TLP:CLEAR

ADGM THREAT INTELLIGENCE NEWSLETTER

PREPARED BY ADGM CTI

CATEGORY	ACTIONABLE
AUDIENCE	ADGM FSRA ENTITIES
DATE	7/5/2026
OVERALL THREAT SCORE	ELEVATED
TARGET SECTOR	FINANCIAL SERVICES
TARGET REGION	UAE, MENA & GLOBAL
ATTRIBUTION	MULTIPLE
TLP	CLEAR

WEEKLY SUMMARY REPORT – 07 May 2026

9	6	0	0
Campaigns	**Vulnerability**	**Cyber Breach**	**Threat Actors**
Threat Campaigns of Potential Relevance to Finance Sector	Actively Exploited & Critical Vulnerabilities	Major Compromises and Breaches	Threat actor activities in the UAE & Middle East impacting Finance Sector

Summary

This week's cybersecurity newsletter covers a blend of social engineering, covert remote-access techniques, and software supply-chain compromise, alongside critical vulnerability patches across widely used enterprise platforms. Highlights include multi-stage phishing and smishing campaigns, AI-assisted deception lures, stealthy persistence methods, legacy-system exploitation, and supply-chain and ransomware cases where design flaws may complicate detection or recovery. From a financial sector perspective, these developments could undermine the resilience of endpoints, development environments, and exposed services, potentially impacting availability and confidentiality if credential theft or hidden backdoors succeed. Financial institutions should note the newsletter's consistent emphasis on rapid patching, strong control of privileged access, software integrity validation, and monitoring for suspicious file execution or installation behaviour.

ADGM THREAT INTELLIGENCE SUMMARY

Name	Threat Severity Rating	TLP	Attribution	Originating Source
OilRig Leveraged Social Unrest Lures for Advanced Multi-Stage Phishing	HIGH	CLEAR	Campaign	Open Source

Executive Summary

Researchers observed OilRig conducting a phishing campaign tied to recent Iranian social unrest, leveraging macro-enabled Excel documents as the initial access vector. The attack chain deploys C#-based loaders that abuse GitHub and Google Drive to retrieve steganographically concealed configuration data, dynamically execute in-memory payloads, and communicate with attackers via encrypted Telegram C2 channels, resulting in a highly covert, cloud-abuse-centric intrusion workflow.

This activity may impact organizations that rely on email and Office workflows and could affect environments where traffic to widely used cloud platforms blends into normal business patterns. Organizations in the financial sector should be aware that the campaign's modular design and encrypted remote-control channel may enable covert access, staged capability delivery, and data movement if initial phishing succeeds.

Technical Details

- The campaign starts with protest-themed phishing lures delivering macro-enabled Excel documents designed to persuade recipients to enable macros.
- Once enabled, VBA (Visual Basic for Applications) macros extract embedded objects from the spreadsheet and decode them to recover C# source code plus supporting configuration content.
- The recovered C# is compiled locally into a malicious loader, reducing the need to deliver a fully built executable in the initial attachment.
- The chain uses a scheduled task for persistence by launching a legitimate executable that loads attacker-controlled .NET components at runtime.
- The loader retrieves encoded configuration from a cloud code repository location, then decodes it to obtain the next staging pointer.
- The next stage pulls an image from cloud storage and uses LSB steganography to extract encrypted configuration hidden inside the picture.
- The extracted configuration is decrypted using Base64 + XOR logic to reveal parameters for module retrieval and command-and-control.
- Follow-on modules are downloaded and loaded dynamically to provide capabilities such as persistence, upload, download, command execution, and program execution.
- Command-and-control is implemented through a Telegram Bot channel with encrypted communications and online signaling behavior.

Recommendations

- Quarantine or warn on macro enabled spreadsheets from external senders, reduce macro execution to approved business cases only.

- Alert on unusual local compilation activity (e.g., C# compilation) followed by newly created scheduled tasks and unexpected .NET loading.
- Baseline and monitor access to common cloud repositories/storage from user endpoints, especially when followed by new process creation.
- Investigate processes that download images and immediately perform decoding/decryption behavior inconsistent with normal viewing or editing.
- Monitor and restrict Telegram API communications from endpoints that do not require it, and triage correlated file/module loading events.

For a detailed breakdown of tactics, techniques, and procedures (TTPs), refer to the Annexure – OilRig Leveraged Social Unrest Lures for Advanced Multi-Stage Phishing

For the indicators of compromise (IOCs), refer to the attached CSV sheet.

Reference to the Source

back to top

Name	Threat Severity Rating	TLP	Attribution	Originating Source
BlueNoroff Blends AI-Generated Fake Meetings with ClickFix and Fileless PowerShell	**HIGH**	**CLEAR**	**Campaign**	**Open Source**

Executive Summary

Researchers have identified a targeted intrusion attributed with high confidence to BlueNoroff, using spear-phishing and a manipulated '*Calendly*' calendar invite that leads victims to a typo-squatted meeting link and a fake Zoom interface. When the victim clicks, the lure page covertly captures live camera feed for reuse and triggers a ClickFix-style clipboard injection, followed by rapid multi-stage credential and data theft focused on crypto wallet extensions.

This activity may impact organizations engaging with Web3/crypto counterparties and could affect executives or staff involved in investment and asset movement workflows, given the campaign's focus on individuals with access and decision authority. Organizations in the financial sector should be aware that the use of realistic "meeting" lures, cloud-style scheduling patterns, and fast post-click execution can reduce the time available for users and controls to intervene.

Technical Details

- The attack begins with impersonation of a reputable figure in fintech legal and a Calendly invitation delivered via email to establish legitimacy and schedule a meeting.
- The calendar event is covertly modified so the meeting link is replaced with a typo-squatted Zoom URL that closely mimics authentic join link structure.
- When the victim clicks, an attacker hosted fake Zoom meeting page loads and the victim may click repeatedly, believing the client is malfunctioning.

- The fake meeting interface captures the victim's live webcam feed and exfiltrates it, enabling reuse as "known participant" lure content in later attacks.
- In parallel, the lure deploys a ClickFix style clipboard injection technique to push malicious content through user interaction.
- The post click chain progresses quickly from initial click to full compromise in under five minutes using a multi-stage credential extraction pipeline.
- Collection focuses on browser-resident data and cryptocurrency wallet extensions, aligning to theft of digital assets and related access paths.
- Arctic Wolf describes components including a PowerShell-based C2 implant, an AES-encrypted browser injection payload, and Telegram Bot API-based screenshot exfiltration.
- Investigators identified additional targets and a production pipeline where stolen webcam footage is combined with AI-generated images to fabricate new fake meeting media.

Recommendations

- Treat unsolicited scheduling invites and last-minute meeting-link changes as suspicious, verify via an out-of-band channel.
- Disable or harden clipboard/"paste-to-fix" behaviors in browsers where feasible, educate users on ClickFix-style prompts.
- Monitor for fileless PowerShell execution patterns and rapid post-click process chains that indicate staged compromise.
- Increase monitoring around browser extension access and credential extraction behaviors, prioritizing users handling digital asset workflows.
- Alert on unexpected screenshot capture and outbound messaging-platform API usage from endpoints that do not require it.

For a detailed breakdown of tactics, techniques, and procedures (TTPs), refer to the Annexure – BlueNoroff Blends AI-Generated Fake Meetings with ClickFix and Fileless PowerShell

For the indicators of compromise (IOCs), refer to the attached CSV sheet.

Reference to the Source

back to top

Name	Threat Severity Rating	TLP	Attribution	Originating Source
PromptMink Supply-Chain Campaign Uses AI-Assisted Code to Steal Crypto Wallet Secrets	HIGH	CLEAR	Campaign	Open Source

Executive Summary

Researchers have reported malicious code introduced into a crypto trading project after an AI-based coding agent added a tainted dependency, enabling theft of sensitive secrets from the host environment. The campaign described as "PromptMink" uses layered "bait" and "payload" packages and has evolved across multiple implementation styles to improve stealth and persistence while targeting crypto- and trading-related development workflows.

This campaign may impact teams building or integrating Web3/crypto tooling, especially where automated assistants suggest dependencies or accelerate commits without deep review. Organizations in the financial sector should be aware that the described technique could affect software supply chains that touch digital-asset operations by exposing secrets that enable downstream access to wallets and funds.

Technical Details

- A malicious dependency was added to an autonomous trading agent project via a commit the report says was co-authored by Anthropic's Claude Opus large language model (LLM).
- The added package presented itself as routine "validation" functionality while actually siphoning secrets from the host environment.
- The campaign uses a two-layer supply-chain approach: "bait" packages appear legitimate and pull in separate "payload" packages that contain the theft logic.
- This layering lets operators swap out exposed payload components while keeping the higher-credibility bait layer stable over time.
- Earlier payloads used obfuscated JavaScript to collect sensitive files commonly used for secrets, then exfiltrate the collected data.
- After detections, the operators shifted to hiding the payload inside large executable bundles to make inspection and takedown harder.
- Later versions added remote-access persistence by installing an attacker-controlled SSH key on supported operating systems.
- The campaign then pivoted to compiled add-on modules written in Rust to reduce obvious payload traits while expanding capability.
- In the Rust phase, the payload expanded to compress and exfiltrate entire projects, including source code, in addition to secrets.
- The report links the activity to a North Korea–linked actor tracked as "Famous Chollima," citing infrastructure and targeting patterns.

Recommendations

- Review and approve dependency additions (especially those suggested by automated tools) before merging, focusing on “new” validation/utility libraries in crypto projects.
- Alert on unusual dependency traits highlighted by the report’s evolution, such as sudden inclusion of large executables or compiled add-on modules in packages.
- Monitor build and developer environments for unexpected secret-access and bulk file collection behavior consistent with the described file-stealing workflow.
- Add detections for unauthorized SSH key installation events on developer/build systems, as the report describes SSH-based persistence in later phases.
- Treat suspected exposure of wallet-related secrets as an incident, since the report states leaked credentials can enable access to wallets and funds.

For the indicators of compromise (IOCs), refer to the attached CSV sheet.

Reference to the Source

back to top

Name	Threat Severity Rating	TLP	Attribution	Originating Source
Long-Lived Supply Chain Backdoor Identified in Widely Deployed WordPress Plugin	HIGH	CLEAR	Campaign	Open Source

Executive Summary

Researchers have identified that multiple sites were running a tampered build of a widely deployed WordPress redirect plugin that reported a legitimate version string but did not match the official repository package. The modified plugin added a hidden content-injection hook for logged-out visitors and also configured an external update channel that could push new plugin code through routine update check.

This campaign may impact organizations operating public-facing WordPress properties, and it could affect environments where plugin integrity is assumed based on version strings alone. Organizations in the financial sector should be aware that a long-lived, supply-chain style backdoor like this could enable silent page-content manipulation and remote code delivery via trusted update mechanisms if left undiscovered.

Technical Details

- The investigation began when a “known issue” alert led to a fleet-wide check that found multiple sites reporting the same plugin version via management tooling.
- Although the version number matched, file comparisons confirmed the installed software was not an official build and appears to have been altered.
- The tampered variant added a function hooked into WordPress content rendering, designed to prepend remotely fetched content into posts/pages.
- The injection was gated to only trigger for logged-out views, meaning administrators reviewing the site while authenticated would not see the injected content.

- The remote request included page and client context (site, requested path, user-agent), enabling selective responses depending on the visitor type.
- The article notes the remote endpoint later stopped resolving, leaving the backdoor "dormant" but still present and ready to reactivate if the endpoint returned.
- A second mechanism embedded an update-checker library and registered a third-party update source, allowing new plugin code to be installed during scheduled update checks.
- Repository history showed the updater capability was introduced through official commits/tags, then later removed from new installs while remaining active on already-deployed versions.
- Archived update metadata described a tampered build version and aligns with the observed modification timestamps across affected sites, supporting the long-lived supply-chain narrative.
- The author highlights integrity verification as the practical detection gap: version numbers appeared normal, while file-level checks would have surfaced drift immediately.

Recommendations

- Run routine plugin integrity verification across WordPress fleets (the article highlights checksum-based verification as the reliable way to detect tampered files).
- Audit plugins for any behavior that pulls content or code from third-party endpoints during page rendering or scheduled update checks.
- Replace the affected redirect plugin with alternatives explicitly suggested in the article to remove reliance on the identified supply-chain path.
- Investigate unexplained public-facing content changes that do not reproduce for authenticated admins, as the backdoor was designed to hide from logged-in review.
- Treat unexpected plugin "update source" changes as high priority, since the article describes this as an avenue for on-demand remote code delivery.

Reference to the Source

back to top

Name	Threat Severity Rating	TLP	Attribution	Originating Source
Phoenix PhaaS Kit Powering Global Smishing-Driven Phishing Campaigns	**HIGH**	**CLEAR**	**Campaign**	**Open Source**

Executive Summary

Researchers have identified the "Phoenix System," a phishing-as-a-service platform driving global smishing campaigns that impersonate banks, telecom providers, and logistics firms to harvest sensitive customer data. The platform combines SMS-based delivery, phishing domains, and real-time victim interaction to improve credential theft and bypass authentication controls.

This activity may impact financial institutions by increasing exposure to brand impersonation and credential theft campaigns targeting customers through trusted communication channels. Organizations in the financial sector should be aware the research describes targeting across financial services alongside

telecommunications and logistics, including features designed to observe sessions in real time and intervene during OTP entry to bypass MFA.

Technical Details

- The Phoenix System operates as a centralized phishing-as-a-service platform, allowing multiple affiliates to run campaigns through a single administrative panel with minimal technical effort.
- Threat actors distribute phishing links via SMS messages that impersonate trusted brands, encouraging victims to click and interact with fraudulent websites.
- Some campaigns leverage rogue Base Transceiver Stations (BTS) to inject SMS messages directly into mobile devices, bypassing telecom filtering mechanisms.
- Two primary phishing themes are used: reward points redemption and failed parcel delivery, both designed to create urgency and drive user action.
- Once users click the link, they are redirected to phishing pages that closely mimic legitimate services to capture login credentials and payment details.
- The platform applies geofencing and IP filtering to ensure only targeted victims can access the phishing pages, reducing exposure to security analysis.
- A centralized dashboard allows attackers to monitor victim activity in real time, including submitted data and device information.
- The system enables "live phishing," where attackers can interact during the session and prompt victims to re-enter OTPs or PINs, helping bypass multi-factor authentication.
- The infrastructure includes thousands of phishing domains, enabling large-scale campaigns and domain rotation to evade detection.
- The phishing kit is distributed via Telegram channels, supporting widespread adoption and coordinated campaign deployment.

Recommendations

- Monitor for SMS-linked brand abuse and phishing pages impersonating your organization and align rapid takedown workflows with telecom coordination where applicable.
- Track newly registered phishing infrastructure patterns described in the research (e.g., large-scale domain churn and infrastructure reuse across themes).
- Continuously monitor and block suspicious or newly registered domains linked to phishing campaigns.
- Enhance customer/staff awareness for urgency-based SMS lures; advise verification via official apps/sites and discourage entering payment data from SMS links.
- Deploy real-time detection for abnormal login behavior and potential misuse of compromised credentials.

For a detailed breakdown of tactics, techniques, and procedures (TTPs), refer to the Annexure – Phoenix PhaaS Kit Powering Global Smishing-Driven Phishing Campaigns

For the indicators of compromise (IOCs), refer to the attached CSV sheet.

Reference to the Source

back to top

Name	Threat Severity Rating	TLP	Attribution	Originating Source
TeamPCP-Linked Supply Chain Compromise in SAP CAP npm Packages	MEDIUM	CLEAR	Campaign	Open Source

Executive Summary

Researchers at Socket reported a suspected supply-chain attack affecting multiple npm packages tied to SAP's JavaScript and cloud application development ecosystem, where compromised releases introduced unexpected installation-time behavior. The modified packages added a preinstall bootstrapper that downloads and runs a platform-specific runtime, then executes a heavily obfuscated payload designed to harvest secrets and enable follow-on compromise.

This campaign may impact developer workstations and CI/CD environments that install SAP CAP (Cloud Application Programming Model) dependencies, because code execution occurs automatically during package installation. Organizations in the financial sector should be aware this could affect software supply chains supporting digital channels or internal platforms if build secrets, cloud credentials, or developer tokens are exposed through compromised dependencies.

Technical Details

- The compromised releases introduced a new preinstall hook that runs automatically during npm install, giving execution before application code runs.
- The preinstall logic acts as a bootstrapper that downloads a platform-specific runtime archive, extracts it, and immediately executes it.
- Socket noted the implementation follows redirects without validating the final destination and uses a Windows execution approach that reduces guardrails.
- The bootstrapper then runs a large, single-line, obfuscated JavaScript payload that hides key strings and intent behind layered decoding logic.
- The payload includes a locale-based guardrail (halting on certain language settings) and then chooses different paths for CI versus developer hosts.
- On developer machines, the payload searches broadly for secrets in common credential locations, environment files, tooling configs, and wallet-related artifacts.
- It also probes cloud credential sources (including metadata-style endpoints) and attempts to pull tokens from developer tooling where available.
- On CI runners, Socket described a memory-focused approach that extracts masked "secret" values from the runner process rather than relying on logs.
- Stolen data is encrypted before exfiltration, using an approach that abuses trusted developer platforms as the transport channel.
- The campaign supports self-propagation by reusing stolen publishing access to inject the payload into additional packages tied to the same ecosystem.

Recommendations

- Review dependency trees and lock files to identify whether the affected SAP CAP related package versions were installed during the exposure window Socket describes.
- Avoid installing the affected versions until more details are confirmed and prefer known-good versions from before the suspicious releases.
- Rotate credentials and tokens that may have been present in developer or build environments, as the payload targets secrets and access material.
- Review CI/CD logs and build telemetry for unexpected network retrievals and binary execution during npm install events.
- Add detection for package-install steps that unexpectedly download runtimes and execute them, especially when it appears newly introduced in a dependency.

For the indicators of compromise (IOCs), refer to the attached CSV sheet.

Reference to the Source

back to top

Name	Threat Severity Rating	TLP	Attribution	Originating Source
SHADOW-EARTH-053 Exploits Legacy Exchange to Deploy Web Shells and ShadowPad Implants	**MEDIUM**	**CLEAR**	**Campaign**	**Open Source**

Executive Summary

Researchers at Trend Micro have identified a China-aligned cyberespionage campaign (tracked as SHADOW-EARTH-053) targeting government and critical infrastructure organizations, primarily across South, East, and Southeast Asia, with at least one NATO-member government also targeted. The campaign gains entry by exploiting vulnerabilities such as '*ProxyLogon chain*' in Microsoft Exchange, then establishes persistence with web shells and stages ShadowPad implants using DLL sideloading of legitimate signed executable.

This campaign may impact organizations that still operate legacy or unpatched Exchange/IIS infrastructure, as the report highlights continued exploitation of long-patched issues for mailbox compromise, credential theft, and prolonged access. Organizations in the financial sector should be aware this could affect environments where email and identity systems are exposed to the internet, because the described tradecraft enables durable footholds and covert data collection once server access is achieved.

Technical Details

- The intrusion begins with exploitation of N-day flaws in public-facing Microsoft Exchange and IIS services to obtain an initial foothold on servers.
- After access, attackers deploy web shells to maintain persistent command execution on the compromised server without needing repeated exploitation.
- The group then introduces ShadowPad as the primary modular implant, using a consistent loader pattern built around DLL sideloading.

- Delivery typically combines a legitimate signed executable, a malicious DLL, and an encrypted payload that is stored in the registry and removed after first use.
- Trend Micro observed scheduled-task persistence that repeatedly launches the sideloaded component at short intervals with elevated privileges.
- Post-compromise discovery includes Active Directory and Exchange reconnaissance executed via web shell context, including enumeration and exports of directory objects.
- The actor uses multiple tunneling/proxy tools in the same environment to preserve outbound connectivity and provide redundancy if one channel is blocked.
- Credential access includes tools and techniques for extracting credentials from memory and directory replication-style methods for elevated access.
- Collection includes mailbox-focused activity, with evidence of exporting and archiving email content from high-profile users for exfiltration.

Recommendations

- Prioritize patching and exposure reduction for internet-facing Microsoft Exchange and IIS servers, as the campaign relies on exploitation of known, older issues.
- Hunt for web-shell-driven server activity by reviewing IIS worker process execution chains and unusual administrative commands run in server contexts.
- Add detections for DLL sideloading patterns involving legitimate signed executables spawning abnormal DLL loads and registry-staged payload behavior.
- Monitor for the appearance of multiple tunneling/proxy utilities on the same host and unexpected long-lived outbound sessions used for covert access.
- Review Exchange logs for suspicious mailbox access or exports, and investigate any password-protected archive files created from email data.

For the indicators of compromise (IOCs), refer to the attached CSV sheet.

Reference to the Source

back to top

Name	Threat Severity Rating	TLP	Attribution	Originating Source
Sandworm Establishes Covert Persistent Backdoors via SSH-over-TOR Tunnels	MEDIUM	CLEAR	Campaign	Open Source

Executive Summary

Researchers have identified suspected Sandworm-linked activity using spear-phishing to deliver a compressed attachment that tricks users into launching a shortcut-based loader, which unpacks and runs a hidden PowerShell-driven installation chain. The chain establishes persistence and covert access by setting up a Tor hidden service and placing an SSH service inside that tunnel, creating a double-encrypted pathway that supports remote interaction and internal service reachability without direct inbound exposure.

This campaign may impact organizations that rely on email workflows and have endpoints capable of running script-based installers, because the flow blends user deception with automated persistence. Organizations in the financial sector should be aware this approach could affect incident visibility and containment, as it is designed to hide management access behind anonymized tunneling and evade routine traffic inspection while enabling data theft operations.

Technical Details

- Initial access starts with spear-phishing that delivers a ZIP containing a shortcut file disguised as a benign document and a hidden, system-styled folder to mislead the user.
- When executed, the shortcut recursively locates the lure archive in the user profile, performs multi-stage extraction, and triggers a hidden PowerShell process to run the main controller script.
- The controller script performs sandbox/VM checks using simple environment heuristics (recent shortcut count and running process count) and exits if thresholds are not met.
- To distract the victim, the script opens a decoy PDF, then deletes dropped artifacts and its own components to reduce disk-forensics visibility.
- Persistence is achieved by registering two scheduled tasks that run at user logon, masquerading as legitimate “repair” activity for common applications.
- One task launches Tor configured as a hidden service, mapping internal services (including remote administration and file-sharing ports) to an onion-accessible endpoint.
- The SSH component is configured to listen only on localhost on a high port, uses public-key authentication (no passwords), and is reachable only through the Tor tunnel.
- Traffic concealment is enhanced through Tor bridge usage and obfs4-style obfuscation, aiming to make tunnel traffic look like random TCP data and bypass deep inspection.
- The script waits for Tor hidden-service materialization, then sends a beacon through the local Tor proxy using repeated retries to ensure operator awareness of the host.

Recommendations

- Restrict execution of shortcut-driven payload chains and enforce tighter controls on script execution behaviors that launch hidden PowerShell processes.
- Monitor for new scheduled tasks created around user logon that reference unexpected binaries or “repair/update” style disguises.
- Hunt for endpoint-side Tor/hidden-service behaviors and local-only SSH listeners that appear shortly after archive extraction activity.
- Alert on signs of traffic obfuscation consistent with Tor bridge usage and obfuscated tunneling, especially when paired with persistence artifacts.
- Reinforce user awareness for spear-phishing lures delivered as compressed attachments and document-themed shortcuts, as described in the infection flow.

For the indicators of compromise (IOCs), refer to the attached CSV sheet.

Reference to the Source

back to top

Name	Threat Severity Rating	TLP	Attribution	Originating Source
VECT Ransomware Functions as Unrecoverable Data Wiper Due to Encryption Design Flaw	MEDIUM	CLEAR	Campaign	Open Source

Executive Summary

Researchers have identified the VECT ransomware campaign, a ransomware-as-a-service operation that distributes malware through a large affiliate network and supply-chain partnerships to target organizations across multiple platforms. The malware is designed to encrypt files and demand ransom, but due to critical flaws, it instead permanently destroys large files, making recovery impossible.

This activity may impact financial institutions that rely on critical data such as databases, backups, and virtual environments, which are primary targets during ransomware incidents. Organizations in the financial sector should be aware that paying the ransom could not restore affected data, increasing the importance of resilience and recovery planning.

Technical Details

- VECT emerged in late 2025 and pursued a “wide open” affiliate approach rather than a small, vetted partner set.
- Access to the ransomware platform was distributed broadly via a forum partnership that granted many members affiliate-level reach.
- The operators also announced a partnership intended to leverage prior software-access footholds as a launchpad for ransomware activity.
- Researchers reviewed the affiliate panel, builder, and three platform payloads and found the core mechanism is fundamentally broken.
- When processing large files, VECT permanently discards required recovery material, making reversal impossible in practice.
- The flaw is described as consistent across Windows, Linux, and ESXi variants and present across all observed versions.
- Several advertised features do not function as claimed, including “speed modes” that are accepted but ignored during execution.
- Anti-analysis/evasion components are described as compiled into the malware but not actually activated during runs.
- The report notes indicators of immature development (unchanged flaws across versions) and says AI-assisted coding cannot be ruled out.
- Despite the flaw, the report cautions that data may still be exfiltrated before disruption and future fixes could increase harm.

Recommendations

- Treat incidents involving this family as potential destructive events: prioritize restoration from clean backups and rapid containment over payment-based recovery.

- Validate backup integrity and recovery time objectives for large-file workloads (VMs, databases, archives), as these are described as most affected.
- Review environments for signs of pre-encryption data collection/exfiltration during incident response, as disruption may not be the only outcome.
- Rotate sensitive credentials and tokens used in developer/build environments if there is any exposure to supply-chain compromise pathways referenced in the report.
- Ensure endpoint and gateway protections are updated to block known variants across Windows, Linux, and ESXi, as the report notes coverage exists for observed samples.

For the indicators of compromise (IOCs), refer to the attached CSV sheet.

Reference to the Source

back to top

Name	Threat Severity Rating	TLP	Attribution	Originating Source
Notepad++ Patches Format String Flaw in Localization Parsing	HIGH	CLEAR	Vulnerability	CSC

Executive Summary

Notepad++ has disclosed and addressed a critical vulnerability (CVE-2026-3008) involving format string injection during parsing of the nativeLang.xml localization configuration file. The issue can be triggered when users run “Find ALL in Current Document” or related search operations with a maliciously crafted localization file.

This vulnerability may impact endpoint stability and could affect application confidentiality if sensitive memory content is exposed during exploitation. Organizations in the financial sector should be aware that user-driven workflows (opening or using a modified localization file and running common search actions) could create a practical pathway for disruption or information exposure on impacted workstations.

Technical Details

- CVE-2026-3008 is rated Critical with CVSS v4: 10, indicating maximum severity in the provided scoring.
- The flaw is a format string injection tied to Notepad++ handling of the find-result-hits parameter inside nativeLang.xml.
- A specially crafted format-string payload embedded in the localization XML is processed without sufficient validation.
- The vulnerability is triggered during search operations such as “Find ALL in Current Document,” where the affected parameter is evaluated.
- Exploitation can cause application instability, including crashes that may produce a denial-of-service condition for the user session.

- The same weakness may disclose sensitive memory contents, potentially exposing internal application data.
- Memory disclosure could assist follow-on exploitation attempts or weaken system security protections by revealing otherwise hidden information.

Recommendations

- Upgrade Notepad++ from 8.9.3 to 8.9.4 or later to remediate CVE-2026-3008.

Vulnerability and affected product details can be found *here*.

back to top

Name	Threat Severity Rating	TLP	Attribution	Originating Source
cPanel Patches Critical Authentication Weakness Affecting cPanel & WHM Access	HIGH	CLEAR	Vulnerability	CSC

Executive Summary

cPanel has addressed a critical authentication vulnerability in cPanel & WHM that may allow unauthorized access to hosting control panels by impacting core authentication mechanisms used in both cPanel and WHM interfaces.

This issue may impact organizations operating affected cPanel deployments and could affect account integrity if unauthorized panel access enables account compromise or administrative access. Organizations in the financial sector should be aware that hosting and control-plane exposure can amplify downstream risk when administrative interfaces are involved, even though full technical details are not yet public.

Technical Details

- The vulnerability affects cPanel & WHM and is characterized as an authentication bypass / unauthorized access issue. It impacts core authentication mechanisms used in both the cPanel and WHM interfaces.
- Severity is described as Critical, though an exact CVSS score is not disclosed in the provided information. The risk level is driven by the potential for unauthorized panel access and account compromise.
- Full technical details have not been publicly disclosed at this time. Despite limited public detail, it is treated as high risk due to the nature of authentication flaws.
- The issue impacts multiple supported versions (specific affected versions are not enumerated in the provided content). Patches are available across several supported release trains.
- Exploit status is not publicly confirmed, but the vulnerability is still treated as high risk. The concern centers on the possibility of unauthorized access leading to administrative control.
- Some hosting providers (example noted: Namecheap) implemented temporary mitigations while deploying official patches.

Recommendations

- Upgrading to one of these versions (or later) is presented as the primary remediation.

- Patched versions listed are: 11.110.0.97, 11.118.0.63, 11.126.0.54, 11.132.0.29, 11.134.0.20, and 11.136.0.5.

Vulnerability and affected product details can be found *here*.

back to top

Name	Threat Severity Rating	TLP	Attribution	Originating Source
Apache MINA Patches Two Critical Unsafe Deserialization RCE Flaws	HIGH	CLEAR	Vulnerability	CSC

Executive Summary

Apache has disclosed two critical vulnerabilities in Apache MINA (Multipurpose Infrastructure for Network Applications) 'CVE-2026-41635' and 'CVE-2026-41409' that enable remote code execution through unsafe deserialization behavior. Both issues stem from allowlist enforcement weaknesses during class resolution and loading, allowing crafted serialized payloads to execute attacker-controlled code paths.

These vulnerabilities may impact exposed applications that use Apache MINA for network communications and accept serialized objects from remote sources, especially where unauthenticated access is possible. Organizations in the financial sector should be aware this could affect service availability and system integrity if vulnerable MINA-based components are reachable and process untrusted serialized input.

Technical Details

- CVE-2026-41635 is caused by inconsistent enforcement of allowlist validation during deserialization in the resolveClass() function.
- Certain execution paths, such as handling static classes or primitive types, bypass the intended classname validation checks.
- This flaw allows attackers to deliver specially crafted serialized objects that evade filtering mechanisms.
- Successful exploitation can result in execution of arbitrary classes, leading to remote code execution.
- CVE-2026-41409 stems from improper initialization order, where allowlist validation occurs after class loading.
- Java automatically executes static initializers during class loading, meaning malicious code can run before validation occurs.
- This enables attackers to embed payloads in static blocks that execute prior to any security checks.
- The issue represents an incomplete fix of a previously identified vulnerability, leaving systems exposed to similar attack techniques.
- Both vulnerabilities are remotely exploitable and may not require authentication depending on application exposure.
- Affected versions include 2.0.x (up to 2.0.27), 2.1.x (up to 2.1.10), and 2.2.x (up to 2.2.5), with fixes released in newer versions.

Recommendations

- Upgrade Apache MINA to 2.2.6, 2.1.11, or 2.0.28 (or later) based on your branch.

Vulnerability and affected product details can be found *here* and *here*.

back to top

Name	Threat Severity Rating	TLP	Attribution	Originating Source
Microsoft Confirms Active Exploitation of Windows Shell Spoofing Flaw	HIGH	CLEAR	Vulnerability	CSC

Executive Summary

Microsoft has confirmed active exploitation of a Windows Shell vulnerability tracked as CVE-2026-32202, a protection mechanism failure that enables network-delivered spoofing when a user executes a crafted malicious file. The issue was originally released on April 14, 2026, and updated on April 27, 2026, with an official patch available.

This vulnerability may impact user trust in file presentation and could affect security workflows because exploitation is confirmed in the wild despite a relatively low CVSS score (4.3). Organizations in the financial sector should be aware that user interaction remains the key trigger, and successful exploitation may expose limited sensitive information even if system integrity and availability are not directly impacted.

Technical Details

- CVE-2026-32202 affects the Windows Shell and is classified as a Protection Mechanism Failure (CWE-693) enabling spoofing.
- The attack vector is network-based with low complexity and no privileges required, but it requires user interaction to execute a malicious file.
- The reported impact is limited confidentiality exposure (C:L) with no integrity or availability impact (I:N/A:N) and unchanged scope.
- Exploitation involves delivering a specially crafted file to the victim, then relying on the victim to run it to trigger spoofed content presentation.
- Microsoft has confirmed the vulnerability is actively exploited in the wild, elevating operational risk beyond the CVSS score.
- The exploit is described as having functional maturity, indicating practical use rather than purely theoretical conditions.
- The issue is reportedly linked to an incomplete patch for CVE-2026-21510, enabling bypass conditions in some scenarios.
- Remediation level is listed as “official patch available,” and the guidance references April 2026 security updates.

Recommendations

- Deploy April 2026 Patch Tuesday updates from Microsoft without delay.
- Ensure EDR/XDR solutions are updated to detect spoofed file execution patterns.

Vulnerability and affected product details can be found *here*.

back to top

Name	Threat Severity Rating	TLP	Attribution	Originating Source
Google Chrome 147 Security Updates Fix Critical Use-After-Free Flaws Across Major Components	HIGH	CLEAR	Vulnerability	CSC

Executive Summary

Google has released Chrome 147 Stable and Extended Stable security updates for Windows, Mac, Linux, and Android, addressing 30 vulnerabilities including multiple critical and high-severity issues across core browser components. Successful exploitation could enable remote code execution, memory corruption, or browser crashes through malicious web content.

These updates may impact organizations with broad browser exposure and could affect endpoint security posture if users browse to attacker-controlled content while running vulnerable builds. Organizations in the financial sector should be aware that widely used browsers are common entry points, and unpatched clients may increase exposure to exploitation paths tied to memory-safety flaws.

Technical Details

- The release includes Chrome 147 Stable and Extended Stable updates for Windows, Mac, Linux, and Android, addressing 30 vulnerabilities. This bundle includes critical and high-severity issues impacting major browser subsystems.
- Critical issues listed include multiple Use-After-Free flaws in components such as Canvas, Accessibility, and Views. Use-after-free conditions can enable memory corruption and, in some cases, code execution.
- High-severity issues include additional Use-After-Free vulnerabilities in GPU, Media, WebRTC, WebView, Cast, and Navigation. These components are commonly reached during normal browsing, media playback, or real-time communications.
- Additional high-severity flaws include Type Confusion in V8, Heap Buffer Overflow in Skia, and Out-of-Bounds Read/Write in ANGLE. These bug classes can contribute to code execution, crashes, or security boundary weakening.
- One high-severity issue is Insufficient Validation of Untrusted Input in Compositing, and another in Feedback. Input validation failures can allow malformed content to drive unsafe states within rendering workflows.

- Medium-severity issues include Heap Buffer Overflow in WebRTC, Integer Overflow in ANGLE, and an additional Use-After-Free in Media. While lower severity, these still contribute to instability and potential exploitation chains.

Recommendations

- Update Chrome across all endpoints to the fixed versions listed (or later), prioritizing user-facing systems that regularly access external web content.
- Fixed versions provided are 147.0.7727.137/138+ (Windows/Mac Stable), 147.0.7727.137+ (Linux Stable), and 147.0.7727.137+ (Android).
- Extended Stable for Windows/Mac is 146.0.7680.216+.

Vulnerability and affected product details can be found *here* and *here*.

back to top

Name	Threat Severity Rating	TLP	Attribution	Originating Source
Remote Code Execution Vulnerability in GitHub	HIGH	CLEAR	Vulnerability	CSC

Executive Summary

GitHub has disclosed a command injection vulnerability (CVE-2026-3854) affecting GitHub[.]com, GitHub Enterprise Cloud (all variants), and GitHub Enterprise Server (GHES), where an authenticated attacker with repository push access could achieve remote code execution using a single git push. The flaw stems from improper sanitization of user-supplied git push option values that were embedded into internal service headers, enabling injection when internal parsing interprets delimiter characters.

This vulnerability may impact organizations running affected GitHub platforms and could affect administrative control of code-hosting infrastructure if an attacker obtains push-level access to a repository. Organizations in the financial sector should be aware that the described weakness can enable RCE without user interaction, which may elevate risk in environments where repository access is broadly provisioned or tightly integrated with build and deployment processes.

Technical Details

- CVE-2026-3854 is a command injection / improper input neutralization issue that can lead to remote code execution. Exploitation requires authentication and repository push permissions and can be performed with a single push action.
- The vulnerable behavior involves user-controlled git push option values being embedded into internal X-Stat headers. Those values were not properly sanitized before being placed into headers used by internal services.
- Internal header parsing relied on semicolon (;) delimiters to separate fields.
- By including delimiter characters in crafted input, an attacker could inject malicious metadata fields during parsing.

- Affected platforms include GitHub.com, GitHub Enterprise Cloud (all variants), and GitHub Enterprise Server (GHES).
- The disclosure notes potential for full compromise of affected instances and cross-tenant exposure in multi-tenant environments.
- Severity is listed as CVSS 8.7 (High) with Network attack vector and no user interaction required.
- Privileges required are "Low" in the sense that only push access is needed to attempt exploitation.

Recommendations

- Upgrade GHES immediately to a patched version: 3.14.25+, 3.15.20+, 3.16.16+, 3.17.13+, 3.18.7+, 3.19.4+, or 3.20.0+.
- Verify patch levels across all GitHub instances (including test and DR environments) to ensure no lagging nodes remain.
- Treat exposed environments as urgent and apply emergency patching where GHES is reachable and actively used.

Vulnerability and affected product details can be found *here*.

back to top

Appendix A - Tactics, Techniques & Procedures (TTPs)

OilRig Leveraged Social Unrest Lures for Advanced Multi-Stage Phishing

Tactics	Techniques	Description
Initial Access	Phishing via document attachment	Social-event-themed Excel files used to lure victims into enabling macros
Execution	User execution; Command and scripting interpreter	Malicious macros trigger execution and compilation of embedded C# code
Persistence	Scheduled task	Persistence achieved by creating scheduled tasks linked to trusted executables
Defense Evasion	Obfuscated and encrypted payloads	Base64 and XOR encryption combined with steganography to hide configuration data
Defense Evasion	Fileless execution	Payloads and modules loaded directly into memory after dynamic compilation
Command and Control	Encrypted channel	Remote control established via encrypted communication through messaging service APIs
Exfiltration	Data transfer over C2 channel	Stolen data transmitted through the established encrypted control channel

BlueNoroff Blends AI-Generated Fake Meetings with ClickFix and Fileless PowerShell

Tactic	Technique	Description
Initial Access	Phishing via Service	Manipulated meeting and scheduling invitations used to lure victims into fake video calls
Execution	Command and Scripting Interpreter	Fileless PowerShell execution triggered through user interaction
Defense Evasion	Fileless Malware	Malicious activity conducted entirely in memory without dropping files
Credential Access	Credentials from Browsers	Extraction of browser-stored credentials and cryptocurrency wallet data
Collection	Screen Capture	Covert capture of live webcam footage from victims
Exfiltration	Exfiltration Over C2 Channel	Stolen credentials, media, and screenshots sent to attacker-controlled infrastructure
Command and Control	Application Layer Protocol	PowerShell-based implant used for ongoing control and tasking

Phoenix PhaaS Kit Powering Global Smishing-Driven Phishing Campaigns

Tactics	Techniques	Description
Initial Access	Phishing via SMS	Delivery of fraudulent SMS messages impersonating trusted brands
Execution	User Execution	Victims manually interact with phishing pages following SMS lures
Credential Access	Credentials from Web Forms	Collection of personal, financial, and payment data
Defense Evasion	Geofencing and Traffic Filtering	Restriction of phishing content to targeted regions and devices
Collection	Automated Data Collection	Centralized harvesting and real-time monitoring of victim input

Appendix B – Threat Severity Ratings & Definitions

In this newsletter, each campaign, breach, and vulnerability has been assigned a severity level—High, Medium, or Low—based on a contextual assessment of its impact, relevance to the financial sector, regional exposure, and exploit status.

A **High** severity rating is assigned to threats that involve widely used technologies, nation-state actors, or actively exploited zero-day vulnerabilities. These incidents pose a significant risk due to their potential for widespread disruption, data compromise, and strategic targeting of core infrastructure.

A **Medium** severity rating reflects threats that are technically impactful but have limited regional exposure or affect less prominent entities. These may include ransomware attacks on mid-sized firms, blockchain-based malware delivery mechanisms, or smart contract exploits in DeFi platforms.

A **Low** severity rating is applied to threats that are peripherally related to the financial sector, have no confirmed impact in the region, or target non-critical systems. These include credential stuffing attacks on platforms outside the MENA region or supply chain breaches that have not yet affected local infrastructure.

Threat Score Ratings & Definitions

1. **Severe**: Widespread, high-impact attacks ongoing. Critical infrastructure affected by destructive malware or multiple zero-day exploits.
2. **High**: Confirmed breaches or coordinated campaigns in progress. High-impact threat actor activity and zero-day exploitation observed.
3. **Elevated**: Active exploitation of known vulnerabilities or targeted campaigns detected. Some breaches or zero-days may surface.
4. **Guarded**: Slight increase in malicious activity. Low-level campaigns or threat actor reconnaissance may be underway.
5. **Normal**: No significant cyber activity beyond baseline. No major breaches, campaigns, or zero-day activity observed.

Appendix C – Traffic Light Protocol (TLP) Definitions and Usage

TLP	When should it be used?	How should it be shared?
TLP:Red	Sources may use TLP:RED when information cannot be effectively acted upon without significant risk for the privacy, reputation, or operations of the organizations involved. For the eyes and ears of individual recipients only, no further.	Recipients may not share TLP:RED information with any parties outside of the specific exchange, meeting, or conversation in which it was originally disclosed. In the context of a meeting, for example, TLP:RED information is limited to those present at the meeting. In most circumstances, TLP:RED should be exchanged verbally or in person.

TLP:Amber+Strict	Sources may use TLP:AMBER+STRICT when information requires support to be effectively acted upon, yet carries risk to privacy, reputation, or operations if shared outside of the organization.	Recipients may share TLP:AMBER+STRICT information only with members of their own organization on a need-to-know basis to protect their organization and prevent further harm.
TLP:Amber	Sources may use TLP:AMBER when information requires support to be effectively acted upon, yet carries risk to privacy, reputation, or operations if shared outside of the organizations involved. Note that TLP:AMBER+STRICT should be used to restrict sharing to the recipient organization only.	Recipients may share TLP:AMBER information with members of their own organization and its clients on a need-to-know basis to protect their organization and its clients and prevent further harm.
TLP:Green	Sources may use TLP:GREEN when information is useful to increase awareness within their wider community.	Recipients may share TLP:GREEN information with peers and partner organizations within their community, but not via publicly accessible channels. Unless otherwise specified, TLP:GREEN information may not be shared outside of the cybersecurity or cyber defense community.
TLP:Clear	Sources may use TLP:CLEAR when information carries minimal or no foreseeable risk of misuse, in accordance with applicable rules and procedures for public release.	Recipients may share this information without restriction. Information is subject to standard copyright rules.

Appendix D - Acronyms & Technical Terms

Term / Acronym	Meaning / Description
AI	Artificial Intelligence: Used in the newsletter to describe automation that helped generate or enhance lures or code in observed campaigns.
AES	Advanced Encryption Standard: Mentioned as part of encryption used for payloads or data handling in described activity. [360.cn], [anchor.host]
AES-GCM	AES in Galois/Counter Mode: Described as an encryption mode used to wrap stolen data in the SAP CAP supply-chain incident. [anchor.host]
Allowlist	A “permit list” of allowed classes/inputs; weaknesses in allowlist enforcement were central to the Apache MINA deserialization flaws.
ANGLE	A graphics abstraction layer referenced as affected by multiple Chrome vulnerabilities (memory-safety and overflow issues).
Anti-analysis	Techniques intended to hinder security analysis; referenced in ransomware feature gaps and phishing platform anti-crawler controls. [360.cn], [reversinglabs.com]

AppDomain	.NET application domain concept referenced in modular loader behavior in the OilRig phishing chain. [blog.checkpoint.com]
Authentication Bypass	A flaw type where authentication checks can be bypassed, enabling unauthorized access (e.g., cPanel & WHM issue).
Base64	An encoding method repeatedly used in attack chains to store/transfer configuration, code, or embedded data. [blog.checkpoint.com], [arcticwolf.com]
Backdoor	A hidden access mechanism enabling persistent or covert control; seen in plugin update-channel abuse and SSH-over-Tor tunnels. [socket.dev], [arcticwolf.com]
Bun	A JavaScript runtime downloaded and executed during npm install in the SAP CAP supply-chain compromise. [anchor.host]
C#	A programming language used to compile malicious components or loaders in multiple chains described. [blog.checkpoint.com]
C2 / Command and Control	Infrastructure/channel attackers use to control compromised systems or receive data; described via Telegram Bot channels and Tor-based access. [blog.checkpoint.com], [arcticwolf.com]
Calendly	A scheduling service abused as part of the initial social engineering pipeline in the BlueNoroff intrusion. [360.cn]
CAP	Cloud Application Programming Model: Mentioned in relation to SAP's JavaScript ecosystem affected by the npm supply-chain compromise. [anchor.host]
CI/CD	Continuous Integration / Continuous Delivery: Build/deployment environments highlighted as high-risk because malicious code executed during dependency installation. [anchor.host]
ClickFix	A clipboard-injection style technique described as used during the fake meeting lure to push malicious content via user interaction. [360.cn]
CSC	UAE Cyber Security Counsil
CVE	Common Vulnerabilities and Exposures: Standard identifier for publicly tracked vulnerabilities (e.g., CVE-2026-32202).
CVSS	Common Vulnerability Scoring System: Severity scoring referenced across multiple vulnerabilities (e.g., 9.8/10/4.3).
CWE	Common Weakness Enumeration: A taxonomy label used to describe vulnerability classes (e.g., CWE-693).
CustomXMLParts	A document feature referenced as a location used to store/retrieve embedded content in malicious Excel lures. [blog.checkpoint.com]
DCSync	A technique referenced as likely used to extract directory credentials by simulating domain controller replication. [socradar.io]
DLL Sideloading	Loading a malicious DLL via a legitimate signed executable; used to stage implants in the espionage campaign. [socradar.io]
DoS	Denial of Service: A condition where a service/app becomes unavailable (e.g., Notepad++ crash risk).
EDR/XDR	Endpoint Detection and Response / Extended Detection and Response: Tools referenced as needing updates/tuning for detecting malicious execution patterns.
ESXi	VMware ESXi: A platform explicitly mentioned as affected by the VECT ransomware variant set. [360.cn]
Exchange / Microsoft Exchange	An email server platform exploited via older vulnerabilities in the espionage campaign. [socradar.io]

EWS	Exchange Web Services: Referenced as used by a custom tool to export mailbox content. [socradar.io]
Fileless	A technique where execution relies on scripts/in-memory activity rather than dropping traditional files; referenced in the BlueNoroff chain. [360.cn]
Format String Injection	A vulnerability type where crafted format strings influence processing, causing crashes or memory disclosure (Notepad++ case).
Geofencing	Traffic filtering based on geography/IP used to restrict who can access phishing pages or content. [reversinglabs.com]
GitHub (as infrastructure)	Referenced as abused for configuration hosting, payload staging, or as an exfiltration channel in supply-chain activity. [blog.checkpoint.com], [anchor.host]
GHES	GitHub Enterprise Server: Affected by the CVE-2026-3854 command-injection vulnerability.
Hidden Service / Onion Service	A Tor service type used to expose internal services via anonymized addressing, enabling stealthy remote reachability. [arcticwolf.com]
IIS	Internet Information Services: A Windows web server component exploited as part of the espionage initial access. [socradar.io]
IOC	Indicator of Compromise: References such as domains/hashes; explicitly excluded from newsletter outputs by requirement.
LNK	Windows shortcut file format used as an execution trigger in spear-phishing delivery chains. [arcticwolf.com]
LSB Steganography	Least Significant Bit steganography: Hiding encrypted configuration inside images, then extracting it during execution. [blog.checkpoint.com]
LSASS	Local Security Authority Subsystem Service: Mentioned in context of credential extraction via memory dumping. [socradar.io]
MFA	Multi-Factor Authentication: A control targeted by “live-phishing” to intercept/force OTP re-entry and bypass protections. [reversinglabs.com]
MHTML	A web archive format referenced in a Chrome vulnerability (“Race in MHTML”).
MITRE ATT&CK	A framework referenced for mapping tactics/techniques (e.g., phishing, execution, impact). [trendmicro.com], [reversinglabs.com]
MTA	Multi-Target Application: Mentioned in relation to SAP deployment workflows and the “mbt” build tool. [anchor.host]
N-day	A vulnerability that is already known/patched but remains exploitable in unpatched environments; described in the Exchange/IIS exploitation. [socradar.io]
nativeLang.xml	A Notepad++ localization file used as the malicious input vector in CVE-2026-3008.
npm	A JavaScript package registry/ecosystem where malicious dependencies and compromised packages were described. [group-ib.com], [anchor.host]
Obfuscation	Intentional code transformation to hinder analysis; described as used for large payloads and single-line scripts. [anchor.host], [group-ib.com]
obfs4	A Tor traffic obfuscation method referenced to disguise Tor communications as ordinary TCP-like traffic. [arcticwolf.com]
OTP	One-Time Password: A step monitored/targeted in phishing flows to bypass MFA via real-time intervention. [reversinglabs.com]

Out-of-Bounds Read/Write	Memory-safety flaw type listed among Chrome high-severity issues (ANGLE component).
PhaaS	Phishing-as-a-Service: A subscription-like platform model that provides templates, dashboards, filtering, and victim management to operators. [reversinglabs.com]
PBKDF2	Password-Based Key Derivation Function 2: Described as used in string decoding/encryption logic within an obfuscated payload. [anchor.host]
Phishing / Spear-phishing	Deceptive messages/pages designed to capture secrets; “spear-phishing” indicates targeted delivery to specific individuals. [360.cn], [blog.checkpoint.com]
PII	Personally Identifiable Information: Described as harvested in phishing operations alongside payment data. [reversinglabs.com]
Plugin Update Checker	A library used by a WordPress plugin to fetch updates from a third-party server, enabling remote code delivery via update checks. [socket.dev]
PowerShell	A Windows scripting engine used for fileless execution, loaders, and stealthy automation in multiple chains. [360.cn], [arcticwolf.com]
Preinstall Script	An npm lifecycle hook that runs during install; abused to download and execute additional code before normal package use. [anchor.host]
ProxyLogon	A chain of Exchange vulnerabilities referenced as an example of the exploited server entry method. [socradar.io]
RAR	An archive format/tool referenced as used for packaging collected data for exfiltration. [socradar.io]
RCE	Remote Code Execution: Ability for an attacker to execute code on a remote system (notably in Apache MINA and GitHub issues).
RDP	Remote Desktop Protocol: A remote access service referenced as mapped/exposed via Tor in the SSH-over-Tor tunneling setup. [arcticwolf.com]
RSA-OAEP	RSA Optimal Asymmetric Encryption Padding: Described as used to encrypt harvested data before exfiltration. [anchor.host]
Sandbox / VM Checks	Environmental checks used to avoid analysis environments; described via heuristic checks before payload continuation. [arcticwolf.com]
Scheduled Task	A Windows persistence mechanism used to run malicious components at logon or periodically. [socradar.io], [arcticwolf.com]
ShadowPad	A modular malware family deployed in the espionage campaign via DLL sideloading and registry staging. [socradar.io]
Smishing	SMS phishing: Fraudulent SMS messages used to deliver phishing links and drive victims to credential/payment harvesting pages. [reversinglabs.com]
SMB	Server Message Block: A file-sharing service referenced as mapped/exposed through Tor hidden services. [arcticwolf.com]
Spoofing	Deceptive presentation designed to mislead users (e.g., Windows Shell spoofing via crafted files).
Steganography	Hiding data inside other content (e.g., images) to conceal configuration or payload details. [blog.checkpoint.com]
Supply Chain Attack	Compromise introduced through software dependencies, packages, plugins, or update mechanisms rather than direct victim targeting. [group-ib.com], [socket.dev], [anchor.host]
Telegram Bot API	A messaging-based mechanism used for command/control or data handling in multiple activities described. [blog.checkpoint.com], [360.cn], [trendmicro.com], [reversinglabs.com]

Tor	The Onion Router: An anonymizing network used to build hidden services and conceal access paths for persistence. [arcticwolf.com]
Type Confusion	A vulnerability class listed among Chrome high-severity items (V8) that can lead to unsafe execution paths.
Use-After-Free (UAF)	A memory-safety flaw type heavily represented in Chrome's listed vulnerabilities across multiple components.
VBA	Visual Basic for Applications: Macro language referenced as used to decode embedded content and trigger multi-stage payloads. [blog.checkpoint.com]
V8	Chrome's JavaScript engine referenced as affected by a "Type Confusion" vulnerability.
Web Shell	A server-side backdoor that enables command execution via web requests; used after Exchange/IIS exploitation. [socradar.io]
WebRTC	Web Real-Time Communication: Chrome component referenced in multiple vulnerabilities (UAF and overflow types).
WHM	WebHost Manager: The administrative interface paired with cPanel, affected by the described authentication issue.
Wiper	Destructive behavior that renders data unrecoverable; described in the VECT case for large files due to design flaws. [360.cn]
WMIC	Windows Management Instrumentation Command-line: Used for lateral movement in the espionage activity. [socradar.io]
XOR	A simple cryptographic operation referenced as part of configuration decryption logic in multi-stage chains. [blog.checkpoint.com], [arcticwolf.com]
X-Stat Headers	Internal service headers referenced as the injection point for crafted git push options in the GitHub vulnerability description.