TLP:CLEAR

ADGM THREAT INTELLIGENCE NEWSLETTER

PREPARED BY ADGM CTI

CATEGORY	ACTIONABLE
AUDIENCE	ADGM FSRA ENTITIES
DATE	11/6/2026
OVERALL THREAT SCORE	ELEVATED
TARGET SECTOR	FINANCIAL SERVICES
TARGET REGION	MENA & GLOBAL
ATTRIBUTION	MULTIPLE
TLP	CLEAR

WEEKLY SUMMARY REPORT – 11 June 2026

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

Summary

This week's threat landscape was defined by large-scale credential theft, stealthy persistence, and disruption risks across communications, cloud, edge, and developer environments. Adversaries used targeted social engineering, covert payloads, and memory-based techniques to steal data and maintain access. Supply chain risks remained significant, with malicious npm packages and a VS Code browser flaw exposing developer and CI/CD secrets. Vulnerabilities in HTTP/2, Envoy, HPE Telco NFV Orchestrator, and SolarWinds Serv-U further highlighted the business risk of internet-facing services. From a financial sector perspective, these developments underscore the need to secure internet-facing services, developer toolchains, cloud and edge environments, and trusted communication channels against theft and disruption. Priorities include patching vulnerable systems, reducing exposure, strengthening identity controls, and tightening third-party access. Attackers continue to pair easy initial access with stealthy persistence, making early detection and strong patch discipline essential.

ADGM THREAT INTELLIGENCE SUMMARY

Error 524 Decoy: A Large-Scale Smishing Operation Targeting Globally [Campaign] [High]

OP-512 Targets Legacy IIS Servers with Custom Web Shell Framework [Campaign] [High]

PCPJack Hijacks Cloud Servers to Build a Hidden SMTP Relay Network [Campaign] [High]

Miasma Supply Chain Campaign Compromises Red Hat Cloud Services npm Packages [Campaign] [Medium]

Staged Malspam Campaign Delivers Fileless .NET Loader Through Multi-Step Infection Chain [Campaign] [Medium]

VerdantBamboo Used Edge Appliances and Managed Access to Sustain Long-Term Intrusions [Campaign] [Medium]

Microsoft Patches github.dev Zero-Day Enabling One-Click GitHub Token Theft [Vulnerability] [High]

Envoy Patches HTTP/2 Bomb Memory Exhaustion Flaw in Downstream HTTP/2 Processing [Vulnerability] [High]

HPE Fixes Multiple Vulnerabilities in Telco NFV Orchestrator [Vulnerability] [High]

SolarWinds Fixes Actively Exploited Serv-U DoS Vulnerability [Vulnerability] [High]

Cisco Catalyst SD-WAN Manager Privilege Escalation Vulnerability Exploited [Vulnerability] [High]

Unpatched Windows Search URI Flaw Exposes NTLMv2 Hashes via SMB [Vulnerability] [Medium]

Name	Threat Severity Rating	TLP	Attribution	Originating Source
Error 524 Decoy: A Large-Scale Smishing Operation Targeting Globally	HIGH	CLEAR	Campaign	Open Source

Executive Summary

Group-IB researchers identified a large and highly organized smishing and phishing campaign that has been active since late 2025, targeting victims across 72 countries by impersonating more than 267 brands, especially in the telecommunications and financial services sectors. The attackers use SMS messages and phishing links to lure users to fake websites designed to steal credit card details and personal information. A key part of the attack is its ability to avoid detection. Victims are first shown fake Cloudflare error pages, such as an "Error 524" timeout message, to make the site look legitimate or inactive, while the real phishing content is only displayed to selected users based on their location and whether they are using a mobile device.

This campaign may impact organizations exposed to brand impersonation and payment fraud, and organizations in the financial sector should be aware that the activity included financial services targets and was designed to collect personal and payment card data through a staged mobile workflow.

Technical Details

- Victims received SMS messages crafted with urgency themes and local-number spoofing, with shortened URLs used to reduce suspicion before the victim clicked.
- After the link was opened, the site checked the visitor's country, language, currency, and mobile user-agent to decide whether to show the phishing page or a decoy.
- Users who did not match the targeting rules were shown fake Cloudflare-style error pages such as 524, 300, or 313, helping the campaign hide its malicious content from researchers and service providers.
- For qualified victims, the phishing page loaded a lightweight single-page application that used Base64-encoded content decoded at runtime instead of exposing the full content immediately.
- The first step asked for a national registry or identification number, creating a familiar entry point before advancing the victim further into the process.
- The next pages displayed a reward or benefit and then requested personal details such as full name, address, email address, and phone number.
- After building trust, the final form requested payment card details, including card number, expiration date, and CVV, with only basic checksum validation.
- Submitted data was exfiltrated in real time through encrypted WebSocket channels using binary payloads, and heartbeat traffic was used to maintain the session.
- After submission, victims were redirected to the legitimate brand website, helping the interaction appear routine and reducing immediate suspicion.
- The infrastructure relied on Cloudflare as a reverse proxy, while origin hosting frequently used Tencent Cloud and Alibaba to support rapid domain cycling and concealment.

Recommendations

- Monitor for SMS lures and phishing pages that imitate rewards, benefits, delivery notices, or account-related communications.
- Prioritize detection of shortened links and mobile-only phishing pages that use geofencing or device checks before revealing malicious content.
- Review fraud monitoring use cases for workflows that collect identity details first and payment card data later in the same session.
- Monitor for suspicious encrypted WebSocket activity associated with phishing pages and staged credential theft workflows.
- Reinforce user awareness around unsolicited SMS messages that create urgency and direct users to claim rewards, benefits, or service-related offers.

For a detailed breakdown of tactics, techniques, and procedures (TTPs), refer to the Annexure – Error 524 Decoy: A Large-Scale Smishing Operation Targeting Globally

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
OP-512 Targets Legacy IIS Servers with Custom Web Shell Framework	HIGH	CLEAR	Campaign	Open Source

Executive Summary

ReliaQuest has identified OP-512 as a newly observed, likely China-linked espionage cluster that targeted a legacy, internet-facing IIS server. The attack appears to have leveraged the exposed web server as the entry point, after which the actor deployed a custom web shell framework that enabled covert remote access, centralized control, and multiple fallback command paths. The operation showed clear signs of long-term persistence, with evidence of access dating back 75 days before the attacker returned to expand control, escalate privileges, and strengthen foothold on the system.

This activity may impact organizations running legacy internet-facing IIS infrastructure, and organizations in the financial sector should be aware that the intrusion relied on stealthy persistence, privilege escalation, and cryptographically protected web shell access designed to evade traditional signature-based detection.

Technical Details

- The compromised host was an IIS server running Windows Server 2016 with end-of-life ".NET" Framework 4.0, creating a plausible attack surface on an internet-facing application.

- Telemetry showed signs of activity on the same server 75 days earlier, indicating the operator had previous access and later returned to continue the intrusion.
- The attacker first placed a malicious ".aspx" (Active Server Pages Extended) file manager web shell in the application upload directory, giving them file operations and a way to register the compromise automatically.
- When the shell was accessed, it encoded its own URL and sent it to attacker-controlled infrastructure through a DNS query, with an HTTP fallback if the DNS request failed.
- The actor then deployed two ".ashx" (ASP[.]NET Web Handler) command handlers to the same directory, adding separate remote command paths after the initial web shell was established.
- These command handlers processed requests through Base64 decoding, RC4 decryption, and RSA signature verification, and each was generated with different keys and randomized code elements.
- After access was established, the attacker loaded multiple post-exploitation tools directly into the IIS worker process memory instead of writing them to disk.
- The operator then attempted privilege escalation using Potato Suite tools and ran encoded privilege-check commands to verify the security context on the host.
- Prevention controls terminated the malicious process, but IIS automatically restarted the worker process, allowing the activity to continue across new process instances.
- The web shells also used time stomping to blend with older files, while compiled malicious DLLs remained in ASP[.]NET temporary compilation directories even after the original shell files were removed.

Recommendations

- Prioritize migration, segmentation, or close monitoring of internet-facing IIS servers running legacy or end-of-life ".NET" applications.
- Hunt for unusual outbound DNS requests from w3wp[.]exe, especially long encoded subdomains that may indicate self-reporting web shell behavior.
- Monitor IIS worker processes for reflective ".NET" assembly loading and other in-memory activity that may not leave clear artifacts on disk.
- Review ASP[.]NET temporary compilation directories for unexpected DLL generation outside normal deployment activity, even if suspicious web files have already been removed.
- Isolate affected hosts during response, as repeated process termination alone may not stop the intrusion if IIS automatically restarts the worker process.

For a detailed breakdown of tactics, techniques, and procedures (TTPs), refer to the Annexure – OP-512 Targets Legacy IIS Servers with Custom Web Shell Framework

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
PCPJack Hijacks Cloud Servers to Build a Hidden SMTP Relay Network	HIGH	CLEAR	Campaign	Open Source

Executive Summary

Researchers uncovered two unauthenticated open directories on infrastructure already tied to PCPJack, exposing source code, binaries, deployment logs, and live command-and-control data that revealed a large-scale cloud abuse operation. The attack appears to start with PCPJack compromising exposed cloud services and vulnerable web applications then stealing credentials, spreading to additional hosts, and establishing persistence on Linux systems. From there, the operators used Sliver and unmodified Chisel binaries to turn compromised AWS, Google Cloud, and Azure servers into persistent SOCKS5 proxies, but only after testing whether each host could successfully relay outbound email.

This campaign may impact organizations running exposed or weakly monitored cloud-hosted Linux workloads, and organizations in the financial sector should be aware that the activity focused on turning hijacked servers into covert email relay infrastructure at scale.

Technical Details

- Hunt[.]io identified exposed directories on attacker infrastructure that revealed deployment scripts, Chisel binaries, state files, and an active working directory without authentication, providing visibility into how the operation was managed.
- The recovered toolkit included Chisel binaries for multiple Linux architectures, allowing the operator to deploy the same tunneling capability across a broad range of cloud-hosted systems.
- The deployment process used Sliver beacons to identify active Linux hosts, assign each one a SOCKS5 proxy port, and push the tunneling payload to compromised systems.
- In the earlier deployment version, hosts were first checked for outbound access to an SMTP service, showing that email relay capability was a core requirement from the start.
- Later versions broadened deployment to more hosts and shifted the SMTP testing to a separate verification process, suggesting the operator prioritized scale before filtering for usable relays.
- After upload, the binary was copied to a hidden path on the victim host and persistence was created through either a systemd service or a recurring cron watchdog.
- The operator also used a diagnostic script to check whether the tunneling binary was present, whether it was running, whether the command server was reachable, and whether persistence had been installed correctly.
- A background verification process continuously tested active tunnels by performing an SMTP handshake and kept only working proxies in the usable pool.
- Verified proxy lists were synchronized to another downstream server, indicating the relay network was maintained as an active service rather than a one-time access operation.

- Recovered state files confirmed one deployment wave uploaded and executed the tooling on 230 compromised hosts, while other artifacts suggested related activity had occurred before the recovered run set.

Recommendations

- Review cloud-hosted Linux workloads for unexpected tunneling tools, hidden binaries, and persistence created through systemd services or cron jobs.
- Monitor for unusual outbound connections associated with reverse tunnels, SOCKS5 proxy behavior, and repeated SMTP validation traffic from server workloads.
- Investigate exposed directories, staging paths, and temporary working locations that may reveal attacker tooling or signs of post-compromise activity.
- Hunt for signs of centralized deployment activity that pushes the same tunneling payload across multiple Linux hosts through remote management or beacon infrastructure.
- Validate that cloud workloads are segmented and closely monitored so compromised hosts cannot be repurposed into relay or proxy infrastructure.

For a detailed breakdown of tactics, techniques, and procedures (TTPs), refer to the Annexure – PCPJack Hijacks Cloud Servers to Build a Hidden SMTP Relay Network

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
Miasma Supply Chain Campaign Compromises Red Hat Cloud Services npm Packages	MEDIUM	CLEAR	Campaign	Open Source

Executive Summary

Researchers have identified a supply chain campaign affecting multiple "@redhat-cloud-services" npm packages, where malicious versions used a preinstall hook to run automatically during npm install, decode staged payloads, and harvest secrets from developer systems and CI/CD environments before the package was even used.

This campaign may impact organizations that rely on affected npm packages in developer workflows or build pipelines, and organizations in the financial sector should be aware that the payload was designed to steal cloud, registry, and CI secrets and potentially propagate further through compromised publishing credentials.

Technical Details

- The campaign affected multiple packages in the "@redhat-cloud-services" npm scope, with malicious code triggered through node index[.]js in the preinstall lifecycle script.

- The malicious index[.]js was unusually large and heavily obfuscated, using staged decoding and encrypted blobs to hide its real behavior from casual review.
- During installation, the loader decrypted additional payloads, wrote them to temporary files, executed them with Bun, and then removed the temporary artifacts.
- The payload collected environment variables and searched local files for sensitive material, including cloud credentials, npm tokens, Git credentials, SSH keys, Kubernetes configuration, and other developer secrets.
- On GitHub Actions runners, the malware attempted to read process memory to recover masked secrets directly from the Runner[.]Worker process.
- Exfiltration was encrypted and sent over HTTPS, and the malware also implemented a GitHub-based fallback channel to store stolen data through repository API activity.
- The campaign also included worm-like behavior, using stolen npm credentials and the bypass_2fa publish option to push additional backdoored package versions.
- Analysis indicated the malicious releases were published through GitHub Actions OIDC from the upstream repository, suggesting the publishing pipeline itself was compromised.

Recommendations

- Treat any developer machine or CI/CD pipeline that installed affected versions as potentially compromised and rotate exposed credentials immediately.
- Review dependency usage and remove or replace affected package versions from build pipelines and developer environments.
- Inspect CI/CD runners for unusual secret access, including attempts to read process memory or collect masked workflow secrets.
- Review npm publishing permissions and tokens for signs of unauthorized package publication or token misuse.
- Monitor developer systems for unexpected persistence or malicious changes in local tooling and workspace configuration files.

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
Staged Malspam Campaign Delivers Fileless .NET Loader Through Multi-Step Infection Chain	MEDIUM	CLEAR	Campaign	Open Source

Executive Summary

Researchers at Huntress have identified a malspam campaign that used a malicious HTML attachment to send recipients through a trusted advertising domain and a personalized lure page, then delivered a ZIP archive containing a JScript loader that launched PowerShell and a fileless ".NET" loader through several staged steps.

This campaign may impact organizations exposed to email-borne malware delivery and organizations in the financial sector should be aware that the activity used layered in-memory execution, anti-analysis checks, telemetry tampering and signed-process injection to reduce visibility after initial access.

Technical Details

- Initial access began with a malspam email carrying an HTML attachment that redirected the recipient through a legitimate high-reputation advertising service before reaching the malicious delivery chain.
- The lure page rebuilt itself using the recipient's email address, dynamically adjusted branding, and displayed location-based details to make the message appear more convincing without requiring a custom page for each target.
- When the victim interacted with the page, the site delivered a ZIP archive containing a heavily padded and obfuscated JScript file rather than the final malware.
- The JScript loader copied itself to a stable public location, relaunched from there, repaired an embedded blob, and dropped a PowerShell script to continue execution.
- The PowerShell stage checked internet connectivity, looked for sandboxing and analysis tools, and rebooted the system if it detected conditions associated with examination or failed staging.
- The next PowerShell stage assembled and reflectively loaded a ".NET" component, then passed a trusted Microsoft-signed binary as part of the execution chain to blend malicious activity with legitimate tooling.
- The ".NET" loader then performed broader anti-analysis checks, cleaned up earlier artifacts, created a nested staging directory, and attempted to weaken host protections before continuing.
- Persistence was established through multiple user-level startup mechanisms, and the final payload was launched with hidden PowerShell execution from the staging location.
- A subsequent ".NET" DLL handled process hollowing, using common RunPE-style APIs to inject the next payload into a legitimate signed process rather than running it directly.
- The malware also profiled the host by enumerating security products and checking hardware details, including GPU-related information, before establishing command-and-control communications.

Recommendations

- Inspect HTML attachments and redirect-based email lures more closely, especially when they rely on trusted third-party domains before handing off to malicious content.
- Review detections for script-based execution chains involving JScript, PowerShell, reflective ".NET" loading, and hidden follow-on execution.
- Monitor for anti-analysis behavior such as debugger checks, sandbox detection, unexpected host reboots, and attempts to evade telemetry collection.
- Hunt for abuse of signed Microsoft utilities and legitimate processes as injection targets to conceal malicious execution.
- Strengthen email security controls that can analyze malicious attachments before delivery and identify staged payload delivery chains early.

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
VerdantBamboo Used Edge Appliances and Managed Access to Sustain Long-Term Intrusions	MEDIUM	CLEAR	Campaign	Open Source

Executive Summary

Researchers at Volexity have identified 'VerdantBamboo' as a sophisticated advanced persistent threat (APT) actor, compromising internet-facing and edge-adjacent systems including an Egnyte Storage Sync appliance, a pfSense firewall, and a Synology NAS, then using valid credentials, web SSL VPN access, and malware implants such as BRICKSTORM, AGENTPSD, and PLENET to regain access and move through victim environments over an extended period.

This campaign may impact organizations that rely on managed infrastructure, edge devices, or systems without strong monitoring coverage, and organizations in the financial sector should be aware that the activity used stealthy footholds on peripheral systems to proxy access into internal environments and Microsoft 365.

Technical Details

- The investigation began after suspicious traffic was observed from a Linux-based Egnyte Storage Sync virtual appliance communicating with a threat-actor-controlled domain behind Cloudflare and using DNS over HTTPS via Google public DNS.
- Volexity determined the appliance had been compromised by VerdantBamboo, which had used valid SSH access through the egnyteservice account to reach the system.

- The actor then abused a sudo configuration weakness that allowed the account to use tee as root, giving it a local privilege escalation path to write files into privileged directories.
- After gaining elevated access, VerdantBamboo deployed BRICKSTORM to the appliance and used a temporary cron-based launch method, while also placing AGENTPSD in crontab as a fallback reverse shell in case the primary implant stopped working.
- Volexity found evidence that these implants had remained on the Storage Sync system for at least 18 months, showing long-term persistence even though the attacker manually relaunched BRICKSTORM when needed.
- The actor used the compromised edge system and credentials obtained through the victim's VPN access path to blend into legitimate traffic and access the victim's Microsoft 365 environment in a way that could evade Conditional Access restrictions.
- During the wider investigation, Volexity found that the victim's managed services provider had also been compromised, including a pfSense firewall carrying a FreeBSD variant of BRICKSTORM with persistence added through startup script modification.
- After initial remediation, VerdantBamboo returned by using stolen administrative credentials on an internet-exposed firewall, enabling web SSL VPN access and using that path to reconnect internally.
- The renewed access was used to reach a Synology NAS, enable SSH, and deploy PLENET, a previously undocumented ".NET" Core backdoor compiled with Native AOT.
- Volexity assessed that VerdantBamboo's approach focused on devices lacking EDR coverage, allowing the actor to maintain covert access on edge infrastructure rather than relying on heavily monitored endpoints.

Recommendations

- Review externally accessible appliances, firewalls, NAS systems, and synchronization platforms for unauthorized SSH access, malware execution, and suspicious outbound traffic.
- Audit administrative and service accounts used on managed infrastructure, especially where access is not protected by MFA or where legacy local accounts remain active.
- Validate that edge devices and managed systems are included in logging, threat detection, and regular security reviews rather than treated as low-visibility infrastructure.
- Review VPN and firewall configurations for unauthorized enablement of remote access services and investigate unusual internal activity sourced from trusted infrastructure paths.
- Assess managed service provider connections and administrative dependencies to identify whether third-party access or compromise could provide a path into internal environments.

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
Microsoft Patches github.dev Zero-Day Enabling One-Click GitHub Token Theft	HIGH	CLEAR	Vulnerability	Open Source

Executive Summary

Microsoft fixed a zero-day vulnerability in the browser version of GitHub[.]dev. A specially crafted repository link could cause a malicious notebook to open, run JavaScript in a VS Code webview, simulate keyboard shortcuts, install attacker-controlled extension code, and steal a user's GitHub OAuth token without any further action from the victim.

This vulnerability may impact organizations that use browser-based development workflows, and organizations in the financial sector should be aware that a stolen token could provide read and write access across all repositories available to the affected user session, rather than only the repository initially opened.

Technical Details

- The issue affected github[.]dev, where GitHub posts a broadly scoped OAuth session token to the browser-based VS Code editor when a repository is opened.
- The attack chained five VS Code behaviors together, turning individually expected functions into a complete token-theft path.
- The first stage abused webview keydown event bubbling, allowing JavaScript inside a sandboxed webview to generate keyboard events treated as real editor input.
- The second stage used a malicious Jupyter notebook, where HTML inside a markdown cell executed JavaScript through an image onerror handler.
- Once running, the payload waited for VS Code to display an extension recommendation notification tied to the repository contents.
- It then triggered the default shortcut for accepting the primary notification action, silently approving installation of a recommended extension.
- The next stage relied on local workspace extensions stored under ".vscode/extensions/," which inherited workspace trust and bypassed the usual publisher trust prompt.
- That local extension added a custom keybinding, which the notebook payload triggered to run extension installation logic with skipPublisherTrust enabled.
- After activation, the malicious extension accessed the pre-loaded GitHub token and queried repository access, showing that the token was not limited to the opened repository.
- Microsoft applied stopgap fixes on June 3, 2026, including blocking the skipPublisherTrust bypass path and preventing keydown events from bubbling out of notebook webviews.

Recommendations

- Confirm that users access the current github[.]dev experience with Microsoft's updated protections in place before opening untrusted repository links.

- Review whether any suspicious extensions were installed through github[.]dev sessions and remove anything not explicitly expected.
- Treat exposed GitHub tokens as compromised and rotate them if there is any indication a user opened a malicious proof-of-concept or similar repository.
- Clear github[.]dev site data in browsers used for sensitive development work if there is concern about prior session trust state or suspicious activity.
- Review extension installation activity, unusual repository enumeration, and unexpected code pushes that may have been performed with a legitimate user token.

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

back to top

Name	Threat Severity Rating	TLP	Attribution	Originating Source
Envoy Patches HTTP/2 Bomb Memory Exhaustion Flaw in Downstream HTTP/2 Processing	HIGH	CLEAR	Vulnerability	Open Source

Executive Summary

Envoy has addressed a high-severity HTTP/2 flaw in downstream request processing that allows a remote client to send crafted cookie headers, use HPACK indexed references to trigger large decoded header allocations, and prolong memory retention through HTTP/2 flow-control stalling until the proxy exhausts memory.

This vulnerability may impact organizations exposing HTTP/2 services through Envoy, and organizations in the financial sector should be aware that successful exploitation could cause service disruption through excessive memory growth and out-of-memory termination under relatively modest attacker-side resource requirements.

Technical Details

- The attack combines two behaviors: incomplete cookie header size accounting during request validation and HPACK limits enforced on encoded bytes rather than total decoded header size.
- An attacker can seed the HPACK dynamic table with a header once and then reuse indexed references so that very small wire input causes much larger header allocations in memory.
- The technique becomes more effective when the attacker advertises a zero-byte flow-control window, preventing the server from completing the response and delaying memory reclamation.
- Calif's testing described strong amplification against Envoy, where a single client could hold roughly 32 GB of server memory in around 10 seconds on Envoy 1.37.2.
- In Envoy, the issue is especially effective when cookie fragments are merged only after header validation, allowing oversized logical cookie content to bypass intended protections.

- Envoy's advisory states the attack can lead to rapid abnormal memory growth and OOM termination, including container exit status 137 in some environments.
- The advisory also notes a secondary operational effect where very large decoded cookies forwarded upstream may exceed upstream header limits and trigger transient failures.

Recommendations

- Upgrade Envoy to a patched release immediately, using the vendor-listed fixed versions where applicable.
- Disable downstream HTTP/2 where operationally feasible until remediation is completed.

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

back to top

Name	Threat Severity Rating	TLP	Attribution	Originating Source
HPE Fixes Multiple Vulnerabilities in Telco NFV Orchestrator	HIGH	CLEAR	Vulnerability	CSC

Executive Summary

HPE has addressed multiple vulnerabilities in Telco Network Function Virtualization Orchestrator that could allow attackers to cause denial-of-service conditions, gain unauthorized remote access, trigger memory corruption or buffer overflow conditions, bypass input validation mechanisms, or otherwise compromise affected systems.

This activity may impact organizations running vulnerable orchestration environments, and organizations in the financial sector should be aware that successful exploitation could affect service availability, system integrity, and the security of virtualized network management platforms depending on deployment and exposure.

Technical Details

- HPE observed multiple vulnerabilities affecting Telco Network Function Virtualization Orchestrator, indicating a broad set of weaknesses across the platform.
- The reported issues included paths that could lead to denial-of-service conditions, which may disrupt normal platform operations and service delivery.
- HPE also noted vulnerabilities that could allow unauthorized remote access, increasing the risk of attacker interaction with affected systems.
- Some of the vulnerabilities could trigger memory corruption or buffer overflow conditions, creating additional risk to system stability and integrity.
- Input validation weaknesses were also identified, which could allow attackers to bypass intended application controls under certain conditions.

- The most severe issue listed was CVE-2025-68121, rated Critical with a CVSS score of 10.0, and described as capable of allowing complete compromise of affected systems.
- Several High severity vulnerabilities were also identified, including CVE-2025-25679, CVE-2025-61726, CVE-2026-2391, CVE-2026-25639, and CVE-2026-35554.
- HPE stated these High severity issues could enable remote attackers to disrupt services, impact system integrity, or compromise availability depending on deployment and exposure.
- Additional Medium severity issues were reported under CVE-2025-11143, CVE-2025-13465, CVE-2025-61728, and CVE-2026-40453.

Recommendations

- Upgrade HPE Telco Network Function Virtualization Orchestrator to version 7.7.0 or later as recommended by HPE.

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

back to top

Name	Threat Severity Rating	TLP	Attribution	Originating Source
SolarWinds Fixes Actively Exploited Serv-U DoS Vulnerability	HIGH	CLEAR	Vulnerability	CSC

Executive Summary

SolarWinds has identified CVE-2026-28318, an unauthenticated denial-of-service vulnerability in Serv-U that can be triggered remotely when an attacker sends a specially crafted HTTP POST request containing the Content-Encoding: deflate header, causing the service to crash or become unavailable.

This vulnerability may impact organizations operating internet-facing Serv-U servers, and organizations in the financial sector should be aware that active exploitation in the wild could affect file transfer availability and disrupt externally accessible services if vulnerable instances remain exposed.

Technical Details

- The vulnerability is tracked as CVE-2026-28318 and affects SolarWinds Serv-U.
- It is classified as an uncontrolled resource consumption / denial-of-service issue.
- The flaw can be exploited remotely over the network, making exposed Serv-U systems reachable attack targets.
- No authentication is required, so an attacker does not need valid credentials to trigger the condition.
- No user interaction is required, allowing the attack to be executed directly against the service.
- The exploit method involves sending a specially crafted HTTP POST request that includes the Content-Encoding: deflate header.

- Successful exploitation causes the Serv-U service to crash or become unavailable, resulting in a denial-of-service condition.
- The issue affects Serv-U version 15.5.4 and earlier.
- The vulnerability is reported as actively exploited in the wild, increasing the urgency of remediation for exposed deployments.

Recommendations

- Upgrade SolarWinds Serv-U to version 15.5.4 HF1 immediately.
- Restrict access to trusted IP addresses wherever operationally possible.
- Review and reduce the exposure of internet-facing Serv-U instances.
- Monitor logs for suspicious HTTP POST requests, especially those associated with abnormal or repeated service interruptions.

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

back to top

Name	Threat Severity Rating	TLP	Attribution	Originating Source
Cisco Catalyst SD-WAN Manager Privilege Escalation Vulnerability Exploited	**HIGH**	**CLEAR**	**Vulnerability**	**CSC**

Executive Summary

Cisco has disclosed a high-severity privilege escalation vulnerability in Catalyst SD-WAN Manager that stems from insufficient validation of user-supplied input in CLI functionality, allowing an authenticated attacker with netadmin privileges to upload a crafted file, inject commands, and execute them with root-level permissions.

This vulnerability may impact organizations using affected SD-WAN Manager deployments, and organizations in the financial sector should be aware that observed exploitation has already resulted in unauthorized configuration changes being pushed to edge devices, which could affect service integrity and administrative control.

Technical Details

- The vulnerability is tracked as CVE-2026-20245 and is rated High severity with a CVSS score of 7.8.
- The issue exists because user-supplied input in the command-line interface functionality is not validated sufficiently before being processed.
- An attacker who successfully exploits this weakness can perform command injection and elevate privileges to the root user on the affected system.
- Exploitation requires authenticated access with netadmin privileges, so the attacker must already have sufficient access to the platform.

- Cisco noted that this access may be obtained through valid credentials or by leveraging previously disclosed vulnerabilities such as CVE-2026-20182 or CVE-2026-20127.
- The vulnerability affects Cisco Catalyst SD-WAN Manager deployments regardless of configuration, including on-prem deployments, Cisco SD-WAN Cloud-Pro, Cisco SD-WAN Cloud (Cisco Managed), and Cisco SD-WAN for Government (FedRAMP).
- Cisco has observed limited exploitation activity in the wild.
- In observed cases, successful exploitation resulted in unauthorized configuration changes being pushed to edge devices.
- Cisco has not released a software update specifically addressing CVE-2026-20245 at the time of publication, though future software releases are expected to include remediation.

Recommendations

- Identify and prioritize affected systems for assessment and remediation.
- Preserve relevant logs and forensic artifacts before making changes.
- Review systems for signs of unauthorized activity or configuration changes.

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

back to top

Name	Threat Severity Rating	TLP	Attribution	Originating Source
Unpatched Windows Search URI Flaw Exposes NTLMv2 Hashes via SMB	**MEDIUM**	**CLEAR**	**Vulnerability**	**Open Source**

Executive Summary

Windows contains an unpatched issue in the search: URI handler that can cause a system to initiate an SMB connection to an attacker-controlled server when a user clicks a crafted link, exposing the user's Net-NTLMv2 hash before the operating system displays an error. Huntress reported that the behavior mirrors the mechanism used in CVE-2026-33829, the Windows Snipping Tool URI-handler issue Microsoft patched in April 2026.

This vulnerability may impact organizations that rely on Windows hosts with outbound SMB access, and organizations in the financial sector should be aware that captured NTLMv2 hashes could support relay activity or offline cracking attempts, depending on how the environment is configured. At the time of reporting, Huntress stated the search: variant had no CVE and no vendor fix.

Technical Details

- Huntress reported that the issue resides in the Windows search: URI handler and can be triggered with a crafted link that uses crumb=location: to point to an attacker-controlled UNC path.

- When the link is activated, Windows attempts to connect to the remote SMB location, which can trigger NTLM authentication and leak the user's Net-NTLMv2 hash.
- The research states that the hash is transmitted before Windows renders the access-denied style error message shown to the user.
- Huntress demonstrated that simple link-click delivery worked from a browser, meaning the attack path did not require malware or a complex exploit chain.
- The behavior was described as technically similar to CVE-2026-33829, which affected the Snipping Tool ms-screensketch: URI handler and was patched by Microsoft on April 14, 2026.
- Microsoft's published metadata for CVE-2026-33829 marked the Snipping Tool issue as Moderate severity and noted that customer action was required to resolve it.
- Huntress reported that the newly described search: variant was disclosed to Microsoft but closed as below the servicing bar, leaving the issue without a CVE or patch at the time of publication.
- The Huntress write-up also noted that search: and search-ms: are wired through the same COM activation path, suggesting the issue sits in shared Windows search handling rather than in only one URI scheme.

Recommendations

- Block outbound SMB traffic on TCP/445 and TCP/139 from hosts that do not require it, as Huntress identified this as the highest-value mitigation for this flaw class.
- Enforce SMB signing so captured hashes are less useful for relay attacks against internal services.
- Disable or restrict NTLM where operationally feasible, after validating application compatibility.

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

back to top

Appendix A - Tactics, Techniques & Procedures (TTPs)

Error 524 Decoy: A Large-Scale Smishing Operation Targeting Globally

Tactics	Techniques	Observed Activity
Initial Access	Phishing via SMS	Distribution of smishing messages impersonating trusted brands
Credential Access	Phishing for Information	Collection of payment card and sensitive user data
Defense Evasion	Obfuscated/Decoy Content	Use of fake Cloudflare Error 524 pages as decoys
Defense Evasion	Traffic Filtering	Geofencing and device fingerprinting to restrict access
Exfiltration	Exfiltration Over WebSocket	Encrypted real-time transmission of stolen data
Execution	User Execution	Victim interaction required to proceed through attack flow
Collection	Input Capture	Harvesting of financial and personal information

OP-512 Targets Legacy IIS Servers with Custom Web Shell Framework

Tactics	Techniques	Observed Activity
Persistence	Server Software Component Web Shell	The attacker deployed a custom framework of three web shells to the compromised IIS server, including a file-management component and two authenticated command handlers.
Command and Control	Application Layer Protocol DNS	The first web shell automatically reported its deployment location through DNS queries using encoded data in the subdomain.
Command and Control	Application Layer Protocol Web Protocols	If the DNS notification failed, the web shell used a web-based fallback channel to transmit the same encoded location data.
Defense Evasion	Timestomp	All three web shells manipulated their file times to blend with surrounding files and hinder timeline-based forensic review.
Defense Evasion / Execution	Reflective Code Loading	Four post-exploitation toolkits were loaded directly into the memory of the web server process, with no evidence of those tools being written to disk.
Privilege Escalation	Exploitation for Privilege Escalation	The attacker used multiple privilege-escalation toolkits to attempt elevation from the web server's limited service context to a higher privilege level.
Discovery	System Owner User Discovery	The actor executed commands to determine the current account context and associated privileges on the compromised host.
Defense Evasion	Obfuscated Files or Information	The command handlers used randomized variable and method names, dead variables, junk comments, and encrypted request handling to complicate static analysis and signature creation

PCPJack Hijacks Cloud Servers to Build a Hidden SMTP Relay Network

ID	Technique	Detail
T1190	Exploit Public-Facing Application	Web application exploitation via PCPJack initial access
T1059.004	Unix Shell	Shell payloads issued through Sliver Execute RPC
T1543.002	Systemd Service	xsync.service installed on root-owned compromised hosts
T1053.003	Cron	Five-minute watchdog crontab on unprivileged sessions
T1564.001	Hidden Files	Dot-prefixed binary names in temporary directories
T1036.004	Masquerade Task or Service	xsync service name imitates system synchronization utility
T1572	Protocol Tunneling	HTTP-wrapped Chisel tunnel connecting to port 9000
T1090.003	Multi-hop Proxy	Per-beacon reverse SOCKS5 proxy, ports 10000 to 14999

T1583.003	Virtual Private Server	C2 and Chisel tunnel aggregation hosted on Contabo VPS (AS51167)
T1048	Exfiltration over Alternative Protocol	SCP transfer of proxy list to Excalibur server
T1049	System Network Connections Discovery	ss -tlnp used to enumerate active Chisel tunnel state
T1057	Process Discovery	pgrep used for idempotency checks before each deployment

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
AGENTPSD	A Python-based fallback backdoor used in the VerdantBamboo intrusion set. It appears to have been intended as a backup access method if the primary malware stopped working.
AI coding assistants	Development tools that help write or suggest code and operate inside authenticated developer environments. In this newsletter, they are relevant because stolen developer credentials could affect the environments these tools rely on.
AI toolchain	The broader set of development tools, editors, assistants, repositories, and automation systems used to build software. Several newsletter items showed that attackers are increasingly targeting this layer for credential theft and software compromise.
AMSI	Anti-Malware Scan Interface. A Windows security interface that helps detect malicious scripts and code execution; the malspam-delivered loader attempted to interfere with this visibility.
Apache httpd	Apache HTTP Server, a widely used web server. It was one of the server platforms discussed in relation to the HTTP/2 Bomb memory exhaustion issue.
ASP.NET	Microsoft's web application framework. It appeared in the IIS intrusion reporting where compiled artifacts remained in ASP.NET directories even after malicious files were removed.
Base64	A common encoding method used to hide or transport data in text form. Across the newsletter, it appeared in phishing kits, staged malware payloads, and encoded command content.
BRICKSTORM	A backdoor malware family used by VerdantBamboo. It was deployed on edge and infrastructure systems to maintain covert access and support proxying or remote control.
Browser-based development environment	A development workspace that runs in the browser instead of a locally installed editor. In the newsletter, github.dev was highlighted as a browser-hosted VS Code environment involved in token theft.
Bun	A runtime used to execute JavaScript or TypeScript. In the npm package compromise, malicious code used it to run decrypted payloads during installation.
C2 / Command and Control	The attacker infrastructure used to communicate with malware on compromised systems, send instructions, or receive stolen data.
Calif	The research source that published the HTTP/2 Bomb write-up referenced in the vulnerability entry. Its write-up described how small HTTP/2 inputs could drive very large memory allocation on vulnerable servers.
Chisel	A tunneling tool used in the PCPJack campaign to turn compromised Linux servers into proxy or relay infrastructure.
CI/CD	Continuous Integration / Continuous Delivery. The automated build, test, and release pipeline used in software development. In the newsletter, compromised npm packages were designed to steal secrets from these environments.
Cloudflare	A content delivery and reverse proxy service. In the newsletter, it appeared both as part of phishing deception through fake error pages and as a fronting layer in attacker infrastructure.
Conditional Access	Security policies that restrict access to cloud services based on conditions such as location, device, or identity. VerdantBamboo was assessed to have used trusted infrastructure paths to blend in and evade such controls.
Cookie crumb	A split portion of an HTTP cookie header. The HTTP/2 Bomb write-up explained that repeated cookie fragments could be abused to amplify memory allocation on the server.
Cron	A scheduled task mechanism used on Linux and Unix-like systems. Attackers used it in multiple cases to run malware automatically or maintain fallback access.
CSC	UAE Cyber Security Counsil
CVE	Common Vulnerabilities and Exposures. A public identifier assigned to a specific software vulnerability.

CVSS	Common Vulnerability Scoring System. A widely used method for describing how severe a vulnerability is likely to be.
Deflate	A compression method referenced in the SolarWinds Serv-U denial-of-service issue. A crafted request using this header caused the service to crash or become unavailable.
Developer workstation	A laptop or desktop used by software developers. In the newsletter, several attacks targeted developer machines because they often contain tokens, keys, and access to source code or automation systems.
DLL	Dynamic Link Library. A Windows binary component used by applications. In the newsletter, malicious DLLs appeared as artifacts of web shell execution and staged malware activity.
DNS	Domain Name System. The internet's naming system that converts domain names into addresses. Attackers in the newsletter also used DNS-related behaviour to hide or support malicious activity.
DNS over HTTPS (DoH)	A method of sending DNS traffic over encrypted HTTPS connections. VerdantBamboo was observed using this behaviour, which can make some name resolution activity less visible in standard monitoring.
DoS	Denial of Service. An attack that causes a service to slow down, crash, or become unavailable.
Dynamic table	In HTTP/2 compression, a temporary memory structure used to store headers for reuse. The HTTP/2 Bomb technique abused this feature to turn small requests into large memory allocations.
Edge appliance	An infrastructure system placed at the boundary of a network, such as a firewall, storage sync appliance, or network device. Several attacks in the newsletter used such systems as stealthy footholds because they often have weaker monitoring.
EDR	Endpoint Detection and Response. Security tooling used to monitor and detect suspicious activity on endpoints and servers. VerdantBamboo specifically benefited from targeting devices that lacked this coverage.
Egnyte Storage Sync	A storage synchronization appliance referenced in the VerdantBamboo investigation. It was one of the systems used to maintain covert access into the victim environment.
Envoy	A proxy and traffic management platform. In the newsletter, it was the product affected by a high-severity HTTP/2 memory exhaustion vulnerability.
Error 524 Decoy	The campaign name used for the global smishing operation that hid mobile phishing behind fake Cloudflare-style error pages.
Exfiltration	The unauthorized transfer of data out of a system, such as tokens, secrets, payment information, or confidential business data.
Flow-control window	An HTTP/2 feature that controls how much data can be sent at one time. In the HTTP/2 Bomb attack, a zero-byte flow-control window was used to hold server memory allocations in place for longer.
Geofencing	A targeting technique that restricts content based on location or device profile. The smishing campaign used it so phishing pages were only shown to selected victims.
GitHub Actions	GitHub's workflow automation platform used for build and deployment tasks. The npm package compromise specifically targeted secrets stored in these environments.
GitHub token / OAuth token	A credential that gives software or users access to GitHub resources. In the VS Code issue, the stolen token could reach all repositories available to the affected user session.
github.dev	GitHub's browser-hosted Visual Studio Code editor. It was the primary environment affected by the one-click token theft issue described in the newsletter.
HPACK	The header compression mechanism used by HTTP/2. The HTTP/2 Bomb technique abused HPACK references so very small network input caused much larger memory use on the target server.
HPE Telco NFV Orchestrator	HPE Telco Network Function Virtualization Orchestrator. A telecom orchestration product covered in the vulnerability section because multiple flaws could affect availability, integrity, and remote access security.

HTTP POST	A type of web request used to send data to a server. In the SolarWinds Serv-U issue, crafted HTTP POST requests triggered the crash condition.
HTTP/2	A modern web protocol designed to improve speed and efficiency. Several vulnerabilities discussed in the newsletter showed how protocol features could still create new denial-of-service risks.
HTTP/2 Bomb	The name used for the denial-of-service technique that combined HPACK compression abuse with HTTP/2 flow-control stalling to force excessive memory retention.
IIS	Internet Information Services. Microsoft's web server platform, targeted by the OP-512 espionage activity.
In-memory execution	A technique where malware runs mainly in memory instead of placing obvious files on disk. This reduces visible evidence and may complicate detection or investigation.
JScript	A Microsoft scripting language variant. In the malspam campaign, it acted as one stage in the infection chain before PowerShell and .NET components were launched.
Jupyter notebook	An interactive notebook file format often used for code, text, and visual content. In the VS Code browser issue, a notebook was used to execute JavaScript inside a webview.
Local workspace extension	A VS Code extension stored inside a project repository or workspace. In the one-click exploitation chain, this behavior was used to bypass normal extension trust expectations.
M365	Microsoft 365. Cloud services referenced in the VerdantBamboo intrusion, where attackers used trusted access paths to reach organizational accounts.
Managed access path	A route into an organization through trusted infrastructure, outsourced management, or edge systems rather than a direct endpoint compromise. VerdantBamboo used this type of path effectively.
Managed Services Provider (MSP)	A third-party provider that manages parts of a customer's IT environment. The VerdantBamboo case showed how compromise of an MSP can create a path into client environments.
Memory exhaustion	A condition where a service consumes enough memory that performance degrades or the process stops. It was central to the Envoy HTTP/2 Bomb issue.
Miasma	The campaign name used in reporting on compromised Red Hat Cloud Services npm packages that stole developer and CI/CD secrets during installation.
Mini Shai-Hulud	A label used to describe the npm campaign because it reused tactics associated with a broader style of install-time credential theft and downstream propagation.
NAS	Network Attached Storage. A storage appliance placed on the network for centralized file access. In the VerdantBamboo case, a Synology NAS became a malware deployment point.
Native AOT	Native Ahead-Of-Time compilation. A .NET build method used by PLENET that made analysis more difficult by compiling malware into a standalone native binary.
nginx	A widely used web server and reverse proxy platform. It was one of the products mentioned in the HTTP/2 Bomb write-up.
npm	A JavaScript package ecosystem and package manager. In the newsletter, compromised npm packages were used to deliver malicious code during installation.
OIDC	OpenID Connect. An identity and authentication method referenced in the compromised npm publishing workflow.
OOM	Out Of Memory. A condition where a process or host exhausts available memory and is often terminated. This was explicitly referenced in the Envoy advisory.
OP-512	The tracking name for the suspected espionage cluster that targeted legacy IIS infrastructure with custom web shells and cryptographic access controls.
Persistence	The methods attackers use to stay on a system after initial compromise, such as scheduled tasks, startup entries, services, or hidden tooling.
PfSense	An open-source firewall platform. In the VerdantBamboo reporting, a pfSense firewall was found compromised with a BRICKSTORM variant.
Phishing	A deceptive technique used to trick someone into clicking a link, opening content, or giving away information.

Pingora	A proxy platform mentioned in the HTTP/2 Bomb write-up as another server family affected by similar memory exhaustion behavior.
Pinning memory	Keeping memory allocations from being released quickly. In the HTTP/2 Bomb case, flow-control behavior allowed the attacker to keep server memory occupied for longer than normal.
PLENET	A .NET Core backdoor deployed by VerdantBamboo on a Synology NAS. It represented another custom malware family used to maintain access.
PoC	Proof of Concept. A demonstration that shows a vulnerability or attack path works in practice.
Potato Suite	A group of privilege escalation tools referenced in the OP-512 reporting. They are used to try to move from lower privileges to higher control on Windows systems.
Preinstall hook	A package installation step that runs before installation completes. In the npm compromise, it was used to launch malicious code automatically during npm install.
Process hollowing / RunPE	A technique where malware starts a legitimate process and then replaces or injects malicious code into it so the activity appears to come from trusted software.
Publisher trust	A VS Code safety model that warns users before installing extensions from unfamiliar publishers. The one-click browser exploit was designed to work around that trust flow.
RC4	Rivest Cipher 4. An encryption algorithm used in the custom OP-512 command handlers.
Reflective loading	A way of loading code directly into memory without using standard file-based execution paths. This technique was observed in the OP-512 and malspam-related reporting.
Reverse proxy	An intermediary system that receives traffic from users and forwards it to a server behind it. Attackers in the newsletter also used such layers to hide origin infrastructure.
RSA	Rivest-Shamir-Adleman. A public-key cryptography method used in the OP-512 command handlers to restrict who could issue commands.
Serv-U	A file transfer product from SolarWinds. It was affected by an unauthenticated remotely triggerable denial-of-service vulnerability covered in the newsletter.
Shai-Hulud	A malware and supply chain attack theme referenced in the npm compromise reporting. In the newsletter context, it described a campaign style involving install-time execution, credential theft, and possible propagation.
Single-Page Application (SPA)	A web application that loads content dynamically within one page rather than navigating between many pages. The smishing campaign used this design to present malicious forms only to selected victims.
Sliver	An offensive security framework referenced in the PCPJack campaign. It was used as part of the infrastructure managing compromised Linux hosts.
Slowloris-style hold	A technique where an attacker holds connections or server resources open for a long time to degrade service. The HTTP/2 Bomb write-up described this concept as one half of the combined attack.
Smishing	SMS-based phishing. Attackers use text messages and malicious links to lure victims into clicking and disclosing information or payment data.
SMTP	Simple Mail Transfer Protocol. The standard protocol for sending email. PCPJack used compromised cloud servers as hidden SMTP relay infrastructure.
SMTP relay network	A group of servers used to forward email traffic. In the PCPJack campaign, compromised cloud servers were turned into covert mail relay infrastructure.
SOCKS5	A proxy protocol that allows traffic to be relayed through another host. PCPJack used it to turn compromised systems into working proxy nodes.
Supply chain attack	An attack that reaches victims through trusted software, packages, dependencies, update channels, or publishing workflows instead of only through direct targeting.
Synology NAS	A network-attached storage product line referenced in the VerdantBamboo case, where attackers enabled SSH and deployed malware after regaining access.
Systemd	A Linux service manager. It was used in the PCPJack campaign as one of the persistence methods on compromised cloud systems.

Timestomping	A technique that changes file timestamps so malicious files appear older or blend in with surrounding legitimate files.
TTPs	Tactics, Techniques, and Procedures. A term used to describe how threat actors operate, including their methods for access, persistence, movement, and evasion.
Tunnel / tunneling	A method for routing traffic through another system. In the PCPJack campaign, tunneling was used to transform cloud servers into hidden proxy and relay infrastructure.
Unauthorized remote access	Access gained to a system without permission, often by using stolen credentials, flaws, or weak controls. This risk appeared across several vulnerabilities and intrusions in the newsletter.
VerdantBamboo	The threat actor name used by Volexity and associated with long-term access through appliances, firewalls, VPN paths, and custom malware.
Virtualized network management platform	A software layer used to manage virtualized network functions and orchestration. The HPE Telco NFV Orchestrator vulnerability coverage related to this type of platform.
VPN	Virtual Private Network. A method of remote access into internal systems. In the VerdantBamboo case, SSL VPN access paths were used to blend into trusted traffic and reach internal services.
VS Code	Visual Studio Code, Microsoft's code editor platform. Multiple newsletter entries touched on the broader risk around development ecosystems built on or connected to it.
VS Code webview	A sandboxed embedded browser frame inside Visual Studio Code. It was central to the one-click GitHub token theft issue because it could pass synthetic key events back to the main editor.
Web shell	A malicious file placed on a web server to give attackers remote control or command execution through that server.
WebSocket / WSS	A protocol for persistent two-way communication between browser and server. The smishing campaign used encrypted WebSocket channels to send harvested information in real time.
Window stall	In the HTTP/2 Bomb context, keeping the server's response blocked by advertising a very small or zero flow-control window so allocated memory stays in use longer.
Workflow secrets	Sensitive values stored for automated build or deployment processes, such as tokens, cloud credentials, or keys. These were a major target in the npm package compromise.
Zero-byte flow-control window	An attacker-controlled HTTP/2 setting that prevents the server from completing its response and helps keep memory allocated longer during exploitation.
Zero-day	A vulnerability disclosed or exploited before defenders have had the normal time to fully respond. The browser-based VS Code token theft issue was presented in that context.