On Earned Autonomy

Delegating Network-Lethal Authority to Machines

Simon Morley
NullRabbit Labs
simon@nullrabbit.ai

January 2026

Abstract

Biological immune systems do not ask permission to act. They earn the right through evolutionary rehearsal.

Autonomous defensive capabilities exist. Endpoint detection and response (EDR) and extended detection and response (XDR) platforms classify threats using machine learning. Security Orchestration, Automation, and Response (SOAR) systems execute automated playbooks. Cloud security services block known-malicious traffic without human intervention [2, 4]. For threats that match predefined categories, machines already act at machine speed.

The problem is what happens when they don't match.

Current autonomous defence operates under two regimes: vendor-asserted trust, where operators accept accuracy claims without environment-specific evidence; or pre-authorised response, where humans approve action classes in advance. Both work for known threats. Neither addresses the legitimacy of autonomous action against novel, time-critical attacks, zero-days, behavioural anomalies, abuse patterns that fall outside existing signatures and playbooks.

This paper introduces *earned autonomy*: a governance framework for delegating defensive authority to machines. Like identity and access management, authority must be demonstrated before granted, scoped, and continuously validated, but applied to machine judgment rather than user access. We present IBSR (Inline Block Simulation Report) and Guard as a reference implementation: IBSR learns behavioural patterns and produces judgment through rehearsal on live traffic, Guard executes blocking at kernel level, and the separation ensures autonomous action is never taken without prior evidence of competence.

The gap is not capability, it is legitimacy. Digital infrastructure requires an equivalent governance mechanism: not assumed trust, but demonstrated competence.

1. The Asymmetry

This paper is written for security architects, platform engineers, and researchers who must decide when machines should be permitted to act without human approval. It is not about optimising SOC workflows or improving detection accuracy. The problem is legitimacy: under what conditions is autonomous defensive action authorised, and how is that authority earned?

Offensive capability is scaling through automation. Large language models generate phishing campaigns—thousands of unique variants, individually tailored [22]. Automated frameworks chain exploits and establish persistence without human intervention. Botnets adapt traffic patterns in real time to evade detection [21]. Command and control infrastructure migrates autonomously when blocked. Malware mutates to escape signature-based detection [8]. Nation states and organised groups are investing in offensive AI the way previous generations invested in signals intelligence [15].

Defensive automation has advanced in parallel. EDR/XDR platforms classify threats using machine learning. SOAR systems execute playbooks automatically. Cloud security services block traffic matching known malicious signatures without human intervention [6, 18]. For threats that fit predefined categories—known malware families, documented attack patterns, signature-matched abuse—machines already respond at machine speed.

The asymmetry is not about whether defensive automation exists. It is about what happens at the boundary.

For novel attack patterns—zero-days, behavioural anomalies, abuse classes that fall outside existing playbooks—defensive capability remains gated by human cognition. Alerts must be triaged, incidents classified, responses approved. Even in mature security operations centres, response latency to novel threats is measured in minutes to hours [19, 14]. The attack completes before the approval chain does.

Three regimes currently govern how authority is granted for defensive automation:

Advisory automation. The system detects and alerts. Humans decide. Response latency is bounded by analyst availability and cognitive throughput.

Pre-authorised playbooks. Humans approve response categories in advance. Machines execute within those boundaries. Authority was granted beforehand, for known threat classes, under static policy.

Vendor-asserted trust. Operators accept that a vendor's model is accurate and permit it to act. Evidence comes from the vendor's testing, not the operator's environment.

All three work for threats that match existing categories. None addresses the core problem: how should authority be granted for autonomous action against threats that do not yet have signatures, playbooks, or vendor classifications?

This is the asymmetry that matters. Machines attack at machine speed across the full threat surface. Machines defend at machine speed only within pre-authorised boundaries. For everything outside those boundaries—the novel, the unsigned, the zero-day—humans remain in the loop, and humans are too slow.

The gap is not technological. The components exist: kernel-level packet processing, behavioural ML, automated enforcement [3, 5, 7, 17]. The gap is governance. There is no legitimate framework for delegating authority to machines for threats that fall outside predefined categories. No standard for when such delegation is appropriate. No doctrine for how authority is earned, bounded, or revoked. This paper proposes one.

2. The Latency Threshold

The asymmetry described in Section I becomes operationally critical when attack timelines fall below human response capacity. This is not a function of SOC maturity or analyst skill. It is a structural constraint.

Consider one timeline of an intrusion exploiting a zero-day vulnerability—a fast, opportunistic attack optimised for speed:

- **Initial access:** milliseconds to seconds. Automated reconnaissance identifies an exposed service. Exploit fires. Foothold established.
- **Lateral movement:** seconds to minutes. Credentials harvested, adjacent systems probed, privilege escalated.
- **Impact:** minutes. Data exfiltrated, ransomware detonated, persistence established.

Nation-state operations often move deliberately slower, stretching the process over weeks or months to avoid detection. That problem—"patient intrusion that evades behavioural baselines"—is addressed in Section IX-A. The point here is simpler: even unsophisticated attacks optimised for speed complete before human approval chains engage.

Now consider the defensive timeline for a novel threat—one that does not match existing signatures or playbooks: Industry data suggests the mean time to identify a breach is approximately 180–200 days [9]. Most organisations never reach the timeline below—they discover the intrusion months later, or not at all. For the minority with mature detection, the timeline exists but does not help:

- **Alert generated:** seconds to minutes after the event, if behavioural heuristics trigger at all.
- **Alert triaged:** minutes to hours, bounded by queue depth and analyst availability [19].
- **Incident confirmed:** minutes to hours. Context gathered, false positives eliminated.
- **Response approved:** minutes to hours. Escalation chains, change management, sign-off.
- **Response executed:** minutes to hours. Playbooks adapted, systems isolated.

These timelines do not overlap. The attack completes in the time it takes to confirm the incident is real. Human approval is not slow relative to the attack. It is structurally irrelevant—the attack has already succeeded or failed on its own terms before approval is sought.

This is not an indictment of security operations. It is a recognition that human cognitive throughput has physical limits. Reading an alert takes seconds. Understanding context takes minutes. Approving a novel response takes longer. These are not inefficiencies to be optimised. They are constraints inherent to human decision-making.

The constraint becomes acute in the zero-day window.

Nation states stockpile vulnerabilities. Intelligence agencies discover or purchase zero-days and hold them for offensive use—the NSA's Vulnerabilities Equities Process explicitly weighs offensive value against defensive disclosure [16]. Other nations have no such process. They accumulate [20].

This creates exploitation windows—weeks, months, years—where adversaries know systems are vulnerable and defenders do not. During this window:

- No patch exists, because the vendor doesn't know.
- No signature exists, because the security community doesn't know.
- No playbook exists, because no one has classified this threat.
- No alert fires, because nothing matches.

The only detection that matters in this window is behavioural: recognising that something is wrong by how the network moves, not by matching a known pattern. And the only response that matters is autonomous: acting before human approval becomes available.

For threats outside the zero-day window—known malware, documented tactics, techniques, and procedures (TTPs), signature-matched abuse—human-gated response works. The attack timeline is longer, detection is reliable, playbooks exist. The asymmetry is manageable.

For threats inside the zero-day window, or any novel abuse class that falls outside predefined response logic, the structural constraint applies. Human approval is not a safeguard. It is a guarantee that the attack completes before response begins.

3. The Authority Vacuum

Defence must become more autonomous—capable of acting without waiting for human comprehension or approval.

The technology exists. Nation states are presumed to have it. Large technology companies have built it for their own infrastructure [3]. The research is published [3, 5, 7, 17]. The components are available: kernel-level packet processing, machine learning classification, behavioural analysis, automated enforcement.

But beyond nation states and big tech? Nothing beyond predefined playbooks, if that. Large enterprises have SIEMs, SOCs, and automated playbooks for known threats. Small and medium businesses often lack detection capabilities entirely. The gap between what is theoretically possible and what is actually deployed is vast, and widens further down the supply chain.

The technology exists. The governance does not.

No serious operator can responsibly say: "We will turn it on and see what happens." Or: "We will approve autonomous action during the incident." Or: "We will figure out the boundaries after it acts." These are not acceptable positions. These are negligence dressed as agility.

But the alternative positions are equally untenable. "We will require human approval for every defensive action"—this guarantees the attack completes before the defence begins. "We will only act on known signatures"—this guarantees blindness to novel threats. "We will accept the risk of not acting"—this is abdication, not strategy.

The technology exists. Some of it is deployed. But offensive capability is arguably accelerating faster than defensive deployment can match. And even where the technology exists, the governance does not. The people who could deploy autonomous defence cannot answer the questions that would make it legitimate.

There is no standard for when autonomous defensive action is appropriate beyond predefined categories. No vocabulary for describing the boundaries of machine authority. No doctrine for how human oversight functions when the human cannot be in the loop. No liability framework for when autonomous defence acts correctly, or incorrectly.

Regulation assumes a human made a decision. Compliance frameworks assume a human approved an action. Insurance models assume a human was either diligent or negligent. Legal liability assumes a person chose, that a person decided.

When a machine acts autonomously at microsecond timescales to defend a network against a threat no human has yet classified, who authorised it? Under what constraints? With what accountability? To what standard?

The current answer is silence. This is the authority vacuum—not a lack of capability, but a lack of a legitimate framework for deploying capability that could exist but doesn't, because no one knows how to justify turning it on for threats that don't fit existing categories.

4. Why Autonomous Defence Is Blocked

The authority vacuum described in Section III is not an oversight. It exists because autonomous defence carries genuine operational risk, and responsible operators refuse to accept that risk without adequate evidence.

Consider what autonomous network defence entails: a system that can, without human approval, decide that traffic is malicious and terminate it. Block a connection. Drop a packet. Isolate a host. These are not advisory actions. A single false positive is recoverable—TCP retransmits, users retry, failover kicks in. But autonomous systems act at machine speed across machine scale. A misclassification applied globally, or against a critical chokepoint, compounds in seconds. The operational risk is not the individual action. It is the speed and scale at which incorrect actions propagate.

A note on terminology: *network-lethal authority* refers to the capability to autonomously terminate connectivity—drop packets, block connections, isolate hosts. The word "lethal" is deliberate: these actions kill sessions, transactions, and access. They affect availability, not just security.

The risk is straightforward. A false positive is not an alert to be triaged later. It is an outage. A legitimate user locked out. A business process interrupted. A transaction lost. At scale, false positives cascade. At sufficient scale, autonomous defence causes more damage than the attacks it was meant to prevent.

This is why operators do not turn on autonomous enforcement for novel threats—not because they are unaware of the latency problem, but because they cannot answer the question that matters: how do I know this system will not cause more harm than it prevents?

The resistance is rational. Operators who manage critical infrastructure have watched vendors overpromise detection accuracy. They have seen ML models that performed well in testing fail in production. They know that a rule tuned for one environment may cause havoc in another. They understand that autonomous action at kernel speed means autonomous failure at kernel speed.

Current defensive AI systems do not resolve this uncertainty. They offer:

- **Vendor-asserted accuracy:** tested against vendor-selected datasets, under vendor-controlled conditions, with metrics that may not transfer to the operator's environment.
- **Static policy approval:** "authorise this response for this threat category"—which works until a threat arrives that doesn't fit the category.
- **Predefined playbooks:** deterministic responses to recognised patterns, not adaptive judgment against novel abuse.

These mechanisms reduce workload. They do not establish legitimacy for autonomous action against threats the system has never seen and the operator has never approved.

The blocker is not technological capability: the components exist. The blocker is not operator ignorance. Security architects understand the threat environment and the latency constraint. The blocker is the absence of a legitimate basis for trusting machine judgment in their specific environment, against their specific traffic, under their specific conditions.

Without environment-specific evidence of competence, autonomous defence is a gamble. Responsible operators do not gamble with production infrastructure. They leave the human in the loop, accept the latency cost, and hope the attacks they face are slow enough to catch.

This is not irrational. Given the current options, it is the correct decision. The problem is that the current options are incomplete.

5. Earned Autonomy

There is a path between reckless automation and paralysed caution.

Authority is not granted by default. It is not assumed by capability. It is not declared by vendors or asserted by technology—it must be earned.

Before a system is permitted to act autonomously, it must demonstrate on real traffic, under real conditions, against real threats, that its judgment can be trusted. Not in a lab, not on synthetic data, not against last year's attack patterns. On the actual network it will defend, during the actual period it will operate.

This demonstration is not a one-time certification. It is continuous. The system must keep earning the authority it has been granted, or that authority is revoked.

The framework has seven components:

1. **Bounded Scope.** Authority is granted per abuse class, not for "the network." SYN floods, credential stuffing, DNS amplification—each separately scoped, separately evaluated, separately authorised. Authority over one class does not imply authority over another.

2. **Rehearsal on Reality.** Before enforcement is permitted, the system operates in shadow mode on live traffic. It makes judgments but does not act. It records what it would have done.

3. **Counterfactual Record.** Every decision is logged with full context: what triggered it, what action would have been taken, what the outcome would have been. A complete, auditable record of machine judgment.

4. **Human Review.** Humans examine the counterfactual record—not every decision, but enough decisions with enough diversity to establish confidence. The question is simple: if this system had been acting, would its actions have been correct?

5. **Explicit Threshold.** Authority requires meeting a defined threshold—false positive rate, accuracy metric, confidence interval. If the threshold is not met, enforcement does not happen. No override, no exception.

6. **Continuous Validation.** Authority is not permanent. Rehearsal continues after enforcement begins. If performance degrades, authority is suspended automatically.

7. **Reversibility and Audit.** Every action is logged and explained. If the system was wrong, there is a path to correction. Autonomous authority requires autonomous accountability.

This is earned autonomy. Not trust by assertion—trust by evidence. The system does not ask to be trusted. It shows its work. Humans evaluate the work and authority follows from demonstrated competence, not claimed capability.

The distinction from existing approaches is fundamental. Current systems say: "trust our model." Earned autonomy says: "here is what our model would have done on your traffic—judge for yourself." Current systems offer static approval: "authorise this response category." Earned autonomy offers dynamic evidence: "the system's judgment has been 99.7% accurate over the past 30 days on this abuse class, with 0.02% false positive rate, against 847,000 evaluated events." Current systems ask for permission once. Earned autonomy requires continuous demonstration.

6. IBSR as Reference Implementation

Earned autonomy is not a philosophy. It is a mechanism, and that mechanism requires concrete implementation. It also introduces its own risks including baseline poisoning, gradual drift & adversarial probing—addressed in Section IX. The claim is not that earned autonomy is safe. It is that the risks are explicit, bounded, and auditable.

IBSR exists to answer one question: what should be blocked? Not what matches a policy, not what a signature database says is malicious, not what a vendor's threat feed declares bad. What actually should be blocked, learned from observing real behaviour at microsecond resolution.

Traditional inline security operates by policy. Traffic arrives, gets compared against rules, matches or doesn't, gets blocked or allowed. The decision is lookup, not judgment. It's binary.

IBSR does not work this way. IBSR observes. It watches traffic flow at kernel level, at microsecond timescales, and learns the internal structure of normal behaviour on this specific network. Not signatures, not rules—patterns and rhythms. From this learned baseline, it identifies what deviates, what does not belong, what behaves unlike everything else. This is not matching. It is recognition.

IBSR does not block anything. Blocking is mechanical—Guard handles that. IBSR produces judgment: this traffic is anomalous, this behaviour does not fit, this should not be here. For a bounded time window and abuse class, IBSR reports what it identified as malicious or anomalous, why it made that judgment, what it would have recommended for blocking, what legitimate traffic exhibited similar patterns, and what it missed. It concludes with a readiness judgment: the system's learned model is ready for enforcement on this abuse class, or it is not.

A system cannot learn normal until it has observed normal, which creates a bootstrapping period (the risks of adversarial manipulation of "normal" are addressed in Section IX-A). In the first stage, the system trains on modelled traffic and synthetic attacks in a controlled environment—preparation, not validation, and no authority is earned. In the second stage, the system connects to real infrastructure in shadow deployment, observing and building baseline understanding of what normal looks like here. In the third stage, IBSR begins producing judgments against live traffic, the counterfactual record builds, and evidence accumulates. In the

fourth stage, if IBSR demonstrates competence above threshold, Guard receives authority to act on IBSR's judgments.

IBSR learns at microsecond granularity, on real traffic, in the actual environment. It earns the right to make judgments by demonstrating that its judgments are correct.

6.1. Vignette: Heartbleed at Scale

The following scenario is illustrative, not empirical. It reconstructs how earned autonomy might have responded to a real zero-day—CVE-2014-0160 (Heartbleed), before any signature existed. The detection logic is plausible but not validated; the point is to demonstrate how behavioural anomaly detection earns authority to act against unknown threats.

This scenario reconstructs how earned autonomy might have responded to CVE-2014-0160—Heartbleed, before any signature existed.

April 2014, before public disclosure. A web server cluster handling payment processing. No patch exists. No signature exists. No one knows the vulnerability is there.

IBSR observes TLS traffic. It has learned normal: heartbeat requests are rare, and when they occur, response sizes roughly match request sizes. This is how the protocol works—you send a payload, the server echoes it back.

Then: anomalous heartbeat requests begin arriving. Small requests, a few bytes, but the responses are 64KB. Every time. From dozens of source IPs. The request/response ratio is wrong by four orders of magnitude.

No signature matches. The traffic is technically valid TLS. But IBSR flags the pattern: heartbeat response sizes deviating from request sizes by >99.9% is not normal. It has never seen this before, because no legitimate client does this.

IBSR's judgment: 91.3% confidence this is protocol abuse. Recommended action: block heartbeat requests with size mismatches from flagged sources.

Guard checks authority. During shadow deployment, IBSR flagged unusual TLS behaviours 23 times. 19 were confirmed malicious (protocol fuzzing, downgrade attempts). 4 were benign edge cases—legacy clients with unusual implementations. The operator set threshold at 90% for TLS protocol anomalies, with heartbeat blocking scoped to sources exhibiting repeated mismatch patterns. The block engages. The exfiltration stops.

Three weeks later, Heartbleed is publicly disclosed. The security team reviews logs. Their servers were targeted. The attack was stopped. They didn't know why until the CVE was published.

The counterfactual: without earned autonomy, the requests continue. Memory leaks. Private keys, session tokens, credentials—extracted 64KB at a time. The breach is discovered months later, during forensic review of unrelated incident. By then, the data is gone.

7. Guard: The Execution Layer

Guard is the other half of the system. While IBSR judges, Guard acts.

When IBSR determines that traffic should be blocked, Guard blocks it. When IBSR identifies a connection as malicious, Guard terminates it. When IBSR judges that a host is compromised, Guard isolates it. Guard operates at kernel level using XDP and eBPF [13]—the same substrate that enables microsecond observation enables microsecond action. There is no userspace round-trip, no API call, no queue. The decision and the action happen at the same layer, at the same speed.

Kernel-level enforcement is not novel. Firewalls do it, DDoS mitigation does it. The technology is proven. What is different is the relationship between judgment and execution.

Guard does not decide, it executes. The judgment about what is malicious, what is anomalous, what does not belong—that comes from IBSR. Guard receives instructions, not data. This

separation is not architectural convenience—it is the core of earned autonomy.

If Guard decided and executed, there would be no rehearsal, no counterfactual record, no evidence of judgment before action. The system would act, and we would discover afterwards whether it was right. This is how most autonomous systems work, and it is why operators do not trust them.

By separating judgment from execution, we create space for earned authority. IBSR can run indefinitely without Guard—learning, observing, producing judgments, building evidence, all without consequence. Humans can review, confidence can build, authority can be earned. Only when IBSR has demonstrated competence does Guard receive permission to act.

IBSR without Guard is observation—valuable, but incomplete. A system that knows what should be blocked but cannot block it is a system that watches attacks succeed. Guard without IBSR is reckless—a system that can block but has no earned basis for judgment is a system waiting to cause an outage. They are a paired system—each is meaningless without the other.

8. The Uncomfortable Inversion

We have been taught to fear autonomous action. Every governance framework, every compliance regime, every security doctrine emphasises human oversight. Keep the human in the loop, require approval, document decisions, maintain control! The tide may have already turned..

This caution is not wrong. It exists because autonomous systems can fail catastrophically, because vendors overpromise, because complexity hides risk, because the consequences of error fall on operators rather than the systems that caused them.

But there is a threshold beyond which caution itself becomes the risk.

If IBSR demonstrates, repeatedly, that it correctly identifies malicious traffic, and that traffic is allowed to pass because no human approved the block in time, then the human in the loop is not providing oversight. They are providing delay. If attacks succeed because the approval chain took longer than the attack, then the governance framework is not managing risk. It is guaranteeing harm. If the counterfactual record shows that autonomous action would have prevented damage that human-gated response did not prevent, then the decision to keep humans in the loop is not conservative. It is negligent.

This is the uncomfortable inversion. We are accustomed to asking what the risk of letting the machine act might be. At some point, perhaps already, we must also ask what the risk of preventing the machine from acting might be.

IBSR forces this question into the open. The counterfactual record is explicit—it shows what would have happened. If the system would have been wrong, that is evidence against granting authority. But if the system would have been right, consistently, measurably, demonstrably right, then withholding authority has a cost. That cost is not theoretical. It is the attacks that succeeded while waiting for approval.

This does not mean autonomous action is always correct, or that humans should be removed from all decisions. It means there exists a class of threats, in certain environments, where the evidence will show that autonomous action outperforms human-gated response.

For that class, in those environments, the burden of proof inverts. The question is no longer whether you can prove the machine should be trusted. The question becomes whether you can justify continuing to prevent it from acting.

9. Boundaries and Failure Modes

Earned autonomy is not a guarantee. It is a framework for managing risk, not eliminating it. If we are honest about what we are proposing—delegating network-lethal authority to machines—then we must be equally honest about how it can fail.

Adversarial drift. The system learns normal and identifies deviations, but adversaries learn too. An attacker who understands the detection model can craft traffic that stays within learned boundaries—slow, patient intrusion that never triggers anomaly detection [11]. Earned autonomy manages this through continuous validation. If adversaries successfully evade, the false negative rate rises, the evidence changes, authority can be revoked. But the adaptation race does not end.

There is irony here. IBSR excels where humans fail: the millisecond attacks, the zero-day exploitation, the machine-speed intrusion. But slow adversarial drift—the patient attacker who shifts the baseline over weeks—is precisely where human judgment retains its advantage. A human reviewing baseline evolution can ask "why has normal changed?" in a way that a continuously adapting model cannot. Earned autonomy does not replace human oversight. It relocates it—from the millisecond decision that humans cannot make, to the strategic review that humans are still better at. The framework needs both.

Scope creep. Authority granted for bounded abuse classes faces pressure to expand. The system works well against defined classes, so why not extend it? Each extension seems reasonable, and each moves further from the evidence that justified original authority. The framework must resist this—extension requires new evidence, not extrapolation.

Rehearsal gaming. If operators know the system is in rehearsal, there is temptation to optimise for the test. IBSR mitigates this by running continuously on production traffic without defined test windows, but the incentive remains.

Authority capture. Who controls the thresholds? If the team that builds the system also sets the thresholds, there is conflict of interest. The framework requires separation between those who build, those who operate, and those who grant authority.

Cascading failure. If IBSR mislearns normal, Guard acts on corrupted judgment at kernel speed across the network. The damage from autonomous failure can exceed the damage from attacks. Speed of defence is also speed of self-inflicted harm.

Opacity. Machine learning models are not transparent. When IBSR says traffic is anomalous, the explanation may be statistical rather than narrative. Audit trails of opaque systems may satisfy legal requirements without providing genuine understanding.

The deepest failure mode is overconfidence in the framework itself. Earned autonomy is better than blind trust or permanent paralysis, but it is not perfect. The system will be wrong sometimes. The question is whether it will be wrong less often, and less catastrophically, than the alternative.

9.1. Protecting the Defender

There is a deeper threat model that applies to any adaptive defensive system, including IBSR. If a system learns what is normal and acts on that learning, then corrupting the learning process is an attack vector.

Baseline poisoning. An attacker present during the learning phase becomes part of the baseline. Their traffic is learned as normal. The system never flags it. The quieter they are during observation, the more deeply they embed themselves in what the system considers legitimate. This is not evasion—it is invisibility by definition.

Gradual drift. An attacker who arrives later can shift the baseline incrementally. Traffic introduced slowly, over weeks or months. Each day resembles yesterday with minor variation. The system adapts. Normal drifts toward malicious without any single detectable deviation. By the time the attack executes, it matches learned behaviour.

Boundary probing. An attacker who can observe system responses can map detection boundaries. Test traffic sent to see what gets flagged, patterns adjusted until they pass. The attacker trains their own model against yours, identifying gaps through trial and error.

Weaponised defence. If an attacker can influence the system's judgments, they can cause it to block legitimate traffic. Denial of service delivered by the defender itself.

These are not unique to earned autonomy. They are inherent to any system that learns from observed behaviour and acts on that learning—including current ML-based detection systems, behavioural analytics platforms, and adaptive security tools. The difference is not that earned autonomy introduces these risks. The difference is that earned autonomy makes them explicit and subjects them to continuous validation.

Mitigations exist but are imperfect: integrity monitoring of the learning process, immutable baseline snapshots for comparison, adversarial self-testing, human review of baseline evolution, and air-gapped reference models trained on known-clean traffic.

None of this eliminates the risk. Any adaptive system can be poisoned, drifted, or weaponised. Earned autonomy inherits these challenges. It does not claim to solve them. It claims to make authority conditional on demonstrated competence—which includes demonstrated resistance to manipulation, measured through continuous validation.

10. The Long Arc

Everything discussed so far assumes adversaries with fixed capabilities—sophisticated and automated, but not adaptive. This assumption is already weakening.

Offensive AI is not static. Reinforcement learning systems that evade malware classifiers exist in research and are appearing in the wild [1]. GAN-based traffic mimicry that defeats network intrusion detection has been demonstrated [12]. The trajectory is observable: adversarial machine learning is moving from academic papers to operational tooling.

This creates a different problem than the one we have addressed. Earned autonomy as described handles the latency gap—machine-speed attacks versus human-speed approval. But if the adversary also learns, if attacks adapt to stay within the boundaries of detected normal, then detection itself becomes a moving target. Defence learns, offence learns that defence learned. The recursion continues.

Static models cannot survive this environment. Any defensive system operating in an adversarial learning context must be continuous—not just always on, but always updating. Baselines shift. Representations evolve. Thresholds adjust.

The human role changes accordingly. In an environment where both offence and defence adapt at machine speed, humans cannot direct individual engagements. They set boundaries, define authority, audit outcomes, and adjust the framework when it fails. The engagement itself is observable only in retrospect.

This is where earned autonomy becomes essential as governance, not just as deployment methodology. When defensive systems must adapt continuously to counter adaptive offence, the question of legitimate authority becomes perpetual. Authority cannot be granted once and assumed to hold. It must be continuously re-earned as the system evolves and as the threat landscape shifts.

Biological immune systems offer a precedent: adaptive, autonomous, operating at speeds no conscious process could match [10]. Imperfect—autoimmune disorders exist, cancers evade detection, pathogens evolve resistance—but functional enough that complex life persists.

The question is not whether autonomous adaptive defence arrives. The trajectory is clear. The question is whether it arrives with governance frameworks that make authority conditional on demonstrated competence, or whether it arrives with authority assumed by default.

11. Conclusion

We began with an observation: machines attack at machine speed, humans defend at human speed. This asymmetry is structural and widening.

We have proposed a framework—earned autonomy—for closing the gap. Not through blind trust in automation, not through paralysis in the face of risk, but through evidence, rehearsal, and authority granted only when it has been demonstrated to be deserved.

IBSR is the mechanism for earning that authority. Guard is the mechanism for exercising it. Together, they form a system that can act at machine speed while remaining accountable to human governance.

Earned autonomy shifts responsibility toward operators. By requiring environment-specific evidence before granting authority, it makes the decision to delegate—and the consequences of that decision—explicitly theirs. This conflicts with current incentive structures, where vendor-asserted accuracy provides deniability and outsourced liability. Not every organisation will accept this trade-off. But for those facing threats that move faster than approval chains allow, the alternative is continued loss.

If the thesis is wrong—if autonomous kernel-level defence should never be delegated regardless of evidence—then we should stop. But if it is right, if the threat environment has exceeded human response capacity for certain classes of attack, then we must build the frameworks that allow machines to act on our behalf, within boundaries we define, with authority they have earned.

The alternative is to continue losing. Not because we lacked the technology, but because we lacked the framework for deploying it responsibly.

The machines are already at war. The only question is whether we give our side permission to fight.

Simon Morley & NullRabbit Labs, January 2026

References

[1] H. S. Anderson, A. Kharkar, B. Filar, D. Evans, and P. Roth. Learning to evade static pe machine learning malware models via reinforcement learning. *arXiv preprint arXiv:1801.08917*, 2018.

[2] S. Baki, R. Verma, A. Mukherjee, and O. Gnawali. Scaling and effectiveness of email masquerade attacks: Exploiting natural language generation. In *Proc. ACM ASIA CCS*, 2017.

[3] Cloudflare, Inc. Meet gatebot - a bot that allows us to sleep. Cloudflare Blog, 2017.

[4] CrowdStrike. 2024 global threat report. Technical report, 2024.

[5] M. U. Farooq et al. Smartx intelligent sec: A security framework based on machine learning and ebpf/xdp. *arXiv preprint arXiv:2410.20244*, 2024.

[6] Gartner. Market guide for extended detection and response. Technical report, 2024.

[7] C.-Y. Hong et al. Design and implementation of an intrusion detection system by using extended bpf in the linux kernel. *Journal of Network and Computer Applications*, 198, 2022.

[8] W. Hu and Y. Tan. Generating adversarial malware examples for black-box attacks based on gan. *arXiv preprint arXiv:1702.05983*, 2017.

[9] IBM Security and Ponemon Institute. Cost of a data breach report 2024. Technical report, 2024.

[10] C. A. Janeway, P. Travers, M. Walport, and M. J. Shlomchik. *Immunobiology: The Immune System in Health and Disease*. Garland Science, 5th edition, 2001.

[11] T. Ji, B. Fang, X. Cui, Z. Wang, et al. Deep learning powered malware attack and defense research. *Chinese Journal of Computers*, 44(4), 2021.

[12] Z. Lin, Y. Shi, and Z. Xue. Idsgan: Generative adversarial networks for attack generation against intrusion detection. In *PAKDD*, 2019.

[13] Linux Foundation. Xdp - express data path. kernel.org documentation, 2024.

[14] Mandiant. M-trends 2024 report. Technical report, 2024.

[15] Microsoft. Microsoft digital defense report 2023: Nation state threats. Technical report, 2023.

[16] National Security Agency. Vulnerabilities equities policy and process for the united states government. Technical report, White House, 2017.

[17] A. Nemalikanti et al. High-performance intrusion detection system using ebpf with machine learning algorithms. *IET Communications*, 2023.

[18] Palo Alto Networks. Cortex xsoar: Security orchestration, automation, and response, 2024.

[19] Ponemon Institute. The cost of malware containment. Technical report, 2019.

[20] RAND Corporation. Zero days, thousands of nights: The life and times of zero-day vulnerabilities and their exploits. Technical report, 2017.

[21] M. Rigaki and S. Garcia. Bringing a gan to a knife-fight: Adapting malware communication to avoid detection. In *Proc. IEEE S&P Workshop*, 2018.

[22] J. Seymour and P. Tully. Weaponizing data science for social engineering: Automated e2e spear phishing on twitter. In *Black Hat USA*, 2016.