INTELLIGENCE BRIEFING
Security Command Center
TLP:CLEAR
2026-03-29 18:34 UTC

GAN-Optimized Phishing Machines Target AI Browsers: The Attack Surface Has Shifted From Users to Agents

SECURITY ANALYSIS | HIGH | CVSS 7.5

SCC Item ID	SCC-STY-2026-0023
Type	Security Analysis
Severity	HIGH
CVSS Base Score	7.5
Affected Products	Perplexity Comet AI Browser (pre-patch), 1Password browser extension (via Comet integration), Gmail (via Comet integration)
Published	2026-03-22
Discovery Source	Rss

Executive Summary

Researchers at Guardio, Trail of Bits, and Zenity Labs demonstrated that Perplexity's Comet AI browser can be manipulated into autonomously executing phishing attacks, credential theft, and 1Password vault takeovers, in some cases completing the full attack chain in under four minutes. The core vulnerability is not a single patchable flaw: it is a structural property of current agentic AI architectures, where verbose reasoning output becomes an adversarial feedback channel and prompt injection via untrusted web content requires no user interaction. Organizations deploying AI browsers or any agentic tooling with access to credential stores should treat this as an active, generalizable threat class, not a vendor-specific incident.

Technical Analysis

The attack surface documented across three independent research efforts represents a meaningful shift in how adversaries can interact with enterprise tooling. Where traditional phishing requires a human to click, read, and act, agentic browsers like Comet can be directed to act autonomously on behalf of the user, and that autonomy becomes the attack vector.

Guardio's demonstration centered on what researchers called "agentic blabbering" - the tendency of LLM-based agents to externalize their reasoning chains in verbose output. Guardio's team used a Generative Adversarial Network to iteratively test malicious web pages against Comet's own reasoning output, using the agent's responses as a real-time oracle to refine the attack payload. The result was a self-optimizing phishing page that could bypass the agent's defenses with no prior knowledge of its internal architecture, only its observable

behavior. Guardio's demonstration showed the full phishing sequence could complete in under four minutes.

Zenity Labs documented a separate path (tracked as PerplexedBrowser) in which an attacker could weaponize Comet's browser agent to take over a victim's 1Password vault. The attack exploited insufficient origin validation and the agent's willingness to act on instructions embedded in untrusted web content, a prompt injection pattern (mapped to CWE-77 and CWE-20) that is well-documented in LLM research but newly demonstrated at this level of credential access. The 1Password integration attack path did not require user interaction beyond the agent having access to the extension.

The Register confirmed an additional exploitation path via calendar invites, establishing that the injection surface extends beyond web pages to any content the agent processes as trusted input.

Relevant MITRE ATT&CK techniques observed across the documented attack paths include T1566 and T1566.002 (phishing and spear-phishing), T1555 (credentials from password stores), T1185 (browser session hijacking), T1557 (adversary-in-the-middle positioning), T1059 (command and scripting interpreter abuse via agent actions), T1041 and T1056 (exfiltration and input capture), and T1539 (session cookie theft).

Perplexity has patched the specific PerplexedBrowser vulnerabilities. However, the researchers across all three groups are consistent on a critical point: the patches address identified instances, not the underlying architecture. Verbose agent reasoning as an adversarial feedback channel, susceptibility to prompt injection from untrusted content, and insufficient trust boundary enforcement between agent and integrated tools are structural properties of current LLM-based agentic systems. Any product built on this architecture carries analogous exposure until the architecture changes.

For security teams, the defensive implication is significant. Existing browser security controls, DLP policies, and phishing defenses were designed for human actors making decisions. An agent that autonomously accesses credential stores, fills forms, and exfiltrates data on a four-minute clock operates outside the assumptions most enterprise controls were built on.

Action Checklist

1. Step 1: Assess exposure, inventory all AI browser tools and agentic AI products deployed in your environment, with specific attention to any that have access to credential managers, email, calendar, or authenticated SaaS sessions.
2. Step 2: Verify patch status, confirm that any deployment of Perplexity Comet has received the vendor patch addressing the PerplexedBrowser vulnerabilities; treat unpatched instances as actively exploitable. (Verify current patch status and version requirements via Perplexity's official security advisories.)
3. Step 3: Restrict agentic tool permissions, apply least-privilege principles to AI browser integrations; revoke or scope down access to credential stores (1Password, browser-native keychains) until trust boundary controls are evaluated by the vendor.
4. Step 4: Update your threat model, incorporate prompt injection via untrusted web content as an active TTP against agentic tooling; map to T1185, T1555, and T1566 in your detection and response playbooks.
5. Step 5: Brief leadership on the structural risk, this is not a single-vendor patch event; communicate to CISOs and relevant stakeholders that any agentic AI with access to sensitive integrations carries analogous architectural exposure until the industry resolves verbose reasoning and origin validation weaknesses.
6. Step 6: Monitor research developments, track follow-up disclosures from Guardio, Trail of Bits, and Zenity Labs; this research area is active and additional agentic browser products are likely to be

examined.

IR / Forensic Enrichment

Triage Priority	URGENT
Escalation Criteria	Escalate to external IR firm immediately if any unpatched Comet instances have active 1Password vault integrations AND recent phishing or credential compromise is detected; this indicates active exploitation with structural architectural vulnerability requiring forensic analysis beyond in-house capability.
Recovery Notes	Post-containment: (1) Reset 1Password master passwords for all affected users and force re-authentication of vault access; (2) Revoke OAuth tokens and session keys for all integrated SaaS applications (Gmail, etc.) that were accessible through Comet; (3) Conduct endpoint sweep for persistence mechanisms—check browser extensions, system start-up folders, and scheduled tasks for unauthorized agentic tools. (4) Update detection rules with extracted IoCs (malicious prompt injection patterns, phishing domains used) and re-scan logs for signs of earlier compromise window.
Forensic Artifacts	Browser extension manifest files and installation metadata (/opt/google/chrome/extensions/ on Linux, ~/Library/Application Support/Google/Chrome/Default/Extensions/ on macOS, HKCU\Software\Google\Chrome\Extensions on Windows) | 1Password vault access and modification logs (itemuse.sqlite, Activity Logs in 1Password client, and server-side audit logs if backup/export available) | Perplexity Comet conversation history and cached prompts (~/.config/perplexity/ or ~/Library/Application Support/Perplexity/) | Network traffic logs and firewall rules showing connections to credential manager APIs, phishing domains, and external cloud storage (DNS logs, proxy/firewall logs, packet captures) | System event logs and process execution history (Windows Event Log 4688 and Sysmon Event 1, auditd on Linux, log stream on macOS) showing launch of Comet and subsequent browser/email activity

Per-Action IR Details

Step 1: Assess exposure — inventory all AI browser tools and agentic AI products deployed in your environment, with specific attention to any that have access to credential managers, email, calendar, or authenticated SaaS sessions.

NIST Phase: Preparation

Reference: NIST 800-61r3 §2.1 (Preparation phase: tools and resources)

Controls: NIST 800-53 CM-8 (Information System Component Inventory), CIS 2.1 (Address Unauthorized Software)

Compensating: Use osquery or Wazuh agent to enumerate browser extensions and running processes: `osquery> SELECT * FROM chrome_extensions;` On macOS, parse ~/Library/Application Support/Google/Chrome/Default/Extensions/ directory. On Windows, query HKLM\SOFTWARE\Wow6432Node\Microsoft\Windows\CurrentVersion\Uninstall for installed software. Cross-reference against approved inventory spreadsheet manually.

Evidence: Capture before inventory: (1) Browser extension manifest files and installation timestamps from user profile directories; (2) Process execution history from sysmon/auditd showing browser and agentic tool launches; (3) Network traffic logs showing outbound connections from browser extensions to credential manager APIs; (4) System registry/plist dumps showing configured extensions and integrations.

Step 2: Verify patch status — confirm that any deployment of Perplexity Comet has received the vendor patch addressing the PerplexedBrowser vulnerabilities; treat unpatched instances as actively exploitable.

NIST Phase: Detection Analysis

Reference: NIST 800-61r3 §3.2.1 (Detection and Analysis: determine whether an incident has occurred)

Controls: NIST 800-53 SI-2 (Flaw Remediation), CIS 7.4 (Perform Software Integrity Checks)

Compensating: Without patch management tooling, manually verify: (1) Launch Perplexity Comet and check Help > About for version number; cross-reference against Perplexity's security advisory page for patched versions; (2) On macOS, run `mdls /Applications/Perplexity.app | grep kMDItemVersion`; on Windows, check application properties or registry HKLM\SOFTWARE\Perplexity. (3) Query Perplexity's public changelog via curl to confirm patch date. Document all findings with screenshots.

Evidence: Capture before remediation: (1) Application binary hashes (SHA-256) of all installed Perplexity Comet instances using `shasum -a 256` or `Get-FileHash` PowerShell cmdlet; (2) Installation and update timestamps from file metadata and OS package manager logs (rpm/apt history on Linux, Windows Update history); (3) Network logs showing any outbound connections from unpatched versions to credential manager endpoints (examine firewall logs for 1Password API domains).

Step 3: Restrict agentic tool permissions — apply least-privilege principles to AI browser integrations; revoke or scope down access to credential stores (1Password, browser-native keychains) until trust boundary controls are evaluated by the vendor.

NIST Phase: Containment

Reference: NIST 800-61r3 §3.2.4 (Containment, Eradication, and Recovery: stop the attack)

Controls: NIST 800-53 AC-6 (Least Privilege), NIST 800-53 AC-3 (Access Enforcement), CIS 5.1 (Establish and Maintain an Inventory of Accounts)

Compensating: For teams without centralized permission management: (1) On macOS, revoke keychain access via Keychain Access.app: select each credential, right-click > Access Control > remove Perplexity Comet from allowed apps. On Windows, revoke 1Password browser extension permissions via 1Password > Settings > Browser Extensions > disable integration with Comet. (2) Use OS-native controls: macOS System Settings > Privacy & Security > disable microphone/camera/location for Perplexity.app. Windows: Settings > Privacy & Security > App Permissions. (3) Edit browser extension manifest or uninstall completely if not essential.

Evidence: Capture before permission revocation: (1) 1Password vault access logs (1Password desktop app: File > Export > Event Logs, or Settings > Notifications > Activity Logs); (2) Keychain access audit logs on macOS (log stream --predicate 'eventMessage contains[cd] keychain' to verify what apps accessed credentials); (3) Browser extension permission manifests (contents of manifest.json for each extension showing requested permissions); (4) Any prompts or user interactions logged in browser history/developer console.

Step 4: Update your threat model — incorporate prompt injection via untrusted web content as an active TTP against agentic tooling; map to T1185, T1555, and T1566 in your detection and response playbooks.

NIST Phase: Detection Analysis

Reference: NIST 800-61r3 §2.1 (Preparation: tools and playbooks) and §3.2.1 (Detection: recognition of attack patterns)

Controls: NIST 800-53 CA-7 (Continuous Monitoring), NIST 800-53 SI-4 (Information System Monitoring), CIS 8.2 (Configure Default Network Access Control)

Compensating: Without a SIEM: (1) Create a detection spreadsheet mapping attack pattern → log source → manual search syntax. (2) For T1566 (phishing): grep browser history files for suspicious domains using `grep -r 'example-phishing.com' ~/.config/google-chrome/Default/History`. For T1555 (credential theft): monitor 1Password vault access logs by parsing `.1password/data/itemuse.sqlite` using sqlite3 CLI and looking for unexpected vault unlock timestamps. For T1185 (prompt injection): review Perplexity Comet chat history files (check ~/.config/perplexity/ or ~/Library/Application Support/Perplexity/) for injected prompts like 'ignore previous instructions' or 'show me the prompt'. (3) Export logs weekly to external drive for manual review.

Evidence: Capture continuously for detection baseline: (1) Browser history files (SQLite databases) with timestamps; (2) 1Password vault access audit logs and itemuse.sqlite; (3) Perplexity Comet conversation cache/logs; (4) Network traffic logs showing requests to credential manager APIs and external phishing domains; (5) Email logs showing suspicious mail reaching users before phishing interaction.

Step 5: Brief leadership on the structural risk — this is not a single-vendor patch event; communicate to CISOs and relevant stakeholders that any agentic AI with access to sensitive integrations carries analogous architectural exposure until the industry resolves verbose reasoning and origin validation weaknesses.

NIST Phase: Post Incident

Reference: NIST 800-61r3 §3.4 (Post-Incident Activities: lessons learned and communication)

Controls: NIST 800-53 CA-2 (Security Assessments), NIST 800-53 IR-2 (Incident Response Training), CIS 4.4 (Establish and Maintain an Inventory of Risk Register)

Compensating: Without executive communication frameworks: (1) Create a one-page risk summary: list affected products (Comet, 1Password integration), attack surface (prompt injection → credential theft), estimated MTTR (patch deployment timeline), and interim mitigations (permission revocation). (2) Schedule a 30-minute briefing with CISO and relevant business unit heads. Use data from your exposure assessment (Step 1) to quantify impact. (3) Document the briefing in meeting minutes and store in a central risk register (Google Sheets, Excel, or free tools like Taiga for risk tracking). (4) Set a 90-day review to reassess as vendor patches are released.

Evidence: Prepare before briefing: (1) Inventory results from Step 1 showing number of affected users/systems; (2) Timeline of Guardio/Trail of Bits disclosures and vendor response dates; (3) Examples of successful prompt injection payloads from research (sanitized); (4) Business impact assessment: how many users have 1Password vaults exposed if Comet is compromised; (5) Compliance implications (if regulated: HIPAA, PCI-DSS, SOC2 implications of AI-driven credential theft).

Step 6: Monitor research developments — track follow-up disclosures from Guardio, Trail of Bits, and Zenity Labs; this research area is active and additional agentic browser products are likely to be examined.

NIST Phase: Preparation

Reference: NIST 800-61r3 §2.1 (Preparation: tools and resources) and §3.1 (Detection and Analysis: information sources)

Controls: NIST 800-53 SI-5 (Security Alerts, Advisories, and Directives), CIS 4.1 (Establish and Maintain a Secure Configuration Management Process)

Compensating: Without threat intelligence feeds: (1) Subscribe to free email alerts from vendor security pages: Perplexity (perplexity.ai/security), 1Password (1password.com/security), Gmail (support.google.com/bugs); (2) Set up Google Alerts for 'Guardio Labs prompt injection', 'Trail of Bits agentic AI', 'Zenity Labs browser security'; (3) Monitor HackerNews and Reddit (/r/cybersecurity, /r/netsec) by checking daily and filtering for 'Comet', 'agentic AI', 'prompt injection'; (4) Add CISA ADP (Authorized Disclosure Protocol) and CVE feeds via RSS: nvd.nist.gov/feeds/json/cve/1.1/nvdcve-1.1-modified.json; (5) Create a shared Google Doc or wiki for team to log new threat intel findings weekly.

Evidence: Maintain continuously: (1) Dated records of all security advisories reviewed (screenshot or PDF); (2) Links and summaries of published PoC code or attack demonstrations; (3) Vendor patch release dates and your remediation timeline; (4) Notes on which other agentic AI products (Claude, ChatGPT, Copilot) have been tested by researchers and any vulnerabilities found; (5) Update threat model document quarterly with new attack patterns discovered.

Detection Guidance

Standard phishing and credential-access detections are insufficient for this threat class because the malicious action is performed by the agent, not the user. Detection must shift toward behavioral anomalies in agent activity.

Log and alert on: credential manager access events (1Password API calls, browser keychain reads) initiated by browser automation processes rather than direct user interaction; browser extension activity generating outbound data transfers to unfamiliar domains, particularly during or immediately after page loads on external sites; calendar invite processing that results in URL navigation or form submission by an automated process; and agent reasoning logs (where accessible) containing instructions sourced from web page content, especially

those directing credential access or data submission.

For organizations with access to endpoint telemetry: flag sequences where a browser process accesses a credential store and initiates an outbound connection within a short time window, this pattern matches the documented exfiltration chain (T1555 followed by T1041).

For email and calendar security gateways: review filtering rules for calendar invite payloads containing embedded URLs or instruction-style text; the calendar-based injection path documented by The Register suggests this is an under-monitored initial access vector for agentic tools.

Note: The detection strategies above assume access to browser extension logs, credential manager audit trails, and endpoint telemetry. Organizations without these capabilities should prioritize: (1) restricting agentic tool permissions, (2) disabling credential store integration until controls can be evaluated, and (3) monitoring for unexpected credential manager access events in any available logs (browser history, OS-level activity monitoring, or SaaS audit trails).

Hunt hypothesis: Are any AI browser or agent processes in your environment making credential store API calls outside of user-initiated authentication flows? This behavioral baseline does not exist in most environments today and should be established before agentic tooling is expanded.

Framework Mappings

MITRE-ATTACK

- **T1555** — Credentials from Password Stores
- **T1059** — Command and Scripting Interpreter
- **T1185** — Browser Session Hijacking
- **T1557** — Adversary-in-the-Middle
- **T1566** — Phishing
- **T1566.002** — Spearphishing Link
- **T1041** — Exfiltration Over C2 Channel
- **T1056** — Input Capture
- **T1539** — Steal Web Session Cookie

NIST-800-53R5

- **CM-7** — Least Functionality
- **SI-3** — Malicious Code Protection
- **SI-4** — System Monitoring
- **SI-7** — Software, Firmware, and Information Integrity
- **AT-2** — Literacy Training and Awareness
- **CA-7** — Continuous Monitoring
- **SC-7** — Boundary Protection
- **SI-8** — Spam Protection
- **SC-8** — Transmission Confidentiality and Integrity
- **SI-10** — Information Input Validation

OWASP-TOP10-2021

- **A02:2021** — Cryptographic Failures
- **A03:2021** — Injection

CIS-V8

- **3.10**
- **16.10**
- **6.3** — Require MFA for Externally-Exposed Applications
- **7.3** — Perform Automated Operating System Patch Management
- **7.4** — Perform Automated Application Patch Management
- **14.2** — Train Workforce Members to Recognize Social Engineering Attacks

HIPAA-SECURITY

- **164.312(e)(1)** — Transmission Security
- **164.312(d)** — Person or Entity Authentication
- **164.308(a)(5)(i)** — Security Awareness and Training
- **164.308(a)(6)(ii)** — Response and Reporting

ISO-27001-2022

- **A.8.26** — Application security requirements
- **A.8.8** — Management of technical vulnerabilities
- **A.5.34** — Privacy and protection of personal information
- **A.5.21** — Managing information security in the ICT supply chain

SOC2-TSC

- **CC6.1** — Logical access security software, infrastructure, and architectures
- **CC7.4** — Responds to identified security incidents
- **CC9.2** — Manages risks associated with vendors and business partners

MITRE ATT&CK Mapping

Technique ID	Technique Name	Tactic
T1555	Credentials from Password Stores	Credential-Access
T1059	Command and Scripting Interpreter	Execution
T1185	Browser Session Hijacking	Collection
T1557	Adversary-in-the-Middle	Credential-Access
T1566	Phishing	Initial-Access
T1566.002	Spearphishing Link	Initial-Access

Technique ID	Technique Name	Tactic
T1041	Exfiltration Over C2 Channel	Exfiltration
T1056	Input Capture	Collection
T1539	Steal Web Session Cookie	Credential-Access

Sources

Source	URL	Tier
Security News	https://thehackernews.com/2026/03/researchers-trick-perplexitys-com...	T3
Perplexity Comet browser hole was exploitable via cal invite	https://www.theregister.com/2026/03/03/perplexity_comet_browser_hol...	T3
Until last month, attackers could've stolen info from Perplexity Comet ...	https://www.reddit.com/r/1Password/comments/1rk4279/until_last_mont...	T3
Agentic Browser Vulnerability Exposed: Perplexity's Comet Hijacked	https://www.linkedin.com/posts/tamir-ishay-sharbat-069496163_today-...	T3
How Attackers Can Hijack Comet to Takeover your 1Password Vault	https://labs.zenity.io/p/perplexedbrowser-how-attackers-can-weaponi...	T3

Generated 2026-03-29 18:34 UTC by TJS Security Command Center