INTELLIGENCE BRIEFING
Security Command Center
TLP:CLEAR
2026-05-02 06:44 UTC

AI-Accelerated Vulnerability Discovery Collapses Defender Patch Windows to Near-Zero

SECURITY ANALYSIS | HIGH | CVSS 7.5

SCC Item ID	SCC-STY-2026-0104
Type	Security Analysis
Severity	HIGH
CVSS Base Score	7.5
Affected Products	Broad ecosystem, no single product; context references CrowdStrike Falcon, OpenAI Project Glasswing, Anthropic Claude (Mythos); OpenBSD (unspecified 27-year-old flaw), Mozilla Firefox (181 exploits generated in internal testing)
Discovery Source	Rss:T1 Threatintel

Executive Summary

AI-assisted vulnerability discovery systems can now identify and weaponize software vulnerabilities in hours, compressing the window between public disclosure and active exploitation from roughly two years in 2019 to under one day according to secondary reporting in 2026 (source confidence: medium; primary disclosure not yet available). According to T3 reporting, vulnerability discovery testing identified thousands of high- and critical-severity flaws, including a 27-year-old OpenBSD vulnerability, and generated working exploit code for multiple platforms. This signals a structural shift in the threat landscape: organizations still operating on weekly or monthly patch cycles are now functionally undefended against AI-accelerated adversaries.

Technical Analysis

The core problem this story surfaces is a velocity mismatch. Defenders have long operated on the assumption that the discover-to-weaponize timeline provides a meaningful response window. That assumption is breaking down. According to secondary reporting, AI-assisted vulnerability discovery and exploit generation has identified thousands of high- and critical-severity vulnerabilities and produced working exploits, including a flaw in OpenBSD that had persisted undetected for 27 years. Mean time from disclosure to exploitation, reported at approximately 2.3 years in 2019, has compressed to under one day in 2026 according to T2/T3 outlets; these figures should be treated as medium-confidence indicators of direction rather than precision metrics until primary technical disclosure is available.

The MITRE ATT&CK footprint associated with this threat model is wide. AI-assisted reconnaissance (T1595) and vulnerability scanning (T1046) lower the bar for initial access. Exploit development (T1587.004) and

acquiring exploits from the market (T1588.006) are now augmented by AI tooling. Once access is established, techniques including exploitation of public-facing applications (T1190), client-side exploitation (T1203), phishing for information (T1598), privilege escalation (T1068), and credential use (T1078) complete the chain. Threat actor groups already operating at speed, including APT28 (FANCY BEAR), stand to gain significant advantage from AI-accelerated offensive tooling.

SANS and the Cloud Security Alliance have flagged that legacy patch cadences (weekly or monthly cycles) and current security team staffing are structurally mismatched to this acceleration. CrowdStrike's Falcon platform is referenced in the context of AI-powered vulnerability discovery on the defender side, and Anthropic's Project Glasswing represents industry efforts to close the defender gap. The Anthropic Glasswing page (T1 source) is the highest-confidence anchor for the defensive-AI framing; specific internal testing figures (exploit counts, vulnerability counts) originate from T2/T3 outlets and have not been confirmed in primary technical disclosure as of this writing.

The structural implication for security operations: patch prioritization models that rely on CVSS scores and disclosure dates are no longer sufficient. Organizations need risk-velocity scoring that factors in AI-assisted exploitation probability, not just static severity ratings. Detection engineering must shift toward behavioral patterns rather than signature-based indicators, because AI-generated exploit chains may not match known signatures at all.

Action Checklist

1. Assess patch velocity exposure: audit your current mean time to patch for critical and high-severity CVEs; compare against the sub-24-hour exploitation window now documented for AI-accelerated adversaries; identify your longest-tail unpatched assets
2. Prioritize by exploitation probability, not CVSS alone: incorporate EPSS scores and threat-intelligence enrichment into your patch prioritization workflow; CVSS base scores do not reflect AI-accelerated weaponization speed
3. Review EDR and behavioral detection coverage: verify that CrowdStrike Falcon or your equivalent EDR is tuned for post-exploitation behavioral patterns (T1068, T1078, T1059, T1203) rather than relying solely on known exploit signatures that AI-generated exploits may bypass
4. Update your threat model for AI-assisted adversaries: add AI-accelerated vulnerability discovery and exploit generation as explicit threat scenarios in your threat register; map to T1587.004, T1588.006, T1595, and T1046; assess which threat actors in your profile are likely to adopt these capabilities
5. Evaluate compensating controls for legacy systems: the 27-year-old OpenBSD flaw discovery signals that AI tooling will surface long-dormant vulnerabilities in legacy and open-source components; audit your software bill of materials (SBOM) for unmaintained or under-reviewed components
6. Brief leadership on the velocity problem: frame the business risk as a structural mismatch between current operational rhythms and attacker speed; this is a resource and process question, not only a technical one; present options including continuous patching pipelines, vulnerability management tooling upgrades, or managed detection services
7. Monitor for primary-source confirmation: the specific testing figures (exploit counts, vulnerability counts) cited in secondary reporting should be cross-referenced against Anthropic's official Glasswing publications and SANS/CSA technical research before citing these metrics in formal risk documentation

IR / Forensic Enrichment

Triage Priority	URGENT
Escalation Criteria	Escalate immediately if CrowdStrike Falcon or Sysmon detects firefox.exe or any OpenBSD/BSD-derived service process spawning unexpected child processes (T1203), if EPSS scores for any open CVE in your environment exceed 0.30 within 24 hours of NVD publication (indicating AI-accelerated weaponization), or if CISA adds a CVE affecting Firefox or BSD-derived components to the KEV catalog — any of these conditions indicates the sub-24-hour exploitation window may already be closing on an asset in your environment.
Recovery Notes	After patching Firefox or any BSD-derived legacy component identified through the SBOM audit, verify patch application with 'firefox --version' or the relevant package manager query and re-scan with Grype to confirm the vulnerable component version is no longer present. Monitor Sysmon Event ID 1 and 3 logs on patched systems for 72 hours post-patch, specifically watching for any process-tree anomalies consistent with T1203 that could indicate exploitation occurred prior to patching. Because AI-generated exploits in the Mythos class are documented to produce working shellcode without prior public proof-of-concept, treat any anomalous process execution on previously vulnerable systems as a potential pre-patch compromise and preserve memory images with WinPmem or LiME before returning systems to full production.
Forensic Artifacts	Firefox process tree logs — Sysmon Event ID 1 (Process Create) with ParentProcessName = firefox.exe; any child process outside firefox.exe's normal execution tree (updater.exe, crashreporter.exe) is a high-fidelity indicator of successful exploitation from the Mythos-class Firefox exploit family | OpenBSD/BSD system package manager history — 'pkg_info -a' output and /var/log/messages entries timestamped around any network-facing service anomaly; the 27-year-old OpenBSD flaw class would likely manifest as an unexpected privilege escalation event in auth.log or a process executing as root from a non-root service context | Network connection logs for post-exploitation C2 — Sysmon Event ID 3 (Network Connection) on firefox.exe or the vulnerable BSD service process making outbound connections to non-browser destinations immediately following the exploitation window; capture with 'tcpdump -i any -w capture.pcap' for offline analysis if no SIEM is available | CrowdStrike Falcon detection history export — query Falcon console for T1068 (Exploitation for Privilege Escalation) and T1203 (Exploitation for Client Execution) detections in the 30 days prior to the patch date on affected Firefox and legacy system assets; this establishes whether exploitation preceded remediation | SBOM delta report — Grype scan output before and after patching, retained as dated JSON artifacts; for AI-discovered legacy flaws in unmaintained components, the pre-patch scan is the only forensic record that the vulnerable version was present, which is required to assess blast radius if a compromise is later suspected

Per-Action IR Details

Assess patch velocity exposure — audit your current mean time to patch for critical and high-severity CVEs; compare against the sub-24-hour exploitation window now documented for AI-accelerated adversaries; identify your longest-tail unpatched assets

NIST Phase: Preparation

Reference: NIST 800-61r3 §2 — Preparation: Establishing IR Capability and Measuring Operational Readiness

Controls: NIST SI-2 (Flaw Remediation), NIST RA-3 (Risk Assessment), CIS 7.1 (Establish and Maintain a Vulnerability Management Process), CIS 7.2 (Establish and Maintain a Remediation Process)

Compensating: Run 'wmic qfe list full' (Windows) or 'rpm -qa --last | head -50' (Linux) to extract patch installation timestamps per host. Export to CSV and compute delta between CVE NVD publication date and patch application date per asset. For network devices and legacy OpenBSD or BSD-derived systems — which the Anthropic Mythos research specifically flagged as harboring 27-year-old unpatched flaws — manually review /var/log/messages and package manager history logs. A 2-person team can script this in Python using the NVD JSON feed (nvd.nist.gov/developers/vulnerabilities) and an internal asset list to flag any asset where MTTP exceeds 24 hours for CVSS 7.0+ CVEs.

Evidence: Before this step: capture a point-in-time snapshot of your vulnerability scanner output (Nessus, OpenVAS, or equivalent) as a baseline artifact. Export the full asset inventory including OS version, patch level, and last-scan timestamp. For any Firefox or OpenBSD/BSD-derived systems in scope, capture installed package lists and version strings now — AI-generated exploits targeting the specific Firefox flaw class referenced in the Mythos testing would leave no pre-exploitation forensic trace, making the pre-patch baseline your only proof-of-state artifact.

Prioritize by exploitation probability, not CVSS alone — incorporate EPSS scores and threat-intelligence enrichment into your patch prioritization workflow; CVSS base scores do not reflect AI-accelerated weaponization speed

NIST Phase: Preparation

Reference: NIST 800-61r3 §2 — Preparation: Building Detection and Prioritization Capability Before Incidents Occur

Controls: NIST SI-5 (Security Alerts, Advisories, and Directives), NIST RA-3 (Risk Assessment), NIST SI-2 (Flaw Remediation), CIS 7.1 (Establish and Maintain a Vulnerability Management Process), CIS 7.2 (Establish and Maintain a Remediation Process)

Compensating: Query the FIRST EPSS API directly at no cost: 'curl https://api.first.org/data/v1/epss?cve=CVE-XXXX-XXXX' to retrieve exploitation probability scores for each CVE in your open findings. Layer this against CISA KEV (Known Exploited Vulnerabilities catalog, downloadable as JSON from cisa.gov/known-exploited-vulnerabilities-catalog) to flag any CVE already weaponized. For AI-accelerated adversary context, treat any EPSS score above 0.10 on a CVSS 7.0+ finding as an immediate-patch trigger rather than the legacy 30-day SLA — the sub-24-hour window documented for Mythos-class systems makes standard SLA cycles operationally obsolete.

Evidence: Before re-prioritizing: export your current patch queue with CVSS scores and original prioritization rationale as a dated artifact. Pull the EPSS score for every open CVE in that queue at the same timestamp. This delta document — CVSS rank versus EPSS rank — is the evidence that justifies re-sequencing patching resources and is required to defend prioritization decisions in a post-incident review per NIST 800-61r3 §4 (Post-Incident Activity). Also capture any threat intelligence feeds showing CVEs recently added to exploit frameworks (Metasploit changelog, ExploitDB new entries) that correspond to your open findings.

Review EDR and behavioral detection coverage — verify that CrowdStrike Falcon or your equivalent EDR is tuned for post-exploitation behavioral patterns (T1068, T1078, T1059, T1203) rather than relying solely on known exploit signatures that AI-generated exploits may bypass

NIST Phase: Detection Analysis

Reference: NIST 800-61r3 §3.2 — Detection and Analysis: Monitoring and Analyzing Indicators of Adverse Events

Controls: NIST SI-4 (System Monitoring), NIST IR-4 (Incident Handling), NIST AU-2 (Event Logging), CIS 8.2 (Collect Audit Logs)

Compensating: For teams without CrowdStrike Falcon, deploy Sysmon with the SwiftOnSecurity or Olaf Hartong modular config (github.com/SwiftOnSecurity/sysmon-config) and enable Event ID 1 (Process Create), Event ID 10 (Process Access), and Event ID 3 (Network Connection). Write Sigma rules targeting: (1) T1203 — browser process (firefox.exe) spawning unexpected child processes such as cmd.exe, powershell.exe, or wscript.exe, which would be the behavioral signature of a successful Firefox exploit from the Mythos exploit class; (2) T1068 — any process executing from a non-standard path immediately following a network connection to firefox.exe; (3) T1059 — script interpreter execution with encoded command-line arguments. Run 'Get-WinEvent -LogName Security -FilterXPath '*[System[EventID=4688]]'' and filter on ParentProcessName = firefox.exe with child processes outside the browser's normal execution tree.

Evidence: Before tuning detection rules: export CrowdStrike Falcon's current prevention policy configuration and any existing custom IOA (Indicator of Attack) rules as a baseline. If using Sysmon, export the current XML config. Capture a 72-hour sample of Sysmon Event ID 1 logs filtered on firefox.exe and any BSD/OpenBSD-adjacent processes to establish a behavioral baseline before rule changes. AI-generated exploits targeting the Firefox flaw classes described in Mythos testing would not match known signature hashes — the only reliable detection artifact is the post-exploitation process tree anomaly, making this behavioral baseline your detection anchor.

Update your threat model for AI-assisted adversaries — add AI-accelerated vulnerability discovery and exploit generation as explicit threat scenarios in your threat register; map to T1587.004, T1588.006, T1595, and T1046; assess which threat actors in your profile (including FANCY BEAR, FAMOUS CHOLLIMA) are likely to adopt these capabilities

NIST Phase: Preparation

Reference: NIST 800-61r3 §2 — Preparation: Developing IR Policies and Maintaining Threat Scenario Awareness

Controls: NIST RA-3 (Risk Assessment), NIST IR-8 (Incident Response Plan), NIST PM-16 (Threat Awareness Program), CIS 7.1 (Establish and Maintain a Vulnerability Management Process)

Compensating: Update your threat register (even a spreadsheet qualifies under NIST IR-8 at small scale) with two new threat scenario rows: (1) 'AI-assisted zero-day discovery targeting legacy components' mapped to T1595 (Active Scanning), T1046 (Network Service Discovery), T1587.004 (Exploits — Develop Capabilities), and T1588.006 (Vulnerabilities — Obtain Capabilities); (2) 'AI-accelerated weaponization of disclosed CVEs within 24-hour window' for FANCY BEAR and FAMOUS CHOLLIMA actor rows. Pull current MITRE ATT&CK Group pages for G0007 (APT28/FANCY BEAR) and G0032 (Lazarus-adjacent FAMOUS CHOLLIMA) from attack.mitre.gov to verify their documented TTPs and note which overlap with post-exploitation chains that AI-generated Firefox or OpenBSD exploits would initiate.

Evidence: Before updating the threat model: export the current version of your threat register with a date stamp — this establishes the pre-update baseline and supports the post-incident review requirement under NIST 800-61r3 §4 to document model changes driven by emerging intelligence. Also capture the current MITRE ATT&CK Navigator layer for your environment so the delta between old and new threat coverage is visible and auditable.

Evaluate compensating controls for legacy systems — the 27-year-old OpenBSD flaw discovery signals that AI tooling will surface long-dormant vulnerabilities in legacy and open-source components; audit your software bill of materials (SBOM) for unmaintained or under-reviewed components

NIST Phase: Preparation

Reference: NIST 800-61r3 §2 — Preparation: Hardening Systems and Reducing Attack Surface Before Incidents

Controls: NIST SI-2 (Flaw Remediation), NIST SA-12 (Supply Chain Protection), NIST CM-7 (Least Functionality), CIS 2.1 (Establish and Maintain a Software Inventory), CIS 2.2 (Ensure Authorized Software is Currently Supported)

Compensating: Generate or update your SBOM using Syft (free, github.com/anchore/syft): run 'syft /path/to/application -o spdx-json > sbom.json' for each application. Feed the resulting SBOM into Grype (github.com/anchore/grype): 'grype sbom:sbom.json' to match components against the NVD and OSV vulnerability databases, which will surface long-dormant CVEs in OpenBSD-derived, BSD-licensed, or unmaintained open-source libraries that Mythos-class AI tooling is specifically documented to target. For any component last updated more than 5 years ago with no active maintainer, treat it as a high-priority compensating control candidate regardless of current CVE status — the OpenBSD 27-year finding demonstrates that absence of a known CVE is not equivalent to absence of a flaw.

Evidence: Before the SBOM audit: run 'find / -name "*.so" -o -name "*.dll" 2>/dev/null | xargs file | grep -i "ELF\|PE32"' to enumerate shared libraries that may not appear in package manager records (a common gap in legacy OpenBSD and embedded Linux environments). Capture the output as a dated artifact. Also collect 'pkg_info -a' (OpenBSD) or 'dpkg -l' / 'rpm -qa' output per host. These pre-audit artifacts are your evidentiary baseline if a previously unknown flaw in a legacy component is later exploited — they demonstrate what was known, when.

Brief leadership on the velocity problem — frame the business risk as a structural mismatch between current operational rhythms and attacker speed; this is a resource and process question, not only a technical one;

present options including continuous patching pipelines, vulnerability management tooling upgrades, or managed detection services

NIST Phase: Post Incident

Reference: NIST 800-61r3 §4 — Post-Incident Activity: Lessons Learned and Communicating Risk to Leadership

Controls: NIST IR-6 (Incident Reporting), NIST IR-8 (Incident Response Plan), NIST PM-9 (Risk Management Strategy), CIS 7.1 (Establish and Maintain a Vulnerability Management Process)

Compensating: Build the leadership brief around two data points your team can pull without enterprise tooling: (1) your MTTP delta from Step 1 versus the sub-24-hour AI exploitation window — expressed as 'our median patch time is X days; adversaries using Mythos-class tools can weaponize a new CVE in under 24 hours, creating a structural exposure window of X-1 days per critical finding'; (2) the CISA KEV catalog entry count for the past 90 days as a proxy for exploitation velocity trends (cisa.gov/known-exploited-vulnerabilities-catalog). Frame the OpenBSD 27-year finding and 181 Firefox exploit generation as concrete evidence that legacy component age is no longer a proxy for safety. Present the three options from the step with rough cost-effort tiers so leadership can make an informed resource decision.

Evidence: Before the briefing: assemble supporting artifacts — MTTP audit output from Step 1, EPSS-ranked patch queue from Step 2, and the current SBOM gap report from Step 5. These are the evidence package that grounds the risk narrative in operational data rather than vendor claims. Note per the advisory that the specific Anthropic Mythos figures (181 Firefox exploits, thousands of high/critical flaws) are from secondary reporting — flag this explicitly in the briefing materials per the advisory's own Step 7 guidance, and do not cite them as confirmed figures in formal risk documentation until primary-source confirmation is available.

Monitor for primary-source confirmation — the specific testing figures (181 Firefox exploits, thousands of high/critical flaws) are sourced from secondary reporting; track Anthropic's official Glasswing disclosures and any SANS or CSA technical publications for confirmed data before citing these figures in formal risk documentation

NIST Phase: Post Incident

Reference: NIST 800-61r3 §4 — Post-Incident Activity: Evidence Quality, Documentation Standards, and Intelligence Integration

Controls: NIST SI-5 (Security Alerts, Advisories, and Directives), NIST IR-5 (Incident Monitoring), NIST AU-11 (Audit Record Retention), CIS 7.1 (Establish and Maintain a Vulnerability Management Process)

Compensating: Set up no-cost monitoring for primary-source confirmation: (1) RSS feed or Google Alert on 'Anthropic security research' and 'Project Glasswing' to catch official disclosures; (2) SANS Internet Storm Center daily diary (isc.sans.edu/diary) for technical corroboration of AI-assisted exploit generation claims; (3) CSA AI Safety Working Group publications at cloudsecurityalliance.org. Create a versioned intelligence log (a dated text file qualifies) that records each secondary-source claim, its current confidence level, and the primary-source confirmation status — this is the documentation artifact required by NIST IR-5 (Incident Monitoring) for tracking evolving threat intelligence. Update your threat register entries from Step 4 with a 'pending confirmation' flag on the specific figures until primary sources publish.

Evidence: Retain all secondary-source materials (article URLs, retrieval dates, quoted figures) as time-stamped artifacts in your intelligence log. Under NIST AU-11 (Audit Record Retention), these records support the audit trail for any risk decisions made on the basis of the current reporting. If Anthropic's official disclosure materially differs from the secondary figures, the dated intelligence log demonstrates due diligence and supports any required correction to formal risk documentation or leadership briefings already delivered.

Detection Guidance

Because this story describes a capability shift rather than a specific active campaign, detection guidance focuses on behavioral patterns consistent with AI-accelerated exploitation attempts and the MITRE techniques in scope.

Reconnaissance and scanning: Watch for high-frequency, low-dwell port and service scanning (T1595, T1046) against your external attack surface, particularly against legacy services and open-source components. Unusual scan patterns from unfamiliar ASNs or cloud egress ranges are a flag.

Exploit delivery and initial access: Monitor for exploitation attempts against public-facing applications (T1190), especially for vulnerabilities disclosed within the last 30 days. Given sub-24-hour weaponization timelines, any newly disclosed CVE affecting your stack should be treated as actively exploited until patched. Correlate web application firewall (WAF) logs with vulnerability disclosure feeds.

Client-side exploitation (T1203): Hunt for unexpected or privilege-escalated child processes (cmd.exe, PowerShell, wscript.exe) spawned by browser processes (Firefox, Chrome, Edge). These are behavioral indicators of exploit delivery regardless of whether the specific exploit signature is known.

Privilege escalation and credential abuse (T1068, T1078): Alert on privilege changes outside change-control windows, unexpected service account logons, and token manipulation events in Windows Security Event Logs (Event IDs 4672, 4673, 4624 type 3 from unusual sources).

Command execution (T1059): Baseline scripting interpreter usage by host and user; alert on deviations, particularly encoded PowerShell commands or unusual Python or shell invocations on servers.

Defensive gap audit: Review whether your vulnerability management tooling has access to EPSS data and can flag newly disclosed CVEs with high exploitation-probability scores within hours of NVD publication, not days. If your workflow requires a human to manually check and queue patches, that workflow is now a structural liability for high-EPSS findings.

Indicators of Compromise

Type	Value	Context	Confidence
TOOL	`Pending — refer to Anthropic Glasswing (anthropic.com/glasswing) and primary SANS/CSA publications for any published indicators`	No specific IOC values (hashes, IPs, domains, C2 infrastructure) are present in the source material for this story; this is a capability-shift report, not a campaign disclosure with attributed infrastructure	LOW

Framework Mappings

MITRE-ATTACK

- **T1588.006** — Vulnerabilities
- **T1203** — Exploitation for Client Execution
- **T1566** — Phishing
- **T1046** — Network Service Discovery
- **T1598** — Phishing for Information
- **T1190** — Exploit Public-Facing Application
- **T1587.004** — Exploits
- **T1068** — Exploitation for Privilege Escalation

- **T1595** — Active Scanning
- **T1059** — Command and Scripting Interpreter
- **T1078** — Valid Accounts

NIST-800-53R5

- **SC-7** — Boundary Protection
- **SI-2** — Flaw Remediation
- **SI-3** — Malicious Code Protection
- **SI-4** — System Monitoring
- **AT-2** — Literacy Training and Awareness
- **CA-7** — Continuous Monitoring
- **SI-8** — Spam Protection
- **CA-8** — Penetration Testing
- **RA-5** — Vulnerability Monitoring and Scanning
- **SI-7** — Software, Firmware, and Information Integrity
- **AC-6** — Least Privilege
- **CM-7** — Least Functionality
- **AC-2** — Account Management
- **IA-2** — Identification and Authentication (Organizational Users)
- **IA-5** — Authenticator Management
- **IA-8** — Identification and Authentication (Non-Organizational Users)
- **SI-16** — Memory Protection
- **SI-10** — Information Input Validation

OWASP-TOP10-2021

- **A07:2021** — Identification and Authentication Failures
- **A03:2021** — Injection

CIS-V8

- **6.3** — Require MFA for Externally-Exposed Applications
- **6.4** — Require MFA for Remote Network Access
- **6.5** — Require MFA for Administrative Access
- **16.10** — Apply Secure Design Principles in Application Architectures
- **7.3** — Perform Automated Operating System Patch Management
- **7.4** — Perform Automated Application Patch Management

SOC2-TSC

- **CC6.1** — The entity implements logical access security software, infrastructure, and architectures over protected information assets

HIPAA-SECURITY

- **164.312(d)** — Person or Entity Authentication

ISO-27001-2022

- **A.8.8** — Management of technical vulnerabilities

MITRE ATT&CK Mapping

Technique ID	Technique Name	Tactic
T1588.006	Vulnerabilities	Resource-Development
T1203	Exploitation for Client Execution	Execution
T1566	Phishing	Initial-Access
T1046	Network Service Discovery	Discovery
T1598	Phishing for Information	Reconnaissance
T1190	Exploit Public-Facing Application	Initial-Access
T1587.004	Exploits	Resource-Development
T1068	Exploitation for Privilege Escalation	Privilege-Escalation
T1595	Active Scanning	Reconnaissance
T1059	Command and Scripting Interpreter	Execution
T1078	Valid Accounts	Defense-Evasion

Sources

Source	URL	Tier
Blog	https://www.crowdstrike.com/en-us/blog/tune-in-future-of-ai-powered...	T3
	https://www.economist.com/science-and-technology/2026/04/29/a-glimp...	T2
	https://finance.yahoo.com/sectors/technology/articles/sans-institut...	T3
	https://www.startuphub.ai/ai-news/artificial-intelligence/2026/anth...	T3
Project Glasswing: Securing critical software for the AI era - Anthropic	https://www.anthropic.com/glasswing	T1

Generated 2026-05-02 06:44 UTC by TJS Security Command Center