INTELLIGENCE BRIEFING
Security Command Center
TLP:CLEAR
2026-04-30 06:30 UTC

Claude Mythos Preview Exposes a Structural Gap: AI Finds Vulnerabilities Faster Than Most Teams Can Fix Them

SECURITY ANALYSIS | HIGH | CVSS 7.5

SCC Item ID	SCC-STY-2026-0095
Type	Security Analysis
Severity	HIGH
CVSS Base Score	7.5
Affected Products	Anthropic Claude Mythos Preview, CrowdStrike Falcon Platform, CrowdStrike Falcon AIDR, CrowdStrike Falcon Data Security, CrowdStrike AgentWorks, Claude Code; downstream exposure across major OS and browser ecosystems
Discovery Source	Rss:T1 Threatintel

Executive Summary

Anthropic's Claude Mythos Preview has demonstrated autonomous discovery of vulnerabilities across major operating systems and browsers, including flaws that persisted undetected despite extensive automated testing. A coalition of 12 major technology vendors, Project Glasswing, is deploying Mythos defensively to close these gaps before adversaries develop equivalent capability. The strategic signal is structural: AI-assisted vulnerability discovery has outpaced the remediation pipelines of most enterprise security programs, creating an expanding window of exposure that threat actors, including state-sponsored groups, are working to exploit.

Technical Analysis

The Mythos Preview disclosure surfaces a problem that security teams have long anticipated but are now confronting at scale: the velocity of vulnerability discovery has structurally decoupled from the velocity of remediation. Mythos autonomously identified vulnerabilities across major operating systems and browser ecosystems, including legacy flaws that survived extensive automated test iterations without detection. These are not exotic, narrow-scope issues. The vulnerability classes involved - memory safety violations (CWE-119, CWE-787), use-after-free conditions (CWE-416), privilege escalation paths (CWE-269), and protection mechanism bypasses (CWE-693) - map directly to the attack chains adversaries use most reliably in production intrusions.

The MITRE ATT&CK techniques associated with this story (T1068 Exploitation for Privilege Escalation, T1190 Exploit Public-Facing Application, T1203 Exploitation for Client Execution, T1587.001 and T1587.004 Develop Capabilities: Malware and Exploits, T1588.006 Obtain Capabilities: Vulnerabilities, T1592 Gather Victim Host Information, T1650 Acquire Access) describe the complete adversarial pipeline: reconnaissance, capability development, initial access, and escalation. Mythos, used defensively, interrupts that pipeline by discovering exploitable conditions before adversaries can weaponize them. Used offensively, the same model class would compress the timeline from discovery to weaponization dramatically.

Project Glasswing, with CrowdStrike, AWS, Microsoft, and Apple among its 12 founding members, is deploying Mythos across their platforms to identify and remediate vulnerabilities at AI-generated speed, per Anthropic's official Project Glasswing documentation. The defensive logic is sound: if autonomous discovery is now tractable, the only meaningful defense is to discover and remediate faster than adversaries can develop equivalent tooling.

The gap that Glasswing cannot close internally is the one that lives in every enterprise security program outside the coalition: patching pipelines built for human-speed vulnerability disclosure, prioritization frameworks that assume months between discovery and weaponization, and engineering bandwidth that cannot absorb hundreds of critical findings in parallel. The remediation velocity gap is not a future risk. It is the current attack surface.

Action Checklist

1. Step 1: Assess exposure, inventory all systems running affected platforms: major operating systems (Windows, macOS, Linux distributions), browser engines (Chromium, WebKit, Firefox/Gecko), and media processing pipelines dependent on components like FFmpeg; any of these is in scope for the vulnerability classes Mythos identified
2. Step 2: Review patch pipeline capacity, audit your current mean time to remediate (MTTR) for critical vulnerabilities; if your pipeline cannot absorb a sustained high-volume disclosure event, identify the specific bottleneck (change control, testing bandwidth, engineering prioritization) and document the gap now
3. Step 3: Harden prioritization infrastructure, configure your vulnerability management platform to weight memory safety (CWE-119, CWE-787), use-after-free (CWE-416), privilege escalation (CWE-269), and protection mechanism bypass (CWE-693) at elevated priority; these are the classes Mythos is discovering at scale
4. Step 4: Update threat model, add AI-accelerated zero-day discovery as a named threat scenario in your threat register; map it to T1068, T1190, T1203, T1587.001, T1587.004, and T1588.006
5. Step 5: Evaluate compensating controls for legacy codebases, audit high-exposure legacy components and evaluate memory-safe reimplementation, sandboxing, or network isolation as compensating controls where patching is not immediately feasible
6. Step 6: Monitor Project Glasswing disclosures via Anthropic's official security advisory channel (glasswing.anthropic.com) and CrowdStrike's public disclosure feeds; establish a standing intake process for coordinated releases
7. Step 7: Brief leadership with remediation gap framing, the board-level risk is not that Mythos exists; it is that most organizations cannot remediate at the speed AI discovery generates findings; quantify your current MTTR against a high-volume disclosure scenario and present the gap as a resource and process risk, not a technology risk

IR / Forensic Enrichment

Triage Priority	URGENT
Escalation Criteria	Escalate to incident commander and notify legal/compliance if any system in the Chromium, WebKit, FFmpeg, or affected OS inventory shows evidence of exploitation (unusual child processes from browser or media-processing parent processes, outbound connections from sandboxed components, or log entries matching Glasswing-disclosed URI or payload patterns) prior to patch deployment, particularly if the affected system processes PII, PHI, or is subject to PCI-DSS scope, as this constitutes a potential breach notification trigger.
Recovery Notes	After patching affected OS, browser engine, and FFmpeg components, verify patch integrity by recomputing binary hashes against vendor-published checksums and confirm no persistence mechanisms were installed during the pre-patch exposure window by auditing scheduled tasks, cron jobs, startup registry keys (HKLM\SOFTWARE\Microsoft\Windows\CurrentVersion\Run), and /etc/cron.d entries on all affected hosts. Monitor CrowdStrike Falcon AIDR and Glasswing feeds continuously for 90 days post-patch, as Mythos-class discovery is ongoing and a patched component may have sibling vulnerabilities disclosed in subsequent advisory batches. Retain pre-patch memory snapshots and log archives for a minimum of 12 months to support any retrospective forensic analysis if a Mythos-discovered CVE is later confirmed actively exploited by a state-sponsored actor against your industry vertical.
Forensic Artifacts	Sysmon Event ID 1 (Process Creation) logs on Windows endpoints filtering for browser engine processes (chrome.exe, msedge.exe, firefox.exe) spawning unexpected child processes (cmd.exe, powershell.exe, wscript.exe, or any executable in %TEMP% or %APPDATA%) — indicative of T1203 (Exploitation for Client Execution) via a Mythos-identified WebKit or Chromium memory safety flaw | FFmpeg process memory dumps and core files from Linux media processing hosts (located in /var/crash/, /tmp/, or the working directory of the ffmpeg process) — a Mythos-class CWE-787 out-of-bounds write in libavcodec would produce a crash artifact recoverable with `volatility3 -f coredump.bin linux.pstree` that can confirm exploitation attempt | Web server and WAF access logs for URI patterns associated with Glasswing-disclosed CVEs against public-facing applications — filter on HTTP 4xx/5xx response codes with unusually large request bodies or malformed media MIME types (video/mp4, image/webp) submitted to endpoints that invoke FFmpeg or browser rendering pipelines, consistent with T1190 exploitation | Linux kernel audit logs (`/var/log/audit/audit.log`) filtered for `syscall=execve` events preceded by `type=AVC` (SELinux denial) or `type=SECCOMP` entries from FFmpeg or browser sandbox processes — a Mythos-class CWE-269 privilege escalation attempt would generate an AVC denial before or after the exploitation attempt if SELinux is in enforcing mode | Windows Security Event Log Event ID 4688 (Process Creation with command line) and Event ID 4697 (Service Installed) on hosts running Chromium-based browsers or FFmpeg pipelines — filter for new service installations or processes launched from browser working directories within 24 hours of a Glasswing advisory publication, as state-sponsored actors (APT41, Lazarus Group) have historically weaponized browser engine flaws within hours of public disclosure

Per-Action IR Details

Step 1: Assess exposure — inventory all systems running affected platforms: major operating systems (Windows, macOS, Linux distributions), browser engines (Chromium, WebKit, Firefox/Gecko), and FFmpeg-dependent media processing pipelines; any of these is in scope for the vulnerability classes Mythos

identified

NIST Phase: Preparation

Reference: NIST 800-61r3 §2 — Preparation: Establishing IR capability and asset visibility before adverse events occur

Controls: NIST SI-5 (Security Alerts, Advisories, and Directives) — track Project Glasswing and CrowdStrike disclosure feeds as authoritative advisory sources, NIST RA-2 (Security Categorization) — categorize systems hosting Chromium/WebKit/FFmpeg pipelines at elevated risk given Mythos-identified memory safety flaw classes, CIS 1.1 (Establish and Maintain Detailed Enterprise Asset Inventory) — enumerate all assets running affected OS kernels, browser engines, and FFmpeg-linked media processing components, CIS 2.1 (Establish and Maintain a Software Inventory) — identify all software packages with FFmpeg linkage (ffmpeg, libavcodec, libavformat) and browser engine versions across the fleet

Compensating: Run `osquery` with the query `SELECT name, version, source FROM deb_packages WHERE name LIKE '%ffmpeg%' OR name LIKE '%chromium%' OR name LIKE '%webkit%';` on Linux hosts; on Windows, use `Get-WmiObject Win32_Product | Where-Object {$_.Name -match 'ffmpeg|chrome|webkit'}` via PowerShell remoting. Produce a CSV; flag any host running FFmpeg < current stable or Chromium below the latest Glasswing-patched build. A 2-person team can complete fleet enumeration in one sprint using these free tools.

Evidence: Before conducting the inventory, snapshot current installed package versions and build hashes for FFmpeg binaries (run `sha256sum /usr/bin/ffmpeg` or `Get-FileHash`) and browser executable hashes. This baseline is required to prove pre-patch state if a Mythos-class zero-day is later confirmed exploited against your environment — without it, you cannot establish the vulnerable window for incident scoping under NIST 800-61r3 §3.2 timeline reconstruction.

Step 2: Review patch pipeline capacity — audit your current mean time to remediate (MTTR) for critical vulnerabilities; if your pipeline cannot absorb a sustained high-volume disclosure event, identify the specific bottleneck (change control, testing bandwidth, engineering prioritization) and document the gap now

NIST Phase: Preparation

Reference: NIST 800-61r3 §2 — Preparation: Identifying resource and process gaps before a high-tempo disclosure event overwhelms response capacity

Controls: NIST SI-2 (Flaw Remediation) — formalize patching SLAs for the CWE-119, CWE-787, CWE-416, and CWE-269 classes Mythos is producing; document current MTTR against those SLAs, NIST IR-8 (Incident Response Plan) — update the IR plan to include a sustained high-volume zero-day disclosure scenario as a named contingency, referencing Project Glasswing as the trigger source, CIS 7.1 (Establish and Maintain a Vulnerability Management Process) — document the specific process step (change control gate, QA testing window, or engineering queue) that caps throughput, CIS 7.2 (Establish and Maintain a Remediation Process) — establish a risk-based remediation track with an expedited lane for Mythos-class memory safety findings (CVSS ≥ 7.0, CWE-119/787/416)

Compensating: Export your last 90 days of vulnerability tickets from whatever tracker you use (Jira, GitHub Issues, spreadsheet) and compute MTTR per severity tier using a simple Python script or Excel pivot. If MTTR for criticals exceeds 30 days, that is your documented gap. For change control bottlenecks, draft a pre-authorized emergency change template scoped specifically to OS security patches and browser engine updates so the approval step does not add days when Glasswing disclosures accelerate.

Evidence: Preserve your current MTTR metrics, open critical vulnerability backlog count, and change control queue depth as a dated snapshot before any remediation sprint begins. This establishes the pre-event baseline required to demonstrate reasonable response effort to regulators or cyber insurance reviewers if a Mythos-discovered zero-day is later weaponized against your environment during the disclosure window.

Step 3: Harden prioritization infrastructure — configure your vulnerability management platform to weight memory safety (CWE-119, CWE-787), use-after-free (CWE-416), privilege escalation (CWE-269), and protection mechanism bypass (CWE-693) at elevated priority; these are the classes Mythos is discovering at scale

NIST Phase: Preparation

Reference: NIST 800-61r3 §2 — Preparation: Tuning detection and triage tools to recognize the specific vulnerability classes expected from AI-accelerated discovery

Controls: NIST RA-3 (Risk Assessment) — formally reassess risk ratings for all open findings in CWE-119, CWE-787, CWE-416, CWE-269, and CWE-693 classes given Mythos evidence that these classes carry decades-long latency in legacy codebases, NIST SI-2 (Flaw Remediation) — update patch SLAs to assign automatic critical-tier classification to any incoming advisory mapped to these five CWE classes regardless of CVSS base score alone, CIS 7.1 (Establish and Maintain a Vulnerability Management Process) — add CWE-class weighting rules to the documented prioritization criteria so Mythos-class findings are not deprioritized by scanners that score them below 9.0, CIS 7.2 (Establish and Maintain a Remediation Process) — create a named remediation track 'AI-Discovery-Class' for findings in these CWE categories originating from Project Glasswing or CrowdStrike AIDR feeds

Compensating: In the absence of an enterprise vuln management platform, maintain a YAML or JSON CWE priority list and use `grep` or `jq` to filter NVD JSON feeds (`https://nvd.nist.gov/vuln/data-feeds`) for incoming CVEs matching CWE-119, CWE-787, CWE-416, CWE-269, CWE-693. Write a daily cron job that pulls the NVD feed, filters on these CWEs, and emails the team — this is achievable in under 50 lines of Python with no budget.

Evidence: Query your vulnerability scanner's historical findings for all open CWE-119, CWE-787, CWE-416, CWE-269, and CWE-693 findings. Export this list with discovery date, asset, and current remediation status before reconfiguring prioritization weights — this pre-change snapshot is your evidence that the old configuration was inadequate if a subsequent exploit targets one of these classes in your environment.

Step 4: Update threat model — add AI-accelerated zero-day discovery as a named threat scenario in your threat register; map it to T1068, T1190, T1203, T1587.001, T1587.004, and T1588.006; assign named threat actors (China, Iran, North Korea, Russia state-sponsored groups) as the primary adversary tier

NIST Phase: Preparation

Reference: NIST 800-61r3 §2 — Preparation: Maintaining current threat intelligence and ensuring the IR plan accounts for identified adversary capabilities

Controls: NIST RA-3 (Risk Assessment) — formally incorporate AI-accelerated zero-day discovery as a threat source in the organizational risk assessment, citing Mythos capability and Project Glasswing scope as evidence, NIST IR-8 (Incident Response Plan) — add a named scenario 'State-Sponsored AI-Assisted Zero-Day Exploitation' mapping to T1068 (Exploitation for Privilege Escalation), T1190 (Exploit Public-Facing Application), T1203 (Exploitation for Client Execution), T1587.001 (Develop Capabilities: Malware), T1587.004 (Develop Capabilities: Exploits), and T1588.006 (Obtain Capabilities: Vulnerabilities), NIST SI-5 (Security Alerts, Advisories, and Directives) — designate CISA KEV, Project Glasswing advisories, and CrowdStrike Falcon intelligence feeds as mandatory intake sources for this threat scenario, CIS 7.1 (Establish and Maintain a Vulnerability Management Process) — tag all vulnerability findings against affected platforms with state-sponsored threat actor relevance to support prioritization when Glasswing disclosures reference geopolitical attribution

Compensating: Use MITRE ATT&CK Navigator (free, browser-based) to create a layer file mapping the six techniques above to your current detection coverage. Color-code gaps in red. Export the SVG and attach it to your threat register entry. This gives leadership a visual representation of detection gaps against the specific adversary tier (APT10, APT41, Lazarus Group, Sandworm) most likely to weaponize Mythos-class findings first.

Evidence: Before updating the threat model, pull current CISA KEV entries and cross-reference against your asset inventory for any already-exploited CVEs in the Chromium, WebKit, Linux kernel, or FFmpeg categories — these are the precedent cases that validate the threat model update and demonstrate to auditors that the new scenario is grounded in observed adversary behavior, not speculation.

Step 5: Evaluate compensating controls for legacy codebases — the 27-year OpenBSD flaw and 16-year FFmpeg bug illustrate that legacy code carries undiscovered critical risk; audit high-exposure legacy components and evaluate memory-safe reimplementation, sandboxing, or network isolation as compensating controls where patching is not immediately feasible

NIST Phase: Containment

Reference: NIST 800-61r3 §3.3 — Containment Strategy: Selecting containment strategy based on damage potential and need to preserve evidence while reducing exposure

Controls: NIST SI-7 (Software, Firmware, and Information Integrity) — deploy integrity verification on FFmpeg binaries and legacy media processing pipeline components to detect unauthorized modification consistent with post-exploitation

persistence, NIST SC-39 (Process Isolation) — enforce process isolation for FFmpeg-linked media processing pipelines; on Linux this means seccomp-BPF profiles or bubblewrap sandboxing; on Windows, use Windows Sandbox or AppContainer for legacy processing jobs, NIST CM-7 (Least Functionality) — disable or network-isolate legacy components that cannot be patched on a timeline consistent with Glasswing disclosure velocity, CIS 4.4 (Implement and Manage a Firewall on Servers) — apply host-based firewall rules blocking outbound connections from FFmpeg processing processes to internet-routable addresses; use `iptables -A OUTPUT -m owner --uid-owner ffmpeg -j DROP` or Windows Firewall outbound rules scoped to the service account

Compensating: For FFmpeg-dependent pipelines that cannot be immediately patched, run them inside `firejail --seccomp --net=none ffmpeg [args]` on Linux — this costs nothing and eliminates network-reachability for a Mythos-class memory safety exploit in libavcodec or libavformat. For Windows legacy components, enable Windows Defender Exploit Guard with Attack Surface Reduction rules targeting untrusted process execution. Document each compensating control with a target patch date so it does not become permanent.

Evidence: Before implementing sandboxing or isolation, capture a memory snapshot and full filesystem listing of each legacy component host using `volatility3` (free) or `winpmem` for Windows memory acquisition. For FFmpeg hosts specifically, record all active network connections (`ss -tulnp` / `netstat -ano`) and running process trees (`ps auxf` / `Get-Process | Select-Object *`). These pre-containment artifacts establish whether exploitation preceded your containment action — critical for determining if the 16-year FFmpeg class flaw was already weaponized against your environment before you isolated it.

Step 6: Monitor Project Glasswing disclosures — track Anthropic's glasswing.anthropic.com channel and CrowdStrike's disclosure feeds for coordinated vulnerability releases; establish a standing intake process so your team is not responding ad hoc when disclosures accelerate

NIST Phase: Detection Analysis

Reference: NIST 800-61r3 §3.2 — Detection and Analysis: Monitoring for adverse events and maintaining current threat intelligence intake

Controls: NIST SI-5 (Security Alerts, Advisories, and Directives) — formally designate glasswing.anthropic.com and CrowdStrike Falcon intelligence feeds as named external advisory sources in the SI-5 implementation; assign a named owner responsible for daily intake, NIST IR-6 (Incident Reporting) — establish internal reporting SLAs: any Glasswing or CrowdStrike disclosure affecting in-scope platforms must be triaged and assigned within 4 hours of publication, NIST AU-6 (Audit Record Review, Analysis, and Reporting) — correlate incoming Glasswing disclosures against existing log data for affected components to determine if exploitation preceded the disclosure, CIS 7.1 (Establish and Maintain a Vulnerability Management Process) — embed the Glasswing intake step as a named procedure in the vulnerability management process document, including escalation path when a disclosure matches an unpatched asset in your inventory

Compensating: Use a free RSS-to-email or RSS-to-Slack bridge (RSSHub, Feedly free tier, or a Python `feedparser` cron job) to monitor the Glasswing feed and CrowdStrike's public blog for disclosure announcements. Write a Sigma rule that fires on ingestion of any advisory containing CWE-119, CWE-787, CWE-416, CWE-269, or CWE-693 and cross-references your asset inventory CSV — this gives a 2-person team an automated first-pass triage without a SIEM.

Evidence: When a new Glasswing disclosure is received, immediately query web server access logs, WAF logs, and reverse proxy logs for URI patterns or user-agent strings associated with exploitation attempts against the disclosed vulnerability class before the advisory was public — adversaries with equivalent AI capability may have weaponized the same flaw ahead of disclosure. For browser-engine disclosures, check endpoint logs for unusual child processes spawned by browser processes (Sysmon Event ID 1, parent process = chrome.exe or firefox.exe, child = cmd.exe, powershell.exe, or any process from a temp directory).

Step 7: Brief leadership with remediation gap framing — the board-level risk is not that Mythos exists; it is that most organizations cannot remediate at the speed AI discovery generates findings; quantify your current MTTR against a high-volume disclosure scenario and present the gap as a resource and process risk, not a technology risk

NIST Phase: Post Incident

Reference: NIST 800-61r3 §4 — Post-Incident Activity: Lessons learned, capability gap documentation, and organizational communication to drive process improvement

Controls: NIST IR-4 (Incident Handling) — document the organizational capability gap between current patch throughput and projected Glasswing/Mythos disclosure velocity as a formal IR capability deficiency requiring resource allocation, NIST IR-8 (Incident Response Plan) — update the IR plan with a high-volume disclosure scenario including explicit resource requirements (FTE hours per CVE, change control time, QA windows) to make the gap quantifiable for leadership, NIST RA-3 (Risk Assessment) — present MTTR gap as a risk acceptance decision requiring explicit leadership sign-off; the risk of unpatched CWE-119/787/416 flaws during the window between Glasswing disclosure and your remediation completion is a named, quantified risk, CIS 7.2 (Establish and Maintain a Remediation Process) — use current MTTR data to project the patch backlog accumulation rate if Glasswing releases 10, 50, or 100 advisories per month; present this as a resource planning table, not a qualitative concern

Compensating: Build the leadership brief in a spreadsheet: column A = historical MTTR for criticals (days), column B = estimated Glasswing disclosures per month (conservative: 10, moderate: 50, aggressive: 100), column C = projected open critical count at 30/60/90 days under each scenario. This costs nothing and converts an abstract AI risk narrative into a concrete backlog and resource ask that a CFO or CTO can act on. Attach the MITRE ATT&CK Navigator gap map from Step 4 as the detection coverage appendix.

Evidence: Compile the following dated artifacts as supporting evidence for the leadership brief: (1) current open critical/high vulnerability count with age distribution, (2) last 90-day MTTR by severity tier, (3) number of assets running unpatched Chromium, WebKit, FFmpeg, or OS kernels in the affected window, and (4) any CISA KEV entries from the past 12 months that affected your fleet. These figures transform the brief from a theoretical risk discussion into a documented, auditable gap assessment — and serve as the baseline against which future remediation investment will be measured.

Detection Guidance

There are no direct IOCs associated with Mythos or Project Glasswing; this story is a structural risk disclosure, not an active campaign with published indicators. Detection focus should shift to two areas.

First, watch for adversary exploitation of the vulnerability classes Mythos has identified before patches are available. In your SIEM and EDR telemetry, prioritize alerting on: privilege escalation chains from unprivileged browser or media-processing contexts (signals CWE-269 exploitation via CWE-119 or CWE-416 primitives); unexpected process spawning from browser renderer processes or media handlers; memory corruption signals in OS kernel logs; and anomalous crash telemetry in browser crash reporting pipelines, which may indicate probing activity before a stable exploit is developed.

Second, hunt for signs of AI-augmented adversary reconnaissance aligned with T1592 (Gather Victim Host Information). Unusually high-volume, structured enumeration of software versions, patch levels, or exposed service banners, particularly from sources with no prior organizational relationship, may indicate automated host profiling consistent with AI-assisted targeting pipelines.

Policy gap audit: review whether your vulnerability disclosure intake process has a defined SLA for memory-safety class vulnerabilities. If it does not, that absence is itself a detectable gap. Run a tabletop against a 50-CVE simultaneous disclosure scenario and document where the pipeline breaks.

Indicators of Compromise

Type	Value	Context	Confidence
URL	`Pending — refer to Anthropic Project Glasswing (anthropic.com/glasswing) for published vulnerability disclosures`	Anthropic's Project Glasswing documentation is the primary source for coordinated vulnerability disclosures resulting from Mythos discovery; specific CVE identifiers and technical indicators will be published through that channel	LOW
URL	`Pending — refer to CrowdStrike blog (crowdstrike.com/blog) for Falcon platform-specific detection guidance tied to Glasswing disclosures`	CrowdStrike is a founding Glasswing member deploying Mythos across Falcon AIDR, Falcon Data Security, and AgentWorks; platform-specific detection content will be published through their standard advisory channel	LOW

Framework Mappings

MITRE-ATTACK

- **T1650** — Acquire Access
- **T1587.004** — Exploits
- **T1068** — Exploitation for Privilege Escalation
- **T1587.001** — Malware
- **T1588.006** — Vulnerabilities
- **T1203** — Exploitation for Client Execution
- **T1190** — Exploit Public-Facing Application
- **T1592** — Gather Victim Host Information

NIST-800-53R5

- **AC-6** — Least Privilege
- **SC-7** — Boundary Protection
- **SI-2** — Flaw Remediation
- **SI-3** — Malicious Code Protection
- **SI-4** — System Monitoring
- **CA-8** — Penetration Testing
- **RA-5** — Vulnerability Monitoring and Scanning
- **SI-7** — Software, Firmware, and Information Integrity
- **SI-16** — Memory Protection
- **SI-10** — Information Input Validation
- **IR-5** — Incident Monitoring

CIS-V8

- **16.10** — Apply Secure Design Principles in Application Architectures
- **5.4** — Restrict Administrator Privileges to Dedicated Administrator Accounts

- **6.8** — Define and Maintain Role-Based Access Control
- **7.3** — Perform Automated Operating System Patch Management
- **7.4** — Perform Automated Application Patch Management

OWASP-TOP10-2021

- **A03:2021** — Injection
- **A01:2021** — Broken Access Control

ISO-27001-2022

- **A.8.8** — Management of technical vulnerabilities
- **A.5.21** — Managing information security in the ICT supply chain
- **A.5.23** — Information security for use of cloud services

SOC2-TSC

- **CC9.2** — Manages risks associated with vendors and business partners
- **CC6.3** — Authorizes, modifies, or removes access

NIST-CSF-2

- **DE.AE-08** — Incidents are declared when adverse events meet the defined incident criteria

MITRE ATT&CK Mapping

Technique ID	Technique Name	Tactic
T1650	Acquire Access	Resource-Development
T1587.004	Exploits	Resource-Development
T1068	Exploitation for Privilege Escalation	Privilege-Escalation
T1587.001	Malware	Resource-Development
T1588.006	Vulnerabilities	Resource-Development
T1203	Exploitation for Client Execution	Execution
T1190	Exploit Public-Facing Application	Initial-Access
T1592	Gather Victim Host Information	Reconnaissance

Sources

Source	URL	Tier
Blog	https://www.crowdstrike.com/en-us/blog/crowdstrike-founding-member-...	T3
	https://thehackernews.com/2026/04/mythos-changed-math-on-vulnerabil...	T3

Source	URL	Tier
	https://www.anthropic.com/glasswing	T1
	https://www.bbc.com/news/articles/crk1py1jgzko	T2
CrowdStrike Secures Anthropic AI Innovation | George Kurtz posted ...	https://www.linkedin.com/posts/georgekurtz_anthropic-claude-mythos-...	T3

Generated 2026-04-30 06:30 UTC by TJS Security Command Center