INTELLIGENCE BRIEFING
Security Command Center
TLP:CLEAR
2026-04-23 13:39 UTC

AI-Driven Vulnerability Discovery Outpaces Remediation: Project Glasswing Exposes Structural Security Gap

SECURITY ANALYSIS | CRITICAL | CVSS 9.5

SCC Item ID	SCC-STY-2026-0077
Type	Security Analysis
Severity	CRITICAL
CVSS Base Score	9.5
Affected Products	Firefox, Linux, FreeBSD, OpenBSD, OpenSSL, FortiGate appliances, major operating systems and browsers (unspecified versions)
Published	2026-04-23T07:30:00
Discovery Source	Rss

Executive Summary

Anthropic's Project Glasswing, built on the Mythos model, demonstrated that AI can discover vulnerabilities across major software stacks, including a 27-year-old OpenBSD flaw, at speeds that human patch cycles cannot match. Reportedly, fewer than 1% of discovered vulnerabilities were remediated, a structural gap that AI-assisted attack tooling is already exploiting: autonomous attack chains have reportedly compromised 2,516 organizations across 106 countries. This is not a single-vendor incident, it is a signal that the asymmetry between AI-accelerated offense and human-paced defense has become operationally consequential.

Technical Analysis

Project Glasswing represents a documented inflection point in AI-assisted vulnerability research. Operating on the Mythos model, the system reportedly achieved a 72.4% autonomous exploit success rate in testing against Firefox and identified a vulnerability in OpenBSD that had persisted undetected for 27 years, a class of flaw that conventional automated scanning and manual auditing had failed to surface across nearly three decades of security review. The vulnerability classes involved are not novel: use-after-free (CWE-416), race conditions (CWE-362), buffer errors (CWE-119), out-of-bounds writes (CWE-787), and code injection (CWE-94) represent the same memory safety and concurrency failure patterns that security teams have managed for years. What has changed is the speed and scale at which they can be discovered and weaponized.

The operationally significant finding is the discovery-remediation gap. If the reported figures are accurate, the vast majority of vulnerabilities identified during the project period remained unpatched, not because vendors were unresponsive, but because the volume and velocity of discovery exceeded the capacity of existing patch development and deployment pipelines. This is a structural problem, not an isolated failure.

The FortiGate MCP campaign component introduces a separate but related concern. An unattributed threat actor, reportedly leveraging AI-assisted tooling according to security news coverage, exploited FortiGate appliances, likely leveraging CVE-class flaws in perimeter devices, using attack chains across 2,516 organizations in 106 countries. The MITRE ATT&CK techniques mapped to this activity include T1595.002 (Vulnerability Scanning), T1190 (Exploit Public-Facing Application), T1588.006 and T1587.004 (acquiring and developing capabilities including exploits), T1068 (Privilege Escalation via Exploitation), T1059 (Command and Scripting Interpreter), T1203 (Exploitation for Client Execution), T1003 (Credential Dumping), and T1486 (Data Encrypted for Impact). This technique chain is consistent with a full kill chain from initial reconnaissance through operational impact, ransomware, data theft, or persistent access, executed at machine speed.

Important verification note: The source quality score for this story is 0.64, and primary sourcing is T3 (secondary news coverage and vendor advisories). The 72.4% exploit success rate, the 2,516 organization compromise figure, and the attribution of attacks to AI-assisted tooling are drawn from news coverage of the Glasswing research, not from a reviewed primary research publication. Specific CVE assignments and CVSS scores have not been independently verified against NVD or CISA KEV. The CVSS 9.5 figure in the item data is an editorial severity estimate, not an assigned score. Security teams should treat quantitative claims as directionally significant but verify against primary sources before operational reliance.

Action Checklist

1. Step 1: Assess exposure, audit your environment for all software and appliances named in this story: Firefox, Linux, FreeBSD, OpenBSD, OpenSSL, and FortiGate devices; prioritize FortiGate perimeter appliances given active exploitation attribution. Note: Specific vulnerable versions have not yet been disclosed. Monitor NVD, Mozilla, and Fortinet PSIRT for CVE assignments and patch releases.
2. Step 2: Review controls, verify that EDR telemetry covers scripting interpreter abuse (T1059), privilege escalation attempts (T1068), and credential access events (T1003); confirm FortiGate firmware is current and management interfaces are not exposed to the internet
3. Step 3: Update threat model, add AI-accelerated vulnerability discovery and autonomous exploit chaining as a named threat pattern in your threat register; the discovery-remediation gap is now a documented attack surface, not a theoretical one
4. Step 4: Audit patch pipeline capacity, assess whether your vulnerability management program can handle discovery velocity that exceeds historical CVE rates; identify where triage, testing, and deployment bottlenecks exist
5. Step 5: Communicate findings, brief leadership on the structural gap between AI-assisted discovery and remediation capacity; frame it as an operational risk to patching SLAs, not as a single product vulnerability
6. Step 6: Monitor developments, track NVD, CISA advisories, Mozilla security advisories, and Fortinet PSIRT for CVE assignments and patches tied to the vulnerability classes described; watch for follow-up research publication from Anthropic or Mythos-related disclosures

IR / Forensic Enrichment

Triage Priority	IMMEDIATE
Escalation Criteria	Escalate immediately if FortiGate event logs show any admin account creation or configuration change not traceable to a known change ticket, or if Sysmon/auditd telemetry reveals any interpreter process (bash, python3, perl) spawned as a child of an OpenSSL-linked service or Firefox, as these are the specific execution pathways consistent with AI-chained autonomous exploits targeting the named software stack; also escalate if your exposure audit reveals any FortiGate management interface resolvable from the public internet, given CISA's documented history of active exploitation against FortiGate perimeter devices.
Recovery Notes	Following containment of any confirmed FortiGate or OpenSSL-related compromise, verify firmware integrity by comparing the installed FortiGate firmware hash against Fortinet's published hash on PSIRT advisories before bringing the appliance back to production — autonomous exploit chains have been observed implanting persistent backdoors in perimeter firmware in prior FortiGate campaigns. Monitor all previously affected hosts for 30 days post-recovery using auditd on Linux and Sysmon on Windows endpoints for re-emergence of T1059 interpreter abuse or T1068 privilege escalation patterns that would indicate incomplete eradication or a second-stage implant. Revalidate that no new admin accounts, API keys, or SSH authorized_keys entries were added to FortiGate or Linux/OpenBSD hosts during the exposure window, as credential persistence is a primary objective of autonomous attack chains operating at the scale described (2,516 organizations).
Forensic Artifacts	FortiGate event logs (Log & Report > Event Logs, filtered to admin-login, system-config-change, and firmware-upgrade event types) — autonomous exploit chains targeting FortiGate perimeter appliances consistently create rogue admin accounts or modify SSL-VPN configurations as a persistence mechanism; these events appear in FortiGate system event logs even when management interface logging is set to minimal verbosity | OpenSSL application logs from web servers and VPN concentrators (/var/log/apache2/error.log, /var/log/nginx/error.log) filtered for TLS handshake failures, malloc/free errors, or segfault-adjacent entries — AI-discovered memory corruption vulnerabilities in OpenSSL (consistent with the 27-year-old OpenBSD flaw class) produce characteristic error patterns before successful exploitation achieves clean execution | Linux auditd syscall logs (/var/log/audit/audit.log) filtered for execve calls with euid=0 from non-root parent processes on hosts running OpenSSL-linked services — privilege escalation via T1068 against a kernel or library vulnerability leaves a syscall trace where the effective UID transitions from unprivileged to root without a corresponding sudo or su event in /var/log/auth.log | FreeBSD/OpenBSD kernel message logs (/var/log/messages, 'dmesg' output) for segfault or memory protection violation entries tied to sshd, httpd, or any OpenSSL-linked daemon — a 27-year-old flaw class in OpenBSD specifically is likely a memory safety issue that produces kernel-level crash artifacts or coredumps in /var/crash before a reliable exploit is achieved | Firefox parent-process telemetry via Sysmon Event ID 1 (Process Create) and Event ID 3 (Network Connection) on Windows endpoints — AI-chained browser exploits targeting Firefox as an entry point produce characteristic child process spawning (cmd.exe, powershell.exe, wscript.exe as children of firefox.exe) and outbound connection attempts from the Firefox process to C2 infrastructure, both of which are captured in Sysmon logs before any EDR behavioral alert fires

Per-Action IR Details

Step 1: Assess exposure — audit your environment for all software and appliances named in this story: Firefox, Linux, FreeBSD, OpenBSD, OpenSSL, and FortiGate devices; prioritize FortiGate perimeter

appliances given active exploitation attribution

NIST Phase: Preparation

Reference: NIST 800-61r3 §2 — Preparation: Establishing IR Capability and Asset Visibility

Controls: NIST SI-2 (Flaw Remediation), NIST SI-5 (Security Alerts, Advisories, and Directives), NIST RA-5 (Vulnerability Monitoring and Scanning), CIS 1.1 (Establish and Maintain Detailed Enterprise Asset Inventory), CIS 7.1 (Establish and Maintain a Vulnerability Management Process)

Compensating: Run 'nmap -sV --script=banner -p 443,8443,8080,10443 ' to identify FortiGate management interfaces exposed externally; cross-reference against Shodan CLI ('shodan search hostname:fortigate') for internet-facing instances. For Linux/OpenBSD hosts, run 'rpm -qa' or 'dpkg -l' piped to grep for OpenSSL version strings; for FreeBSD run 'pkg info | grep -E "openssl|firefox"'. Document every host running OpenSSL versions predating 3.x as highest priority given AI-discovered class vulnerabilities tend to target memory management primitives present in older codebases.

Evidence: Before remediating, snapshot FortiGate running config via 'get system status' and 'show full-configuration' — capture firmware version, exposed management interface bindings (HTTPS/SSH on WAN), and any recently added admin accounts. On Linux/OpenBSD systems, record current OpenSSL version ('openssl version -a'), linked library inventory ('ldd /usr/bin/openssl'), and /etc/os-release to establish a pre-patch baseline for post-incident comparison. Save FortiGate event logs from Log & Report > System Events filtered to the last 30 days before any changes.

Step 2: Review controls — verify that EDR telemetry covers scripting interpreter abuse (T1059), privilege escalation attempts (T1068), and credential access events (T1003); confirm FortiGate firmware is current and management interfaces are not exposed to the internet

NIST Phase: Detection Analysis

Reference: NIST 800-61r3 §3.2 — Detection and Analysis: Monitoring and Telemetry Coverage

Controls: NIST SI-4 (System Monitoring), NIST AU-2 (Event Logging), NIST AU-12 (Audit Record Generation), NIST IR-4 (Incident Handling), CIS 8.2 (Collect Audit Logs)

Compensating: Deploy Sysmon with SwiftOnSecurity config (github.com/SwiftOnSecurity/sysmon-config) and verify Event ID 1 (Process Create) captures interpreter spawning — specifically: cmd.exe, powershell.exe, python3, perl, bash launched as child processes of Firefox, sshd, or any OpenSSL-linked service. For T1068 on Linux, enable auditd rules: 'auditctl -a always,exit -F arch=b64 -S execve -F euid=0 -k priv_esc' to catch SUID/SGID abuse. For FortiGate, enable 'config log eventfilter' with 'set admin enable' and 'set system enable' to capture firmware tampering and admin login events, then forward via syslog to a central rsyslog host.

Evidence: Query Sysmon Event ID 1 logs for processes spawned by Firefox parent PID or sshd on OpenBSD/FreeBSD hosts within the past 30 days. On FortiGate, export Log & Report > Traffic Logs and Event Logs filtering on admin login source IPs and configuration changes — autonomous exploit chains targeting FortiGate (consistent with prior MITRE T1190 External Remote Services campaigns against FortiGate CVEs such as CVE-2023-27997) leave admin account creation events and REST API calls in these logs. Capture '/var/log/auth.log' on Linux and '/var/log/authlog' on OpenBSD for privilege escalation indicators aligned with T1068.

Step 3: Update threat model — add AI-accelerated vulnerability discovery and autonomous exploit chaining as a named threat pattern in your threat register; the discovery-remediation gap is now a documented attack surface, not a theoretical one

NIST Phase: Preparation

Reference: NIST 800-61r3 §2 — Preparation: Threat Modeling and IR Plan Maintenance

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), CIS 7.2 (Establish and Maintain a Remediation Process)

Compensating: Create a threat register entry using a free risk register template (NIST IR 8286A provides a free Community Profile template) with the following fields specific to this threat pattern: Threat Name = 'AI-Autonomous Exploit Chain'; Attack Surface = 'Pre-patch window on OpenSSL, Firefox, FortiGate firmware'; Likelihood Driver = 'Discovery-remediation velocity gap (Glasswing demonstrated sub-hour discovery vs. weeks-long patch cycles)';

Impact = 'Perimeter compromise via FortiGate + lateral movement via T1059/T1068 on Linux/OpenBSD hosts'. Link this entry to your existing patch SLA policy so SLA breaches automatically elevate the residual risk score.

Evidence: Before updating the threat model, pull your historical patch SLA metrics from your ticketing system (Jira, ServiceNow, or even a spreadsheet) for OpenSSL, Firefox, and FortiGate firmware over the past 24 months — this establishes your actual discovery-to-remediation gap baseline, which the Glasswing findings suggest is structurally exploitable. Document mean time to patch (MTTP) per product line as forensic evidence of organizational exposure window if an incident later requires breach timeline reconstruction under NIST IR-5 (Incident Monitoring).

Step 4: Audit patch pipeline capacity — assess whether your vulnerability management program can handle discovery velocity that exceeds historical CVE rates; identify where triage, testing, and deployment bottlenecks exist

NIST Phase: Preparation

Reference: NIST 800-61r3 §2 — Preparation: Resource and Capability Readiness

Controls: NIST SI-2 (Flaw Remediation), NIST CA-7 (Continuous Monitoring), NIST PM-4 (Plan of Action and Milestones Process), CIS 7.2 (Establish and Maintain a Remediation Process), CIS 7.3 (Perform Automated Operating System Patch Management), CIS 7.4 (Perform Automated Application Patch Management)

Compensating: Map your patch pipeline stages on a whiteboard or spreadsheet: (1) NVD/CISA KEV ingestion, (2) affected asset identification, (3) patch testing in non-prod, (4) deployment approval, (5) verification. For each stage, record the current cycle time and responsible party. For FortiGate specifically, test whether your team can execute a firmware upgrade on a perimeter device within 72 hours of a PSIRT advisory — this is the operationally relevant threshold given AI-assisted tooling can develop working exploits within hours of disclosure based on Glasswing's demonstrated capability. Use CISA's free Known Exploited Vulnerabilities catalog (cisa.gov/known-exploited-vulnerabilities-catalog) as your triage input and measure how long after KEV addition your environment achieves full remediation.

Evidence: Pull your vulnerability scanner output (OpenVAS/GVM is free) filtered to Firefox, OpenSSL CVEs, and FortiGate advisories from the past 12 months; calculate the delta between CVE NVD publication date and your verified-remediated date for each. This gap analysis is the primary forensic artifact establishing organizational exposure duration — it directly quantifies how much of your environment was in the pre-patch window that AI-assisted attack tooling exploits. Preserve this data in case of regulatory inquiry under breach notification requirements.

Step 5: Communicate findings — brief leadership on the structural gap between AI-assisted discovery and remediation capacity; frame it as an operational risk to patching SLAs, not as a single product vulnerability

NIST Phase: Post Incident

Reference: NIST 800-61r3 §4 — Post-Incident Activity: Lessons Learned and Organizational Improvement

Controls: NIST IR-6 (Incident Reporting), NIST IR-8 (Incident Response Plan), NIST PM-3 (Information Security and Privacy Resources), CIS 7.1 (Establish and Maintain a Vulnerability Management Process)

Compensating: Prepare a one-page risk brief structured as: (1) Current state — your MTTP for FortiGate firmware and OpenSSL patches vs. AI-assisted exploit development timelines documented in Project Glasswing reporting; (2) Gap quantification — number of assets in your environment running affected software stacks with no current CVE but within the structural exposure window; (3) Resource ask — specific headcount, tooling, or process changes needed to close the gap. Use the CISA Known Exploited Vulnerabilities catalog entry counts for FortiGate (historically 20+ KEV entries) as a concrete reference point for leadership to understand vendor-specific exploitation frequency.

Evidence: The primary evidence for this briefing is your own patch gap data from Step 4 combined with the Glasswing-reported statistic that fewer than 1% of AI-discovered vulnerabilities were remediated at time of disclosure — document this ratio against your own remediation percentage for the affected product stack. If your environment had any FortiGate management interfaces exposed per the Step 1 audit, include that finding as a concrete current-state risk indicator. Preserve all briefing materials as they establish the organizational awareness baseline, which is relevant if a subsequent incident triggers regulatory notification requirements.

Step 6: Monitor developments — track NVD, CISA advisories, Mozilla security advisories, and Fortinet PSIRT for CVE assignments and patches tied to the vulnerability classes described; watch for follow-up research

publication from Anthropic or Mythos-related disclosures

NIST Phase: Detection Analysis

Reference: NIST 800-61r3 §3.2 — Detection and Analysis: Threat Intelligence Integration

Controls: NIST SI-5 (Security Alerts, Advisories, and Directives), NIST SI-4 (System Monitoring), NIST IR-5 (Incident Monitoring), CIS 7.1 (Establish and Maintain a Vulnerability Management Process), CIS 8.2 (Collect Audit Logs)

Compensating: Configure free RSS/Atom feed aggregation (Feedly free tier or a self-hosted FreshRSS instance) to ingest: NVD CVE feed filtered to vendors 'mozilla', 'openssl', 'fortinet', 'linux', 'freebsd', 'openbsd'; CISA KEV RSS feed; Fortinet PSIRT advisories (fortiguard.fortinet.com/psirt); and Mozilla Foundation Security Advisories (mozilla.org/en-US/security/advisories/). Set a daily 15-minute triage window to review new entries against your asset inventory from Step 1. For Mythos/Glasswing-specific research, set a Google Scholar alert for 'Project Glasswing', 'Mythos model vulnerability', and 'AI vulnerability discovery autonomous exploit' to catch peer-reviewed follow-up publications before they are widely weaponized.

Evidence: When a new CVE is assigned to any of the named products, immediately query your FortiGate logs for the specific service or interface class the CVE targets — for example, if a new OpenSSL CVE targets TLS handshake processing, query FortiGate SSL inspection logs and Linux syslog for TLS negotiation anomalies in the 30 days prior to CVE publication, as AI-assisted exploit chains may have been active in the pre-disclosure window. Preserve a 90-day rolling archive of FortiGate traffic logs, Linux /var/log/syslog, and OpenSSL application logs (Apache/Nginx access and error logs) per NIST AU-11 (Audit Record Retention) to support retroactive analysis when new CVEs are disclosed.

Detection Guidance

Given the MITRE techniques mapped to this campaign, focus detection efforts on three areas.

Perimeter and initial access: Review FortiGate VPN and management interface logs for anomalous authentication attempts, unexpected firmware queries, or lateral movement originating from appliance IPs. T1190 exploitation of public-facing appliances often leaves traces in web application logs as malformed requests or unexpected HTTP method patterns.

Post-exploitation behavior: Hunt for T1059 activity, unusual scripting interpreter invocations (Bash, Python, PowerShell) spawned from processes that do not normally execute them. Correlate with T1068 privilege escalation attempts: look for processes spawning with elevated privileges without a corresponding user-initiated sudo or equivalent event. T1003 credential dumping leaves artifacts in LSASS access events on Windows and in /etc/shadow access attempts on Linux systems.

Memory safety exploitation patterns: Given the memory safety vulnerability classes involved (CWE-416 use-after-free and CWE-787 out-of-bounds write), exploitation against browsers like Firefox typically manifests as renderer process crashes followed immediately by unexpected network connections or process spawning. Enable crash telemetry and correlate browser crash events with outbound connection anomalies.

Discovery-phase indicators: T1595.002 vulnerability scanning from external sources targeting your perimeter appliances may appear in firewall logs as systematic port sweeps or repeated probes against management ports (8443, 443, 22) from rotating IP ranges.

Log sources to prioritize: FortiGate syslog and PSIRT advisories, EDR process creation and network events, browser crash telemetry, authentication logs for perimeter devices, and outbound DNS for unexpected or newly registered domains.

Indicators of Compromise

Type	Value	Context	Confidence
TOOL	`Pending — refer to Anthropic Project Glasswing primary research disclosure and Fortinet PSIRT FG-IR-26-076 for published indicators`	AI-assisted exploit tooling (Mythos model) and FortiGate MCP campaign indicators including C2 infrastructure, payload hashes, and scanning signatures have been referenced in news coverage but specific values were not published in available T3 sources	LOW

Framework Mappings

MITRE-ATTACK

- **T1588.006** — Vulnerabilities
- **T1587.004** — Exploits
- **T1486** — Data Encrypted for Impact
- **T1003** — OS Credential Dumping
- **T1595.002** — Vulnerability Scanning
- **T1190** — Exploit Public-Facing Application
- **T1203** — Exploitation for Client Execution
- **T1068** — Exploitation for Privilege Escalation
- **T1059** — Command and Scripting Interpreter

NIST-800-53R5

- **CP-9** — System Backup
- **CP-10** — System Recovery and Reconstitution
- **AC-6** — Least Privilege
- **IA-5** — Authenticator Management
- **SI-4** — System Monitoring
- **CA-8** — Penetration Testing
- **RA-5** — Vulnerability Monitoring and Scanning
- **SC-7** — Boundary Protection
- **SI-2** — Flaw Remediation
- **SI-7** — Software, Firmware, and Information Integrity
- **SI-3** — Malicious Code Protection
- **CM-7** — Least Functionality
- **SI-16** — Memory Protection
- **SI-10** — Information Input Validation

CIS-V8

- **16.10** — Apply Secure Design Principles in Application Architectures
- **7.3** — Perform Automated Operating System Patch Management

- **7.4** — Perform Automated Application Patch Management

OWASP-TOP10-2021

- **A03:2021** — Injection

ISO-27001-2022

- **A.8.8** — Management of technical vulnerabilities
- **A.5.21** — Managing information security in the ICT supply chain

SOC2-TSC

- **CC9.2** — Manages risks associated with vendors and business partners

MITRE ATT&CK Mapping

Technique ID	Technique Name	Tactic
T1588.006	Vulnerabilities	Resource-Development
T1587.004	Exploits	Resource-Development
T1486	Data Encrypted for Impact	Impact
T1003	OS Credential Dumping	Credential-Access
T1595.002	Vulnerability Scanning	Reconnaissance
T1190	Exploit Public-Facing Application	Initial-Access
T1203	Exploitation for Client Execution	Execution
T1068	Exploitation for Privilege Escalation	Privilege-Escalation
T1059	Command and Scripting Interpreter	Execution

Sources

Source	URL	Tier
Security News	https://thehackernews.com/2026/04/project-glasswing-proved-ai-can-f...	T3
Security Advisories for Firefox - Mozilla	https://www.mozilla.org/en-US/security/known-vulnerabilities/firefox/	T3
AI Just Hacked One Of The World's Most Secure Operating Systems	https://www.forbes.com/sites/amirhusain/2026/04/01/ai-just-hacked-o...	T3

Source	URL	Tier
FreeBSD : Firefox -- Multiple vulnerabilities (b704d4b8-4b87-1...	https://www.tenable.com/plugins/nessus/240167	**T3**
OpenSSL CVE-2025-15467 - PSIRT | FortiGuard Labs - Fortinet	https://fortiguard.fortinet.com/psirt/FG-IR-26-076	**T3**

Generated 2026-04-23 13:39 UTC by TJS Security Command Center