/security-center/advisory/intel-sa-00828.html`

Zenbleed	SI, FW	CVE-2023-20593	Reported 15 May 2023; AMD-SB-7008 24 Jul 2023; client firmware to OEMs 21 Nov 2023 to 20 Mar 2024, embedded 15 Apr 2024	Three	https://lock.cmpxchg8b.com/zenbleed.html
GoFetch	SI, LR	None named in the chapter	Disclosed to Apple 5 Dec 2023; public March 2024	Three	https://gofetch.fail/
Rowhammer and Phoenix	SI, FW	CVE-2025-6202 (Phoenix)	ISCA 2014 paper; Phoenix published September 2025	Three	https://comsec.ethz.ch/research/dram/phoenix/
Thunderspy	SI, FW	None named in the chapter	Published 2020; no software fix for affected designs	Three	https://thunderspy.io/
Intel Management Engine flaws	SI, FW	INTEL-SA-00086	Advisory 20 Nov 2017; fix through each system maker	Three	https://www.intel.com/content/www/us/en/security-center/advisory/intel-sa-00086.html
AMI MegaRAC BMC authentication bypass	SI, FW, EX	CVE-2024-54085	Added to CISA KEV 25 Jun 2025	Three	https://www.cisa.gov/known-exploited-vulnerabilities-catalog
Microsoft 2011 Secure Boot certificate expiry	FW, ID	None (certificate expiry)	KEK CA 2011 24 Jun 2026; UEFI CA 2011 27 Jun 2026; Windows Production PCA 2011 19 Oct 2026	Four	https://support.microsoft.com/en-us/topic/windows-secure-boot-certificate-expiration-and-ca-updates-7ff40d33-95dc-4c3c-8725-a9b95457578e
BlackLotus	FW, HV, KN	CVE-2022-21894, CVE-2023-24932	Patch 11 Jan 2022; ESET analysis 1 Mar 2023; KB5025885 from 9 May 2023; enforcement date np as of 30 Sep 2026	Four, Nine	https://msrc.microsoft.com/update-guide/vulnerability/CVE-2023-24932
LogoFAIL	FW	VU#811862	Announced 29 Nov 2023; details and CERT/CC note 6 Dec 2023; fix per OEM firmware	Four	https://kb.cert.org/vuls/id/811862
PKfail	FW, ID	VU#455367	Binarly report 25 Jul 2024; CERT/CC note 30 Aug 2024	Four	https://kb.cert.org/vuls/id/455367

BootHole	FW	CVE-2020-10713	Published 29 Jul 2020; SBAT first shipped in shim 15.3, March 2021	Four	https://eclypsium.com/blog/theres-a-hole-in-the-boot/
CrowdStrike Channel File 291	MU, DR, KN, ID	None (not an exploited vulnerability)	Content published 19 Jul 2024 04:09 UTC, reverted 05:27 UTC; PIR 24 Jul 2024; RCA 6 Aug 2024; Delta suit filed 25 Oct 2024	Five, Thirteen	https://www.crowdstrike.com/wp-content/uploads/2024/08/Channel-File-291-Incident-Root-Cause-Analysis-08.06.2024.pdf
Stuxnet	DR, ID, KN, EX	Microsoft Security Advisory 2286198; four Windows flaws, not itemized	Earliest sample June 2009; reported 17 Jun 2010; certificates revoked 16 and 22 Jul 2010	Thirteen, Five	https://docs.broadcom.com/doc/security-response-w32-stuxnet-dossier-11-en
Bring your own vulnerable driver (Dell dbutil_2_3.sys)	DR, KN	CVE-2021-21551	Abused autumn 2021; ESET report 30 Sep 2022	Five	https://www.welivesecurity.com/2022/09/30/amazon-themed-campaigns-lazarus-netherlands-belgium/
Storm-0558	ID	None (a key and a validation gap)	Access from 15 May 2023; detected 15 and 16 Jun 2023; MSRC 11 Jul 2023; CSRB report dated 20 Mar 2024, released 2 Apr 2024	Six, Nine	https://www.cisa.gov/sites/default/files/2025-03/CSRBReviewOfTheSummer2023MEOIntrusion508.pdf
Uber contractor account	ID	None	September 2022; update revised through 19 Sep 2022	Six	https://www.uber.com/newsroom/security-update/
MGM Resorts	ID	None	September 2023; Form 8-K 5 Oct 2023	Six	https://www.sec.gov/Archives/edgar/data/789570/000119312523251667/d461062d8k.htm
Okta support system	ID	None	28 Sep to 17 Oct 2023; reported 20 Oct; root cause 3 Nov 2023	Six	https://sec.okta.com/articles/2023/11/unauthorized-access-oktas-support-case-management-system-root-cause
Kaseya VSA	MU, EX	CVE-2021-30116	Reported 6 Apr 2021; SaaS fix 26 Jun; attack 2 Jul; on-premises fix 11 Jul 2021	Seven	https://csirt.divd.nl/cases/DIVD-2021-00011/

ConnectWise ScreenConnect	MU	CVE-2024-1709	Reported 13 Feb 2024; bulletin 19 Feb; CISA KEV 22 Feb 2024	Seven	https://www.connectwise.com/company/trust/security-bulletins/connectwise-screenconnect-23.9.8
Stryker	MU, ID	None in the public record	Incident 11 Mar 2026; CISA alert 18 Mar; Form 8-K/A 9 Apr 2026	Seven	https://www.cisa.gov/news-events/alerts/2026/03/18/cisa-urges-endpoint-management-system-hardening-after-cyberattack-against-us-organization
Kaspersky Final Determination	MU	None	Announced 20 Jun 2024; updates barred from 29 Sep 2024	Seven	https://www.federalregister.gov/documents/2024/06/24/2024-13532/final-determination-case-no-icts-2021-002-kaspersky-lab-inc
SolarWinds SUNBURST	LR, MU, ID	None named in the chapter	Build compromise from Sep 2019; injection from 20 Feb 2020; ED 21-01 13 Dec 2020; Form 8-K 14 Dec 2020	Eight	https://www.sec.gov/Archives/edgar/data/1739942/000162828020017451/swi-20201214.htm
xz-utils backdoor	LR, MU	CVE-2024-3094	First patch to the mailing list 29 Oct 2021; first commit by the account merged 7 Feb 2022; 5.6.0 released 24 Feb 2024; found 29 Mar 2024	Eight	https://www.openwall.com/lists/oss-security/2024/03/29/4
Log4Shell	LR	CVE-2021-44228	Reported 24 Nov 2021; 2.15.0 public 10 Dec 2021; ED 22-02 17 Dec 2021	Eight	https://www.cisa.gov/sites/default/files/2023-02/CSRB-Report-on-Log4j-PublicReport-July-11-2022-508-Compliant.pdf
3CX desktop app	LR, MU, ID	None named in the chapter	Reported 29 Mar 2023; Mandiant findings 20 Apr 2023	Eight	https://cloud.google.com/blog/topics/threat-intelligence/3cx-software-supply-chain-compromise
Shai-Hulud npm worm	LR, ID	None	CISA alert 23 Sep 2025	Eight	https://www.cisa.gov/news-events/alerts/2025/09/23/widespread-supply-chain-compromise-impacting-npm-ecosystem

CircleCI secrets rotation	ID, MU	None	Disclosed 4 Jan 2023; tokens rotated by 7 Jan 2023	Nine	https://circleci.com/blog/january-4-2023-security-alert/
Apple Rapid Security Response 16.5.1 (a) and (c)	MU, FR	None named in the chapter	Shipped 10 Jul 2023; replaced 12 Jul 2023	Nine	https://support.apple.com/en-us/106354
Operation Triangulation	FR, LR, KN, SI, HV	CVE-2023-41990, CVE-2023-32435, CVE-2023-32434, CVE-2023-38606	Fixes 23 Jan, 27 Mar, 21 Jun, 24 Jul 2023; first report 1 Jun 2023; full chain 27 Dec 2023	Ten	https://securelist.com/operation-triangulation-the-last-hardware-mystery/111669/
FORCEDENTRY sandbox escape	FR, SP	None assigned for the escape	Citizen Lab 13 Sep 2021; Project Zero 31 Mar 2022	Ten, Thirteen	https://projectzero.google/2022/03/forcedentry-sandbox-escape.html
Trident	FR, KN	CVE-2016-4655, CVE-2016-4656, CVE-2016-4657	Citizen Lab 24 Aug 2016; iOS 9.3.5 25 Aug 2016	Ten, Thirteen	https://citizenlab.ca/2016/08/million-dollar-dissident-iphone-zero-day-nso-group-uae/
Edge appliances (Citrix, Fortinet, Ivanti, Cisco)	EX	CVE-2023-4966, CVE-2024-21762, CVE-2023-46805, CVE-2024-21887	KEV 18 Oct 2023, 9 Feb 2024, 10 Jan 2024; ED 24-01 19 Jan 2024; ED 25-03 25 Sep 2025	Eleven	https://www.cisa.gov/news-events/directives/ed-24-01-mitigate-ivanti-connect-secure-and-ivanti-policy-secure-vulnerabilities
Heartbleed	LR	CVE-2014-0160	OpenSSL advisory 7 Apr 2014	Twelve	https://www.openssl.org/news/secadv/20140407.txt
NotPetya	MU, KN, ID	None named in the chapter	27 Jun 2017; attributed 15 Feb 2018; indictment announced 19 Oct 2020; Merck appeal ruling 1 May 2023	Thirteen, Twelve	https://www.cisa.gov/news-events/alerts/2017/07/01/petya-ransomware
Pegasus record (BLASTPASS; WhatsApp v. NSO)	FR, KN	None listed in this book	Entity List 4 Nov 2021; BLASTPASS 7 Sep 2023; liability ruling 20 Dec 2024; verdict 6 May 2025; injunction 17 Oct 2025	Thirteen	https://citizenlab.ca/2023/09/blastpass-nso-group-iphone-zero-click-zero-day-exploit-captured-in-the-wild/
GTG-1002, AI-orchestrated espionage	AG, ID	None	Detected mid-September 2025; report 13 Nov 2025	Sixteen	https://assets.anthropic.com/m/ec212e6566a0d47/original/Disrupting-the-first-reported-AI-orchestrated-cyber-espionage-campaign.pdf

Big Sleep and SQLite	LR	CVE-2025-6965	First find 1 Nov 2024; CVE-2025-6965 announced 15 Jul 2025	Sixteen	https://blog.google/technology/safety-security/cybersecurity-updates-summer-2025/

APPENDIX G

The Visibility Map

This book tracks five events that change who controls a system (a help-desk reset or a firmware setting change can too): kernel code loads, firmware changes, a token is issued, a management channel pushes a policy, and an agent acts. This map shows which record each platform keeps for each event, and who can read it. Use it to fill column 6 of the Trust Ledger, "Who can see it". Mechanisms are as of November 2026, checked against vendor documentation and kernel sources; full URLs are in Appendix E.

How to read it

- **A cell names a record, not a guarantee.** Many records exist only after someone turns them on: a Sysmon install, an auditd rule, Android security logging, a log export. Until you have done that, the cell is a capability, not evidence.
- **The best record is often off the device.** Sign-ins, management actions and API calls are kept by the identity provider, the MDM or the cloud service. Their retention and price are set by your license (Chapter Six).
- **On Android, `logcat` is not a fleet tool.** Google marks the `READ_LOGS` permission "not for use by third-party applications". Fleet evidence comes from the security and network logs that the device owner app (your EMM) collects.

Kernel code, firmware and boot (events 1 and 2)

Event	Windows	macOS	iOS	Linux	Android
1 Driver or kernel code load	`CodeIntegrity/Operational` log: unsigned driver attempts (3001), App Control blocks (3077) and audit-mode would-blocks (3076). Sysmon event 6, "Driver loaded", with hash and signature, once you install Sysmon (optional Sysinternals tool). Security event 4697, "A service was installed in the system", when its audit subcategory is on	Endpoint Security `ES_EVENT_TYPE_NOTIFY_KEXTLOAD`, and `AUTH_KEXTLOAD` to allow or deny; seen only by an ES client, such as your EDR	Kernel code comes from Apple alone (Chapter Five). Apple documents Endpoint Security for macOS only	auditd rule on `init_module` and `finit_module` (key `module-load` in the sample rules); tracepoint `module:module_load` for eBPF tools. eBPF programs do not pass through `init_module`: the kernel writes an audit record of type `BPF` (`op=LOAD`, `op=UNLOAD`) from Linux 5.6	Kernel and modules come from Google, the chip vendor and the device maker. The security log has no module-load event; it logs app starts (`TAG_APP_PROCESS_START`)
2 Firmware write or boot measurement	Intune compliance through Device Health Attestation: Secure Boot, code integrity and BitLocker state, measured at boot. Measured Boot logs (the TCG log) in `C:\Windows\Logs\MeasuredBoot`; API `Tbsi_Get_TCG_Log`; decode with TBSLogGenerator or PCPTool	Managed Device Attestation to the MDM (macOS 14, Apple silicon): signed device properties and the Secure Enclave OS version	Managed Device Attestation (iOS 16, A11 chip or later). Fresh attestations limited to one per device every seven days	TPM event log at `/sys/kernel/security/tpm0/binary_bios_measurements`; IMA list `ascii_runtime_measurements` if IMA is on; `fwupdmgr get-history` for updates fwupd applied	Security log `TAG_OS_STARTUP`: Verified Boot state (green, yellow, orange) and dm-verity mode. Key attestation and Play Integrity verdicts go to the app's own server
Firmware and silicon	The TPM holds PCR values; the firmware's event log, in the format of the TCG PC Client Platform Firmware Profile, says what was measured into them. The log is evidence only when a verifier replays it against a signed TPM quote. On Apple devices and Android, the mechanisms above report results (attestation, verdicts, a boot-state field), not a raw boot log. Limits of attestation: Chapter Seventeen.				

Sign-ins, pushes and agents (events 3 to 5)

Event	Windows	macOS	iOS	Linux	Android
3 Token issuance or sign-in	Security log 4624 on the device; Entra ID sign-in logs in the cloud	ES `ES_EVENT_TYPE_NOTIFY_AUTHENTICATION` and `..._LOGIN_LOGIN`; the identity provider's log	The identity provider's sign-in log	auditd `USER_AUTH` and `USER_LOGIN` records; records carry `auid`, the ID the user logged in with	Security log `TAG_KEYGUARD_DISMISSED` and `TAG_KEYGUARD_DISMISS_AUTH_ATTEMPT`; tokens in the identity provider's log
4 Configuration or policy push	Intune audit log: actor, target, changed properties; cannot be disabled. On the device: `DeviceManagement-Enterprise-Diagnostics-Provider/Admin`	MDM command `Status` (`Acknowledged`, `Error`, `NotNow`) on the MDM server; ES `ES_EVENT_TYPE_NOTIFY_PROFILE_ADD` on the Mac	MDM command `Status`, kept by the MDM server	No OS management channel: the configuration tool's own log, plus auditd `-w` watches on the files it changes	Security log admin events (`TAG_USER_RESTRICTION_ADDED`, `TAG_PASSWORD_COMPLEXITY_SET`, `TAG_REMOTE_LOCK`) with the admin package; EMM console log
5 AI agent or automation action	What the agent's account does: Security log, Sysmon process events. Cloud calls: Microsoft Graph activity logs, which include AI clients	ES `ES_EVENT_TYPE_NOTIFY_EXEC` for processes it starts, under the host app; identity and API logs	Service-side identity and API logs, under the user's identity	auditd syscall records, tagged with the `auid` of the user who started it	`TAG_APP_PROCESS_START` (process, APK hash); network logging of DNS lookups and TCP connections per app

THREE CELLS THAT CHANGE DECISIONS

The agent row has no agent column. None of the operating-system mechanisms above has an event class for "an agent did this". They record an account, a process or an API call. Where the platform gives the agent its own account, as Windows does in its agent workspace (Chapter Sixteen), the logs can tell agent from user. Elsewhere the agent acts as the user, and the record says the user did it. Cloud API logs can name the client app that made a call, as Microsoft Graph activity logs do, and a process record names the binary; but neither says whether a person or the agent inside that app decided. Give each agent its own identity, or in most device logs its actions are indistinguishable from the user's.

The firmware row is mostly a result, not a log. On Windows and Linux you can read the raw measurement log. On Apple devices and Android you receive a result: a signed attestation or verdict, or a boot-state field. Both are useful; they are different evidence, and the Ledger cell should say which one you hold.

The Android row depends on your EMM. Security logging is called by the device owner, or by the profile owner of an organization-owned work profile, which sees less. Network logging needs Android 8 for a device owner and Android 12 for a work profile. When an unaffiliated user or profile exists, device-wide logs pause, and full buffers are discarded.

The identity layer keeps most of the record

Three of the five events leave their best record in the cloud, and that record has a clock. Entra ID keeps sign-in and audit logs for seven days on the free tier and 30 days on P1 and P2. Microsoft Graph activity logs, an audit trail of every Graph API request with app, user, IP address and token details, need P1 or P2 and are kept only if you stream them to Log Analytics, storage or Event Hubs. Microsoft's own rough estimate for a 1,000-user tenant is about 14 GiB of storage a month. The Okta System Log keeps a 90-day sliding window. The Intune audit log can be read through Graph for two years. Microsoft Purview Audit keeps Microsoft 365 records, MailItemsAccessed among them, for 180 days in Audit (Standard) and one year in Audit (Premium); MailItemsAccessed is the log the State Department case turned on.

These numbers are why Chapter Fourteen calls visibility a cost line, and why the State Department case in Chapter Six turned on a log the customer had paid for. Export each log to a store that the console's own administrators cannot delete, as Chapter Seven recommends for every push channel: write-once or append-only storage under a separate administrator, with synchronized clocks on every source, and an alert when an export stops.

How to use it in the Ledger

- **One row, its records.** For each Ledger row, find its event above and write in column 6 each record that holds it (often one on the device, one in the management or identity service), with who reads it and how long it is kept.
- **Tag it honestly.** A mechanism in this table is **D**, documented. It becomes **O**, observed, only when you have seen it fire on your own fleet and found the event in your store. A vendor's promise to keep a log is **C**; a guess is **I**; a blank is **U** (Chapter One).
- **Test it like an isolation.** Load a test driver, push a harmless policy, sign in a test account, run a test agent task. If the record does not arrive, the cell is U, whatever the documentation says. The same rule proves an Isolate decision (Chapter Fifteen).
- **Watch the logger itself.** Android logs `TAG_LOGGING_STOPPED` and `TAG_LOG_BUFFER_SIZE_CRITICAL` at 90% of the buffer; Sysmon logs its own configuration changes (event 16). Alert on the records that say the records stopped.

> *A mechanism that exists is documented. A record you have seen arrive is observed. Only the second belongs in column 6 as O.*

APPENDIX H

Annotated Contract Clauses

Chapter Fifteen named seven clauses that carry a Trust Ledger decision into a contract. This appendix gives each one as text a buyer can put in a request for proposal, with the case behind it and the Ledger column it fixes. The wording is a starting point for your counsel, not legal advice; adapt the numbers to your own clocks. After the seven come the standard supplier terms Chapter Fifteen puts beside them, and a short table of the related asks that other chapters made.

How to use the clauses

- **Start from the Ledger, not the template.** Use a clause where a row's cell reads U, np or nc and the vendor holds the authority. Note the clause number in the optional "contract clause reference" column.
- **Ask for evidence, not promises.** Each clause asks the vendor to report what actually happened on the last cases, so the cell can move from I or U to C, and then to O once you measure it.
- **Expect refusals and settle in writing.** In the author's experience (Chapter Fifteen), vendors rarely accept a hard T1 for third-party components, uncapped liability for an outage, or a customer veto on security content during an active campaign; they more often accept staged rollout, customer-defined rings for non-urgent content, a named recovery contact, release notes for content and a printed support end.

Clause 1. Patch SLA to T1

Text. "For any vulnerability in the Product that the Supplier knows to be exploited, the Supplier will make a fix available for each device class the Customer runs within [N] days of becoming aware of the exploitation, by severity as set out in Schedule [X]. Each quarter the Supplier will report its actual time from awareness to fix availability for its last three critical cases."

Why. Edge appliances and firmware fixes arrive on the slowest clocks in the book (Chapters Four and Eleven), and the first fix for one device class can come weeks before the fix for yours. The report turns a promise into a dated record.

Column fixed. 4, Patch clock (the T0 to T1 part). **Rows:** FW, DR, LR, EX.

Clause 2. Staged rollout and customer rings for content

Text. "Every update to the Product, including every content, rule or configuration update that changes the behavior of components running with kernel or administrator privilege, will be deployed first to a canary group before general release. The Customer may assign its devices to deployment rings and choose where and when content updates are deployed to each ring."

Why. On 19 July 2024 one content update reached every Windows host running sensor 7.11 or later that was online between 04:09 and 05:27 UTC, with no customer ring in its way (Chapter Five). CrowdStrike's root cause analysis then listed staged deployment of template instances and customer control over where and when Rapid Response Content is deployed. Content is not code, but it can change privileged behavior as much as a release can.

Columns fixed. 3, Exposure and blast radius (one push no longer reaches every device), and 5, Stop, revoke, recover. **Rows:** DR, MU.

Clause 3. Pause and kill switch

Text. "The Supplier will provide a documented means for the Customer to pause each update and content channel to the Customer's devices within one hour, and will maintain the ability to stop a faulty release globally. Both will be tested jointly at least once a year, and the Supplier will record the measured time from the decision to stop until no further device received the release."

Why. A stop that has never been drilled is not a stop (Chapter Nine). The vendor's global stop protects new devices; only the customer's pause protects the customer from a release the vendor still thinks is good.

Column fixed. 5, Stop, revoke, recover (the stop, and the date of the last drill). **Rows:** MU, DR, AG.

Clause 4. Recovery support

Text. "If a Supplier release or content update leaves Customer devices unable to start or unreachable by the Supplier's console, the Supplier will provide recovery tools, technical staff and a recovery procedure that works without the Product's console and without network access to the device, and will commit to the recovery target stated in Schedule [X]. The Supplier will name a recovery contact available at all hours."

Why. Recovery from a boot loop was often hands on keyboard, device by device, with a BitLocker recovery key for each (Chapters Five and Nine). The vendor's revert stopped new failures in 78 minutes; the recovery clock ran for days.

Column fixed. 5, Stop, revoke, recover (the restore step and the time to R90). **Rows:** KN, DR, FW.

Clause 5. Support end date

Text. "The Supplier will state in this Agreement the date until which it will provide security fixes for the Product, for its firmware and for every component it bundles, and will give the Customer notice of any change to those dates no less than [one replacement cycle, in months] in advance."

Why. A support end that is not written down is found during an incident (Chapter Eleven). For products placed on the EU market, the Cyber Resilience Act sets a support period of at least five years unless the product is expected to be in use for less (Article 13(8)), so many vendors will have to state this date anyway.

Column fixed. 7, Support end. **Rows:** all.

Clause 6. Firmware support

Text. "For each hardware model supplied, the Supplier will state who builds, signs and ships firmware fixes, the channel through which they are delivered (such as Windows Update, the Linux Vendor Firmware Service or the Supplier's own tool), and the period after the model's last sale during which firmware fixes will be provided, including updates to Secure Boot certificates and keys."

Why. Fixes for firmware flaws come from the firmware supply chain, not from a revocation list (Chapter Four): LogoFAIL was fixed only by each maker's new firmware, and some makers updated the 2023 Secure Boot certificates only on platforms still in service.

Columns fixed. 2, Who can change it (the named builder, signer and channel), and 7, Support end. **Rows:** SI, FW.

Clause 7. Notification clocks

Text. "The Supplier will notify the Customer within [N] hours of becoming aware of an exploited vulnerability in the Product or of a breach of the Supplier's own systems that could affect the Customer, and will provide the logs and artifacts the Customer needs to investigate, retained for at least [period]."

Why. The Customer's own regulator clocks start from the Customer's awareness (Chapter Fourteen), so a late notice from a supplier spends the Customer's hours. CISA's Secure by Design Pledge (May 2024) asks vendors to give customers evidence of intrusions; under the CRA, a manufacturer selling in the EU must send an early warning of an actively exploited

vulnerability to its national CSIRT and ENISA within 24 hours from 11 September 2026 (Article 14(2)), and must inform the users affected, with any mitigations or corrective measures (Article 14(8)). Article 14(8) is the model for this clause's duty to the customer; Article 14(2) is the model for its clock.

Column fixed. 6, Who can see it, and the header's regulator clocks. **Rows:** ID, MU, LR.

The seven clauses on one page

Clause	Ledger column it fixes	Rows	Case behind it
1 Patch SLA to T1	4 Patch clock	FW, DR, LR, EX	Edge appliances, firmware (Chapters Four, Eleven)
2 Staged rollout and customer rings	3 Exposure and blast radius; 5 Stop, revoke, recover	DR, MU	Channel File 291 (Chapter Five)
3 Pause and kill switch	5 Stop, revoke, recover	MU, DR, AG	Untested brakes (Chapter Nine)
4 Recovery support	5 Stop, revoke, recover	KN, DR, FW	Boot loops recovered by hand (Chapters Five, Nine)
5 Support end date	7 Support end	All	The long tail; CRA Article 13(8) (Chapter Eleven)
6 Firmware support	2 Who can change it; 7 Support end	SI, FW	LogoFAIL, the 2026 certificates (Chapter Four)
7 Notification clocks	6 Who can see it; header regulator clocks	ID, MU, LR	Regulator clocks (Chapter Fourteen); CRA Article 14(2) and 14(8)

Standard supplier terms beside the seven

The seven clauses carry the Ledger decision; they are not the whole contract. Chapter Fifteen lists the standard terms that belong beside them, and the book's cases add three of their own. For Halbrook's Irish subsidiary the first group is not optional: DORA Article 30

requires terms such as audit and access rights and exit strategies in contracts for ICT services that support critical or important functions. Contracts vary by jurisdiction; this is not legal advice.

Term	What to ask for	Why (chapter)	Column
Software bill of materials	An SBOM with each release, and an exploitability statement (VEX) where a listed flaw does not affect the product	A bundled library is a row you cannot see without it (Chapter Eight)	1 Component; 4 Patch clock
Audit rights	Audit and access rights, or independent assurance reports, covering the build, signing and update systems	The build and the push channel are authorities (Chapters Seven, Eight); DORA Article 30	2 Who can change it; 6 Who can see it
Sub-processors	Notice of sub-processors and subcontractors that run or reach the service, and of changes to them	An outside console or support system can be the way in (Chapters Six, Seven)	2 Who can change it; EX row
Liability and indemnity	Liability caps with carve-outs for the failures this book records, and indemnity	Who pays when a push fails is decided before it fails (Chapters Twelve, Thirteen)	Optional "who pays on failure"
Exit and termination assistance	Data export, a transition period and help moving to a replacement, at a stated price	A supplier can fail, be sold or be barred by a dated order (Chapters Seven, Fifteen); DORA Article 30	7 Support end; 8 Decision and owner
Signing-key custody	Where release and update signing keys are held (for example in an HSM), who can approve a signature, and how a compromised key is revoked	Stolen keys and compromised builds produced signed malware in Stuxnet, SolarWinds and 3CX (Chapters Eight, Thirteen)	2 Who can change it; 5 Stop, revoke, recover

Security logs in the base price	The sign-in, admin and data-access logs column 6 needs, included without a premium tier, exportable through an API, kept for a stated period	The State Department saw Storm-0558 only because its license tier included the log (Chapter Six)	6 Who can see it
Pre-release ring	Access to each release, content update included, in a test tenant or early ring before general release	Customer rings are the stop a customer owns (Chapters Five, Nine)	5 Stop, revoke, recover

Related asks from other chapters

Ask	Why (chapter)	Column
Host-patch notice and maintenance windows for a rented virtual-desktop host	The provider patches the silicon under your sessions on its own clock (Chapter Three)	4 Patch clock; 6 Who can see it
The console's administrator list and login logs from each managed service provider and SaaS console vendor	Whoever controls the console controls the fleet (Chapter Seven)	2 Who can change it; 6 Who can see it
Which components still run in kernel mode, how content is staged by default, and how to pause it, in writing	The move of Windows sensors to user mode runs on vendor dates (Chapter Five)	2 Who can change it; 5 Stop, revoke, recover
Build provenance and the build platform	Signed is not safe: the build is an authority (Chapter Eight)	2 Who can change it
A dated post-quantum roadmap for signing and connection keys	Firmware and appliance signatures lag by a hardware generation (Chapter Seventeen)	7 Support end; 8 Decision and owner
For an AI agent: the model is rented, but the identity, the tool list and the approval for irreversible actions stay with you	An agent is an authority with its own row (Chapter Sixteen)	2 Who can change it; 5 Stop, revoke, recover

Method reference: NIST SP 800-161 Revision 1 (May 2022, updated November 2024) is the practice of cybersecurity supply chain risk management behind these clauses; the Trust Ledger is the one-page view that points to it (Chapter Fifteen).

APPENDIX E

Reading List and Sources

Part 1 is a short shelf of primary sources behind the whole book: standards, vendor security guides, incident reports, court and government records and original papers. Part 2 lists the sources each chapter relied on, taken from the source notes kept with every chapter, so a reader can check any claim at its origin. Sources were checked in November 2026; web pages change, so check the date on anything you rely on.

Part 1: Primary sources for the whole book

Source	Why read it	Where
W. H. Ware (ed.), "Security Controls for Computer Systems", Defense Science Board Task Force report, RAND R-609-1 (1970, reissued 1979)	The first official study of shared computers and who controls them.	https://www.rand.org/pubs/reports/R609-1.html
J. P. Anderson, "Computer Security Technology Planning Study", US Air Force ESD-TR-73-51 (October 1972)	Where the reference monitor idea comes from.	https://csrc.nist.gov/files/pubs/conference/1998/10/08/proceedings-of-the-21st-nissc-1998/final/docs/early-cs-papers/ande72a.pdf
J. H. Saltzer and M. D. Schroeder, "The Protection of Information in Computer Systems", Proceedings of the IEEE (1975)	The design principles, such as least privilege, that every chapter tests.	https://www.cs.virginia.edu/~evans/cs551/saltzer/
K. Thompson, "Reflections on Trusting Trust", Communications of the ACM (August 1984)	Why you cannot fully trust code you did not build yourself.	https://www.cs.cmu.edu/~rdriley/487/papers/Thompson_1984_ReflectionsonTrustingTrust.pdf
NIST SP 800-193, Platform Firmware Resiliency Guidelines (May 2018)	Protect, detect and recover for the firmware under the OS.	https://csrc.nist.gov/pubs/sp/800/193/final
UEFI Forum, UEFI Specification 2.10, Secure Boot and Driver Signing	How Secure Boot keys and signature databases work.	https://uefi.org/specs/UEFI/2.10/32_Secure_Boot_and_Driver_Signing.html
Microsoft Support, Windows Secure Boot certificate expiration and CA updates	The 2011 Secure Boot certificates that expire in 2026, and the move to new ones.	https://support.microsoft.com/en-us/topic/windows-secure-boot-certificate-expiration-and-ca-updates-7ff40d33-95dc-4c3c-8725-a9b95457578e
Microsoft, KB5025885, Boot Manager revocations for CVE-2023-24932 (BlackLotus, May 2023)	How a boot loader is revoked and why it takes years.	https://support.microsoft.com/kb/5025885
Apple, Apple Platform Security guide	Apple's own account of the Secure Enclave, boot chain and OS integrity.	https://support.apple.com/guide/security/welcome/web
Android Open Source Project, Verified Boot	How Android checks each stage of boot and shows the device state.	https://source.android.com/docs/security/features/verifiedboot

NIST SP 800-40 Rev. 4, Guide to Enterprise Patch Management Planning (April 2022)	Patching as routine maintenance, the basis of the patch clock.	https://csrc.nist.gov/pubs/sp/800/40/r4/final
CISA, Known Exploited Vulnerabilities Catalog	Which flaws are exploited in the wild, with federal fix deadlines.	https://www.cisa.gov/known-exploited-vulnerabilities-catalog
CrowdStrike, Channel File 291 Incident Root Cause Analysis (6 August 2024)	The vendor's own account of the 19 July 2024 outage.	https://www.crowdstrike.com/wp-content/uploads/2024/08/Channel-File-291-Incident-Root-Cause-Analysis-08.06.2024.pdf
Cyber Safety Review Board, Review of the Summer 2023 Microsoft Exchange Online Intrusion (March 2024)	How a stolen signing key opened cloud mail (Storm-0558).	https://www.cisa.gov/sites/default/files/2025-03/CSRBReviewOfTheSummer2023MEOIntrusion508.pdf
Cyber Safety Review Board, Review of the December 2021 Log4j Event (July 2022)	Why one open source library became a long tail of risk.	https://www.cisa.gov/sites/default/files/2023-02/CSRB-Report-on-Log4j-PublicReport-July-11-2022-508-Compliant.pdf
CISA, Emergency Directive 21-01, SolarWinds Orion Code Compromise (December 2020)	The government order that followed a signed, poisoned update.	https://www.cisa.gov/news-events/directives/ed-21-01-mitigate-solarwinds-orion-code-compromise
A. Freund, "backdoor in upstream xz/liblzma leading to ssh server compromise", oss-security list (29 March 2024)	The first report of the xz-utils backdoor.	https://www.openwall.com/lists/oss-security/2024/03/29/4
IETF, RFC 9334, Remote ATtestation procedureS (RATS) Architecture (January 2023)	How a device proves its state to someone else.	https://www.rfc-editor.org/rfc/rfc9334.html
NIST, FIPS 203, 204 and 205 approved (August 2024)	The first post-quantum standards for key exchange and signatures.	https://csrc.nist.gov/news/2024/post quantum-cryptography-fips-approved
Regulation (EU) 2024/2847, Cyber Resilience Act (October 2024)	The EU law that makes makers patch and report for a support period.	https://eur-lex.europa.eu/eli/reg/2024/2847/oj/eng

Part 2: Sources by chapter

Reader's Brief

- SEC cybersecurity disclosure rules adopted 26 July 2023; Form 8-K Item 1.05 "generally be due four business days after a registrant determines" an incident is material: https://www.sec.gov/newsroom/press-releases/2023-139
- US banking agencies' computer-security incident notification rule: notify "no later than 36 hours after the banking organization determines" a notification incident ...: https://www.fdic.gov/news/financial-institution-letters/2021/fil21074.html
- DORA, Regulation (EU) 2022/2554, Article 64: "It shall apply from 17 January 2025": https://eur-lex.europa.eu/eli/reg/2022/2554/oj/eng
- GDPR Article 33(1): notify the supervisory authority "not later than 72 hours after having become aware of it", where feasible, "unless the personal data breach is ...: https://eur-lex.europa.eu/eli/reg/2016/679/oj/eng
- Cyber Resilience Act, Regulation (EU) 2024/2847, Article 71: applies from 11 December 2027; Article 14 (reporting) from 11 September 2026: https://eur-lex.europa.eu/eli/reg/2024/2847/oj/eng
- PCI DSS applies to system components that store, process or transmit cardholder data, or could affect its security (scoping section of PCI DSS v4.0.1): https://www.pcisecuritystandards.org/document_library/

Prologue: One Power Button, One Approval

- Microsoft delivers Intel microcode updates through Windows Update (summary of Intel microcode KB articles): https://support.microsoft.com/en-us/help/4093836
- Secure Boot key roles: OEMs may manage the PK or use the Microsoft-managed PK ("the preferred PK for the Windows ecosystem"); KEK holds Microsoft's key; DB holds Windows ...: https://learn.microsoft.com/en-us/windows-hardware/manufacture/desktop/windows-secure-boot-key-creation-and-management-guidance
- Windows Production PCA 2011 expires 19 October 2026; devices without the 2023 certificates "will continue to start and operate normally" but cannot receive new ...: https://support.microsoft.com/en-us/topic/windows-secure-boot-certificate-expiration-and-ca-updates-7ff40d33-95dc-4c3c-8725-a9b95457578e
- Memory integrity "is enabled by default on many new Windows 11 devices that meet hardware and compatibility requirements": https://learn.microsoft.com/en-us/windows/security/hardware-security/enable-virtualization-based-protection-of-code-integrity
- ELAM drivers are "initialized before other boot-start drivers": https://learn.microsoft.com/en-us/windows-hardware/drivers/install/early-launch-antimalware
- "Starting with Windows 10, version 1607, Windows will not load any new kernel-mode drivers which are not signed by the Dev Portal": https://learn.microsoft.com/en-us/windows-hardware/drivers/install/kernel-mode-code-signing-policy--windows-vista-and-later-
- Windows Hello for Business: "the user's private key protected by the device's security module" plus a PIN; "the PIN never leaves the device": https://learn.microsoft.com/en-us/windows/security/identity-protection/hello-for-business/
- BitLocker recovery password is a 48-digit number; firmware and boot-manager changes trigger recovery; access to recovery passwords in Microsoft Entra ID "can also be ...: https://learn.microsoft.com/en-us/windows/security/operating-system-security/data-protection/bitlocker/recovery-overview
- Microsoft announced a monthly security release cycle on 1 October 2003: https://learn.microsoft.com/en-us/archive/blogs/test/10-years-of-update-tuesdays
- CrowdStrike: hosts running Falcon sensor for Windows 7.11 and above "online between Friday, July 19, 2024 04:09 UTC and Friday, July 19, 2024 05:27 UTC, may be ...: https://www.crowdstrike.com/blog/falcon-update-for-windows-hosts-technical-details/
- CrowdStrike preliminary post-incident review (04:09 UTC update, 05:27 UTC revert): https://www.crowdstrike.com/en-us/blog/falcon-content-update-preliminary-post-incident-report/
- Microsoft: "CrowdStrike's update affected 8.5 million Windows devices, or less than one percent of all Windows machines" (Microsoft's estimate): https://blogs.microsoft.com/blog/2024/07/20/helping-our-customers-through-the-crowdstrike-outage/
- Microsoft recovery steps: delete C-00000291*.sys; BitLocker recovery key may be required: https://support.microsoft.com/en-us/help/5042421
- Microsoft recovery steps: delete C-00000291*.sys; BitLocker recovery key may be required: https://support.microsoft.com/en-us/help/5042429
- ELAM "can load a Microsoft or non-Microsoft anti-malware driver before all non-Microsoft boot drivers"; Measured Boot: firmware "stores in the TPM a hash of the ...: https://learn.microsoft.com/en-us/windows/security/operating-system-security/system-security/secure-the-windows-10-boot-process
- Intune compliance policies use Microsoft's Device Health Attestation service to check BitLocker, Secure Boot and code integrity at boot: https://learn.microsoft.com/en-us/intune/device-security/compliance/ref-windows-settings
- Keylime: "remote boot attestation and runtime integrity measurement", follows the TCG TPM 2.0 specification; a CNCF sandbox project: https://keylime.dev/
- At reset an x86 processor begins fetching its first instruction from a fixed address (the reset vector) in platform firmware: https://www.intel.com/content/www/us/en/developer/articles/technical/intel-sdm.html
- Measured boot: firmware extends measurements into TPM platform configuration registers (PCRs) and records each event in an event log kept outside the TPM: https://trustedcomputinggroup.org/resource/pc-client-specific-platform-firmware-profile-specification/

Chapter 1. Who Controls It: Authorities, Clocks and the Trust Ledger

- Ware (chair), *Security Controls for Computer Systems*, RAND R-609, February 1970; classified Confidential; declassified by ARPA on 10 October 1975; reissued as R-609-1 ...: https://www.rand.org/pubs/reports/R609-1.html
- Anderson, *Computer Security Technology Planning Study*, ESD-TR-73-51, October 1972: the reference validation mechanism "must be tamper proof", "must always be invoked", ...: https://csrc.nist.gov/files/pubs/conference/1998/10/08/proceedings-of-the-21st-nissc-1998/final/docs/early-cs-papers/ande72a.pdf

- Saltzer and Schroeder, "The Protection of Information in Computer Systems", Proc. IEEE 63(9), September 1975; least privilege and fail-safe defaults wording: https://www.cs.virginia.edu/~evans/cs551/saltzer/
- Thompson, "Reflections on Trusting Trust", Communications of the ACM 27(8), August 1984: "You can't trust code that you did not totally create yourself.": https://www.cs.cmu.edu/~rdriley/487/papers/Thompson_1984_ReflectionsonTrustingTrust.pdf
- Morris worm released the evening of 2 November 1988: https://spaf.cerias.purdue.edu/tech-reps/823.pdf
- Microsoft announced monthly security releases on 1 October 2003: https://learn.microsoft.com/en-us/archive/blogs/test/10-years-of-update-tuesdays
- Windows XP Service Pack 2 released to manufacturing 6 August 2004: https://learn.microsoft.com/en-us/archive/blogs/mswanson/windows-xp-service-pack-2-rtm-available-to-msdn-subscribers
- App Store opened 10 July 2008: https://www.apple.com/newsroom/2008/07/10iPhone-3G-on-Sale-Tomorrow/
- "All third-party apps are sandboxed" (iOS): https://support.apple.com/guide/security/sec15bfe098e/web
- Microsoft Security Advisory 2286198 (CVE-2010-2568), published 16 July 2010: https://learn.microsoft.com/en-us/security-updates/securityadvisories/2010/2286198
- Windows 8 general availability 26 October 2012: https://learn.microsoft.com/en-us/archive/blogs/serverandtools/windows-8-arrives-worldwide
- x86 PCs certified for Windows 8 must have Secure Boot on by default: https://technet.microsoft.com/library/dn486875.aspx
- Heartbleed, CVE-2014-0160, OpenSSL advisory 7 April 2014 (TLS heartbeat read overrun): https://openssl-library.org/news/secadv/20140407.txt
- Spectre and Meltdown published 3 January 2018, "before an originally coordinated disclosure date of January 9, 2018": https://security.googleblog.com/2018/01/todays-cpu-vulnerability-what-you-need.html
- SolarWinds: CISA Emergency Directive 21-01, 13 December 2020: https://www.cisa.gov/news-events/directives/ed-21-01-mitigate-solarwinds-orion-code-compromise
- CrowdStrike content update 19 July 2024 04:09 UTC, reverted 05:27 UTC (78 minutes): https://www.crowdstrike.com/en-us/blog/falcon-content-update-preliminary-post-incident-report/
- Microsoft's estimate: about 8.5 million Windows devices affected: https://blogs.microsoft.com/blog/2024/07/20/helping-our-customers-through-the-crowdstrike-outage/
- Microsoft 2011 Secure Boot certificates: KEK CA 2011 expires 24 June 2026, UEFI CA 2011 27 June 2026, Windows Production PCA 2011 19 October 2026: https://support.microsoft.com/en-us/topic/windows-secure-boot-certificate-expiration-and-ca-updates-7ff40d33-95dc-4c3c-8725-a9b95457578e
- NIST SP 800-40 Rev. 4, *Guide to Enterprise Patch Management Planning: Preventive Maintenance for Technology*, final 6 April 2022; patching as preventive maintenance and …: https://csrc.nist.gov/pubs/sp/800/40/r4/final
- BOD 22-01 (3 November 2021) created the KEV catalog: remediate "within 6 months for vulnerabilities with a CVE ID assigned prior to 2021 and within two weeks for all …: https://www.cisa.gov/news-events/directives/bod-22-01-reducing-significant-risk-known-exploited-vulnerabilities
- BOD 26-04, 10 June 2026: deadlines set by asset exposure, KEV status, exploit automation and technical impact; shortest timeline three days (with forensic triage); KEV …: https://www.cisa.gov/news-events/directives/bod-26-04-prioritizing-security-updates-based-risk
- Android update policy: postpone once for up to 30 days; freeze periods of at most 90 days: https://developer.android.com/reference/android/app/admin/SystemUpdatePolicy
- Apple MDM software update deferral: 1 to 90 days: https://support.apple.com/guide/deployment/dep0578d8b8a/web
- DORA Article 28(3): financial entities "shall maintain and update … a register of information in relation to all contractual arrangements on the use of ICT services …: https://eur-lex.europa.eu/eli/reg/2022/2554/oj/eng
- Evergreen WebView2 Runtime "updates automatically"; updates can be disabled by an administrator: https://learn.microsoft.com/en-us/microsoft-edge/webview2/concepts/distribution
- Defender monthly platform and engine updates roll out gradually; if no update channel is configured, Microsoft assigns the device a channel: https://learn.microsoft.com/en-us/defender-endpoint/manage-gradual-rollout
- NIST Cybersecurity Framework 2.0 (February 2024) includes the Cybersecurity Supply Chain Risk Management category, GV.SC: https://nvlpubs.nist.gov/nistpubs/CSWP/NIST.CSWP.29.pdf

Chapter 2. Five Platforms, Five Trust Models

- iOS 27 and macOS Golden Gate 27 released 14 September 2026; latest security releases 27.2.1 on 9 November 2026: https://support.apple.com/en-us/100100

- Windows 11 26H2 available 29 September 2026; 25H2 30 September 2025; 24H2 1 October 2024; 25H2 and 26H2 delivered as enablement packages: https://learn.microsoft.com/en-us/windows/release-health/windows11-release-information
- Windows 11 26H2 available 29 September 2026; 25H2 30 September 2025; 24H2 1 October 2024; 25H2 and 26H2 delivered as enablement packages: https://support.microsoft.com/help/5054156
- Windows 11 26H2 available 29 September 2026; 25H2 30 September 2025; 24H2 1 October 2024; 25H2 and 26H2 delivered as enablement packages: https://support.microsoft.com/help/5121794
- Android developer verification: from 30 September 2026 in Brazil, Indonesia, Singapore and Thailand on certified devices; global rollout 2027; unregistered apps …: https://android-developers.googleblog.com/2025/08/elevating-android-security.html
- Android developer verification: from 30 September 2026 in Brazil, Indonesia, Singapore and Thailand on certified devices; global rollout 2027; unregistered apps …: https://developer.android.com/developer-verification/guides
- PK managed by the OEM or the Microsoft-managed PK ("the preferred PK for the Windows ecosystem"); Microsoft 3rd-party UEFI CA and Option ROM CA for third-party drivers …: https://learn.microsoft.com/en-us/windows-hardware/manufacture/desktop/windows-secure-boot-key-creation-and-management-guidance
- Secured-core PCs: "3rd party UEFI CA not trusted by default, with BIOS option for enabling trust": https://learn.microsoft.com/en-us/windows-hardware/design/device-experiences/oem-highly-secure-11
- Kernel-mode code signing since Windows 10 version 1607, and its exceptions (upgrades, Secure Boot off, certificates issued before 29 July 2015): https://learn.microsoft.com/en-us/windows-hardware/drivers/install/kernel-mode-code-signing-policy--windows-vista-and-later-
- "Since the Windows 11 2022 update, the vulnerable driver blocklist is enabled by default for all devices": https://learn.microsoft.com/en-us/windows/security/application-security/application-control/app-control-for-business/design/microsoft-recommended-driver-block-rules
- Memory integrity on by default on many new devices that qualify: https://learn.microsoft.com/en-us/windows/security/hardware-security/enable-virtualization-based-protection-of-code-integrity
- Pluton firmware authored by Microsoft and delivered through Windows Update: https://learn.microsoft.com/en-us/windows/security/hardware-security/pluton/microsoft-pluton-security-processor
- Apple silicon boot: Boot ROM "the first step in the chain of trust", LLB, iBoot; Secure Enclave Boot ROM: https://support.apple.com/guide/security/secac71d5623/web
- Startup security: Full Security default; kernel extensions need Reduced Security, enabled by the owner in recoveryOS: https://support.apple.com/guide/security/sec7d92dc49f/web
- Startup security: Full Security default; kernel extensions need Reduced Security, enabled by the owner in recoveryOS: https://support.apple.com/guide/security/sec8e454101b/web
- Signed System Volume from macOS 11; SPTM with TXM "replaces the PPL" on A15 or later and M2 or later: https://support.apple.com/guide/security/secd698747c9/web
- Signed System Volume from macOS 11; SPTM with TXM "replaces the PPL" on A15 or later and M2 or later: https://support.apple.com/guide/security/sec8b776536b/web
- Developer ID signing and notarization; Gatekeeper: https://support.apple.com/guide/security/sec5599b66df/web
- EU alternative marketplaces from iOS 17.4 (March 2024); notarization "applies to all apps, regardless of their distribution channel": https://developer.apple.com/support/dma-and-apps-in-the-eu/
- Japan changes with iOS 26.2 under the Mobile Software Competition Act (announced 17 December 2025): https://www.apple.com/newsroom/2025/12/apple-announces-changes-to-ios-in-japan/
- Background Security Improvements from iOS, iPadOS and macOS 26.1; first released 17 March 2026: https://support.apple.com/en-us/126604
- Software update deferral 1 to 90 days (declarative configuration): https://support.apple.com/guide/deployment/dep0578d8b8a/web
- Lockdown Mode in iOS 16 or later and macOS Ventura or later: https://support.apple.com/en-us/105120
- shim: "a first-stage UEFI bootloader" validating the next stage "against a built-in certificate"; Machine Owner Keys: https://github.com/rhboot/shim
- SBAT revokes by generation number instead of image hash; first shipped in shim 15.3 (March 2021): https://github.com/rhboot/shim/blob/main/SBAT.md
- SBAT revokes by generation number instead of image hash; first shipped in shim 15.3 (March 2021): https://github.com/rhboot/shim/releases/tag/15.3

- Kernel lockdown LSM in Linux 5.4 (released 24 November 2019), integrity and confidentiality modes: https://kernelnewbies.org/Linux_5.4
- RHEL 10.0 GA 20 May 2025; ten-year Full and Maintenance support; SELinux enforcing by default on new installs: https://access.redhat.com/articles/3078
- RHEL 10.0 GA 20 May 2025; ten-year Full and Maintenance support; SELinux enforcing by default on new installs: https://access.redhat.com/support/policy/updates/errata
- Ubuntu 24.04 LTS: standard security maintenance to May 2029, Ubuntu Pro to May 2034, Legacy add-on to May 2039; AppArmor loaded by default: https://ubuntu.com/about/release-cycle
- Ubuntu 24.04 LTS: standard security maintenance to May 2029, Ubuntu Pro to May 2034, Legacy add-on to May 2039; AppArmor loaded by default: https://ubuntu.com/server/docs/how-to/security/apparmor/
- LVFS: "a secure portal which allows hardware vendors to upload firmware updates": https://fwupd.org
- Android Verified Boot from "a hardware-protected root of trust to the bootloader"; orange state when unlocked: https://source.android.com/docs/security/features/verifiedboot
- GKI: "moving SoC and board support out of the core kernel into loadable vendor modules"; required from Android 12 for kernel 5.10 or higher: https://source.android.com/docs/core/architecture/kernel/generic-kernel-image
- AVF introduced in Android 13; pKVM is the standard hypervisor: https://source.android.com/docs/core/virtualization
- App sandbox: unique UID per app, "a kernel-level Application Sandbox": https://source.android.com/docs/security/app-sandbox
- "In Android 5.0 and later, SELinux is fully enforced": https://source.android.com/docs/security/features/selinux
- Mainline introduced in Android 10, delivered through Google Play system updates: https://source.android.com/docs/core/ota/modular-system
- Android partners notified "at least a month before publishing the bulletin": https://source.android.com/docs/security/bulletin/2026/2026-11-01
- Pixel 8 (4 October 2023): seven years of software updates, including OS and security updates: https://blog.google/products/pixel/google-pixel-8-pro/
- Titan M2 in Pixel 6 (October 2021); StrongBox from Android 9: https://security.googleblog.com/2021/10/
- Titan M2 in Pixel 6 (October 2021); StrongBox from Android 9: https://developer.android.com/privacy-and-security/keystore
- Android 17 stable release, 16 June 2026: https://blog.google/products/android/
- SystemUpdatePolicy: one 30-day postponement (monthly security patches not affected); freeze periods up to 90 days block all updates including security patches: https://developer.android.com/reference/android/app/admin/SystemUpdatePolicy
- Windows 11 IoT Enterprise LTSC 2024: start 1 October 2024, end of support 10 October 2034: https://learn.microsoft.com/en-us/lifecycle/products/windows-11-iot-enterprise-ltsc-2024
- EU DMA: Commission designated six gatekeepers on 6 September 2023: https://digital-markets-act.ec.europa.eu/commission-designates-six-gatekeepers-under-digital-markets-act-2023-09-06_en
- Windows 11 IoT Enterprise LTSC 2024 reaches end of support on 10 October 2034 (Fixed Policy): https://learn.microsoft.com/en-us/lifecycle/end-of-support/end-of-support-2034
- Permissive Security tells iBoot to "accept locally Secure Enclave-signed boot objects"; "Apple doesn't provide or support custom XNU kernels"; the LocalPolicy is signed …: https://support.apple.com/guide/security/startup-disk-security-policy-control-sec7d92dc49f/web
- Permissive Security tells iBoot to "accept locally Secure Enclave-signed boot objects"; "Apple doesn't provide or support custom XNU kernels"; the LocalPolicy is signed …: https://support.apple.com/guide/security/localpolicy-signing-key-creation-and-management-sec1f90fbad1/web
- Android Enterprise Recommended: knowledge-worker devices get 90-day security updates; makers publish "the end date when the device is guaranteed to receive security …: https://support.google.com/work/android/answer/16751490
- fapolicyd is "a framework for an application allowlist solution"; by default it trusts software installed by the RPM package manager: https://docs.redhat.com/en/documentation/Red_Hat_Enterprise_Linux/10/html/security_hardening/blocking-and-allowing-applications-by-using-fapolicyd
- GKI monthly releases contain a tested boot.img with "a Google inserted certificate to attest that the binaries were built from a known source code baseline": https://source.android.com/docs/core/architecture/kernel/gki-releases

Chapter 3. Silicon and the Computers Inside the Computer

- Azure: "majority of Azure infrastructure has already been updated"; automatic reboot of remaining VMs from 3:30pm PST, 3 January 2018; self-service window ended: https://azure.microsoft.com/en-us/blog/securing-azure-customers-from-cpu-vulnerability/
- Project Zero post (Jann Horn, 3 Jan 2018): reported to Intel, AMD and ARM on 2017-06-01; variants CVE-2017-5753 (bounds check bypass), CVE-2017-5715 (branch target ...: https://googleprojectzero.blogspot.com/2018/01/reading-privileged-memory-with-side.html
- Posting "before an originally coordinated disclosure date of January 9, 2018" and why: https://security.googleblog.com/2018/01/todays-cpu-vulnerability-what-you-need.html
- CVE-2017-5753 and 5715 = Spectre, 5754 = Meltdown; "We don't know" if abused; Meltdown affects "effectively every processor since 1995 (except Intel Itanium and Intel ...: https://meltdownattack.com/
- Apple: Meltdown mitigations in iOS 11.2, macOS 10.13.2, tvOS 11.2; Spectre mitigations in iOS 11.2.2 and macOS 10.13.2 Supplemental Update: https://support.apple.com/en-us/101886
- macOS 10.13.2 released 6 Dec 2017 (HT208331): https://support.apple.com/en-ca/103178
- iOS 11.2.2 and macOS 10.13.2 Supplemental Update released 8 Jan 2018: https://support.apple.com/en-us/103179
- Linux 4.14.11 released 2 Jan 2018 with kernel page table isolation: https://cdn.kernel.org/pub/linux/kernel/v4.x/ChangeLog-4.14.11
- Microsoft ADV180002 released 3 Jan 2018: https://msrc.microsoft.com/update-guide/vulnerability/ADV180002
- INTEL-SA-00088, released 3 Jan 2018, includes "updated Intel microprocessor microcode to our customers and partners": https://www.intel.com/content/www/us/en/security-center/advisory/intel-sa-00088.html
- Intel's microcode could cause "higher than expected reboots and other unpredictable system behavior"; on 22 January Intel recommended customers stop deploying it; ...: https://support.microsoft.com/kb/4078130
- Microsoft ships Intel microcode updates through Windows Update, WSUS and the Update Catalog: https://support.microsoft.com/kb/4093836
- Downfall CVE-2022-40982: Gather instruction; reported to Intel 24 Aug 2022; "affected processors have been around since 2014": https://downfall.page/
- INTEL-SA-00828 released 8 Aug 2023: https://www.intel.com/content/www/us/en/security-center/advisory/intel-sa-00828.html
- GDS mitigation: "performance impacts of up to 50%" for specific workloads; MSR opt-out; gather_data_sampling=off; SGX attests up to date only when FIT-loaded "(for ...: https://www.intel.com/content/www/us/en/developer/articles/technical/software-security-guidance/technical-documentation/gather-data-sampling.html
- Zenbleed CVE-2023-20593: AMD-SB-7008 initial publication 24 Jul 2023; Rome microcode 0x0830107B dated 2023-06-06; AGESA dates per product (2023-11-07 to 2024-04-15); ...: https://www.amd.com/en/resources/product-security/bulletin/amd-sb-7008.html
- Zenbleed reported to AMD 15 May 2023 by Tavis Ormandy: https://lock.cmpxchg8b.com/zenbleed.html
- Linux microcode loader: early load from initrd; reapplied on resume; use case "beyond the OEM End-Of-Life support"; late loading off by default since 5.19: https://docs.kernel.org/arch/x86/microcode.html
- Linux commit 4e2c719782a8 (authored 22 Apr 2025), adds /sys/devices/system/cpu/vulnerabilities/old_microcode; quote on keeping microcode up to date: https://github.com/torvalds/linux/commit/4e2c719782a8
- GoFetch: disclosed to Apple 5 Dec 2023, 107 days before public release; DIT on M3 disables the DMP, not on M1 and M2: https://gofetch.fail/
- Kim et al., "Flipping Bits in Memory Without Accessing Them", ISCA 2014: 110 of 129 modules; industry aware since at least 2012 (Intel patent applications): https://users.ece.cmu.edu/~yoonguk/papers/kim-isca14.pdf
- Phoenix (CVE-2025-6202): all 15 SK Hynix DDR5 DIMMs tested; embargo to 15 Sept 2025; privilege escalation in default settings; AMD client BIOS update not independently ...: https://comsec.ethz.ch/research/dram/phoenix/
- Thunderspy: 7 vulnerabilities; "cannot be fixed in software"; "will require a silicon redesign"; no fix for systems purchased before 2019: https://thunderspy.io/
- Kernel DMA Protection uses the IOMMU; requires virtualization (IOMMU) enabled in firmware; state visible in Windows Security and msinfo32: https://learn.microsoft.com/en-us/windows/security/hardware-security/kernel-dma-protection-for-thunderbolt
- Windows 11 minimum requirement: TPM 2.0: https://support.microsoft.com/en-us/windows/windows-11-specifications-and-system-requirements-86c11283-ea52-4782-9efd-7674389a7ba3

- Pluton: designed by Microsoft, built into SoCs; Microsoft authors firmware, delivered through Windows Update; loads from SPI flash at boot, newer from Windows Update at …: https://learn.microsoft.com/en-us/windows/security/hardware-security/pluton/microsoft-pluton-security-processor
- Secure Enclave: "dedicated secure subsystem"; own boot ROM, "immutable code"; verifies sepOS signature: https://support.apple.com/guide/security/secure-enclave-sec59b0b31ff/web
- Titan M2 (Pixel 6): "fully designed and developed by Google", in-house RISC-V; Trusty open-source TEE: https://security.googleblog.com/2021/10/pixel-6-setting-new-standard-for-mobile.html
- Caliptra: "IP and firmware for an integrated Root of Trust block", targets datacenter-class SoCs: https://github.com/chipsalliance/Caliptra
- INTEL-SA-00086 (20 Nov 2017): ME 6.x to 11.x, SPS, TXE; "check with your system manufacturer for updated firmware": https://www.intel.com/content/www/us/en/security-center/advisory/intel-sa-00086.html
- CVE-2024-54085 (AMI MegaRAC SPx) added to CISA KEV 25 Jun 2025: https://www.cisa.gov/known-exploited-vulnerabilities-catalog
- Copilot+ PCs: NPU of "more than 40 trillion operations per second (TOPS)": https://learn.microsoft.com/en-us/windows/ai/npu-devices/
- Measurement: torvalds/linux drivers/base/cpu.c, commit 87590ce6e373 (7 Jan 2018) adds the vulnerabilities folder with 3 entries; master on 18 Nov 2026 (latest change …: https://github.com/torvalds/linux/blob/master/drivers/base/cpu.c
- EntrySign (CVE-2024-56161): insecure hash in AMD microcode signature validation lets a local administrator load malicious microcode; Zen 1 to Zen 4, Zen 5 added 7 April …: https://github.com/google/security-research/security/advisories/GHSA-4xq7-4mgh-gp6w
- EntrySign fix requires an OEM BIOS update; OS microcode packages cannot fully resolve it alone (AMD-SB-7033, AMD-SB-3019): https://ubuntu.com/security/vulnerabilities/entrysign
- SinkClose (CVE-2023-31315): AMD-SB-7014 first published 9 Aug 2024; revised 14 Aug (Ryzen 3000 "Matisse" mitigation target), 19 Aug (mitigation available), 30 Oct …: https://www.amd.com/en/resources/product-security/bulletin/amd-sb-7014.html
- /sys/devices/system/cpu/vulnerabilities files report "Not affected", "Vulnerable" or "Mitigation: $M": https://www.kernel.org/doc/Documentation/ABI/testing/sysfs-devices-system-cpu

Chapter 4. Firmware and Boot: Keys with End Dates

- The three 2011 certificates, roles, stores, expiry dates (KEK CA 2011 24 Jun 2026; UEFI CA 2011 27 Jun 2026; Windows Production PCA 2011 19 Oct 2026) and 2023 successors …: https://support.microsoft.com/kb/5062710
- Expiry: device "will still start and operate normally"; "will no longer be able to receive new security protections for the early boot process"; "New Secure Boot and …: https://support.microsoft.com/kb/5079373
- Confidence levels; "High Confidence": "Microsoft has validated this device class is safe for updates": https://support.microsoft.com/kb/5062713
- OEMs "must create, sign, and submit updates for the new Microsoft Corporation KEK CA 2023"; PK enrollment and KEK rules: https://learn.microsoft.com/en-us/windows-hardware/manufacture/desktop/windows-secure-boot-key-creation-and-management-guidance
- Windows 8 lifecycle start date 30 Oct 2012: https://learn.microsoft.com/en-us/lifecycle/products/windows-8
- UEFI Secure Boot: PK, KEK, authorized and forbidden signature databases; image in forbidden database "will not be started": https://uefi.org/specs/UEFI/2.10/32_Secure_Boot_and_Driver_Signing.html
- UEFI is based on Intel's EFI 1.10 specification, maintained by the UEFI Forum: https://uefi.org/faq
- Dell: "The computer is still able to boot"; Default database updated by BIOS flash, Active database by Windows Update; platforms with EoSL after 31 Dec 2025 updated; …: https://www.dell.com/support/kbdoc/en-us/000390990/secure-boot-transition-faq
- Red Hat: systems continue to boot after June 2026; shim signed by 2011 key boots while the certificate is not removed or revoked; dual-signed shims from RHEL 9.8 and …: https://www.redhat.com/en/blog/expiration-secure-boot-signing-certificates-2026
- Apple boot chain: boot ROM "immutable code", Apple Root CA public key, iBoot verifies kernel; Recovery and DFU on failure: https://support.apple.com/guide/security/boot-process-for-iphone-and-ipad-devices-secb3000f149/web
- Mac Apple silicon: Full, Reduced, Permissive Security; changes from recoveryOS; LocalPolicy; kexts need Reduced Security: https://support.apple.com/guide/security/startup-disk-security-policy-control-sec7d92dc49f/web
- Android Verified Boot: rollback protection "to a vulnerable version of Android": https://source.android.com/docs/security/features/verifiedboot/avb

- Android device states: unlocking wipes data partitions; user-settable root of trust on Pixel "in addition to the built-in root of trust": https://source.android.com/docs/security/features/verifiedboot/device-state
- BlackLotus (ESET, 1 Mar 2023): first in-the-wild UEFI bootkit bypassing Secure Boot on fully updated systems; "sold on hacking forums for $5,000 since at least October …: https://www.welivesecurity.com/2023/03/01/blacklotus-uefi-bootkit-myth-confirmed/
- CVE-2022-21894 released 11 Jan 2022; CVE-2023-24932 released 9 May 2023: https://msrc.microsoft.com/update-guide/vulnerability/CVE-2023-24932
- KB5025885: phases (9 May 2023; 11 Jul 2023; evaluation from 9 Apr 2024; deployment from July 2024, not enabled by default); enforcement "will not begin before January …: https://support.microsoft.com/kb/5025885
- LogoFAIL announced 29 Nov 2023; malicious images on the ESP or unsigned sections of a firmware update; Black Hat Europe; Wojtczuk and Tereshkin, Black Hat USA 2009, BMP …: https://www.binarly.io/blog/the-far-reaching-consequences-of-logofail
- CERT/CC VU#811862, first published 6 Dec 2023, "Image files in UEFI can be abused to modify boot behavior"; AMI, Insyde, Phoenix affected; Phoenix statement: https://kb.cert.org/vuls/id/811862
- PKfail (Binarly, 25 Jul 2024): "DO NOT TRUST" / "DO NOT SHIP" issuers; more than 10% of images, 8% for the past 4 years; first May 2012, latest June 2024; private key …: https://binarly.io/blog/pkfail-untrusted-platform-keys-undermine-secure-boot-on-uefi-ecosystem
- CERT/CC VU#455367 (30 Aug 2024): firmware update; PK replacement tool; Intel: affected products no longer supported: https://kb.cert.org/vuls/id/455367
- BootHole CVE-2020-10713 (Eclypsium, 29 Jul 2020): grub.cfg "not signed"; dbx needed; dbx updates offered manually first because of bricking risk: https://eclypsium.com/blog/theres-a-hole-in-the-boot/
- Microsoft ADV200011 released 29 Jul 2020: https://msrc.microsoft.com/update-guide/vulnerability/ADV200011
- SBAT: BootHole added 3 certificates and 150 hashes to dbx, 10 kB of 32 kB; revocation by generation: https://github.com/rhboot/shim/blob/main/SBAT.md
- LVFS used by all major Linux distributions; Dell 9667 firmware files, Lenovo ThinkPad 4203: https://fwupd.org/lvfs/vendors/
- Measurement: tianocore/edk2 MdeModulePkg/Library/BaseBmpSupportLib/BmpSupportLib.c added 22 Dec 2017 (4123d6da7095), last change 5 Dec 2021 (1436aea4d570); header text …: https://github.com/tianocore/edk2/commits/master/MdeModulePkg/Library/BaseBmpSupportLib/BmpSupportLib.c
- Intel Boot Guard, from 4th-generation Core platforms, verifies the initial boot block; the OEM key's public portion "is then programmed into field programmable fuses …: https://www.uefi.org/sites/default/files/resources/UEFI%20RoT%20white%20paper_Final%208%208%2016%20%28003%29.pdf
- AMD Platform Secure Boot sets one-time fuses that bind the processor to the OEM's firmware signing key (AMD statement to ServeTheHome; trade press): https://www.servethehome.com/amd-psb-vendor-locks-epyc-cpus-for-enhanced-security-at-a-cost/
- MSI breach: Binarly reported image-signing keys for 57 MSI products and Intel Boot Guard keys for 116 (later counts higher); Intel: Boot Guard OEM keys are generated by …: https://www.bleepingcomputer.com/news/security/intel-Investigating-leak-of-intel-boot-guard-private-keys-after-msi-breach/
- MSI breach: Binarly reported image-signing keys for 57 MSI products and Intel Boot Guard keys for 116 (later counts higher); Intel: Boot Guard OEM keys are generated by …: https://www.binarly.io/blog/leaked-msi-source-code-with-intel-oem-keys-how-does-this-affect-industry-wide-software-supply-chain
- ESET, "LoJax: First UEFI rootkit found in the wild" (27 to 28 September 2018): survives OS reinstallation and disk replacement; not properly signed, so Secure Boot is …: https://blog.eset.ie/2018/09/28/lojax-first-uefi-rootkit-found-in-the-wild-courtesy-of-the-sednit-group/
- x86 PCs certified for Windows must have Secure Boot on by default, trust Microsoft's certificate, and let the user disable Secure Boot: https://learn.microsoft.com/en-us/windows/security/operating-system-security/system-security/secure-the-windows-10-boot-process
- CVE-2024-7344 (ESET, 14 Jan 2025): a UEFI application signed under Microsoft Corporation UEFI CA 2011 could load unsigned code at boot; "Microsoft released official …: https://www.welivesecurity.com/en/eset-research/under-cloak-uefi-secure-boot-introducing-cve-2024-7344/
- CVE-2024-7344 (ESET, 14 Jan 2025): a UEFI application signed under Microsoft Corporation UEFI CA 2011 could load unsigned code at boot; "Microsoft released official …: https://kb.cert.org/vuls/id/529659

Chapter 5. Privileged Code: Kernels, Hypervisors, Drivers and the Morning of 19 July 2024

- Rapid Response Content update published 19 July 2024 at 04:09 UTC to Windows hosts running sensor 7.11 and above; crashes on systems online 04:09 to 05:27 UTC; Mac and …: https://www.crowdstrike.com/wp-content/uploads/2024/07/CrowdStrike-PIR-Executive-Summary.pdf
- Faulty channel file C-00000291*.sys timestamp 04:09 UTC; reverted version 05:27 UTC or later; more than 97% of Windows sensors online as of 24 July 2024, 5 pm PT, stated …: https://www.crowdstrike.com/falcon-content-update-remediation-and-guidance-hub/
- RCA dated 6 Aug 2024: IPC Template Type added with sensor 7.11 in February 2024; defined 21 input fields, integration code supplied 20; wildcard in 21st field in tests …: https://www.crowdstrike.com/wp-content/uploads/2024/08/Channel-File-291-Incident-Root-Cause-Analysis-08.06.2024.pdf
- Microsoft (David Weston, 20 July 2024): "We currently estimate that CrowdStrike's update affected 8.5 million Windows devices, or less than one percent of all Windows …: https://blogs.microsoft.com/blog/2024/07/20/helping-our-customers-through-the-crowdstrike-outage/
- Microsoft (27 July 2024): kernel drivers give visibility, early load and tamper resistance "at the cost of resilience"; crash analysis confirms "a read-out-of-bounds …: https://www.microsoft.com/en-us/security/blog/2024/07/27/windows-security-best-practices-for-integrating-and-managing-security-tools/
- Microsoft Recovery Tool (20 July 2024, one path; Safe Mode path added in the 21 July update): WinPE path may need the BitLocker recovery key; Safe Mode path may avoid it …: https://techcommunity.microsoft.com/blog/intunecustomersuccess/new-recovery-tool-to-help-with-crowdstrike-issue-impacting-windows-endpoints/4196959
- Windows Resiliency Initiative announced 19 Nov 2024; Quick Machine Recovery to execute targeted fixes from Windows Update; security products to run in user mode; private …: https://blogs.windows.com/windowsexperience/2024/11/19/windows-security-and-resiliency-protecting-your-business/
- Quick Machine Recovery generally available on Windows 11 24H2 (KB5062660, July 2025); on by default on Home, off by default on Pro, Education and Enterprise: https://techcommunity.microsoft.com/blog/windows-itpro-blog/resilience-in-action-for-windows-devices/4434571
- Windows endpoint security platform reached general availability on Windows 11 on 17 Nov 2026; user-mode APIs for file system, process, registry and Object Manager; …: https://techcommunity.microsoft.com/blog/windows-itpro-blog/whats-next-windows-endpoint-security-platform-and-driver-quality/4572663
- Same announcement on the Windows Experience Blog (17 Nov 2026): https://blogs.windows.com/windowsexperience/2026/11/17/windows-security-at-ignite-2026-start-secure-protect-what-matters-prepare-for-ai/
- Memory integrity (HVCI) moves kernel-mode code integrity into the VBS environment, which "assumes the kernel can be compromised"; on by default on many new Windows 11 …: https://learn.microsoft.com/en-us/windows/security/hardware-security/enable-virtualization-based-protection-of-code-integrity
- Driver signing: from Windows 10 1607 new kernel drivers must be signed by the Dev Portal; exceptions for upgrades, Secure Boot off, certificates issued before 29 July …: https://learn.microsoft.com/en-us/windows-hardware/drivers/install/kernel-mode-code-signing-policy--windows-vista-and-later-
- Vulnerable driver blocklist enabled by default for all devices since the Windows 11 2022 update; also enforced with memory integrity, Smart App Control or S mode; …: https://learn.microsoft.com/en-us/windows/security/application-security/application-control/app-control-for-business/design/microsoft-recommended-driver-block-rules
- Stuxnet drivers signed with a compromised Realtek certificate revoked 16 July 2010 by Verisign; JMicron-signed driver found 17 July 2010, certificate revoked 22 July …: https://www.cs.umd.edu/class/fall2023/cmsc614/papers/stuxnet-dossier.pdf
- ESET (30 Sept 2022): autumn 2021 attacks on an aerospace company employee in the Netherlands and a political journalist in Belgium; first recorded abuse of …: https://www.welivesecurity.com/2022/09/30/amazon-themed-campaigns-lazarus-netherlands-belgium/
- Apple: Kernel Integrity Protection; SPTM supported on A15 or later and M2 or later and replaces PPL; PPL in iOS, iPadOS, visionOS and watchOS, not macOS; TXM with SPTM: https://support.apple.com/guide/security/operating-system-integrity-sec8b776536b/web
- Apple: macOS 10.15 system extensions run in user space; DriverKit for USB and HID drivers; kexts not recommended; on Apple silicon kexts need Reduced Security and …: https://support.apple.com/guide/security/securely-extending-the-kernel-sec8e454101b/web
- Apple Platform Deployment: endpoint security tools among agents built without kernel extensions; MDM approval of system extensions: https://support.apple.com/guide/deployment/system-and-kernel-extensions-in-macos-depa5fb8376f/web
- Android pKVM isolates pVMs and Android into mutually distrusted environments; host cannot access pVM pages without sharing; SoC vendors must meet requirements: https://source.android.com/docs/core/virtualization/security
- Linux 5.4 released 24 Nov 2019 with the lockdown LSM to strengthen the boundary between root and the kernel: https://kernelnewbies.org/Linux_5.4

- Linux Security Modules framework and documented LSMs (SELinux, AppArmor, Landlock and others): https://docs.kernel.org/admin-guide/LSM/index.html
- eBPF verifier: control-flow check, then descent of "all possible paths": https://docs.kernel.org/bpf/verifier.html
- Linux 6.1 released 11 Dec 2022 with initial support for Rust: https://kernelnewbies.org/Linux_6.1
- Linux kernel CVE process: team assigns CVEs to any bug fix it identifies; none for unfixed issues or unsupported versions: https://docs.kernel.org/process/cve.html
- Linux kernel became a CVE Numbering Authority, February 2024: https://openssf.org/blog/2024/02/14/linux-kernel-achieves-cve-numbering-authority-status/
- Android memory-safety share 76% in 2019 and 24% in 2024; 2024 extrapolated to full year (Google, 25 Sept 2024): https://security.googleblog.com/2024/09/eliminating-memory-safety-vulnerabilities-Android.html
- Android memory-safety vulnerabilities below 20% of total in 2025; Android 6.12 kernel first with Rust support enabled and a production Rust driver (Google, 13 Nov 2025): https://blog.google/security/rust-in-android-move-fast-fix-things/
- Measurement: LOLDrivers repository, commit 39c70ca64ae9b66650e2c50e011eec6f7354e9cf (27 Oct 2026), file loldrivers.io/content/api/drivers.json: 713 entries (588 ...: https://github.com/magicsword-io/LOLDrivers/blob/39c70ca64ae9b66650e2c50e011eec6f7354e9cf/loldrivers.io/content/api/drivers.json
- Sensor Update Policies select the latest sensor, N-1 or N-2; Sensor Content ships only with sensor releases, Rapid Response Content as separate configuration updates; ...: https://www.crowdstrike.com/en-us/blog/falcon-content-update-preliminary-post-incident-report/
- Content update policies: ring assignment per content category (ga, ea, pause); pause not allowed for system-critical content: https://developer.crowdstrike.com/cac/terraform/provider/content-update-policy/resources/content-update-policy/
- Windows Resiliency Initiative (26 June 2025): MVI 3.0 partners must test incident response and follow safe deployment practices; "Security product updates must be ...: https://blogs.windows.com/windowsexperience/2025/06/26/the-windows-resiliency-initiative-building-resilience-for-a-future-ready-enterprise/
- eBPF bounded loops: verifier support merged for Linux 5.3 (merge commit 94079b64255f, 'bpf-bounded-loops'): https://lwn.net/Articles/795402
- Red Hat knowledge base: "Kernel panic observed after booting 5.14.0-427.13.1.el9_4.x86_64 by falcon-sensor process"; "eBPF program causes kernel panic on kernels ...: https://access.redhat.com/solutions/7068083
- Microsoft Defender Antivirus: monthly platform and engine updates roll out gradually through update channels; security intelligence updates are separate: https://learn.microsoft.com/en-us/defender-endpoint/manage-gradual-rollout
- eBPF: the name began as "extended Berkeley Packet Filter": https://ebpf.io/what-is-ebpf/

Chapter 6. Identity and Keys: Who Can Sign, Who Can Log In

- CSRB report dated 20 March 2024: 22 organizations and over 500 individuals compromised in May and June 2023; actor assessed affiliated with the PRC for espionage; tokens ...: https://www.cisa.gov/sites/default/files/2025-03/CSRBReviewOfTheSummer2023MEOIntrusion508.pdf
- DHS released the CSRB report on 2 April 2024: https://www.dhs.gov/news/2024/04/02/cyber-safety-review-board-releases-report-microsoft-online-exchange-incident-summer
- MSRC (11 July 2023): Storm-0558 access beginning 15 May 2023; about 25 organizations; forged tokens with an acquired MSA consumer signing key; tokens blocked and key ...: https://www.microsoft.com/en-us/msrc/blog/2023/07/microsoft-mitigates-china-based-threat-actor-storm-0558-targeting-of-customer-email
- MSRC (6 Sept 2023): April 2021 crash-dump hypothesis; common key metadata endpoint introduced September 2018; libraries did not check key scope; mail system accepted ...: https://www.microsoft.com/en-us/msrc/blog/2023/09/results-of-major-technical-investigations-for-storm-0558-key-acquisition
- DHS memo of 20 January 2025 terminated advisory committee memberships, including the CSRB (secondary report quoting the memo): https://www.nextgov.com/cybersecurity/2025/01/dhs-cyber-review-board-cleaned-out-trump-move-eliminate-misuse-resources/402386
- Uber security update (15 to 19 Sept 2022): EXT contractor account; password likely bought on the dark web after malware on a personal device; repeated two-factor ...: https://www.uber.com/newsroom/security-update/
- MGM Resorts Form 8-K (5 Oct 2023): about $100 million negative impact to Adjusted Property EBITDAR in September; less than $10 million one-time expenses in Q3; insurance ...: https://www.sec.gov/Archives/edgar/data/789570/000119312523251667/d461062d8k.htm
- Attackers' claim relayed by a research group: an employee found on LinkedIn and a call to the help desk; MGM did not confirm (secondary, labeled): https://www.securityweek.com/ransomware-gang-takes-credit-for-highly-disruptive-mgm-resorts-attack/

- FBI and CISA advisory AA23-320A (16 Nov 2023, revised 29 July 2025): calls to convince help-desk staff to reset passwords and/or transfer MFA tokens; MFA fatigue; SIM ...: https://www.cisa.gov/news-events/cybersecurity-advisories/aa23-320a
- Okta (20 Oct 2023): stolen credential used to access the support case management system; HAR files may contain session tokens: https://sec.okta.com/articles/2023/10/tracking-unauthorized-access-oktas-support-system/
- Okta root cause (3 Nov 2023): access 28 Sept to 17 Oct 2023; 134 customers (less than 1%); sessions of 5 customers hijacked; service account exposed via an employee's ...: https://sec.okta.com/articles/2023/11/unauthorized-access-oktas-support-case-management-system-root-cause
- Okta (8 Feb 2024): investigation closed; IP binding, MFA for protected admin actions: https://sec.okta.com/harfiles
- FIDO Alliance: passkey definition; phishing resistant; FIDO2 = WebAuthn and CTAP; synced versus device-bound passkeys: https://fidoalliance.org/passkeys/
- Windows Hello for Business: key pair bound to the TPM if available or in software; policy needed to guarantee hardware keys; Windows Hello usable as a FIDO2 authenticator: https://learn.microsoft.com/en-us/windows/security/identity-protection/hello-for-business/how-it-works
- Windows 11 hardware requirement: TPM 2.0: https://learn.microsoft.com/en-us/windows/whats-new/windows-11-requirements
- Apple Secure Enclave: dedicated secure subsystem integrated into Apple SoCs: https://support.apple.com/guide/security/secure-enclave-sec59b0b31ff/web
- Android Keystore: key material never enters the app process; StrongBox on devices with Android 9 or higher, with its own CPU, secure storage and true random-number ...: https://developer.android.com/privacy-and-security/keystore
- Android Keystore and TEE-backed KeyMint: https://source.android.com/docs/security/features/keystore
- CA/Browser Forum code-signing baseline requirements (v3.11.0): from 2023-06-01 subscriber private keys generated, stored and used in a suitable hardware crypto module: https://cabforum.org/working-groups/code-signing/requirements/
- Okta (29 Nov 2023): the actor ran and downloaded a report with "the names and email addresses of all Okta customer support system users"; all Workforce Identity Cloud ...: https://sec.okta.com/october-security-incident-recommended-actions
- Microsoft (25 Jan 2024): Midnight Blizzard attack detected 12 January 2024; password spray compromised "a legacy, non-production test tenant account" that "did not have ...: https://www.microsoft.com/en-us/security/blog/2024/01/25/midnight-blizzard-guidance-for-responders-on-nation-state-attack/
- Mandiant (10 June 2024): UNC5537 accessed Snowflake customer instances with credentials stolen by infostealer malware; about 165 potentially exposed organizations ...: https://cloud.google.com/blog/topics/threat-intelligence/unc5537-snowflake-data-theft-extortion
- Marks and Spencer full-year results (21 May 2025): cyber incident expected to cost about £300m of 2025/26 operating profit, before mitigation, insurance and trading ...: https://corporate.marksandspencer.com/media/press-releases/full-year-results-52-weeks-ended-29-march-2025
- M&S chairman's oral evidence to the House of Commons Business and Trade sub-committee (8 July 2025): entry through "sophisticated impersonation" in which a third party ...: https://committees.parliament.uk/event/24543
- Microsoft (12 July 2022): adversary-in-the-middle phishing proxies the sign-in and steals the session cookie, letting attackers in "even if the target's MFA is enabled"; ...: https://www.microsoft.com/en-us/security/blog/2022/07/12/from-cookie-theft-to-bec-attackers-use-aitm-phishing-sites-as-entry-point-to-further-financial-fraud/
- Microsoft (19 July 2023): Audit (Standard) customers to receive detailed email-access logs and more than 30 other log types, rolling out from September 2023; default ...: https://www.microsoft.com/en-us/security/blog/2023/07/19/expanding-cloud-logging-to-give-customers-deeper-security-visibility/
- Microsoft (20 May 2024): new Standard logs including MailItemsAccessed available in public preview to Worldwide and Gov cloud customers: https://techcommunity.microsoft.com/t5/security-compliance-and-identity/increased-security-visibility-through-new-standard-logs-in/ba-p/4144454
- Microsoft (SFI April 2025 progress report): in September 2024 Entra ID and MSA access token signing keys moved "to hardware-based security modules (HSMs) and ...: https://www.microsoft.com/en-us/security/blog/2025/04/21/securing-our-future-april-2025-progress-report-on-microsofts-secure-future-initiative/
- Cloudflare (9 August 2022): at least 76 employees received phishing texts, three entered credentials; attackers "could not get past the hard key requirement" (FIDO2 ...: https://blog.cloudflare.com/2022-07-sms-phishing-attacks/
- Microsoft Learn: "Number matching is enabled for all Authenticator push notifications": https://learn.microsoft.com/en-us/entra/identity/authentication/how-to-mfa-number-match

Chapter 7. The Management Plane: Everyone Who Can Push

- Stryker identified a cybersecurity incident on 11 March 2026 that "resulted in a global disruption to the Company's Microsoft environment"; "no indication of ransomware ...: https://www.sec.gov/Archives/edgar/data/310764/000119312526102460/d76279d8k.htm

- Stryker 8-K (12 Mar 2026): operations disrupted, including order processing, manufacturing and shipping; connected products not believed impacted: https://www.sec.gov/Archives/edgar/data/310764/000119312526104431/d101097d8k.htm
- Stryker 8-K (23 Mar 2026): disruption to the corporate network "including but not limited to the Microsoft environment"; the threat actor "used a malicious file to run …: https://www.sec.gov/Archives/edgar/data/310764/000119312526118634/d94012d8k.htm
- Unit 42 letter (20 Mar 2026, Exhibit 99.1): compromise "involving impacts to Stryker's Entra ID environment, servers, and workstations": https://www.sec.gov/Archives/edgar/data/310764/000119312526118634/d94012dex991.htm
- Stryker 8-K/A (9 Apr 2026), Item 1.05: material impact on operations and on first-quarter 2026 results; fully operational as of the amendment: https://www.sec.gov/Archives/edgar/data/310764/000119312526149607/d112875d8ka.htm
- Press reports (secondary) that attackers used the device-management console's remote-wipe command; device counts reported vary and are not used: https://www.bleepingcomputer.com/news/security/medtech-giant-stryker-offline-after-iran-linked-wiper-malware-attack/
- CISA alert (18 Mar 2026): activity "targeting endpoint management systems of U.S. organizations" that "misuses legitimate endpoint management software"; recommends …: https://www.cisa.gov/news-events/alerts/2026/03/18/cisa-urges-endpoint-management-system-hardening-after-cyberattack-against-us-organization
- Intune Multi Admin Approval: protected resources (apps, compliance and configuration policies, wipe/retire/delete, RBAC, Windows scripts, device categories); approver …: https://learn.microsoft.com/en-us/intune/intune-service/fundamentals/multi-admin-approval
- Play Integrity API verdicts come from Google Play; developers told to plan for "the unlikely event of a large-scale Play Integrity API outage": https://developer.android.com/google/play/integrity/overview
- Kaseya learned of the incident on Friday 2 July 2021; told on-premises customers to shut down VSA servers; SaaS servers taken offline as a precaution: https://helpdesk.kaseya.com/hc/en-gb/articles/4403440684689-Important-Notice-July-2nd-2021
- CISA (12 July 2021): Kaseya released VSA 9.5.7a for on-premises customers: https://www.cisa.gov/news-events/alerts/2021/07/12/kaseya-provides-security-updates-vsa-premises-software-vulnerabilities
- DIVD case: Kaseya informed 6 April 2021; fixes on 10 April (9.5.5) and 8 May (9.5.6); 26 June 2021 Kaseya released 9.5.7 on SaaS resolving CVE-2021-30116 and …: https://csirt.divd.nl/cases/DIVD-2021-00011/
- CVE-2021-30116: unauthenticated credential leak and business logic flaw in VSA before 9.5.7, exploited in the wild in July 2021: https://csirt.divd.nl/cves/CVE-2021-30116/
- CISA-FBI guidance (4 July 2021): MFA on MSP accounts, restrict RMM traffic to known IP pairs, RMM admin interfaces behind VPN or firewall: https://www.cisa.gov/news-events/alerts/2021/07/04/cisa-fbi-guidance-msps-and-their-customers-affected-kaseya-vsa-supply-chain-ransomware-attack
- CISA, NSA, MS-ISAC advisory AA23-025A (25 Jan 2023) on malicious use of legitimate RMM software: https://www.cisa.gov/news-events/cybersecurity-advisories/aa23-025a
- ConnectWise ScreenConnect 23.9.8 bulletin (19 Feb 2024): reported 13 Feb 2024; 23.9.7 and prior affected; cloud instances mitigated within 48 hours; on-premises must …: https://www.connectwise.com/company/trust/security-bulletins/connectwise-screenconnect-23.9.8
- CISA added CVE-2024-1709 (authentication bypass creating an administrator account) to KEV on 22 Feb 2024: https://www.cisa.gov/news-events/alerts/2024/02/22/cisa-adds-one-known-exploited-connectwise-vulnerability-cve-2024-1709-catalog
- BIS Final Determination on Kaspersky announced 20 June 2024; three Kaspersky entities added to the Entity List; continued users "will not face legal penalties"; …: https://media.bis.gov/press-release/commerce-department-prohibits-russian-kaspersky-software-u.s.-customers
- From 12:00 AM EDT 20 July 2024 no new agreements with US persons; from 12:00 AM EDT 29 September 2024 no "anti-virus signature updates and codebase updates": https://www.bis.gov/kaspersky
- Federal Register, Final Determination, Case No. ICTS-2021-002, Kaspersky Lab, Inc. (24 June 2024): https://www.federalregister.gov/documents/2024/06/24/2024-13532/final-determination-case-no-icts-2021-002-kaspersky-lab-inc
- Kaspersky statement (secondary, TechCrunch, 26 Sept 2024): migration began early September; on Windows "done automatically" so users "would not experience a gap in …: https://techcrunch.com/2024/09/26/kaspersky-defends-automatically-force-replacing-its-software-without-users-explicit-consent
- Apple: Erase Device MDM command; Mac conditions (Apple silicon with bootstrap token or T2): https://support.apple.com/guide/deployment/erase-devices-dep0a819891e/web
- Apple: device management command list; supervision through Automated Device Enrollment or Apple Configurator: https://support.apple.com/guide/deployment/device-management-command-list-dep789n2k1qp/web
- Systems Management Server officially launched 7 November 1994 (Microsoft, "ConfigMgr at 25"): https://www.microsoft.com/en-us/microsoft-365/blog/2017/09/26/configmgr-25/

- Apple previewed iPhone OS 4 on 8 April 2010 with Mobile Device Management to "wirelessly configure, query and even wipe or lock managed iPhones": https://www.apple.com/newsroom/2010/04/08Apple-Previews-iPhone-OS-4/
- CVE-2025-59287, WSUS: "allows an unauthorized attacker to execute code over a network"; CVSS 3.1 9.8; published 14 Oct 2025; added to CISA KEV 24 Oct 2025: https://nvd.nist.gov/vuln/detail/CVE-2025-59287
- CISA (24 Oct 2025): out-of-band update released 23 October 2025 for a WSUS flaw "that a prior update did not fully mitigate"; KEV addition 24 October 2025: https://www.cisa.gov/news-events/alerts/2025/10/24/microsoft-releases-out-band-security-update-mitigate-windows-server-update-service-vulnerability-cve

Chapter 8. The Software Supply Chain: Signed Is Not Safe

- Freund, "backdoor in upstream xz/liblzma leading to ssh server compromise", oss-security, 29 Mar 2024 08:51 -0700; failed login 0.299 s before, 0.807 s after; backdoor …: https://www.openwall.com/lists/oss-security/2024/03/29/4
- Debian (29 Mar 2024): "no Debian stable versions are known to be affected"; compromised packages in testing, unstable, experimental from 5.5.1alpha-0.1 (uploaded …: https://lists.debian.org/debian-security-announce/2024/msg00057.html
- Red Hat (29 Mar 2024, updated 30 Mar): Fedora Rawhide and Fedora 40 beta received affected versions; "No versions of Red Hat Enterprise Linux (RHEL) are affected": https://www.redhat.com/en/blog/urgent-security-alert-fedora-41-and-rawhide-users
- XZ Utils project: 5.6.0 and 5.6.1 tarballs contain a backdoor and "were created and signed by Jia Tan"; clean releases 29 May 2024: https://tukaani.org/xz-backdoor/
- Measurement: github.com/tukaani-project/xz at f54febc540 (18 Nov 2026): first commit authored "Jia Tan" dated 2022-05-05 (the account's earliest, as "jiat75", …: https://github.com/tukaani-project/xz
- SolarWinds Form 8-K (14 Dec 2020): vulnerability in updates released March to June 2020; introduced by a compromise of the Orion build system and "not present in the …: https://www.sec.gov/Archives/edgar/data/1739942/000162828020017451/swi-20201214.htm
- SolarWinds, "New Findings From Our Investigation of SUNBURST" (11 Jan 2021): timeline begins September 2019; October 2019 release tested code insertion; SUNBURST …: https://www.solarwinds.com/blog/new-findings-from-our-investigation-of-sunburst
- CISA AA20-352A: malicious DLL "signed by the legitimate SolarWinds code signing certificate"; activity beginning in at least March 2020; ED 21-01 released 13 Dec 2020: https://www.cisa.gov/news-events/cybersecurity-advisories/aa20-352a
- US Treasury (15 Apr 2021) names the SVR as responsible for the SolarWinds Orion exploit: https://home.treasury.gov/news/press-releases/jy0127
- UK NCSC (15 Apr 2021): "highly likely" the SVR was responsible: https://www.ncsc.gov.uk/news/uk-and-us-call-out-russia-for-solarwinds-compromise
- SEC Litigation Release 26423: SEC v. SolarWinds and Timothy G. Brown dismissed with prejudice by joint stipulation (20 Nov 2025), No. 1:23-cv-09518 (S.D.N.Y.); most …: https://www.sec.gov/enforcement-litigation/litigation-releases/lr-26423
- CSRB, Review of the December 2021 Log4j Event (11 July 2022): reported 24 Nov 2021 by an Alibaba Cloud Security engineer; fix committed 5 Dec; 2.15.0-rc1 tagged 6 Dec; …: https://www.cisa.gov/sites/default/files/2023-02/CSRB-Report-on-Log4j-PublicReport-July-11-2022-508-Compliant.pdf
- Apache Log4j security page: CVE-2021-44228 fixed in 2.15.0 (Java 8), 2.12.2 (Java 7), 2.3.1 (Java 6); reported by Chen Zhaojun, Alibaba Cloud Security Team: https://logging.apache.org/security.html
- CISA ED 22-02 (17 Dec 2021): inventory by 23 Dec, reports by 28 Dec; later closed and moved to BOD 22-01: https://www.cisa.gov/news-events/directives/ed-22-02-mitigate-apache-log4j-vulnerability-closed
- Debian DSA-5020-1 (11 Dec 2021): apache-log4j2 fixed in 2.15.0-1~deb11u1 (bullseye): https://lists.debian.org/debian-security-announce/2021/msg00206.html
- VMSA-2021-0028: initial 10 Dec 2021; workaround information added 11 Dec; revisions through .13 on 14 Feb 2022; revision .10 (27 Jan 2022) updated vCenter Server: https://support.broadcom.com/web/ecx/support-content-notification/-/external/content/SecurityAdvisories/0/23618
- Measurement: github.com/apache/logging-log4j2, 2.x head 572cd83040 (18 Nov 2026): JndiLookup.java added f1a0cac60f (2013-07-18, "LOG4J2-313 - Add JNDILookup"); present …: https://github.com/apache/logging-log4j2
- Mandiant (20 Apr 2023): 3CX compromised via trojanized X_TRADER installer still downloadable from Trading Technologies' site in 2022, signed with a certificate expiring …: https://cloud.google.com/blog/topics/threat-intelligence/3cx-software-supply-chain-compromise
- CrowdStrike (29 Mar 2023): malicious activity from the legitimate signed 3CXDesktopApp on Windows and macOS: https://www.crowdstrike.com/en-us/blog/crowdstrike-detects-and-prevents-active-intrusion-campaign-targeting-3cxdesktopapp-customers/

- Trading Technologies could not verify Mandiant's assertions (secondary): https://techcrunch.com/2023/04/20/3cx-supply-chain-xtrader-mandiant
- Cisco Talos, "The MeDoc Connection" (July 2017): NotPetya (Nyetya) installations came through the M.E.Doc update system: https://blog.talosintelligence.com/the-medoc-connection/
- CISA (23 Sept 2025): "Shai-Hulud" npm worm, over 500 packages; credential harvesting, self-publishing; pin to releases before 16 Sept 2025, rotate credentials, ...: https://www.cisa.gov/news-events/alerts/2025/09/23/widespread-supply-chain-compromise-impacting-npm-ecosystem
- SLSA v1.0 released 19 April 2023: https://slsa.dev/blog/2023/04/slsa-v1-final
- SLSA v1.0 threats: "does not address source threats"; source threats include "an authorized individual introducing an unauthorized change": https://slsa.dev/spec/v1.0/threats
- Thompson, "Reflections on Trusting Trust", Commun. ACM 27(8), Aug 1984, 761 to 763; "You can't trust code that you did not totally create yourself.": https://www.cl.cam.ac.uk/teaching/2223/R209/Reflections-Trusting-Trust.pdf
- FBI: Morris worm released 2 November 1988 from an MIT computer; about 6,000 of about 60,000 internet computers hit within 24 hours: https://www.fbi.gov/history/cases-and-criminals/morris-worm
- macOS Gatekeeper checks that software from outside the App Store comes from an identified developer, is notarized and is unaltered; Apple issues notarization tickets ...: https://support.apple.com/guide/security/protecting-against-malware-sec469d47bd8/web
- macOS app code signing process (Developer ID, notarization): https://support.apple.com/guide/security/app-code-signing-process-sec3ad8e6e53/web
- iOS apps distributed through alternative channels in the EU pass Apple notarization (a review applying to all apps whatever the channel): https://developer.apple.com/support/dma-and-apps-in-the-eu/
- Play App Signing: Google manages and protects the app signing key; the developer holds an upload key: https://support.google.com/googleplay/android-developer/answer/9842756
- Android app signing by the developer's key: https://developer.android.com/studio/publish/app-signing
- Public analysis of the xz backdoor: the payload "must be encrypted and signed with the attacker's ED448 key"; the backdoor validates against a hardcoded Ed448 public key: https://github.com/amlweems/xzbot
- CISA (18 Mar 2025): tj-actions/changed-files compromised (CVE-2025-30066), affected runs 12 Mar 00:00 UTC to 15 Mar 12:00 UTC; secrets such as access keys, GitHub ...: https://cisa.gov/news-events/alerts/2025/03/18/supply-chain-compromise-third-party-github-action-cve-2025-30066
- Datadog Security Labs: Shai-Hulud 2.0 identified 24 November 2025; at least 796 unique npm packages (1,092 versions) backdoored through an added preinstall script: https://securitylabs.datadoghq.com/articles/shai-hulud-2.0-npm-worm/
- GitHub (22 Sept 2025): removal of 500+ compromised packages; plan to deprecate legacy classic tokens, limit granular token lifetimes to seven days, expand trusted ...: https://github.blog/security/supply-chain-security/our-plan-for-a-more-secure-npm-supply-chain/
- CISA, Minimum Requirements for Vulnerability Exploitability eXchange (VEX) (21 April 2023): https://www.cisa.gov/resources-tools/resources/minimum-requirements-vulnerability-exploitability-exchange-vex

Chapter 9. When Trust Must Be Withdrawn: Stop, Revoke, Roll Back, Recover

- NIST SP 800-40 Rev. 4, "Guide to Enterprise Patch Management Planning: Preventive Maintenance for Technology", final 6 April 2022; "frames patching as a critical ...: https://csrc.nist.gov/pubs/sp/800/40/r4/final
- Windows Update client policies: defer quality updates "for a period of up to 30 days"; pause "for a period of up to 35 days": https://learn.microsoft.com/en-us/windows/deployment/update/waas-configure-wufb
- Hotpatch for Windows 11 Enterprise 24H2 generally available on Intel and AMD devices 2 April 2025 (Arm64 on 8 July 2025): https://techcommunity.microsoft.com/blog/windows-itpro-blog/hotpatch-for-windows-client-now-available/4399808
- Hotpatch plan: "eight hotpatch months, and four baseline months, planned each year"; baseline months need a restart: https://learn.microsoft.com/en-us/windows/deployment/windows-autopatch/manage/windows-autopatch-hotpatch-updates
- Apple MDM: "a custom deferral period from 1 to 90 days" on supervised devices: https://support.apple.com/guide/deployment/install-and-enforce-software-updates-depd30715cbb/web
- Android 10 introduced Mainline; devices "can receive Mainline updates from the Android Google Play system update feature": https://source.android.com/docs/core/ota/modular-system
- Linux livepatch: "allowing for function calls to be redirected; thus, fixing critical functions without a system reboot": https://docs.kernel.org/livepatch/livepatch.html
- Play Console: halting a staged rollout means "no additional users will receive the app version": https://support.google.com/googleplay/android-developer/answer/6346149

- Background Security Improvements supported from iOS, iPadOS and macOS 26.1: https://support.apple.com/en-us/126604
- First Background Security Improvements 17 March 2026 (iOS/iPadOS 26.3.1 (a), macOS 26.3.1 (a) and 26.3.2 (a)), WebKit, CVE-2026-20643: https://support.apple.com/en-us/102657
- Users "can also elect to remove all Background Security Improvements currently applied on their device": https://support.apple.com/guide/security/rapid-security-responses-sec87fc038c2/web
- Rapid Security Response iOS 16.5.1 (a), 10 July 2023 (WebKit, CVE-2023-37450); replaced by 16.5.1 (c) on 12 July 2023 "to fix an issue that prevents some websites from …: https://support.apple.com/en-us/106354
- Android A/B updates: if an update "fails to boot, the device will reboot back into the old partition and remains usable": https://source.android.com/docs/core/ota/ab
- Android Verified Boot rollback protection: refuses to boot a version lower than the recorded one: https://source.android.com/docs/security/features/verifiedboot/verified-boot
- shim SBAT: "a mechanism to require a specific level of resistance to UEFI Secure Boot bypasses": https://github.com/rhboot/shim/blob/main/SBAT.md
- CrowdStrike Channel File 291: released 19 July 2024 04:09 UTC, reverted 05:27 UTC: https://www.crowdstrike.com/blog/falcon-content-update-preliminary-post-incident-report/
- CrowdStrike: "~99% of Windows sensors are online as of July 29 at 5pm PT" (sensor connectivity, not business recovery): https://www.crowdstrike.com/falcon-content-update-remediation-and-guidance-hub/
- Microsoft manual fix KB5042421: "You may be asked to enter your BitLocker recovery key.": https://support.microsoft.com/en-us/topic/b1c700e0-7317-4e95-aeee-5d67dd35b92f
- Microsoft recovery tool KB5042429, released 20 July 2024, two repair options (WinPE and safe mode) on bootable media: https://techcommunity.microsoft.com/blog/intunecustomersuccess/new-recovery-tool-to-help-with-crowdstrike-issue-impacting-windows-endpoints/4196959
- KB5025885 (CVE-2023-24932): initial deployment 9 May 2023, second 11 July 2023, evaluation from 9 April 2024, enforcement "will not begin before January 2026", no date …: https://support.microsoft.com/en-us/topic/how-to-manage-the-windows-boot-manager-revocations-for-secure-boot-changes-associated-with-cve-2023-24932-41a975df-beb2-40c1-99a3-b3ff139f832d
- Storm-0558: "Microsoft revoked all MSA signing [keys] which were valid at the time of the incident, including the actor-acquired MSA key" (14 July 2023): https://www.microsoft.com/en-us/security/blog/2023/07/14/analysis-of-storm-0558-techniques-for-unauthorized-email-access/
- CircleCI, 4 January 2023: "Immediately rotate any and all secrets stored in CircleCI"; all GitHub OAuth tokens rotated on customers' behalf by 7 January 2023: https://circleci.com/blog/january-4-2023-security-alert/
- BitLocker recovery keys stored in Microsoft Entra ID (Entra-joined) or AD DS (domain-joined): https://learn.microsoft.com/en-us/windows/security/operating-system-security/data-protection/bitlocker/recovery-overview
- FileVault: "escrow the personal recovery key with your device management service": https://support.apple.com/guide/deployment/manage-filevault-with-device-management-dep0a2cb7686/web
- macOS recoveryOS: "Full Security and Reduced Security can be set using Startup Security Utility from recoveryOS": https://support.apple.com/guide/security/startup-disk-security-policy-control-sec7d92dc49f/web
- Quick Machine Recovery generally available on Windows 11 24H2 with KB5062660, July 2025: https://techcommunity.microsoft.com/blog/windows-itpro-blog/resilience-in-action-for-windows-devices/4434571
- QMR uses "a secure and connected Windows Recovery Environment (WinRE) to scan Windows Update for remediation options"; "Cloud remediation is disabled by default on …: https://learn.microsoft.com/en-us/windows/configuration/quick-machine-recovery/
- Windows 98 generally available 25 June 1998 with Windows Update, "a centralized Web-based resource site": https://news.microsoft.com/source/1998/04/14/microsoft-announces-windows-98-is-scheduled-to-be-available-on-june-25/
- 1 October 2003: Microsoft announced a monthly security bulletin cadence; first monthly release October 2003: https://www.microsoft.com/en-us/msrc/blog/2013/10/10-years-of-update-tuesdays
- Windows XP SP2 released to manufacturers 6 August 2004, delivered through Automatic Updates; users told to "simply turn on the Automatic Updates feature": https://news.microsoft.com/source/2004/08/06/microsoft-releases-windows-xp-service-pack-2-with-advanced-security-technologies-to-computer-manufacturers/
- SafeBreach (Alon Leviev), Windows Downdate: takes over Windows Update to downgrade system components while the machine "will appear fully up-to-date"; presented at Black …: https://www.safebreach.com/blog/downgrade-attacks-using-windows-updates/

- Microsoft KB5042562 (first published 13 Aug 2024): CVE-2024-21302 mitigation is a Microsoft-signed revocation policy (SkuSiPolicy.p7b); once applied with Secure Boot on, …: https://support.microsoft.com/en-us/topic/guidance-for-blocking-rollback-of-virtualization-based-security-vbs-related-security-updates-b2e7ebf4-f64d-4884-a390-38d63171b8d3
- CircleCI incident report (12 Jan 2023): attacker "leveraged malware deployed to a CircleCI engineer's laptop in order to steal a valid, 2FA-backed SSO session": https://circleci.com/blog/jan-4-2023-incident-report/
- CA/Browser Forum ballot SC-081v3: voting ended 11 April 2025, passed; maximum TLS certificate validity reduced from 398 days to 47 days in steps from March 2026 to March …: https://cabforum.org/2025/04/11/ballot-sc081v3-introduce-schedule-of-reducing-validity-and-data-reuse-periods/
- CrowdStrike root cause analysis (August 2024): "New capabilities have been implemented and deployed to our cloud that allow customers to control how Rapid Response …: https://crowdstrike.com/wp-content/uploads/2024/08/Executive-Summary_Root-Cause-Analysis_Channel-File-291.pdf
- Certificate Transparency: public, append-only logs of issued TLS certificates (IETF RFC 9162): https://www.rfc-editor.org/rfc/rfc9162

Chapter 10. Zero-Click: From the Screen to the Silicon

- Kaspersky, 1 June 2023: found while "monitoring the network traffic of our own corporate Wi-Fi network dedicated for mobile devices" (KUMA); "Without any user …: https://securelist.com/operation-triangulation/109842/
- Kaspersky (B. Larin), 27 December 2023: chain "used four zero-days and was designed to work on iOS versions up to iOS 16.2"; 37C3 talk "Operation Triangulation: What You …: https://securelist.com/operation-triangulation-the-last-hardware-mystery/111669/
- Apple PPL "is designed to prevent user space code from being modified after code signature verification is complete": https://support.apple.com/guide/security/operating-system-integrity-sec8b776536b/web
- iOS 16.3 (23 January 2023): FontParser CVE-2023-41990, "Processing a font file may lead to arbitrary code execution", credit Apple, "Entry added September 8, 2023": https://support.apple.com/en-us/HT213606
- iOS 16.4 (27 March 2023): WebKit CVE-2023-32435, entry added 21 June 2023, credit Kaspersky researchers: https://support.apple.com/en-us/HT213676
- iOS 16.5.1 and iOS 15.7.7 (21 June 2023): Kernel CVE-2023-32434; "Apple is aware of a report that this issue may have been actively exploited against versions of iOS …: https://support.apple.com/en-us/HT213814
- iOS 16.5.1 and iOS 15.7.7 (21 June 2023): Kernel CVE-2023-32434; "Apple is aware of a report that this issue may have been actively exploited against versions of iOS …: https://support.apple.com/en-us/HT213811
- iOS 16.6 and iOS 15.7.8 (24 July 2023): Kernel CVE-2023-38606, "An app may be able to modify sensitive kernel state"; iOS 15.7.8 also fixes CVE-2023-41990: https://support.apple.com/en-us/HT213841
- iOS 16.6 and iOS 15.7.8 (24 July 2023): Kernel CVE-2023-38606, "An app may be able to modify sensitive kernel state"; iOS 15.7.8 also fixes CVE-2023-41990: https://support.apple.com/en-us/HT213842
- Project Zero (Beer, Groß), 31 March 2022: FORCEDENTRY sandbox escape from IMTranscoderAgent to "arbitrary code execution in the context of CommCenter", "notable for …: https://projectzero.google/2022/03/forcedentry-sandbox-escape.html
- Citizen Lab, 24 August 2016, Trident: "The one-click vector involves sending the target a normal SMS text message with a link to a malicious website.": https://citizenlab.ca/2016/08/million-dollar-dissident-iphone-zero-day-nso-group-uae/
- iOS 9.3.5 (25 August 2016): CVE-2016-4655, CVE-2016-4656 (Kernel), CVE-2016-4657 (WebKit), credit Citizen Lab and Lookout: https://support.apple.com/en-us/HT207107
- BlastDoor: "a new, tightly sandboxed 'BlastDoor' service which is now responsible for almost all parsing of untrusted data in iMessages" (28 January 2021): https://projectzero.google/2021/01/a-look-at-imessage-in-ios-14.html
- Lockdown Mode: "Most message attachment types are blocked, other than certain images, video, and audio.": https://support.apple.com/en-us/105120
- Lockdown Mode announced 6 July 2022; JIT JavaScript compilation named among limited web technologies: https://www.apple.com/newsroom/2022/07/apple-expands-commitment-to-protect-users-from-mercenary-spyware/
- MTE documented in the Arm v8.5 Architecture Reference Manual (Google, August 2019): https://security.googleblog.com/2019/08/adopting-arm-memory-tagging-extension.html
- Pixel 8: "there's finally a production handset that allows you to enable MTE!"; developer option with Android 14: https://projectzero.google/2023/11/first-handset-with-mte-on-market.html
- Pixel 8: "there's finally a production handset that allows you to enable MTE!"; developer option with Android 14: https://security.googleblog.com/2023/11/mte-promising-path-forward-for-memory.html

- Android 16 Advanced Protection: "For supported apps, MTE will help prevent apps from corrupting memory.": https://support.google.com/android/answer/16339980
- Apple Memory Integrity Enforcement, 9 September 2025: "always-on memory-safety protection covering key attack surfaces, including the kernel and over 70 userland ...: https://security.apple.com/blog/memory-integrity-enforcement/
- Android bulletins mark exploited issues: "There are indications that the following may be under limited, targeted exploitation.": https://source.android.com/docs/security/bulletin/2025-12-01
- Measurement: FreeType src/truetype/ttinterp.c at commit d7339729 (17 November 2026): opcodes 0x8F and 0x90 labeled /* ADJUST */ and routed to Ins_UNKNOWN; label from ...: https://gitlab.freedesktop.org/freetype/freetype
- Citizen Lab, Triple Threat (18 April 2023): targets with Lockdown Mode briefly received real-time warnings of PWNYOURHOME attempts; "we have not seen PWNYOURHOME ...: https://citizenlab.ca/2023/04/nso-groups-pegasus-spyware-returns-in-2022/
- Project Zero (16 March 2023): eighteen 0-day vulnerabilities in Exynos modems; four allowed Internet-to-baseband remote code execution; attacker needs only to know the ...: https://projectzero.google/2023/03/multiple-internet-to-baseband-remote-rce.html
- Apple threat notifications: alerts to users who may have been "individually targeted by mercenary spyware attacks", by Lock Screen alert, email and Apple Account banner; ...: https://support.apple.com/en-us/102174
- Apple iOS 18.6.2 (20 Aug 2025): CVE-2025-43300, ImageIO out-of-bounds write; "may have been exploited in an extremely sophisticated attack against specific targeted ...: https://support.apple.com/en-us/124925
- WhatsApp advisory (August 2025): CVE-2025-55177, incomplete authorization of linked device synchronization messages; "in combination with an OS-level vulnerability on ...: https://www.whatsapp.com/security/advisories/2025

Chapter 11. The Long Tail: End of Support, Edge Appliances and Security Debt

- CISA BOD 26-02, "Mitigating Risk From End-of-Support Edge Devices", 5 February 2026: inventory listed devices in 3 months, decommission them in 12, all end-of-support ...: https://www.cisa.gov/news-events/directives/bod-26-02-mitigating-risk-end-support-edge-devices
- Microsoft Fixed Lifecycle Policy: "A defined support and servicing Lifecycle timeline at the time of product launch"; "A minimum of five years Mainstream Support"; ...: https://learn.microsoft.com/en-us/lifecycle/policies/fixed
- Windows 10 end of support 14 October 2025; commercial ESU "$61 USD per device for Year One... The price doubles every consecutive year, for a maximum of three years.": https://learn.microsoft.com/en-us/windows/whats-new/extended-security-updates
- Windows 10 IoT Enterprise LTSC 2021: extended end 13 January 2032 (listed as 1/14/2032 6:59:59 AM PT): https://learn.microsoft.com/en-us/lifecycle/products/windows-10-iot-enterprise-ltsc-2021
- Windows 10 IoT Enterprise LTSC 2019: extended end 9 January 2029: https://learn.microsoft.com/en-us/lifecycle/products/windows-10-iot-enterprise-ltsc-2019
- Windows 11 IoT Enterprise LTSC 2024: start 1 October 2024, extended end 10 October 2034 (listed as 10/11/2034 6:59:59 AM PT): https://learn.microsoft.com/en-us/lifecycle/products/windows-11-iot-enterprise-ltsc-2024
- Windows 10 Enterprise LTSC 2021 ends 12 January 2027; paid ESU to 8 January 2030: https://learn.microsoft.com/en-us/lifecycle/products/windows-10-enterprise-ltsc-2021
- Windows 11 Enterprise LTSC 2024 ends 9 October 2029: https://learn.microsoft.com/en-us/lifecycle/products/windows-11-enterprise-ltsc-2024
- Secure Boot 2011 certificates: KEK CA 2011 24 June 2026; UEFI CA 2011 27 June 2026; Windows Production PCA 2011 19 October 2026: https://support.microsoft.com/en-us/topic/windows-secure-boot-certificate-expiration-and-ca-updates-7ff40d33-95dc-4c3c-8725-a9b95457578e
- Apple: "not all known security issues are addressed in previous versions": https://support.apple.com/guide/deployment/about-software-updates-depc4c80847a/web
- iOS 16.7.11 (31 March 2025): CVE-2025-24200 and CVE-2025-43200 (Messages entry added 11 June 2025), plus CVE-2025-24201; "may have been exploited in an extremely ...: https://support.apple.com/en-us/122346
- iOS 15.8.8 and iOS 16.7.16 released 11 May 2026: https://support.apple.com/en-us/100100
- Google, 4 October 2023: "seven years of software support for the Pixel 8 and Pixel 8 Pro": https://blog.google/products-and-platforms/devices/pixel/software-support-pixel-8-pixel-8-pro/
- Pixel updates for 7 years "starting from when the device first became available on the Google Store in the US": https://support.google.com/pixelphone/answer/4457705
- Samsung Galaxy S24 (January 2024): "seven generations of OS upgrades and seven years of security updates": https://news.samsung.com/global/enter-the-new-era-of-mobile-ai-with-samsung-galaxy-s24-series

- Fairphone 5: "eight years of updates and software support, guaranteeing a future proof performance until 2031": https://www.fairphone.com/legal/fairphone-5-tagline-explained
- kernel.org longterm kernels: 6.18 and 6.12 to Dec 2028; 6.6 and 6.1 to Dec 2027; 5.15 and 5.10 to Dec 2026 (6.6 released 2023-10-29); "Each new longterm kernel usually ...: https://www.kernel.org/category/releases.html
- Regulation (EU) 2024/2847 (CRA) Art. 13(8): "the support period shall be at least five years" unless expected use is shorter; applies from 11 December 2027, Article 14 ...: https://eur-lex.europa.eu/eli/reg/2024/2847/oj/eng
- Mandiant M-Trends 2024 (2023 data): exploits 38% of initial vectors; "the first and third most targeted vulnerabilities were related to edge devices": https://services.google.com/fh/files/misc/m-trends-2024.pdf
- Mandiant M-Trends 2025 (2024 data): exploits 33%; "the most frequently exploited vulnerabilities affected security devices" at the network edge: https://services.google.com/fh/files/misc/m-trends-2025-en.pdf
- Verizon DBIR 2025: edge devices and VPNs 22% of vulnerability exploitation targets, almost eight-fold from 3%; about 54% fully remediated, median 32 days: https://www.verizon.com/business/resources/reports/2025-dbir-executive-summary.pdf
- GTIG 2024 zero-day review (29 April 2025): 75 zero-days; 44% enterprise; 20 security and networking: https://cloud.google.com/blog/topics/threat-intelligence/2024-zero-day-trends
- GTIG 2025 zero-day review (5 March 2026): 90 zero-days; 43 (48%) enterprise software and appliances; 21 security and networking: https://cloud.google.com/blog/topics/threat-intelligence/2025-zero-day-review
- Mandiant, "How Low Can You Go? An Analysis of 2023 Time-to-Exploit Trends" (16 October 2024): 138 vulnerabilities, 97 zero-days, 41 n-days, average time to exploit 5 days: https://cloud.google.com/blog/topics/threat-intelligence/time-to-exploit-trends-2023
- Mandiant, "How Low Can You Go? An Analysis of 2023 Time-to-Exploit Trends" (16 October 2024): 138 vulnerabilities, 97 zero-days, 41 n-days, average time to exploit 5 days: https://cloud.google.com/blog/ja/topics/threat-intelligence/time-to-exploit-trends-2023
- KEV dates: CVE-2023-4966 (Citrix NetScaler) 18 October 2023; CVE-2024-21762 (FortiOS SSL VPN) 9 February 2024; CVE-2023-46805 and CVE-2024-21887 (Ivanti) 10 January 2024: https://www.cisa.gov/known-exploited-vulnerabilities-catalog
- CISA ED 24-01 (19 January 2024), Ivanti Connect Secure and Policy Secure; Supplemental Direction V1 (31 January 2024): "disconnect all instances", deadline 2 February ...: https://www.cisa.gov/news-events/directives/ed-24-01-mitigate-ivanti-connect-secure-and-ivanti-policy-secure-vulnerabilities
- CISA ED 25-03 (25 September 2025), Cisco ASA and Firepower: end-of-support hardware: "Permanently disconnect these devices on or before September 30, 2025.": https://www.cisa.gov/news-events/directives/ed-25-03-identify-and-mitigate-potential-compromise-cisco-devices
- Apple UK PSTI statement of compliance, iPad Pro 13-inch (A2926): "Defined Support Period: Minimum 5 Years from the first supply date"; PSTI regulations enforcement date ...: https://three.co.uk/content/dam/threedigital/terms-and-conditions/psti/apple/a2926-ipad-pro-wi-fi-13-inch.pdf
- Regulation (EU) 2023/1670 (ecodesign, smartphones and tablets): applies from 20 June 2025; operating-system updates "from the date of end of placement on the market to ...: https://eur-lex.europa.eu/eli/reg/2023/1670/oj/eng
- Android Security Bulletin, July 2025 (published 7 July 2025): "There are no Android security patches in the July 2025 Android Security Bulletin": https://source.android.com/docs/security/bulletin/2025-07-01
- Android Security Bulletin, September 2025: more than 100 CVEs across the 2025-09-01 and 2025-09-05 patch levels: https://source.android.com/docs/security/bulletin/2025-09-01
- Barracuda ESG notice (6 June 2023, updated 15 June 2023), CVE-2023-2868: "Compromised ESG appliances must be immediately replaced regardless of patch version level.": https://trust.barracuda.com/security/information/esg-vulnerability

Chapter 12. Who Pays for Security: The Market

- Apple, 10 Oct 2025: top Apple Security Bounty award doubled to $2 million for exploit chains with mercenary-spyware goals; bonuses can take a payout "in excess of $5 ...: https://security.apple.com/blog/apple-security-bounty-evolved/
- Google paid more than $17 million (shown as $17.1 million, 747 researchers) through its Vulnerability Reward Programs in 2025, more than 40% above 2024: https://blog.google/security/vrp-2025-year-in-review/
- Microsoft, 5 Aug 2025: $17 million in bounty rewards in the past year to 344 researchers from 59 countries: https://www.microsoft.com/msrc/blog/2025/08/microsoft-bounty-program-year-in-review-17-million-in-rewards
- Crowdfense exploit acquisition program, price list updated April 2024 (advertising): "iOS Zero Click Full Chain (e.g. iMessage): from 5 to 7 M USD"; "Android Zero Click ...: https://www.crowdfense.com/exploit-acquisition-program/
- Crowdfense exploit acquisition program, price list updated April 2024 (advertising): "iOS Zero Click Full Chain (e.g. iMessage): from 5 to 7 M USD"; "Android Zero Click ...: https://securityaffairs.com/161584/hacking/crowdfense-30m-exploit-acquisition-program.html

- Google announced the Wiz acquisition in March 2025 ($32 billion headline) and completed it on 11 March 2026: https://www.nasdaq.com/press-release/google-completes-acquisition-wiz-2026-03-11
- Microsoft Secure Future Initiative announced 2 November 2023 (Brad Smith; Charlie Bell engineering memo): https://blogs.microsoft.com/on-the-issues/2023/11/02/secure-future-initiative-sfi-cybersecurity-cyberattacks/
- OpenSSL advisory for CVE-2014-0160 (Heartbleed), 7 April 2014: https://www.openssl.org/news/secadv/20140407.txt
- Steve Marquess (OpenSSL Software Foundation), April 2014: the foundation typically received about $2,000 a year in donations: http://veridicalsystems.com/blog/of-money-responsibility-and-pride/index.html
- Core Infrastructure Initiative announced by the Linux Foundation on 24 April 2014 after Heartbleed; founding companies pledged funding over three years (secondary report): https://www.itnews.com.au/news/linux-foundation-pours-millions-into-critical-internet-infrastructure-383837
- OpenSSF launched by the Linux Foundation on 3 August 2020, merging the Core Infrastructure Initiative and other efforts: https://www.linuxfoundation.org/press/press-release/technology-and-enterprise-leaders-combine-efforts-to-improve-open-source-security
- Alpha-Omega launched 1 February 2022 under OpenSSF with an initial $5 million from Microsoft and Google (secondary report): https://siliconangle.com/2022/02/01/google-microsoft-back-new-industry-initiative-fix-open-source-vulnerabilities/
- Sovereign Tech Fund (Germany, 2022), now the Sovereign Tech Agency: more than EUR 24.6 million to more than 60 open-source projects (European Commission OSOR case study, ...: https://interoperable-europe.ec.europa.eu/collection/open-source-observatory-osor/document/funding-open-source-case-study-sovereign-tech-fund
- Merck & Co. v. ACE American Insurance Co., N.J. Super. Ct. App. Div., No. A-1879-21, decided 1 May 2023: the "hostile/warlike action" exclusion did not bar coverage for ...: https://law.justia.com/cases/new-jersey/appellate-division-published/2023/a-1879-21.html
- Merck and its remaining insurers settled in January 2024, before the New Jersey Supreme Court ruled (secondary report): https://www.insurancejournal.com/news/national/2024/01/05/754582.htm
- Mondelez v. Zurich American (Illinois state court): Zurich relied on a war exclusion for a claim over $100 million; settled in late 2022 during trial, terms undisclosed ...: https://therecord.media/mondelez-and-zurich-reach-settlement-in-notpetya-cyberattack-insurance-suit
- Lloyd's Market Bulletin Y5381 (August 2022): standalone cyber-attack policies from 31 March 2023 must, at a minimum, exclude losses from war and from state-backed ...: https://assets.lloyds.com/media/e9a4058c-4c25-48d6-99d3-0c466987e099/Y5381%20Market%20Bulletin%20-%20Cyber-attack%20exclusions.pdf
- LMA model clauses (LMA5564 series) rely primarily on attribution by the government of the state where the affected computer system is located (law-firm summary): https://www.dacbeachcroft.com/en/What-we-think/war-exclusions-in-cyber-policies-an-overview
- UK and US attributed NotPetya on 15 February 2018 (see Chapter 13 sources): https://www.gov.uk/government/news/foreign-office-minister-condemns-russia-for-notpetya-attacks
- US National Cybersecurity Strategy, 2 March 2023, Strategic Objective 3.3: "Shift liability for insecure software products and services"; legislation should prevent ...: https://www.whitehouse.gov/wp-content/uploads/2023/03/National-Cybersecurity-Strategy-2023.pdf
- Executive Order 14306, 6 June 2025 (90 FR 24723, 11 June 2025), amends EO 13694 and EO 14144; strikes FAR Council secure-software requirements; earlier orders not ...: https://www.govinfo.gov/content/pkg/FR-2025-06-11/html/2025-10804.htm
- CRS Insight IN12570 on changes to national cyber policy (June 2025): https://www.everycrsreport.com/reports/IN12570.html
- Regulation (EU) 2024/2847 (Cyber Resilience Act): in force 10 December 2024; Article 14 applies from 11 September 2026; Chapter IV from 11 June 2026; general application ...: https://eur-lex.europa.eu/eli/reg/2024/2847/oj/eng
- CRA Article 14 early warning within 24 hours of becoming aware of an actively exploited vulnerability or severe incident (Ireland NCSC guidance): https://ncsc.gov.ie/cra/
- Directive (EU) 2024/2853 (Product Liability Directive), adopted 23 October 2024: "product" includes software (Article 4(1)); does not apply to free and open-source ...: https://eur-lex.europa.eu/eli/dir/2024/2853/oj/eng
- Log4Shell (CVE-2021-44228), December 2021: told in Chapter 8: https://logging.apache.org/log4j/2.x/security.html

Chapter 13. The Record: Spyware, Sabotage and the Outage

- Symantec W32.Stuxnet Dossier, version 1.4 (February 2011), Falliere, O Murchu, Chien: VirusBlokAda report 17 June 2010; earliest sample June 2009, spreading through ...: https://docs.broadcom.com/doc/security-response-w32-stuxnet-dossier-11-en
- Citizen Lab, "The Million Dollar Dissident" (Trident), published 24 August 2016; Apple iOS 9.3.5, 25 August 2016: https://citizenlab.ca/2016/08/million-dollar-dissident-iphone-zero-day-nso-group-uae/

- Citizen Lab, "Hide and Seek: Tracking NSO Group's Pegasus Spyware to Operations in 45 Countries", 18 September 2018; list may include non-customer countries: https://citizenlab.ca/2018/09/hide-and-seek-tracking-nso-groups-pegasus-spyware-to-operations-in-45-countries/
- Citizen Lab, FORCEDENTRY, 13 September 2021; Apple iOS 14.8 the same day: https://citizenlab.ca/2021/09/forcedentry-nso-group-imessage-zero-click-exploit-captured-in-the-wild/
- Citizen Lab, BLASTPASS, 7 September 2023: zero-click chain delivered Pegasus to an employee of a Washington DC civil society organization; CVE-2023-41064 and …: https://citizenlab.ca/2023/09/blastpass-nso-group-iphone-zero-click-zero-day-exploit-captured-in-the-wild/
- CERT-EU advisory 2023-061 on the BLASTPASS CVEs: https://cert.europa.eu/publications/security-advisories/2023-061/
- BIS adds NSO Group and Candiru (with Positive Technologies and Computer Security Initiative Consultancy) to the Entity List; rule effective and published 4 November …: https://regulations.justia.com/regulations/fedreg/2021/11/04/2021-24123.html
- BIS press release on the NSO Group Entity List action: https://www.bis.gov/press-release/commerce-adds-nso-group-other-foreign-companies-entity-list-malicious-cyber-activities
- US Treasury sanctions on members of the Intellexa Consortium (Predator spyware), 5 March 2024: https://home.treasury.gov/news/press-releases/jy2155
- WhatsApp Inc. v. NSO Group Technologies, N.D. Cal. No. 4:19-cv-07123-PJH, filed 29 October 2019, Judge Phyllis J. Hamilton; about 1,400 devices alleged: https://www.pacermonitor.com/public/case/33085233/WhatsApp_Inc_et_al_v_NSO_Group_Technologies_Limited_et_al
- Order of 20 December 2024 granting partial summary judgment on liability (CFAA, California CDAFA, breach of contract): https://storage.courtlistener.com/recap/gov.uscourts.cand.350613/gov.uscourts.cand.350613.465.0.pdf
- Jury verdict 6 May 2025: $444,719 compensatory, $167,254,000 punitive (counsel's announcement and press): https://www.davispolk.com/experience/trial-victory-meta-and-whatsapp-spyware-case
- Order of 17 October 2025: permanent injunction barring NSO from breaking into or attempting to break into WhatsApp's systems; punitive damages capped at a 9 to 1 ratio, …: https://cyberscoop.com/whatsapp-wins-injunction-against-nso-group-spyware-damages-reduced/
- Order of 17 October 2025: permanent injunction barring NSO from breaking into or attempting to break into WhatsApp's systems; punitive damages capped at a 9 to 1 ratio, …: https://www.securityweek.com/nso-ordered-to-stop-hacking-whatsapp-but-damages-cut-to-4-million/
- Ninth Circuit appeal No. 25-7380: stay of injunction denied 28 January 2026; NSO opening brief 11 February 2026; WhatsApp answering brief 13 March 2026; pending at …: https://dockets.justia.com/docket/circuit-courts/ca9/25-7380
- Apple Inc. v. NSO Group, N.D. Cal. No. 3:21-cv-09078-JD, filed 23 November 2021; Apple moved to dismiss in September 2024 citing risk to security information; dismissed …: https://www.courthousenews.com/wp-content/uploads/2025/07/apple-vs-nso-dismissal.pdf
- US-CERT alert TA17-181A (1 July 2017, revised 15 February 2018): attackers "used a backdoor to compromise M.E. Doc's development environment as far back as April 14, …: https://www.cisa.gov/news-events/alerts/2017/07/01/petya-ransomware
- UK Foreign Office minister Lord Ahmad, 15 February 2018: UK government judges the Russian government responsible; NCSC assesses the Russian military "almost certainly …: https://www.gov.uk/government/news/foreign-office-minister-condemns-russia-for-notpetya-attacks
- White House press secretary statement, 15 February 2018: "the Russian military launched the most destructive and costly cyber-attack in history": https://www.presidency.ucsb.edu/documents/statement-the-press-secretary-the-notpetya-cyber-attack
- DOJ, 19 October 2020: six officers of GRU Unit 74455 charged in W.D. Pa. (indictment returned 15 October 2020), including for NotPetya; "nearly one billion USD in losses …: https://www.justice.gov/opa/pr/six-russian-gru-officers-charged-connection-worldwide-deployment-destructive-malware-and
- Indictment text: https://www.justice.gov/opa/press-release/file/1328521/dl
- Andy Greenberg, Wired, 22 August 2018: "more than $10 billion in total damages, according to a White House assessment confirmed to WIRED by former Homeland Security …: https://www.wired.com/story/notpetya-cyberattack-ukraine-russia-code-crashed-the-world/
- CrowdStrike preliminary post incident review: release 04:09 UTC, reverted 05:27 UTC on 19 July 2024; commitments to staggered deployment with canary and more customer …: https://www.crowdstrike.com/en-us/blog/falcon-content-update-preliminary-post-incident-report/
- CrowdStrike root cause analysis, 6 August 2024: https://www.crowdstrike.com/wp-content/uploads/2024/08/Channel-File-291-Incident-Root-Cause-Analysis-08.06.2024.pdf

- Microsoft (David Weston), 20 July 2024: "We currently estimate that CrowdStrike's update affected 8.5 million Windows devices, or less than one percent of all Windows ...: https://blogs.microsoft.com/blog/2024/07/20/helping-our-customers-through-the-crowdstrike-outage/
- Adam Meyers testimony, House Homeland Security subcommittee, 24 September 2024: "we let our customers down"; not a cyberattack: https://homeland.house.gov/wp-content/uploads/2024/09/2024-09-24-HRG-CIP-Testimony-Meyers.pdf
- Delta SEC filing, August 2024: about $380 million revenue impact and about $170 million costs; about 7,000 flight cancellations (secondary reports of the filing): https://fortune.com/2024/08/09/delta-380-million-lost-revenue-crowdstrike-outage-cancel-thousands-flights/
- CrowdStrike counsel's August 2024 letter: liability contractually capped in the single-digit millions (Reuters via Insurance Journal): https://www.insurancejournal.com/news/national/2024/08/05/787017.htm
- Delta Air Lines v. CrowdStrike, Superior Court of Fulton County, complaint 25 October 2024; motion to dismiss granted in part and denied in part 16 May 2025; later ...: https://www.sec.gov/Archives/edgar/data/0001535527/000153552726000025/crwd-20260430.htm
- Claims proceeding after 16 May 2025 include gross negligence and computer trespass (secondary report): https://www.bankinfosecurity.com/judge-lets-deltas-cyber-failure-suit-vs-crowdstrike-proceed-a-28443
- Channel File 291 reached Windows hosts running sensor 7.11 and later that were online between 04:09 and 05:27 UTC on 19 July 2024 (CrowdStrike preliminary post incident ...: https://www.crowdstrike.com/falcon-content-update-remediation-and-guidance-hub/
- Executive Order 14093, 27 March 2023: US agencies shall not make operational use of commercial spyware that poses significant counterintelligence or security risks (88 ...: https://www.govinfo.gov/link/cpd/executiveorder/14093
- Google, 13 May 2025: Advanced Protection extended to a device-level setting in Android 16: https://security.googleblog.com/2025/05/advanced-protection-mobile-devices.html
- SolarWinds (2020) and xz (2024) build-system cases: told in Chapter 8: https://www.cisa.gov/news-events/directives/ed-21-01-mitigate-solarwinds-orion-code-compromise
- Microsoft Security Bulletin MS10-046 (Windows Shell shortcut flaw used by Stuxnet), released out of band 2 August 2010: https://learn.microsoft.com/en-us/security-updates/securitybulletins/2010/ms10-046
- Microsoft Security Bulletin MS10-061 (Print Spooler flaw used by Stuxnet to spread), released 14 September 2010: https://learn.microsoft.com/en-us/security-updates/securitybulletinsummaries/2010/ms10-sep
- Microsoft Security Bulletin MS17-010 (SMBv1 flaws used by EternalBlue and EternalRomance), released 14 March 2017: https://learn.microsoft.com/en-us/security-updates/securitybulletins/2017/ms17-010

Chapter 14. The Cost of Trust: Control, Concentration and a Day Down

- Interagency computer-security incident notification rule (OCC, Federal Reserve, FDIC), 86 FR 66424 (November 2021): a bank notifies its primary federal regulator as soon ...: https://occ.gov/news-issuances/bulletins/2021/bulletin-2021-55.html
- Rule effective 1 April 2022, full compliance 1 May 2022 (law-firm summary): https://cdp.cooley.com/36-hour-breach-notification-rule-banking-organizations/
- SEC, 26 July 2023: Form 8-K Item 1.05 generally due four business days after a registrant determines a cybersecurity incident is material: https://www.sec.gov/news/press-release/2023-139
- Item 1.05 requirements effective 18 December 2023; smaller reporting companies 15 June 2024; materiality determination without unreasonable delay (law-firm summary): https://www.hunton.com/privacy-and-information-security-law/sec-adopts-final-public-company-cybersecurity-disclosure-rules
- Regulation (EU) 2022/2554 (DORA) applies from 17 January 2025: https://eur-lex.europa.eu/eli/reg/2022/2554/oj/eng
- Commission Delegated Regulation (EU) 2025/301, Article 5(1): initial notification within four hours of classification as major and no later than 24 hours from awareness; ...: https://www.springlex.eu/en/packages/dora/rts-ir-regulation/article-5/
- GDPR Article 33(1): notify the supervisory authority without undue delay and, where feasible, not later than 72 hours after becoming aware of a personal data breach: https://eur-lex.europa.eu/eli/reg/2016/679/oj/eng
- CRA Article 14 timelines: early warning within 24 hours; notification within 72 hours; final report within 14 days after a corrective measure is available ...: https://ncsc.gov.ie/cra/
- CRA Article 71: Article 14 applies from 11 September 2026: https://eur-lex.europa.eu/eli/reg/2024/2847/oj/eng
- Delta SEC filing, August 2024: about $380 million revenue impact, about $170 million costs (secondary reports of the filing; about $550 million in total): https://www.nbcnews.com/business/business-news/delta-says-chaos-crowdstrike-outage-cost-550-million-rcna165864

- Parametrix, 24 July 2024: $5.4 billion direct financial loss to US Fortune 500 companies excluding Microsoft; insured loss $540 million to $1.08 billion (insurer's ...: https://parametrixinsurance.com/in-the-news/crowdstrike-to-cost-fortune-500-5-4-billion-insured-loss-range-of-540-million-to-1-08-billion
- CrowdStrike counsel, August 2024: liability to Delta contractually capped at an amount in the single-digit millions (Reuters via Insurance Journal): https://www.insurancejournal.com/news/national/2024/08/05/787017.htm
- CrowdStrike preliminary post incident review: staggered deployment with canary and more customer control over content delivery: https://www.crowdstrike.com/en-us/blog/falcon-content-update-preliminary-post-incident-report/
- Quick machine recovery generally available on Windows 11, version 24H2 (KB5062660), enabled by default on Home and disabled by default on Pro, Education and Enterprise ...: https://techcommunity.microsoft.com/blog/windows-itpro-blog/resilience-in-action-for-windows-devices/4434571/
- Quick machine recovery generally available on Windows 11, version 24H2 (KB5062660), enabled by default on Home and disabled by default on Pro, Education and Enterprise ...: https://learn.microsoft.com/en-us/windows/configuration/quick-machine-recovery
- Maersk chairman Jim Hagemann Snabe, World Economic Forum, January 2018: about 4,000 servers, 45,000 PCs and 2,500 applications reinstalled in ten days after NotPetya ...: https://www.theregister.com/2018/01/25/after_notpetya_maersk_replaced_everything
- European supervisory authorities designate critical ICT third-party providers under DORA, 18 November 2025: https://www.esma.europa.eu/press-news/esma-news/european-supervisory-authorities-designate-critical-ict-third-party-providers
- UK critical third parties regime: PRA PS16/24 and FCA PS24/16 (12 November 2024), rules in force 1 January 2025, applying to providers designated by HM Treasury: https://www.fca.org.uk/publications/policy-statements/ps24-16-operational-resilience-critical-third-parties-uk-financial-sector
- Bank service provider notification applies to a bank service provider performing covered services, when it determines that an incident has materially disrupted or ...: https://www.federalregister.gov/documents/2021/11/23/2021-25510/computer-security-incident-notification-requirements-for-banking-organizations-and-their-bank

Chapter 15. Own, Rent, Pin, Isolate: The Decision and the Contract

- CrowdStrike Channel File 291 root cause analysis dated 6 August 2024; mitigation "Customers can choose where and when Rapid Response Content updates are deployed"; ...: https://www.crowdstrike.com/wp-content/uploads/2024/08/Channel-File-291-Incident-Root-Cause-Analysis-08.06.2024.pdf
- CRA Art. 13(8): manufacturer handles vulnerabilities of the product "including its components" for the support period; support period at least five years unless expected ...: https://eur-lex.europa.eu/legal-content/EN/TXT/HTML/?uri=CELEX:32024R2847
- CRA Art. 71: applies from 11 December 2027; Art. 14 from 11 September 2026; published in OJ 20 November 2024, in force on the twentieth day (10 December 2024): https://eur-lex.europa.eu/eli/reg/2024/2847/oj/eng
- NIST SP 800-161 Rev. 1, "Cybersecurity Supply Chain Risk Management Practices for Systems and Organizations", May 2022, Update 1 on 1 November 2024; C-SCRM definition ...: https://csrc.nist.gov/pubs/sp/800/161/r1/upd1/final
- CISA Secure by Design Pledge (May 2024): goals include security patches and evidence of intrusions (logs for customers): https://www.cisa.gov/securebydesign/pledge
- CISA Secure by Design Pledge document dated May 2024: https://www.cisa.gov/sites/default/files/2024-05/CISA%20Secure%20by%20Design%20Pledge_508c.pdf
- Windows: quality updates deferred up to 30 days, paused up to 35 days; feature updates deferred up to 365 days: https://learn.microsoft.com/en-us/windows/deployment/update/waas-configure-wufb
- Apple documents update visibility for devices on a 90-day deferral: https://support.apple.com/guide/deployment/manage-software-updates-depc4c80847a/web
- Android device owner: one 30-day postponement; freeze periods up to 90 days with a 60-day buffer; makers may exempt important security updates: https://developer.android.com/work/dpc/system-updates
- Windows 11 IoT Enterprise LTSC 2024: start 1 October 2024, extended support to October 2034: https://learn.microsoft.com/en-us/lifecycle/products/windows-11-iot-enterprise-ltsc-2024
- Windows 10 IoT Enterprise LTSC 2021: start 16 November 2021, extended support to January 2032: https://learn.microsoft.com/en-us/lifecycle/products/windows-10-iot-enterprise-ltsc-2021
- Windows 10 2015 LTSB: start 29 July 2015, extended support ended October 2025: https://learn.microsoft.com/en-us/lifecycle/products/windows-10-2015-ltsb
- Regulation (EU) 2022/2554 (DORA), Article 30: key contractual provisions for ICT services, including audit rights and exit strategies for critical or important functions: https://eur-lex.europa.eu/eli/reg/2022/2554/oj/eng
- Microsoft, Windows Resiliency Initiative (November 2024): work with endpoint security partners through the Microsoft Virus Initiative to focus on adopting Safe ...: https://www.microsoft.com/en-us/windows/business/windows-resiliency-initiative

- CISA and FBI, Secure by Demand Guide: How Software Customers Can Drive a Secure Technology Ecosystem (6 August 2024): questions buyers can ask software manufacturers: https://www.cisa.gov/news-events/alerts/2024/08/06/cisa-releases-secure-demand-guidance

Chapter 16. AI on Both Sides of the Stack

- Anthropic detected the operation in mid-September 2025; assesses "with high confidence" a Chinese state-sponsored group it designated GTG-1002; roughly 30 targets; "a ...: https://assets.anthropic.com/m/ec212e6566a0d47/original/Disrupting-the-first-reported-AI-orchestrated-cyber-espionage-campaign.pdf
- Anthropic summary post dated 13 November 2025: https://www.anthropic.com/news/disrupting-AI-espionage
- Model Context Protocol introduced 25 November 2024: https://www.anthropic.com/news/model-context-protocol
- Mandiant TTE: 63 days (2018 to 2019), 44 (2020 to start of 2021), 32 (2021 to 2022), five days (2023); 138 vulnerabilities, 97 zero-days, 41 n-days; 15 outliers ...: https://cloud.google.com/blog/topics/threat-intelligence/time-to-exploit-trends-2023
- M-Trends 2026 (23 March 2026): mean time to exploit "dropped to an estimated -7 days": https://cloud.google.com/blog/topics/threat-intelligence/m-trends-2026
- Big Sleep (Project Zero and Google DeepMind), 1 November 2024: exploitable stack buffer underflow in SQLite, fixed the same day, before an official release; "first ...: https://projectzero.google/2024/10/from-naptime-to-big-sleep.html
- Google, 15 July 2025: Big Sleep found CVE-2025-6965 based on Google Threat Intelligence; "known only to threat actors"; "first time" an AI agent used to foil ...: https://blog.google/technology/safety-security/cybersecurity-updates-summer-2025/
- DARPA AIxCC results, 8 August 2025: 54 of 63 synthetic vulnerabilities found ("54 unique synthetic vulnerabilities in the Final Competition's 63 challenges"; DARPA ...: https://www.darpa.mil/news/2025/aixcc-results
- Google DeepMind CodeMender, 6 October 2025: 72 security fixes upstreamed, human review before submission: https://deepmind.google/blog/introducing-codemender-an-ai-agent-for-code-security/
- Windows experimental agentic features: off by default; only an administrator can enable; separate agent accounts; agent workspace in a separate session; access to six ...: https://support.microsoft.com/en-us/windows/ai/ai-features/experimental-agentic-features
- Apple App Intents: "Make content and actions discoverable by Apple Intelligence and support system experiences like Siri, Spotlight, Shortcuts"; iOS 16.0, macOS 13.0: https://developer.apple.com/documentation/appintents
- Android AppFunctions: Android 16 and later; callers need EXECUTE_APP_FUNCTIONS; experimental with limited access; guidance to expose only safe, non-destructive tasks: https://developer.android.com/ai/appfunctions
- OWASP Top 10 for LLM Applications 2025: LLM01 Prompt Injection, LLM06 Excessive Agency: https://genai.owasp.org/llm-top-10/
- OWASP Top 10 for Agentic Applications for 2026, published 9 December 2025: https://genai.owasp.org/resource/owasp-top-10-for-agentic-applications-for-2026/
- Apple Private Cloud Compute, 10 June 2024: devices send data only to nodes running verifiable images whose measurements match the public transparency log: https://security.apple.com/blog/private-cloud-compute/
- DARPA Cyber Grand Challenge final, 4 August 2016; Mayhem (ForAllSecure) won: https://www.darpa.mil/news/2016/cyber-grand-challenge-winners
- Saltzer and Schroeder, "The Protection of Information in Computer Systems" (1975 version): least privilege definition quoted: https://www.cs.virginia.edu/~evans/cs551/saltzer/
- Outside researchers questioned the GTG-1002 report, citing the absence of indicators of compromise and the autonomy figure (press, November 2025): https://www.bleepingcomputer.com/news/security/anthropic-claims-of-claude-ai-automated-cyberattacks-met-with-doubt/
- Outside researchers questioned the GTG-1002 report, citing the absence of indicators of compromise and the autonomy figure (press, November 2025): https://arstechnica.com/security/2025/11/researchers-question-anthropic-claim-that-ai-assisted-attack-was-90-autonomous/
- XBOW, June 2025: autonomous system reached first place on HackerOne's US leaderboard after about 90 days of submissions (company account): https://xbow.com/blog/top-1-how-xbow-did-it/
- curl, January 2026: Daniel Stenberg announced the end of curl's bug bounty at the end of January 2026, to remove the incentive for made-up, often AI-generated, reports: https://www.theregister.com/2026/01/21/curl_ends_bug_bounty/
- Arup confirmed in May 2024 that it was the company that lost HK$200 million (about US$25 million) in a Hong Kong fraud using a deepfaked video call: https://scmp.com/news/hong-kong/law-and-crime/article/3263151/uk-multinational-arup-confirmed-victim-hk200-million-deepfake-scam-used-digital-version-cfo-dupe

- Google Threat Intelligence Group, "AI Threat Tracker: Advances in Threat Actor Usage of AI Tools" (5 November 2025): first malware families seen using a language model …: https://cloud.google.com/blog/topics/threat-intelligence/threat-actor-usage-of-ai-tools

Chapter 17. The Next Trust Migration: Memory Safety, Attestation and Post-Quantum Keys

- NIST IR 8547 final, October 2026 (final 27 October 2026): quantum-vulnerable ECDSA, EdDSA, RSA signatures and DH/ECDH/MQV key establishment at "≥ 112 bits of classical …: https://nvlpubs.nist.gov/nistpubs/ir/2026/NIST.IR.8547.pdf
- NIST IR 8547 page and initial public draft dated 12 November 2024: https://csrc.nist.gov/pubs/ir/8547/final
- NIST news (27 October 2026): EO 14412 requires high-value assets and high-impact systems to "migrate by 2030 for key establishment and 2031 for digital signatures"; 2035 …: https://www.nist.gov/news-events/news/2026/10/nist-report-provides-guidelines-transition-post-quantum-cryptography
- FIPS 203 (ML-KEM, from CRYSTALS-Kyber), FIPS 204 (ML-DSA, from CRYSTALS-Dilithium), FIPS 205 (SLH-DSA, from SPHINCS+) approved 13 August 2024; FALCON-derived standard in …: https://csrc.nist.gov/news/2024/postquantum-cryptography-fips-approved
- NIST selects HQC, 11 March 2025; backup to ML-KEM on different math; final standard expected 2027: https://www.nist.gov/news-events/news/2025/03/nist-selects-hqc-fifth-algorithm-post-quantum-encryption
- NSA CNSA 2.0 advisory (September 2022): signing prefer by 2025, exclusive by 2030; browsers, servers, cloud prefer 2025, exclusive 2033; networking prefer 2026, …: https://media.defense.gov/2022/Sep/07/2003071834/-1/-1/0/CSA_CNSA_2.0_ALGORITHMS_.PDF
- OpenSSH 9.0 released 2022-04-08 with sntrup761x25519 hybrid key exchange by default: https://www.openssh.org/txt/release-9.0
- OpenSSH 10.0 released 2025-04-09; mlkem768x25519-sha256 used by default: https://www.openssh.org/txt/release-10.0
- Chrome 131 switches hybrid key exchange from Kyber to ML-KEM (post 13 September 2024): https://security.googleblog.com/2024/09/a-new-path-for-kyber-on-web.html
- Chrome 131 stable on 12 November 2024: https://developer.chrome.com/release-notes/131
- iOS 26, iPadOS 26, macOS Tahoe 26, visionOS 26 advertise hybrid quantum-secure key exchange in TLS 1.3 automatically: https://support.apple.com/en-us/122756
- iOS 26 security content released 15 September 2025: https://support.apple.com/en-us/125108
- Microsoft: PQC APIs (ML-KEM, ML-DSA) generally available in Windows 11 24H2 and 25H2, Windows Server 2025, .NET 10; 18 November 2025: https://techcommunity.microsoft.com/blog/microsoft-security-blog/-/4469093
- MSRC, 16 July 2019: "~70% of the vulnerabilities Microsoft assigns a CVE each year continue to be memory safety issues": https://www.microsoft.com/en-us/msrc/blog/2019/07/a-proactive-approach-to-more-secure-code
- Google, 13 November 2025: memory-safety vulnerabilities below 20% of Android vulnerabilities in 2025; about 0.2 per MLOC in Rust vs "closer to 1,000" per MLOC in C/C++; …: https://blog.google/security/rust-in-android-move-fast-fix-things/
- NSA and CISA, "Memory Safe Languages: Reducing Vulnerabilities in Modern Software Development", 24 June 2025: https://www.cisa.gov/resources-tools/resources/memory-safe-languages-reducing-vulnerabilities-modern-software-development
- Android MTE: platform support from Android 13; Pixel 8 and later listed (Pixel 8 shipped with Android 14); off by default for apps; developer option on some devices: https://developer.android.com/ndk/guides/arm-mte
- Apple Memory Integrity Enforcement, 9 September 2025: all iPhone 17 models and iPhone Air; A19 and A19 Pro; EMTE spec with Arm released 2022; synchronous; kernel and …: https://security.apple.com/blog/memory-integrity-enforcement/
- CHERI: SRI International and University of Cambridge; DARPA support since 2010: https://www.cl.cam.ac.uk/research/security/ctsrd/cheri/
- RFC 9334, RATS Architecture, January 2023: roles, Evidence and Attestation Result definitions, freshness methods, "always a race condition possible": https://www.rfc-editor.org/rfc/rfc9334.html
- Measurement: Chromium at commit 16c0007869adaf2f41a88a29bebd0d522eea8fc6 (2026-11-17), git ls-files excluding third_party and vendor paths: 327 .rs files, 110,380 …: https://chromium.googlesource.com/chromium/src
- Arm, January 2022: first Morello boards shipping to researchers through the UK Digital Security by Design programme: https://newsroom.arm.com/news/morello-research-program-hits-major-milestone-with-hardware-now-available-for-testing
- CISA and FBI, Product Security Bad Practices, version 2 (January 2025): "develop new product lines in memory safe languages"; for existing products, "publish a memory …: https://www.cisa.gov/resources-tools/resources/product-security-bad-practices
- Google, November 2025: Rust vulnerability density of about 0.2 per million lines estimated from about five million lines and one potential memory-safety flaw found …: https://security.googleblog.com/2025/11/rust-in-android-move-fast-fix-things.html

- NIS Cooperation Group, A Coordinated Implementation Roadmap for the Transition to Post-Quantum Cryptography, published 23 June 2025: start by end of 2026, high-risk ...: https://digital-strategy.ec.europa.eu/en/library/coordinated-implementation-roadmap-transition-post-quantum-cryptography
- UK NCSC, Timelines for migration to post-quantum cryptography, 20 March 2025: by 2028 discovery and initial plan; by 2031 highest-priority migration; by 2035 complete: https://www.ncsc.gov.uk/guidance/pqc-migration-timelines

Epilogue: The Last Authority Is You

- CRA Art. 14(2): 24-hour early warning of actively exploited vulnerabilities, applying from 11 September 2026 (Larkfield's duty): https://eur-lex.europa.eu/legal-content/EN/TXT/HTML/?uri=CELEX:32024R2847
- The author's runtime work under the jc10 name: https://jc10.ai/

Appendix C. The Five-OS Trust Reference Table

- Windows Hello for Business: if "Use a hardware security device" is not enabled, the TPM is still preferred but devices can provision with software keys when the TPM is ...: https://learn.microsoft.com/en-us/windows/security/identity-protection/hello-for-business/policy-settings
- fapolicyd controls the execution of applications based on a user-defined policy (RHEL application allowlisting): https://docs.redhat.com/en/documentation/red_hat_enterprise_linux/9/html/security_hardening/assembly_blocking-and-allowing-applications-using-fapolicyd_security-hardening
- GKI: devices shipping with kernel 5.10 or higher on Android 12 must ship the GKI kernel; GKI releases contain a tested boot.img with a Google-inserted certificate; SoC ...: https://source.android.com/docs/core/architecture/kernel/gki-releases
- GKI overview: https://source.android.com/docs/core/architecture/kernel/generic-kernel-image
- Memory Integrity Enforcement in all models of iPhone 17 and iPhone Air (9 September 2025): https://security.apple.com/blog/memory-integrity-enforcement/
- Apple notarization for iOS apps distributed outside the App Store in the EU: https://developer.apple.com/support/dma-and-apps-in-the-eu/
- Clevis: automated decryption framework with Tang (network) and TPM2 pins, including LUKS binding: https://github.com/latchset/clevis
- systemd-cryptenroll enrolls TPM2, FIDO2 and recovery keys in LUKS2 volumes: https://www.freedesktop.org/software/systemd/man/latest/systemd-cryptenroll.html
- Red Hat Satellite (fleet management for RHEL): https://www.redhat.com/en/technologies/management/satellite
- Red Hat Insights: https://www.redhat.com/en/technologies/management/insights
- Canonical Landscape (Ubuntu systems management): https://ubuntu.com/landscape

Appendix D. Measuring the Patch and Recovery Clocks

- Windows Update for Business reports store data in a Log Analytics workspace, tables UCClient, UCClientUpdateStatus, UCUpdateAlert and others: https://learn.microsoft.com/en-us/windows/deployment/update/wufb-reports-overview
- UCClient fields: OSBuild, OSRevisionNumber, OSSecurityUpdateStatus, LastWUScanTime: https://learn.microsoft.com/en-us/windows/deployment/update/wufb-reports-schema-ucclient
- UCClientUpdateStatus fields: TargetKBNumber, UpdateReleaseTime, UpdateInstalledTime, RestartRequiredTime, ClientSubstate: https://learn.microsoft.com/en-us/windows/deployment/update/wufb-reports-schema-ucclientupdatestatus
- Intune Windows Update Distribution Report: release date, devices on this update or later, last check-in; CSV export; cached for three days: https://learn.microsoft.com/en-us/intune/device-updates/windows/monitor-quality-updates
- Windows 11 release information: availability date, build and KB per update: https://learn.microsoft.com/en-us/windows/release-health/windows11-release-information
- WSUS Update Status Summary and Computer Status Summary reports (WSUS 3.0 era documentation): https://learn.microsoft.com/en-us/previous-versions/windows/it-pro/windows-server-2008-R2-and-2008/cc708481(v=ws.10
- SUSDB public views vComputerTarget, vUpdateInstallationInfoBasic (state values installed, pending reboot, failed, not installed): https://learn.microsoft.com/en-us/archive/blogs/wsus/baseline-compliance-report-using-public-wsus-views
- Get-HotFix returns InstalledOn; lists only component-based servicing updates: https://learn.microsoft.com/en-us/powershell/module/microsoft.powershell.management/get-hotfix
- UBR registry value with CurrentBuildNumber gives the build revision; each package raises it: https://learn.microsoft.com/en-us/windows-hardware/manufacture/desktop/oem-deployment-of-windows-desktop-editions
- Apple MDM DeviceInformation query keys OSVersion, BuildVersion, SupplementalBuildVersion: https://developer.apple.com/documentation/devicemanagement/deviceinformationcommand/command-data.dictionary/queries-data.dictionary

- Declarative device management status item device.operating-system.supplemental.build-version and related software update status items: https://developer.apple.com/documentation/devicemanagement/statusdeviceoperatingsystemsupplementalbuildversion
- Apple security releases page lists releases with dates; issues not disclosed until fixes are available: https://support.apple.com/en-us/100100
- Jamf Pro API v4 computers-inventory: operatingSystem.build, supplementalBuildVersion, general.reportDate, general.lastContact; v1 deprecated 30 June 2025: https://developer.jamf.com/jamf-pro/reference/get_v4-computers-inventory
- Jamf Pro API v2 mobile-devices/detail filters on osBuild and osSupplementalBuildVersion: https://developer.jamf.com/jamf-pro/reference/get_v2-mobile-devices-detail
- Android Management API device resource: softwareInfo.securityPatchLevel, androidBuildNumber, systemUpdateInfo.updateReceivedTime, lastStatusReportTime; softwareInfo ...: https://developers.google.com/android/management/reference/rest/v1/enterprises.devices
- Android security bulletin overview: monthly bulletins with publication dates and patch levels; device and chipset makers publish their own bulletins: https://source.android.com/docs/security/bulletin/asb-overview
- rpm -qa --last orders packages by install time, newest first: https://rpm.org/docs/latest/man/rpm.8
- rpm query format INSTALLTIME:date: https://rpm.org/docs/latest/man/rpm-queryformat.7
- dpkg.log line format: date, time, action, package, version: https://manpages.ubuntu.com/manpages/noble/en/man1/dpkg.1.html
- /var/log/apt/history.log records package state changes: https://www.debian.org/releases/trixie/release-notes/upgrading.html
- needs-restarting -r exits 1 when a reboot is required, 0 otherwise: https://dnf-plugins-core.readthedocs.io/en/latest/needs_restarting.html
- Ubuntu unattended upgrades and /var/run/reboot-required: https://ubuntu.com/server/docs/how-to/software/automatic-updates/
- Ubuntu Security Notices: https://ubuntu.com/security/notices
- Red Hat security advisories (RHSA), with CSAF feed: https://access.redhat.com/security/updates/advisory
- Microsoft Entra audit log records "Read BitLocker key" (category KeyManagement): https://learn.microsoft.com/en-us/entra/architecture/security-operations-devices
- Entra audit activity reference: https://learn.microsoft.com/en-us/entra/identity/monitoring-health/reference-audit-activities
- WSUS deprecated (announced 20 September 2024): no new capabilities, current functionality preserved, updates still published through WSUS: https://techcommunity.microsoft.com/blog/windows-itpro-blog/windows-server-update-services-wsus-deprecation/4250436
- Win32_BIOS class property SMBIOSBIOSVersion gives the BIOS version reported by SMBIOS: https://learn.microsoft.com/en-us/windows/win32/cimwin32prov/win32-bios
- Secure Boot certificate update guidance: registry value UEFICA2023Status under HKLM\SYSTEM\CurrentControlSet\Control\SecureBoot\Servicing; event 1801 (updated ...: https://support.microsoft.com/en-us/topic/secure-boot-certificate-updates-guidance-for-it-professionals-and-organizations-e2b43f9f-b424-42df-bc6a-8476db65ab2f
- fwupdmgr get-devices lists devices fwupd can update, with firmware versions: https://fwupd.github.io/libfwupdplugin/fwupdmgr.html
- kpatch list shows installed and loaded patch modules: https://github.com/dynup/kpatch/blob/master/man/kpatch.1
- canonical-livepatch status shows the Livepatch client state and applied patches: https://ubuntu.com/security/livepatch/docs/client/status

Appendix G. The Visibility Map

- App Control and Code Integrity events are logged in Microsoft-Windows-CodeIntegrity/Operational, covering executables, DLLs and drivers; 3076 is the audit-mode block ...: https://learn.microsoft.com/en-us/windows/security/application-security/application-control/app-control-for-business/operations/event-id-explanations
- Sysmon is a Sysinternals service and driver that, once installed, logs to the Windows event log; event ID 6 "Driver loaded" includes hashes and signature information; ...: https://learn.microsoft.com/en-us/sysinternals/downloads/sysmon
- Measured Boot logs record PCR changes and are located in C:\Windows\Logs\MeasuredBoot; they can be decoded with TBSLogGenerator.exe or PCPTool.exe: https://learn.microsoft.com/en-us/troubleshoot/windows-client/windows-security/decode-measured-boot-logs-to-track-pcr-changes
- Tbsi_Get_TCG_Log retrieves the Windows Boot Configuration Log, also called the TCG log: https://learn.microsoft.com/en-us/windows/win32/api/tbs/nf-tbs-tbsi_get_tcg_log

- Security event 4624, "An account was successfully logged on", is generated on the computer where the logon session was created (Security channel): https://learn.microsoft.com/en-us/previous-versions/windows/it-pro/windows-10/security/threat-protection/auditing/event-4624
- Windows MDM events are logged under Microsoft-Windows-DeviceManagement-Enterprise-Diagnostics-Provider, Admin channel by default: https://learn.microsoft.com/en-us/windows/client-management/mdm-collect-logs
- Intune audit logs record create, update, delete, assign and remote actions; auditing is enabled for all customers and cannot be disabled; exports show the actor and …: https://learn.microsoft.com/en-us/intune/governance/monitor-audit-logs
- Entra ID audit and sign-in logs are retained seven days on Free and 30 days on P1 and P2; Microsoft Graph activity logs require P1 or P2 and are not retained unless …: https://learn.microsoft.com/en-us/entra/identity/monitoring-health/reference-reports-data-retention
- Microsoft Graph activity logs are an audit trail of all HTTP requests Microsoft Graph receives for a tenant, including from AI clients; fields include AppId, UserId, …: https://learn.microsoft.com/en-us/graph/microsoft-graph-activity-logs-overview
- Okta maintains a 90-day sliding window of System Log events: https://developer.okta.com/docs/concepts/monitor/
- Endpoint Security event types include ES_EVENT_TYPE_NOTIFY_KEXTLOAD, ES_EVENT_TYPE_AUTH_KEXTLOAD, ES_EVENT_TYPE_NOTIFY_EXEC, ES_EVENT_TYPE_NOTIFY_AUTHENTICATION, …: https://developer.apple.com/documentation/endpointsecurity/es_event_type_t
- Managed Device Attestation: iOS 16, iPadOS 16.1, macOS 14, tvOS 16 or later; A11 Bionic or later and Apple silicon Macs; properties verified through the Secure Enclave …: https://support.apple.com/guide/deployment/managed-device-attestation-dep28afbde6a/web
- MDM command responses carry a Status of Acknowledged, Error, CommandFormatError, Idle or NotNow: https://developer.apple.com/documentation/devicemanagement/sending-mdm-commands-to-a-device
- Linux audit sample rules watch kernel module insertion with -a always,exit -S init_module,finit_module -F key=module-load: https://github.com/linux-audit/audit-userspace/blob/master/rules/43-module-load.rules
- auditctl: auid is the original ID the user logged in with (loginuid); -w inserts a watch on a file system object: https://github.com/linux-audit/audit-userspace/blob/master/docs/auditctl.8
- Linux audit record types USER_AUTH and USER_LOGIN: https://github.com/linux-audit/audit-userspace/blob/master/lib/msg_typetab.h
- Linux kernel defines the module_load tracepoint: https://github.com/torvalds/linux/blob/master/include/trace/events/module.h
- Linux exposes the firmware TPM event log in securityfs as binary_bios_measurements under the TPM's directory: https://github.com/torvalds/linux/blob/master/drivers/char/tpm/eventlog/common.c
- Preboot firmware keeps a log entry for each value hashed into a PCR, used for remote attestation; reference is the TCG PC Client Platform Firmware Profile: https://docs.kernel.org/security/tpm/tpm_event_log.html
- TCG PC Client Platform Firmware Profile defines requirements for platform firmware to initialize and interact with a TPM 2.0 device (version 1.06 revision 52, December …: https://trustedcomputinggroup.org/resource/pc-client-specific-platform-firmware-profile-specification/
- IMA exposes ascii_runtime_measurements and binary_runtime_measurements in securityfs: https://github.com/torvalds/linux/blob/master/security/integrity/ima/ima_fs.c
- A verifier replays the event log against a TPM quote rather than trusting the log alone: https://ima-doc.readthedocs.io/en/latest/event-log-format.html
- fwupdmgr get-history shows the firmware update history stored by fwupd: https://fwupd.github.io/libfwupdplugin/fwupdmgr.html
- Android READ_LOGS: "Not for use by third-party applications, because Log entries can contain the user's private information": https://developer.android.com/reference/android/Manifest.permission#READ_LOGS
- Android SecurityLog tags: TAG_OS_STARTUP (Verified Boot state green, yellow or orange, and dm-verity mode), TAG_APP_PROCESS_START (process name, APK SHA-256), …: https://developer.android.com/reference/android/app/admin/SecurityLog
- setSecurityLoggingEnabled is called by the device owner or the profile owner of an organization-owned managed profile; non-device-owner callers see fewer or redacted …: https://developer.android.com/reference/android/app/admin/DevicePolicyManager#setSecurityLoggingEnabled(android.content.ComponentName,%20boolean
- Network logging records DNS lookups and TCP connections per app; Android 8 for a device owner, Android 12 for a work profile owner; paused while an unaffiliated user or …: https://developer.android.com/work/dpc/logging
- Play Integrity verdicts are read by the app's server; MEETS_STRONG_INTEGRITY on Android 13 and higher requires security updates in the last year: https://developer.android.com/google/play/integrity/verdicts

- Key attestation certificate chains are sent to a separate trusted server for validation: https://developer.android.com/privacy-and-security/security-key-attestation
- Intune Windows compliance Device Health settings use the Device Health Attestation service: require BitLocker (measured at boot), Secure Boot, code integrity: https://learn.microsoft.com/en-us/intune/intune-service/protect/compliance-policy-create-windows
- Security event 4697, "A service was installed in the system" (Audit Security System Extension): https://learn.microsoft.com/en-us/previous-versions/windows/it-pro/windows-10/security/threat-protection/auditing/event-4697
- Linux audit record type AUDIT_BPF (1334) for the BPF subsystem; program LOAD and UNLOAD records, added in Linux 5.6 (commit bae141f54be8): https://github.com/torvalds/linux/blob/master/include/uapi/linux/audit.h
- Purview Audit (Standard) retains records 180 days (logs from 17 October 2023); Audit (Premium) retains Exchange, SharePoint and Entra records one year; MailItemsAccessed …: https://learn.microsoft.com/en-us/purview/audit-solutions-overview
- MailItemsAccessed for investigating compromised accounts: https://learn.microsoft.com/en-us/purview/audit-log-investigate-accounts

Appendix H. Annotated Contract Clauses

- CRA Article 14(2)(a): early warning of an actively exploited vulnerability within 24 hours of awareness; Article 14(8): the manufacturer shall inform impacted users of …: https://eur-lex.europa.eu/legal-content/EN/TXT/HTML/?uri=OJ:L_202402847
- CrowdStrike, 20 July 2024: customers running Falcon sensor for Windows 7.11 and above, online between 04:09 and 05:27 UTC on 19 July 2024, may be impacted: https://www.crowdstrike.com/en-us/blog/falcon-update-for-windows-hosts-technical-details/
- DORA, Regulation (EU) 2022/2554, Article 30: key contractual provisions, including audit, access and inspection rights and exit strategies for ICT services supporting …: https://eur-lex.europa.eu/eli/reg/2022/2554/oj/eng

THE SERIES

Every Pixel Matters

One book for the whole stack, then one book for each layer that decides what your customers see. The companion series, GDCD in Code, covers how to govern the layers you choose to own.

Book 1	**From Bit to Pixel**: the whole stack, history, own or rent
Book 2	**From Code to Glass**: the rendering layer, from code to glass (this book)
Book 3	**From Silicon to Screen**: security from the chip to the screen
Book 4	**Data in Motion**: storage, networking, distributed systems
Book 5	**The Owned Stack**: toolchain, IDE, building an owned system
Book 6	**The Pixel Runtime**: building our own runtime (planned)

Juan Cardena

Enterprise AI & Agentic Systems Architect · blog.jcardena.com

Download the latest edition at blog.jcardena.com/en/ebooks/from-silicon-to-screen.

Every Pixel Matters · Book 3 · From silicon to screen.