DRILLAPP: Edge Debugging API Weaponized for Ukrainian Espionage

Laundry Bear Advances Browser-as-Implant Technique Against Defense Targets

Unofficial AI-assisted Research

Cloud Security Alliance AI Safety Initiative

2026-03-18

Key Takeaways

A threat actor tentatively assessed as Russian-linked — tracked as Laundry Bear (also known as UAC-0190 and Void Blizzard) with low-to-medium confidence — has deployed a novel JavaScript-based backdoor, DRILLAPP, that exploits Microsoft Edge's built-in debugging capabilities to conduct covert surveillance against Ukrainian entities. Rather than introducing malicious code into operating system processes or exploiting software vulnerabilities, DRILLAPP launches Edge in headless mode with a collection of security-disabling command-line flags that grant the browser unfettered access to the local file system, microphone, camera, and screen — without triggering user-facing permission prompts. The technique represents a notable advance in living-off-the-land (LOLBAS) tradecraft, turning a widely-deployed enterprise browser into a browser-based surveillance tool with significant collection capabilities while leaving no custom kernel drivers, injected DLLs, or signed-binary tampering for endpoint detection tools to surface.

The campaign was first observed in early February 2026 and proceeded through at least two distinct variants, each refining the delivery mechanism and expanding the backdoor's capabilities [1][2]. Attribution to Laundry Bear carries low-to-medium confidence and rests on overlaps in lure themes, infrastructure patterns, and malware family lineage connecting DRILLAPP to the PLUGGYAPE backdoor that CERT-UA attributed to the same actor in January 2026 [3]. Organizations supporting Ukrainian defense, government, and civil society operations — particularly those relying on Signal, WhatsApp, or charity-related networks as operational channels — face elevated risk and should treat browser debugging interfaces as an active attack surface requiring immediate hardening.

Background

Laundry Bear: A Newly Disclosed Russian Espionage Actor

Laundry Bear emerged publicly in May 2025, when Microsoft Threat Intelligence and Dutch intelligence agencies — the AIVD and MIVD — jointly attributed a persistent espionage campaign to the group, with Microsoft tracking the actor under the name Void Blizzard [4]. The disclosure established that the threat actor had been conducting operations since at least mid-2024, with targeting spanning NATO member

governments, defense contractors, communications providers, healthcare systems, educational institutions, and media organizations across Europe and North America. Dutch officials assessed that virtually all European Union and NATO member countries had been targeted at some point during the campaign period [4].

Laundry Bear's documented early tradecraft relied on credential abuse and cloud API exploitation rather than custom malware [4][5]. The group authenticated directly to Microsoft Exchange Online, SharePoint Online, and Microsoft Teams using stolen authentication tokens and session cookies — the acquisition mechanism for which has not been publicly confirmed — conducting bulk exfiltration of email and files. Password spraying against Entra ID served as a supplementary access vector. This approach, while unsophisticated in its tooling, proved effective against organizations that lacked robust anomaly detection on legitimate cloud authentication events.

The PLUGGYAPE campaign, disclosed by CERT-UA in January 2026, marked a shift toward custom tooling [3][6]. In that operation, attackers used Signal and WhatsApp to reach Ukrainian Armed Forces personnel with messages impersonating charitable foundations. Clicking links in these messages led victims to download password-protected archives containing a Python-based executable packaged with PyInstaller. When run, the executable deployed the PLUGGYAPE backdoor, which profiled the host, registered a unique victim identifier with attacker infrastructure, and awaited remote command execution. An updated variant introduced in December 2025 improved obfuscation and replaced the original C2 channel with MQTT-based communications [3]. If the attribution to the same actor holds, DRILLAPP would represent a further evolution of this tactical progression — substituting browser exploitation for Python packaging and expanding the available surveillance capabilities considerably.

The Mechanics of Browser Debug Mode as an Attack Surface

Microsoft Edge, like all Chromium-based browsers, exposes an internal administrative interface called the Chrome DevTools Protocol (CDP) when launched with the `--remote-debugging-port` flag [7][8]. CDP is the same protocol used by legitimate development tools — including Visual Studio Code, Playwright, and Puppeteer — to inspect running browser sessions, evaluate JavaScript in any page context, intercept network traffic, and read or modify any data accessible within the browser's process. When the debugging port is active, any local process that can connect to it gains the same access as a developer with full DevTools open against a live session.

The risk that exposed CDP interfaces pose to enterprise environments has been documented by the security research community for several years. Work published by SpecterOps demonstrated that Chrome's remote debugger port enables extraction of session cookies, plaintext credentials, and locally stored data — without exploiting any vulnerability, simply by connecting to the port and issuing valid CDP commands [8]. (Chromium introduced targeted protections against certain CDP-based cookie

extraction techniques in July 2023; DRILLAPP's use of CDP for file download operations represents a distinct attack vector not addressed by those mitigations.) The key insight that DRILLAPP operationalizes is that CDP's reach extends beyond session data: when Edge is launched with additional flags that disable sandboxing and bypass security policies, the browser's user-consent permission gates are bypassed, granting the JavaScript running inside the headless instance access to operating system resources that the browser would ordinarily gate behind user permission dialogs.

The command-line flags DRILLAPP employs to achieve this include `--no-sandbox`, which removes the Chromium sandbox that isolates browser processes from the operating system; `--disable-web-security`, which eliminates the same-origin policy; `--allow-file-access-from-files`, which grants the browser direct access to the local file system; `--use-fake-ui-for-media-stream`, which suppresses permission dialogs for camera and microphone access; `--auto-select-screen-capture-source=true`, which enables silent screen capture; and `--disable-user-media-security`, which removes the last remaining media permission checks [1] [2]. None of these flags represent vulnerabilities in the conventional sense — they are documented developer options that Microsoft includes for legitimate testing purposes. That design choice makes them an attractive target for threat actors seeking to operate below the threshold that signature-based detection can reliably catch.

Security Analysis

Campaign Timeline and Delivery Mechanism Evolution

Researchers at Lab52 identified the first DRILLAPP campaign variant in early February 2026, with an earlier sample showing an attempted connection to a third-party domain dating to January [2]. The initial variant used Windows Shortcut (LNK) files as the delivery mechanism. When a victim executed the LNK file — delivered through lure content themed around the Come Back Alive charitable foundation and Starlink installation imagery, consistent with social engineering themes targeting Ukrainian military personnel — the shortcut created an HTML Application (HTA) file in the system's temporary folder [1][2]. The HTA file loaded a remote, obfuscated JavaScript payload hosted on Pastefy, a legitimate paste-sharing service, and copied the LNK file into the Windows Startup folder to establish persistence across reboots.

A second variant emerged in late February 2026 with meaningful refinements. The delivery mechanism shifted from LNK files to Control Panel (.CPL) files, which provide a different execution pathway that may bypass certain LNK-specific detection rules. Lure themes in this variant incorporated state audit service documents and weapons seizure reports, suggesting the actor adapted social engineering content to targets beyond the military charity context [2]. The decoy documents in Variant 2 were displayed via legitimate Ukrainian government websites, lending the lures additional credibility. The backdoor capabilities in Variant 2 also expanded, adding recursive file listing and batch file upload functionality to the surveillance suite present in the original version — evidence of active development and operational iteration during the campaign period.

The DRILLAPP Kill Chain

Once the HTA file executes, it uses Pastefy as a dead drop resolver: rather than hardcoding C2 server addresses in the malware itself — a practice that enables fast takedowns through IP or domain blocking — the HTA retrieves a WebSocket URL from a Pastefy-hosted document [1]. This indirect addressing technique, documented across multiple threat actors as a C2 evasion pattern, allows the operator to update the active C2 endpoint by modifying a paste rather than redeploying malware to already-compromised hosts. The WebSocket URL retrieved from Pastefy serves as the real-time command channel through which the attacker issues instructions and receives surveillance data.

Before establishing the C2 connection, DRILLAPP performs device fingerprinting using canvas fingerprinting — a technique that renders off-screen graphics through the browser and hashes the rendering output to produce a device identifier that is stable across sessions and resistant to cookie clearing [1]. The backdoor also performs timezone detection, logging geographic locale against a list that includes Ukraine, the United Kingdom, Germany, France, Russia, the United States, Canada, Australia, Japan, China, India, Brazil, Italy, Spain, and Poland, and defaulting to a U.S. profile when no match is found [1][2]. The inclusion of Russia in this list is analytically notable: for a campaign attributed to a Russian actor targeting Ukrainian entities, Russia's presence may indicate operator testing from Russian infrastructure, coverage of Russian diaspora communities in Western countries, or intelligence collection against Russian-adjacent targets — though no definitive conclusion is available from current evidence. This profiling information likely supports victim triage, enabling operators to prioritize high-value targets, avoid unintended infections in regions outside the operation's scope, or inform automated payload variant selection.

DRILLAPP's documented C2 commands include audio capture from the system microphone, webcam video recording, screenshot capture of the active display, and file access and exfiltration from anywhere the browser's elevated file system permissions reach [1][2]. In Variant 2, the file access capability was extended to support recursive directory enumeration and batched upload operations, increasing the

throughput available for bulk file collection. The download pathway — bringing files from the attacker's infrastructure onto the target machine — relies specifically on CDP's ability to inject JavaScript that simulates user interaction, circumventing the browser's normal restrictions on programmatic file downloads [2].

Attribution Assessment

The link between DRILLAPP and Laundry Bear carries low-to-medium confidence, grounded in behavioral and thematic overlap rather than definitive technical proof. The charity-themed lures in the initial DRILLAPP variant directly echo the social engineering imagery used in the January 2026 PLUGGYAPE campaign that CERT-UA attributed to UAC-0190 [3][6]. Both campaigns leverage public text-sharing platforms for payload hosting — Pastefy in DRILLAPP's case, comparable services in prior operations — a pattern that reduces infrastructure costs and attribution surface while leveraging services that enterprise firewalls frequently whitelist [2]. Researchers characterize DRILLAPP as representing early-stage campaign activity consistent with continued development of the same threat actor's toolset, and assess that the malware is still maturing based on the functional additions observed between Variant 1 and Variant 2 [2].

The broader context of Laundry Bear's operational tempo reinforces this assessment. The group has demonstrated a consistent pattern of operational adaptation since its initial identification in mid-2024, progressing from pure credential abuse toward increasingly capable and diverse custom tooling within roughly eighteen months [4][5]. DRILLAPP's browser-exploitation approach fits the trajectory of an actor seeking to extend collection capabilities while maintaining the LOLBAS philosophy that has characterized its operations — avoiding the custom kernel implants and signed binary tampering that trigger high-confidence endpoint detections.

Recommendations

Immediate Actions

The most direct mitigation available to enterprise defenders is to disable remote debugging in Microsoft Edge through Group Policy. Microsoft's Edge Browser Policy documentation includes the `RemoteDebuggingAllowed` policy, which can be set to disabled, blocking the `--remote-debugging-port` flag from activating the CDP interface [9]. Deploying this policy across enterprise endpoints removes the specific mechanism DRILLAPP depends on for its file-download capability,

though it does not prevent the broader pattern of Edge being launched with security-disabling flags. Organizations should note that this policy applies specifically to Microsoft Edge; organizations with Chromium-based browsers from other vendors should identify equivalent controls. Compliance with the policy should be verified through an endpoint configuration scan before treating the control as effective.

Endpoint detection should be extended to alert on Microsoft Edge — or any Chromium-based browser — launched from processes other than interactive user sessions, particularly when the process tree includes Mshta (T1218.007) or Windows Startup folder modifications (T1547.001). The specific combination of `--headless`, `--no-sandbox`, and `--disable-web-security` flags in command-line arguments associated with Edge process creation constitutes a high-fidelity indicator of malicious use in environments where headless browser automation is not part of legitimate developer or CI/CD workflows; organizations should baseline browser process launches against expected build and test automation activity before deploying this detection rule broadly. SIEM correlation rules that join LNK or CPL file creation events in temporary folders with subsequent browser process launches can surface the infection chain's early stages before surveillance activity begins.

Network defenders should review proxy and firewall policies governing access to paste-sharing services including Pastefy and similar platforms. While blanket blocking of these services may not be feasible in all environments, WebSocket connections from browser processes to paste-service infrastructure represent an anomalous behavioral pattern that warrants alerting. Traffic inspection for WebSocket connections originating from Edge processes launched without a visible user interface is a practical detection point in environments with SSL inspection capabilities.

Short-Term Mitigations

Organizations supporting Ukrainian government, defense, or civil society operations should treat their employee communication channels — particularly Signal, WhatsApp, and social media platforms — as active initial-access vectors for this threat actor. User awareness training specific to charity-themed and government document-themed lures, combined with explicit guidance that file downloads through messaging applications should be treated as high-risk regardless of apparent sender, addresses the social engineering entry point that DRILLAPP and PLUGGYAPE share. The consistent use of credible charitable organizations as cover makes these lures difficult to dismiss through general phishing skepticism; named-organization awareness — training that specifically identifies the Come Back Alive Foundation and similar entities as recurring lure vehicles — may improve detection rates over general phishing skepticism training alone, given the lures' credibility within the target community.

The Windows Startup folder persistence mechanism (T1547.001) is detectable through file integrity monitoring configured to alert on new executable or shortcut files written to `%APPDATA%\Microsoft\Windows\Start Menu\Programs\Startup` and equivalent locations. This control is broadly applicable, relatively low-noise in most enterprise environments, and addresses a persistence technique that appears across multiple Laundry Bear campaign variants. Implementing application allowlisting that prevents unsigned scripts from executing through Mshta reduces the HTA-based loader's effectiveness without requiring DRILLAPP-specific signatures.

Strategic Considerations

DRILLAPP illustrates a trend that security architects should factor into enterprise browser governance strategies: the debugging and automation interfaces built into modern browsers are increasingly treated by threat actors as an attack surface analogous to PowerShell, WMI, and other Windows administrative tooling. Just as the security community spent a decade building detection and response capabilities around PowerShell-based LOLBAS techniques, browser debugging APIs have been documented as an attack surface for several years and are now seeing operationalization by state-linked threat actors at scale. Organizations should evaluate whether their current endpoint monitoring covers browser process behavior with the same fidelity as shell interpreter behavior, and should include browser debugging API abuse in their red team exercise programs to validate detection coverage.

For organizations in sectors that Laundry Bear has persistently targeted — defense supply chains, government agencies, healthcare, and communications infrastructure — zero trust architecture principles apply directly to this threat. The actor's documented initial access pattern relies on legitimate cloud credential abuse, meaning that identity-centric controls — phishing-resistant MFA, conditional access policies that evaluate device compliance state, continuous session risk evaluation — form the foundational defense layer. DRILLAPP represents the post-access persistence and collection capability; preventing initial credential compromise through robust identity hygiene limits the actor's ability to reach the deployment stage where DRILLAPP becomes relevant.

CSA Resource Alignment

The DRILLAPP campaign engages several dimensions of the Cloud Security Alliance's published guidance and frameworks. The threat's reliance on living-off-the-land techniques — exploiting legitimate browser functionality rather than deploying novel malware — reflects exactly the class of endpoint behavior that the CSA Cloud Controls Matrix (CCM) v4.0's Threat and Vulnerability Management domain (TVM) addresses through requirements for behavioral-based endpoint monitoring, not merely

signature-based detection [10]. Organizations implementing CCM controls should verify that their TVM controls explicitly cover browser process anomaly detection and not solely file-hash or network-signature-based approaches.

The CSA's guidance on AI Organizational Responsibilities is relevant in a less obvious but consequential way: as organizations adopt AI-assisted security operations, the same Chrome DevTools Protocol and headless browser automation that DRILLAPP exploits is used by a growing number of AI agent frameworks that perform browser-based task execution [11]. Security teams should review their internal AI tool deployments to confirm that browser instances spun up by AI agents are not launched with the same security-disabling flags that DRILLAPP weaponizes, and that CDP interfaces exposed during AI agent operation are bound to loopback addresses with appropriate access controls.

This threat scenario maps conceptually to MAESTRO's Layer 3 (Agent Orchestration) and Layer 5 (Tool and API Integration) threat categories, both of which apply when a browser debugging interface is exposed as an accessible tool to a software agent — whether that agent is controlled by a human operator running DRILLAPP or an AI framework with inadequate sandboxing [12]. Organizations building AI agent pipelines that leverage browser automation should treat these MAESTRO layer controls as applicable to their browser integration architecture.

The CSA Zero Trust guidance's principle of least-privilege access is directly implicated by the attack: DRILLAPP's effectiveness depends on Edge being able to launch with privileged flags from a process that gained execution through social engineering. A Zero Trust endpoint posture that enforces application launch constraints through allowlisting, monitors for privilege escalation in process trees, and continuously validates device integrity would surface this behavior before surveillance collection begins [13].

References

[1] The Hacker News, "DRILLAPP Backdoor Targets Ukraine, Abuses Microsoft Edge Debugging for Stealth Espionage," March 2026. https://thehackernews.com/2026/03/drillapp-backdoor-targets-ukraine.html

[2] Lab52, "DRILLAPP: New Backdoor Targeting Ukrainian Entities with Possible Links to Laundry Bear," March 2026. https://lab52.io/blog/drillapp-new-backdoor-targeting-ukrainian-entities-with-possible-links-to-laundry-bear/

[3] SOCPrime, "UAC-0190 Attack Detection: Fake Charity Lures Used to Deploy the PLUGGYAPE Backdoor Against the Ukrainian Armed Forces," January 2026. https://socprime.com/blog/uac-0190-attacks-detection/

[4] CyberScoop, "New Russian State-Sponsored APT Quickly Gains Global Reach, Hitting Expansive Targets," May 2025. https://cyberscoop.com/laundry-bear-void-blizzard-russia-apt/

[5] The Record from Recorded Future News, "Dutch Intelligence Unmasks Previously Unknown Russian Hacking Group 'Laundry Bear'," May 2025. https://therecord.media/laundry-bear-void-blizzard-russia-hackers-netherlands

[6] The Hacker News, "PLUGGYAPE Malware Uses Signal and WhatsApp to Target Ukrainian Defense Forces," January 2026. https://thehackernews.com/2026/01/pluggyape-malware-uses-signal-and.html

[7] Google Chrome Developers, "Chrome DevTools Protocol." https://chromedevtools.github.io/devtools-protocol/

[8] Justin Bui / SpecterOps, "Hands in the Cookie Jar: Dumping Cookies with Chromium's Remote Debugger Port," SpecterOps Blog, December 2020. https://specterops.io/blog/ [Note: Original posts.specterops.io URL has been retired; article available via site:specterops.io "Hands in the Cookie Jar" search.]

[9] Microsoft, "Microsoft Edge Browser Policy Documentation: RemoteDebuggingAllowed," Microsoft Learn. https://learn.microsoft.com/en-us/deployedge/microsoft-edge-browser-policies/remotedebuggingallowed

[10] Cloud Security Alliance, "Cloud Controls Matrix v4.0," CSA. https://cloudsecurityalliance.org/research/cloud-controls-matrix/

[11] Cloud Security Alliance, "AI Organizational Responsibilities Working Group," CSA AI Safety Initiative, 2025. https://cloudsecurityalliance.org/research/working-groups/ai-organizational-responsibilities/ [Note: URL resolves to the general CSA Working Groups index; consult the CSA AI Safety Initiative for specific AI organizational responsibility publications.]

[12] Cloud Security Alliance, "MAESTRO: A Threat Modeling Framework for AI Agent Systems," CSA AI Safety Initiative, 2025. https://cloudsecurityalliance.org/research/working-groups/ai-safety-initiative/ [Note: URL returned 404 at time of publication; consult the CSA AI Safety Initiative working group for the current canonical URL of the MAESTRO framework document.]

[13] Cloud Security Alliance, "Zero Trust Working Group," CSA. https://cloudsecurityalliance.org/research/working-groups/zero-trust/ [Previously cited as the Software Defined Perimeter Working Group, which has been folded into the active Zero Trust Working Group.]