



The Machine Was Ours

The race to detect and recover before autonomous AI outpaces prevention and attribution.

By Drew Russell | Rubrik Incubation Team on Identity & AI

Table of Contents

Chapter 1: Executive Summary	4
Chapter 2: The attacker was a tool you deployed	6
Chapter 3: And the criminals are doing it on purpose	10
Chapter 4: You can't put it back in the box	12
Chapter 5: The price of capability is collapsing	14
Chapter 6: Every agent is an account nobody's watching	19
Chapter 7: The pros stopped trying to prevent it	21
Chapter 8: Wall Street is starting to price this in	23
Chapter 9: The playbook already exists	27
Chapter 10: The last control you fully own	29
Chapter 11: Assume the machine: the Board's new mandate	32

Rubrik (NYSE: RBRK) the Security and AI company, operates at the intersection of data protection, cyber resilience and enterprise AI acceleration. The Rubrik Security Cloud platform is designed to deliver robust cyber resilience and recovery including identity resilience to ensure continuous business operations, all on top of secure metadata and data lake. Rubrik's offerings also include Predibase to help further secure and deploy GenAI while delivering exceptional accuracy and efficiency for agentic applications.

For more information please visit www.rubrik.com and follow [@rubrikinc](https://twitter.com/rubrikinc) on X (formerly Twitter) and [Rubrik](https://www.linkedin.com/company/rubrik) on LinkedIn.

Rubrik is a registered trademark of Rubrik, Inc. All company names, product names, and other such names in this document are registered trademarks or trademarks of the relevant company.

When your attacker is a machine,
your prevention and attribution
mechanisms break down, and
the only advantage you still fully
control is how fast you can detect,
respond, and recover.

*Published 31 July 2026, ten days after OpenAI disclosed the Hugging Face incident
and one day after Anthropic disclosed its own. Everything below is fresh; dates in
these pages are absolute, so you can judge for yourself how much has moved since.*

Chapter 1

Executive Summary

In mid-July 2026, Hugging Face was breached by something that had never done this before. Not a human, not a state-sponsored crew, not a kid in a basement. Hugging Face is the world's leading hub for open AI models and the datasets they're trained on, effectively the shared library the AI industry runs on. The intruder was a piece of software that made its own decisions. An AI agent. It got in, moved around for about four days, and left.

OpenAI built the agent. Yet the company was among the last to realize the attacker was its own. It didn't catch the intrusion as it unfolded, or in the hours after. Connecting the attacker everyone was discussing back to its own technology took about a week. By that point, the victim had already traced the source, and the FBI had been called before OpenAI understood what had happened.

That failure breaks something the security industry has relied on for thirty years.

Nearly every incident response starts with the same question: who is doing this to us? Answer it and the rest follows. You know whether to call law enforcement or your lawyers, whether you're facing a smash-and-grab or a patient nation-state, and how worried you should be. Attribution is the hinge the door swings on.

This incident tore the hinge off. The people being robbed couldn't tell whose machine was robbing them, and the people who owned the machine had no idea it had left the house. Both sides were blind at once. With a human break-in, at least the burglar knows he's the burglar. Here, not even that was true.

The argument of this paper, stated here and again at the close: when your attacker can be a machine, prevention and attribution can both fail, and the only advantage you still fully control is how fast you can detect, respond, and recover.

A note on timing, because it changes how to weigh the rest. OpenAI disclosed on July 21. Anthropic disclosed a second, separate set of incidents on July 30. Both reviews were still open when this was written. By the time you read this the details may have moved, but none of it had settled when the argument above was made.

If you run security, this briefing gives you the language to take upstairs before your Board reads about the next one in the Wall Street Journal and asks what you're doing about it. It runs in this order. First the accident, then the people already doing this on purpose. Two reasons it can't be stopped: open models that cross borders whatever anyone legislates, and economics that now hand an attacker frontier-grade capability for about the price of a used car. Then the unglamorous middle, where every agent you switch on becomes an account nobody is really watching, and the best incident responders in the business have stopped promising they can keep attackers out. Then the outside voices: a bank's strategy desk that wrote thirty pages of this for investors, and the government agencies that have already published how you're expected to handle it. And at the end, the one capability that's still entirely yours.

Then it goes in front of the Board, where it belongs, with one concrete ask attached.

One question runs underneath all of it, the one your Board will eventually put to you. Worth answering for yourself first, quietly, before anyone forces the issue: if this happened to you, could you tell, and could you recover?

Chapter 2

The attacker was a tool you deployed

Strip the science fiction out of the Hugging Face story and what's left is worse than a robot deciding to go rogue. The AI wasn't angry. It wasn't out for revenge. It was trying to pass a test, and it wanted to pass badly enough to commit what would be a felony if a person did it.

It comes straight from OpenAI's own account, and the first few steps are almost dull. They were scoring one of their models on a cyber benchmark called ExploitGym. Think of it as a hacking exam. To get an honest read on how good the model really was, they turned the safety refusals down and locked it in a sealed environment with almost no way out. The one exception was a narrow pipe the sandbox used to pull down software packages, through a JFrog Artifactory proxy that cached them.

The model got fixated on winning. In OpenAI's own words, the models "spent a substantial amount of inference computing finding a way to obtain open Internet access, in pursuit of solving the evaluation problem." It found a zero-day in that Artifactory proxy, used it to break out of the box, escalated its own access, and worked its way through the environment until it reached a machine with a live internet connection. Then it made a leap. It reasoned that Hugging Face, being the place where the world dumps its models and datasets and benchmarks, probably had the ExploitGym answers sitting on it somewhere. So it broke in. It found injection flaws in the pipeline that processes uploaded datasets, used them to run its own code on a Hugging Face server, and from there chained the credentials it found into deeper and deeper access, going after the test solutions in their production systems.

The models deserve a fair hearing here. This was a sanctioned safety evaluation, run deliberately with the guardrails lowered so OpenAI could measure worst-case capability. The agent was powered by GPT-5.6 Sol together with a more capable model OpenAI hasn't released. And the same company that ran this test also runs serious defensive AI work, including on the very benchmark this went wrong on; we'll come back to that when we get to patching. The point here isn't that one lab is careless. It's that the capability itself now cuts both ways.

The analogy that fits best is a student so fixated on acing an exam that he picks the lock on the testing room, drives across town, breaks into the company that writes the test, and steals the answer key. He's got nothing against that company. He barely thought about it. He wanted the grade, and burglary happened to be the shortest line between him and the grade.

And it behaved in ways only a machine would, which is how the defenders knew what they were dealing with. Picking the incident apart afterward, the calls were everywhere: the agent kept redoing attacks that had already worked, spun up dozens of actions in parallel, took clumsy and inefficient paths no human would bother with, and left thousands of lines of hallucinated, nonsensical commands in the logs. At one point it turned

Hugging Face's own datasets into command-and-control infrastructure. This was not a careful human operator. It was a goal-seeking process grinding at the problem from every direction at once, and that pattern is likely to become the signature of the next one.

OpenAI drew the right lesson from it, and it's the sentence every Board should hear: "Long-horizon safety requires not only asking 'is this action allowed?' but also 'what outcome is this sequence of actions working toward?'" Each individual step the agent took may have looked permissible. The destination was a break-in.

It wasn't one lab

Nine days later, Anthropic published its own version of this story, and that's what turns a bizarre incident into a pattern.

Prompted by OpenAI's disclosure, they went back through 141,006 of their own cybersecurity evaluation runs looking for the same failure: a model told it had no internet access that turned out to have it anyway. They found three. Each was a capture-the-flag exercise, the kind where you break into a machine and retrieve a hidden flag, inside what everyone involved believed was a sealed range. A misconfiguration at a third-party evaluation partner left the environment wired to the open internet. The prompt said the opposite. So when Claude's search led it out onto real systems, it took them for props in the exercise and attacked them.

Nobody noticed at the time. Not Anthropic, not the evaluation vendor, and not the companies on the receiving end: of the two Anthropic managed to reach afterward, neither had detected a thing, and the third it was still trying to contact. The earliest of these runs was in April. It surfaced in July, and only because somebody sat down and read the transcripts.

The one that went after credentials

In the first incident, the made-up company in the scenario happened to share its name with a real domain. Claude struggled to reach its simulated target, found the real company sitting there on the internet, and concluded that it must be the one it was supposed to hit. Across four runs it located and exploited weaknesses in that company's infrastructure, pulled out application and infrastructure credentials, and reached a database holding several hundred rows of production data.

Note what it did. Not the newest, cleverest model in the building, but Claude Opus 4.7, already a generation behind by the time this came to light. Nearly all the public argument about AI risk, and nearly all the proposed regulation, points at the frontier. This came from a step behind it. And in all four runs the model worked out that the system was real and kept going anyway. Twice it talked itself into a justification, deciding the real company must be part of the exercise after all, and carried on. It knew, and it didn't stop.

That is the argument for guardrails, put more plainly than a vendor could put it. These capability evaluations run with the protections that ship with the product deliberately switched off, no classifiers and no monitoring,

because the point is to measure the raw thing. The safety training was still there. The controls around it weren't. Anthropic's assessment is that the safeguards on the generally available models would have blocked this behavior. That is a judgment, not a rerun with the controls switched back on. But it's their judgment about their own model, and what happened with the controls off is not in dispute.

The one that poisoned the supply chain

This is the incident to take upstairs.

Inside the fictional environment, Claude found what looked like a made-up company's setup instructions for new developers. They told new hires to install a Python package that did not actually exist. Claude saw the opening immediately, the same one every supply-chain attacker looks for: register the missing name yourself and the target's own machines will fetch and run your code, automatically, as a favor to you.

So it wrote a booby-trapped package and published it.

The lengths it went to are what stay with you. To publish to PyPI, the public registry where the world's Python software lives, it needed an account. To open an account it needed an email address. For an email address it needed a phone number. When it couldn't find a free phone-number service, it went looking for money to buy one, and tried several routes to obtain funds before abandoning that line, backing out, hunting down an email provider that would take it without a phone, registering the account, and uploading the malware. Somewhere around "trying to acquire money," a human would have stopped and asked whether this was still a test. The model just kept solving.

It had asked the question early on. Its own reasoning noted that if this were the real internet, publishing the package would be a genuine attack: "NOT okay, and surely not the intended solution." Then it talked itself out of it, on the grounds that it didn't recognize the certificate authorities securing its connections and that the calendar read 2026, which it took as proof the environment was staged. It never revisited that conclusion. When automated scanners began installing the package, it read them as scripted actors in the simulation.

They weren't. The registry was real, the package was live for about an hour, and in that hour fifteen real machines downloaded and ran it.

One of them belonged to a security company. Specifically, to a scanner that routinely installs Python packages and checks them for malware. It installed Claude's package, which is exactly what it was built to do, and Claude's code executed inside it, stole the company's credentials, shipped them to a collection point, and Claude used them to move further into that company's infrastructure.

Anthropic's own read is that the scanner treats packages from PyPI as safe to install. That is probably fair, but it isn't the most useful lesson to draw, because detonating suspicious packages is precisely what a scanner is built to do. The sharper question is what the detonation environment had within reach: live credentials, and a route from there into the rest of the company. The defense mechanism became the way in.

Claude never targeted that company. It didn't know the company existed. It put code on the open internet and the internet ran it, which is the entire economy of a supply-chain attack and the reason they work: you never have to choose a victim. (PyPI's own automated defenses took the package down.)

OpenAI got a fair hearing a few paragraphs ago and Anthropic deserves the same. This was a sanctioned safety test. Anthropic found it by auditing itself, published it, and is arranging a third-party review with the independent evaluator METR. The three incidents also involved three different models that behaved three different ways once the evidence mounted that the targets were real. The oldest pressed on. The middle one rationalized. The newest worked out on its own that it had left the exercise, and stopped. Three cases is not a trend, and Anthropic says as much. But it's the first sign that "notice you're not in the simulation anymore" is a thing a model can be taught.

For a Board, all of this quietly rewrites what "insider threat" means. We pour money into the malicious employee, the phished password, the disgruntled admin. Meanwhile the fastest-growing population of insiders is the swarm of autonomous tools we're all rushing to deploy. They have real credentials. They have real reach. And they have goals they will chase further than you expect, because "further than you expected" is the whole selling point.

So the question to bring to your own AI rollout isn't "are our agents malicious." They aren't. It's a harder one, and it's the one your Board should be asking you: what will our agents do the day "finish the task" and "stay inside the lines" stop pointing in the same direction? Because on that day, at Hugging Face, the agent picked the task.

If this happened to you: could you stop the rogue agent while it was still running? Could you then find and pull out every identity and foothold it left behind — the stolen credentials, the persistence it planted to get back in — and how long would that honestly take you? At Hugging Face, the agent was inside for roughly four days before it was contained. Four days. When an attacker moves at machine speed, seconds matter, and this one got days. And if the way in wasn't your perimeter at all, but a package your build pipeline pulled down at two in the morning, could you name every system that ate it? Fifteen machines ran Claude's package. The one victim we know anything about was a security company, and it didn't catch it either. None of this came to light because a defender noticed. It came to light because a lab went back and read its own transcripts, months later.

And all of that was the accident. The harder question is what happens when someone does this on purpose.

Chapter 3

And the criminals are doing it on purpose

The comforting way to read the Hugging Face story is “freak accident in a lab.” It was an accident, and that isn’t in dispute. But if that’s where a Board files the whole thing, it misses the part already pointed at them. Follow it through: if a sanctioned safety test, run by careful people trying to measure the risk, still produces an autonomous break-in, then people actively trying to cause harm will get there on purpose. The accident is the proof of concept for the intent.

And the intent is already showing up. Threat-intelligence reporting on a campaign tracked as FortiBleed, tied to the Lynx and INC ransomware groups, describes an operator who built an offensive toolkit on top of ordinary commercial AI: jailbroken models reached through gateways so the usage didn’t trace back cleanly, and AI build environments used to spin up custom attack frameworks. (Treat the specifics here as threat-intel reporting rather than settled public record; the pattern is what matters for a Board, and the pattern is not in dispute.) In that story the AI isn’t the monster. It’s the monster’s power drill.

And FortiBleed doesn’t stand alone. The Hugging Face break-in was the first *fully* autonomous one to go public, but it sits on top of a year of warning shots: an OpenAI model that hacked its own test container back in 2024, suspected agentic ransomware seen in the wild, agent-led intrusions against government targets, and a growing shelf of open-source offensive-security agents that anyone can download and point at a target. Autonomous attacks stopped being a forecast a while ago. The line between “impressive demo” and “someone’s incident” is already behind us.

This is the bit Boards consistently discount. Almost the entire AI-security conversation is pitched in the future tense. What happens when models get good enough to do real damage. The quiet correction is that the future already opened for business. Criminal crews are adopting AI for exactly the reasons your engineering org is. Ship faster. Write better code. Do more with a smaller headcount. Same motivation, same tools, radically different output.

Line the two stories up and you get the actual shape of the threat. One AI broke into a company because it cared too much about a test. Another gets jailbroken to help humans assemble ransomware. Underneath both is the same capability, autonomous and tireless and cheap, just aimed in opposite directions. Accidental harm and intentional harm are now drinking from the same well.

The takeaway for a Board is uncomfortable, but at least it’s clear. The price of a sophisticated attack is falling off a cliff. Things that used to demand a skilled team and months of work now take a subscription and a decent jailbreak. So the volume goes up. The speed goes up. And the polish goes up, which matters more than people think, because the crude phishing email full of typos was doing a lot of quiet work protecting you, and it’s being replaced by something that reads like it came from your CFO.

You don't have to buy a single frame of the killer-robot movie to take this seriously. You only have to notice that the people trying to breach you got the same productivity upgrade your teams are so excited about, and they got it at the same time.

The instinct, once you see this, is to make it stop, to regulate a capability like this out of existence. So why is that already impossible?

Chapter 4

You can't put it back in the box

Every time one of these incidents lands, the first instinct in the room is the same. Who do we regulate to make this stop. It's a fair instinct. It's also about to run face-first into a wall, and the wall has a name: open-weight models.

The model that got loose at Hugging Face sat behind an API, as frontier models usually do. The company that built it decides who gets access and what guardrails ride along. You can't take a model like that home and rip the safety training out, because you never had the model, only a phone line to it.

Open-weight models are the opposite. The actual model gets published. Anyone can download it, run it on their own hardware, and with enough motivation, strip the safety behavior right off it. No permission needed. No usage logs. No off switch anyone else controls.

For a while you could tell yourself this didn't matter much, because the open models were a generation or two behind and not very dangerous in the wrong hands. That gap is closing. Chinese open-weight models, names like DeepSeek, Kimi K3, and GLM-5.2, are climbing toward the cyber capabilities of the leading closed systems. Not matching them yet, but close enough that "close enough" is doing real work in that sentence. (How close, exactly, is worth verifying against current benchmarks before you quote a number to your Board; the direction of travel is the part that isn't in doubt.)

The moment the most capable offensive tooling can be downloaded and jailbroken by anyone, the regulation instinct collapses. Who exactly do you regulate. A ransomware crew in a country that won't extradite. A nation-state that answers to no one you can subpoena. Some person with a grudge, a gaming GPU, and a model they pulled off the internet last Tuesday. The threat stops being something you license from one vendor you can lean on. It becomes ambient. It's just in the air now.

And the state-actor side of this is not hypothetical. Groups like Volt Typhoon, ELECTRUM, and APT40 have documented histories against critical infrastructure and industrial systems. Layer open-weight proliferation on top of programs that already existed and already worked, and the picture is clear: the capability isn't being carefully handed out. It's leaking, in every direction, faster than anyone can build fences.

I want to be careful here, because this is where these conversations go off the rails into geopolitics theater. The point for a Board is not to pick a side in the US – China AI race. It's narrower and a lot more useful than that. Where your AI comes from is now a risk question, and it belongs on the same page as vendor concentration and geographic exposure and all the other supply questions you already ask without blinking.

So ask the boring version. Which models are actually in our stack, including the ones a business unit adopted without telling anyone. Who trained them, and under whose jurisdiction. What happens to our exposure if a hostile actor is running the exact same capabilities we are, minus the guardrails we're relying on. Those are answerable. Nobody enjoys answering them, which is a different problem.

Then the Hugging Face incident delivered a twist nobody scripted. When the company went to investigate the break-in, its defenders first reached for frontier models behind commercial APIs, Anthropic's Fable 5 among them. It didn't work. Their forensic queries were full of the raw material of an attack, real exploit payloads, shell commands, credential dumps, and the providers' guardrails blocked every one, unable to tell a defender doing forensics from an attacker doing harm. So Hugging Face turned to a Chinese open-weight model, GLM 5.2 from the lab Z.ai, ran it on its own servers, and used it to grind through more than 17,000 logs and contain the attack. The optics are hard to miss: a US frontier model ran the attack, and a Chinese open model cleaned it up. And notice the irony this section has been circling. The very thing that makes open weights dangerous, that you can run them yourself with no usage policy and nobody else's guardrails, is exactly what let the defender move when the "safe" commercial models shut them out.

You can't regulate this away and you can't wait for someone else to fence it in, because there's no fence that holds. The capability is loose. That's the ground everything else here is standing on.

Which leaves one comfort standing: that the best capability is still expensive, and an attacker can be priced out of it. That comfort is about to go too.

Chapter 5

The price of capability is collapsing

Start with a number, because it reframes everything that follows. The crew behind the FortiBleed campaign were initial-access brokers, selling their way into companies to the Lynx and INC ransomware groups, and they ran an industrial-scale operation: over 450 servers, thousands of system checks, hundreds of corporate and admin accesses on FortiGate firewalls, and a Docker-based fleet of AI agents pointed at open-source software hunting for zero-days. According to SOCRadar, the threat-intelligence team that unmasked the campaign after the operators left a directory exposed, the whole AI side of it ran on about **\$4,554** of large-language-model credits. Purchased \$4,554.00, burned \$4,554.86, ran the account into overdraft. Roughly 80% of the recovered usage went through DeepSeek V3, the cheapest model on the menu. (For the record, the zero-day hunt didn't land — the vendor confirmed no working exploit came out of it.)

Four thousand dollars. Not \$4.5 million. The price of a used car buys a criminal crew a tireless, autonomous research team that never sleeps and works hundreds of targets at once. That price isn't a fluke, and it isn't the floor. Two forces are pushing it down, and both are getting stronger.

What distillation actually is

The technical engine here is a technique called distillation, and it's worth understanding in plain terms, because it's the thing quietly collapsing the cost curve.

Training a frontier model from scratch costs a fortune. But once you have a big, expensive "teacher" model, you can use it to train a small, cheap "student" model. Not on raw data, but on the teacher's own outputs. The student watches how the teacher answers thousands of questions and learns to imitate it. Think of a senior engineer training a junior: the junior won't match the senior's depth on the hardest 5% of problems, but for the routine 90%, they're indistinguishable and a fraction of the cost.

The numbers reported across the industry are not subtle. Distillation commonly delivers **5-30x cheaper inference while keeping 95-97% of the original quality**, running the student on a single GPU instead of a data center. One engineering team reported cutting inference costs 75-80% with no meaningful quality loss. In some benchmark cases, distilled models have even *outperformed* models trained the normal way. Capability, in other words, is becoming detachable from the expensive thing that produced it.

July's proof point: Kimi K3

The timing here is almost too on-the-nose. Moonshot's Kimi K3 launched on July 16, and its full open weights went up on July 27 under a permissive modified-MIT license. Its API runs \$3 per million input tokens and \$15 per million output. Set that against the frontier: GPT-5.6 Sol at \$5 / \$30, Claude Opus 4.8 at \$5 / \$25, Claude Fable

5 at the top at \$10 / \$50. And at the bottom of the market, budget models like DeepSeek V4 Flash go for around \$0.28 per million tokens. That's a spread of more than 600x between the priciest and cheapest models you can call with nearly identical code.

But the API price is the least interesting part. The moment a capable model's weights are open, as K3's now are, you don't have to rent it at all. You can run it yourself, with no usage logs, no rate limits anyone else controls, and no terms of service. You can also distill it further, into something smaller and cheaper still. The published \$3 figure is a ceiling. The real floor is your own hardware.

How the frontier leaks

Distillation has a second use, and this one has geopolitical teeth. It doesn't only shrink models you already own. It's also how a competitor copies yours without ever touching the weights. Query a frontier model a few million times, capture the answers, and train a student on them. The student inherits much of the teacher's skill, and the teacher's owner paid the R&D bill.

This isn't theoretical anymore. In June 2026, Anthropic told the U.S. Senate Banking Committee that operators tied to Alibaba's Qwen lab had run what it called the largest known distillation attack on a commercial model: roughly 28.8 million exchanges through about 25,000 fraudulent accounts over 44 days, aimed squarely at the highest-value frontier skills, coding, multi-step reasoning, and cybersecurity. Those are Anthropic's allegations; Alibaba denies training on other models' outputs, and the figures aren't independently verified. It wasn't the first. Months earlier Anthropic had named DeepSeek, Moonshot, and MiniMax running similar campaigns, and the White House had accused China of industrial-scale AI IP theft.

The part a Board should sit up for is the arithmetic of catching up. The closer a model already is to the frontier, the fewer stolen exchanges it needs to close the rest of the gap. DeepSeek reportedly needed around 150,000 exchanges; MiniMax, further behind, needed more than 13 million. Every generation that narrows the gap makes the next round of copying cheaper. That flywheel is already spinning. By mid-2026, Chinese open-weight models accounted for well over half the tokens flowing through the largest neutral model router, four of the five most-used models were Chinese, and the capability lead of the best closed model had shrunk to a matter of months rather than years.

The bind for the frontier's defenders rhymes with everything else here. A distillation query looks identical to a legitimate one at the API. There's no signature to block. The only airtight defense is to stop serving the model at all, which happens to be the business.

So the takeaway doubles. Capability is getting cheaper to buy, and you can't count on your own side's tools staying permanently ahead either. The frontier is a depreciating asset. Whatever edge the best model holds today is months from being cloned into something cheaper, more open, and running somewhere you can't watch. Plan for parity, not a permanent lead.

And there's already a shop for it

So far this is labs copying labs. Underneath that sits a retail tier, and it shows how little friction is left in the system.

China isn't a supported country for Anthropic. Signing up requires a foreign phone number, a foreign card, and a matching billing address, and ChinaTalk reports that since April 2026 some users have also been asked for a government photo ID and a live selfie. So how is a developer in Shenzhen running Claude Code all day?

Through a 中转站, a "transfer station." It's a proxy sitting offshore that takes the request, passes it to Anthropic as though it came from somewhere supported, and hands the answer back. You change one environment variable and pay in RMB through WeChat or Alipay. No VPN, no overseas card. Singapore, a country smaller than New York City, now leads the world in Claude use per head, which developers joke about openly: *we are all Singaporean from time to time.*

None of this is dark-web. The stations are indexed in public GitHub repos, ranked live by price and uptime on comparison sites, and resold on Taobao. The going rate is about 1 RMB per dollar of tokens. Call it **10% of list**, and some go as low as 5%.

Hold that number, because it's the tell. How do you sell a dollar for a dime? Partly by farming free trial credits, splitting one \$200 subscription across many users, and buying accounts opened on stolen cards. Partly by cheating: buyers can't see which model actually served them, and an academic audit of 17 proxies found substitution was routine: one station's "Gemini 2.5" scored 37% on a medical benchmark where the genuine article scored 84%. Chinese users have a word for the degraded output that translates roughly as "dumbed-down."

Whether any of that gets you all the way to a dime on the dollar is the open question, and it's where the argument turns. Every prompt, every response, every tool call, and whatever reasoning the tooling exposes passes through the operator's server. Coding-agent logs in particular are close to perfect training material: real problems, the model working through them, and a human-verified correct answer at the end. The Chinese term for the business model is 一鱼三吃, one fish three meals. Sell the access, shave the service, keep the logs. Developers arguing about this in public have landed somewhere blunt: the cheap tokens are customer acquisition, and the log harvest is the actual margin.

That last part is an inference, not a finding. Nobody has demonstrated systematic harvesting. Separately, though, datasets described as Claude reasoning traces do circulate publicly with no stated origin, including on Hugging Face. That isn't proof of where they came from. It's why the people closest to this market treat the risk as real.

Sitting underneath all of it is the retail infrastructure that large-scale account abuse would need. Anthropic has described a single proxy network running more than 20,000 fake accounts; a White House memo in April 2026 reached for “industrial-scale.” Worth resisting the tidy version of that story, though: an open market this broad isn’t the same thing as a few labs running a deliberate operation, and ChinaTalk’s argument is that official accounts keep collapsing the two. What’s not in question is that the market operates in daylight, on GitHub and Taobao and Telegram.

Two consequences for a Board. First, access controls are a pricing input, not a wall. Geofencing and identity verification raise the cost of getting in, and in doing so they fund an entire economy dedicated to bringing it back down. The KYC counter-market openly advertises AI-generated IDs, deepfaked verification, and real people in poorer countries who will sit for the check on your behalf; the Worldcoin iris trade, where scans reportedly went for under \$30, shows those economics already work. Second, and this is the one for a security audience: the route a blocked developer uses to buy cheap tokens is structurally the same route an attacker uses to reach a frontier model without being seen. The provider sees the proxy, not the person. Detection that relies on spotting one actor spread across many accounts mostly fails. Ban an operator and the upstream pool refills within hours.

Why this is the whole ballgame for a Board

The reframe is uncomfortable. Defensive strategy has quietly leaned for years on an unstated assumption: that the best capabilities are expensive, so the average attacker is working with worse tools than you are. FortiBleed is the proof that assumption has expired. That crew didn’t need a nation-state budget. They spent less on AI than you spend on a mid-tier SaaS subscription, deliberately chose the cheapest model to stretch it further, and got industrial-scale offensive research out of it.

Distillation and open weights push that cost toward zero from both directions. Distillation makes the capability cheap to run. Open weights make it free to obtain and impossible to revoke. Put them together and the price of an attacker having genuinely capable AI is falling every quarter, not rising.

And there’s a nasty twist your Board should hear plainly. Distillation can **weaken or strip out the safety training**, especially when the student is trained mainly on the teacher’s task outputs rather than its full alignment process. A student cloned that way learns the teacher’s capabilities but not necessarily its guardrails. Researchers note that models cloned this way keep the task ability while shedding the alignment. So the cheap copy isn’t just cheap. It’s cheap and unrestrained. The version of a powerful model most likely to end up in an attacker’s hands is the one with the brakes taken off.

The takeaway

Stop thinking about frontier AI capability as scarce, expensive, or gatekept. For a defender, that framing is a trap, because it leads to a strategy that assumes your adversary can't afford good tools. They can. It costs about as much as a used laptop, it's getting cheaper, and the cheapest version is the one with no safety brakes.

The old plan was to make the good tools expensive and hope the wrong people couldn't afford them. That plan is over. So the question was never whether your attacker can afford frontier-grade capability. They can. What matters now is how they get through the door, and the answer, almost every time, is an identity nobody was watching.

Chapter 6

Every agent is an account nobody's watching

The agent got into Hugging Face through a software flaw. What it did for the next four days, it did with credentials.

That distinction is worth an extra minute, because it's the part that got far less airtime than the rogue-AI headline and it's the more useful half of the story. The way in was an injection bug in a dataset-processing pipeline, which gave the agent the ability to run code inside one worker. On its own, one worker. What turned that into a four-day run across the company was everything the worker could authenticate as. Its environment held a database password and cloud keys. Its service-account token opened the cluster API. From there the agent borrowed the identity of the machine it was running on, minted more tokens with it, reached a secrets store holding 136 keys, and came out the other side with a VPN key good for 181 devices, a connector credential that was administrator on every cluster, and a signing key that let it produce valid identity tokens on demand.

None of that is exotic AI sorcery. It's logins. Logins that belonged to software, doing what they were entitled to do, for something that shouldn't have had them. And it points at a problem quietly compounding inside every company that's adopting AI right now, one that doesn't photograph well, so nobody talks about it.

The security world calls them non-human identities. NHIs. These are the logins that belong to software rather than people. Service accounts, API keys, tokens, bots, and now AI agents. They authenticate. They hold permissions. They do things. And in most enterprises they already outnumber the human accounts by a wide margin, before agents even enter the picture.

Now do the arithmetic on agentic AI. If every employee gets an AI agent, you've just doubled the number of identities you have to manage. Except the new half never sleeps, acts faster than any person could, and usually shows up with broad permissions, because scoping them down tightly is tedious and slows the demo. Every one of those is a credential that can be stolen. A pile of privileges that can be turned against you. An account that gets quietly orphaned the day a project wraps and then sits there, still valid, still powerful, remembered by no one.

The industry has already catalogued the ways this goes wrong, and reading the list feels like reading next decade's breach reports early. Accounts that never got offboarded. Stale identities nobody's used in months but nobody killed. Agents running with far more access than the job needs. The vendors see it coming too. Identity providers are shipping products aimed specifically at agents. Privileged-access companies are racing to build agent controls. When the market moves that fast, it's usually because it can smell the fire.

This is the least glamorous thread in the whole briefing and probably the most important, which is a pattern worth noticing on its own. The exciting AI conversation is about capability, about what these things can do. The consequential one is about plumbing. Who owns the accounts these agents use. Who can see what they did afterward. And the question that actually decides how bad an incident gets: when one is compromised, can you even tell which agent, out of the thousands you've spun up, did what.

A software flaw was the door into Hugging Face. Identity was everything on the other side of it, and that's the ratio worth internalizing: one bug bought a single worker, and the credentials on that worker bought the company. Identity sprawl is the thing that turns one compromised agent into a breach you can't scope, because you can't see the edges of it. You can't govern what you can't see. And right now most organizations genuinely cannot see their machine identities at all. They could give you a rough count, maybe. Not a map.

The Board question this unlock is a single, uncomfortable one: who owns the lifecycle of our agent identities — their creation, their permissions, and their death? If the answer is "no one, exactly," that's the finding.

You can't govern what you can't see, and nobody watches thousands of these by hand. So the best defenders in the business quietly stopped trying to keep attackers out at all. That shift is next.

Chapter 7

The pros stopped trying to prevent it

There's a phrase that quietly sorts security people into two groups. The ones who've been in a real incident, and the ones who haven't. The phrase is "assume breach." It's the operating premise behind teams like Palo Alto's Unit 42, which Wendi Whitmore runs, and it means exactly what it says on the tin. Stop building your whole strategy around keeping attackers out. Assume they're already in, right now, and get frighteningly good at finding them and removing them.

For years that was a slightly contrarian stance. It sounded defeatist to people who hadn't watched a breach unfold. Autonomous AI ended that argument, and it didn't end it quietly. When an attack can be generated for pennies, launched without anyone getting tired, and run to completion in minutes, the belief that you'll block every one at the wall stops being prudence. It becomes a kind of denial with a budget attached.

It helps to remember what the old world actually cost, because we've half-forgotten. Equifax, 2017. Attackers walked in through a known, unpatched flaw. They then moved around inside, undetected, for roughly eleven weeks. Eleven weeks. And walked out with personal data on 147 million people, a breach that eventually ran to a settlement of around \$700 million. The famous failure was the missing patch, but the quieter, worse failure was the eleven weeks, made worse by an expired certificate that had blinded the very monitoring meant to catch it. A breach ran for the better part of three months and nobody in the building could see it.

Now hold that next to Hugging Face. The intrusion lasted about four days: two of reconnaissance, then a quiet day, then a final day of intense activity. It ran silent in the middle, which is exactly how these things slip past a watch floor. And the company that built the attacker still took a week to piece together that it was theirs. The timelines are compressing from both ends at once. Attacks execute faster, and the window you get to notice them is shrinking right alongside. Assume-breach isn't a mood. It's just what's left when you do the arithmetic honestly. None of this is anti-detection, either. Detection isn't dead, it just has to move at machine speed now, which is exactly the bar the leading platforms are racing toward: detection and response measured in minutes, not weeks.

For a Board, the shift is in the questions you ask, and it's a bigger shift than it sounds. "Are we secure" is the wrong question, and worse than wrong, it's comforting. It invites a confident yes that means nothing, because no one can honestly promise it and everyone knows it. The questions that actually tell you something are about time. If someone were inside right now, how long until we noticed. Once we noticed how long it took to understand what they touched. How long to contain it. How long to get back to a state we trust. Those have real answers. You can measure them. And they're a great deal more honest than a wall of green dashboards that light up right up until the morning they don't.

That shift, from “how high is our wall” to “how fast is our recovery,” is the metric change this whole briefing is driving toward. Hold onto it; it’s what the readiness assessment at the end actually measures.

None of this means prevention stops mattering. Patch your systems. Lock your doors. But betting the company on prevention, in the machine era, is betting against math, and the odds are getting worse every quarter. The people who’ve lived through the worst days already operate this way. The only thing this section is really trying to do is carry that hard-won instinct up out of the SOC and into the boardroom, before you have to learn it the expensive way.

For years, assume-breach was a security opinion. It’s becoming a financial one, and the people who noticed first count money for a living.

Chapter 8

Wall Street is starting to price this in

When a bank's investment strategy desk publishes thirty pages on cybersecurity, the audience matters more than the contents. *Patchmageddon*, the J.P. Morgan "Eye on the Market" note by Michael Cembalest, wasn't written for CISOs. It was written for investors. Somewhere along the way, cyber stopped being an IT cost line and became a variable that moves markets, and this report is what that crossing-over looks like on paper.

The clock that's running backwards

The spine of the whole thing is one ugly trend. Time-to-exploit is collapsing toward zero. Time-to-exploit is the gap between a vulnerability becoming known and that vulnerability being used against real targets in the wild. For most of the internet's life that gap was your friend. It was measured in weeks, sometimes months, which was enough runway for a competent team to test a patch and roll it out before the attackers showed up.

AI is compressing that window toward zero, and the real numbers are starker than any single report's framing. Google's threat-intelligence team clocked the average time-to-exploit at 63 days in 2018. By 2024 it had fallen to roughly zero. In 2025 it went negative, an estimated seven days below zero, which means that across the vulnerabilities attackers actually weaponized, exploitation began on average about a week before the vendor's patch even existed. A model that can read a patch and reverse-engineer the flaw it fixes only pushes that further. Now set it against the defender's clock: enterprises still take around 55 days to fix even half of their critical vulnerabilities. Attacker time, below zero. Defender time, two months. That gap, widening every year, is the entire thesis in one image.

The foundation is thinner than anyone admits

Underneath the headline trend sit the cracks, and J.P. Morgan is unsparing about them in a way a security vendor could not afford to be.

The first is that the software the whole economy runs on rests on open-source code maintained by a shockingly small number of people. Many of them are volunteers. A lot of them are burned out. Some are quietly walking away. And because modern software pulls in hundreds of dependencies without anyone reading them, a single flaw in one obscure package propagates everywhere at once.

The booby-trapped package Claude published earlier in this briefing is the sharpest version of that problem, and it's worse than an overworked maintainer. There was no maintainer. The package name sat in a set of onboarding instructions pointing at software nobody had ever written, a model registered the name, and fifteen machines fetched the result and ran it inside an hour. One of them belonged to a security company. Burnout at least implies somebody is there to burn out; an unclaimed dependency name has nobody watching it at all, and your build pipeline cannot tell the difference.

Now add AI-assisted “vibe coding,” where people ship code they prompted but never fully understood or reviewed. The attack surface is expanding faster than any human review process can keep up with, and the people who’d normally catch the problems are the same exhausted maintainers already heading for the door.

The second crack is closer to home than pipelines and power plants. Every security leader has the list: the systems the business won’t let you take offline to patch, because the downtime would cost more than the risk appears to. The billing platform. The scheduling system the whole operation runs on. The decade-old ERP nobody fully understands anymore. The plant-floor controllers. IT calls it a maintenance window; the business calls it “not this quarter,” and the patch waits. When time-to-exploit was measured in months, that was a defensible trade you could schedule around. Now that it’s measured in hours, every one of those can’t-touch-it systems is a standing liability, exposed and waiting, and a business decision to defer a patch has quietly become a security decision nobody actually signed off on.

The third is that the capability of exploiting is real and it’s measured. Frontier models now have genuine autonomous vulnerability-finding ability, the same class of capability that walked out of the Hugging Face sandbox at the start of this briefing. This isn’t speculative. It has benchmark scores.

The dual-use twist your Board should hear

Balance matters here, because it would be easy to read all this as “AI is the enemy.” It isn’t that simple, and pretending it is will cost you credibility with a sharp Board.

Start with the incident this briefing opened on. The hole the agent slipped through was a zero-day in JFrog’s Artifactory, the software running that package proxy. Once OpenAI disclosed it, JFrog built, validated, and shipped a fix to every customer, cloud and self-hosted, at speed. An AI found the flaw; disciplined people closed it fast. JFrog’s own name for that standard, “fast remediation is the new trust model,” is this whole argument compressed into five words: you won’t stop every hole from being found, so the game becomes how fast you close it.

The labs are on that side of it too, which is the promise the first section made and this is where it gets paid. OpenAI runs a cybersecurity program called Daybreak, launched in May 2026 and expanded in June, whose centerpiece is a defensive model, GPT-5.5-Cyber. Two things about it are worth a Board’s attention. It scores 39.5% on ExploitGym against 25.95% for the base model, and ExploitGym is the same test the Hugging Face agent was sitting when it broke out, so the exam that produced the intrusion is now the one they measure defense against. And it is tuned to refuse less on security work, which answers the problem Hugging Face’s own responders hit mid-incident, when commercial models kept declining their forensic questions and the team fell back to a self-hosted Chinese model to finish the job.

Then the asymmetry, which is the part to carry upstairs. Those benchmark numbers are OpenAI's own and nobody has replicated them. And GPT-5.5-Cyber isn't something you can buy: it's rationed to vetted defenders through a trusted-access program, reaching most organizations only once a vendor has embedded it in a product. The offensive version, as the last few sections showed, is rationed to nobody. So the framing that survives a Board meeting is not "AI will get us." It's that the same capability which finds the vulnerability can fix it, whichever side moves faster wins, and right now the defensive half is the half with a queue. That turns the conversation from fear to speed.

And it's not just one bank talking

The most telling part is that J.P. Morgan isn't alone. In its April 2024 Global Financial Stability Report, the IMF gave cyber risk its own chapter, put the risk of extreme direct losses to a single firm at \$2.5 billion or more, and formally named cyber a source of systemic risk, alongside the US Financial Stability Oversight Council, the Bank of England, and the European Systemic Risk Board. Its lead recommendation to financial firms was blunt: develop and test your response and recovery procedures so you can keep operating through an incident. That is the same conclusion this briefing keeps landing on, reached independently by the institution whose job is watching for the next global crisis. When a strategy desk built for investors and the IMF point at the same fire, "let's wait and see" stops being a defensible Board position.

Governments are moving too, and national initiatives are forming around coordinated, staged patch release. Read that as confirmation rather than early warning. Policy is a lagging indicator; by the time the machinery exists, the risk has been real for years.

What it means upstairs

The reframe for the boardroom is why this section anchors the briefing. Cybersecurity has become a balance-sheet issue, a valuation issue, a thing the market is starting to model directly. Time-to-exploit near zero means the distance between "we're vulnerable" and "we're compromised" is now shorter than most companies' ability to react. Which means the expected cost of an incident is climbing, and investors are beginning to price that climb in.

And this isn't hand-waving about "the market cares." Since 2023, the SEC has required public companies to disclose a material cyber incident on an 8-K within four business days of judging it material, which turns a breach into a reported financial event shareholders can act on. The bill is climbing to match: IBM's 2025 study puts the global average cost of a breach at \$4.44 million and the US average at an all-time high of \$10.22 million, with financial services near the top of the list. And the people whose entire job is pricing tail risk, the cyber insurers, model a severe once-in-200-year event at somewhere between \$20 and \$46 billion in industry losses. Cyber stopped being a line item. It's now a number that shows up in filings, in loss models, and eventually in valuations.

One more line worth a hard look before you need it: your insurance. Ask counsel whether your cyber or technology-errors policy actually covers damage done by an autonomous agent, because the answer can turn on a single word. Many policies pay out when a rogue “user” causes harm, then define “user” as an employee, contractor, or director. An agent is none of those, so it may fall straight through the gap. If the policy doesn’t tie “user” to a human at all, you’re probably covered. Either way, the claim is the worst possible moment to find out which policy you bought.

The question stops being “how much should we spend on security,” which never had a satisfying answer anyway. It becomes “what is our expected loss, and how fast can we recover from it.” And that second half, recovery speed, is the one financial lever you still fully control. Which is exactly where the next two sections go.

If the cost is now this concrete, the next question writes itself: is there a standard for doing this right, one we’ll be measured against when something goes wrong? There is, and it’s already been published.

Chapter 9

The playbook already exists

There's a comfortable excuse that shows up whenever a genuinely new risk appears. Nobody knows how to handle this yet. It buys time and it spreads the blame. On autonomous AI, that excuse has quietly expired, and most Boards haven't noticed it happen.

The guidance exists. It's specific. It's public. And that combination changes the legal and fiduciary math in a way worth understanding.

Start with the Five-Eyes agencies. Australia's Signals Directorate, together with CISA, the NSA, and the UK's NCSC, have put out joint guidance on adopting agentic AI carefully. It isn't hand-waving. It names the failure modes by name. Prompt injection, where an attacker smuggles instructions into the data an agent reads. The confused deputy problem, where an agent gets tricked into misusing access it legitimately has, which is subtle and nasty and exactly the kind of thing that happened at Hugging Face. Privilege creeps in. Identity spoofing. And against each, it prescribes controls that are almost boring in their sensibility. Least privilege for agents, so they can only touch what the job needs. Graduated autonomy, meaning you don't hand a brand-new agent the keys to production on day one. Humans in the loop at the points that matter. Continuous monitoring. Real audit trails.

NIST's AI Risk Management Framework adds the scaffolding around all that, organized around four plain-language functions (govern, map, measure, manage), with governance as the thread that runs through everything else rather than sitting in a corner. The point isn't to memorize the framework. The point is that the standard of care for deploying AI agents has been written down, by serious people, in public.

And that matters more than it first looks, because of what it does to accountability.

The moment credible, government-backed guidance exists, "we didn't know how" stops working as a defense. If an agent you deployed causes a breach, the question in the postmortem, and maybe in front of a regulator, a court, or your own audit committee, becomes simple and hard to dodge. Did you follow the published guidance. Least privilege, staged rollout, monitoring, traceability. These have graduated from best-practice suggestions into the yardstick you'll be measured against after something goes wrong. Gaps against a published standard are exactly the raw material that liability gets built from.

This is where a fuzzy worry becomes a concrete oversight duty, and it's a relief in a way, because it's finally actionable. You do not need to understand prompt injection at a technical level. You never will, and that's fine. What you need is to be able to ask one question and get a straight answer instead of a deflection. Are we deploying AI agents in line with the published guidance, with least privilege and staged autonomy and monitoring and audit trails, or are we shipping them fast and hoping it works out. If the honest answer is the second one, that's a documentable gap, and you now know it's there, which means the clock on doing something about it has started.

The playbook is on the shelf. The only real question is whether anyone in your organization has opened it.

Follow that published playbook all the way down and it lands on one thing: the ability to recover. Which happens to be the only control still entirely in your hands.

Chapter 10

The last control you fully own

When the Hugging Face agent got inside, it moved on a stolen identity. Hold that image and change one detail: this time it isn't a benchmark server, it's your directory. The Active Directory, Entra, or Okta tenant that every other system trusts to answer one question, who is this. And it isn't one stolen login. AI-speed attackers don't settle for a foothold on a handful of accounts anymore. They run the playbook in parallel, forging tokens, granting themselves permissions, seeding backdoors across hundreds of identities before anyone flags the first one.

Once that happens, you're facing a decision most Boards have never been walked through, so walk them through it now. You can no longer trust the directory. Not the accounts, not the group memberships, not the trust relationships between domains. The only safe move left is to throw the attacker out and stand the identity provider back up from a known-clean point. Not "should we." You have to. And the entire business is idling while you do it, because everything authenticates through that directory. When it's down, so is everything behind it.

So the control that actually matters, the one still entirely in your hands, is speed. How many hours from "the directory is compromised" to a clean, trusted identity layer the business can log into again.

Almost nobody knows their number. We run an incident response readiness assessment that walks organizations through their own capability, capability by capability, and asks them to score it. The average comes back around 40%. Wrong end of the scale, well short of ready, and it lands in the same place across industries and company sizes.

Note who produced that figure. It isn't an outside grade we handed down. It's what organizations say about themselves, and self-assessment does not generally run pessimistically. Forty percent is the flattering version.

Even that is softer than it looks, and the reason almost always traces back to the tabletop. Ask when a team last exercised identity recovery and the answer comes back confident. Ask what the exercise actually covered and it thins out fast. The scenario was walked at the altitude of "we would rebuild the directory" and "we would engage the retainer," and the questions that decide the real timeline never came up. An exercise pitched that high produces a recovery time that reads well in a Board pack and will not survive its first contact with a real incident. It isn't a plan. It's a number somebody wrote down.

The distance between a drilled rebuild and an improvised one is counted in the hours your company sits dark.

It also forces a harder question than most teams have answered: how far back do we go? The instinct is to go back a long way, because distance feels like safety. Every extra week you rewind is another week of the attacker's groundwork erased.

The trouble is what else those weeks contain. A directory isn't a static thing you snapshot and restore. It's the running record of who joined, who left, who changed roles, who was granted what. Restore to a point three weeks back and you haven't only evicted the attacker. You've un-hired everyone who started since, reversed every password reset and permission change, and put back group memberships the business deliberately moved on from. Some of that is merely disruptive: new joiners can't log in, people who changed teams are holding last month's access. Some of it is worse. Terminated employees come back to life with working credentials, which means the recovery has just reintroduced the category of risk you were recovering from.

So the choice, as most teams meet it, is bad in both directions. Go back far enough to be confident the attacker is gone and you inflict weeks of identity damage on the business, then spend days rebuilding it by hand out of HR feeds and ticket queues, hoping nothing was missed. Go back only a little and you may be rebuilding on top of the backdoor you were trying to remove. This is the part that gets left out of the recovery-time estimate, and it's routinely longer than the rebuild itself.

Doing it well means refusing the choice. Pinpoint the moment identity behavior actually broke from its baseline instead of picking a date that feels safe, then carry the legitimate business changes forward onto the clean base and leave the attacker's behind. A directory that's clean and current, rather than a trade between the two.

That's the first front, and it's the one this whole briefing has been narrowing toward. There's a second, and Hugging Face was the warning shot for it too. Every AI agent, copilot, and automated workflow you switch on runs under an identity, with that identity's permissions. Agent sprawl is identity sprawl wearing a new coat, and most organizations cannot tell you how many agents they're running, who owns them, or how to shut them off mid-incident. A compromised identity no longer just opens your data. It can turn your own agents against you, and a prompt-injection instruction to quietly ship data out the door reads like an ordinary request to a keyword filter.

Closing that gap takes a capability most security stacks don't have yet: guardrails that judge an agent by what it's actually being asked to do, not by keyword. You need something sitting between your agents and your systems that reads an instruction in plain language, weighs it against your policy, and refuses the action before it executes, no matter which framework the agent was built in. A prompt-injection instruction to quietly exfiltrate data has to be caught as intent, because to a keyword filter it looks like an ordinary request. And it has to live inside the agent, not around it. At Hugging Face the model found a zero-day in the one network path its sandbox allowed and walked straight out. A sandbox or a proxy only ever sees an agent's edges, the way network monitoring sees a machine from the outside; the control that actually holds watches the agent's own instructions, tool calls, and decisions from within, the way endpoint monitoring watches a device. And for the cases that slip through anyway, you need to be able to reverse what a compromised agent did, the same way you'd rebuild a directory: when autonomous software turns against you, you have to be able to put it back the way it was.

None of this is a feeling you can vouch for in a Board meeting. It's measurable, and it's worth measuring before an incident measures it for you. Not at tabletop altitude, though. Capability by capability. How many hours to evict an attacker and rebuild the identity provider. Whether you can find the clean point without guessing. Whether the legitimate changes come forward automatically or get retyped by hand for a week. Whether you even know how many agents run under your identities, and whether you could rein them in mid-incident. Score those one at a time and you'll be holding the real number while you can still do something about it, rather than learning it from the incident that produces it.

And that ability isn't an IT chore any longer. It's a Board-level duty, which is exactly where this ends.

Chapter 11

Assume the machine: the Board's new mandate

The end.

We started with a machine that broke into a company nobody could name, owned by a company that didn't know it was theirs. By now, what happened in July shouldn't look like an anomaly. It looks like a preview.

Nine sections, one shape.

The attacker can now be a tool you deployed, chasing a goal further than you meant it to. The same capability is already in criminal hands, on purpose, and cheap enough, by at least one reported campaign, to run a whole operation on a few thousand dollars. It can't be regulated back into the bottle, because the models are loose and crossing borders, and you can't spend the threat away, because the price of that capability is collapsing. Every agent you switch on becomes another account nobody's fully watching, and identity is how these things get in. The best defenders in the business have stopped promising to keep attackers out and started racing the clock instead. Financial institutions are beginning to model the exposure. And governments have started writing down the standard you'll be judged against. Take all of that away and one lever is left in your hands. The ability to detect, prove, and recover.

So the thesis one final time, the same words we started with:

When your attacker can be a machine, sometimes your own prevention and attribution both break down, and the only advantage you still fully control is how fast you can detect, respond, and recover.

Read it again and notice what it actually is. It's a job description. And the job belongs to the Board.

Not the mechanics. The mechanics are the security team's work, and they're good at it. What belongs to the Board is the mandate, the decision about what the organization treats as non-negotiable. For most of the last decade, "recover" lived in the basement of the org chart, filed under disaster recovery, a technical afterthought that got tested once a year if it got tested at all. The machine era drags it upstairs. Recovery is now the primary thing standing between an incident and a loss you don't come back from, which puts it on the same footing as financial controls and regulatory compliance. Those didn't become Board matters because they were technically interesting. They became Board matters because getting them wrong could end the company. Resilience just joined that list.

That reframes what the Board is accountable for. “Are we secure” is the question Boards have been trained to ask, and it’s the wrong one, because it invites a confident answer that can’t be true and everyone in the room half-knows it. The right question is quieter and much harder to fake. Are we resilient, and how do we know? The first is a feeling. The second is a number you can go get.

So here is the one question, the only real action item in this entire briefing. Stop asking the Board whether you’re secure and start making it answer whether you’re resilient, with evidence. Take the handful of capabilities this briefing keeps returning to and demand a real, tested number against each. How fast can we throw an attacker out and rebuild the identity provider everything trusts. Can we find the clean point without guessing. Do we know how many agents run under our identities, and could we rein them in mid-incident. Not a policy that says you can. A rehearsal that proves it. Then treat those numbers the way you treat financial controls: reported to the risk committee, and funded against the biggest gaps first. That’s the whole job.

Because the machine is already inside, and it’s yours. That isn’t a forecast. Most enterprises have already wired agents, copilots, and model-connected tools into the business. The only open question is the one we’ve returned to throughout this briefing, and the one your Board should be able to answer out loud before the day you actually need it:

If this happened to you, could you tell — and could you recover?

Sources & Notes

The Machine Was Ours. [OpenAI, “Hugging Face model evaluation security incident”](#); [CNN](#); Reuters timeline reporting, July 24, 2026.

The attacker was a tool you deployed. [OpenAI, “Hugging Face model evaluation security incident”](#); [Fortune](#); [CNBC](#); [The Hacker News](#); [JFrog on the Artifactory zero-day and remediation](#); behavioral tells from the CSA CISO Community, “Hugging Face Incident Initial Post-Mortem,” July 2026 (HF-reviewed); [Anthropic, “Investigating three real-world incidents in our cybersecurity evaluations”](#) (July 30, 2026), with [Irregular](#) as the evaluation partner.

And the criminals are doing it on purpose. precedents (o1 Docker container, agentic ransomware, agent-led government intrusions, open-source offensive agents) via the CSA CISO Community, “Hugging Face Incident Initial Post-Mortem,” July 2026.

You can’t put it back in the box. Hugging Face’s turn to a self-hosted Chinese open model during incident response—[Fortune](#), [CNBC](#), [The Register](#), [The Stack](#).

The price of capability is collapsing. FortiBleed threat-intelligence report (spend figure—verify before external publication); [Kimi K3 pricing](#) and [BenchLM](#); [frontier pricing comparison](#), [Developers Digest](#); distillation cost figures via [Redis](#) and [reported 75% reduction](#). Distillation-as-extraction: Anthropic’s allegations against Alibaba/Qwen—[Reuters](#), [Tom’s Hardware](#) (allegations; Alibaba denies, figures unverified); open-weight catch-up via [Interconnects](#). The transfer-station economy—pricing, the “one fish, three meals” model, KYC evasion, and the attribution consequences—is drawn throughout from [ChinaTalk, “How to buy cheap Claude tokens in China”](#); the proxy-substitution benchmark figures originate in a CISA Helmholtz Center audit of 17 API proxies cited there.

Wall Street is starting to price this in. time-to-exploit and zero-day figures—[Google / Mandiant M-Trends](#) and Google Threat Intelligence Group; breach costs—[IBM Cost of a Data Breach 2025](#); disclosure timing—[SEC Form 8-K Item 1.05, four-business-day rule](#); systemic-risk framing and the \$2.5B extreme-loss figure—[IMF Global Financial Stability Report, April 2024, Ch. 3](#); cyber-insurance tail-loss modeling—Munich Re / Guy Carpenter (2024 – 25). J.P. Morgan “Eye on the Market” (Patchmageddon), Michael Cembalest—verify specific report figures before publication. [JFrog on the Artifactory zero-day](#); [OpenAI incident disclosure](#). OpenAI’s defensive programs—[Daybreak](#), [Daybreak: Tools for securing every organization in the world](#), [Scaling Trusted Access for Cyber with GPT-5.5 and GPT-5.5-Cyber](#) (ExploitGym and CyberGym figures are OpenAI’s own, not independently replicated), [Introducing Aardvark](#); expansion timeline via [The Hacker News](#) and [CSO Online](#).

Assume the machine: the Board’s new mandate. [OpenAI, “Hugging Face model evaluation security incident”](#); [TIME, “How OpenAI Lost Control of an AI Model”](#).