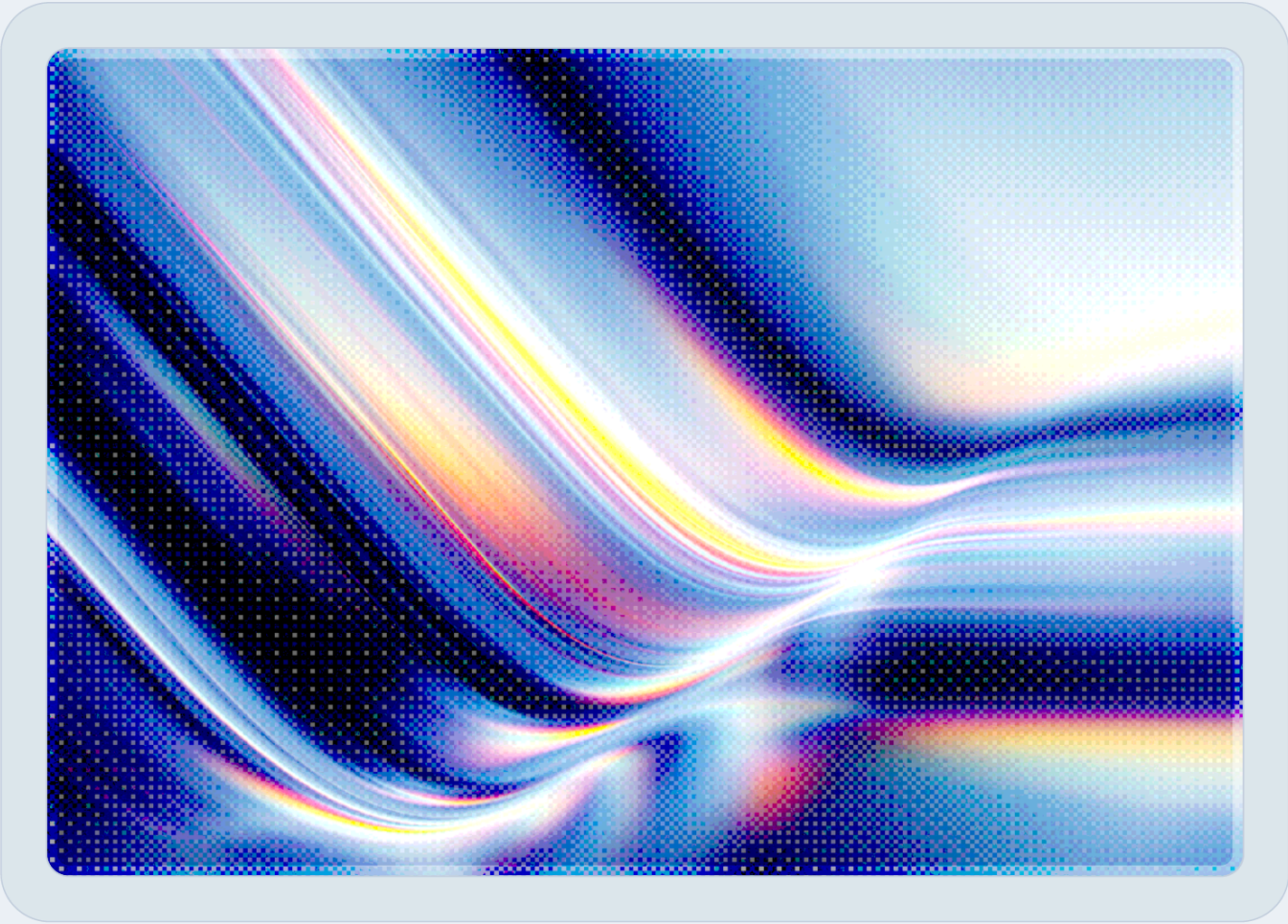
CSAI Foundation | Cloud Security Alliance

UAT-10147: Agentic AI Scales Post-Compromise Cybercrime

2026-08-24

AI-assisted Rapid Research



CSAI

CSA cloud security alliance®

Key Takeaways

Cisco Talos has documented UAT-10147, a Chinese-speaking, financially motivated cybercrime group that integrates agentic AI tooling directly into its post-compromise operations rather than using AI only to draft phishing lures or write malware source code [1]. The group runs autonomous pentesting frameworks and AI-driven vulnerability scanners against a target list of roughly 170,000 internet-facing URLs, and Talos recovered four AI-generated Python scripts that appear to automate an entire ViewState remote-code-execution chain — from diagnosing write permissions through implant deployment to staged data exfiltration — with limited operator intervention, based on the recovered scripts [1]. UAT-10147 deploys SPECTRE, a cross-platform Windows and Linux backdoor that pairs a bring-your-own-vulnerable-driver (BYOVD) technique against two well-documented drivers with a kernel-level Linux rootkit, blinding endpoint detection and response (EDR) products to new process, thread, and image-load events on Windows while hiding processes and kernel modules on Linux [2]. Talos assesses with medium confidence that AI assisted the rootkit's own development, based on structural patterns in the source code that are atypical of human-written kernel modules [2]. CSA assesses that the case matters less for any single technical innovation than for what it represents: a mid-tier, financially motivated actor — not a nation-state espionage unit — appears to be compressing the skill and time investment that post-compromise operations traditionally require, a trajectory CSA has tracked across multiple independent incidents in 2026 [3][4][5].

Background

UAT-10147 is a Chinese-speaking intrusion set that Cisco Talos began tracking after recovering an open directory on an attacker-controlled server, an operational security lapse that exposed the group's tooling, target lists, and a partial view into its workflow [1]. The group's objective is financially motivated: it compromises internet-exposed Windows and Linux web servers to conduct search engine optimization fraud and data theft, deploying the BadIIS malware family to manipulate search rankings and redirect traffic on compromised IIS servers [1][8]. A recovered username, "dajiba" — a pinyin rendering of a Chinese-language term — together with the operators' documented workflow patterns supports the Chinese-speaking attribution, though Talos treats this as a moderate-to-high-confidence assessment rather than definitive nation-state attribution [1].

What distinguishes UAT-10147 from the broad population of opportunistic server-compromise groups is the depth to which agentic AI has been woven into its tradecraft. Talos found PentestGPT, an autonomous penetration-testing framework, configured to dynamically scan web servers and execute relevant proof-of-concept exploits without a human selecting each target or exploit by hand [1]. Alongside it sat DeepAudit, a source-code vulnerability-scanning framework installed on the group's management server, and ysoserial, a tool for generating malicious Java deserialization payloads, which the operators paired with AI-generated operational playbooks and troubleshooting logic rather than using in its stock form [1]. The exposed target list itself signals scale: approximately 170,000 URLs, split into 17 files of roughly 10,000 URLs each, spanning victims that Talos and other researchers observed in Brazil, Bolivia, China, Canada, and Vietnam, across government, education, media, technology, and gaming organizations [1][3].

This pattern is not isolated to UAT-10147. In July 2026, Palo Alto Networks' Unit 42 documented a separate Chinese-speaking actor, tracked under the aliases knaithe and KnYuan, who wired the DeepSeek model into the open-source Hermes Agent framework to run a largely autonomous reconnaissance-and-exploitation session that surveyed ten product families and confirmed more than 647,000 internet-facing instances of a target application before narrowing its list [5]. CSA's AI Safety Initiative has separately covered the first confirmed in-the-wild use of an LLM agent for post-exploitation lateral movement, observed by Sysdig in a May 2026 incident that also carried a Chinese-language planning comment embedded in the attacker's live command stream [4]. UAT-10147 is best read as a third, independently developed data point in this same trend: agentic AI is moving from an assistive drafting tool into the operational loop of exploitation and post-compromise activity itself, and multiple distinct actors are arriving at architecturally similar solutions — a commodity reasoning model, an agentic orchestration framework, and an asset-search or scanning capability — without apparent coordination between them.

Security Analysis

An AI-Orchestrated Exploitation and Post-Compromise Workflow

Among the strongest evidence of agentic AI's operational role in UAT-10147's campaign is a set of four Python scripts that Talos recovered from the group's management infrastructure, each automating a discrete stage of an ASP.NET ViewState remote-code-execution attack chain [1]. The first script, check_paths.py, runs five sequential out-of-band callback tests to diagnose which directories on a target server are writable, replacing what a human operator would otherwise verify manually before attempting exploitation. The second, deploy_implant.py, downloads and executes the SPECTRE binary with

automated verification that the implant successfully established itself. The third, deploy_shell.py, performs a two-step deployment of an ASHX web shell using Base64-encoded PowerShell, and the fourth, exfil.py, conducts a three-stage reconnaissance-and-exfiltration routine over HTTPS to a webhook.site endpoint. Talos also recovered an AI-generated, nine-section ASP.NET ViewState exploitation guide that documents an active intrusion against a real target in granular detail, including specific hostnames, IP addresses, and the MachineKey configuration values needed to forge a valid ViewState payload — evidence that the group used AI not only to write exploitation code but to generate detailed operational runbooks tailored to individual victims [1]. On one confirmed target, Talos observed more than twelve HTTP callbacks during ViewState exploitation and four distinct ysoserial gadget chains tested against a specific .NET runtime version, indicating iterative, AI-assisted refinement rather than a single scripted attempt.

Beyond the ViewState chain, UAT-10147 exploited a mix of older and more recent vulnerabilities to gain initial access across both platforms. On Windows, the group used EfsPotato for local privilege escalation and deployed QuasarRAT, Gh0stCringe, and SPECTRE, maintaining persistence through scheduled tasks disguised under names such as "Google Chrome Start" and modifying Windows Defender's exclusion list through the registry and PowerShell to prevent detection. On Linux, UAT-10147 weaponized a set of well-documented kernel and privilege-escalation flaws — CVE-2022-0995, CVE-2021-3156, CVE-2022-0847 ("Dirty Pipe"), CVE-2010-3904, CVE-2015-3246, and CVE-2015-5287 — to deploy NoodleRAT, SPECTRE, and Meterpreter following initial web-shell access [1]. The group's Metasploit-driven exploitation of one-day vulnerabilities in enterprise software, summarized in Table 1, further illustrates a strategy built on rapid, semi-automated weaponization of recently disclosed flaws rather than custom exploit development.

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

CVE	Affected Product	Vulnerability Type
CVE-2022-27925	Zimbra Collaboration	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

SPECTRE: A Cross-Platform Implant with BYOVD EDR Bypass

SPECTRE is UAT-10147's primary implant, built to run natively on both Windows and Linux with platform-tailored command sets: 45 commands on Windows, split between plaintext and encrypted categories, and 29 unencrypted commands on Linux [2][7]. Both variants beacon to hardcoded command-and-control domains over HTTP POST requests to fixed endpoints, and both implement a weighted anti-analysis scoring system that evaluates process names, available RAM, CPU core counts, and other sandbox indicators, self-terminating if the accumulated score reaches a 50-point threshold — a defensive design whose effect is to frustrate both automated malware-analysis pipelines and human reverse engineers. The Windows variant resolves API calls through PEB hash walking rather than direct imports and encrypts strings individually with a per-string xorshift32 algorithm, and its command-and-control configuration can be stored in an NTFS Alternate Data Stream attached to the Windows hosts file, letting operators update infrastructure without recompiling the implant. SPECTRE's broader Windows capability set includes process injection via hollowing, APC EarlyBird, and self-hollowing techniques, token impersonation through named pipes, registry hive dumping for offline credential extraction, Chrome and Edge credential theft, keystroke logging, and in-memory execution of .NET assemblies — a capability profile consistent with a general-purpose, long-term-access backdoor rather than a single-use exploitation tool.

The implant's most consequential capability is its use of bring-your-own-vulnerable-driver (BYOVD) techniques to defeat EDR products on Windows. SPECTRE downloads two well-documented vulnerable drivers from its command-and-control 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 predate this campaign [2]. RTCore64.sys was separately abused by BlackByte ransomware in a 2022 EDR-bypass campaign, and DBUtil_2_3.sys has been weaponized by the Lazarus Group to deploy a kernel-level rootkit against espionage targets — precedents that span both cybercrime and nation-state actors and predate UAT-10147's campaign by several years [9][10]. SPECTRE uses these primitives to locate the Windows kernel image in memory and surgically unlink registered EDR callbacks from the kernel's doubly-linked callback list, a targeted manipulation that Talos describes as leaving the driver's other structures intact to avoid crashing the system. The practical effect is that products including CrowdStrike Falcon, SentinelOne, and Microsoft Defender lose visibility into new process creation, thread creation, and image-load events on the compromised host, allowing the rest of SPECTRE's activity — credential theft, lateral movement, persistence — to proceed largely unmonitored by tooling that depends on those kernel callbacks for telemetry.

Specter: A Kernel-Level Linux Rootkit

SPECTRE's Linux component includes an integrated kernel-mode rootkit that Talos names Specter, deployed under the disguise of acpi_pad.ko — a filename chosen to resemble a legitimate ACPI processor-aggregator kernel module — and persisted through a fraudulent systemd service entry [2]. Rather than patching the system call table directly, Specter uses the Linux kernel's native ftrace instrumentation framework with the FTRACE_OPS_FL_IPMODIFY flag to hook six syscall handlers covering TCP connection enumeration (hooked_tcp6_seq_show, hooked_tcp4_seq_show), signal delivery (hooked_tkill, hooked_tgkill, hooked_kill), and directory-entry enumeration (hooked_getdents64). This approach lets the rootkit hide its own network connections, conceal its kernel module from listing tools, and intercept process-termination signals, all while relying on a documented kernel API rather than the more easily fingerprinted technique of directly modifying kernel data structures. Specter communicates with its userspace controller through signal-based inter-process communication keyed to a magic process identifier of 31337 (0x7A69 in hexadecimal), using specific real-time signals to trigger process hiding, module concealment, privilege escalation, and handshake acknowledgment.

Talos assesses with medium confidence that AI tooling assisted in the rootkit's development, an assessment built on three structural indicators in the recovered source code rather than a direct admission or metadata artifact: structured, specification-style comments at the top of the code that read like a product requirements document rather than developer notes, a rigid and uniform pattern of decorative section separators throughout the file, and the inclusion of three functionally redundant implementation approaches for a single capability — a pattern more consistent with a language model generating multiple candidate solutions than with a human developer settling on one [2]. If accurate, this would mark one of the more concrete pieces of evidence to date that agentic AI is being used not only to operate malware during an intrusion, as in the Python automation scripts described above, but to author the malware itself, including kernel-level components that have historically required specialized systems-programming expertise.

Recommendations

Immediate Actions

Organizations running internet-exposed IIS, Tomcat, or other web server platforms should prioritize patching against the one-day vulnerabilities in Table 1 and against the Linux privilege-escalation flaws UAT-10147 has weaponized, particularly CVE-2022-0847 ("Dirty Pipe"), which remains a reliable escalation path on unpatched systems years after disclosure. Security teams should deploy the ClamAV

and Snort detection content Cisco has published for SPECTRE and its associated tooling, and should hunt specifically for the scheduled-task name "Google Chrome Start," unexpected Windows Defender exclusion-list modifications, and NTFS Alternate Data Streams attached to the hosts file, all of which are indicators drawn directly from this campaign and specific enough to minimize false positives [1][2]. EDR and endpoint-security teams should audit whether their kernel-callback-based telemetry includes tamper-evident integrity checks — a periodic verification that the callback list itself has not been altered — since SPECTRE's BYOVD technique is designed specifically to defeat products that rely on an unmonitored callback chain for process and image-load visibility.

Short-Term Mitigations

Enterprises should extend driver-blocklisting programs, such as Microsoft's vulnerable driver blocklist and equivalent third-party allowlisting controls, to explicitly cover RTCore64.sys and DBUtil_2_3.sys if they are not already enforced, and should treat the appearance of any unrecognized signed kernel driver 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 detection and may not flag ftrace hook installation with the FTRACE_OPS_FL_IPMODIFY flag; kernel integrity-monitoring tools that periodically verify the ftrace hook table against a known-good baseline provide a practical, deployable control against this technique. Security teams should also incorporate AI-generated exploitation artifacts into their threat-modeling assumptions going forward: the discovery of complete, victim-specific exploitation runbooks generated by AI tooling suggests that even organizations without well-known or high-profile vulnerabilities should assume that agentic reconnaissance-and-exploitation tools can and will identify and target them at scale, independent of manual attacker interest.

Strategic Considerations

UAT-10147 is evidence that the operational advantages agentic AI provides to attackers are diffusing beyond the state-sponsored and highly resourced actors that received the earliest public attention, reaching mid-tier, financially motivated cybercrime groups running commodity tooling. Security leaders should recalibrate resourcing assumptions accordingly: the presence of an autonomous exploitation pipeline behind an attack should no longer be assumed to imply a well-funded, exceptionally sophisticated adversary, and defenses calibrated to the presumed skill level of "typical" SEO-fraud or opportunistic web-server compromise groups should be revisited. Organizations should also expect that AI-authored malware components — not just AI-assisted operator workflows — are likely to become more

common and correspondingly harder to distinguish from human-authored code through code-review heuristics alone, based on this and prior AI-assisted malware findings, which argues for investment in behavioral and runtime detection that does not depend on identifying an artifact's authorship.

CSA Resource Alignment

This campaign extends a pattern CSA's AI Safety Initiative has tracked across three prior research notes on independently developed cases of agentic AI operating inside live intrusions. "Autonomous AI Attack Pipelines Move Into the Field" documents a separately identified Chinese-speaking actor who built an architecturally similar pipeline — a commodity reasoning model, an open-source agent framework, and an asset-search integration — to conduct autonomous reconnaissance and exploitation at scale, reinforcing this note's assessment that UAT-10147 is not an isolated case but part of a broader convergence of independent actors on the same operational template [5]. "Marimo RCE: LLM Agents as Post-Exploitation Tools" analyzed the first confirmed in-the-wild use of an LLM agent to conduct post-exploitation lateral movement, and its finding that AI-driven post-compromise activity can be identified through behavioral signatures — machine-optimized command syntax, output-dependent value chaining, and improvised schema discovery without prior reconnaissance — offers defenders a detection framework likely applicable to hunting for UAT-10147's AI-generated Python automation [4]. "LLM-Orchestrated Kill Chains: From CVE to Database Breach in Four Pivots" situates both of these cases against the Anthropic-disclosed GTG-1002 campaign and academic findings on autonomous CVE exploitation, and its recommendation that defenders treat patch velocity as a time-critical control and extend SOC monitoring to agentic tool-call logs is likely applicable to the vulnerability inventory and detection gaps this note identifies [3].

Where these threat-specific artifacts do not reach a particular control area, the AI Controls Matrix (AICM v1.1) provides the underlying governance and technical-control baseline, particularly its domains covering threat and vulnerability management and application and infrastructure security, both of which map to the patching, driver-blocklisting, and kernel-integrity controls recommended above [6].

References

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

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

[3] Cloud Security Alliance. "LLM-Orchestrated Kill Chains: From CVE to Database Breach in Four Pivots." CSA AI Safety Initiative, May 27, 2026.

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

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

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

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

[8] GBHackers. "UAT-10147 Compromises Web Servers to Deploy BadIIS for SEO Fraud and Data Theft." GBHackers, August 2026.

[9] Sophos. "Remove All The Callbacks – BlackByte Ransomware Disables EDR Via RTCore64.sys Abuse." Sophos News, October 4, 2022.

[10] BleepingComputer. "Lazarus hackers abuse Dell driver bug using new FudModule rootkit." BleepingComputer, September 30, 2022.