INTELLIGENCE BRIEFING
Security Command Center
TLP:CLEAR
2026-06-16 07:18 UTC

SearchLeak: Single-Click Prompt Injection Attack Enables Enterprise Data Exfiltration via Microsoft Copilot

SECURITY ANALYSIS | HIGH | CVSS 7.5

SCC Item ID	SCC-STY-2026-0213
Type	Security Analysis
Severity	HIGH
CVSS Base Score	7.5
Affected Products	Microsoft Copilot (enterprise deployments; patched by Microsoft)
Published	2026-06-15T15:27:48
Discovery Source	Rss

Executive Summary

Researchers disclosed SearchLeak, a three-stage prompt-injection attack chain that allowed an attacker to exfiltrate enterprise data from Microsoft Copilot with a single user click. The attack exploited how Copilot's search-augmented generation ingests untrusted external content, embedding hidden instructions that redirected the AI to silently retrieve and transmit sensitive internal data. Microsoft has patched this specific instance, but the underlying attack class, indirect prompt injection, remains an unsolved architectural problem across AI-integrated enterprise platforms, signaling a durable and escalating threat category for any organization deploying large language models against internal data.

Technical Analysis

SearchLeak operates as a three-stage indirect prompt injection chain targeting Microsoft Copilot's search-augmented generation (RAG-like) pipeline. In stage one, an attacker embeds malicious instructions inside content that Copilot is likely to retrieve during a search query, such as a publicly accessible document or web page. In stage two, when a user initiates a Copilot interaction that causes the model to fetch that attacker-controlled content, the hidden instructions are ingested as trusted input by the LLM, without the user seeing or approving them. In stage three, the injected instructions direct Copilot to formulate and issue an outbound request, typically via a hidden URL, that carries enterprise data to an attacker-controlled destination. The entire exfiltration chain completes after the user's initial single click, with no further interaction required.

The attack maps to multiple ATT&CK techniques: T1190 (Exploit Public-Facing Application) for abusing the Copilot surface, T1059 (Command and Scripting Interpreter) in the sense that the LLM prompt functions as a

command interpreter executing attacker-supplied instructions, T1567 (Exfiltration Over Web Service) for the outbound data transmission via hidden URL, and T1071.001 (Application Layer Protocol: Web Protocols) for the covert exfiltration channel. The initial content delivery vector aligns with T1566 (Phishing) when attacker content is delivered through social engineering.

The weaknesses exploited, CWE-20 (Improper Input Validation), CWE-77 (Command Injection), CWE-116 (Improper Encoding or Escaping of Output), and CWE-200 (Exposure of Sensitive Information), are not novel. They are decades-old categories now manifesting in a new architectural layer: the AI inference pipeline. This is the core analytical point. Patching the specific SearchLeak vector does not close the class. Academic work on EchoLeak (arxiv.org/html/2509.10540v1), a zero-click variant of the same prompt-injection family, demonstrates that researchers are actively mapping the full attack surface of LLM-integrated enterprise tools. The EchoLeak variant requires no user click at all, placing it at a higher severity tier.

The defensive gap SearchLeak exposes is not a single misconfiguration. It is a structural gap in how enterprise AI platforms handle the trust boundary between internally trusted data and externally retrieved content. Enterprise deployments of Copilot, and by extension any AI assistant with retrieval or browsing capabilities, implicitly grant the model authority to act on instructions regardless of instruction source. Until AI platforms enforce strict separation between data and instruction planes, the category remains exploitable across products and vendors.

Action Checklist

1. Step 1: Assess exposure, inventory every Microsoft Copilot deployment across the enterprise, including Microsoft 365 Copilot, Copilot for Security, and any custom Copilot Studio agents; confirm whether each deployment has web search or external content retrieval enabled.
2. Step 2: Verify patch status, confirm Microsoft's patch for the SearchLeak-specific vector has been applied to all Copilot-enabled tenants; review Microsoft's security update documentation for the specific tenant configuration changes required, if any.
3. Step 3: Audit Copilot data permissions, apply least privilege to Copilot's data access scope per NIST AC-6 (Least Privilege); Copilot inherits the permissions of the authenticated user, meaning over-permissioned accounts create over-permissioned AI agents; reduce Copilot's reachable data to what each user role genuinely requires.
4. Step 4: Restrict or disable external content retrieval, evaluate whether Copilot's web search and external URL retrieval features are operationally necessary; if not, disable them to eliminate the attacker-controlled content ingestion vector; align this with NIST AC-4 (Information Flow Enforcement) to enforce approved information flow paths.
5. Step 5: Instrument Copilot activity for anomaly detection, enable and review audit logs for Copilot interactions per NIST AU-2 (Event Logging) and AU-6 (Audit Record Review, Analysis, and Reporting); hunt for outbound HTTP requests originating from Copilot sessions to domains outside the organization's approved list, and for sessions accessing unusually broad data scopes relative to the user's normal pattern.
6. Step 6: Update AI threat model, add indirect prompt injection as a formal threat category in your threat register, mapped to T1059, T1567, and T1190; treat any AI system with retrieval or browsing capabilities as a potential execution environment for attacker-supplied instructions, not just a passive assistant.
7. Step 7: Expand review beyond Microsoft, SearchLeak and EchoLeak establish that this attack class affects any LLM-integrated platform with retrieval capabilities; apply the same data-permission audit,

content-source restriction, and logging controls to other AI tools in the environment (Google Workspace AI, Salesforce Einstein, GitHub Copilot, and similar).

8. Step 8: Communicate risk to leadership, brief the CISO and relevant stakeholders on the distinction between patch deployment (completed) and category-level risk (ongoing); frame this as an architectural trust-boundary problem in AI integration that requires sustained governance, not a closed incident.

IR / Forensic Enrichment

Triage Priority	URGENT
Escalation Criteria	Escalate immediately to CISO and legal/privacy counsel if Microsoft Purview CopilotInteraction audit logs reveal any session where Copilot accessed and transmitted files containing PII, PHI, financial records, or credentials to an external domain — this crosses the threshold for breach notification assessment under GDPR Article 33, HIPAA §164.412, or applicable state privacy law.
Recovery Notes	After patch verification and permission scoping are complete, restore any externally-facing Copilot features only after confirming web search and external URL retrieval controls are locked to an approved-domain allowlist. For 90 days post-containment, run weekly queries against Microsoft Purview CopilotInteraction logs hunting for sessions with cross-site data aggregation patterns (single session accessing more than 5 SharePoint sites) or outbound requests to non-Microsoft domains. Monitor Microsoft's security advisories for new indirect prompt injection disclosures affecting Copilot, as the SearchLeak/EchoLeak research lineage suggests continued researcher focus on this attack surface.
Forensic Artifacts	Microsoft Purview Unified Audit Log — CopilotInteraction record type: contains UserId, session ObjectId, AppAccessContext (internal files/sites accessed per session), and timestamp; primary evidence of whether the SearchLeak three-stage chain (retrieval, injection, exfiltration) completed and what data was in scope | Proxy or DNS logs correlated to Copilot session timestamps: outbound HTTP requests to non-Microsoft domains initiated during authenticated Copilot sessions are the exfiltration-channel artifact; filter by user-agent strings associated with Microsoft 365 Copilot service calls | SharePoint and OneDrive access logs (Purview Audit, SharePoint Unified Logging Service): file-level access events correlated to Copilot session IDs showing which documents were retrieved during a potentially injected session — specific to SearchLeak's internal data aggregation stage | Microsoft Entra ID (Azure AD) sign-in logs and Conditional Access logs for Copilot-licensed users: establishes the authentication context and session token scope active at the time of any anomalous Copilot interaction, relevant because Copilot inherits the full permission scope of the authenticated user | Copilot Studio agent conversation transcripts and plugin execution logs (Power Platform Admin Center > Analytics): for environments using custom Copilot Studio agents, these logs capture the full prompt-response chain including any injected instructions retrieved from attacker-controlled external content

Per-Action IR Details

Step 1: Assess exposure — inventory every Microsoft Copilot deployment across the enterprise, including Microsoft 365 Copilot, Copilot for Security, and any custom Copilot Studio agents; confirm whether each deployment has web search or external content retrieval enabled.

NIST Phase: Preparation

Reference: NIST 800-61r3 §2 — Preparation: establishing visibility into assets and capabilities before or during an incident

Controls: CIS 1.1 (Establish and Maintain Detailed Enterprise Asset Inventory), CIS 2.1 (Establish and Maintain a Software Inventory)

Compensating: Query Microsoft 365 admin center (admin.microsoft.com) under Settings > Copilot to enumerate licensed users and enabled plugins. For Copilot Studio agents, export the environment list via Power Platform Admin Center CLI: `pac admin list --environment`. Cross-reference against Azure AD app registrations to identify any OAuth-connected Copilot extensions with external retrieval scopes. Document in a spreadsheet: deployment name, web search enabled (Y/N), external connector enabled (Y/N), user population.

Evidence: No live-state alteration occurs in this step. Capture the current Copilot configuration state before any changes: export Microsoft 365 admin center Copilot settings screenshots or JSON via Graph API (`GET /beta/admin/serviceAnnouncement/healthOverviews`) as a baseline. Preserve current Entra ID (Azure AD) group-to-Copilot-license assignments so post-change comparisons are possible.

Step 2: Verify patch status — confirm Microsoft's patch for the SearchLeak-specific vector has been applied to all Copilot-enabled tenants; review Microsoft's security update documentation for the specific tenant configuration changes required, if any.

NIST Phase: Eradication

Reference: NIST 800-61r3 §3.4 — Eradication: removing the vulnerability or condition that enabled the incident

Controls: CIS 7.3 (Perform Automated Operating System Patch Management), CIS 7.4 (Perform Automated Application Patch Management), CIS 7.1 (Establish and Maintain a Vulnerability Management Process)

Compensating: Microsoft 365 Copilot is a SaaS service; tenant admins cannot apply patches directly but must verify that Microsoft's server-side fix is active by confirming the tenant is not on a deferred update ring. Check Message Center in Microsoft 365 admin center for the specific advisory tied to SearchLeak (filter by 'Copilot' and 'security'). Confirm no legacy Copilot Studio agent or custom plugin is pinned to a pre-patch API version by reviewing connector API endpoint versions in Power Platform Admin Center.

Evidence: Before confirming eradication, capture Copilot audit logs from Microsoft Purview Compliance Portal (Audit > Search) filtered to CopilotInteraction events for the 30 days preceding patch confirmation — these logs represent the window of potential exploitation and must be preserved before any session or configuration changes that could affect log retention. Export to immutable storage (Azure Blob with WORM policy or equivalent).

Step 3: Audit Copilot data permissions — apply least privilege to Copilot's data access scope per NIST AC-6 (Least Privilege); Copilot inherits the permissions of the authenticated user, meaning over-permissioned accounts create over-permissioned AI agents; reduce Copilot's reachable data to what each user role genuinely requires.

NIST Phase: Containment

Reference: NIST 800-61r3 §3.3 — Containment: limiting the scope of damage and reducing attacker-accessible resources

Controls: NIST AC-6 (Least Privilege), NIST AC-3 (Access Enforcement), CIS 3.3 (Configure Data Access Control Lists), CIS 5.4 (Restrict Administrator Privileges to Dedicated Administrator Accounts)

Compensating: Run Microsoft's SharePoint Permission Reports or use PnP PowerShell (`Get-PnPSiteCollectionAdmin`, `Get-PnPGroupMembers`) to enumerate over-shared SharePoint sites and OneDrive libraries accessible to Copilot-licensed users. Use Microsoft Entra ID Access Reviews (free tier available) to identify accounts with broad Exchange, SharePoint, or Teams permissions that exceed their role definition. Revoke broad sharing links (organization-wide links) on sensitive SharePoint sites that Copilot-licensed users can access, replacing with explicit group-based permissions.

Evidence: Before revoking permissions or modifying sharing configurations, capture a point-in-time export of current SharePoint site access reports and Entra ID group memberships for all Copilot-licensed users. If any Copilot session logs indicate a SearchLeak-style broad data retrieval event (unusually wide scope of files accessed in a single session), preserve those specific CopilotInteraction audit log entries from Microsoft Purview with file/site identifiers intact — this is the blast-radius evidence for any breach notification analysis.

Step 4: Restrict or disable external content retrieval — evaluate whether Copilot's web search and external URL retrieval features are operationally necessary; if not, disable them to eliminate the attacker-controlled content ingestion vector; align this with NIST AC-4 (Information Flow Enforcement) to enforce approved information flow paths.

NIST Phase: Containment

Reference: NIST 800-61r3 §3.3 — Containment: eliminating the attacker-controlled input channel that enabled the indirect prompt injection

Controls: NIST AC-4 (Information Flow Enforcement), CIS 4.4 (Implement and Manage a Firewall on Servers), CIS 4.2 (Establish and Maintain a Secure Configuration Process for Network Infrastructure)

Compensating: Disable Copilot web search via Microsoft 365 admin center: Settings > Search & Intelligence > Configurations > toggle off 'Web content' for Copilot. For Copilot Studio agents, remove external HTTP connector actions and restrict data sources to internal SharePoint/Graph only within the agent's topic configuration. At the network perimeter, use Windows Defender Firewall or a free pfSense instance to block outbound connections from Copilot plugin callback URLs to non-approved external domains — reference Microsoft's published list of Copilot service endpoints to baseline approved egress.

Evidence: Before disabling external retrieval, capture any proxy or DNS logs showing Copilot-session-correlated outbound requests to external domains — these are the exfiltration channel artifacts specific to SearchLeak. In environments with Zscaler, Squid, or similar proxy, filter logs by the user-agent strings associated with Microsoft Copilot service calls and export requests to non-Microsoft domains made during authenticated Copilot sessions. This evidence establishes whether exfiltration occurred and to which destination.

Step 5: Instrument Copilot activity for anomaly detection — enable and review audit logs for Copilot interactions per NIST AU-2 (Event Logging) and AU-6 (Audit Record Review, Analysis, and Reporting); hunt for outbound HTTP requests originating from Copilot sessions to domains outside the organization's approved list, and for sessions accessing unusually broad data scopes relative to the user's normal pattern.

NIST Phase: Detection Analysis

Reference: NIST 800-61r3 §3.2 — Detection and Analysis: establishing monitoring capability and hunting for indicators of SearchLeak-style exploitation

Controls: NIST AU-2 (Event Logging), NIST AU-6 (Audit Record Review, Analysis, And Reporting), NIST AU-3 (Content Of Audit Records), CIS 8.2 (Collect Audit Logs)

Compensating: Enable Microsoft Purview Unified Audit Log for CopilotInteraction events if not already active (requires E3/E5 or Purview add-on). Query via PowerShell: `Search-UnifiedAuditLog -RecordType CopilotInteraction -StartDate -EndDate `. Hunt for sessions where the `AppAccessContext.IsRecursiveAccess` field is true or where a single session retrieves files from more than 5 distinct SharePoint sites — this pattern is consistent with SearchLeak's data-aggregation stage. For outbound exfiltration detection without a SIEM, use Microsoft Defender for Cloud Apps (free 30-day trial) to create a policy alerting on Copilot sessions with external data transfer activity.

Evidence: This is a detection step; preserve all CopilotInteraction audit log entries before any log rotation or retention-policy truncation occurs. Specifically capture: the `UserId`, `ObjectId` (the prompt/response pair identifier), `AppAccessContext` fields showing which internal resources were accessed per session, and the timestamp correlation between Copilot sessions and any outbound DNS/proxy requests to attacker-controlled domains. These log entries are the primary forensic record of whether the SearchLeak three-stage chain (retrieval → injection → exfiltration) completed successfully.

Step 6: Update AI threat model — add indirect prompt injection as a formal threat category in your threat register, mapped to T1059, T1567, and T1190; treat any AI system with retrieval or browsing capabilities as a potential execution environment for attacker-supplied instructions, not just a passive assistant.

NIST Phase: Post Incident

Reference: NIST 800-61r3 §4 — Post-Incident Activity: updating threat models and organizational knowledge based on lessons learned from the SearchLeak disclosure

Controls: CIS 7.1 (Establish and Maintain a Vulnerability Management Process), CIS 7.2 (Establish and Maintain a Remediation Process)

Compensating: Document indirect prompt injection as a threat scenario in a free threat modeling tool such as OWASP Threat Dragon or Microsoft Threat Modeling Tool. Create a one-page threat register entry: threat name 'Indirect Prompt Injection via LLM Retrieval Augmentation', affected assets (all RAG-enabled AI tools), attack narrative referencing SearchLeak/EchoLeak, inherent risk HIGH (CVSS 7.5 reference), residual risk after controls. Note: the MITRE ATT&CK technique IDs referenced in the original step (T1059, T1567, T1190) describe attacker behaviors for threat modeling narrative purposes — they are not controls and are not listed in the controls field.

Evidence: No live-state alteration in this step. Archive the SearchLeak research disclosure, Microsoft's official advisory, and any internal Copilot audit log exports as supporting documentation for the threat register entry. This creates the evidentiary basis for demonstrating due diligence if regulatory inquiry follows.

Step 7: Expand review beyond Microsoft — SearchLeak and EchoLeak establish that this attack class affects any LLM-integrated platform with retrieval capabilities; apply the same data-permission audit, content-source restriction, and logging controls to other AI tools in the environment (Google Workspace AI, Salesforce Einstein, GitHub Copilot, and similar).

NIST Phase: Post Incident

Reference: NIST 800-61r3 §4 — Post-Incident Activity: applying lessons learned to expand controls beyond the immediately affected system to address the broader indirect prompt injection attack class

Controls: CIS 1.1 (Establish and Maintain Detailed Enterprise Asset Inventory), CIS 2.1 (Establish and Maintain a Software Inventory), NIST AC-6 (Least Privilege), NIST AC-4 (Information Flow Enforcement)

Compensating: Build a one-page AI tool inventory: for each tool (Google Workspace AI, Salesforce Einstein, GitHub Copilot, etc.), document whether it has retrieval/browsing enabled, what internal data scopes it can access, and whether audit logging is available. Use this as a repeatable checklist. For GitHub Copilot specifically, review repository access scopes granted to the GitHub Copilot app in GitHub organization settings and restrict to minimum required repos. For Google Workspace AI, audit the 'connected apps' under Google Admin Console > Security > API Controls.

Evidence: No live-state alteration. Before conducting permission audits on each platform, export current permission and configuration states as baseline snapshots — Google Admin SDK reports, Salesforce Connected App OAuth token lists, GitHub Copilot organization policy exports. These baselines document pre-remediation exposure and support any post-audit comparison.

Step 8: Communicate risk to leadership — brief the CISO and relevant stakeholders on the distinction between patch deployment (completed) and category-level risk (ongoing); frame this as an architectural trust-boundary problem in AI integration that requires sustained governance, not a closed incident.

NIST Phase: Post Incident

Reference: NIST 800-61r3 §4 — Post-Incident Activity: communicating lessons learned and residual risk to leadership to drive sustained governance investment

Controls: NIST AC-1 (Policy And Procedures), CIS 7.1 (Establish and Maintain a Vulnerability Management Process)

Compensating: Prepare a one-page executive brief using the following structure: (1) What happened — SearchLeak demonstrated single-click enterprise data exfiltration via Microsoft Copilot using indirect prompt injection; (2) What was done — patch verified, retrieval restricted, permissions scoped, logging enabled; (3) What remains — indirect prompt injection is an unsolved class-level problem affecting all RAG-enabled AI tools; (4) What is needed — an AI governance policy, quarterly permission reviews, and budget for Purview or equivalent audit tooling. Deliver in writing so the risk acceptance or investment decision is documented.

Evidence: No live-state alteration. Attach to the brief: the preserved Copilot audit log exports from Steps 2 and 5, the AI tool inventory from Step 7, and the threat register entry from Step 6. This package constitutes the evidentiary record that the organization detected, contained, and assessed the SearchLeak exposure and supports defensible risk-acceptance documentation if a future incident surfaces the same attack class.

Detection Guidance

Detection for indirect prompt injection attacks on Copilot is challenging because the attack chain operates within legitimate user workflows. Focus hunting on behavioral anomalies rather than known-bad signatures.

Log sources to enable and review:

- Microsoft 365 Unified Audit Log: Copilot interaction logs (CopilotInteraction audit record type), focusing on sessions that result in outbound URL fetches or large data retrievals relative to the initiating query scope. Aligns with NIST AU-2 and AU-6; CIS 8.2 (Collect Audit Logs).

- Proxy and DNS logs: Hunt for outbound HTTP/HTTPS requests initiated within the timeframe of Copilot sessions to domains not on the organization's approved vendor or content list, particularly to newly registered or low-reputation domains. Maps to T1071.001 and T1567.

- DLP telemetry: Review Data Loss Prevention alerts for data egress events correlated with Copilot session activity. Any Copilot-associated transfer of structured internal data (email content, SharePoint documents, Teams messages) to external endpoints warrants investigation.

- Identity and access logs: Audit the permissions of accounts with active Copilot licenses per NIST AC-2 (Account Management); flag accounts with access to sensitive data stores whose Copilot interactions show retrieval patterns inconsistent with their role.

Behavioral hunting hypotheses:

- Copilot session initiates a web retrieval, followed within the same session by an outbound request to a domain that does not match any resource the user's query would logically require.

- Copilot session accesses a significantly larger volume of internal documents or emails than the user's historical baseline for similar query types.

- Outbound URL constructed during a Copilot session contains Base64-encoded or URL-encoded strings, consistent with data serialized for exfiltration per T1567.

- Copilot interaction log shows retrieval of content from a URL that was not explicitly typed by the user, suggesting the model followed an embedded link from ingested content.

D3FEND countermeasures applicable to this threat:

- D3-UAP (User Account Permissions): Restrict Copilot's reachable data to role-appropriate scope.

- D3-MFA (Multi-factor Authentication): Enforce MFA on all accounts with Copilot access, consistent with CIS 6.3 and CIS 6.5, to reduce the value of session hijacking as a secondary vector.

- D3-PBWSAM (Proxy-based Web Server Access Mediation): Route Copilot's external retrieval traffic through an inspecting proxy to detect and block requests to untrusted or unexpected destinations.

- D3-SFA (System File Analysis): Monitor Copilot configuration and tenant policy files for unauthorized changes that could re-enable restricted retrieval features.

Indicators of Compromise

Type	Value	Context	Confidence
URL	`Pending — refer to Dark Reading (darkreading.com/application-security/copilot-searchleak-attack-1-click-data-theft) and the EchoLeak academic paper (arxiv.org/html/2509.10540v1) for any published indicators`	No specific attacker-controlled domains, IP addresses, or payload hashes were extractable from the provided source material; source reporting focuses on the attack technique rather than a specific campaign with attributed infrastructure.	LOW

Framework Mappings

MITRE-ATTACK

- **T1071** — Application Layer Protocol
- **T1567** — Exfiltration Over Web Service
- **T1189** — Drive-by Compromise
- **T1059** — Command and Scripting Interpreter
- **T1071.001** — Web Protocols
- **T1190** — Exploit Public-Facing Application
- **T1566** — Phishing

NIST-800-53R5

- **CA-7** — Continuous Monitoring
- **SC-7** — Boundary Protection
- **SI-4** — System Monitoring
- **CM-7** — Least Functionality
- **SI-3** — Malicious Code Protection
- **SI-7** — Software, Firmware, and Information Integrity
- **CA-8** — Penetration Testing
- **RA-5** — Vulnerability Monitoring and Scanning
- **SI-2** — Flaw Remediation
- **AT-2** — Literacy Training and Awareness
- **SI-8** — Spam Protection
- **SI-10** — Information Input Validation
- **AC-3** — Access Enforcement
- **SC-28** — Protection of Information at Rest

OWASP-TOP10-2021

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

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

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

HIPAA-SECURITY

- **164.312(a)(1)** — Access Control
- **164.308(a)(6)(ii)** — Response and Reporting

SOC2-TSC

- **CC7.4** — Responds to identified security incidents

MITRE ATT&CK Mapping

Technique ID	Technique Name	Tactic
T1071	Application Layer Protocol	Command-And-Control
T1567	Exfiltration Over Web Service	Exfiltration
T1189	Drive-by Compromise	Initial-Access
T1059	Command and Scripting Interpreter	Execution
T1071.001	Web Protocols	Command-And-Control
T1190	Exploit Public-Facing Application	Initial-Access
T1566	Phishing	Initial-Access

Sources

Source	URL	Tier
Security News	https://www.darkreading.com/application-security/copilot-searchleak...	T3
	https://www.darkreading.com/application-security/copilot-searchleak...	T3
	https://www.darkreading.com/application-security/github-confirms-br...	T3
	https://www.darkreading.com/application-security/lessons-ai-hacking...	T3

Source	URL	Tier
EchoLeak: The First Real-World Zero-Click Prompt Injection Exploit ...	https://arxiv.org/html/2509.10540v1	T2