September 18, 2025

ShadowLeak: The First Service-Side Leaking, Zero-click Indirect Prompt Injection Vulnerability

Key Insights:

- **ShadowLeak** is the first service-side leaking, zero-click indirect prompt injection (IPI) vulnerability in ChatGPT, discovered and responsibly disclosed to OpenAI by Radware. OpenAI has confirmed and fixed the vulnerability.
- **Business Impact**: Exposure of PII/PHI, deal data, legal strategy and credentials; potential regulatory violations (GDPR/CCPA/SEC), reputational harm and downstream fraud
- **Zero-click Vulnerability**: No need to click or open a malicious link; ordinary assistant tasks trigger the exfiltration
- **Service-Side Leak**: Sensitive and private data is leaked directly from OpenAI's servers, without being funneled through ChatGPT's client executing on an organization managed device, making the data leak nearly impossible to detect by the impacted organization
- **Scope**: any data source connected to ChatGPT is subject to be leveraged by the agent and its information leaked from OpenAI's infrastructure

Executive Summary

ShadowLeak is a newly discovered zero-click indirect prompt injection (IPI) vulnerability that occurs when OpenAI's ChatGPT is connected to enterprise Gmail and allowed to browse the web. An attack takes advantage of the vulnerability by sending a legitimate-looking email that quietly embeds malicious instructions in invisible or non-obvious HTML. When an employee asks the assistant to "summarize today's emails" or "research my inbox about a topic," the agent ingests the booby-trapped message and, without further user interaction, exfiltrates sensitive data by calling an attacker-controlled URL with private parameters (e.g., names, addresses, and internal and sensitive information). It's important to note that the web request is performed by the agent executing in OpenAI's cloud infrastructure, causing the leak to originate directly from OpenAI's servers. Unlike earlier disclosed indirect prompt injection vulnerabilities, the malicious request and the private data never pass through the ChatGPT client. As a result, the affected organization is left with no clear traces to monitor and no forensic evidence to analyze at the organization's boundary.

This class of exploit aligns with the broader risks described in the emerging Internet of Agents—autonomous, tool-using AI that act across protocols and services. As organizations wire these assistants into mailboxes, CRMs, HR systems and SaaS, the business risk shifts from “what the model says” to “what the agent does.”

How ShadowLeak Works

An attacker sends a legitimate-looking email to a corporate mailbox. Inside the email body, hidden in tiny fonts, white-on-white text or formatting metadata, sit instructions for the assistant—not for the employee. When the employee later asks the assistant to analyze or summarize their inbox, the agent ingests the booby-trapped message and performs “authorized” actions: it follows a link, calls an external URL that the attacker controls and appends private details pulled from recent emails—names, addresses, identifiers—into the query string. No one clicks a malicious link; there’s no attachment to analyze in a sandbox. The agent directly leaks the data while completing a routine task.

The zero-click property makes the vulnerability impossible to prevent. Traditional awareness training assumes a human decision point—a moment to pause, inspect and refuse. ShadowLeak removes that moment. It abuses capabilities you deliberately enabled to boost productivity: mailbox access, web browsing and autonomous tool use. From the outside, the traffic looks like sanctioned assistant activity. From the inside, guardrails focused on safe output don’t catch what really matters here—covert, tool-driven actions.

The service-side leaking is the crux of the vulnerability. Because there is no malicious URL or sensitive data passed to the ChatGPT client and the client does not perform the web request, there is no data leaking from an organization’s boundary, leaving the organization blind to the event.

Phishing the Assistant

The attacker’s craft is as much **social engineering for machines** as it is for people. In repeated runs, the attack worked roughly half the time with a simple instruction and a plain exfiltration URL, such as https://hr-service.net/{params}. A determined adversary using better prompts and a domain reflecting the malicious prompt’s intent can get much better results. In testing, success rates improved considerably when urgency was added to the prompt instruction and the exfiltration endpoint was made to resemble a compliance check with an employee directory lookup endpoint: https://compliance.hr-service.net/public-employee-lookup/{params}. The agent’s internal reasoning now treats the malicious prompt as part of an urgent HR compliance task.

Why It Matters

ShadowLeak weaponizes the very capabilities that make AI assistants useful: email access, tool use and autonomous web calls. It results in silent data loss and unlogged actions performed “on behalf of the user," bypassing traditional security controls that assume intentional user clicks or data leakage prevention at the gateway level.

Beyond Email

To understand why ShadowLeak matters beyond email, consider the architecture that underpins modern assistants. Enterprises are moving from single chatbots to networks of goal-driven agents that read mail, consult tools, call APIs and increasingly collaborate with other agents. Protocols such as Anthropic's Model Context Protocol (MCP) and Google's Agent-to-Agent (A2A) standardize interactions between agents and agent systems. MCP gives agents hands to operate and access external data; A2A gives them a language to coordinate with each other. With MCP, the AI becomes your system administrator; with A2A, it's your emissary. Combine them and you have a roaming delegate that negotiates and acts across systems you don't fully control. Productivity might soar, but predictability, auditability and control will fall unless you redesign your governance to match.

This is the **Internet of Agents**, and it is already here. It behaves less like a set of endpoints and more like a mesh of autonomous actors. Identity becomes fluid, instructions are opaque, and authority is transitive and distributed. In such an environment, prompt injections don't just bend a single response; they can cascade. Compromise an agent with a hidden instruction, nudge it to invoke a poisoned tool via MCP, and you create a chained exploit that looks benign in any single log line but malignant in composition. The uncomfortable scenario isn't theoretical. Chained compromise, misaligned agent objectives and even zero-click lateral movement across an A2A mesh are precisely the risk patterns security research has begun to document.

ShadowLeak is a vivid example of the **threat you don't see**. It leverages the assistant's sanctioned powers to create business impact. For HR and Legal, that could mean silent leakage of PII or case strategy during routine mailbox digests. For Finance, invoice and pricing details exfiltrated as part of month-end prep. For Sales, pipeline summaries that quietly include account data in outbound requests. For IT and SecOps, tokens and secrets that slipped into a thread and then slipped out to an attacker endpoint—no alarms, just "normal" assistant behavior. The pattern is consistent: The damage arises not from a bad sentence on the screen but from a behind-the-scenes action your policies didn't explicitly constrain.

A New Threat Surface

If this sounds eerily familiar, it's because we've seen the early contours already. Assistants now summarize emails, browse the web and coordinate with other agents as a matter of course. In that world, every piece of content becomes a potential trigger, every tool a potential exfil path and every agent a potential amplifier.

Prior work on zero-click IPIs in office assistants showed how hidden instructions inside content could trigger unintended actions without user interaction. Unlike prior disclosed vulnerabilities, however, ShadowLeak does not pass the malicious URL and private data to the client to make the web request on behalf of the agent by leveraging image rendering. In ShadowLeak, the web request is made directly from the agent running in OpenAI's cloud infrastructure. This makes ShadowLeak stealthier and almost impossible to detect by the affected organization compared to earlier disclosed vulnerabilities.

Recommendations

Start by reframing assistants as privileged actors, not chat features. If a human with the same mailbox and access to sensitive information would be subject to strict controls, so should the agent. Sanitize all inbound HTML before LLM ingestion so hidden instructions are stripped or neutralized; flatten documents to safe text where possible. Then turn on lights in the black box by logging every agent action with who/what/why metadata, so you can investigate and, crucially, deter.

It is, however, important to note that traditional detection methods don't work on malicious AI prompts. Classic SQL or command injection exploits target tightly structured programming languages, so their payloads follow a limited set of detectable patterns. Prompt attacks, by contrast, are written in natural language with virtually endless ways to express the same intent. Spotting that intent requires semantic analysis from an LLM or equivalent—not the regular-expression or state-machine engines many tools rely on today.

Even strict monitoring and whitelisting of external URLs by the organization will not mitigate this new attack vector. Because the web request is made from OpenAI's infrastructure, there is no network traffic with private data leaving the network of the organization itself. Earlier research by Aim Labs on the EchoLeak vulnerability demonstrated that rendering an agent-provided Markdown image in the ChatGPT client resulted in private data being funneled through a Microsoft Teams URL that sat on the "safe" allowlist of an organization. Such request could also sail past egress controls because the Teams host was trusted, and the open redirect then forwarded the full string—secrets and all—to the attacker's server. In ShadowLeak, however, the ChatGPT client within the organization's infrastructure never takes a suspicious action. It is the agent executing in the context of OpenAI's infrastructure that makes the call and, by consequence, leaks the data outside of the control and visibility of the organization.

Governance will have to catch up with capability. For example, update supplier questionnaires and contract processes to require prompt-injection resilience testing and sanitization upstream of model ingestion. Segment assistant permissions so that "reading" and "acting" are different scopes and different service accounts. Build a maturity model for autonomy that starts with read-only analysis and elevates to supervised actions only after security review. Then red-team the whole stack with zero-click IPI playbooks and poisoned tool tests before you give a green light to broad deployment. The goal isn't to block agentic AI; it's to deploy it safely with the same discipline you apply to any system that can move money, touch customer data or change state in your environment.

Finally, accept that offense is learning fast. Dark-market AIs have professionalized, and thousands of MCP endpoints now expose toolchains that can be nudged or poisoned if you're not paying attention. The defensive posture that works in this new terrain is the one that treats model prompts and outputs like untrusted input, instrument agents like production systems and assumes attackers will chain protocols the same way your architects chain services. The businesses that pull ahead will be those that embrace the agent economy while insisting on security as a feature, not an afterthought. In other words, make the assistant useful—but make it accountable.

Summary

ShadowLeak is a wake-up call, not a reason to turn the lights off. The promise of assistants that lift the weight of email drudgery, knowledge search and workflow glue is real and worth pursuing. The way to get there is straightforward, if not simple: treat agents as privileged actors, harden the inputs they see, constrain the outputs they can send and hold them to the same standards of logging, review and governance as any other critical system or user. Do that, and you can keep the productivity—and your data.

EFFECTIVE DDOS PROTECTION ESSENTIALS

Hybrid DDoS Protection – Use on-premises and cloud DDoS protection for real-time DDoS attack prevention that also addresses high-volume attacks and protects from pipe saturation

Behavioral-Based Detection – Quickly and accurately identify and block anomalies while allowing legitimate traffic through

Real-Time Signature Creation – Promptly protect against unknown threats and zero-day attacks

Web DDOS Tsunami Protection – Automated immediate detection and mitigation of Web DDOS encrypted high RPS and morphing attacks

A Cybersecurity Emergency Response Plan – Turn to a dedicated emergency team of experts who have experience with Internet of Things security and handling IoT outbreaks

Intelligence on Active Threat Actors – High fidelity, correlated and analyzed data for preemptive protection against currently active known attackers

For further network and application protection measures, Radware urges companies to inspect and patch their network to defend against risks and threats.

EFFECTIVE WEB APPLICATION SECURITY ESSENTIALS

Full OWASP Top-10 coverage against defacements, injections, etc.

Low false positive rate using negative and positive security models for maximum accuracy

Auto-policy generation capabilities for the widest coverage with the lowest operational effort

Bot protection and device fingerprinting capabilities to overcome dynamic IP attacks and achieve improved bot detection and blocking

Securing APIs by filtering paths, understanding XML and JSON schemas for enforcement, and using activity tracking mechanisms to trace bots and guard internal resources

Flexible deployment options including on-premises, out-of-path, virtual or cloud-based

LEARN MORE AT RADWARE'S SECURITY RESEARCH CENTER

To know more about today's attack vector landscape, understand the business impact of cyberattacks, or learn more about emerging attack types and tools, visit Radware's Security Research Center. Additionally, visit Radware's Quarterly DDoS & Application Threat Analysis Center for quarter-over-quarter analysis of DDoS and application attack activity based on data from Radware's cloud security services and threat intelligence.

THIS REPORT CONTAINS ONLY PUBLICLY AVAILABLE INFORMATION, WHICH IS PROVIDED FOR GENERAL INFORMATION PURPOSES ONLY. ALL INFORMATION IS PROVIDED “AS IS” WITHOUT ANY REPRESENTATION OR WARRANTY OF ANY KIND, EITHER EXPRESS OR IMPLIED, INCLUDING WITHOUT LIMITATION, ANY IMPLIED WARRANTIES THAT THIS REPORT IS ERROR-FREE OR ANY IMPLIED WARRANTIES REGARDING THE ACCURACY, VALIDITY, ADEQUACY, RELIABILITY, AVAILBILITY, COMPLETENESS, FITNESS FOR ANY PARTICULAR PURPOSE OR NON-INFRINGEMENT. USE OF THIS REPORT, IN WHOLE OR IN PART, IS AT USER’S SOLE RISK. RADWARE AND/OR ANYONE ON ITS BEHALF SPECIFICALLY DISCLAIMS ANY LIABILITY IN RELATION TO THIS REPORT, INCLUDING WITHOUT LIMITATION, FOR ANY DIRECT, SPECIAL, INDIREC, INCIDENTAL, CONSEQUENTIAL, OR EXAMPLARY DAMAGES, LOSSES AND EXPENSES ARISING FROM OR IN ANY WAY RELATED TO THIS REPORT, HOWEVER CAUSED, AND WHETHER BASED ON CONTRACT, TORT (INCLUDING NEGLIGENCE) OR OTHER THEORY OF LIABILITY, EVEN IF IT WAS ADVISED OF THE POSSIBILITY OF SUCH DAMAGES, LOSSES OR EXPENSES. **CHARTS USED OR REPRODUCED SHOULD BE CREDITED TO RADWARE**