INTELLIGENCE BRIEFING
Security Command Center
TLP:CLEAR
2026-03-29 18:35 UTC

FAUX#ELEVATE: Fake Resume Campaign Weaponizes Legitimate Services to Steal Credentials and Mine Monero in 25 Seconds

THREAT CAMPAIGN | **HIGH** | CVSS 7.5

SCC Item ID	SCC-CAM-2026-0097
Type	Threat Campaign
Severity	HIGH
CVSS Base Score	7.5
Affected Products	Chromium-based browsers (ChromElevator ABE bypass), Mozilla Firefox, Microsoft Defender, Windows UAC/Registry, WinRing0x64.sys kernel driver, Dropbox (staging abuse), mail.ru SMTP (exfiltration abuse), WordPress sites (C2 abuse); targets domain-joined Windows enterprise machines in French-speaking environments
Published	2026-03-24
Discovery Source	Rss

Executive Summary

An active phishing campaign tracked as FAUX#ELEVATE is targeting French-speaking corporate environments by distributing malware disguised as resume documents. Within approximately 25 seconds of opening the file, the malware steals browser-stored credentials and exfiltrates them before silently installing a cryptocurrency miner, all while evading detection through Dropbox staging and compromised WordPress infrastructure. Organizations with French-speaking employees, domain-joined Windows environments, or Chromium-based browser deployments face direct exposure to credential compromise and unauthorized resource consumption.

Technical Analysis

FAUX#ELEVATE delivers heavily obfuscated VBScript payloads (T1059.005, T1027) via spear-phishing attachments (T1566.001) disguised as resumes. Execution is gated by a WMI domain-join check (T1497.001) that suppresses activity on non-enterprise machines, actively evading sandbox detection mechanisms. The attack chain completes credential theft in ~25 seconds via app-bound encryption (ABE) bypass targeting Chromium privilege elevation mechanisms (T1555.003), then deploys XMRig for Monero mining (T1496). Privilege escalation uses WinRing0x64.sys (CVE-2020-14979, CWE-269), a known-vulnerable kernel driver still flagged by Microsoft Defender as VulnerableDriver:WinNT/Winring0, to gain kernel-level access (T1068). UAC

bypass (T1548.002) and Defender disablement (T1562.001) follow. Exfiltration routes through mail.ru SMTP (T1041, T1071.003). C2 communications use compromised WordPress sites (T1102). Payload staging abuses Dropbox (T1105). Post-execution cleanup (T1070, T1070.004) removes artifacts to inhibit forensic recovery. Affected targets: domain-joined Windows machines, Chromium-based browsers, Firefox, Windows Defender, UAC/Registry. No CVE is assigned to the campaign itself; the driver abuse references CVE-2020-14979. Relevant CWEs: CWE-693 (protection mechanism failure), CWE-269 (improper privilege management), CWE-494 (download of code without integrity check), CWE-506 (embedded malicious code), CWE-311 (missing encryption of sensitive data). No vendor patch exists for the campaign; the driver vulnerability has been known since 2020. Source: Securonix threat research, reported via The Hacker News (2026-03-04).

Action Checklist

1. Step 1, Block vulnerable driver: Add WinRing0x64.sys to your WDAC or Driver Block List policy; Microsoft Defender detects it as VulnerableDriver:WinNT/Winring0, confirm the definition is current and the alert is not suppressed.
2. Step 2, Block known-abused infrastructure: Restrict or alert on outbound connections to mail.ru SMTP endpoints and unapproved Dropbox domains at the perimeter; flag C2 traffic patterns to low-reputation WordPress-hosted domains.
3. Step 3, Hunt for VBScript execution: Query EDR and SIEM for VBScript (wscript.exe, cscript.exe) spawned from user-context processes (Outlook, browser downloads) in the last 30 days, especially on domain-joined machines; correlate with WMI query activity (T1497.001) and Chromium privilege elevation process invocations.
4. Step 4, Audit browser credential stores: Identify any domain-joined endpoints where Chromium privilege elevation mechanisms or equivalent ABE bypass tooling may have executed; treat affected browser credential stores as compromised and initiate password resets for exposed accounts.
5. Step 5, Harden email and endpoint controls: Enforce attachment policies that block or sandbox macro-capable and script-capable files from external senders; review VBScript execution policy via Group Policy (Software Restriction Policies or AppLocker) to prevent user-space script execution on enterprise endpoints.

IR / Forensic Enrichment

Triage Priority	IMMEDIATE
Escalation Criteria	If forensic analysis reveals credential exfiltration from more than 10 domain-joined machines, or if Monero mining signatures are detected on any production servers, escalate to Chief Information Security Officer (CISO) and activate external IR firm engagement within 4 hours; if targeting government or critical infrastructure entities, notify CISA and relevant sector ISACs immediately.

Recovery Notes	Post-eradication recovery requires mandatory password resets for all affected accounts with 30-day monitoring for lateral movement via Windows Event Log 4624/4648 correlation. Conduct full browser credential store audit across enterprise (Chrome, Edge, Firefox) and implement browser-level credential protection (Windows Hello for Business, FIDO2). Deploy compensating controls from Step 5 organization-wide within 7 days and validate with log analysis to confirm VBScript execution blocking is effective without breaking legitimate workflows. Schedule 30-day post-incident review to assess control effectiveness and update threat hunting signatures.
Forensic Artifacts	Windows Event Logs: Security (Event ID 4624, 4648, 1102), System (Event ID 7 driver load), Sysmon Operational (Event ID 1 process creation, Event ID 11 file events) | WMI Event Log: Microsoft-Windows-WMI-Activity/Operational (Event ID 11 WmiQueryLanguage queries) | User-space artifacts: C:\Users*\AppData\Local\Temp, C:\Users*\Downloads, %TEMP% for .vbs/.js/.wsf/.hta files with recent modification times | Browser credential stores: Chrome Local State, Login Data, History (C:\Users*\AppData\Local\Google\Chrome\User Data\Default); Firefox key4.db, logins.json (C:\Users*\AppData\Roaming\Mozilla\Firefox\Profiles) | Registry and persistence: HKLM\System\CurrentControlSet\Services (driver load history), HKCU\Software\Microsoft\Windows\CurrentVersion\Explorer\RunMRU, HKCU\Software\Microsoft\Windows\CurrentVersion\Explorer\RecentDocs, HKLM\SOFTWARE\Microsoft\Windows\CurrentVersion\Run | Network and exfiltration: Proxy/firewall logs (outbound SMTP to mail.ru, Dropbox domains, WordPress C2), network packet captures (.pcap) showing C2 beaconing patterns, mail server logs for external SMTP relay abuse | Monero mining indicators: Process and driver memory artifacts (WinRing0x64.sys kernel driver loaded), Task Scheduler history for scheduled miner tasks, performance baseline (CPU/GPU usage anomalies), Prefetch files (*chrome*.pf*, *wscript*.pf*)

Per-Action IR Details

Step 1 — Block vulnerable driver: Add WinRing0x64.sys to your WDAC or Driver Block List policy; Microsoft Defender detects it as VulnerableDriver:WinNT/Winring0 — confirm the definition is current and the alert is not suppressed.

NIST Phase: Preparation

Reference: NIST 800-61r3 §2.1 (preparation phase); §3.2.1 (detection and analysis of kernel-level compromise)

Controls: NIST 800-53r5 SI-7 (Information System Monitoring), NIST 800-53r5 AC-6 (Least Privilege), CIS Controls v8 6.1 (Establish and Maintain Application Allow-lists)

Compensating: For organizations without WDAC: Export WinRing0x64.sys file hash (SHA256) and distribute to all endpoints; use free hash-checking utilities (e.g., fciv.exe built into Windows) in scheduled Task Scheduler jobs to audit for presence; alternatively, use Autoruns (Sysinternals, free) to manually enumerate driver load paths on at-risk machines quarterly. If EDR unavailable, configure Windows Defender Exploit Guard via Group Policy to block unsigned or low-reputation drivers.

Evidence: Capture driver load events from Windows Event Log (System channel, Event ID 7, filter on 'WinRing0'); export HKLM\System\CurrentControlSet\Services registry hive before blocking to document pre-existing loads; preserve all .sys files in C:\Windows\System32\drivers matching the pattern 'winring*' for hash comparison and malware analysis.

Step 2 — Block known-abused infrastructure: Restrict or alert on outbound connections to mail.ru SMTP endpoints and unapproved Dropbox domains at the perimeter; flag C2 traffic patterns to low-reputation WordPress-hosted domains.

NIST Phase: Containment

Reference: NIST 800-61r3 §3.2.2 (containment strategies); §3.2.3 (eradication); §4 (incident handling recovery recommendations on network segmentation)

Controls: NIST 800-53r5 SC-7 (Boundary Protection), NIST 800-53r5 CA-3 (System Interconnections), CIS Controls v8 13.1 (Maintain and Enforce Network-Based URL Filters)

Compensating: Without next-gen firewall: configure Windows Defender Firewall outbound rules (Group Policy Computer Configuration > Windows Defender Firewall with Advanced Security > Outbound Rules) to block mail.ru SMTP (25, 465, 587 to mail.ru IP ranges); document Dropbox shared domain list (*.dropbox.com, *.dropboxusercontent.com) and add to firewall deny list. For WordPress C2: use free DNS filtering (e.g., Quad9, NextDNS free tier) to block low-reputation domains; manually audit browser history on suspected machines for WordPress referrers and block via hosts file. Monitor netstat output via scheduled Task Scheduler script capture.

Evidence: Capture proxy/firewall logs for all outbound SMTP connections (filter source port 25, 465, 587) and Dropbox domains for past 90 days before implementing block; export Windows Firewall logs (C:\Windows\System32\LogFiles\Firewall\pfirewall.log); preserve network packet captures (.pcap) from suspected infected endpoints showing egress traffic patterns; document baseline DNS queries to mail.ru and WordPress domains from uninfected control machines for comparison.

Step 3 — Hunt for VBScript execution: Query EDR and SIEM for VBScript (wscript.exe, cscript.exe) spawned from user-context processes (Outlook, browser downloads) in the last 30 days, especially on domain-joined machines; correlate with WMI query activity (T1497.001) and ChromElevator process invocations.

NIST Phase: Detection Analysis

Reference: NIST 800-61r3 §3.1 (preparation for detection); §3.2.1 (detection and analysis with log analysis and event reconstruction)

Controls: NIST 800-53r5 AU-2 (Audit Events), NIST 800-53r5 SI-4 (Information System Monitoring), CIS Controls v8 8.5 (Implement Application Allowlisting for Scripts), CIS Controls v8 8.8 (Disable Unused Scripts and Interpreters)

Compensating: Without EDR/SIEM: manually export Windows Event Logs (Microsoft-Windows-Sysmon/Operational if Sysmon installed free; otherwise System and Security channels) filtering Event ID 1 (process create) for wscript.exe and cscript.exe parent processes from outlook.exe, explorer.exe, chrome.exe, firefox.exe. Query WMI Event Log (Applications and Services > Microsoft > Windows > WMI-Activity > Operational, Event ID 11) for WmiQueryLanguage queries. Search user Temp folders (C:\Users*\AppData\Local\Temp, C:\Users*\Downloads) for recent .vbs, .js, .wsf files modified in past 30 days; use dir /s /od commands to sort by date. Check registry HKCU\Software\Microsoft\Windows\CurrentVersion\Explorer\RunMRU and RecentDocs for script references.

Evidence: Export full Windows Event Log (Security, System, Sysmon Operational) for all domain-joined machines for past 30 days before hunting (preserve .evtx files); capture process creation tree screenshots or logs showing parent-child relationships (wscript.exe parent = explorer/Outlook); preserve all .vbs/.js/.wsf files found in user Temp/Downloads for static malware analysis; extract Windows.old folder if available (post-reboot forensics); capture WMI repository (C:\Windows\System32\wbem\Repository) as forensic image for offline WMI event analysis.

Step 4 — Audit browser credential stores: Identify any domain-joined endpoints where ChromElevator or equivalent ABE bypass tooling may have executed; treat affected browser credential stores as compromised and initiate password resets for exposed accounts.

NIST Phase: Eradication

Reference: NIST 800-61r3 §3.2.3 (eradication phase); §4 (post-incident activities and lessons learned); NIST 800-53r5 IA-2, IA-4 (authentication and access controls)

Controls: NIST 800-53r5 IA-2 (Authentication), NIST 800-53r5 IA-4 (Identifier Management), NIST 800-53r5 AC-2 (Account Management), CIS Controls v8 6.2 (Restrict Administrator Privileges to Dedicated Administrator Accounts)

Compensating: Without EDR: manually inspect Chrome/Edge credential store on suspected machines — export Chrome Local State and Login Data from C:\Users*\AppData\Local\Google\Chrome\User Data\Default; use free SQLite viewer tools to inspect encrypted credentials (decryption requires direct memory access on domain-joined machines, but file presence confirms exposure). For Firefox, check C:\Users*\AppData\Roaming\Mozilla\Firefox\Profiles*.default-release for key4.db and logins.json; presence of these files indicates credential storage was accessible. Cross-reference with login timestamps in Windows Event Log

(Security channel, Event ID 4624) to identify when credentials may have been extracted. Initiate manual password reset workflow for all accounts that have logged in via affected browsers in past 30 days.

Evidence: Capture Chrome/Edge Local State, Login Data, and History database files before credential reset (preserve encrypted state); export Firefox key4.db, logins.json, and history.sqlite; preserve Windows Event Log 4624 (logon events) and 4648 (explicit credential use) for past 30 days to identify affected accounts; document all Chrome/Edge/Firefox browser processes with timestamps from Task Scheduler history or prefetch files (C:\Windows\Prefetch*chrome*.pf*, *firefox*.pf*); capture memory dumps (if possible) or process handles from affected machines to identify any open credential store access.

Step 5 — Harden email and endpoint controls: Enforce attachment policies that block or sandbox macro-capable and script-capable files from external senders; review VBScript execution policy via Group Policy (Software Restriction Policies or AppLocker) to prevent user-space script execution on enterprise endpoints.

NIST Phase: Recovery

Reference: NIST 800-61r3 §3.2.4 (recovery phase); §4 (post-incident recommendations for hardening); NIST 800-53r5 SI-4, CA-7 (system monitoring and security assessments)

Controls: NIST 800-53r5 SI-3 (Malware Protection), NIST 800-53r5 SI-7 (Information System Monitoring), NIST 800-53r5 AC-6 (Least Privilege), CIS Controls v8 8.2 (Restrict Administrative Privileges to Dedicated Administrator Accounts), CIS Controls v8 13.1 (Maintain and Enforce Network-Based URL Filters)

Compensating: Without enterprise email gateway: use Outlook rules at organizational level to flag or move .vbs, .js, .wsf, .hta, .scr attachments to quarantine folder (Group Policy User Configuration > Administrative Templates > Microsoft Outlook > Outlook 2016 > Security > Attachment Handling). Deploy AppLocker via Group Policy (Computer Configuration > Windows Settings > Security Settings > Application Control Policies > AppLocker) with rules blocking wscript.exe, cscript.exe from non-System32 paths; set to 'Audit' mode first for 2 weeks to baseline, then 'Enforce'. Alternatively, use Windows Defender Application Control (WDAC) policy to audit and block script execution. Enable VBScript disable in Group Policy (Computer Configuration > Policies > Administrative Templates > Windows Components > Windows Script Host > Prevent execution of VB scripts) as fallback. Test hardening on pilot group before full deployment.

Evidence: Baseline current email attachment volume before policy implementation by reviewing mail server logs or SMTP transport rules for past 30 days; document existing VBScript execution allowlists (if any) in AppLocker/SRP; preserve current Group Policy objects (.gpo exports) before hardening changes for rollback capability; capture endpoint process execution logs for 1 week post-hardening to confirm no legitimate business process breakage.

Detection Guidance

Behavioral indicators: (1) wscript.exe or cscript.exe spawned from a mail client or browser download context on a domain-joined machine; (2) WMI query for domain membership (Win32_ComputerSystem DomainRole property) immediately preceding script execution, query EDR process telemetry for this sequence; (3) Chromium privilege elevation process invoked outside of a legitimate browser update workflow; (4) WinRing0x64.sys loaded as a kernel driver, query Windows Event Log (System, Event ID 7045 for new service installs) and Sysmon Event ID 6 (driver load); (5) outbound SMTP connections to mail.ru from non-mail-server endpoints (firewall/proxy logs); (6) XMRig process signatures or high sustained CPU utilization on endpoints combined with outbound connections to Monero mining pools (common pool ports: 3333, 4444, 5555, 7777, 14444); (7) mass file deletion or overwrite activity following script execution (T1070.004), Sysmon Event ID 23 (file delete) in a short burst post-execution. SIEM query starting point: correlate wscript.exe/cscript.exe process creation (Sysmon EID 1) + WMI activity (EID 19/20/21) + outbound SMTP (firewall logs, destination AS belonging to mail.ru) within a 60-second window on the same host. Source attribution: Securonix campaign report (FAUX#ELEVATE); Microsoft Defender Antivirus documentation (T1 source).

Indicators of Compromise

Type	Value	Context	Confidence
DOMAIN	`mail.ru (SMTP endpoints)`	Abused for credential exfiltration via SMTP; outbound connections from non-mail-server endpoints should be alerted or blocked	MEDIUM
DOMAIN	`dropbox.com (staging paths, unspecified)`	Abused for payload staging; specific URLs not publicly disclosed in available sources — flag anomalous Dropbox download activity from script execution context	MEDIUM
HASH	`WinRing0x64.sys (specific hash not disclosed in available sources)`	Vulnerable kernel driver used for privilege escalation; Microsoft Defender signature: VulnerableDriver:WinNT/Winring0 — use Defender detection as the practical indicator	HIGH
DOMAIN	`Compromised WordPress sites (specific domains not publicly disclosed)`	Used for C2 communications; hunt for DNS/HTTP connections to low-reputation WordPress-hosted domains from endpoint processes, particularly post-script execution	LOW

Framework Mappings

MITRE-ATTACK

- **T1102** — Web Service
- **T1543.003** — Windows Service
- **T1027** — Obfuscated Files or Information
- **T1041** — Exfiltration Over C2 Channel
- **T1548.002** — Bypass User Account Control
- **T1033** — System Owner/User Discovery
- **T1555.003** — Credentials from Web Browsers
- **T1059.005** — Visual Basic
- **T1071.003** — Mail Protocols
- **T1053.005** — Scheduled Task
- **T1496** — Resource Hijacking
- **T1105** — Ingress Tool Transfer
- **T1070.004** — File Deletion
- **T1068** — Exploitation for Privilege Escalation

- **T1070** — Indicator Removal
- **T1566.001** — Spearphishing Attachment
- **T1497.001** — System Checks
- **T1562.001** — Disable or Modify Tools
- **T1082** — System Information Discovery

NIST-800-53R5

- **SI-3** — Malicious Code Protection
- **SI-4** — System Monitoring
- **CA-7** — Continuous Monitoring
- **SC-7** — Boundary Protection
- **AC-6** — Least Privilege
- **SI-2** — Flaw Remediation
- **AT-2** — Literacy Training and Awareness
- **SI-8** — Spam Protection
- **SI-7** — Software, Firmware, and Information Integrity
- **CM-3** — Configuration Change Control
- **SR-2** — Supply Chain Risk Management Plan
- **SC-13** — Cryptographic Protection

OWASP-TOP10-2021

- **A01:2021** — Broken Access Control
- **A08:2021** — Software and Data Integrity Failures

CIS-V8

- **5.4**
- **6.8**
- **2.5**
- **2.6**
- **6.3** — Require MFA for Externally-Exposed Applications
- **14.2** — Train Workforce Members to Recognize Social Engineering Attacks
- **15.1** — Establish and Maintain an Inventory of Service Providers

HIPAA-SECURITY

- **164.312(d)** — Person or Entity Authentication
- **164.308(a)(5)(i)** — Security Awareness and Training
- **164.308(a)(6)(ii)** — Response and Reporting
- **164.312(e)(1)** — Transmission Security

SOC2-TSC

- **CC6.1** — Logical access security software, infrastructure, and architectures
- **CC7.4** — Responds to identified security incidents

- **CC9.2** — Manages risks associated with vendors and business partners

ISO-27001-2022

- **A.8.8** — Management of technical vulnerabilities
- **A.5.34** — Privacy and protection of personal information
- **A.5.21** — Managing information security in the ICT supply chain
- **A.8.24** — Use of cryptography

NIST-CSF-2

- **GV.SC-01** — Cybersecurity supply chain risk management program

MITRE ATT&CK Mapping

Technique ID	Technique Name	Tactic
T1102	Web Service	Command-And-Control
T1543.003	Windows Service	Persistence
T1027	Obfuscated Files or Information	Defense-Evasion
T1041	Exfiltration Over C2 Channel	Exfiltration
T1548.002	Bypass User Account Control	Privilege-Escalation
T1033	System Owner/User Discovery	Discovery
T1555.003	Credentials from Web Browsers	Credential-Access
T1059.005	Visual Basic	Execution
T1071.003	Mail Protocols	Command-And-Control
T1053.005	Scheduled Task	Execution
T1496	Resource Hijacking	Impact
T1105	Ingress Tool Transfer	Command-And-Control
T1070.004	File Deletion	Defense-Evasion
T1068	Exploitation for Privilege Escalation	Privilege-Escalation
T1070	Indicator Removal	Defense-Evasion
T1566.001	Spearphishing Attachment	Initial-Access
T1497.001	System Checks	Defense-Evasion
T1562.001	Disable or Modify Tools	Defense-Evasion

Technique ID	Technique Name	Tactic
T1082	System Information Discovery	Discovery

Sources

Source	URL	Tier
Security News	https://thehackernews.com/2026/03/hackers-use-fake-resumes-to-steal...	T3
Microsoft Defender Antivirus alert - VulnerableDriver:WinNT/Winring0	https://support.microsoft.com/en-us/windows/microsoft-defender-anti...	T1
is this winring0x64.sys a virus? : r/computerviruses - Reddit	https://www.reddit.com/r/computerviruses/comments/1imce43/is_this_w...	T3
Add warning about WinRing0x64 or update it? CVE-2020-14979 ...	https://github.com/seerge/g-helper/issues/3424	T3
Operationalizing browser exploits to bypass Windows Defender ...	https://www.ibm.com/think/x-force/operationalizing-browser-exploits...	T3

Generated 2026-03-29 18:35 UTC by TJS Security Command Center