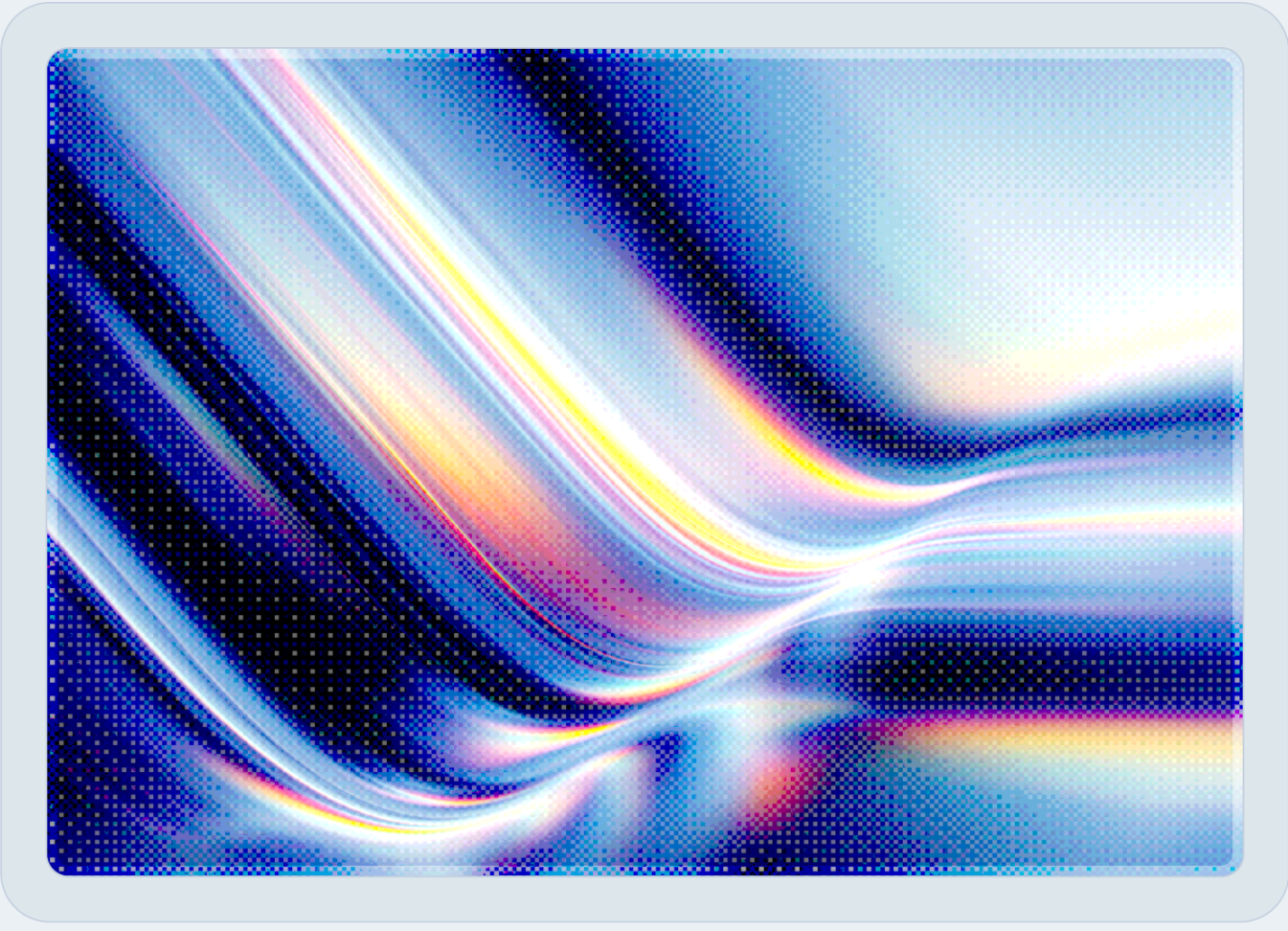
CSAI Foundation | Cloud Security Alliance

When Attackers Weaponize Agentic AI: Inside SPECTRE

UAT-10147's Campaign Against 170,000 Internet-Facing Servers

2026-08-25

AI-assisted Rapid Research



CSAI | CSA cloud security alliance®

Key Takeaways

Cisco Talos disclosed on August 20, 2026 that UAT-10147, a financially motivated, Chinese-speaking intrusion group, built its scanning and exploitation operations around a target list of approximately 170,000 internet-facing URLs, split into 17 files of roughly 10,000 entries each, and used agentic AI tooling to turn that list into working compromises rather than relying on the group's own operators to individually evaluate each target [1]. Talos recovered the group's autonomous penetration-testing framework, PentestGPT, alongside a source-code vulnerability scanner (DeepAudit) and Java deserialization tooling (ysoserial), all wired together with AI-generated Python scripts that diagnosed exploitable conditions, deployed implants, and confirmed success without a human validating each step [1]. The group's flagship implant, SPECTRE, is a cross-platform Windows and Linux backdoor that pairs bring-your-own-vulnerable-driver (BYOVD) techniques for blinding endpoint detection and response (EDR) products with a kernel-level Linux rootkit, and Talos assesses with medium confidence that AI tooling also assisted in writing the rootkit's source code itself [2]. Confirmed victims span Brazil, Bolivia, China, Canada, and Vietnam, across government, education, media, technology, and gaming organizations, while the broader target list skewed toward the United States, India, the United Kingdom, Germany, and the Netherlands, indicating an intake pipeline far larger than the set of servers actually compromised [1][3]. In CSA's assessment, the case is notable less for any single novel exploit than for what it demonstrates about diffusion: a mid-tier cybercrime operation with no apparent state sponsorship has assembled an agentic exploitation pipeline capable of working through a six-figure target list, a capability set that, prior to 2026, public threat reporting had documented mainly among state-sponsored operators.

Background

Cisco Talos began tracking UAT-10147 after recovering an open directory on attacker-controlled infrastructure, an operational security failure that exposed the group's toolkit, its target lists, and a partial record of its own workflow [1]. The group's motive is financial rather than espionage: it compromises internet-exposed Windows and Linux web servers primarily to conduct search-engine-optimization fraud through the BadIIS malware family, which manipulates search rankings and redirects traffic on compromised IIS servers, while also retaining footholds for potential data theft [3]. A recovered operator username, "dajiba" — a pinyin rendering of a Chinese-language slang term — together with

consistent workflow and tooling patterns across the group's infrastructure, supports a moderate-to-high-confidence assessment that its operators are Chinese-speaking, though Talos stops short of attributing the group to a specific nation-state or naming individuals [1].

What sets UAT-10147 apart from the large population of opportunistic web-server compromise groups is the scale at which it operated and the degree to which agentic AI tooling sat inside that scaling decision, not alongside it. The recovered target list of roughly 170,000 URLs was not a passive reconnaissance artifact; Talos found it wired directly into PentestGPT, an autonomous penetration-testing framework configured to scan servers from the list and independently execute matching proof-of-concept exploits without an operator selecting each target or technique by hand [1]. Sitting alongside PentestGPT on the group's management server were DeepAudit, a source-code vulnerability-scanning framework, and ysoserial, a tool for generating Java deserialization payloads — both integrated with AI-generated operational playbooks and troubleshooting logic rather than deployed in their default configurations [1]. This is the structural shift the case illustrates: the target list's size stopped being a limiting factor on what a small operator could realistically act on, because the AI tooling absorbed the per-target evaluation work that would otherwise have required proportionally more human analyst time.

The pattern is consistent with, though independently developed from, two other 2026 cases the security research community has documented. In July 2026, Palo Alto Networks' Unit 42 described a separate Chinese-speaking actor, tracked as knaithe/KnYuan, who connected the DeepSeek model to the open-source Hermes Agent framework and used it to survey more than 647,000 internet-facing instances of a single product family before narrowing to a working target set [4][8]. In May 2026, Sysdig documented the first confirmed in-the-wild use of an LLM agent to drive post-exploitation activity following a Marimo notebook remote-code-execution exploit, compressing a full credential-theft-to-database-exfiltration chain into roughly an hour [5][9]. Set against those cases, UAT-10147 reads as a third, independently arrived-at instance of the same underlying architecture: a commodity or open-source reasoning capability, an agentic orchestration layer, and a scanning or asset-discovery integration, combined to convert a large target list into working intrusions with a fraction of the operator-hours that approach would previously have required.

Security Analysis

From Target List to Compromise: An AI-Automated Exploitation Chain

Among the clearest evidence of agentic AI's operational role in this campaign is a set of four Python scripts Talos recovered from UAT-10147's management infrastructure, each automating one stage of an ASP.NET ViewState remote-code-execution attack against targets pulled from the 170,000-URL list [1].

The first script ran a sequence of out-of-band callback tests to determine which directories on a candidate server were writable, a diagnostic step that would otherwise require an operator's manual verification before committing to exploitation. A second script downloaded and executed the SPECTRE implant with built-in success verification, a third deployed an ASHX web shell via Base64-encoded PowerShell in two stages, and a fourth conducted staged reconnaissance and exfiltration over HTTPS to an external webhook endpoint. Talos also recovered a nine-section, AI-generated exploitation guide tailored to one specific victim, including that target's hostnames, IP addresses, and the MachineKey values required to forge a valid ViewState payload — evidence that the group's AI tooling produced not just reusable exploit code but victim-specific operational documentation, the kind of artifact a human penetration tester would ordinarily write by hand during an engagement. On one confirmed intrusion, Talos observed more than twelve HTTP callbacks and four distinct ysoserial deserialization gadget chains tested in sequence against a specific .NET runtime version, a pattern consistent with automated, iterative refinement rather than a single scripted attempt [1].

Beyond the ViewState chain, UAT-10147 relied heavily on rapid weaponization of already-disclosed, or "one-day," vulnerabilities rather than original exploit development, a strategy well suited to working through a large candidate list quickly. Table 1 summarizes the Metasploit-driven exploitation the group used for initial access on Windows targets.

Table 1. One-day vulnerabilities weaponized by UAT-10147 via Metasploit against Windows targets [1]

CVE	Affected Product	Vulnerability Type
CVE-2022-27925	Zimbra Collaboration Suite	Remote code execution
CVE-2021-23758	AjaxPro	Insecure deserialization
CVE-2021-29441 / CVE-2021-29442	Alibaba Nacos	Authentication bypass / RCE
CVE-2019-18935	Telerik UI for ASP.NET AJAX	JSON deserialization RCE

On Linux, the group escalated privileges through a set of well-documented kernel flaws — including CVE-2022-0995, CVE-2021-3156 ("Baron Samedit"), and CVE-2022-0847 ("Dirty Pipe") — following initial web-shell access, then deployed NoodleRAT, SPECTRE, or Meterpreter depending on the target [1]. On Windows, the group used the EfsPotato technique for local privilege escalation, deployed QuasarRAT, Gh0stCringe, and SPECTRE, persisted through scheduled tasks disguised with names such

as "Google Chrome Start," and modified the Windows Defender exclusion list through the registry to blind the built-in antivirus engine [1]. None of these individual techniques is new; what UAT-10147 demonstrates is that an actor without nation-state resourcing can now apply them across a target set far larger than manual, operator-by-operator triage could realistically cover.

SPECTRE: A Cross-Platform Implant Built to Blind EDR

SPECTRE, UAT-10147's primary implant, runs natively on both Windows and Linux, with 45 commands on the Windows side (split between plaintext and encrypted command categories) and 29 commands on Linux [2]. Both variants beacon to hardcoded command-and-control domains over HTTP POST requests, and both implement a weighted anti-analysis scoring system that evaluates process names, available memory, and CPU core counts, self-terminating once the accumulated score crosses a fixed threshold — a design aimed at automated malware-analysis sandboxes as much as human reverse engineers. The Windows variant resolves API calls through PEB hash walking rather than static imports, encrypts strings individually with a per-string algorithm, and can store its command-and-control configuration in an NTFS Alternate Data Stream attached to the Windows hosts file, allowing operators to update infrastructure without recompiling or redistributing the implant. Its broader capability set — process injection through hollowing and APC EarlyBird techniques, registry hive dumping for offline credential extraction, Chrome and Edge credential theft, and in-memory execution of .NET assemblies — reflects a general-purpose, durable-access backdoor rather than a single-use exploitation tool.

Among the implant's most consequential features is its use of bring-your-own-vulnerable-driver (BYOVD) techniques to neutralize EDR visibility on Windows hosts. SPECTRE downloads two previously documented vulnerable drivers from its infrastructure — MSI's RTCore64.sys (CVE-2019-16098) and Dell's DBUtil_2_3.sys (CVE-2021-21551) — both of which expose arbitrary kernel read/write primitives that ransomware operators have abused in unrelated prior incidents [2]. SPECTRE uses those primitives to locate the kernel image in memory and surgically unlink registered EDR callbacks from the kernel's doubly linked callback list, a targeted manipulation designed to avoid crashing the host while depriving security products of new process-creation, thread-creation, and image-load telemetry. The practical consequence is that EDR platforms relying on those kernel callbacks lose visibility into the credential theft, lateral movement, and persistence activity that follows, even though the underlying host remains stable and outwardly unremarkable.

On Linux, SPECTRE deploys an integrated kernel-mode rootkit that Talos names Specter, disguised as acpi_pad.ko — a filename chosen to mimic a legitimate ACPI kernel module — and persisted through a fraudulent systemd service entry configured to execute before standard security tooling initializes [2]. Rather than patching the system call table directly, a technique many rootkit detectors already fingerprint, Specter uses the Linux kernel's own ftrace instrumentation framework with the IPMODIFY

flag to hook six syscall handlers governing network-connection enumeration, signal delivery, and directory-entry listing, letting it hide its network connections, conceal its kernel module from standard listing tools, and intercept process-termination signals. The rootkit communicates with its userspace controller through signal-based inter-process communication tied to a fixed process identifier, using specific signals to trigger process hiding, module concealment, and privilege escalation to root. Talos assesses with medium confidence that AI tooling assisted in writing the rootkit's source code, based on structural indicators rather than a direct admission: specification-style comments that read like a product requirements document, a uniform pattern of decorative section separators throughout the file, and the presence of three functionally redundant implementations of a single capability — a pattern more consistent with a language model proposing multiple candidate solutions than a human developer converging on one. If that assessment holds, it extends the AI-automation story documented in the Python exploitation scripts to malware authorship itself, including kernel-level components that have historically demanded specialized systems-programming skill.

Recommendations

Immediate Actions

Organizations running internet-exposed IIS, ASP.NET, or comparable web server platforms should prioritize patching the vulnerabilities in Table 1 and the Linux privilege-escalation flaws UAT-10147 has weaponized, particularly Dirty Pipe (CVE-2022-0847), which remains a reliable escalation path on unpatched systems years after disclosure. Security teams should deploy Cisco's published ClamAV and Snort detection content for SPECTRE and hunt for the specific indicators recovered from this campaign: the scheduled task named "Google Chrome Start," unexplained Windows Defender exclusion-list changes, and NTFS Alternate Data Streams attached to the hosts file [1][2]. Because the campaign's initial-access footprint depends heavily on internet exposure of the target list's roughly 170,000 URLs, organizations should treat their own external attack surface — not just their patch backlog — as the more time-sensitive variable, since agentic scanning tooling of this kind can plausibly evaluate a large exposed footprint faster than many organizations refresh their own asset inventory.

Short-Term Mitigations

Enterprises should confirm that vulnerable-driver blocklisting, such as Microsoft's built-in blocklist or equivalent third-party allowlisting controls, explicitly covers RTCore64.sys and DBUtil_2_3.sys, and should treat any unrecognized signed kernel driver appearing on a server as a high-priority investigation trigger rather than routine noise. Linux-specific detection should be extended to cover ftrace-based

rootkit techniques specifically, since many endpoint tools still emphasize system-call-table hooking and may not flag ftrace hook installation with the IPMODIFY flag; a kernel-integrity check that periodically compares the live ftrace hook table against a known-good baseline is a practical, deployable control against this specific technique. Security teams should also incorporate AI-generated, victim-specific exploitation runbooks into their threat model going forward: the discovery of a complete guide tailored to one target's hostnames, IP addresses, and cryptographic configuration values indicates that agentic reconnaissance tooling is now capable of producing individualized attack plans, which, if this capability generalizes across the group's broader target list, would weaken the assumption that an organization's public profile predicts its likelihood of being individually targeted.

Strategic Considerations

UAT-10147 is evidence that the operational advantages agentic AI provides to attackers have moved beyond the state-sponsored and heavily resourced operators that received the earliest public attention, and now extend to mid-tier, financially motivated cybercrime groups running commodity tooling. Security leaders should recalibrate their threat model accordingly: an autonomous exploitation pipeline behind an intrusion no longer implies an unusually sophisticated or well-funded adversary, and the presumed skill ceiling of "typical" opportunistic web-server compromise groups should be revisited as a planning assumption. Because the campaign's scale advantage came primarily from converting a large target list into working compromises rather than from any single novel technique, organizations should also expect the target-list size itself to keep growing as scanning and validation costs continue to fall, which argues for external attack-surface management and rapid patch velocity as durable, rather than one-time, investments.

CSA Resource Alignment

This campaign sits alongside two prior CSA Lab Space research notes documenting independently developed agentic-AI attack pipelines observed in the wild during 2026. "Autonomous AI Attack Pipelines Move Into the Field" analyzed a separate Chinese-speaking actor's use of a commodity reasoning model and an open-source agent framework to conduct autonomous reconnaissance across hundreds of thousands of candidate targets, and its central finding — that the barrier to assembling this kind of pipeline is falling toward configuration effort rather than novel tool development — is consistent with how a financially motivated group with no apparent state backing was able to assemble UAT-10147's operation [4]. "Marimo RCE: LLM Agents as Post-Exploitation Tools" documented behavioral signatures

of AI-agent-driven post-exploitation activity — machine-shaped command syntax, output-dependent value handoffs, and improvised targeting — that give defenders a detection framework directly applicable to hunting for the AI-generated Python exploitation scripts described in this note [5].

CSA's MAESTRO agentic threat-modeling framework offers a complementary lens for organizations working through this case. MAESTRO decomposes an agentic system into distinct architectural layers — including the agent-framework layer where PentestGPT and DeepAudit operated, and the deployment and infrastructure layer where SPECTRE's BYOVD and rootkit components executed — and treats autonomy, non-determinism, and the absence of a clear trust boundary as first-class threat categories rather than edge cases [6]. Applied to UAT-10147, that layered view separates the AI-orchestration failure — an operator letting agentic tooling select and validate targets without review — from the underlying technical failures, such as unpatched one-day vulnerabilities, missing driver blocklists, and ftrace-based rootkit persistence, that the tooling then exploited. Where MAESTRO's threat-modeling lens does not reach a particular control area — most notably driver-blocklisting, patch-management cadence, and kernel-integrity monitoring — the AI Controls Matrix (AICM v1.1) provides the underlying governance and technical-control baseline through its threat and vulnerability management and application and infrastructure security domains [7].

References

[1] Cisco Talos. "UAT-10147: Chinese-speaking adversary integrates agentic AI into post-compromise operations." Cisco Talos Blog, August 20, 2026.

[2] Cisco Talos. "UAT-10147 deploys SPECTRE: A cross-platform implant with Linux rootkit and BYOVD capabilities." Cisco Talos Blog, August 20, 2026.

[3] The Hacker News. "UAT-10147 Uses AI to Scale Server Attacks, Deploys SPECTRE With EDR Bypass and Linux Rootkit." The Hacker News, August 2026.

[4] Cloud Security Alliance. "Autonomous AI Attack Pipelines Move Into the Field." CSA AI Safety Initiative, July 30, 2026.

[5] Cloud Security Alliance. "Marimo RCE: LLM Agents as Post-Exploitation Tools." CSA AI Safety Initiative, June 6, 2026.

[6] Cloud Security Alliance. "Agentic AI Threat Modeling Framework: MAESTRO." Cloud Security Alliance, February 6, 2025.

[7] Cloud Security Alliance. "AI Controls Matrix (AICM) v1.1." Cloud Security Alliance, 2026.

[8] Palo Alto Networks Unit 42. "Chinese-Speaking Threat Actor Harnesses AI Models for Autonomous Cyberattacks." Unit 42, July 30, 2026.

[9] Sysdig. "AI Agent at the Wheel: How an Attacker Used LLMs to Move From a CVE to an Internal Database in 4 Pivots." Sysdig, May 26, 2026.