TLP:CLEAR

ADGM THREAT INTELLIGENCE NEWSLETTER

PREPARED BY ADGM CTI

CATEGORY	ACTIONABLE
AUDIENCE	ADGM FSRA ENTITIES
DATE	14/5/2026
OVERALL THREAT SCORE	ELEVATED
TARGET SECTOR	FINANCIAL SERVICES
TARGET REGION	MENA & GLOBAL
ATTRIBUTION	MULTIPLE
TLP	CLEAR

WEEKLY SUMMARY REPORT – 14 May 2026

8	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 Middle East impacting Finance Sector

Summary

This week's cybersecurity newsletter covers credential-focused attacks and rising vulnerability risks across cloud, and AI services. It highlights social engineering leading to token compromise, malvertising-based malware, supply chain abuse, Linux persistence tools and data theft frameworks. It also details urgent fixes for key vulnerabilities, including an exploited PAN-OS portal zero-day and an unauthenticated memory leak in an LLM hosting service. From a financial sector perspective, these activities raise risks of account takeover, credential theft, and edge-device compromise, potentially impacting access control and sensitive data. Organizations should prioritize patching internet-facing services, strengthen authentication and portal security, rotate exposed credentials, enhance monitoring , unusual logins and suspicious admin actions to reduce risk.

ADGM THREAT INTELLIGENCE SUMMARY

Multi-Stage "Code of Conduct" Phishing Chain Enables AiTM Token Theft [Campaign] [High]

Operation Exploiting Misconfigured Servers impacting the region [Campaign] [High]

MuddyWater False Flag Activity Masquerading as Chaos Ransomware [Campaign] [High]

OceanLotus Linked Supply-Chain Intrusion Targeting Windows and Linux Devices Via ZiChatBot Malware [Campaign] [High]

PCPJack Worm Targets Exposed Cloud Services to Steal Credentials at Scale [Campaign] [Medium]

Multi-Stage AutoIt Loader Leading to Vidar Infostealer Campaign [Campaign] [Medium]

Fake Claude Lure Delivers DonutLoader and Beagle Backdoor [Campaign] [Medium]

Quasar Linux a Supply Chain Linux RAT with Rootkit and PAM Backdoor [Campaign] [Medium]

Active Exploitation of Palo Alto PAN-OS Zero-Day Critical Flaw [Vulnerability] [High]

Ivanti EPMM Vulnerabilities Enable Authentication Bypass and Remote Code Execution [Vulnerability] [High]

cPanel & WHM Vulnerabilities Enable Code Execution and Privilege Escalation [Vulnerability] [High]

Qualcomm Vulnerabilities Enable Remote Code Execution and Firmware Compromise Across Devices [Vulnerability] [High]

Multiple Cisco Vulnerabilities Enable Code Execution, Authentication Bypass, and DoS Attacks [Vulnerability] [High]

Critical Unauthenticated Memory Leak Flaw in Ollama [Vulnerability] [Medium]

Name	Threat Severity Rating	TLP	Attribution	Originating Source
Multi-Stage "Code of Conduct" Phishing Chain Enables AiTM Token Theft	HIGH	CLEAR	Campaign	Open Source

Executive Summary

Researchers have identified a multi-stage social engineering campaign using "code of conduct" themed phishing emails, PDF lures, and CAPTCHA-gated landing pages that guide victims through a staged workflow, culminating in credential harvesting via adversary-in-the-middle (AiTM) phishing infrastructure.

This campaign may impact organizations in the financial sector as it could affect identity security by enabling session hijacking and MFA bypass through token interception, allowing unauthorized access to sensitive systems.

Technical Details

- The campaign used internal compliance/regulatory-themed branding, with polished enterprise-style templates and authenticity cues to increase plausibility and pressure recipients with time-bound prompts.
- Attackers distributed messages in distinct waves over a short window, targeting tens of thousands of users across thousands of organizations and multiple countries.
- Emails instructed targets to open a "personalized attachment," relying on a PDF that contained the call-to-action link initiating the credential-theft workflow.
- The sending approach leveraged a legitimate email delivery service to send messages that appeared fully authenticated and legitimate from attacker-controlled domains, improving delivery and trust.
- Initial landing involved a CAPTCHA gate presented as session validation, likely intended to deter automated analysis and sandbox detonation.
- After the first gate, an intermediate staging page claimed the documentation was encrypted and required authentication, priming users for the next step.
- Users were prompted for an email address and faced additional human-verification (a second CAPTCHA) to further filter automation and reinforce "legitimacy."
- The final stage varied by device type (mobile vs. desktop), then pushed users toward a sign-in option that initiated an AiTM session-hijack flow.
- The AiTM technique proxied authentication in real time to capture tokens, enabling access without relying solely on harvested passwords.

Recommendations

- Enable link and attachment protections and tune mail security settings to better quarantine and retroactively remove phishing messages.
- Conduct targeted phishing awareness training and simulations that mirror compliance-themed lures and multi-step "verification" workflows.
- Apply conditional access policies to restrict token usage and detect abnormal authentication behavior.

- Use browser and network protections that block known malicious sites and reduce exposure to credential-harvesting pages.
- Monitor for suspicious token activity and anomalous sign-ins and use automated disruption/response capabilities to contain active phishing-led intrusions.

For a detailed breakdown of tactics, techniques, and procedures (TTPs), refer to the Annexure – Multi-Stage "Code of Conduct" Phishing Chain Enables AiTM Token Theft

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
Operation Exploiting Misconfigured Servers impacting the region	HIGH	CLEAR	Campaign	Open Source

Executive Summary

Researchers have identified a campaign targeting government entities, uncovered through an exposed staging server that revealed attacker tooling, command-and-control infrastructure, and stolen data, with operations involving webshell deployment, credential access, and data exfiltration.

This campaign may impact organizations in the financial sector as it could affect regional entities through similar exploitation of exposed infrastructure, credential harvesting, and persistent access techniques used to target sensitive data environments.

Technical Details

- The operation was uncovered because a misconfigured server exposed directories containing C2 code, payloads, logs, and "loot," offering visibility into the actor's end-to-end workflow.
- Recovered materials showed a custom ASP.NET-style webshell used for command execution, returning output directly to the operator and enabling interactive control.
- Multiple scripts referenced a second webshell path consistently across targeting scripts, suggesting a standardized execution mechanism within the compromised environment.
- Tooling referenced potential initial access via a DotNetNuke SSRF vulnerability (versions prior to a fixed release), though the article notes this path could not be confirmed from available data alone.
- Separate scripts and logs reflected attempts against email infrastructure using ProxyShell-related exploitation, alongside credential-based access attempts against public-facing portals.
- Post-compromise activity included systematic enumeration and extraction of user tables and judicial/committee-related datasets, plus collection of sensitive registry hives.
- The actor maintained a Python-based HTTP C2 paired with a PowerShell beacon that polled for tasks, returned results in encoded chunks and logged inbound traffic with timestamps.

- Privilege escalation automation was present, including a Windows token abuse technique executed via database command execution and later refined toward in-memory execution.
- Logs captured attempted persistence via a masqueraded scheduled task and activity consistent with efforts to weaken host protection, with at least one persistence attempt blocked.

Recommendations

- Regularly audit and secure internet-facing infrastructure to prevent exposure of directories and sensitive assets.
- Enforce strong authentication controls and monitor for brute-force or abnormal login attempts.
- Detect and block webshell activity through endpoint and server monitoring tools.
- Patch and harden systems vulnerable to common exploitation techniques, including web and email servers.
- Monitor outbound traffic and system logs for signs of data exfiltration and unauthorized access.

For a detailed breakdown of tactics, techniques, and procedures (TTPs), refer to the Annexure – Operation Exploiting Misconfigured Servers impacting the region

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
MuddyWater False Flag Activity Masquerading as Chaos Ransomware	HIGH	CLEAR	Campaign	Open Source

Executive Summary

Researchers have identified a targeted intrusion that initially appeared to be a Chaos ransomware incident, but forensic analysis assessed it as a "false flag" operation consistent with state-sponsored tradecraft. The activity combined high-touch social engineering with remote interaction to harvest credentials and manipulate MFA, followed by persistence and data exfiltration rather than typical ransomware encryption workflows.

This campaign could affect organizations in the financial sector because similar remote-interaction social engineering, MFA manipulation, and long-term access techniques may be reused to gain trusted footholds and enable covert data theft. Organizations in the financial sector should be aware that the observed focus on persistence and exfiltration over encryption may complicate detection and response when attackers operate using legitimate accounts and remote management tools.

Technical Details

- The intrusion began with targeted social engineering via Microsoft Teams, where the actor initiated external chats and moved into interactive screen-sharing sessions.
- During live sessions, the actor performed basic discovery and guided users to expose VPN-related information and enter credentials into locally created text files.

- The actor explicitly instructed victims to modify MFA settings to include attacker-controlled devices, enabling account compromise beyond simple password capture.
- Analysis also identified a phishing-style verification page themed around remote assistance, indicating impersonation-based credential harvesting alongside live interaction.
- After credential compromise, the actor authenticated to internal systems using multiple compromised accounts and expanded access through interactive remote sessions.
- Persistence was established using legitimate remote management tooling (including DWAgent and AnyDesk) and continued internal access via RDP.
- The payload chain included a downloader that collected host identifiers, registered to attacker infrastructure, fetched additional components, and then self-deleted.
- A custom RAT masqueraded as a legitimate WebView2-related application and implemented anti-analysis techniques (e.g., string obfuscation and sandbox/VM checks).
- The RAT supported command execution (cmd/PowerShell), file operations, and interactive shell management while polling periodically for tasks and returning results.
- Rapid7 assessed attribution to MuddyWater with moderate confidence based on technical artifacts including a specific code-signing certificate and infrastructure overlaps.

Recommendations

- Restrict or tightly govern external Microsoft Teams chat and screen-sharing workflows, and alert on unsolicited helpdesk-style outreach patterns.
- Strengthen MFA change controls and monitoring, including alerts for new device registrations and unusual MFA reconfiguration driven by user interaction.
- Hunt for suspicious use of legitimate remote management tools and unexpected RDP activity, especially shortly after user screen-sharing events.
- Monitor endpoints for downloader-like behavior (host profiling, staged component retrieval, periodic polling) and enforce execution controls to reduce payload chaining.
- Prioritize detection for data-exfiltration behaviors and long-lived access over encryption-only signals, aligning response playbooks to “ransomware-branded” intrusions that focus on theft and persistence.

For a detailed breakdown of tactics, techniques, and procedures (TTPs), refer to the Annexure – MuddyWater False Flag Activity Masquerading as Chaos Ransomware

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
OceanLotus Linked Supply-Chain Intrusion Targeting Windows and Linux Devices Via ZiChatBot Malware	HIGH	CLEAR	Campaign	Open Source

Executive Summary

Researchers have identified a supply-chain campaign in which malicious Python wheel packages uploaded to PyPI (Python Package Index) covertly delivered a dropper that installs a newly named malware family, ZiChatBot, targeting both Windows and Linux systems. The packages presented legitimate functionality on their PyPI pages, but their underlying purpose was to deploy hidden payloads during installation and import.

This campaign could affect organizations in the financial sector because development and automation environments that rely on third-party Python dependencies may be exposed to covert implant delivery across mixed Windows/Linux estates. Organizations in the financial sector should be aware that the described approach leverages trusted package ecosystems and a public chat service's REST APIs for command execution, which may reduce reliance on traditional dedicated C2 infrastructure.

Technical Details

- Malicious wheel packages began appearing on PyPI in July 2025 and were designed to imitate legitimate libraries to entice downloads.
- The attacker used a "benign looking" package to pull the malicious one as a dependency, helping conceal the payload from casual review.
- On installation, the wheel extracts a platform-specific dropper (Windows DLL or Linux shared object) onto the host.
- Importing the library triggers execution of initializer code that loads the dropper into the Python process and invokes an exported function.
- The dropper deploys ZiChatBot, establishes auto-run persistence, and then removes traces by deleting the dropper and a related script from the library folder.
- The dropper decrypts embedded strings and payload data (AES-CBC: Advanced Encryption Standard - Cipher Block Chaining), then decompresses content to stage ZiChatBot components.
- The Linux branch places an executable under a temporary location and uses scheduled execution via cron for persistence.
- ZiChatBot does not use a dedicated C2 server instead, it uses REST APIs of the public team chat application Zulip as its C2 mechanism.
- ZiChatBot's primary supported control is executing received shellcode and signaling completion via an in-channel response behavior.
- The report assesses a possible OceanLotus link based on tooling similarity (including a quantified similarity score) between the dropper and earlier OceanLotus-linked droppers.

Recommendations

- Review and tighten dependency controls for Python projects, including restricting untrusted packages and strengthening validation for new or unusual wheel installs.

- Monitor developer and automation hosts for Python import triggered DLL/.SO loading patterns that indicate a library acting as a dropper.
- Hunt for persistence mechanisms consistent with the report's behavior, including user-level auto-run and cron-based scheduled execution on Linux.
- Add detections for staged payload deployment workflows that decrypt and decompress embedded data prior to execution.
- Review outbound connectivity to public chat-service REST APIs if not business-required and investigate systems repeatedly attempting such API-based polling behavior.

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 Worm Targets Exposed Cloud Services to Steal Credentials at Scale	MEDIUM	CLEAR	Campaign	Open Source

Executive Summary

Researchers have identified PCPJack a credential theft framework that steals credentials from cloud, container, developer, productivity, and financial services, sends the data to attacker-controlled systems, and tries to spread to other machines. It targets exposed services like Docker, Kubernetes, Redis, MongoDB, RayML, and vulnerable web apps, allowing it to move both across the internet and within compromised environments. Instead of using crypto mining, it focuses on stealing access that can be used for fraud, spam, extortion, or resale.

This campaign could affect organizations in the financial sector because the toolset targets a wide range of cloud, container, developer, productivity, and financial-service credentials, which may enable downstream fraud, spam, extortion, or resale of access rather than cryptocurrency mining.

Technical Details

- PCPJack was discovered via hunting that surfaced a Linux bootstrap script whose first actions evict and delete tooling associated with TeamPCP, indicating an operator intent to "replace" prior infections.
- The initial script prepares the environment, installs a suitable Python runtime, builds a virtual environment, then pulls multiple Python modules that collectively orchestrate scanning, credential parsing, movement, and encryption.
- Persistence is established differently based on privilege, using a service-based mechanism when executed with elevated permissions and scheduled execution checks when not, then launching the orchestrator.
- The orchestrator drives credential theft by collecting secrets from configuration sources (including environment-based secrets), SSH materials, and cloud/container contexts, then parsing findings with a dedicated extractor module.

- Target discovery and external propagation leverage large-scale hostname sources and iterative scanning logic designed for broad coverage while minimizing repeat scanning during runtime.
- PCPJack attempts to spread by targeting exposed services such as container and orchestration platforms, databases, ML cluster services, and vulnerable web applications for both external propagation and internal lateral movement.
- Command and control is handled through a public messaging platform channel model, where infected nodes post stolen data and periodically check for operator commands.
- The lateral movement component enumerates internal assets and abuses orchestration APIs and exposed management surfaces to extract secrets, execute commands in workloads, and attempt host-level access via container escape techniques.
- Exfiltrated data is encrypted in message-sized chunks using modern key exchange and authenticated encryption, but the framework may fall back to plaintext if required crypto dependencies are unavailable.
- SentinelLABS also noted a separate staging toolset that combines credential harvesting with deployment of an additional beaconing payload, broadening post-compromise capability.

Recommendations

- Audit and restrict exposure of container engines, orchestration APIs, and cloud-adjacent databases/services to reduce opportunistic propagation paths described in the framework.
- Monitor for eviction behavior that kills or removes rival tooling artifacts, followed by immediate creation of new persistence (service- or scheduler-based) and execution of Python orchestrator logic.
- Hunt for credential discovery patterns across configuration sources, environment-based secrets, SSH materials, and container/orchestration contexts consistent with the reported collection workflow.
- Detect anomalous internal API enumeration of orchestration resources (namespaces, pods, secrets, config objects) and suspicious container operations indicative of escape attempts.
- Review outbound use of public messaging platforms for command retrieval and data posting in server environments and investigate repeated polling/command-check behaviors aligned with the described C2 model.

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
Multi-Stage AutoIt Loader Leading to Vidar Infostealer Campaign	**MEDIUM**	**CLEAR**	**Campaign**	**Open Source**

Executive Summary

Researchers have identified a legitimate Windows scripting tool ‘AutoIt’ is often misused by attackers to run hidden malicious code, and in this case, a compiled AutoIt script (Replies[.]scr) was used as a loader to

execute an external payload and initiate network communication. The activity was linked to Vidar malware, which is known for stealing credentials, browser data, cryptocurrency wallets, and system information.

This campaign may impact organizations in the financial sector because the reported end payload is designed to harvest credentials, browser data, cryptocurrency wallet information, and system details, which could affect account security and sensitive access across user and administrative contexts.

Technical Details

- The execution chain began with a commonly abused hack tool launching a command shell to progress the next stages, suggesting user-driven execution as the initial trigger.
- A disguised document-type file was renamed into a batch script and executed, demonstrating file-extension masquerading to bypass simplistic file-type controls.
- The script performed environment checks and enumerated running processes, indicating attempts to identify and potentially disrupt security-related tooling.
- Payload staging used a native extraction utility to unpack additional components before launching an AutoIt-compiled loader with an external parameter.
- Analysis describes a builder-style workflow where data was progressively extracted from staged files, filtered, written into the loader component, then executed.
- Execution flow included a synchronization delay mechanism, consistent with staged timing control prior to subsequent activity.
- The extracted payload exhibited anti-analysis behavior, including anti-debug checks and detection of instrumentation callbacks often associated with monitoring tools.
- Network communication was implemented using WinINet APIs, constructing outbound HTTP(S) GET requests that appeared consistent with configuration retrieval or beaconing.
- The malware performed DNS resolution for an additional domain prior to outbound activity, reflecting dynamic infrastructure usage ahead of communication.
- Post-execution cleanup routines deleted dropped/staged artifacts and terminated processes to reduce forensic traces and hinder retrospective analysis.

Recommendations

- Isolate suspected affected endpoints promptly to reduce the risk of continued outbound communication and follow-on payload activity.
- Perform full system reimaging where this chain is confirmed, as the report notes the ability to stage and execute additional malware.
- Reset potentially exposed credentials (including browser, email, VPN, and administrative) and invalidate active sessions to limit reuse of stolen access.
- Enforce MFA broadly across critical services to reduce unauthorized access following credential theft.
- Strengthen monitoring of DNS and outbound traffic for suspicious connections and restrict execution of unauthorized/untrusted tools to prevent recurrence.

For a detailed breakdown of tactics, techniques, and procedures (TTPs), refer to the Annexure – Multi-Stage AutoIt Loader Leading to Vidar Infostealer Campaign

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
Fake Claude Lure Delivers DonutLoader and Beagle Backdoor	MEDIUM	CLEAR	Campaign	Open Source

Executive Summary

Researchers at Sophos have identified a malicious imitation of the Claude AI website used in a malvertising-driven delivery chain that installs an archive-based payload and abuses DLL (Dynamic Link Library) sideloading to launch an in-memory loader. The infection flow decrypts and executes shellcode that Sophos attributes to DonutLoader, which then loads a previously undocumented backdoor dubbed as "Beagle".

This campaign may impact organizations in the financial sector because it leverages brand-driven user deception and signed binary abuse to achieve stealthy execution, which could affect endpoint integrity and credential security if the backdoor is established. Organizations in the financial sector should be aware that the observed toolchain supports remote command execution and file operations, enabling follow-on activity that may lead to data exposure.

Technical Details

- The operation begins with a fake Claude lookalike site promoted via malvertising, pushing users toward a product download. The delivered archive is unusually large and initiates an installer-based execution flow.
- The installer drops multiple components into a startup execution location, setting up persistence-by-startup behavior. A legitimate, signed updater binary is used as the loader host for the next stage.
- The chain relies on DLL sideloading: the signed executable loads a malicious DLL placed beside it. This mirrors patterns Sophos notes are commonly seen in sideloading-centric campaigns.
- The malicious DLL decrypts an accompanying encrypted blob by reversing data and applying XOR with a derived key. It then runs the decrypted shellcode using a thread-creation technique for execution.
- Sophos identifies the decrypted shellcode as Donut/DonutLoader, an in-memory loader framework. This stage is used to avoid dropping a large final payload directly at the outset.
- DonutLoader loads the final payload, assessed as a simple backdoor "Beagle" with basic remote administration commands. Supported actions include command execution, file transfer, and filesystem manipulation.
- The backdoor uses encrypted communications with AES (Advanced Encryption Standard) and a fresh 16 byte random IV (Initialization Vector) per packet. It supports both Transmission Control Protocol (TCP) and User Datagram Protocol (UDP) transport formats with distinct packet layouts.
- Commanding uses structured messages distinguishing victim "beacon" traffic from inbound tasking. Sophos describes message type fields indicating outbound status/heartbeat versus inbound commands.

- Sophos observed additional samples reusing the same XOR key across previous months, with variations in loaders and decoys. Some variants delivered different shellcode families while retaining the sideloading-style staging.

Recommendations

- In user guidance and secure browsing controls, reduce exposure to malvertising by warning against sponsored search results for popular AI tools and software downloads.
- Monitor startup execution locations for newly introduced executables and adjacent DLL/data-file trios indicative of sideloading chains.
- Add behavioral detections for installers that stage encrypted blobs and then spawn a signed host process that loads a DLL from its working directory.
- Validate endpoint controls can detect and block common sideloading abuse patterns and loader behaviors and ensure telemetry retention supports post-event reconstruction.

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
Quasar Linux a Supply Chain Linux RAT with Rootkit and PAM Backdoor	MEDIUM	CLEAR	Campaign	Open Source

Executive Summary

Researchers have identified Quasar Linux (QLNX), a stealthy Linux remote access trojan that uses fileless execution, rootkit capabilities, and a PAM (Pluggable Authentication Module) backdoor to harvest developer and cloud credentials, enabling persistent access and supply chain compromise.

This campaign may impact organizations in the financial sector as QLNX targets developer and DevOps credentials across the software supply chain, which could affect CI/CD access, cloud administration, and trusted package publication workflows used to support business services.

Technical Details

- QLNX is described as a comprehensive Linux implant that combines evasion, persistence, keylogging, and credential harvesting alongside remote access capabilities.
- The 'LD_PRELOAD' is an environment variable in Linux and Unix-like operating systems that allows you to load custom shared libraries before any other libraries when a program starts.
- The malware embeds C source code for its PAM backdoor and LD_PRELOAD rootkit inside the binary and compiles components on the victim host using local tooling.
- It deploys interception mechanisms through the system preload mechanism for broad visibility, and also uses eBPF (Extended Berkeley Packet Filter) based hiding to conceal processes, files, and network artifacts.

- QLNX achieves fileless execution by copying itself into memory and re-executing, then deleting the original on-disk binary to reduce forensic footprint.
- It profiles the host environment like privilege level, kernel features, containerization signals, and available tools to selectively enable capabilities like compilation, injection, and surveillance.
- Credential harvesting targets high-value developer and cloud artifacts (package registry tokens, Git credentials, cloud provider and container configs, CI/CD-related secrets, and environment files).
- The PAM backdoor is designed to intercept authentication credentials in plaintext during login, expanding access beyond stored secrets and config files.
- Surveillance features include keylogging, screenshot capture, and clipboard monitoring, supporting broader collection beyond credentials alone.
- Post-compromise capabilities include tunneling/port forwarding, scanning, SSH-based movement, and a peer-to-peer mesh design intended to improve resilience.

Recommendations

- Monitor Linux systems for unexpected modifications to preload-based interception and for newly introduced shared objects compiled locally as part of runtime deployment behavior.
- Prioritize hardening and monitoring of developer/DevOps endpoints by auditing access to credential-bearing files and tokens highlighted in the report and rotating exposed secrets.
- Alert on suspicious persistence across user and system scopes, including new service definitions, reboot-scheduled tasks, startup entries, and shell-profile modifications.
- Hunt for fileless execution patterns and stealth behaviors described (memory re-execution, process masquerading, log wiping, and hidden artifacts consistent with rootkit activity).
- Investigate unusual tunneling and peer-to-peer style connectivity from Linux hosts, especially when correlated with credential access and system profiling activity.

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
Active Exploitation of Palo Alto PAN-OS Zero-Day Critical Flaw	**HIGH**	**CLEAR**	**Vulnerability**	**CSC**

Executive Summary

Palo Alto Networks has addressed CVE-2026-0300 (CVSS 9.3), a critical zero-day buffer overflow in the PAN-OS User-ID Authentication Portal (Captive Portal) that enables unauthenticated attackers to achieve arbitrary code execution with root privileges via specially crafted packets.

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-2026-0300 is a buffer overflow vulnerability in the User-ID Authentication Portal (Captive Portal) service of PAN-OS.
- Exploitation is unauthenticated and remote, with successful attacks resulting in arbitrary code execution with root privileges.
- The issue applies only to PA-Series and VM-Series firewalls configured to use the User-ID Authentication Portal feature.
- Internet-exposed Captive Portal instances are highlighted as the highest-risk deployments, with limited in-the-wild exploitation confirmed against exposed systems.
- A public proof-of-concept (PoC) exploit was shared by an independent researcher, demonstrating how unauthenticated packets can lead to root-level RCE.
- Unit 42 assessed observed exploitation as likely associated with a state-linked cluster tracked as CL-STA-1132, with activity reported as successful since 29 April 2026.
- Post-exploitation activity described includes shellcode injection, deployment of tunneling utilities (EarthWorm, ReverseSocks5), and Active Directory enumeration using credentials likely obtained from the firewall.
- The threat actor was also observed deliberately erasing logs and other forensic artifacts to impede detection and investigation.

Recommendations

- Restrict User-ID Authentication Portal access to trusted internal IP addresses only.
- Block portal exposure from the public internet or untrusted zones.
- Apply Threat Prevention Signature updates (available for PAN-OS 11.1+).
- Review Device > User Identification > Authentication Portal Settings to confirm exposure status.

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

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

back to top

Name	Threat Severity Rating	TLP	Attribution	Originating Source
Ivanti EPMM Vulnerabilities Enable Authentication Bypass and Remote Code Execution	**HIGH**	**CLEAR**	**Vulnerability**	**CSC**

Executive Summary

Ivanti has released security updates addressing five high-severity vulnerabilities in Endpoint Manager Mobile (EPMM), including authentication bypass, privilege escalation, certificate validation weaknesses, arbitrary method invocation, and remote code execution. Limited active exploitation has been confirmed for CVE-2026-6973, which requires administrative authentication.

These vulnerabilities may impact enterprise mobile device management and certificate trust mechanisms. Organizations in the financial sector should be aware, as these vulnerabilities could affect administrative control, device enrollment integrity, and secure access within mobile environments.

Technical Details

- CVE-2026-5786 is an improper access control vulnerability allowing authenticated low-privilege users to escalate privileges and gain administrative access.
- CVE-2026-5787 enables unauthenticated attackers to impersonate Sentry hosts and obtain valid CA-signed client certificates due to improper certificate validation.
- CVE-2026-5788 allows unauthenticated attackers to invoke arbitrary methods, increasing the risk of unauthorized application interaction.
- CVE-2026-6973 is an input validation flaw enabling authenticated administrators to execute remote code; limited exploitation has been observed.
- CVE-2026-7821 allows unauthenticated attackers to enroll unauthorized devices, potentially leading to information disclosure and compromise of device identity.
- The vulnerabilities impact multiple layers, including authentication, certificate validation, and API access within EPMM.
- Affected versions include Ivanti EPMM 12.8.0.0 and earlier releases across deployments.
- Organizations impacted by earlier vulnerabilities may remain at risk if credential rotation was not completed.

Recommendations

- Immediately apply vendor-provided security updates for Ivanti EPMM.
- Fixed versions include 12.6.1.1, 12.7.0.1, and 12.8.0.1, which also address previously disclosed vulnerabilities.
- Verify and enforce strong administrative authentication controls.
- Rotate administrative credentials, especially if prior compromise is suspected.
- Review logs for unauthorized administrative activity or suspicious API usage.
- Restrict administrative access to trusted networks/IP ranges where possible.

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

back to top

Name	Threat Severity Rating	TLP	Attribution	Originating Source
cPanel & WHM Vulnerabilities Enable Code Execution and Privilege Escalation	**HIGH**	**CLEAR**	**Vulnerability**	**CSC**

Executive Summary

cPanel has released security updates addressing multiple vulnerabilities in cPanel & WHM, including flaws that allow authenticated attackers to execute arbitrary Perl code, perform file reads, and escalate privileges

or trigger denial-of-service conditions. The most severe issues (CVE-2026-29202 and CVE-2026-29203) carry high CVSS scores of 8.8.

These vulnerabilities may impact systems relying on web hosting management platforms. Organizations in the financial sector should be aware, as exploitation could affect system integrity, enable unauthorized access, and disrupt service availability.

Technical Details

- CVE-2026-29202 arises from improper input validation in the "plugin" parameter of the create_user API, enabling arbitrary Perl code execution under an authenticated system user.
- CVE-2026-29203 involves unsafe symlink handling, allowing modification of file permissions using chmod, resulting in denial-of-service or potential privilege escalation.
- CVE-2026-29201 allows arbitrary file reads due to insufficient validation of feature file names in administrative calls.
- CVE-2026-41940, a previously disclosed critical flaw, has been reportedly weaponized in active attacks.
- Threat actors have leveraged the vulnerability to deploy Mirai botnet variants targeting vulnerable systems.
- Ransomware identified as “Sorry” has also been deployed in observed exploitation activities.
- Attack chains may involve authenticated access followed by code execution or privilege escalation to gain control over systems.
- Exposure may lead to unauthorized file access, system manipulation, and service disruption.
- Multiple versions of cPanel & WHM are affected, with patches released across supported branches.

Recommendations

- Immediately update cPanel & WHM to the latest patched versions across all affected systems.

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

back to top

Name	Threat Severity Rating	TLP	Attribution	Originating Source
Qualcomm Vulnerabilities Enable Remote Code Execution and Firmware Compromise Across Devices	**HIGH**	**CLEAR**	**Vulnerability**	**CSC**

Executive Summary

Qualcomm has disclosed multiple critical and high-severity vulnerabilities affecting its software and firmware components, including flaws enabling remote code execution, privilege escalation, and denial-of-service. The most critical vulnerability (CVE-2026-25254) allows unauthenticated attackers to execute arbitrary code without user interaction.

These vulnerabilities may impact a wide range of connected devices leveraging Qualcomm technologies. Organizations in the financial sector should be aware, as exploitation could affect device integrity, firmware security, and operational reliability across enterprise and industrial environments.

Technical Details

- CVE-2026-25254 is a critical improper authorization flaw in the Software Center SocketIO interface, allowing unauthenticated remote attackers to execute arbitrary code.
- CVE-2026-25293 is a buffer overflow in powerline communication firmware, enabling adjacent attackers to deliver malicious payloads.
- CVE-2026-25262 involves write-what-where memory corruption in the Primary Bootloader, allowing secure boot bypass and persistent firmware compromise.
- CVE-2026-25255 exposes dangerous functionality via the gRPC server interface, enabling privilege escalation within the Software Center.
- CVE-2026-24082 is a use-after-free vulnerability in automotive GPU components, potentially causing instability in infotainment or telemetry systems.
- CVE-2025-47408 involves untrusted pointer reference in firmware, allowing memory corruption through crafted driver-level inputs.
- CVE-2025-47401 and CVE-2025-47403 are buffer over-read vulnerabilities in WLAN components, leading to transient denial-of-service conditions.
- Several vulnerabilities require no user interaction or minimal proximity, increasing exploitation feasibility in exposed environments.
- Successful exploitation can lead to full compromise of application environments, firmware persistence, or device instability.
- The vulnerabilities impact a broad ecosystem, including smartphones, automotive platforms, and industrial IoT systems.

Recommendations

- Identify all Qualcomm-based assets within the environment and prioritize patching based on exposure and criticality.

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

back to top

Name	Threat Severity Rating	TLP	Attribution	Originating Source
Multiple Cisco Vulnerabilities Enable Code Execution, Authentication Bypass, and DoS Attacks	**HIGH**	**CLEAR**	**Vulnerability**	**CSC**

Executive Summary

Cisco has released security updates addressing multiple vulnerabilities across several products, including Unity Connection, Identity Services Engine (ISE), and network management platforms. These vulnerabilities allow attackers to execute arbitrary code, bypass authentication, conduct denial-of-service attacks, and access sensitive information.

These updates may impact enterprise networking and communication systems. Organizations in the financial sector should be aware, as exploitation could affect system availability, compromise authentication controls, and expose sensitive operational data.

Technical Details

- Multiple vulnerabilities impact Cisco Unity Connection, enabling remote code execution and server-side request manipulation through crafted requests.
- Weaknesses in Cisco Identity Services Engine allow authentication bypass and stored cross-site scripting, undermining access control mechanisms.
- Cisco SG350/SG350X switches are affected by denial-of-service vulnerabilities triggered via SNMP requests.
- Network orchestration platforms are exposed to connection exhaustion attacks, potentially disrupting service availability.
- Cisco IoT Field Network Director contains multiple vulnerabilities affecting system integrity and management operations.
- Information disclosure vulnerabilities in network management platforms may expose sensitive configuration and operational data.
- Exploitation paths may involve chained techniques combining authentication bypass, data access, and service disruption.
- The vulnerabilities affect a wide range of enterprise and network-facing services across Cisco environments.
- Successful exploitation can lead to operational disruption, unauthorized access, or manipulation of enterprise systems.

Recommendations

- Apply Cisco security updates across all affected systems as soon as possible.

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

back to top

Name	Threat Severity Rating	TLP	Attribution	Originating Source
Critical Unauthenticated Memory Leak Flaw in Ollama	**MEDIUM**	**CLEAR**	**Vulnerability**	**CSC**

Executive Summary

Ollama (AI model runner) has disclosed a critical vulnerability (CVE-2026-7482) affecting its platform that allows unauthenticated attackers to leak entire process memory through a heap out-of-bounds read flaw in the GGUF model loader, a component responsible for parsing and loading AI model files. The issue can be exploited via crafted model files sent to exposed API endpoints without requiring authentication.

This vulnerability may impact organizations using self-hosted AI infrastructure. Organizations in the financial sector should be aware, as exploitation could expose sensitive data such as API keys, system prompts, and internal processing data from AI-driven applications.

Technical Details

- CVE-2026-7482 is a heap out-of-bounds read vulnerability in the GGUF model loader, enabling unauthorized memory access.
- The flaw occurs when attacker supplied GGUF files specify tensor offsets and sizes exceeding actual file boundaries, causing memory over-read.
- Exploitation requires no authentication, as critical API endpoints (/api/create and /api/push) lack built-in access controls.
- Attackers upload malicious model files via API calls to trigger the vulnerability during model creation workflows.
- The exposed memory may include environment variables, API keys, system prompts, and user interaction data from the AI process.
- Exfiltration is achieved by pushing the compromised model artifact containing leaked memory to attacker-controlled registries.
- The attack chain typically involves three API calls upload, create, and push executed without detection or service disruption.
- The vulnerability affects Ollama versions prior to 0.17.1, with patched releases addressing the issue.
- Exposure risk increases for deployments configured to listen on all network interfaces or accessible over the internet.
- Public proof-of-concept code is available, increasing the likelihood of exploitation attempts.

Recommendations

- Upgrade Ollama to version 0.17.1 or later to remediate the vulnerability.

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

back to top

Appendix A - Tactics, Techniques & Procedures (TTPs)

Multi-Stage "Code of Conduct" Phishing Chain Enables AiTM Token Theft

Tactics	Techniques	Description
Initial access	Phishing	Phishing emails
Persistence	Valid Account	Threat actors sign in with stolen valid entities

Operation Exploiting Misconfigured Servers impacting the region

Tactics	Techniques	Description
Reconnaissance	Active Scanning (T1595)	Early-stage activity reflected reconnaissance and probing of multiple government-facing services before structured post-compromise operations.
Initial Access	Exploit Public-Facing Application (T1190)	Exploitation attempts were documented against exposed server-side components and web platforms to gain access.
Initial Access	Brute Force (T1110)	Credential-based attempts against public portals were observed as part of initial access activity.
Persistence	Web Shell (T1505.003)	Persistent access in the justice ministry environment was maintained via a webshell path used to execute commands remotely.
Execution	Command and Scripting Interpreter Windows Command Shell (T1059.003)	Commands were executed on compromised hosts through a web-accessible mechanism invoking Windows shell execution.
Discovery	System Information Discovery (T1082)	Default and routine commands returned identity and host information to the operator during sessions.
Discovery	System Network Configuration Discovery (T1016)	Network configuration details were collected via routine command output during operator interaction.
Credential Access	OS Credential Dumping SAM (T1003.002)	Sensitive Windows registry hives associated with credential material were extracted from the environment.
Credential Access	Brute Force Password Cracking (T1110.002)	Application membership credential material was collected and prepared for offline cracking attempts.
Privilege Escalation	Exploitation for Privilege Escalation (T1068)	Privilege escalation was performed using a token-impersonation style approach executed via database command functionality.
Command and Control	Protocol Tunneling (T1572)	Tunneling tooling was present to establish encrypted tunnels through network controls.
Collection	Data from Local System (T1005)	Judicial case data, committee decisions, and database content were gathered from internal systems for staging.
Exfiltration	Exfiltration Over Alternative Protocol SSH (T1048.003)	Operator notes described retrieval workflows consistent with secure remote copy-style collection from staged "loot."

MuddyWater False Flag Activity Masquerading as Chaos Ransomware

Tactics	Techniques	Description
Initial Access	T1566 – Phishing (Spearphishing via Service)	Attackers gain initial access by sending malicious Microsoft Teams messages to trick users into executing actions or sharing credentials.
Execution	T1059 – Command and Scripting Interpreter	Execution of system commands such as ipconfig, whoami, etc., to interact with the compromised system.

Discovery	T1082 – System Information Discovery	Collection of host-level system details (OS, hostname, hardware info) from infected machines.
Discovery	T1016 – System Network Configuration Discovery	Attackers identify network configurations using commands like ipconfig to understand connectivity and interfaces.
Credential Access / Persistence / Privilege Escalation	T1078 – Valid Accounts	Use of stolen or harvested credentials to authenticate and maintain access within the environment.
Credential Access	T1056 – Input Capture	Capturing user credentials when victims enter them into attacker-controlled files or phishing pages.
Persistence / Defense Evasion	T1556 – Modify Authentication Process	Manipulation of MFA settings, such as adding attacker-controlled devices to bypass authentication controls.
Lateral Movement	T1021.001 – Remote Services: RDP	Use of Remote Desktop Protocol (RDP) to move laterally and access internal systems.
Command and Control / Persistence	T1219 – Remote Access Tools	Use of legitimate tools like DWAgent and AnyDesk for ongoing remote control and persistence.
Persistence / Privilege Escalation	T1543 – Create or Modify System Process	Installation of malicious services (e.g., DWAgent) to ensure persistent execution.
Defense Evasion / Execution	T1055 – Process Injection / Proxy Execution	Abuse of legitimate processes (e.g., renamed Python binary) to execute malicious code stealthily.
Command and Control / Execution	T1105 – Ingress Tool Transfer	Downloading malicious payloads (e.g., ms_upd.exe) using tools such as curl.
Exfiltration	T1041 – Exfiltration Over C2 Channel	Data is exfiltrated through established command-and-control communication channels.
Defense Evasion	T1027 – Obfuscated/Encrypted Files or Information	Use of encrypted configuration files (e.g., visualwincomp.txt) to evade detection and analysis.
Defense Evasion	T1497 – Virtualization/Sandbox Evasion	Malware performs checks to detect virtualized or sandboxed environments and avoids execution if detected.
Defense Evasion	T1622 – Debugger Evasion	Techniques employed to evade debugging tools used by analysts during investigation.
Command and Control	T1071 – Application Layer Protocol	Use of standard web protocols (HTTP/HTTPS) for communication with C2 servers.
Command and Control	T1573 – Encrypted Channel	Encryption of C2 communications to prevent detection and interception.
Initial Access / Persistence	T1133 – External Remote Services	Use of compromised VPN accounts to gain access to internal networks from external locations.
Discovery	T1087 – Account Discovery	Enumeration of user accounts on systems using built-in commands.
Discovery	T1018 – Remote System Discovery	Identification of other systems in the network for potential lateral movement

Multi-Stage AutoIt Loader Leading to Vidar Infostealer Campaign

Tactics	Techniques	Description
TA0001 - Initial Access	T1204.002 – User Execution: Malicious File	User downloaded and executed microsofttoolkit.exe (hacktool), serving as the initial entry point into the system
TA0002 - Execution	T1204.002 – User Execution: Malicious File	User executed microsofttoolkit.exe, initiating the infection under the assumption of legitimate software activation
TA0002 - Execution	T1059.003 – Command and Scripting Interpreter: Windows Command Shell	Batch/script-based execution used to stage further activity

TA0002 - Execution	T1027 – Obfuscated/Compressed Files	Payload staged via compressed or embedded format using extract32.exe
TA0002 - Execution	T1140 – Deobfuscate/Decode Files or Information	Extraction process used to unpack the next-stage payload
TA0002 - Execution	T1059 – Command and Scripting Interpreter	AutoIt-based loader executed via script-like behavior
TA0002 - Execution	T1218 – Signed Binary Proxy Execution	.scr (AutoIt compiled binary) used as a loader to execute malicious logic
TA0005 - Defense Evasion	T1036 – Masquerading	A .dot file was renamed to .bat to bypass basic file-type restrictions
TA0005 - Defense Evasion	T1562.001 – Disable or Modify Security Tools	taskkill.exe used to terminate security-related processes
TA0005 - Defense Evasion	T1059.003 – Command Shell	findstr.exe leveraged for filtering and identifying security processes
TA0005 - Defense Evasion	T1070.004 – Indicator Removal on Host: File Deletion	Malware deleted dropped files to remove evidence
TA0005 - Defense Evasion	T1489 – Service Stop	Processes terminated to reduce forensic artifacts and evade detection
TA0011 - Command and Control	T1071.001 – Application Layer Protocol: Web Protocols	Malware communicated with C2 over HTTP/HTTPS
TA0011 - Command and Control	T1573 – Encrypted Channel	Encrypted communication used to evade detection
TA0010 - Exfiltration	T1041 – Exfiltration Over C2 Channel	Stolen data (credentials, browser data) exfiltrated via C2 channel

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
Active Directory (AD)	Microsoft directory service referenced as a post-exploitation target for enumeration using credentials obtained from a compromised firewall.
AES	Encryption algorithm referenced in malware components for protecting configuration or communications.
AiTM	Adversary-in-the-middle: A phishing technique that proxies real-time authentication to capture session tokens and enable immediate access.
API endpoint	Application interface path used to perform actions such as create/push models; abused to trigger and exfiltrate memory disclosure.
Authentication bypass	Gaining access without valid authentication; referenced as a potential outcome in some product vulnerability updates.
Authentication token	A session artifact captured during authentication that can allow access without relying only on a password.
AutoIt	A legitimate Windows scripting language abused to create compiled loaders that execute malicious payloads.
Backdoor	Malware that provides remote control capabilities such as command execution and file operations after compromise.
Buffer overflow	A memory corruption flaw type enabling code execution by overwriting memory boundaries (e.g., PAN-OS CVE-2026-0300).
C2	Command and Control: Infrastructure and protocols attackers use to send commands and receive results from compromised systems.
CAPTCHA	A human verification step used in phishing chains as gating to deter automation and sandbox analysis.
Chisel	A tunneling tool referenced as present in recovered payloads to establish encrypted tunnels through firewalls.
chmod	Unix permission-change operation referenced as part of a symlink-handling vulnerability outcome.
CI/CD	Continuous Integration/Continuous Delivery: Development pipeline context referenced as a potential pivot path when DevOps credentials are harvested.
Clipboard monitoring	Collection of clipboard content, referenced as part of data harvesting/surveillance capability set.
Cloud worm	A propagating framework that scans and spreads across exposed cloud services to harvest and exfiltrate credentials.
Credential harvesting	Collection of secrets from configs, environment variables, keys, tokens, and other sources to enable further access.
Cron	Linux scheduling mechanism used for persistence (e.g., ZiChatBot Linux persistence via crontab).
CSC	UAE Cyber Security Counsil
CVE	Common Vulnerabilities and Exposures identifier used to track publicly reported vulnerabilities (e.g., CVE-2026-7482).
CVSS	Common Vulnerability Scoring System: Severity score noted for vulnerabilities (e.g., CVSS 9.1, 9.3).
Data exfiltration	Unauthorized extraction of data from a victim environment, emphasized in incidents that focused on theft over encryption.

DLL sideloading	A technique where a legitimate signed executable loads a malicious DLL placed in its directory, enabling stealthy code execution.
Docker socket	A privileged management interface abused to execute actions against containers and enable spread/escape patterns.
DonutLoader / Donut	An in-memory loader shellcode used to execute payloads without a traditional on-disk malware footprint.
DoS	Denial of Service: Disrupting availability of a service or system (e.g., connection exhaustion, SNMP-related DoS).
DotNetNuke (DNN)	A web platform referenced in the Oman intrusion where scripts and hardcoded paths aligned to DNN structures.
eBPF	Linux feature referenced as used for kernel-level hiding of PIDs, filenames, and ports.
Environment variables	Host process configuration values that may be present in memory and exposed via the vulnerability.
Exposed services	Internet- or network-reachable services attackers scan to gain access (e.g., container/orchestration/datastore services).
False flag	An operation designed to appear like a different threat type or group, complicating attribution (e.g., ransomware-branded intrusion assessed as state-sponsored).
GGUF	Model file format discussed as the attack vehicle for manipulating tensor metadata to trigger heap reads.
GPU	Graphics Processing Unit: Component referenced in an automotive vulnerability affecting system stability.
gRPC	Remote procedure call interface referenced as an exposed surface enabling privileged function access in some ecosystems.
IDOR	Insecure Direct Object Reference: Access control weakness enabling access to objects/resources not intended for the requester.
IMDS	Instance Metadata Service: Referenced as a credential source in cloud environments (noted in lateral movement logic).
IoT	Internet of Things: Device category referenced as impacted by both product updates and chipset vulnerabilities.
Keylogger	Surveillance capability capturing keystrokes, referenced as part of Linux implant capability set.
Kubernetes	Container orchestration platform targeted for secret access and lateral movement in cloud credential theft tooling.
LD_PRELOAD	Linux mechanism used to load attacker-controlled shared objects for system-wide interception/hiding.
Loader / Dropper	A staged component that prepares, decrypts, or reconstructs the next payload (e.g., AutoIt loader chain).
Log cleanup / artifact removal	Deliberate deletion of logs, crash records, or dropped files to impede investigation and detection.
LZMA	Compression method used to unpack decrypted payload data in the ZiChatBot dropper flow.
Malvertising	Malicious advertising / search-result abuse used to drive users to fake download sites that distribute malware.
Masquerading	Disguising file type/extension to bypass controls (e.g., renaming a ".dot" file to ".bat").
memfd_create / execveat	Linux syscalls referenced for fileless in-memory execution and re-execution behavior.
Memory leak	Exposure of process memory contents to an attacker; central to "Bleeding Llama" in Ollama.
MFA	Multi-Factor Authentication: An authentication method requiring more than one verification factor; AiTM can bypass non-phishing-resistant MFA by capturing tokens.
MongoDB	Database service referenced as targeted for credential harvesting in cloud propagation tooling.

Out-of-bounds read	Reading beyond a valid buffer boundary, leading to unintended memory disclosure (the core bug mechanism described).
PAM	Pluggable Authentication Module: Used as a backdoor mechanism to intercept plaintext credentials during authentication.
PAN-OS	Firewall operating system referenced as affected by a critical captive portal buffer overflow.
Persistence	Techniques used to maintain access (e.g., remote management tools, scheduled tasks, system services, cron).
Phishing	Social engineering that tricks users into interacting with attacker-controlled content to harvest access or data (e.g., compliance-themed lures).
PoC	Proof of Concept: Public research code demonstrating exploitability (e.g., CVE-2026-0300 PoC).
PowerShell beacon	A script-based implant that polls for tasks and returns results, used in a Python HTTP C2 framework.
Privilege escalation	Gaining higher permissions than intended (e.g., via unsafe permission changes or admin access control issues).
Prompt / System prompt	User inputs and system instructions for LLM behavior, cited as content that may appear in leaked process memory.
ProxyShell	A set of Exchange exploitation techniques referenced in tooling and targeting attempts in the Oman intrusion analysis.
PyPI	Python Package Index: Public repository where malicious wheel packages were uploaded to distribute ZiChatBot.
Quantization (F16/F32)	Conversion of tensor data types; manipulation of conversion choices was used to keep leaked heap data readable.
RaaS	Ransomware-as-a-Service: A model where ransomware operators provide tooling/infra for affiliates, sometimes used as a “false flag” cover.
RAT	Remote Access Trojan: Malware providing remote control over an infected host (noted in multiple Linux and Windows campaigns).
RCE	Remote Code Execution: Ability to run attacker-controlled code on a target system remotely (often high impact).
RDP	Remote Desktop Protocol: Used for interactive remote access and lateral activity after credential compromise.
Redis	Datastore referenced as targeted for secret discovery and persistence via cron rewrite behavior.
Registry Run key	Windows auto-run mechanism used to persist a payload via user startup execution.
Remote management tools	Legitimate admin tools abused for persistence and control (e.g., DWAgent/AnyDesk referenced in a false-flag intrusion).
Root privileges	Highest-level system privileges; referenced as the execution level achievable in PAN-OS exploitation.
Rootkit	A stealth component designed to hide processes/files/ports and intercept system calls, referenced in a Linux RAT.
SNMP	Simple Network Management Protocol: Management protocol referenced in a switch-related denial-of-service vulnerability.
Social engineering	Manipulation of users through convincing narratives and interaction (e.g., Teams-based screen-sharing) to obtain credentials or MFA changes.
SocketIO	Interface referenced as a component surface in a critical RCE scenario in chipset bulletin items.
SSRF	Server-Side Request Forgery: A flaw type referenced as a likely initial access path in DotNetNuke tooling (not confirmed in the source).
Supply chain attack	Abuse of trusted ecosystems (e.g., PyPI packages or developer credentials) to distribute malicious code at scale.
Symlink	Symbolic link mechanism abused in unsafe handling scenarios that can impact file operations and permissions.

Telegram C2	Use of Telegram channels/bots for command retrieval and data exfiltration in a cloud credential framework.
TLS	Encrypted communication channel referenced for malware network protocols and data transfer.
Tunneling utilities	Public tools used post-exploitation to create covert channels (e.g., EarthWorm and ReverseSocks5 in PAN-OS exploitation).
User-ID Authentication Portal / Captive Portal	PAN-OS service where CVE-2026-0300 resides; exposure is highest when reachable from untrusted networks.
Vidar	An information-stealing malware referenced as the final payload in the AutoIt multi-stage chain.
Webshell	A server-side script enabling remote command execution via HTTP requests, used for persistent access in the Oman intrusion.
Wheel package	Python distribution format abused to bundle droppers targeting Windows and Linux.
WinINet	Windows networking API used by malware to construct HTTP(S) requests for beaconing/config retrieval.
WLAN	Wireless Local Area Network: Wireless subsystem referenced in chipset vulnerabilities causing transient service disruption.
XOR	Simple encoding/encryption method used in multiple stages (e.g., payload decode routines).
XSS	Cross-Site Scripting: Web vulnerability where attacker-controlled script is injected/executed in a user's browser session.
ZiChatBot	Malware family delivered via PyPI packages; uses public Zulip REST APIs rather than a traditional dedicated C2 server.
Zulip REST APIs	Public chat application APIs used by ZiChatBot for command retrieval and execution signaling.