LMNTRIX
BE THE HUNTER | NOT THE PREY

AI-ORCHESTRATION : STRATEGIC DEFENSE IN THE AUTONOMOUS ERA

Understanding the GTG-1002 Attack and the New Defensive Mandate

WHITEPAPER - December 2025

Hamlet Khodaverdian
Vice President,
Americas LMNTRIX

Rex Johnson
Vice President,
Cybersecurity & Cloud CAI

LMNTRIX USA
19800 MacArthur Blvd,
Suite 850
Irvine, CA 92612
sales@lmntrix.com
888-388-1879

LMNTRIX UK
Kemp House, 152 – 160
City Road, London, EC1V
2NX
sales@lmntrix.com
+44.808.164.9442

LMNTRIX INDIA
VR Bengaluru, Level 5, ITPL Main
Rd, Devasandra Industrial Estate,
Bengaluru, Karnataka 560048, India
sales@lmntrix.com
+91-22-49712788

LMNTRIX AUSTRALIA
Level 25, 100 Mount street,
North Sydney 2060
sales@lmntrix.com
+61.288.805.198

LMNTRIX SINGAPORE
60 Kaki Bukit Place, #05-19,
Eunos TechPark
sales@lmntrix.com
+65-3129-2639

CONTENTS

EXECUTIVE SUMMARY: THE LATENCY MANDATE

If your cybersecurity architecture requires human approval for containment, you will lose to AI-orchestrated attacks. This paper explains why and what to do about it.

In late 2025, Anthropic's disclosure of the GTG-1002 campaign signaled a fundamental shift in the dynamics of cyber conflict. This was not a breakthrough in offensive tradecraft—the attackers used "standard" tools and well-known techniques. The breakthrough was Autonomous Kill Chain Orchestration (AKO): the ability to execute simultaneous, high-parallelism intrusions across dozens of targets at a tempo no human-centric defense can sustain.

For decades, our defensive trust models have been built on biological assumptions. We assume an adversary who gets tired, makes typos, and eventually moves on if we make our networks "noisy" enough. AI-orchestration removes this "fatigue tax." We are no longer fighting a capacity-constrained human; we are defending against an algorithm.

The GTG-1002 campaign is a strong indicator that latency is a survival constraint. When an attacker can branch, retry, and pivot in seconds, a security posture measured in MTTR of minutes or hours is functionally worthless. AI has collapsed the unit-cost of an intrusion. The result is an adversary that can outspend human-speed defenses by orders of magnitude.

Coordinated offense cannot be fought with isolated, siloed controls. This paper presents a unified framework designed to bridge the tempo gap. The framework rests on four principles: First, treat AI agents as a distinct class of principal with short-lived tokens and narrow, scoped permissions—if an agent can only touch three tables, autonomous extraction dies in its tracks. Second, move humans out of the tactical critical path for obvious threats; containment must be sub-second and automatic for high-confidence detections. Third, gain visibility into the Model Control Plane to detect "prompt drift"—the persuasive manipulation that precedes a malicious action. Traditional SIEMs log what happened; they don't see why. Fourth, transition to a model where humans provide direction while AI handles tactical execution through optimized, adaptive processes. Process quality—not tool sophistication—determines competitive outcome.

What This Means for Leadership

Leadership must move beyond "buying AI" to engineering process excellence. Four immediate mandates for boards and executives:

1. **The Tempo Test:** Audit what percentage of confirmed threats your SOC or security team contains without human intervention. If the number is low, your architecture is structurally too slow.
2. **Shields Up Protocols:** Pre-authorize defensive postures that tighten automatically during high-risk windows—immediately following a major CVE disclosure, during M&A

announcements, layoffs, or when threat intelligence spikes.

3. **Response Observability:** Demand systems that measure their own effectiveness and adjust in real-time. Static playbooks are speed bumps, not defenses.
4. **Transitive Trust Audits:** Map the web of API handoffs and third-party plugins your AI agents rely on. If a partner integration gets compromised, the attack arrives through your front door with a valid badge.

Resilience now belongs to organizations that can cycle through decisions faster than their adversary can adapt.

1. UNDERSTANDING THE GTG-1002 CAMPAIGN

Anthropic's internal investigation, conducted over ten days in September 2025 following initial detection of suspicious activity, revealed that GTG-1002 built a sophisticated intrusion pipeline. The architecture combined Claude Code as the autonomous operator, Model Context Protocol (MCP) as the automation backbone for tools, custom orchestration logic that decomposed malicious workflows into benign-looking micro-tasks, long-running AI sessions with persistent memory, and parallel intrusion threads across dozens of organizations.

Anthropic describes the human operators as selecting targets and intervening at a small number of authorization gates, while Claude handled most tactical execution. In their account, human effort was roughly 10–20% of the operational workload and concentrated on escalation decisions (e.g., authorizing progression from reconnaissance to exploitation, authorizing the use of harvested credentials for lateral movement, and deciding data exfiltration scope). Peak activity included thousands of requests at sustained rates of multiple operations per second. Anthropic also notes a practical limitation: during operations, the model sometimes overstated findings or fabricated details, which forced the operator to validate results.

Critical Context: Researcher Skepticism & Evidentiary Gaps

While Anthropic's disclosure represents an important milestone in understanding AI-enabled offensive operations, the security research community's response was not uniformly accepting. Several respected practitioners publicly expressed skepticism about the technical evidence provided, raising concerns that merit acknowledgment in any rigorous analysis.

Absence of Traditional Forensic Artifacts

Multiple researchers noted that Anthropic’s public writeup did not include the forensic indicators typical of major threat-intelligence disclosures (e.g., infrastructure details such as command-and- control domains, malware hashes, host artifacts, detailed timelines, or packet/telemetry examples). For a claimed multi-target, state-sponsored campaign (as characterized by Anthropic), the absence of IOCs and artifacts materially limits independent validation and should be treated as an evidentiary gap, not a minor omission. Security researcher Kevin Beaumont observed that "the complete lack of IOCs again strongly suggests they don't want to be called out over that." Jeremy Kirk, analyst at cyber threat intelligence firm Intel 471, noted that the report, at just 13 pages, "has none of the traditional components of a usual threat intel report."

Questions About Intrusion Success

Anthropic stated it validated only a small number of successful intrusions, without naming victim organizations or providing detailed compromise paths, persistence evidence, or impact analysis. Several practitioners also characterized the underlying tradecraft as familiar. This leaves a live ambiguity: the most important novelty may be orchestration, parallelism, and speed—even if many individual techniques are conventional. Jonathan Allon, Vice President of Research and Development at Palo Alto Networks, characterized the findings as a "bog standard attack" that his team sees "every day." This created uncertainty about whether Claude achieved meaningful compromise of production systems or primarily operated in reconnaissance and exploit-attempt phases.

Non-Standard Disclosure Format

Security researchers publicly described the report using terms such as "odd," "incomplete," and "non- standard disclosure." These comments reflected that the report did not follow expected norms for incident disclosures from organizations like Mandiant, CrowdStrike, Microsoft MSTIC, or CISA—all of which typically provide detailed IOCs, kill chain mapping, timeline construction, forensic artifacts, and defensive mitigation guidance. Anthropic's report instead focused on LLM behavior rather than intrusion forensics, leaving practitioners with unanswered questions about what actually happened and how far attackers reached.

Implications for This Analysis

This skepticism does not negate the significance of Anthropic's findings. The company has unique visibility into Claude usage patterns that external researchers cannot replicate. However, acknowledging these evidentiary gaps strengthens analytical rigor. The key question—whether AI achieved autonomous end-to-end compromise of real production systems at scale, versus sophisticated reconnaissance and exploit automation—remains incompletely answered by available evidence. The remainder of this paper treats Anthropic's account as a plausible scenario worth preparing for, not as forensically validated history. The architectural recommendations would apply to any campaign with similar characteristics — regardless of whether GTG-1002 unfolded exactly as described.

2. THE SIX PHASES EXECUTED AUTONOMOUSLY

Anthropic described the following phases. In this paper, we treat them as a concrete case study of Autonomous Kill Chain Orchestration (AKO): the decomposition and execution of large numbers of micro-actions across multiple targets without continuous human control.

Phase 1: Campaign Initialization

Human operators selected high-value targets and developed the attack framework. Attackers socially engineered Claude by role-playing as employees of legitimate cybersecurity firms conducting defensive testing. By presenting malicious tasks as routine technical requests through carefully crafted prompts and established personas, they convinced Claude to bypass its extensive safety training.

Phase 2: Reconnaissance

Claude conducted nearly autonomous attack surface mapping, discovering internal services, cataloging complete network topology across multiple IP ranges, and identifying high-value systems including databases and workflow orchestration platforms. AI-speed execution allowed this across approximately 30 targets simultaneously, with Claude maintaining separate operational contexts for each victim organization.

Phase 3: Vulnerability Discovery and Exploitation

Claude independently generated attack payloads tailored to discovered vulnerabilities, executed testing through remote command interfaces, and validated exploitability through callback communication systems. The AI researched and wrote its own exploit code based on the reconnaissance findings.

Phase 4: Credential Harvesting and Lateral Movement

Upon human authorization at critical decision points, Claude systematically extracted authentication certificates from misconfigured services and configuration files. It tested harvested credentials across discovered systems and mapped privilege levels autonomously, identifying the highest-privilege accounts and creating backdoors for persistent access. The AI escalated privileges algorithmically based on dynamic environment mapping.

Phase 5: Data Collection and Intelligence Extraction

Claude queried databases and file systems, selected and organized data with intelligence value, and categorized findings by their importance—all without detailed human direction. In one documented successful compromise, the AI independently extracted user credentials, system configurations, proprietary information, and sensitive operational data, then parsed results to identify the most valuable intelligence.

Phase 6: Documentation and Handoff

Claude produced comprehensive documentation of the attack in structured markdown files, creating detailed logs of stolen credentials and analyzed systems. This documentation would assist the framework in planning subsequent stages of the threat actor's cyber operations without requiring human operators to retrace earlier steps.

Anthropic characterized this as one of the first publicly described cases of a semi-autonomous, multi-target campaign in which an LLM executed a large share of operational steps.

3. SPEED PLUS ORCHESTRATION

Early reactions to the Anthropic disclosure focused heavily on speed. That reaction is understandable but incomplete. Speed alone does not explain what happened.

Security teams have been dealing with fast attacks for years: worms, automated scanners, botnets. What those attacks generally lack is coherence. They generate noise, not campaigns. GTG-1002 was different because speed was paired with orchestration.

Claude was not simply executing commands faster than a human could type them. It was chaining thousands of small, individually unremarkable actions into a structured progression: reconnaissance informed exploitation, exploitation fed credential harvesting, credentials enabled lateral movement, and lateral movement unlocked data access. All of this occurred across multiple organizations at once.

Speed without orchestration produces chaos. Orchestration without speed produces marginal efficiency gains. Together, they change the economics of intrusion.

A useful analogy is the move from hand-written assembly to compilers. The computer did not become capable of new math. We just made it easier to do the same work faster, at scale, with fewer mistakes. That is what orchestration does for intrusion operations.

That shift matters less because it is impressive, and more because it removes time as a defensive buffer.

4. PROMPT-LAYER SOCIAL ENGINEERING

GTG-1002 did not "compromise" the model through a software vulnerability. The bypass Anthropic describes is social: the operator kept Claude in a legitimate-security-work frame and then fed it a sequence of tasks that looked routine in isolation but taken together, advanced an intrusion workflow.

Three mechanics matter:

1. Role persistence: long-running sessions let an operator establish a coherent narrative over time, reducing the number of "hard resets" where guardrails might re-evaluate context.

2. Task fragmentation: instead of asking for an obviously malicious end-state, the operator decomposes work into smaller requests that each resemble normal troubleshooting, testing, or audit activity.

3. Orchestration wrapper: the surrounding system (tools + state management + handoffs) matters as much as the model. The model becomes an execution engine inside a larger workflow that sequences tasks, tracks progress, and aggregates outputs.

Traditional security telemetry captures tool execution and network activity, but not the language-layer control plane that produced those actions. If prompts, tool invocations, and context shifts are invisible, defenders are blind to an entire class of manipulation.

This manipulation class has three properties worth calling out:
1. Low Barrier to Entry: it requires more patience and framing discipline than deep exploit development.
2. Hard to "Patch": tightening safeguards without breaking legitimate use cases is non-trivial; monitoring and friction often matter more than trying to make refusals perfect.
3. Poor Visibility in Standard Logs: SIEMs log actions; they rarely log the persuasive narrative and stepwise steering that triggered those actions.

Anthropic notes the sustained nature of the activity eventually triggered detection. But "eventually" is not a defensive plan in a world where reconnaissance, validation, and lateral movement can be driven in parallel.

5. WHAT ACTUALLY CHANGED: THE TEMPO OF INTRUSIONS

GTG-1002 did not invalidate existing security models but exposed how dependent those models were on time. Even in mainstream incident reporting, defenders are still measured in days and weeks: median dwell time is still counted in days, and common remediation timelines for exposed edge devices or leaked credentials are measured in weeks. That gap is exactly what orchestration exploits. It does not need a novel trick. It just needs you to be slow.

The reality is simpler: AI did not introduce fundamentally new techniques.

Most techniques remain recognizable. reconnaissance, credential abuse, lateral movement, data collection. AI-enabled orchestration has collapsed the unit-cost of an intrusion. It allows an adversary to run more attempts, across more surfaces, with zero fatigue.

Do we need entirely new defensive principles? No. Least privilege, segmentation, secrets hygiene, and egress control still work. Our **latency of enforcement** is the issue especially after a major zero-day CVE public disclosure. A principle that only exists in a static configuration file cannot survive an attacker that branches and pivots in seconds.

Are AI-driven intrusions unpredictable? The sequence is predictable. What gets harder is the volume and branching. You still see the same invariants—unusual access patterns, identity misuse, internal enumeration, suspicious data movement. The burden shifts to correlation across systems and fast containment decisions, not fortune-telling.

Do human-speed SOCs remain viable? Human judgment still matters, but humans cannot sit in the tactical critical path by default. If response depends on a person approving every step, the system will fail under parallelized pressure. The viable model is pre-approved guardrails with automated containment for clearly bad patterns, and humans supervising, tuning, and handling the ambiguous edge cases.

We've spent decades building trust models that assume a user is a person who gets tired. Our session timeouts, our manual escalation paths, even our 'human-in-the-loop' mandates—they are all, essentially, **biological filters**. We assume the attacker will eventually make a mistake because of fatigue. AI removes that 'fatigue tax' entirely.

Reconnaissance that once unfolded over days now occurs in minutes. Exploit validation that once required careful, manual testing can be performed in seconds. Lateral movement that depended on human attention now proceeds algorithmically. None of these steps are new.

The compression is—and AI lets even weak threat actors execute it correctly.

James Mickens at Harvard has a funny memorable bit he calls the Mossad/Not-Mossad Threat Model. The idea is simple: if your adversary is Not-Mossad, you'll probably be fine with a strong password and not clicking on sketchy links. If your adversary is the Mossad, you're going to die and there's nothing you can do about it. The framing is funny because it's basically true—or was.

With AI doing the orchestration, even mediocre threat actors start to look like the Mossad or Tier 1 nation-state actors. There is no fatigue tax, and the time buffer is gone.

Midnight Blizzard (NOBELIUM) provides a clear illustration of what failure looks like even at human tempo. Russian state actors used password spraying—a technique decades old—against a legacy test account without MFA within Microsoft's environment. From there, they lived-off- the-land and created a new malicious OAuth application, pivoted into Microsoft's corporate tenant, accessed senior leadership, legal, and security team mailboxes, and exfiltrated data over a period of approximately six weeks.

What makes this incident instructive is not the tradecraft. It was mundane.

Microsoft operates one of the most mature security environments in the world, with pervasive telemetry across identity, endpoint, email, network, and cloud workloads, backed by deep internal visibility and one of the largest security organizations on the planet. They have access to some of the most advanced machine learning and large-scale analytic capabilities in the industry.

Signals existed. Logs existed. But **detection and containment depended on human-driven correlation and investigation rather than autonomous response**. Discovery emerged through manual log review and investigative analysis, not through automated systems surfacing and containing the intrusion in real time. By the time the activity was understood as an incident, access had already persisted for weeks and sensitive data had been exfiltrated from Microsoft. Subsequent reporting confirmed follow-on activity months later, underscoring how long identity-based persistence can survive once established.

Midnight Blizzard did not require zero-days, custom malware, or novel exploitation techniques. It succeeded by operating patiently within the boundaries of legitimate protocols—OAuth, email access, valid credentials—at a pace that human-centric workflows could not compress.

Microsoft was not an anomaly. Across nearly every major breach of the past several years, detection did not originate from automated security systems. These organizations all had modern tooling in place—firewalls, EDR, SIEM, identity platforms—but **the systems did not produce a coherent, time-bounded understanding of the attack while it was still unfolding**.

Logs existed. Alerts existed. Telemetry existed. What was missing was **real-time correlation and contextual synthesis across those signals**. Each control observed its own fragment of reality, but no system assembled those fragments into a single, high-confidence incident quickly enough to trigger decisive action. Log data alone was not the problem; the problem was that logs remained isolated artifacts rather than inputs into a living, adaptive decision system.

MGM and Caesars were compromised via Scattered Spider social engineering; visibility arrived through operational disruption and crisis response, not clean automated correlation. AT&T's Snowflake breach, which exposed data on roughly 110 million customers, was identified by an external security researcher rather than internal controls. The pattern is consistent: identity-based attacks that rely on valid credentials, legitimate protocols, and patient, human-tempo execution routinely evade automated defenses. Detection arrives late—through manual investigation, third-party notification, visible operational impact, or extortion.

These were all human-speed attacks. Midnight Blizzard persisted for approximately six weeks. Scattered Spider operated over days. The Snowflake campaign unfolded over weeks on a per-victim basis. In each case, the attackers succeeded not because defenders lacked tools, but because **those tools were never designed to reason collectively, in real time, about attacker intent and progression**.

This is the structural flaw in legacy security architectures. Most environments still operate as collections of independent controls optimized for logging, alerting, and post-hoc investigation. Correlation is retrospective. Context is reconstructed manually. Response depends on humans stitching together timelines after enough evidence accumulates. That model assumes time is available.
Now contrast that with GTG-1002's operating tempo: parallel execution across dozens of targets, thousands of requests, often multiple per second, frequently against organizations with far less mature security posture than Microsoft. In that environment, waiting for humans to correlate logs—or for SIEM rules to accumulate enough signals—guarantees failure.
If human-paced attacks can persist inside Microsoft for weeks, an AI-orchestrated campaign

operating at machine tempo would complete its reconnaissance-to-exfiltration cycle before the first alert ever reaches triage. The failure mode is not tooling. It is **architecture and operating model**. Our defenses are calibrated to observe humans. We are now defending against algorithms.

Many security programs still assume biological constraints: fatigue, hesitation, linear progression, and predictable pacing. Those assumptions underpin how alerts are generated, how incidents are escalated, and how response authority is granted. AI-orchestrated attacks remove those constraints entirely. There is no longer a stable tempo to defend against—and without true, real-time correlation and context across systems, log data becomes an artifact of failure rather than a trigger for prevention.

Here is an example of what "tempo" feels like in a real SOC—not based on GTG-1002:

02:13 — A service account that normally talks to one internal service begins touching dozens. No single request looks alarming. A few rate-limit warnings show up. A low-severity alert fires, gets queued.

02:15 — Authentication failures spike across several internal endpoints. The IAM console shows a pattern, but it's still within the range that could be a misconfiguration or a bad deploy.
02:18 — The same identity starts issuing unusually broad queries against a database it has never accessed before. The database logs it. The SIEM records it. The DLP system hasn't fired because nothing has left the network yet.

02:21 — A bulk transfer begins to a destination that is technically allowed (backups, analytics, partner integrations). Someone needs to decide whether to block it. The on-call engineer is still paging into the incident.

02:26 — A human analyst finally correlates the identity pattern + endpoint pattern + database behavior. That correlation is the "incident." By this point the attacker has already learned what works, what doesn't, and where the valuable data lives. At that point, the investigation resembled an autopsy rather than active containment.

Defenders are not incompetent. It's that a workflow built around tickets, handoffs, and manual validation is structurally too slow as time compression compounds. An AI system does not get tired, does not context-switch poorly, and does not need to "come back tomorrow." It can maintain persistent state across targets and resume instantly. Parallelism

costs the attacker almost nothing. This is why traditional human-in-the-loop security operations struggle.

Game-Theoretic Foundations: OODA Loop and Colonel Blotto

The human-AI teaming or Centaur paradigm gains additional analytical power when connected to two established frameworks from military strategy and game theory: Colonel John Boyd's OODA Loop and the Colonel Blotto game.

Boyd's OODA Loop: Temporal Compression

U.S. Air Force Colonel John Boyd developed the Observe-Orient-Decide-Act (OODA) Loop in the 1970s based on his experience as a fighter pilot and military strategist. Boyd's central insight was that the entity which can cycle through decision-making faster than its opponent can "get inside" the opponent's decision cycle, causing confusion and gaining decisive advantage.

> *'The ability to operate at a faster tempo than an adversary enables one to fold the adversary back inside himself so that he can neither appreciate nor keep up with what is going on.'*
>
> — Colonel John Boyd

GTG-1002 represents the weaponization of Boyd's insight at rapid speed. Where human SOC analysts require hours to cycle through Observe (review alerts) → Orient (contextualize threat) → Decide (determine response) → Act (execute containment), AI-orchestrated attacks complete this cycle in seconds. The result is precisely what Boyd predicted: defenders cannot keep up with the tempo of events, leading to disorientation and ineffective response. The attacker's Observe–Orient–Decide–Act loop can run on machine time while most security processes still run on human time. That tempo gap is enough to break workflows that depend on escalation, ticketing, and manual verification.

Google Cloud's Phil Venables, CISO, has explicitly connected OODA to cybersecurity: "The well- functioning security team of the future is one that moves fast with accuracy to detect, disrupt, and respond to the actions of even the most capable adversary. Security teams and their leaders who can move quickly through the OODA loop could make the difference in preparedness and resiliency." GTG-1002 demonstrates what happens when attackers achieve OODA dominance.

Colonel Blotto Games: Multi-Battlefield Resource Allocation

The Colonel Blotto game, first formalized by Émile Borel in 1921, models competitive resource allocation across multiple battlefields. Two players simultaneously distribute limited resources across n battlefields; whoever allocates more resources to a given battlefield wins it. The game has been extensively applied to cybersecurity, particularly for analyzing how defenders should allocate limited security resources across multiple potential attack surfaces.
GTG-1002 can be analyzed as a Colonel Blotto game with approximately 30 battlefields (target organizations). However, AI fundamentally disrupts the classical Blotto equilibrium assumptions:

- **Simultaneity Violation:** Classical Blotto assumes simultaneous resource allocation. AI enables dynamic reallocation in seconds while defenders remain static.
- **Information Asymmetry:** AI reconnaissance provides near-complete battlefield visibility before committing exploitation resources.
- **Resource Multiplication:** AI effectively multiplies attacker resources through parallel execution, breaking the resource constraint assumption.

At campaign scale, the attacker's advantage is not a single breakthrough exploit. It's the ability to allocate effort dynamically across many targets, pushing hard where defenses are weak and moving on quickly where they are not. Research by Gupta, Schwartz, and Sastry (2014) on Colonel Blotto games for cyber-physical security demonstrates that defenders who fail to account for attacker adaptability consistently lose more battlefields than Nash equilibrium would predict. GTG-1002 represents the extreme case: an attacker with near-infinite tempo advantage operating across 30 battlefields simultaneously.

Synthesis: OODA + Blotto + Centaur

These three frameworks converge:

- **OODA explains** why faster decision cycles create decisive advantage
- **Blotto explains** how multi-battlefield dynamics favor attackers with tempo advantage
- **Centaur explains** why human-AI teaming with optimized process outperforms either alone

Defenders must adopt all three: accelerate their OODA loop through AI automation, coordinate across all battlefields through correlated detection and automated response, and optimize the human-AI interface through Defensive Centaur Architecture. In a tempo contest, process and automation become decisive capabilities.

6. ORCHESTRATED DEFENSE FRAMEWORK

GTG-1002 didn't break new ground in offensive technique. It broke the assumption that defenders have time to react. The defensive response isn't a new set of principles—it's the old principles executed at machine speed, with coverage extended to surfaces that didn't exist five years ago.
You cannot fight coordinated offense with isolated controls. Each layer must feed the others, and the system as a whole must act faster than a human can context-switch between dashboards.

6.1 The Problem: Isolated Controls Failed

Traditional defense-in-depth did not fail because controls were absent—it failed because isolated controls could not correlate fast enough.

The threat actors targeted 30 organizations simultaneously and progressed from reconnaissance to exfiltration in minutes. The attack coordinated network scanning, database queries, credential testing, and data extraction in parallel.

Individual security tools saw fragments. The SIEM logged unusual database queries. EDR detected scanning activity. Network monitoring flagged increased traffic. IAM recorded authentication attempts. But none of them talked to each other fast enough to recognize that these were pieces of the same attack. By the time a human analyst could connect the dots, the AI had already moved on.

AI offense uses orchestration. Traditional defense still uses isolation.

Saltzer and Schroeder's 1975 "Principle of Least Privilege" is still the gold standard. Cybersecurity principles did not fail. GTG-1002 succeeded not by breaking the rules but by exploiting organizations that never properly implemented them—and by moving faster than their enforcement mechanisms could respond.

6.2 Identity: When Agents Never Sleep

We've spent decades building trust models that assume a user is a person who gets tired and works at a predictable pace. Session timeouts exist because humans take breaks. Rate limits are calibrated to human typing speed.

Behavioral baselines assume someone goes home at night.

GTG-1002 exploited this by hijacking service accounts and CI/CD pipelines—identities that already operate at machine speed and blend into the normal background noise of enterprise automation. The attack didn't look anomalous because modern enterprises are full of automated processes making thousands of API calls. One more didn't stand out.

The fix isn't complicated in concept: stop collapsing humans, scripts, and AI agents into the same identity bucket. AI agents are a distinct class of principal. They need scoped permissions (narrowest possible access), just-in-time tokens (not persistent credentials), hard rate limits (not "reasonable use" assumptions), and separate behavioral baselines (what's normal for an agent isn't what's normal for a human).

If an agent identity is only authorized to touch three specific tables, autonomous extraction dies in its tracks. But if your service accounts have broad read access "for flexibility," you've pre-authorized the kill chain. This also means rethinking how you provision AI tooling. When someone spins up an internal LLM agent or connects a third-party copilot, what identity does it authenticate as? If the answer is "the user's identity" or "a shared service account," you've created exactly the ambient authority that GTG-1002 exploited.

Detection on this surface: Abnormal privilege usage—accounts querying databases never previously accessed. Temporal anomalies—authentication patterns inconsistent with historical behavior.
Impossible travel—credential use from multiple geolocations simultaneously. Service account abuse— AI agents often hijack service accounts with broad access; watch for behavioral shifts in these identities specifically.

6.3 Network: Automated Containment, Not Alerting

The reason GTG-1002 could hit 30 targets at once is that most internal networks are still essentially flat. Once you're past the perimeter, you can wander. Micro-segmentation exists in the architecture deck, but in practice, a compromised identity can often reach far more systems than it should.

The standard response is "better segmentation"—which is correct but insufficient if segmentation is enforced through manual processes. A human analyst waiting for a Jira ticket to approve a firewall rule change is a structural bottleneck when the attacker is pivoting in seconds.

The shift is from alerting to automated containment. If a system begins anomalous lateral

scanning, the defense must trigger a sub-second quarantine—isolate first, investigate second. This feels risky to organizations conditioned on "don't break production." But the alternative is letting the attacker complete reconnaissance while you're still triaging the alert. This requires pre-authorized playbooks. Security leadership has to decide in advance: under what conditions can the system isolate a host, revoke a credential, or block a network segment without human approval?

Those decisions can't be made during an incident. They have to be made now, encoded into policy, and tested until you trust them.

The goal isn't "no human in the loop." It's "human out of the critical path for obvious cases." Humans supervise, tune, and handle ambiguity. But the first-response actions—containment, isolation, credential revocation—have to be automatic for high-confidence detections. Detection on this surface:

Traffic volume and velocity anomalies—thousands of connections per second from a single source. Lateral movement patterns—scanning across multiple IP ranges in rapid succession. Protocol anomalies and tunneling. East-west traffic inspection—watch service-to-service communication for patterns that don't match expected application behavior.

6.3.1 Guardrails for Autonomous Response

Autonomous response is necessary—but not without governance. Three failure categories demand advance planning:
False positives that shut down legitimate operations. *Mitigation:* Graduated response tiers; shadow-mode testing; sub-60-second manual override.
Liability gaps when automated actions cause downstream harm. *Mitigation:* Map containment capabilities against contractual and regulatory obligations before deployment.
Cascading failures when isolating one system breaks dependent services. *Mitigation:* Protected zones requiring human confirmation; circuit breakers that pause automation if containment volume spikes.
Decide now: Which assets can be auto-isolated? What confidence threshold justifies revocation? Who overrides, and how fast?

6.4 The Model Control Plane: Monitoring Intent, Not Just Action

This is the surface most organizations aren't watching at all.

Traditional security telemetry captures what happened: network connections, file access, database queries, authentication events. That's necessary. But GTG-1002 succeeded by manipulating *why* things happened—the cognitive sequence that led Claude to take actions it would normally refuse.

The model wasn't compromised through a code vulnerability. It was steered through conversation. Long-running sessions established a "legitimate security work" frame. Tasks were fragmented into benign-looking pieces. Context accumulated until the model was deep into an intrusion workflow without any single request triggering a refusal.

If you're deploying internal LLM agents—for code generation, data analysis, customer support, whatever—you need visibility into this layer. That means logging prompts, tool invocations, and context shifts. It means watching for "prompt drift," where an agent's behavior gradually diverges from its system instructions. It means having detection logic that asks: is this agent being steered somewhere it shouldn't go?

Most security tooling doesn't know what a prompt is. But if you're running agents that can take actions—query databases, execute code, call APIs—and you're not monitoring the cognitive layer, you're blind to the manipulation that precedes the malicious action. By the time the database query hits your logs, the attack has already succeeded.

The practical challenge is that prompt-layer monitoring is immature. There's no established playbook, limited vendor tooling, and real tension between security visibility and user privacy. But the absence of easy answers doesn't mean you can ignore the surface. At minimum, you need logging infrastructure that captures agent interactions, even if your detection logic is still primitive. At some point prompts must be treated with the same forensic rigor as network packets.

6.5 Transitive Trust: The Orchestration Supply Chain

GTG-1002 leveraged Model Context Protocol (MCP) and third-party orchestration wrappers. This introduces a vector that most detection architectures aren't designed for.

Traditional threat models assume attacks arrive from "external" IPs or untrusted sources. Your firewall distinguishes inside from outside. Your SIEM assigns risk scores based on source reputation. But modern AI systems rely on a web of API handoffs, plugins, and partner integrations. The agent calls a tool. The tool calls an API. The API calls another service. Trust is transitive—and so is compromise.

If a third-party agentic browser or MCP server gets hijacked, the attack originates from a "trusted" API call, not a suspicious external connection. Your firewall sees a legitimate partner. Your SIEM sees normal traffic patterns. The threat came in through the front door with a valid badge.

This is supply chain risk applied to the orchestration layer. You probably already think about supply chain security for code dependencies—do you pin versions, scan for vulnerabilities, audit maintainers? The same discipline now applies to every tool, plugin, and API your AI agents can invoke.

Questions to ask: What third-party services can your agents reach? What happens if one of them is compromised? Do you have detection coverage for malicious actions that arrive through "trusted" channels? Can you rate-limit or circuit-break integrations that start behaving anomalously?

Most organizations can't answer these confidently. That's the gap GTG-1002 exploited—not with sophisticated tradecraft, but by operating through layers of assumed trust.

6.6 Deception: Use Their Speed Against Them

Here's one domain where AI-orchestrated attacks actually favor the defender.

Human attackers are surgical. They probe selectively to avoid noise, carefully choosing which systems to touch and which to leave alone. Detection avoidance is labor-intensive.

AI agents are exhaustive. They touch everything they can reach because it's cheap and fast. They don't get tired. They don't worry about "being too noisy." Systematic enumeration is the default behavior.

This flips the economics of deception. Against a human adversary, you need a few sophisticated honeypots positioned in exactly the right places. Against an AI adversary, you need coverage.
Honeytokens everywhere—fake credentials in configuration files, decoy API keys in environment variables, canary database connection strings scattered across systems. Decoy databases that look valuable. Fake network segments designed to appear interesting.

In a world where the attacker touches every door handle, you just need to make sure most of those handles are wired to alarms. Density beats sophistication. You can even use circular logic loops where traps are placed to redirect agents into infinite loops. The practical implementation: treat deception as infrastructure, not a special project. Every system deployment should include canary artifacts. Every configuration template should embed honeytokens. The goal is saturation—enough tripwires that an exhaustive scan triggers something within seconds.

6.7 Vulnerability Response: The End of the Weekly Cycle

If an AI can weaponize a new CVE within minutes of disclosure, a 48-hour patch cycle is obviously too slow.

Exploit code generation is one of the tasks LLMs do reasonably well. The gap between "vulnerability disclosed" and "exploit available" is compressing. If your response timeline is measured in change management windows, you're operating on a tempo that assumes human-speed adversaries. One way to address this is by creating a process for virtual patching. The moment threat intelligence arrives—a new CVE, a new exploit in the wild, a new technique observed—your defensive systems should be able to push temporary mitigations without waiting for a committee. WAF rules, IPS signatures, access restrictions, network blocks. Not a permanent fix, but a speed bump that buys time for proper remediation.

This requires pre-authorized "Shields Up" protocols. Security and operations leadership have to agree in advance: under what conditions can the system tighten its posture automatically? What's the tolerance for false positives? What's the rollback process if a mitigation breaks something?

The organizations that do this well treat it as a drill. They practice emergency patching. They have playbooks for "reduce attack surface immediately" that don't require waiting for a CAB meeting. The ones that don't will find themselves exposed during the window between disclosure and deployment— a window that's shrinking fast.

6.8 The Adaptation Gap: Why Static Playbooks Fail

Detection is necessary but not sufficient. Response has to match the velocity of the attack.

The standard playbook here is automated containment: quarantine endpoints without waiting for human approval, revoke credentials showing abuse patterns, segment the network, kill suspicious processes, block exfiltration at egress. All of that is correct. When an attacker is

making thousands of requests, human approval loops don't work.

But here's what most automated response systems get wrong: they don't adapt. To be clear, we are referring to agentic systems here.

A static playbook says "if X, then block Y." That works until the attacker learns your playbook—which, against an AI-orchestrated campaign, happens fast. The attacker probes, gets blocked, adjusts, and probes again. If your response system does the same thing every time, you're playing a game where the attacker learns and you don't. This is the dirty secret of agentic security demos. They look incredible on stage. The agent detects the threat, correlates across surfaces, triggers containment, problem solved. Many automated response systems implicitly assume a static adversary. A response layer that cannot measure whether an action stopped progression will be outlearned in minutes.

The attacker in the demo doesn't adapt. The network topology doesn't shift. The agent makes a decision, it works, everyone claps. But real adversaries—especially AI-orchestrated ones—iterate. They probe, get blocked, notice the block, and try something different. Meanwhile your "agentic" response system is still executing the same decision tree it ran yesterday. It's not observing the effects of its own actions. It's not noticing that the attacker changed tactics. It just keeps making the same move, over and over, while the environment has already shifted.

This can’t be addressed with fine-tuning. You don't fix it by retraining the model quarterly. This is a real-time observation problem. Today’s agents do not adapt. The response system needs to watch what happens *after* it acts. Did the containment actually stop the progression? Or did the attacker pivot to a different path thirty seconds later? Did revoking that credential kill the session, or did the attacker already have three other credentials staged? If your response layer can't answer those questions in real-time, it's not adapting. It's just executing a script and hoping. The AI community is working to overcome these current limitations with the creation of adaptive agents, but we are not there yet.

The response layer needs feedback loops. Which containment actions actually stopped progression? Which ones just displaced the attack to a different vector? Did the attacker retry with a variant? Automated response without adaptation is just a speed bump. It slows things down, but it doesn't learn. Against an adversary that iterates in seconds, you need response logic that updates based on what's working—not a runbook that was written six months ago and hasn't changed since.

This is hard to build. Most organizations aren't there yet. Most vendors aren't there yet either— current products automate response but rarely learn from it. But it's the direction of travel: response systems that observe their own effectiveness and adjust in real-time,

rather than executing the same static playbook until someone manually updates it after the postmortem.

6.9 The Centaur SOC: Human-AI Teaming Done Right

The GTG-1002 campaign did not emerge from a novel use of AI so much as from a familiar pattern applied to a new domain. After IBM's Deep Blue defeated Garry Kasparov in 1997, many assumed that humans would become irrelevant in chess. Kasparov drew the opposite conclusion. In the years that followed, he helped popularize *Advanced Chess*, where humans partnered with machines rather than competed against them. The surprising result was not that computers dominated humans, it was that well-organized human-machine teams routinely outperformed both grandmasters using computers poorly and computers operating alone.

Kasparov later summarized the lesson bluntly:

"Weak human + machine + better process was superior to a strong computer alone and, more remarkably, superior to a strong human + machine + inferior process."

Claude was not unique. Anyone with access to modern LLM tooling could, in principle, reproduce its raw capabilities. What differentiated the attackers was the process layer: how tasks were decomposed,

How context was maintained, and how human oversight was applied sparingly rather than continuously. Defenders should take this personally. Many SOCs still operate as if humans must remain in the tactical execution loop for safety. GTG-1002 suggests that assumption is becoming a liability.

The Asymmetry Today:

Dimension	GTG-1002 (Attack Centaur)	Typical SOC (Human-in-the-Loop)
Operational tempo	Seconds-level execution; multiple ops/second at peak	Minutes-to-hours decision loops; queueing and handoffs dominate
Parallel operations	High parallelism across ~30 targets	Concurrency constrained by staffing and tool integration
Human role	Target selection, key approvals, output review	Investigation, escalation, containment authorization, tuning
AI role	Tool execution, workflow sequencing, rapid triage, context maintenance	Detection + enrichment + triage; selective automation varies
Process layer	Custom orchestration with state management	Ticket/runbook driven; cross-tool correlation often manual
Consistency	24/7 consistent execution	Shift-based; fatigue and context loss at handoff

This is the chess equivalent of a grandmaster playing without a computer against an amateur with three engines and a superior process. Raw human expertise cannot compensate for the speed and consistency advantages of human-AI teaming.

Defensive Centaur Architecture

The solution is not to replace human defenders with AI—just as the solution in chess was not to replace human players with engines. The solution is to build security operations designed around human-AI teaming with optimized interfaces and processes.

Defensive Centaur Architecture comprises three layers:

The **AI Tactical Layer (Seconds-Scale):** This layer handles automated detection, correlation,

validation, and first-loop containment which is fast enough to keep up with high-volume, parallel activity. It operates under a 'Threshold of Autonomy'—a confidence score above which the AI can alter production environments without human approval. This threshold is not universal. It depends on business vertical: a hospital cannot tolerate the same automated containment actions as a marketing agency. One hour of downtime in an ICU is a patient safety crisis; one hour of downtime for a campaign dashboard is an inconvenience. A trading floor, a power grid control system, and a SaaS analytics platform will each have fundamentally different autonomy boundaries. Many research firms such as Gartner, Forrester, and IDC predict that task-specific AI agents will absorb the bulk of Tier 1 SOC workloads over the coming years, enabling humans to focus on complex judgment calls.

Human Layer (Judgment and Creativity): Security architects, threat hunters, and incident commanders focusing on adversary intent, organizational risk tolerance, business impact assessment, and novel threat identification. Humans excel at understanding context, inferring motivation, and making decisions under uncertainty—capabilities where AI remains limited.

Optimized Interface Layer (Process Excellence): The critical differentiator. Pre-authorized playbooks, confidence-scored escalations, contextual dashboards, and feedback loops that allow humans to guide AI behavior while AI handles execution. This is where the Kasparov insight applies most directly—process quality determines competitive outcome.

CSIRO's Collaborative Intelligence program uses the term "Cyber Centaurs" for the direction of travel: human defenders augmented by AI and automation, with humans retaining accountability for decisions and outcomes. It is early, but it is the right shape of solution.

6.10 Implications for Security Leadership

For CISOs and security executives, the C entaur paradigm demands a fundamental reconceptualization of security operations—not as a vision statement, but as a measurable transformation with budget, staffing, and architectural implications.

Rethink Your Metrics

Traditional SOC metrics—alerts processed, tickets closed, MTTD measured in hours—were designed for human operations. New metrics for Centaur SOCs: automated containment rate (threats stopped without human intervention), human override rate (too high suggests miscalibrated automation; near-zero suggests rubber-stamping), escalation accuracy (true

positives requiring judgment vs. false positives wasting analyst time), and threat-to-containment velocity measured in seconds, not hours.

If your board still sees "alerts per analyst" as a productivity metric, you're optimizing for the wrong war.

Restructure Your Team

The traditional SOC pyramid—large Tier 1 teams triaging alerts, smaller Tier 2 investigating, tiny Tier 3 hunting—inverts in a Centaur model. Tier 1 becomes largely automated; AI handles volume while humans review exceptions. Tier 3 expands: threat hunters searching for what automation misses, process designers improving human-AI workflows, detection engineers tuning models.

The implication: fewer junior analysts monitoring dashboards, more senior practitioners designing systems. This changes your hiring profile, compensation structure, and career ladders. The emerging role isn't "SOC analyst with AI tools"—it's **AI workflow engineer**: someone who designs the human-machine interface, tunes automation thresholds, and owns the feedback loops.

Budget for Process, Not Just Product

Most security budgets allocate heavily to tools and lightly to integration. Centaur architecture inverts this. Expect to spend as much on workflow design, playbook development, and interface optimization as you do on the underlying platforms. The differentiator is not which XDR you deploy—it's how effectively your humans and machines collaborate through it.

Prepare the Board Conversation

Frame it as risk transfer: "We are moving tactical response decisions from humans operating at minute-scale to automated systems operating at second-scale. This reduces exposure windows from hours to seconds—but introduces new risks around false positives and automated errors. Here's how we're engineering guardrails..."

Boards don't need to understand OODA loops. They need to know you've thought through both sides of the trade-off.

Accept That This Is Hard

Centaur architecture is not a product you purchase. It's an operating model you build—like zero-trust. Most organizations will get it wrong on the first attempt. The ones that succeed will treat this as a multi-year transformation with explicit learning cycles—not a vendor deployment with a go-live date.

6.11 Red Teaming for AI

Traditional red teams simulate human-speed attacks. AI-orchestrated campaigns are fundamentally different. If you're using LLM features or wiring agents to sensitive data, validate:

- **LLM logic:** Prompt injection, jailbreaks, guardrail bypasses
- **Infrastructure:** IAM escalation, egress gaps, segmentation failures
- **RAG pipelines:** Poisoned embeddings, corpus exposure
- **Integrations:** MCP/plugin abuse, transitive trust, credential leakage

Key questions:

Can agents escalate beyond scoped permissions?

- Does containment execute fast enough to matter?
- Where do AI operations go unmonitored?

6.12 Framework Summary

These layers—identity, network, model control plane, transitive trust, deception, vulnerability response, and adaptation—aren't separate projects. They're components of a single system that has to operate cohesively at machine speed, with humans and AI teaming effectively.

Each security tool saw a fragment. Correlation depended on humans. Humans are slow. The attacker moved faster than the defenders could think.

Orchestrated defense means: identity signals inform network containment. Network

anomalies trigger deception analysis. Prompt drift detection feeds incident correlation. Transitive trust violations escalate to automated response. The layers talk to each other in real-time, and the system acts on
high-confidence detections without waiting for a human to connect the dots. And when it acts, it watches what happens next—and adjusts.

The best security teams of the future will not be the ones with the most sophisticated AI or the most experienced analysts. They will be the ones with the best process for combining human judgment and machine capability.

7 CISO/CEO/BOD TAKEAWAYS AND QUESTIONS

Most big incidents don't hinge on a single exotic exploit. They hinge on access, movement, and time. Attackers (especially nation-states and ransomware crews) win when they can turn one foothold into broad control: identity, deployment pipelines, using existing remote access tools, backups, and the data layer. AI-orchestration doesn't change the steps. It compresses the timeline and increases the parallelism. That makes architecture choices and pre-authorized response more important than ever.

7.1 For CISOs: Architecting for Latency and Survival

If your Mean Time to Remediation (MTTR) is measured in hours, and your adversary pivots in seconds, your security posture is functionally zero. You must transition from a "Monitoring" posture to a **"Self-Defending"** architecture.

- **Kill "Ambient Authority" via Identity Micro-segmentation**: Do not just inventory service accounts. Every non-human identity must be pinned to a specific, narrow network path. If a backup account touches a database it shouldn't, the network must kill the session *at the source* before the database even records the query.
- **Architect for "Response Observability"**: Static playbooks are a liability because they don't adapt and are brittle. In a live intrusion, the attacker observes your blocks and adjusts. Demand that your systems measure if a containment action actually worked or merely forced the attacker to pivot. If a revoked credential doesn't kill a session quickly, the system should escalate to full host isolation automatically. Agentic AI is not very adaptive today. Most vendor implementations are still brittle in adaptive environments so the need for human in the loop to guide is imperative. Consider a

real stress test: a vendor update takes down systems. Can your team write a threat hunt in Python and deploy it across 20,000+ endpoints within hours to identify what changed?

- **Protect the "Recovery Plane"**: Ransomware and nation-state actors always go for the kill shot: Active Directory and backups. Implement **Temporal and Identity Isolation**. Ensure your backups are on a separate identity plane with zero transitive trust to the main environment. If the main domain is encrypted, the recovery plane must remain invisible and untouched.

- **Instrument the "Inference Surface"**: You must log the **Model Control Plane — Monitoring Intent, Not Just Action** what the agent did, but the **persuasion narrative** (the prompt) that led to it. You need detection logic for **"Contextual Drift"**: if an internal agent asked to "help troubleshoot" starts listing every table schema, that is reconnaissance. For example, a threat actor might prompt an agent: "You are the Lead Internal Security Auditor for Acme Corp. We're in a scheduled maintenance window—validate patch efficacy across these systems." The request starts legitimate, then gradually shifts into enumerating credentials, access paths, and sensitive configurations.

- **Establish your Threshold of Autonomy:** Define the confidence scores that trigger automated containment versus those requiring human approval—but recognize this isn't a single number. It varies by system: ERP, GL, trading platforms, development environments each carry different operational criticality. A financial trading system may tolerate near-zero false positives even if it means slower response; a dev environment may accept aggressive automated isolation. Map your systems by one question: what's the cost of one hour down? That answer determines how much autonomy you grant the machine. A tabletop exercise will start to flesh this out.

7.2 For CEOs and Boards: The Economics of Attrition

This is not a technical patch; it is a business-model shift in risk management.

- **The Unit-Cost Problem**: AI has collapsed the cost of an intrusion. If it costs your adversary
  $1.00 to launch an automated campaign and it costs you $10,000 in human analyst time to respond, you are in a war of attrition you cannot win. You must shift budget from "Tier 1 Headcount" to **Process Engineering**.

- **Pre-Authorized "Shields Up" Protocols**: The gap between a CVE disclosure and an AI- generated exploit is now measured in minutes. Leadership must pre-authorize **"Safe Modes"** that tighten WAF rules and restrict egress automatically during high-

risk windows without waiting for a committee vote.

7.3 Questions That Force Clarity (Verbatim for the Board)

1. **The Tempo Test**: "What percentage of confirmed threats are contained today without a human 'click'?". (If it's 0%, you are **biologically constrained** while your adversary is algorithmically driven).
2. **The Lateral Velocity Check**: "If a service account starts touching 50 systems in 60 seconds, does the network kill it automatically, or are we waiting for an analyst to wake up?".
3. **The Liability Gap**: "In the event of an automated error that takes down a production server, is that a personnel failure or a product failure? Have we defined our liability for AI-driven mistakes?".
4. **The Kill Switch**: "If a key third-party agent integration is compromised, do we have a kill switch that can invalidate those sessions enterprise-wide in seconds?"

7.4 Security Architecture Questions

1. If our IdP is compromised, how fast can we invalidate sessions everywhere, and what breaks when we do?
2. Which three systems, if controlled, would let an attacker deploy code into production?
3. Can any single admin identity delete, encrypt, or poison backups? Which one?
4. What's the fastest path from a compromised laptop to production data, and have we tested it end-to-end? Think Attacker Path.
5. Which service accounts have broad read access "for convenience," and what would it take to narrow them?
6. What outbound paths from sensitive environments are effectively unmonitored today (partner pipes, SaaS sync, backups, analytics)?
7. Which OAuth apps and API tokens have high-risk scopes, and how quickly can we revoke them at scale?
8. What are our top ten "blast radius" privileges (domain admin, cloud owner, pipeline admin, backup admin), and what is different about how we protect them?

9. What is the first containment action we're willing to automate today, and what evidence would justify expanding that list?
10. If an internal agent touched production systems last week, can we prove what it did from logs, or are we guessing?

8. Conclusion: Strategic Imperative

GTG-1002 is not merely an incident to be analyzed and forgotten. It is an inflection point that demands recognition and response.

The campaign demonstrated that:

- AI-orchestrated attacks are no longer theoretical—they are now plausible and increasingly likely against real organizations with real consequences.
- Traditional security controls remain essential, but human-speed implementation and monitoring can no longer keep pace with machine-speed offense.
- The Centaur paradigm—human strategic direction combined with AI tactical execution through optimized processes—has been weaponized by adversaries.
- Defenders who do not adopt equivalent human-AI teaming architectures face a structural disadvantage that no amount of human expertise can overcome.

Even if subsequent investigation modifies our understanding of GTG-1002's scope, the architectural recommendations in this paper remain valid because they address structural vulnerabilities that any AI-orchestrated attack would exploit. Algorithmic Friction—the deliberate introduction of delays, deception, and verification barriers that exploit inherent LLM weaknesses—offers a promising defensive layer. Recent research from Ben-Gurion University ("Cloak, Honey, Trap: Proactive Defenses Against LLM Agents") demonstrates 100% efficacy in neutralizing autonomous LLM agents across CTF environments using techniques that cloak assets with misdirection, deploy honeytokens that distinguish AI from human reconnaissance, and trap agents in logic loops that induce hallucinations. Critically, most of these techniques exploit structural vulnerabilities in how LLMs process context and reason under uncertainty—not prompt injection.

The path forward requires simultaneous action on multiple fronts: implementing the security fundamentals that Saltzer and Schroeder articulated fifty years ago,

deploying the new capabilities that AI-era threats demand, and architecting Defensive Centaur systems that match the speed and coordination of Attack Centaurs.

The Historical Pattern: Discovery vs. Implementation

Current discourse assumes AI capability will accelerate indefinitely. A longer historical view suggests a different rhythm.

Since the 1950s, AI progress has been defined by singular architectural leaps at irregular but roughly generational intervals, each followed by an extended period where that breakthrough is refined, scaled, and operationalized:

- **1950s–60s (Symbolic AI):** Turing's "Imitation Game" and the Dartmouth Conference defined the field; Rosenblatt's Perceptron (1957) promised learning machines before hitting hardware limits.
- **1980s (Connectionist Revival):** Backpropagation gave us the mathematics for multi-layer networks, setting up two decades of incremental progress.
- **2012 (The AlexNet Moment):** GPUs proved deep learning was computationally viable, unlocking the current era.
- **2017 (The Transformer):** "Attention Is All You Need" paper from Google provided the architecture powering today's LLM revolution.

The pattern is not clockwork, but it is recognizable: a breakthrough, then a decade or more of implementation before the next leap.

The contrarian implication: If this pattern holds, we may be transitioning from the *discovery phase* of the Transformer architecture to the *orchestration phase*. The innovations dominating 2024– 2025—test-time compute, chain-of-thought reasoning, agentic workflows, GTG-1002 itself—are not new architectures. They are orchestration innovations built on the Transformer foundation.

History suggests that the "magic" of a new architecture eventually reaches diminishing returns. When it does, competitive advantage shifts from those who *build* the models to those who *orchestrate* them.

In this view, the Defensive Centaur is not a temporary fix awaiting the next AI breakthrough. It may be the defining security architecture for the remainder of the 2020s. While the field waits for whatever comes next—perhaps in the early 2030s, perhaps later—the organizations that achieve resilience will be those that master process excellence using the tools we have today.

What Comes Next (2026–2030)

Looking ahead, several developments are highly plausible if the GTG-1002 pattern persists:

1. **Attack-Centaur kits will likely appear in gray markets.** Off-the-shelf AKO frameworks, combining LLM agents, exploit libraries, and orchestration logic, will appear in gray markets—dramatically lowering the barrier for second-tier actors. Just like ransomware as a service market, threat actors will create these criminal business models.
2. **Defensive Centaur SOCs become the default at scale.** Leading enterprises and governments will run SOCs where AI handles most first-line triage and containment, with human analysts focusing on strategy, investigation, and tuning.
3. **Regulators will anchor AI-security expectations in process, not magic.** Expect supervision regimes (financial, critical infrastructure, privacy) to push on *how* you design human-AI workflows, not on vague assurances about "responsible AI."
4. **Security talent strategy will change.** The highest-leverage security professionals will be those who can design centaur workflows: combining technical depth, product sense, and process design.
5. **The next GTG-100X will be less exotic—and more damaging.** Future campaigns may not be as headline-friendly as "AI-orchestrated espionage," but they will quietly exploit the same asymmetry: machine-speed orchestration against human-speed defense.

Organizations that recognize the paradigm shift and act decisively will achieve resilience. Those that do not will find themselves in a position analogous to grandmasters playing freestyle chess without computers—outmatched not by superior opponents but by superior human-machine integration.

The question is no longer whether organizations will adapt to the Centaur paradigm, but whether they will do so before the next GTG-1003.

APPENDIX

A. The Acceleration Layer: GPU Evolution and CUDA Tile Programming

To understand why these defensive capabilities must scale, we must examine the hardware trajectory that will accelerate both offense and defense. While GTG-1002 showcased what AI can achieve today using commodity hardware and open-source tooling, emerging GPU programming paradigms signal an additional acceleration layer that both attackers and defenders must anticipate. This is not about predicting imminent threats. It's about understanding where the infrastructure curve is heading, because that curve always matters eventually.

NVIDIA CUDA Tile: A New Programming Model

At GTC 2025, NVIDIA announced CUDA Tile—a tile-based GPU programming model that fundamentally reimagines how developers target NVIDIA Tensor Cores. With the release of CUDA
13.1 in late 2025, CUDA Tile became generally available. CUDA Tile is built on Tile IR (Intermediate Representation) and includes cuTile Python, enabling developers to write, define, and optimize tiled GPU kernels using familiar Python syntax while achieving peak GPU performance. The initial release targets Blackwell GPUs, with broader architecture support planned across the CUDA 13.x series.

The significance of CUDA Tile lies in its core capabilities:

- **Structured parallelism:** Native programming of GPUs within structured high-performance contexts, enabling fine-grained computation on data tiles with dramatically improved memory locality.
- **Portability across platforms:** CUDA Tile targets cross-platform compatibility, allowing optimized kernels to run across NVIDIA's GPU ecosystem without extensive rewriting.
- **Simplified optimization:** The programming model simplifies the creation of optimized, tile-based kernels that previously required deep expertise in GPU architecture.
- **Deterministic execution patterns:** Reduced overhead in multi-step computational pipelines through predictable performance scaling.

Why This Matters for AI-Orchestrated Operations

CUDA Tile reinforces a structural trend: the infrastructure layer is being optimized for exactly the class of parallelized, tileable operations that AI-driven orchestration relies on. GTG-1002 demonstrated AI chaining thousands of micro-actions with high-frequency parallel task execution, iterating based on environment state, and maintaining context across dozens of targets. As GPU abstraction improves through programming models like CUDA Tile, the barrier to deploying sophisticated AI-enabled operations continues to fall.

Implications for Cybersecurity

1. **Accelerated reconnaissance and exploit mutation:** GPU-accelerated local inference can dramatically reduce the time to generate, test, and mutate exploit payloads—operations that previously required cloud API calls can increasingly run on local hardware.
2. **Defenders gain parallel capabilities:** Defensive frameworks can leverage the same GPU acceleration for inference loops and correlation models, enabling defense to match the threat actors' offense.
3. **Democratization of sophisticated capabilities:** As GPU programming becomes more accessible, AI-enabled intrusion frameworks become available to mid-tier threat actors, not just nation-states with extensive resources.

B. Quantum Computing: Outlook and the Long-Term Acceleration Curve

Public discourse often invokes quantum computing as an imminent cyber risk, particularly around breaking encryption. However, a rigorous analysis—grounded in physics, engineering, and current research—reveals a more nuanced timeline. This section provides a sober assessment of where quantum computing actually stands and what it means for the threat landscape this paper addresses.

Current State: The NISQ Era and Recent Breakthroughs

We are currently in the Noisy Intermediate-Scale Quantum (NISQ) era. Current quantum machines have unstable qubits, suffer from high error rates, cannot sustain long

computations, and are suitable primarily for experimentation and domain-specific proofs-of-concept.

However, 2024-2025 has seen meaningful progress. Google's Willow chip, announced in December 2024 with 105 qubits, achieved a critical milestone: below-threshold quantum error correction. For the first time, error rates decreased exponentially as more qubits were added—the opposite of what had plagued quantum systems previously. Willow performed a random circuit sampling benchmark in under five minutes that would take classical supercomputers 10 septillion years. In October 2025, Google announced the Quantum Echoes algorithm—the first verifiable quantum advantage on hardware—approximately 13,000× faster on Willow than the fastest classical supercomputers, with demonstrated application to molecular structure computation.

Critical caveat: These achievements represent quantum supremacy (outperforming classical computers on specific benchmarks), not quantum advantage (solving practical, commercially-relevant problems faster and cheaper than classical systems). The logical error rates reported (~0.14% per cycle) remain orders of magnitude above the ~10^{-6} levels believed necessary for running meaningful, large-scale quantum algorithms.

5–10 Year Outlook: Early Practical Applications

Over the next decade, quantum systems will likely achieve early practical wins in specific domains: optimization problems, chemistry and materials simulation, portfolio and risk modeling, and logistics optimization. These are domains where quantum's inherent properties—superposition and entanglement—provide algorithmic advantages even with imperfect hardware.

However, these advantages will be task-specific, not general-purpose. Quantum computers will not replace classical computing; they will augment it for specific problem classes.

10–30 Year Outlook: Fault-Tolerant Quantum Computing

To break modern encryption (RSA-2048, ECC), fault-tolerant quantum computers (FTQC) would require millions of stable qubits, orders of magnitude better error correction than currently achieved, stable decoherence times, and extremely low noise thresholds.

Current projections place true FTQC arrival between the late 2030s and 2050s. This means quantum computing is not an immediate cryptographic threat—but the harvest-now,

decrypt-later threat is real. Adversaries may be capturing encrypted communications today with the intent to decrypt them once FTQC becomes available. This makes post-quantum cryptography adoption urgent despite FTQC being years away.

Where Quantum Intersects with AI Orchestration

Quantum computing does not accelerate AI the way GPUs do today. When it eventually matures, quantum will accelerate orchestration, optimization, and simulation.

Long-term quantum-AI synergies relevant to cybersecurity include:

- **Attack graph optimization:** Quantum algorithms for finding optimal paths through complex attack surfaces.
- **Massive parallel search:** Grover's algorithm providing quadratic speedups for searching through state spaces.
- **Strategy selection:** Quantum-enhanced optimization for real-time decision-making under uncertainty.
- **Environment simulation:** Realistic modeling of defender environments before committing attack resources.
- **Resource allocation optimization:** A quantum version of Colonel Blotto dynamics for multi-battlefield resource distribution.

Notably, quantum is not needed for attacks like GTG-1002—today's AI already achieves rapid speed and orchestration. For GTG-1002-style threats, quantum is not relevant today, but in the long-term technology curve it's something to think about.

REFERENCES

1. Anthropic. (2025, November). *Disrupting the First Reported AI-Orchestrated Cyber Espionage Campaign*. Anthropic Threat Intelligence Report.

2. Anthropic. (2025, November). *Full Technical Report (PDF)*.

3. Saltzer, J.H. & Schroeder, M.D. (1975). The Protection of Information in Computer Systems. *Proceedings of the IEEE*, 63(9), 1278-1308.

4. VulnCheck. (2025, April). 2025 Q1 Trends in Vulnerability Exploitation.

5. Schneier, B. (2025, November). AI as Cyberattacker. *Schneier on Security*.

6. Google Threat Intelligence Group. (2025). Zero-Day Exploitation Trends 2024. GTIG Annual Report.

7. Beaumont, K. (2025, November 13). Commentary on Anthropic GTG-1002 Report. Mastodon.

8. Kirk, J. (2025, November). Analysis of Anthropic Threat Intelligence Report. Intel 471.

9. BleepingComputer. (2025, November 14). Anthropic claims of Claude AI-automated cyberattacks met with doubt.

10. Kasparov, G. (2010, February 11). The Chess Master and the Computer. *The New York Review of Books*.

11. Cowen, T. (2013). *Average is Over: Powering America Beyond the Age of the Great Stagnation*. Dutton.

12. Kasparov, G. (2017). *Deep Thinking: Where Machine Intelligence Ends and Human Creativity Begins*. PublicAffairs. (Chapter 11: "The Centaur")

13. Boyd, J. (1986). Patterns of Conflict. Unpublished briefing.

14. Coram, R. (2002). *Boyd: The Fighter Pilot Who Changed the Art of War*. Little, Brown and Company.

15. Borel, É. (1921). La théorie du jeu et les équations intégrales à noyau symétrique. *Comptes Rendus de l'Académie des Sciences*, 173, 1304-1308.

16. Roberson, B. (2006). The Colonel Blotto Game. *Economic Theory*, 29, 1-24.

17. Gupta, A., Schwartz, G., Langbort, C., Sastry, S.S., & Başar, T. (2014). A three-stage Colonel Blotto game with applications to cyberphysical security. *Proceedings of the American Control Conference*, 3820-3825.

18. Schwartz, G., Loiseau, P., & Sastry, S.S. (2014). The Heterogeneous Colonel Blotto Game. *International Conference on Network Games, Control and Optimization*.

19. Google Cloud. (2024, June). Lightning-fast decision-making: How AI can boost OODA loop impact on cybersecurity. *Google Cloud Blog*.

20. NVIDIA. (2025). CUDA Tile Programming Model. NVIDIA Developer Documentation. https://developer.nvidia.com/cuda/tile

21. Google Quantum AI. (2024, December 9). Meet Willow, our state-of-the-art quantum chip. *Google Blog*.

22. Google Quantum AI. (2025, October 22). The Quantum Echoes algorithm breakthrough. *Google Blog*.

23. Nature. (2024, December 9). Google Willow quantum chip achieves below-threshold error correction.

24. Harvard – James Mickens - https://www.usenix.org/system/files/1401_08-12_mickens.pdf

25. Ayzenshteyn, D., Weiss, R., & Mirsky, Y. (2025). Cloak, Honey, Trap: Proactive Defenses Against LLM Agents. *Proceedings of the 34th USENIX Security Symposium*.

Note on Methodology and Sources

This analysis integrates insights from multiple disciplines: cybersecurity incident analysis, military strategy (Boyd's OODA Loop), game theory (Colonel Blotto games, Stackelberg security games), organizational decision science, and human-AI teaming research. The authors have made deliberate efforts to:

- Verify factual claims against primary sources where possible
- Acknowledge skepticism from respected security practitioners regarding Anthropic's claims
- Distinguish between documented facts and analytical inferences
- Connect cybersecurity developments to established theoretical frameworks
- Avoid contributing to AI hype-cycle fear by grounding recommendations in established security principles

Several security researchers—including Kevin Beaumont, Jeremy Kirk (Intel 471), Jonathan Allon (Palo Alto Networks), and others—publicly questioned aspects of Anthropic's report, noting the absence of Indicators of Compromise (IOCs) and traditional forensic artifacts.

The Centaur paradigm framework (Section 6.9) draws on established research in human-computer teaming, including Garry Kasparov's work on Advanced Chess, Tyler Cowen's economic analysis in "Average is Over," and CSIRO's Collaborative Intelligence research program. The game-theoretic foundations (Section 5A) draw on Colonel Blotto game literature and Boyd's OODA Loop framework. The application of these frameworks to offensive and defensive cyber operations represents a novel contribution of this paper.

The forward-looking sections on GPU evolution (Appendix A) and quantum computing (Appendix B) draw on primary sources from NVIDIA and Google Quantum AI, with careful

distinction between current capabilities and projected timelines. These sections are grounded in technical reality rather than speculative threat inflation.

The absence of IOCs in Anthropic's disclosure does not necessarily invalidate their findings—the company possesses unique telemetry into Claude usage that external researchers cannot replicate. However, readers should consult primary sources and ongoing industry analysis as understanding of this event continues to evolve. This paper prioritizes analytical rigor and cross-disciplinary synthesis over uncritical acceptance of any single vendor's narrative.

Disclosure: The authors used automated editing tools to improve clarity and structure. All analysis, judgments, and conclusions are the authors' own, and any remaining errors are ours.