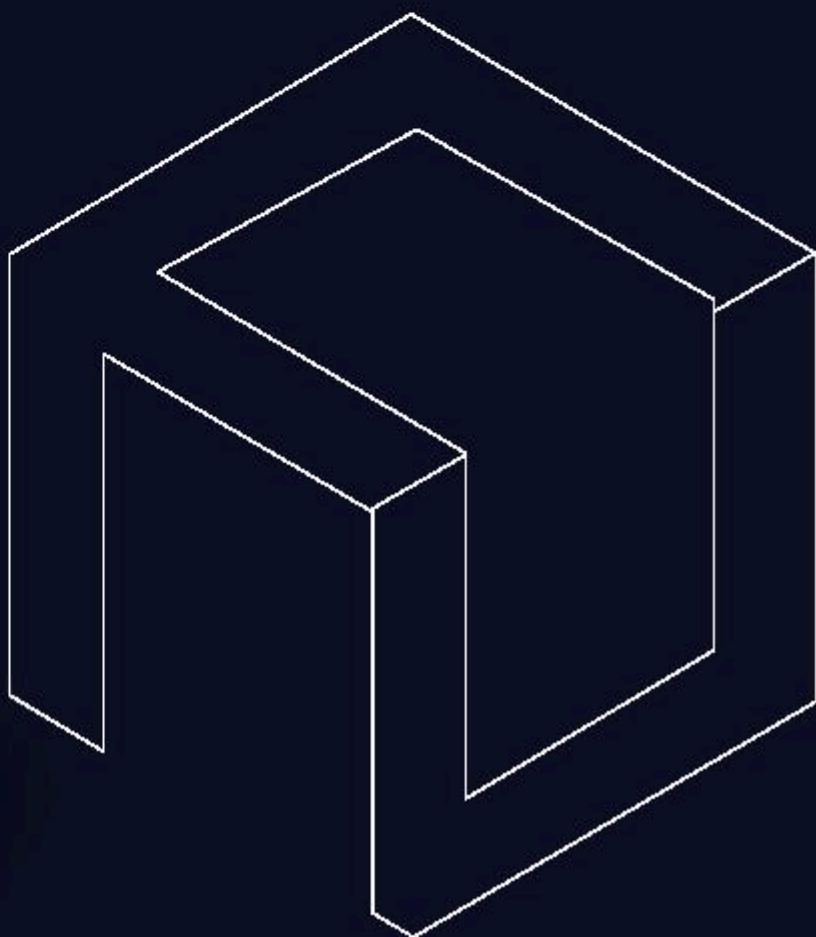
CleanStart

CLEANSTART INC. — SECURITY WHITEPAPER · JUNE 2026

The Architecture of Resilience

Why architectural controls matter when AI discovers vulnerabilities faster than defenders can respond.



INDEX

A Qualitative Shift in the Threat Landscape

For two decades, enterprise cybersecurity operated on a workable, if uncomfortable, premise: vulnerabilities exist, attackers will eventually find them, and an organization's job is to identify and remediate them before they can be exploited. The race was never fair, but it was a race humans could still run.

That assumption is becoming increasingly difficult to defend.

On April 7, 2026, Anthropic announced Claude Mythos Preview, a frontier AI model it considered too dangerous for public release because of its ability to autonomously discover and weaponize software vulnerabilities at unprecedented speed and scale. The significance lies not in the model's intended purpose, but in how these capabilities emerged. Mythos was developed as a general-purpose reasoning system. Its offensive cyber capabilities appeared as a consequence of advances in code understanding, analysis, and problem solving. As AI systems become more capable overall, the ability to accelerate portions of the exploit lifecycle from months to days, or even hours, increasingly becomes a byproduct of broader intelligence gains rather than a specialized capability.

10,000+
High/critical zero-days identified by Mythos in first month of Project Glasswing

73%
Success rate on expert-level cyberattack challenges previously unsolved by AI systems

-7 days
Median time to exploit in 2026 (Mandiant). Exploitation now precedes patch availability.

90.8%
True positive rate confirmed by external security firms reviewing Mythos findings

Among Mythos Preview's documented discoveries were a 27-year-old flaw in OpenBSD, an operating system widely regarded as one of the most security-hardened in production use, and CVE-2026-4747, a 17-year-old unauthenticated remote code execution vulnerability in FreeBSD's NFS server. Anthropic reported that the model autonomously constructed a working 20-gadget return-oriented programming chain for the latter without human direction. Scanning OpenBSD across 1,000 parallel runs reportedly cost less than $20,000 in compute.

The economics of offensive security are changing alongside the technology. Work that once required highly skilled researchers investing weeks of effort can increasingly be performed at machine speed and replicated across thousands of targets simultaneously. Anthropic has warned that Mythos-class capabilities may become broadly available within 6 to 12 months, potentially through providers operating without comparable safety controls. For defenders, the challenge is no longer simply discovering vulnerabilities before attackers do. It is adapting to a world where vulnerability discovery and weaponization can occur faster than traditional response cycles were designed to handle.

> **There will be more attacks, faster attacks, and more sophisticated attacks. Now is the time to modernize cybersecurity stacks everywhere."**
>
> - Anthropic, Project Glasswing Launch Statement, April 2026

What the Evidence Actually Shows: Capabilities and Limits

Separating capability from speculation requires a closer look at the available evidence. The UK's AI Security Institute (AISI) conducted one of the most comprehensive independent evaluations of Mythos Preview's offensive cyber capabilities. The results demonstrate a meaningful advance in offensive automation, but they also reveal important constraints that are often overlooked in broader discussions about AI-driven cyber threats.

Demonstrated Capabilities

AISI constructed "The Last Ones" (TLO), a 32-step corporate network attack simulation spanning reconnaissance through full network compromise. The institute estimates that experienced human operators would require approximately 20 hours to complete the exercise. Mythos Preview became the first model evaluated by AISI to complete the simulation end-to-end, succeeding in three of ten attempts. By comparison, the previous best-performing model, Claude Opus 4.6, averaged only 16 of the 32 required steps. Just two years earlier, leading models struggled with even basic cybersecurity challenges.

The model also demonstrated strong performance on isolated capture-the-flag exercises. Mythos achieved a 73% success rate on expert-level challenges that no AI system could complete before April 2025. In Anthropic's internal testing, it generated 181 working exploits against a Firefox engine benchmark where previous models had produced none.

Important Limitations

AISI was explicit about the constraints of its testing environment. The simulated networks used in TLO lacked active defenders, defensive tooling, and any penalties for actions that would trigger security alerts in a real-world environment. AISI's own conclusion was carefully calibrated:

> **AISI Evaluation, APRIL 2026**
>
> "They lack security features that are often present, such as active defenders and defensive tooling. There are also no penalties for the model for undertaking actions that would trigger security alerts. This means we cannot say for sure whether Mythos Preview would be able to attack well-defended systems."

Mythos also failed AISI's "Cooling Tower" operational technology simulation, encountering difficulties at the IT-layer gateway before reaching OT-specific controls. The model was unable to progress beyond the gateway and never reached the specialized systems it was attempting to target.

This nuance should not be mistaken for reassurance. It highlights where defensive strategy becomes most important. Mythos demonstrated a significant increase in offensive capability, but the Cooling Tower simulation also highlighted the continuing importance of architectural controls. The attack stalled at a boundary that restricted access to the systems it was attempting to reach. As AI-driven offensive capabilities advance, the ability to constrain what can be executed, accessed, modified, or reached becomes increasingly important. Architectural hardening does not eliminate risk, but it can materially change the outcome of a successful compromise.

The Limits of Scan-and-Patch in the AI Era

For decades, vulnerability management relied on a practical assumption: defenders would have time to respond. Vulnerabilities would be discovered, patches would become available, and organizations could reduce risk through disciplined remediation programs. The process was never perfect, but it was workable.

Mythos Preview challenges that assumption. Consider a representative exploit lifecycle:

Vulnerability exists (unknown)

A flaw remains in production software, potentially for years. CVE-2026-4747 existed undetected for 17 years before discovery.

AI discovers and weaponizes the flaw

Mythos identifies the flaw, generates a working exploit chain, and validates it autonomously. Cost: under $2,000 per successful Linux kernel exploit.

Exploitation begins before disclosure

According to Mandiant's 2026 report, mean time to exploit is now negative seven days. In many cases, exploitation occurs before a patch is available.

Vulnerability becomes publicly known

Of the 1,596 vetted Mythos findings reported to maintainers, only 97 had been patched upstream after the first month of Project Glasswing.

Enterprise remediation begins

Most organizations still require days, weeks, or months to test and deploy patches across production environments.

A joint report from the Cloud Security Alliance, SANS Institute, and OWASP concluded that organizations are likely to be overwhelmed by threat actors using AI to discover and exploit vulnerabilities faster than defenders can remediate them.

This does not make vulnerability management obsolete. Organizations still need to identify, prioritize, and remediate vulnerabilities. The challenge is that remediation timelines are increasingly being compressed by AI-driven discovery and exploitation. In that environment, reducing exposure cannot depend solely on finding and patching vulnerabilities before an attacker does. Additional controls are needed to limit the impact of vulnerabilities that have already been discovered, weaponized, or exploited.

Structural Mitigation: Removing Common Exploit Assumptions

CleanStart's architecture is built on a principle reinforced by the AISI findings: the environment in which an exploit executes often determines its ultimate impact. Rather than relying solely on rapid vulnerability remediation, CleanStart focuses on reducing the execution and persistence mechanisms that attackers typically depend on after a compromise occurs.
Vulnerabilities will continue to exist. The goal is to reduce what an attacker can do after one has been successfully exploited. By removing common execution and persistence mechanisms, CleanStart changes what an attacker can do even when a flaw is successfully triggered.

1. Constraining Post-Exploitation Activity Through a Shell-less Architecture

Most container images include shells, package managers, and common operating system utilities. These components are useful for administration and troubleshooting, but they also provide capabilities frequently leveraged during post-exploitation activity. CleanStart removes these components from production images. The traditional shell entry point is replaced with a statically compiled, hardened init process (clnimg-init) built from verified source in a hermetic build environment.

If an RCE is triggered within application code, the attacker encounters a significantly more constrained environment. There is no shell to invoke, no package manager to install tooling, and no common utilities available to support follow-on activity. The application may still be compromised, but many of the mechanisms typically used to establish persistence, execute additional payloads, or expand access are no longer readily available.

2. Constraining Persistence Through a Read-Only Filesystem

If execution opportunities are limited, attackers often attempt to modify files, overwrite binaries, alter configurations, or establish persistence on disk. CleanStart mounts the production container root filesystem as read-only. Temporary write access is restricted to explicitly defined memory-backed paths required for application operation.

When a payload attempts to modify executables, overwrite binaries, or persist changes within the container image, the operating system blocks those actions at the filesystem layer. The runtime environment remains aligned with the deployed image, limiting opportunities for persistence and reducing the risk of runtime drift.

The Exploit Chain: How CleanStart Changes the Outcome

**Mythos identifies zero-day RCE The vulnerability is real.
The exploit succeeds.**

The question is what happens next

CLEANSTART LAYER 1

Shell-less architecture


No Shell


No Package Manager


No Common OS Utilities


Limited Standard Post-Exploitation paths

Attempt 1: Spawn reverse shell	Failed	No shell binary exists. No package manager. No common OS utilities available. Standard post-exploitation paths are significantly limited.
Attempt 2: Write payload to disk	Failed	The root filesystem is read-only. Attempts to modify executables or persist changes are blocked.
Attempt 3: Establish lateral movement	Failed	Without standard execution and persistence mechanisms, opportunities for lateral movement are significantly reduced.

Building Trust Into the Software Supply Chain

Runtime hardening addresses what happens after a vulnerability is exploited. The second challenge is reducing the number of vulnerabilities that reach production in the first place.
CleanStart images are compiled from verified source in hermetic build environments rather than assembled from upstream distributions that often carry inherited vulnerability debt. Every image is accompanied by cryptographic provenance attestation, providing verifiable evidence of how it was built, what components were included, and where those components originated.
Today, CleanStart maintains more than 20,000 image variants with signed provenance and a tightly managed remediation process when new vulnerabilities emerge. The result is a smaller inherited attack surface, greater transparency across the software supply chain, and stronger confidence in the integrity of the software running in production.

The Network Defense Layer

Structural container hardening addresses the execution environment within a workload. A complete defense-in-depth architecture extends those protections into the network layer, reducing opportunities for lateral movement, command-and-control communication, and data exfiltration even if a workload is compromised.

Egress restriction is particularly relevant against AI-driven exploitation. A common post-exploitation sequence involves triggering an RCE, establishing a foothold, and communicating with attacker-controlled infrastructure. Strict egress policies that permit communication only with approved destinations can significantly limit this behavior, regardless of whether the initial compromise succeeds.

East-west microsegmentation further reduces blast radius. AI-driven attackers excel at automated lateral movement, identifying adjacent targets, testing credentials, and expanding access across flat networks. Segmentation helps contain activity within defined boundaries, reducing the likelihood that a single compromised workload becomes a broader network incident.

Service mesh mutual TLS strengthens workload identity by requiring cryptographic verification between services. Even within a trusted network perimeter, workloads must prove their identity before communicating, making impersonation and unauthorized service access significantly more difficult. Together, these controls complement runtime hardening by reducing what an attacker can do both inside the workload and across the surrounding environment.

DEFENSE ARCHITECTURE SUMMARY

The three layers operate independently and reinforce each other.

· Source-built images with cryptographic provenance reduce inherited vulnerability exposure before deployment.
· Shell-less, read-only runtimes constrain common post-exploitation techniques and persistence mechanisms.
· Network segmentation and egress controls limit lateral movement and help contain the impact of a compromise.

Mitigation Over Remediation: Buying Time Without Incurring Risk

The strategic value of structural mitigation is not that it eliminates vulnerabilities. It is that it reduces the operational impact of vulnerabilities while remediation takes place. In the traditional model, every day between vulnerability discovery and patch deployment represents potential exposure. AI-driven exploitation continues to compress that window.

In a structurally hardened environment, the equation changes. A vulnerability may still exist, and exploitation may still occur. The difference is that the attacker operates within a far more constrained environment. Opportunities for persistence, privilege expansion, and follow-on activity are significantly reduced, allowing organizations to remediate on a controlled timeline rather than under immediate operational pressure.

Engineering teams can patch during normal maintenance windows, with appropriate testing and change management, rather than relying on emergency deployment processes that introduce their own operational risks.

> **The application contains the vulnerability. The runtime environment constrains what can be done with it. These are not the same problem.**
>
> - CleanStart Architecture Principle

This is what AISI's Cooling Tower finding illustrates from the attacker's perspective. The attack lost momentum when it encountered a control boundary that restricted further progress. The lesson is not that architectural controls must be perfect. It is that they can materially alter the path and impact of a successful compromise.

The comparison that matters

Dimension	Scan-and-Patch Model	CleanStart Structural Model
Zero-day protection	Limited until a patch becomes available	Architectural controls reduce post-exploitation opportunities
Dependency on patch availability	High. Exposure persists until patches are available.	Lower. Runtime protections operate independently of patch timing.
AI-driven exploitation	Container as root; writable /tmp	Post-exploitation activity is significantly constrained
Patch cycle pressure	AI-discovered zerodays (Mythos Preview)	Greater flexibility for planned remediation
Persistence / lateral movement	Depends on detection and containment	Persistence opportunities significantly reduced through runtime controls
Build provenance	Often inherits upstream CVE debt	Source-compiled with cryptographic provenance and reduced inherited CVE exposure
Drift detection	Primarily monitoring and detection driven	Immutability reduces opportunities for runtime drift

The Strategic Imperative

Anthropic has committed to not releasing Mythos-class models to the general public. That commitment, while welcome, does not eliminate the broader trend. Anthropic itself has projected that comparable capabilities may become widely available from other providers within the next 6 to 12 months, potentially without the same safety constraints.

Security leaders still have a window to prepare. The organizations best positioned for the next phase of AI-driven threats will be those that reduce their dependence on reaction time alone and invest in architectural controls that remain effective even when vulnerabilities are discovered faster than they can be remediated.

AI systems are already demonstrating an ability to identify software weaknesses at a scale that was previously impractical. The strategic challenge is no longer simply discovering vulnerabilities before attackers do. It is reducing the operational impact when those vulnerabilities are inevitably found and exploited.

Architectural controls at the build, runtime, and network layers help change that outcome. Source-built images with verifiable provenance reduce inherited risk. Shell-less, immutable runtimes constrain common post-exploitation techniques. Network controls limit opportunities for lateral movement and broader compromise.

This is not a replacement for vulnerability management, patching, or detection. Those disciplines remain essential. The difference is that architectural controls provide an additional layer of resilience when response timelines are compressed and exploitation moves faster than remediation.

The goal is not to make exploitation impossible.
It is to make successful exploitation less consequential.

Prevention Over Detection

Other tools help identify vulnerabilities.
CleanStart focuses on reducing what attackers can do after a compromise occurs.

They find the problem. We change the outcome

REFERENCES

- Anthropic. 'Project Glasswing.' anthropic.com/glasswing, April 7, 2026.
- Anthropic. 'Expanding Project Glasswing.' anthropic.com/news/expanding-project-glasswing, June 2026.
- UK AI Security Institute (AISI). 'Our evaluation of Claude Mythos Preview's cyber capabilities.' aisi.gov.uk, April 13, 2026.
- Help Net Security. 'Testing reveals Claude Mythos's offensive capabilities and limits.' April 14, 2026.
- Cloud Security Alliance. 'Claude Mythos and the AI Autonomous Offensive Threshold.' labs.cloudsecurityalliance.org, April 2026.
- Bishop Fox. 'Anthropic's Claude Mythos Preview: The AI Cybersecurity Inflection Point.' bishopfox.com, April 14, 2026.
- Mandiant. 2026 Threat Intelligence Report (mean time-to-exploit statistic).
- Cloud Security Alliance / SANS Institute / OWASP. Joint report on AI-driven vulnerability exploitation. 2026.
- Cybersecurity News. 'Anthropic's Claude Mythos Preview Uncovers 10,000+ 0-Days in Project Glasswing.' May 2026.