TLP:CLEAR

ADGM THREAT INTELLIGENCE NEWSLETTER

PREPARED BY ADGM CTI

CATEGORY	ACTIONABLE
AUDIENCE	ADGM FSRA ENTITIES
DATE	4/6/2026
OVERALL THREAT SCORE	ELEVATED
TARGET SECTOR	FINANCIAL SERVICES
TARGET REGION	MENA & GLOBAL
ATTRIBUTION	MULTIPLE
TLP	CLEAR

WEEKLY SUMMARY REPORT – 4 June 2026

13	5	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 cybersecurity newsletter highlights a broad mix of identity-driven intrusions, mobile banking malware, destructive and exfiltration-focused attacks, and the continued abuse of trusted tools, cloud services, and software supply chains to gain access, persist, and steal data. Across the reported activity, attackers repeatedly used social engineering, phishing, vishing, poisoned search results, trojanized software, and compromised management workflows to move from initial access into credential theft, remote control, destructive actions, and wider cloud or endpoint compromise. These threats are particularly relevant to financial institutions, as they may affect cloud identities, collaboration platforms, endpoint management systems, public-facing web infrastructure, developer environments, and Windows endpoints. The newsletter also highlights several high-severity vulnerabilities, underscoring the need to accelerate patching, tighten privileged access controls, verify file and agent integrity, monitor unusual administrative activity, and strengthen defenses around authentication, configuration, and exposed services.

ADGM THREAT INTELLIGENCE SUMMARY

OverlayPhantom: The Android Banking Trojan Uses Fake Updates to Steal Credentials [Campaign] [High]

Identity Compromise Enables Multi-Layer Cloud Intrusion by Storm-2949 [Campaign] [High]

Screening Serpens Uses New RAT Variants in Espionage Campaigns [Campaign] [High]

Kali365 Uses Device Code Phishing to Hijack Microsoft 365 Access Tokens [Campaign] [High]

Campaign Combines Data Exfiltration and Destructive Actions Across Critical Sectors [Campaign] [High]

Threat Actors Use Teams Vishing and Cloud Services to Deploy Nimbus RAT [Campaign] [High]

FortiClient EMS Flaw Used to Deliver EKZ Infostealer Through Fake Patch [Campaign] [High]

A Cryptojacking Campaign Abusing ScreenConnect and Microsoft .NET Utilities [Campaign] [High]

Phishing Campaign Deploys JavaScript-Driven PureLogs Variant to Steal Sensitive Data [Campaign] [High]

TeamPCP Alleged GitHub Breach via Trojanized VS Code Extension [Campaign] [Medium]

BadIIS Enables Scalable Traffic Manipulation Through a Builder-Driven Malware Ecosystem [Campaign] [Medium]

MSHTA Legacy Utility Enables Multi-Stage Malware Delivery on Windows [Campaign] [Medium]

Phantom Killer Uses Vulnerable Driver Access to Terminate Security Processes [Campaign] [Medium]

Atlassian May 2026 Bulletin Addresses Critical and High-Severity Product Flaws [Vulnerability] [High]

NGINX Flaws Expose Internet-Facing Deployments to Memory Corruption and Potential Code Execution [Vulnerability] [High]

Trend Micro Addresses Actively Exploited Apex One On-Premise Vulnerability [Vulnerability] [High]

Notepad++ Addresses High-Severity Flaws - Immediate Patch Recommended [Vulnerability] [High]

MiniPlasma Revives a Windows Cloud Files Privilege Escalation Weakness [Vulnerability] [High]

Name	Threat Severity Rating	TLP	Attribution	Originating Source
OverlayPhantom: The Android Banking Trojan Uses Fake Updates to Steal Credentials	HIGH	CLEAR	Campaign	Open Source

Executive Summary

Researchers have identified OverlayPhantom, an Android banking trojan distributed through malicious URLs that use trusted app lures and a fake update process to trick users into installing the malware. Once installed, it abuses Android Accessibility features, displays fake login overlays on top of targeted apps and supports real-time screen streaming and remote device control.

Organizations in the financial sector should be aware that the malware is configured to target more than 180 banking, financial, and cryptocurrency applications across 10 countries. Its ability to steal credentials, monitor screen activity, and carry out remote actions could affect mobile users accessing financial services through infected Android devices.

Technical Details

- The campaign starts with malicious URLs that deliver a dropper application disguised as a trusted app, using social engineering to persuade victims to begin installation.
- The dropper then shows a fake update screen and a guided setup flow that encourages the user to enable Accessibility permissions.
- After installation, the malware hides behind the appearance of a trusted Android service, making it harder for users to spot and remove.
- With Accessibility access in place, the malware gains broad control of the device and can monitor what application is active on the screen.
- It checks the active application against a built-in list of targeted banking, finance, and cryptocurrency apps stored in its code.
- When a target app is opened, the malware loads a matching fake HTML page and places it over the legitimate app as a seamless overlay.
- Victims may believe they are interacting with the real app, but the overlay captures usernames, passwords, and payment card details entered on screen.
- The malware also supports more than 30 remote commands, allowing the operator to simulate gestures, manipulate the device, and trigger fake notifications.
- In addition, it can stream the device screen in near real time, giving the operator visual access to sensitive activity taking place on the phone.
- This campaign does not appear to involve any specifically identified UAE banking applications.

Recommendations

- Restrict Android app installations to trusted official sources and block installation attempts that originate from unsolicited links.

- Review and tightly control Accessibility permissions, especially when newly installed apps request elevated device interaction.
- Alert users to fake update screens, unexpected installation prompts, and suspicious login overlays that imitate trusted applications.
- Enable MFA on financial applications to reduce the impact of credentials captured through fraudulent overlays.
- Keep Android devices and apps updated and use mobile security tools that can detect malicious behavior and screen-capture abuse.

For a detailed breakdown of tactics, techniques, and procedures (TTPs), refer to the Annexure – OverlayPhantom: The Android Banking Trojan Uses Fake Updates to Steal Credentials

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
Identity Compromise Enables Multi-Layer Cloud Intrusion by Storm-2949	HIGH	CLEAR	Campaign	Open Source

Executive Summary

Researchers at Microsoft has identified a campaign in which the threat actor dubbed as “Storm-2949” used targeted social engineering around self-service password reset requests to trick users into approving MFA prompts, reset their passwords, replace authentication methods, and take over cloud identities like Azure, Azure SQL, Azure Storage, Microsoft Entra ID, Microsoft 365 and so on. The actor then used those compromised accounts to move from Microsoft 365 into Azure resources, access secrets, and exfiltrate data from production-linked services.

This activity could affect organizations that rely on cloud identities to manage business-critical applications, storage, and administrative functions across shared environments. For financial institutions with similar cloud exposure, the campaign shows how abuse of legitimate management features may enable broad access to sensitive data and production resources without depending heavily on traditional malware.

Technical Details

- The campaign began with targeted social engineering designed to get users to approve MFA prompts during a fraudulent self-service password reset process.
- Once a user approved the prompts, the actor reset the password, removed existing authentication methods, and enrolled a new authenticator to keep access.
- After the initial takeover, the actor used automated API queries to search the directory for users, applications, and privileged access paths.

- The compromised accounts were then used to access cloud file storage services and download large volumes of sensitive files from shared locations.
- The intrusion expanded into cloud infrastructure where the actor used privileged role assignments to reach web applications, secrets stores, storage accounts, and databases.
- By accessing a secrets store, the actor obtained credentials such as connection strings and identity details that helped them move deeper into the environment.
- The actor also changed access settings on database and storage resources to support data theft and then removed some of those changes to reduce visibility.
- In parallel, they abused virtual machine management features to create administrator access, run scripts, and attempt additional credential harvesting.
- A remote monitoring tool was later deployed on multiple systems, alongside attempts to weaken antivirus protections and remove forensic traces.

Recommendations

- Strengthen protections around self-service password reset workflows and closely review unusual MFA approval patterns tied to account recovery activity.
- Monitor for rapid changes to authentication methods, especially when they follow password resets or occur across multiple users.
- Review privileged cloud roles and restrict access to management actions that can expose application publishing profiles, secrets, storage keys, or VM access.
- Track configuration changes affecting database firewalls, storage network access, and virtual machine extensions to identify suspicious administrative activity.
- Correlate identity, cloud, and endpoint telemetry to detect attacks that move from compromised user accounts into wider cloud infrastructure.

For a detailed breakdown of tactics, techniques, and procedures (TTPs), refer to the Annexure – Identity Compromise Enables Multi-Layer Cloud Intrusion by Storm-2949

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
Screening Serpens Uses New RAT Variants in Espionage Campaigns	HIGH	CLEAR	Campaign	Open Source

Executive Summary

Researchers have identified a campaign in which Screening Serpens used tailored recruitment-themed lures, impersonated trusted brands and services, and delivered malicious archives that triggered DLL sideloading and 'AppDomainManager' hijacking to load six new RAT (Remote Access Trojan) variants. The activity targeted entities in the region, with the malware designed to disable native '.NET' security visibility before the main payload executed.

This campaign may impact organizations in the financial services sector that rely on user engagement with external recruitment or meeting-related content, particularly when cloud-hosted or internet-facing workflows are part of the process. It also highlights how loaders designed for stealth, combined with frequently changing infrastructure, can hinder detection efforts by limiting telemetry before the malware proceeds with command execution and data exfiltration.

Technical Details

- The campaign used targeted spear phishing lures designed to look legitimate, including fake recruitment material and impersonated meeting or application content to persuade victims to open malicious archives.
- After the archive was opened, the infection chain relied on DLL sideloading so malicious components could run under the appearance of trusted software.
- In the activity observed in mid-April 2026, the attackers deployed a newer MiniUpdate variant as part of the same broader espionage effort.
- The malware used 'AppDomainManager' hijacking to take control during the startup of a legitimate '.NET' application before the main program began.
- This execution method disabled event tracing and bypassed validation checks through a legitimate configuration file, helping the malware reduce security visibility.
- The first stage then showed a fake loading interface while quietly staging the next malware files and preparing persistence.
- Persistence was created through a scheduled task, allowing the renamed program to relaunch and continue the infection chain after the initial execution.
- In the second stage, the loader checked whether it was running in the expected environment before handing control to the final RAT payload.
- The newer variants expanded command handling and included the ability to split large files into smaller chunks for exfiltration. This likely improved reliability and reduced visibility during data theft.

- The campaign also used separate sets of rotating command-and-control infrastructure for different targets and variants. This approach improved resilience and reduced overlap between operations.

Recommendations

- Review controls around archive-based delivery and closely inspect unsolicited recruitment, meeting, or application-themed files before users open them.
- Monitor for DLL sideloading behavior and unusual use of '.NET' configuration files that alter normal application startup behavior.
- Investigate scheduled task creation tied to newly dropped or renamed executables, especially when linked to user-facing decoy activity.
- Track attempts to suppress event tracing or bypass verification checks, as these may indicate efforts to reduce endpoint visibility.
- Watch for staged exfiltration behavior, including repeated outbound transfers that may indicate files are being uploaded in smaller chunks.

For a detailed breakdown of tactics, techniques, and procedures (TTPs), refer to the Annexure – Screening Serpens Uses New RAT Variants in Espionage Campaigns

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

Reference to the Source

back to top

Name	Threat Severity Rating	TLP	Attribution	Originating Source
Kali365 Uses Device Code Phishing to Hijack Microsoft 365 Access Tokens	HIGH	CLEAR	Campaign	Open Source

Executive Summary

Kali365 is an emerging Phishing-as-a-Service (PhaaS) which was first observed in April 2026 and mainly distributed through Telegram. It enables threat actors to gain persistent access to Microsoft 365 environments by capturing OAuth tokens, allowing them to bypass MFA without stealing user credentials. The platform lowers the barrier for less-skilled attackers by offering AI-generated phishing lures, automated campaign templates, real-time target tracking dashboards, and token capture capabilities.

This campaign may impact organizations in the financial services sector that rely on Microsoft 365 for email, collaboration, and file-sharing workflows. By capturing access and refresh tokens instead of passwords, the activity may impact account security and allow continued access to Outlook, Teams, and OneDrive without requiring additional MFA challenges.

Technical Details

- Kali365 is described as a phishing-as-a-service platform that has primarily been distributed through Telegram since it was first observed in April 2026. It lowers the barrier to entry by giving less-technical actors access to phishing infrastructure and token capture features.
- The attack starts with a phishing email that impersonates trusted cloud productivity and document-sharing services. The lure is designed to make the target believe the request is part of a normal business workflow.
- The phishing message contains a device code and instructions to visit a legitimate Microsoft verification page. This use of a real verification page helps the attacker make the process look authentic to the target.
- When the target enters the device code, they unknowingly authorize the attacker's device to access the account. The attacker does not need to intercept the user's credentials to complete this step.
- After authorization, the attacker captures OAuth access and refresh tokens from the session. These tokens provide a way to maintain access without repeatedly asking for the password.
- With the stolen tokens, the attacker can access Microsoft 365 services such as Outlook, Teams, and OneDrive. This access can continue without triggering additional MFA prompts for the attacker.
- The platform also offers AI-generated phishing lures, automated campaign templates, and real-time tracking dashboards. These features support broader and easier operation of phishing campaign.

Recommendations

- Restrict OAuth device authorization flow to limit or block the use of device authentication codes where possible. This can help prevent this style of token theft.
- Create a conditional access policy to block OAuth device authorization flow for all users, with only limited exceptions for required business processes. Review these exceptions carefully before deployment.
- Audit existing device code flow usage to identify legitimate dependencies before enforcing restrictions. This helps reduce the risk of disrupting approved workflows.
- Block authentication transfer policies that allow users to transfer authentication from computers to mobile devices. This reduces another path that could support misuse of the flow.
- If device code flow cannot be fully restricted, exclude emergency access accounts to prevent lockouts. This supports resilience while tightening defensive controls.

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
Campaign Combines Data Exfiltration and Destructive Actions Across Critical Sectors	HIGH	CLEAR	Campaign	Open Source

Executive Summary

Researchers have identified a campaign which uses authenticated remote access, administrative consoles, and custom scripts to exfiltrate data and then destroy virtual machines, storage volumes, databases, backups, and operating system files across multiple victim environments. The activity combined scripted automation with hands-on-keyboard actions, allowing the operator to move through virtualization, database, backup, and web-hosting systems using legitimate management tools and tailored exfiltration utilities.

The implications of this campaign may affect organizations in the financial services sector, particularly in environments where cloud, virtualization, database, and backup systems are tightly interconnected. The combined use of data exfiltration tools and destructive techniques could hinder recovery efforts, especially when attackers simultaneously target stored data, backup chains, and administrative systems during a single intrusion.

Technical Details

- The campaign involved two main phases of attacker activity: stealing data and then carrying out destructive actions inside victim environments. The operators used both automated scripts and direct use of normal administrative tools to cause damage.
- In some cases, the attackers deleted virtual machines from management consoles and removed underlying disk files. They also accessed guest systems and deleted partitions and volumes through standard disk management tools.
- The operators used remote access into victim systems and then interacted with web servers, database tools, and local file systems. This gave them the ability to take databases offline, delete them, and erase hosted application content.
- Backup systems were also targeted during the intrusions. The attackers issued destructive delete actions against backup repositories, removing entire backup chains rather than only individual files.
- In another environment, a custom Python script was used to iterate through a list of SQL Server targets. The script forced user databases offline and then deleted them in sequence across multiple systems.
- The attackers also manually removed operating system and application folders from compromised hosts. In one case, the remote session dropped immediately after the deletions, indicating the destructive activity had succeeded.
- For exfiltration, the operators compressed stolen data into archive volumes and placed them on victim web servers before pulling the files back from external infrastructure. This allowed them to move data out through systems already under their control.
- The report also describes a custom Python-based receiver that accepted encrypted file chunks and reassembled them after upload. In addition, a bespoke uploader was used to collect files from local drives and shared locations before sending them to hardcoded infrastructure.

Recommendations

- Closely monitor administrative actions involving virtualization platforms, disk management tools, database consoles, and backup software, as these may be used for destructive activity after access is obtained.
- Review and restrict privileged access across application, database, virtualization, and backup environments to reduce the chance of one intrusion affecting multiple critical systems.
- Investigate unusual archive creation, staged file uploads to public-facing web locations, and large outbound transfers that may indicate data exfiltration activity.
- Monitor for scripted database administration actions and unexpected deletion activity across multiple servers, especially when performed in rapid sequence.
- Ensure backup protections include controls against direct repository deletion and regularly test recovery procedures for scenarios involving both data theft and destructive actions.

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
Threat Actors Use Teams Vishing and Cloud Services to Deploy Nimbus RAT	HIGH	CLEAR	Campaign	Open Source

Executive Summary

Researchers have identified a campaign in which threat actors used mailbox flooding, Microsoft Teams voice phishing, and Quick Assist to gain remote access before deploying a Java-based remote access trojan known as Nimbus RAT. In the observed intrusion, the actor posed as IT helpdesk staff, guided the user through remote assistance, and delivered the payload from a compromised Microsoft 365 tenant, with the full sequence taking less than 20 minutes from first Teams contact to RAT execution.

This campaign may impact organizations in the financial services sector that rely on Microsoft 365 collaboration workflows and remote support tools for everyday operations. Because the malware uses Google Drive and Google Sheets for command-and-control and can capture credentials through deceptive prompts, the campaign may impact visibility and account security even when network traffic appears benign.

Technical Details

- The attack begins with mailbox flooding, where the target receives a large volume of subscription confirmation emails in a short period. This creates confusion before the social engineering phase starts.
- Shortly afterward, the victim receives a Microsoft Teams message or call from an actor-controlled account posing as IT support. The message references the spam problem and offers help.
- Once the user engages, the actor walks them into launching Quick Assist and granting remote access. This gives the operator a direct path to continue the intrusion.

- The payload is then downloaded from a compromised Microsoft 365 tenant and executed on the target system. In the observed case, the transition from contact to malware execution was very rapid.
- Nimbus RAT is a self-contained Java implant that includes its own Java runtime. This allows it to run on Windows systems even if Java is not already installed.
- For command-and-control, the malware uses Google Drive and Google Sheets. This helps the traffic blend in with legitimate cloud service activity.
- The malware can steal credentials by showing either a fake Windows Security prompt or invoking the real credential prompt API. In both cases, it tries to obtain two password entries from the user.
- Persistence is not installed automatically by the RAT itself. Instead, the operator stages persistence separately, meaning early host isolation can stop the infection from surviving a reboot.
- The malware also supports in-memory execution of second-stage Java code. This allows the operator to expand functionality without dropping more files to disk.

Recommendations

- Detect and alert on mailbox flooding early, as it may provide the first visible sign of the attack chain before the Teams contact begins.
- Review and restrict external Teams communication from trial tenants where possible, as these were a major delivery path in the observed dataset.
- Train users to treat unsolicited Teams helpdesk outreach and remote support requests as suspicious, especially after unusual email activity.
- Rely on process-level and behavioral monitoring for detection since the malware's use of Google services may limit the value of simple network blocking.
- Isolate affected hosts quickly if this activity is suspected, because operator-established persistence may not survive if response occurs before follow-on action.

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
FortiClient EMS Flaw Used to Deliver EKZ Infostealer Through Fake Patch	HIGH	CLEAR	Campaign	Open Source

Executive Summary

Researchers have identified a campaign exploiting CVE-2026-35616 in FortiClient EMS (Endpoint Management Server) to push a malicious script to managed endpoints through normal management workflows. The activity modified EMS configuration, triggered script execution when endpoints established a

VPN tunnel, and then used PowerShell to download and run "EKZ Infostealer" disguised as a legitimate endpoint patch.

Organizations in the financial sector should be aware that this technique could affect environments that rely on centralized endpoint management to distribute trusted configuration changes at scale. Because the campaign used the management pathway itself to reach multiple devices and steal browser credentials, cookies, and autofill data, it may impact access to cloud services and internal applications beyond the initially affected hosts.

Technical Details

- The campaign began with exploitation of CVE-2026-35616, an improper access control issue that allowed unauthenticated requests to be handled as privileged administrative actions. This gave the attacker a way to interact with EMS functions without valid credentials.
- After access was gained, the attacker changed EMS settings, including remote access profile and endpoint policy configuration. These changes were used to prepare malicious script execution on managed devices.
- The malicious logic was tied to VPN configuration, so the script ran when affected endpoints established a tunnel. This allowed the attacker to trigger execution through a process that appeared to be part of normal endpoint management.
- The script then launched PowerShell and attempted to download the payload using several fallback methods. Once retrieved, the payload was executed silently on the endpoint.
- The downloaded file, named to resemble a patch, was identified as "EKZ Infostealer". It was described as a previously unreported browser credential stealer first observed in May 2026 during this campaign.
- "EKZ Infostealer" targets Chromium-based and Firefox-based browsers. It can collect saved passwords, cookies, and autofill data such as payment card details, addresses, and phone numbers.
- The malware stages the harvested information in a log file and the PowerShell script later sends the results out over HTTP. This created a simple but effective collection and exfiltration chain.
- The report notes that stolen cookies may allow attackers to reuse authenticated sessions. This could let them reach additional services without triggering another MFA prompt.

Recommendations

- Upgrade FortiClient EMS to a fixed version as soon as possible to reduce exposure to this campaign. The report specifically recommends moving affected deployments to a remediated release.
- Restrict access to the EMS management port so that only trusted IP ranges can reach it. This can reduce the chance of unauthorized interaction with the management plane.
- Monitor for certificate-authentication anomalies and unexpected remote access profile changes, as the report highlights these as early high-signal indicators of exploitation.
- Review endpoint activity for unusual PowerShell execution tied closely to VPN connection events. This behavior was directly linked to payload delivery in the observed campaign.
- Investigate signs of credential staging and rapid follow-on exfiltration from managed endpoints, especially when browser data appears to be the collection focus.

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
A Cryptojacking Campaign Abusing ScreenConnect and Microsoft .NET Utilities	HIGH	CLEAR	Campaign	Open Source

Executive Summary

Researchers at Microsoft have identified an active cryptojacking campaign in which malicious download sites promoted through poisoned search results, and in some cases AI chatbot responses, impersonated trusted system utility software to reach users likely to own high-performance GPUs (Graphics Processing Unit). The infection chain used a malicious archive, DLL sideloading, silent installation of remote access software, and a follow-on dropper that hollowed legitimate signed '.NET' utilities to launch mining activity and maintain control of the device.

Organizations in the financial sector should be aware that this activity could affect users who search for common utility software outside approved channels, especially on high-performance workstations. Beyond unauthorized GPU mining, the use of persistent remote access and trusted signed processes may impact visibility into follow-on activity and could affect efforts to detect data theft, lateral movement, or additional payload delivery.

Technical Details

- The campaign begins when users search for popular system utilities and are redirected to lookalike sites through poisoned search results. In some observed cases, users were also directed to malicious domains through AI chatbot recommendations.
- These fake sites offer a download that appears legitimate, but the archive contains a real utility paired with a malicious DLL. When the user launches the program, DLL sideloading starts the malicious chain without obvious warning signs.
- The malicious DLL silently installs remote access software, giving the attacker a persistent way to interact with the compromised system. This step establishes control before the mining payload is delivered.
- After remote access is available, the attacker drops a second-stage loader that copies itself to a hidden location and creates several persistence methods. These include scheduled tasks, registry autoruns, and a startup shortcut so the malware relaunches regularly.
- The loader then selects a legitimate signed .NET utility and hollows the process so malicious code runs under a trusted process name. This helps the mining activity blend in with normal system activity.
- The malware also adds antivirus exclusions and performs anti-analysis checks before moving forward. It checks for virtualized environments and common analyst tools, then exits if suspicious conditions are detected.

- Once active, the malware profiles the host, including CPU, GPU, memory, operating system, user activity, and security posture. It uses this information to decide when mining should run and when it should pause.
- The mining component is downloaded at runtime rather than embedded in the loader. The malware also checks regularly that persistence remains in place and restores missing entries if they are removed.

Recommendations

- Turn on cloud-delivered protection and keep endpoint detection in blocking mode to improve coverage against rapidly changing tooling used in this campaign.
- Enable network and web protection so malicious download sites involved in poisoned search result delivery can be blocked earlier in the attack chain.
- Use browser protections that can identify and block malicious or deceptive websites, especially for users who frequently download utility software.
- Avoid storing enterprise credentials in browsers or in personal password managers protected by individual accounts, as this increases browser-based credential exposure.
- Turn on the attack surface reduction rule that blocks executable files from running unless they meet trusted, age, or prevalence criteria.

For a detailed breakdown of tactics, techniques, and procedures (TTPs), refer to the Annexure – A Cryptojacking Campaign Abusing ScreenConnect and Microsoft .NET Utilities

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
Phishing Campaign Deploys JavaScript-Driven PureLogs Variant to Steal Sensitive Data	HIGH	CLEAR	Campaign	Open Source

Executive Summary

Researchers have identified a phishing campaign that delivers a PureLogs variant through purchase-order-themed emails carrying a malicious archive attachment. When the attachment is opened, an obfuscated JavaScript file launches a hidden PowerShell script, which then uses process hollowing to inject a downloader into a trusted Windows process and pull a fileless plugin into memory.

This campaign may impact organizations in the financial services sector that could affect users who handle supplier, procurement, or document-related email workflows. The malware's fileless execution, layered encryption, and broad collection of browser, application, messaging, and cryptocurrency-related data may impact visibility into credential theft and sensitive data exposure on infected endpoints.

Technical Details

- The campaign begins with phishing emails disguised as purchase orders and includes a compressed attachment meant to convince the recipient to open it. This social engineering step is used to trigger the rest of the infection chain.
- Inside the archive is an obfuscated JavaScript file. When executed, it decrypts hidden PowerShell code, writes it to a temporary script file, and runs it in a hidden window.
- The PowerShell stage contains encrypted and encoded content that is decoded in memory. This results in a fileless script that avoids dropping the main payload openly to disk.
- That script then loads two '.NET' modules in memory and uses process hollowing to launch one of them inside a trusted Windows process. This helps the malware blend its activity with normal system behavior.
- Once running, the injected module extracts a downloader from its resources, decrypts it, and loads it directly in memory. The downloader then reaches out to its command server and requests an additional plugin.
- The downloaded plugin is a fileless PureLogs variant that remains in memory. It is heavily obfuscated and is used as the main information-stealing component of the campaign.
- PureLogs collects screenshots, system details, saved browser credentials, cookies, autofill data, and information from messaging, email, file transfer, remote access, and cryptocurrency wallet applications. The collected data is compressed and encrypted before transmission.
- The campaign uses multiple layers of encryption and staged memory execution throughout the chain. This design makes detection harder for defenses that rely mainly on simple file-based signatures.

Recommendations

- Strengthen email filtering for invoice, purchase order, and other document-themed messages, especially when they contain compressed attachments. This can reduce exposure at the first step of the campaign.
- Disable unnecessary script execution where possible, since the attack relies on JavaScript and PowerShell to move from the email lure to in-memory payload delivery.
- Monitor for hidden or unusual PowerShell activity launched by scripting engines. This behavior is a key transition point in the observed infection chain.
- Watch for signs of process hollowing involving trusted Windows processes, as the malware uses this technique to run the downloader and plugin with reduced visibility.
- Hunt for broad credential and data collection activity across browsers and common user applications, as the malware is designed to gather multiple categories of sensitive information from a single host.

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

Reference to the Source

back to top

Name	Threat Severity Rating	TLP	Attribution	Originating Source
TeamPCP Alleged GitHub Breach via Trojanized VS Code Extension	MEDIUM	CLEAR	Campaign	Open Source

Executive Summary

Researchers have identified a campaign in which a compromised code editor extension was published with injected code that silently launched a hidden package fetch when a developer opened a workspace, then executed an obfuscated payload to harvest credentials, cloud tokens, and CI/CD secrets. The attack used a stolen contributor token and stolen marketplace publishing credentials, allowing the malicious version to appear legitimate long enough to reach users before removal.

This campaign may impact organizations in the financial services sector that this activity may impact software development, DevOps, and cloud engineering environments where developer workstations hold broad access to repositories, package registries, CI/CD systems, and secret stores. The reported follow-on activity could affect internal code security and trusted build processes, as the intrusion was linked to unauthorized access to internal repositories after an employee device was compromised by the poisoned extension.

Technical Details

- A malicious extension version was published and remained available only briefly, but the advisory later stated that internal telemetry indicated thousands of activations despite initially lower marketplace download counts.
- When the extension activated, it silently launched a hidden package execution task disguised as normal setup behavior, reducing the chance that the user would notice anything unusual.
- That task pulled code from a hidden orphan commit placed inside the official repository, allowing the attacker to deliver the real payload without embedding it directly in the extension package.
- The downloaded payload used anti-analysis checks, detached itself into the background, and then began parallel collection of credentials and secrets from developer and cloud tooling.
- Reported collection targets included repository tokens, package registry credentials, cloud secrets, vault data, private keys, and other development-related secrets available on disk, in environment variables, and in some cases process memory.
- Stolen data was exfiltrated through multiple channels, including HTTPS, the platform API, and DNS, so blocking a single path would not fully disrupt the theft.
- The payload also contained logic related to trusted publishing and provenance workflows, creating concern that stolen tokens could support downstream package abuse that appears legitimate.
- Separate reporting stated that the poisoned extension compromise led to unauthorized access to internal repositories, with the claimed scale described as directionally consistent with the ongoing investigation.

Recommendations

- Immediately update to the fixed extension version or later, and ensure the compromised version is no longer present in affected developer environments.

- Terminate any suspicious background processes linked to the malicious extension and remove persistence artifacts created during execution.
- Rotate every credential reachable from impacted developer machines, including tokens, secrets, and SSH keys, and audit related access logs for misuse.
- Review developer endpoints for unexpected hidden package execution and other unauthorized background activity tied to extension startup behavior.
- Harden extension and release publishing workflows with stronger approval controls so a single compromised contributor account cannot publish malicious updates.

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
BadIIS Enables Scalable Traffic Manipulation Through a Builder-Driven Malware Ecosystem	**MEDIUM**	**CLEAR**	**Campaign**	**Open Source**

Executive Summary

Researchers have identified a BadIIS variant that appears to function as a commodity malware offering used by multiple Chinese-speaking cybercrime groups. The malware is designed to manipulate web traffic on compromised IIS (Internet Information Services) servers and is supported by a dedicated builder tool that lets operators generate customized payloads for traffic redirection, reverse proxying, content hijacking, and backlink injection.

Organizations in the financial sector should be aware that this activity could affect internet-facing IIS environments that support public websites or web applications. Because the toolset is actively maintained, customizable, and supported by additional installation and persistence components, it may impact website integrity, search visibility, and trust in web content served through compromised infrastructure.

Technical Details

- The investigated BadIIS variant is described as a commodity tool rather than a single-actor implant. The report assesses that multiple cybercrime groups likely use it under a malware-as-a-service style model.
- The malware's core purpose is traffic manipulation on compromised IIS servers. Its observed functions include redirecting visitors, acting as a reverse proxy for crawlers, replacing site content, and injecting backlinks for SEO (Search Engine Optimization) fraud.
- A dedicated builder tool allows operators to create custom configurations and embed them directly into payloads. This makes the malware easier to adapt for different victims and campaign goals.
- The builder supports several modes of abuse, including redirecting users to illicit content, serving hidden content to search engines, and hijacking website pages at selected rates.

- Some customized builds were designed to evade specific security products or change behavior based on browser language and environment. This suggests an active effort to improve effectiveness and reduce detection.
- Beyond the main malware, the same author developed installer, dropper, and service-based tools to deploy the IIS payload. These tools automate installation and support persistence across server restarts.
- The latest installation workflow separates deployment into multiple stages and stores backup copies of the payload. This allows the malware to be restored if the active component is removed.

Recommendations

- Review internet-facing IIS servers for unauthorized module registration, unexpected content changes, and abnormal traffic redirection behavior. These are central to the malware's observed function.
- Investigate IIS environments for suspicious reverse proxy behavior or hidden content delivery to search engine crawlers. The builder specifically supports this type of abuse.
- Look for signs of staged installation activity and service-based persistence, especially where server behavior survives cleanup or reappears after restart.
- Monitor for unexplained changes that affect page content, metadata, backlinks, or search engine indexing behavior. These may indicate SEO-focused manipulation on compromised servers.
- Prioritize integrity checks on public-facing web infrastructure where trust, branding, and content consistency are operationally important.

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
MSHTA Legacy Utility Enables Multi-Stage Malware Delivery on Windows	MEDIUM	CLEAR	Campaign	Open Source

Executive Summary

Researchers have identified continued abuse of MSHTA (Microsoft HTML Application), a legacy Windows utility that can execute script content from local or remote files and is still available by default on many systems. The activity spans several malware delivery chains in which attackers use MSHTA to retrieve HTA content, launch hidden scripts, and hand execution to later stages such as PowerShell, WScript, or final payloads.

Organizations in the financial sector should be aware that this technique could affect users exposed to phishing, fake software downloads, or human-verification lures that trigger script execution through trusted Windows components. Because MSHTA is used as a lightweight staging mechanism in multi-stage and often

fileless chains, it may impact visibility into early infection activity and could affect efforts to stop credential theft or follow-on malware delivery before deeper compromise occurs.

Technical Details

- The campaign centers on MSHTA, a signed Windows utility that attackers use to run HTA files containing JavaScript or VBScript. Its trusted nature helps the first stage blend into normal system activity.
- Across the observed activity, MSHTA was mainly used as a staging tool rather than the final payload. It commonly retrieved remote script content and passed execution to the next step in the chain.
- One common delivery path involved fake or cracked software lures. In those cases, the user executed a disguised installer that ultimately launched a renamed MSHTA binary to contact attacker-controlled infrastructure.
- The remote HTA content then decoded and launched the next payload, often through obfuscated script logic. This allowed the attacker to keep early stages lightweight and flexible.
- Another observed chain used fake human-verification pages shared through messaging platforms. Victims were tricked into pasting and running a command that launched MSHTA with a remote HTA file.
- That HTA ran directly in memory, hid its window, decoded a second-stage script, and then launched PowerShell to continue execution. This reduced obvious on-disk evidence during the early stages.
- The later PowerShell stages were heavily obfuscated and, in the analyzed case, led to an in-memory assembly load used to execute an information stealer.
- The report also describes another chain in which MSHTA launched a remote HTA that quickly shifted to PowerShell-based persistence and payload delivery. This shows the same utility being reused across different malware families.

Recommendations

- Move away from MSHTA in administrative workflows wherever possible, as the report notes its legitimate use is declining while malicious use remains active.
- Apply layered technical controls and user awareness measures, since the observed campaigns relied heavily on social engineering as well as trusted system binaries.
- Monitor for unusual MSHTA usage, especially when it retrieves remote script content or appears alongside HTA, PowerShell, or WScript execution.
- Investigate fake software download themes and human-verification lures, as these were directly used to start the infection chains described in the report.
- Prioritize visibility into multi-stage, fileless execution behavior, particularly where MSHTA hands off execution to hidden or heavily obfuscated scripts.

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
Phantom Killer Uses Vulnerable Driver Access to Terminate Security Processes	MEDIUM	CLEAR	Campaign	Open Source

Executive Summary

A security researcher has described Phantom Killer, a technique that abuses a signed driver associated with a PC management utility to terminate processes from user mode. The analysis shows that the driver exposes a device interface without meaningful access restrictions, allowing a user process to send a process identifier to the driver and trigger kernel-level termination of the selected process.

Organizations in the financial sector should be aware that this technique could affect Windows environments where vulnerable signed drivers are already present or can be introduced through a bring-your-own-vulnerable-driver scenario. Because the described behavior may impact endpoint protection visibility by terminating security processes before follow-on activity, it could affect detection and containment during the early stages of intrusion activity.

Technical Details

- The described technique centers on a signed driver linked to a PC management utility and presented as a strong candidate for a bring-your-own-vulnerable-driver attack. The researcher noted that the driver appeared trusted and initially had no detections when examined.
- During reverse engineering, the researcher found that the driver creates a device object and symbolic link that can be reached from user mode. The write-up states that the device object was not secured with a restrictive access control configuration.
- The analysis of the driver's open handling showed no meaningful access checks when a user process attempted to get a handle to the driver. This means even a low-privileged process may be able to communicate with it.
- The driver was described as using a single control path that accepts a 4-byte buffer. That value is then passed into a function identified by the researcher as the core process-killing routine.
- According to the write-up, the routine takes a process ID, obtains a handle to the target process, and terminates it from kernel mode. This makes the driver useful for targeting high-value processes that are otherwise difficult to stop.
- The researcher outlined two weaponization paths: abuse of the driver if it is already loaded on the system or loading it as part of a bring-your-own-vulnerable-driver chain and then using it to terminate security tooling.

Recommendations

- Identify and review the presence of vulnerable signed drivers in Windows environments, especially those that expose user-accessible device interfaces without strong access restrictions.
- Monitor for unusual user-mode interaction with kernel drivers and for attempts to terminate security processes through non-standard system activity.

- Restrict the ability to load additional kernel drivers and review controls that could limit bring-your-own-vulnerable-driver scenarios.
- Investigate sudden loss of endpoint protection processes as a possible precursor to credential theft, tooling execution, or broader post-compromise activity.
- Prioritize remediation or blocking of known vulnerable drivers that can be reached from low-privileged user space.

For a detailed breakdown of tactics, techniques, and procedures (TTPs), refer to the Annexure – Phantom Killer Uses Vulnerable Driver Access to Terminate Security Processes

Reference to the Source

back to top

Name	Threat Severity Rating	TLP	Attribution	Originating Source
Atlassian May 2026 Bulletin Addresses Critical and High-Severity Product Flaws	**HIGH**	**CLEAR**	**Vulnerability**	**CSC**

Executive Summary

Atlassian has released its May 2026 Security Bulletin addressing multiple vulnerabilities across several Data Center and Server products, including Bamboo, Bitbucket, Confluence, Fisheye/Crucible, Jira Software, and Jira Service Management. The bulletin includes critical- and high-severity issues that could allow broken authentication, remote code execution, denial of service, file inclusion, cross-site scripting, information disclosure, and security misconfiguration attacks.

These vulnerabilities may impact environments that depend on Atlassian platforms for development, collaboration, ticketing, and service workflows. If affected versions remain unpatched, the combined presence of authentication, code execution, file inclusion, and denial-of-service weaknesses could affect system availability, application security, and access to sensitive operational information.

Technical Details

- The bulletin covers multiple Atlassian Data Center and Server products, showing that the exposure is spread across collaboration, development, and service management platforms rather than a single application.
- Two critical-severity issues were highlighted: CVE-2026-29145 for broken authentication and session management, and CVE-2026-22732 for security headers omission. These issues could weaken core protection controls if left unresolved.
- Several high-severity vulnerabilities affect shared dependencies and product components, increasing the chance that multiple products may be exposed through common libraries or embedded services.

- The reported high-severity issues include remote code execution flaws tied to dependencies such as mchange-commons-java, c3p0, and jackson-core. These are among the most serious issues because they may enable attacker-controlled code execution.
- Multiple denial-of-service vulnerabilities were also addressed, including issues tied to product components and dependencies such as ActiveMQ, PostgreSQL, and commons-fileupload. These could disrupt normal application availability.
- The bulletin also lists file inclusion vulnerabilities affecting Jira products and Jira Service Management. These issues may expose systems to unauthorized file handling or unsafe content loading.
- Information disclosure, injection, request smuggling, and cross-site scripting issues were also included, showing that the risk spans both backend processing and user-facing web functionality.

Recommendations

- Update affected Atlassian products to the fixed or latest released versions listed in the bulletin as soon as possible.

Detailed Vulnerability Details and Affected Products can be found *here*.

back to top

Name	Threat Severity Rating	TLP	Attribution	Originating Source
NGINX Flaws Expose Internet-Facing Deployments to Memory Corruption and Potential Code Execution	**HIGH**	**CLEAR**	**Vulnerability**	**CSC**

Executive Summary

NGINX has disclosed two critical vulnerabilities, CVE-2026-9256 in ngx_http_rewrite_module and CVE-2026-8711 in the NGINX JavaScript module, both of which can be triggered by crafted HTTP requests and may lead to heap buffer overflows in worker processes. The rewrite issue occurs when overlapping PCRE captures are used in rewrite directives with multiple capture references, while the JavaScript issue occurs when js_fetch_proxy is configured with client-controlled variables and a location invokes ngx.fetch().

Organizations in the financial sector should be aware that these vulnerabilities may impact internet-facing NGINX deployments used for web delivery, proxying, or application traffic handling. Because both issues are described as data plane exposures that may cause worker restarts and could affect code execution under specific conditions, unpatched systems may face service disruption and increased risk where exposed configurations are present.

Technical Details

- CVE-2026-8711 affects the NGINX JavaScript module when js_fetch_proxy builds proxy URLs from client-controlled variables such as HTTP headers, arguments, or cookies. Improper handling of these values can corrupt memory in the worker process.

- The reported outcome for CVE-2026-8711 is a heap-based buffer overflow that may lead to denial of service and, under specific conditions, possible remote code execution.
- A separate issue, referred to as “nginx-poolslip” vulnerability was publicly revealed as an unpatched zero-day on May 21, 2026, after being autonomously discovered by NebSec's AI security agent.
- It was officially tracked and assigned CVE-2026-9256 the following day on May 22, 2026.
- The flaw is a critical heap buffer overflow in the NGINX rewrite module triggered by specific overlapping regular expression patterns, reportedly affects NGINX 1.31.0.

Recommendations

- Update NGINX JavaScript (njs) to version 0.9.9 or later wherever affected versions are in use.
- For CVE-2026-9256, requiring an upgrade to NGINX 1.30.2 or 1.31.1.

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

Detailed Vulnerability Details and Affected Products can be found *here* and *here*.

back to top

Name	Threat Severity Rating	TLP	Attribution	Originating Source
Trend Micro Addresses Actively Exploited Apex One On-Premise Vulnerability	**HIGH**	**CLEAR**	**Vulnerability**	**CSC**

Executive Summary

Trend Micro has disclosed multiple vulnerabilities affecting TrendAI Apex One, Trend Micro Apex One as a Service, and TrendAI Vision One Endpoint Security - Standard Endpoint Protection. Among them, CVE-2026-34926 affects Apex One On-Premise and is reportedly being actively exploited in the wild, where successful exploitation could allow an authenticated attacker to tamper with arbitrary files, distribute malicious code to security agents, or escalate privileges within the affected environment.

Organizations in the financial sector should be aware that this activity may impact environments that rely on Apex One On-Premise for endpoint protection and centralized agent management. The combination of an actively exploited path traversal issue and several local privilege escalation flaws could affect trust in management infrastructure and may increase the risk of unauthorized changes to protected endpoints if vulnerable systems remain in use.

Technical Details

- The disclosure covers several Trend Micro security products, including TrendAI Apex One, Trend Micro Apex One as a Service, and TrendAI Vision One Endpoint Security - Standard Endpoint Protection. This indicates that the exposure spans both on-premise and service-linked endpoint security environments.
- The most urgent issue is CVE-2026-34926, which is described as a relative path traversal vulnerability affecting TrendAI Apex One On-Premise. The issue is reportedly being actively exploited in the wild.

- Successful exploitation of CVE-2026-34926 could allow an authenticated attacker to tamper with arbitrary files. This creates a direct risk to file integrity within the affected environment.
- The same vulnerability could also allow attackers to distribute malicious code to security agents. This is particularly important because it affects the trust relationship between the management server and protected endpoints.
- The disclosure also lists multiple local privilege escalation vulnerabilities affecting TrendAI Apex One and TrendAI Vision One Endpoint Security - Standard Endpoint Protection. These include CVE-2026-34927, CVE-2026-34928, CVE-2026-34929, CVE-2026-34930, CVE-2026-45206, and CVE-2026-45207, all described as origin validation error issues with CVSS v3.1 7.8 High.
- An additional high-severity issue, CVE-2026-45208, is described as a time-of-check time-of-use (TOCTOU) local privilege escalation vulnerability. This broadens the exposure beyond a single weakness type.

Recommendations

- Patch affected Trend Micro products without delay, with priority given to environments running Apex One On-Premise.
- Validate the integrity of management servers and deployed agents to check for signs of tampering or unauthorized code distribution.

Detailed Vulnerability Details and Affected Products can be found *here*.

back to top

Name	Threat Severity Rating	TLP	Attribution	Originating Source
Notepad++ Addresses High-Severity Flaws - Immediate Patch Recommended	**MEDIUM**	**CLEAR**	**Vulnerability**	**Open Source**

Executive Summary

Notepad++ released version 8.9.6.1 to address three vulnerabilities, including a crash issue caused by malformed structures, an arbitrary code execution flaw through config.xml, and an arbitrary code execution flaw through shortcuts.xml. The release notes and security pages show these fixes were issued as a rushed security update and were presented as important vulnerability remediations for affected users.

Organizations in the financial sector should be aware that version 8.9.6.1 itself was later found to contain a high-severity bypass affecting the “shortcuts[.]xml” protection logic, and the advisory lists version 8.9.6.2 as the patched release. This could affect environments where Notepad++ is used on managed Windows endpoints, especially if configuration files or command execution features are relied on in daily workflows.

Technical Details

- Version 8.9.6.1 was released on May 26, 2026, and was described as fixing three security issues. These included one crash vulnerability and two arbitrary code execution issues tied to configuration handling.

- The fixed issues in 8.9.6.1 were listed as CVE-2026-48770, CVE-2026-48778, and CVE-2026-48800. The release and download pages both identify these as the core security fixes in that version.
- Shortly after, a separate advisory reported a bypass in version 8.9.6.1 affecting the protection added for the "shortcuts[.]xml" issue. The advisory lists v8.9.6.1 as affected and v8.9.6.2 as patched.
- The bypass was described as high severity and could lead to arbitrary code execution without user confirmation. The advisory attributes the issue to weaknesses in path handling and link resolution checks.
- The write-up explains that the trusted-directory check did not canonicalize paths before validating them. As a result, crafted command paths could pass the trust check while still resolving outside the intended trusted locations.

Recommendations

- Upgrade Notepad++ to version 8.9.6.2 or later to address the bypass affecting version 8.9.6.1.

Detailed Vulnerability Details and Affected Products can be found *here*, *here* and *here*.

back to top

Name	Threat Severity Rating	TLP	Attribution	Originating Source
MiniPlasma Revives a Windows Cloud Files Privilege Escalation Weakness	MEDIUM	CLEAR	Vulnerability	Open Source

Executive Summary

Microsoft has disclosed CVE-2020-17103, a Windows Cloud Files Mini Filter Driver elevation of privilege vulnerability, as an Important security issue that could allow a low-privileged local attacker to gain higher privileges on an affected system. The vulnerability impacts the Cloud Files Mini Filter Driver component and is tracked as a privilege management weakness tied to local access rather than remote exploitation.

Organizations in the financial sector should be aware that this issue may impact Windows endpoints and servers where local privilege escalation could support broader post-compromise activity. The availability of public attention around this flaw could affect remediation urgency, especially where patch validation and endpoint hardening have not been recently reviewed.

Technical Details

- CVE-2020-17103 is identified as a Windows Cloud Files Mini Filter Driver Elevation of Privilege Vulnerability. It is associated with the Windows Cloud Files Mini Filter Driver component rather than a remote-facing service.
- Microsoft rates the issue as Important with a CVSS 3.1 score of 7.0, while NVD (National Vulnerability Database) lists it as High severity with a CVSS 3.1 score of 7.8. This difference may affect internal prioritization if organizations rely on a single scoring source.
- The vulnerability requires local access with low privileges and does not require user interaction, which means exploitation begins after an attacker already has some level of execution on the system.

- Microsoft's vector reflects high attack complexity, while NVD records low attack complexity, showing a difference in exploitability assessment between the two sources.
- The weakness enumeration references improper privilege management, indicating the flaw relates to how privileged operations are handled within the affected component.
- Microsoft's original publication stated the issue was not publicly disclosed and not exploited at that time, with an assessment of Exploitation Less Likely.

Recommendations

- Apply the official Microsoft fix associated with CVE-2020-17103 without delay.
- Verify patch compliance on Windows systems where the Cloud Files Mini Filter Driver is present.

Detailed Vulnerability Details and Affected Products can be found *here* and *here*.

back to top

Appendix A - Tactics, Techniques & Procedures (TTPs)

OverlayPhantom: The Android Banking Trojan Uses Fake Updates to Steal Credentials

Tactic	Technique	Procedure
Initial Access (TA0027)	Phishing (T1660)	OverlayPhantom is distributed via phishing sites
Persistence (TA0028)	Event Triggered Execution: Broadcast Receivers (T1624.001)	OverlayPhantom implemented a broadcast receiver for screen capturing
Defense Evasion (TA0030)	Hide Artifacts: Suppress Application Icon (T1628.001)	OverlayPhantom hides its icon
Defense Evasion (TA0030)	Obfuscated Files or Information (T1406)	Malware uses obfuscated strings
Defense Evasion (TA0030)	Masquerading: Match Legitimate Name or Location (T1655.001)	OverlayPhantom masquerades as Google Play Service
Credential Access (TA0030)	Abuse Accessibility Features (T1453)	OverlayPhantom abuses Accessibility service
Discovery (TA0032)	Software Discovery (T1418)	OverlayPhantom checks the installed application list against the target list
Collection (TA0035)	Screen Capture (T1513)	OverlayPhantom captures screen content
Command & Control (TA0037)	Application Layer Protocol (T1437)	OverlayPhantom communicates with C2 over TCP
Command & Control (TA0037)	Non-Standard Port (T1509)	OverlayPhantom uses a non-standard port
Exfiltration (TA0036)	Exfiltration Over C2 Channel (T1646)	OverlayPhantom exfiltrates data to the C&C server

Identity Compromise Enables Multi-Layer Cloud Intrusion by Storm-2949

Tactic	Observed activity
Initial access	Sign-in activity from attacker infrastructure to compromised identities
Initial access	Sign-in and authentication activity to Azure resources
Execution	Various types of execution-related suspicious activity by an attacker were observed
Persistence	Attacker device registered as MFA method
Persistence	ScreenConnect installed on Azure VMs
Defense evasion	Attempts to tamper with Microsoft Defender Antivirus
Defense evasion	Manipulation of Azure Storage account, Key Vault, and SQL database configurations
Credential access	Secret extraction from Azure Key Vault
Credential access	Attempted theft of workload identity tokens using Azure VM Run Command
Credential access	Credential harvesting from endpoints through ScreenConnect
Credential access	Publishing Azure App Service web app profile for credential access
Credential access	Listing Azure storage account access keys for access
Discovery	Domain and system discovery commands run on virtual machines
Lateral movement	Traversal between cloud resources and applications
Exfiltration	Data exfiltration from Azure Storage accounts and other resources
Exfiltration	Data exfiltration from file storage services

Screening Serpens Uses New RAT Variants in Espionage Campaigns

Tactic	Technique	Observed Activity
Initial Access	Phishing (Spear-phishing)	Use of highly tailored recruitment lures and impersonation of trusted entities to deliver malicious payloads
Execution	User Execution	Victims execute malicious files packaged as job application or meeting materials
Execution	DLL Sideloading	Malware execution via sideloaded libraries from legitimate applications
Persistence	Scheduled Task / Job	Establishment of persistence through automated execution mechanisms
Defense Evasion	Modify Application Behavior	Use of AppDomainManager hijacking to disable security controls during application initialization
Command and Control	Remote Access Software	Deployment of multi-functional RATs enabling remote control and data exfiltration
Delivery	SEO poisoning	Search engine manipulation to distribute malicious payloads
Collection	Data exfiltration	Extraction of sensitive information from compromised systems

A Cryptojacking Campaign Abusing ScreenConnect and Microsoft .NET Utilities

Tactic	Observed activity
Execution	Unusual ScreenConnect service creation activity
Execution	Malicious DLL sideloading linked to autorun.dll
Execution	ScreenConnect Installation activity
Execution	Defender detection of crypto mining framework binary

Execution	MDAV detection of suspicious DLL
Persistence	Scheduled task creation activity associated with malicious binary
Persistence	Malicious ASEP linked with malicious binary execution
Persistence	Suspicious .LNK file in startup folder
Defense Evasion	Antivirus exclusion added by malicious binary
Defense Evasion	Process hollowing activity to malicious binary
Command and control	Attacker executing malicious commands via ScreenConnect

Phantom Killer Uses Vulnerable Driver Access to Terminate Security Processes

Tactic	Technique	Observed Activity
Defense Evasion	Impair Defenses	Termination of EDR processes using a weaponized driver
Privilege Escalation	Exploitation for Privilege Escalation	Leveraging kernel driver capabilities for elevated access
Defense Evasion	Abuse Elevation Control Mechanism	Use of signed driver to bypass OS protections
Execution	Native API	Interaction with driver through low-level system calls
Defense Evasion	Rootkit	Kernel-level manipulation to disable security tools

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,	Recipients may share this information without restriction. Information is subject to standard copyright rules.

	in accordance with applicable rules and procedures for public release.	

Appendix D - Acronyms & Technical Terms

Term / Acronym	Meaning / Description
Ababil of Minab	The name used by the attacker persona in the destructive and exfiltration-focused intrusion campaign covered in the newsletter. It was associated with data theft and system destruction across several victim environments.
Accessibility Service	An Android feature that can assist users with device interaction. In the newsletter, malware abused it to monitor app activity, simulate actions, and capture sensitive input on infected devices.
ActiveMQ	A messaging component referenced in the Atlassian bulletin as part of a denial-of-service exposure. It appears as a backend dependency that could affect application availability if vulnerable.
AES	Advanced Encryption Standard. It is a common encryption method used in several malware chains described in the newsletter to protect stolen data or payloads during transfer and loading.
Amatera	Another information stealer mentioned in the MSHTA reporting. It appeared as an alternative final payload delivered through the same loader ecosystem.
AMSI	Antimalware Scan Interface. In the MSHTA-related attack chain, a PowerShell stage was described as using an AMSI bypass before loading the next payload in memory.
API	Application Programming Interface. It is a software interface used by applications and services to exchange data or perform actions; attackers used APIs for tasks such as directory discovery, token abuse, and platform interaction.
App Service	A cloud-hosted web application service referenced in the cloud intrusion campaign. Attackers targeted these hosted applications as part of their broader access and data-exfiltration objectives.
AppDomainManager Hijacking	A .NET execution technique that manipulates application startup so malicious code runs before the main program. In the observed campaign, it was used to disable visibility mechanisms and launch malware more stealthily.
Arbitrary Code Execution	The ability for an attacker to run code of their choosing on a system. Several vulnerabilities and malicious workflows in the newsletter created a risk of this outcome.
ASLR	Address Space Layout Randomization. It is a memory protection feature that makes exploitation harder by randomizing where code and data are loaded in memory. Some vulnerabilities noted that code execution becomes more feasible if ASLR is disabled or bypassed.
ASR Rule	Attack Surface Reduction rule. These are protective controls that help block risky behaviors such as untrusted executable launches or suspicious script activity.
Authentication Method	A configured way for a user to verify identity, such as an app or phone factor. In the cloud intrusion campaign, attackers changed authentication methods to retain access after resetting passwords.
Backlink Injection	The insertion of links into web content to influence search engine ranking or transfer website reputation to other domains. This was one of BadIIS's monetization features.
BadIIS	A malware family that infects IIS web servers and manipulates website traffic, content, and search engine behavior for monetization and redirection purposes.
Buffer Overflow	A memory corruption condition where more data is written than a memory area can safely hold. Several vulnerabilities in the newsletter involved heap-based buffer overflows that could cause crashes or possibly code execution.

BYOVD	Bring Your Own Vulnerable Driver. A technique in which a trusted but flawed driver is introduced or abused so an attacker can perform privileged actions from user space.
C2 / Command and Control	Infrastructure used by attackers to send instructions to malware and receive stolen data or status information from compromised systems.
Canonicalize / Canonicalization	The process of converting a file path into its final, normalized form before trust or safety checks are applied. The Notepad++ bypass existed because path validation did not do this first.
ClipBanker	A malware family referenced in the MSHTA reporting that is primarily associated with hijacking cryptocurrency-related clipboard activity.
Clipboard Hijacking	A technique where malware modifies clipboard content to redirect or alter user actions, often for fraud or credential theft. It was referenced in malware capabilities and related delivery chains.
Cloud Files Mini Filter Driver	A Windows component associated with cloud-backed file handling. CVE-2020-17103 affects this driver and can allow local privilege escalation.
Cloud Identity	A user or service account used to access cloud services and administrative features. Several reported campaigns focused on compromising these identities to expand access across cloud platforms.
Conditional Access Policy	A policy used to control when and how users can access services. It was recommended as a defense against device code phishing.
Config File / Configuration File	A file that stores application settings and trusted behaviors. Several vulnerabilities in the newsletter involved unsafe processing of configuration files that could enable code execution or trust bypasses.
CountLoader	A loader family delivered through MSHTA-based attack chains. It was used to retrieve and stage commodity information stealers and related payloads.
Credential Stealer / Infostealer	Malware designed to collect usernames, passwords, cookies, autofill data, tokens, or other sensitive information from a system. Multiple entries described this type of malware.
Cryptojacking	Unauthorized use of a system's resources to mine cryptocurrency. One campaign specifically targeted systems with stronger GPUs to increase mining value.
CSC	UAE Cyber Security Counsil
CVE	Common Vulnerabilities and Exposures. It is the standard identifier used to track publicly disclosed security vulnerabilities.
CVSS	Common Vulnerability Scoring System. It is a severity rating framework used to indicate the potential impact and exploitability of a vulnerability.
Data Plane	The part of a service or application that processes normal operational traffic and requests. The NGINX advisories noted that the reported issues affected data plane activity rather than management functions.
Dead-drop	A covert communication technique where commands or data are retrieved indirectly from a third-party service rather than through a direct attacker connection. The malicious extension's Python backdoor used this style of command retrieval.
Device Code Flow	An authentication workflow that allows a device or session to be authorized using a code. It was central to the token-hijacking phishing activity described in the newsletter.
Device Code Phishing	A phishing method that tricks a user into entering an attacker-supplied device code on a legitimate sign-in page, which then authorizes the attacker's access.
DLL Sideloading	A technique where a legitimate program is made to load a malicious DLL from the same location or expected path. It was used in multiple campaigns to start execution without exploiting a software flaw directly.
DNS Tunneling	A method of sending data through DNS requests or responses instead of more obvious network channels. One supply chain payload used it as a backup exfiltration path.
DOM-based XSS	A form of cross-site scripting where the browser's document object model is manipulated so malicious script can run in the user's context. It appeared in the Jira-related vulnerability bulletin.

DoS	Denial of Service. A condition where a system or service becomes unavailable or unstable because it cannot continue processing normally.
Dropper	A first-stage component whose purpose is to install or launch another malware payload. Several campaigns in the newsletter used droppers before the main malicious code executed.
EDR	Endpoint Detection and Response. Security tooling used to monitor, detect, and respond to malicious activity on endpoints. Several campaigns attempted to evade, weaken, or terminate these controls.
EKZ Infostealer	The credential-stealing malware delivered through the FortiClient EMS exploitation chain. It collects browser credentials, cookies, and autofill data from infected systems.
Emmenhtal Loader	A multi-stage loader referenced in the MSHTA-related reporting. It used remote HTA content and heavily obfuscated script stages to deliver later malware.
EMS	Endpoint Management Server. In the newsletter, this referred to a centralized management system whose trusted workflows were abused to push malicious scripts to managed endpoints.
ETW	Event Tracing for Windows. A telemetry mechanism used for visibility into application and runtime behavior. One campaign attempted to disable ETW through configuration-based manipulation.
Exfiltration	Unauthorized transfer of data out of a system or environment. Many of the campaigns combined collection with outbound movement of files, credentials, or cloud data.
File Inclusion	A weakness that allows a system or application to include unintended files in processing. The Atlassian bulletin referenced file inclusion issues affecting Jira-related products.
FileFiend	A custom uploader tool described in the destructive intrusion report. It was used to collect files and send them to attacker-controlled infrastructure.
Fulcio	A certificate service within the Sigstore ecosystem. It was referenced in the malicious extension analysis as part of the signing and trusted publishing logic embedded in the payload.
Heap Buffer Overflow	A memory corruption issue involving heap memory allocation. It can cause a worker process to crash and, under certain conditions, could support code execution.
HTA	HTML Application. A file type that can carry script content and be executed by MSHTA on Windows. It was used as an early-stage loader format in several malware chains.
HTTP Request Smuggling	A technique that exploits differences in how systems interpret HTTP requests, allowing request boundaries to be misread. It appeared in the Atlassian security bulletin.
IaaS	Infrastructure as a Service. A cloud service model that provides virtualized infrastructure such as virtual machines and storage. It was part of the environment targeted in the cloud intrusion campaign.
IIS	Internet Information Services. A Microsoft web server platform. One malware family in the newsletter specifically targeted IIS environments to manipulate traffic and content.
Impersonation Token	A Windows token used by a thread or process to act under a particular security context. It was part of the exploit discussion around MiniPlasma-related privilege escalation reporting, though the final CVE classification remains a local elevation issue.
Infostealer	A broad term for malware focused on collecting and exporting credentials, cookies, tokens, and other user or system data. Several entries described infostealers delivered through trusted workflows or phishing.
js_fetch_proxy	An NGINX JavaScript directive used to define proxy behavior for ngx.fetch(). In the advisory, unsafe use of client-controlled values in this directive could trigger a heap overflow.
Kali365	The phishing-as-a-service kit described in the newsletter that captures Microsoft 365 access tokens through device code phishing instead of directly stealing passwords.
Key Vault	A secret-storage service used to hold credentials, keys, or other sensitive values. In the cloud intrusion case, attackers pivoted to this resource to obtain secrets and move deeper into the environment.

Lateral Movement	The process of moving from one compromised account, service, or system to others inside the same environment. The cloud intrusion and remote access campaigns showed this pattern.
LaunchAgent	A macOS persistence mechanism that causes software to run automatically. In the extension incident, it was used by the backdoor written after the main payload executed.
LPE	Local Privilege Escalation. A technique or vulnerability that lets a user or process gain higher privileges on a system after some local execution already exists.
LummaStealer	A commodity information stealer referenced in the MSHTA delivery chains. It is focused on stealing credentials, browser data, and session-related information.
MaaS	Malware-as-a-Service. A criminal model where malware or supporting tools are sold or shared so multiple actors can operate them. The BadIIS ecosystem was assessed as fitting this model.
Memory Corruption	A class of flaw where memory is handled unsafely, causing instability or potentially enabling code execution. It was central to several NGINX vulnerabilities described in the newsletter.
MFA	Multi-Factor Authentication. A security control that requires more than one verification factor during sign-in. Attackers repeatedly worked around MFA through token theft, approval abuse, or session reuse.
MiniJunk V2	A second malware family referenced in the same Screening Serpens campaign. It represents an updated remote access capability used alongside other intrusion tooling.
MiniPlasma	The name used in the newsletter title for the Windows Cloud Files privilege escalation weakness tied to CVE-2020-17103. It refers to the local privilege escalation issue affecting the Cloud Files Mini Filter Driver.
MiniUpdate	A malware family identified in the Screening Serpens reporting. It was used as part of a staged intrusion chain and supported file theft and remote command execution.
MSHTA	Microsoft HTML Application Host. A legacy Windows utility that can run HTA files and embedded scripts. It was used as a trusted system binary in multiple malware delivery chains.
NGINX JavaScript (njs)	A JavaScript module for NGINX that allows request-handling logic to run scripts. A critical vulnerability affected this module when unsafe client-controlled values were used in proxy URL construction.
ngx.fetch()	A fetch operation used in NGINX JavaScript to make outbound requests. In the reported vulnerability, it became dangerous when fed with client-controlled values through vulnerable configuration.
ngx_http_rewrite_module	An NGINX module used to rewrite URLs and arguments during request handling. One of the critical NGINX vulnerabilities affected this component.
Nimbus RAT	The Java-based remote access trojan delivered after Teams-based social engineering and remote support abuse. It gave attackers remote control and credential theft capability on compromised endpoints.
njs	NGINX JavaScript. A scripting module for NGINX that can be used to run JavaScript within request handling logic. One critical vulnerability affected this component when js_fetch_proxy used client-controlled variables.
OAuth Token	A token that grants access to a service after authorization. In the newsletter, attackers captured access and refresh tokens to maintain service access without repeatedly using passwords.
OIDC	OpenID Connect. It is an identity and token framework used in modern authentication and trusted publishing workflows; the malicious extension targeted OIDC-related publishing paths and tokens.
On-Premise	A deployment model where software runs inside the organization's own environment rather than as a hosted service. This distinction mattered in the exploited Apex One vulnerability entry.
OpenVSX	An alternative extension marketplace for VS Code-compatible editors. The advisory noted that the compromised extension was also briefly available there.

Origin Validation Error	A weakness where software does not correctly verify the source or origin of an action or request. In the Trend Micro entry, several local privilege escalation issues were described this way.
Orphan Commit	A source code commit that is not reachable through normal project branches. In the extension supply chain incident, the malicious payload was fetched from such a hidden commit inside the official repository.
OverlayPhantom	The name given to the Android banking malware campaign described in the newsletter. It uses deceptive app delivery, fake overlays, and screen monitoring to steal financial credentials and other sensitive information from mobile users.
PaaS	Platform as a Service. A cloud service model that provides managed application platforms. The cloud intrusion activity specifically expanded from identities into PaaS resources.
Patch Bypass	A condition where a fix is incomplete or can be circumvented, leaving a vulnerability effectively exploitable. This was explicitly noted in the Notepad++ security advisory for version 8.9.6.1.
Path Traversal	A flaw where crafted path input can escape the intended directory or trust boundary, allowing access to unintended locations or execution paths. It was central to the Notepad++ bypass advisory.
PCRE	Perl-Compatible Regular Expressions. These are advanced pattern-matching expressions used in software configurations; the NGINX vulnerability involved unsafe handling of overlapping PCRE capture groups in rewrite rules.
Persistence	A method attackers use to maintain access after initial compromise, including across reboots or service restarts. Many entries described persistence through scheduled tasks, backdoors, or service-based installers.
Phantom Killer	The name used in the newsletter for a technique that abuses a signed driver to terminate processes from user mode, including security-related processes.
Phishing	A social engineering technique that tricks users into taking unsafe actions, such as clicking a link, opening a file, or authorizing access. It was the starting point for multiple campaigns in the newsletter.
PoC	Proof of Concept. A demonstration showing how a vulnerability or technique can be exercised in practice. The MiniPlasma entry specifically noted public attention around a PoC.
Process Hollowing	A technique where a legitimate process is started and then replaced or injected with malicious code so the malware runs under a trusted process identity. It appeared in both malware delivery and mining campaigns.
Provenance	Evidence showing how software was built and where it came from. In the extension incident, the stolen publishing and signing paths raised concern that malicious software could appear legitimate.
Proxy URL	A URL used by a proxy feature or workflow to forward traffic. In the NGINX JavaScript flaw, improper use of client-controlled values in a proxy URL led to a heap overflow condition.
PureLogs	A fileless information-stealing malware variant delivered through a JavaScript, PowerShell, and process hollowing chain. It was used to collect credentials and other sensitive data from compromised Windows devices.
PurpleFox	A persistent malware threat referenced in the MSHTA analysis as part of the range of malware families that still use MSHTA in execution chains.
Quick Assist	A built-in remote assistance tool on Windows. Attackers used it after Teams-based vishing to gain hands-on access to a victim system.
RAT	Remote Access Trojan. Malware that gives an attacker remote control over an infected system or device. Several entries described RAT delivery through phishing, social engineering, or staged loaders.
RBAC	Role-Based Access Control. A permissions model that grants access based on assigned roles. In the cloud intrusion case, privileged RBAC assignments allowed deeper access into cloud resources.

RCE	Remote Code Execution. A condition where an attacker can cause code to run on a target system from a remote position. Several product vulnerabilities in the newsletter carried this risk.
Registry Run Key	A Windows autostart location used to launch programs automatically when a user signs in. Malware used these keys as one of several persistence methods.
Rekor	A transparency log service within the Sigstore ecosystem. It records signing information so that published software provenance can be verified later.
Relative Path Traversal	A form of path traversal where relative directory changes are used to reach or modify unintended files. It was the exploited vulnerability type reported for Trend Micro Apex One On-Premise.
Remote Support Tool	A legitimate assistance or administration utility that can be abused by attackers to gain interactive control of a device. Quick Assist and ScreenConnect were examples in the newsletter.
Reverse Proxy	A proxy function that retrieves content from one source and serves it onward as if it were local. BadIIS supported this to manipulate how search engines view compromised sites.
Rewrite Directive	A web server rule that changes a requested URL or its parameters before processing continues. In the NGINX advisory, a specific rewrite pattern could lead to a heap buffer overflow.
SaaS	Software as a Service. A cloud service model where the application is delivered as a hosted service. The cloud intrusion campaign began with SaaS-focused data theft before expanding further.
Scheduled Task	A Windows mechanism used to launch programs at certain times or system events. Multiple campaigns used scheduled tasks for persistence or staged execution.
Screen Streaming	Real-time or near-real-time transmission of screen content from a compromised device to an attacker. OverlayPhantom used this to observe on-screen activity.
ScreenConnect	A legitimate remote monitoring and management tool that attackers abused for persistent access and follow-on activity. It appeared in both intrusion and cryptojacking-related reporting.
Screening Serpens	The threat group name used in the newsletter for the actor behind targeted espionage campaigns that used tailored lures, staged payloads, and remote access trojans.
SEO Fraud	Search Engine Optimization fraud. It refers to manipulating search ranking or indexing to misdirect traffic or inflate the visibility of malicious or illicit content.
SEO Poisoning	A technique where search results are manipulated so malicious sites rank or appear more convincingly to users searching for software or content. It was used to steer victims toward fake utility downloads.
Session Cookie	A browser-stored value that can keep a user authenticated to a service. Theft of session cookies may allow attackers to reuse active sessions without triggering another sign-in challenge.
Sigstore	A software signing ecosystem used to verify software origin and integrity. The trojanized extension payload included logic related to Sigstore certificate and signing workflows.
SLSA	Supply-chain Levels for Software Artifacts. It is a framework for software build provenance and integrity assurance, referenced in the malicious extension's signing-related logic.
Social Engineering	The use of deception or trust-based manipulation to make a target take an unsafe action. It appeared across many campaigns, including phishing, vishing, fake updates, and fraudulent support or recruitment themes.
SSPR	Self-Service Password Reset. A user-facing account recovery process. In the cloud intrusion campaign, attackers abused this process and related MFA approval flows to gain persistent access.
Storm-2949	The identifier used in the report for the actor behind a multi-layer cloud intrusion that began with identity compromise and expanded into cloud applications, secrets, storage, and virtual machines.

Supply Chain Attack	A compromise that uses trusted software, updates, packages, extensions, or related workflows to reach victims indirectly. The trojanized developer extension incident is a clear example.
SYSTEM Privileges	The highest local privilege level on Windows. A successful privilege escalation to SYSTEM gives an attacker broad control over the machine.
TeamPCP	The name referenced in follow-on reporting about the trojanized developer extension incident. It is associated with the operation that led to broader developer environment and repository exposure.
Teams Vishing	Voice phishing conducted through Microsoft Teams. Attackers used external Teams messages and calls to impersonate helpdesk staff and persuade users to grant remote access.
Time-of-Check Time-of-Use (TOCTOU)	A flaw where a condition changes between the moment it is checked and the moment it is used, allowing unsafe behavior. This vulnerability type was listed among the Trend Micro local privilege escalation issues.
TLS	Transport Layer Security. A protocol used to secure data in transit. One malware family pinned a TLS certificate to help protect or verify its command-and-control connection.
TOCTOU	Time-of-Check Time-of-Use. This is a flaw where a condition changes between the moment it is checked and the moment it is used, creating an opportunity for unsafe behavior or privilege escalation.
Token Hijacking	Abuse or theft of access tokens so an attacker can use a service without needing the original password again. This was a major feature of the device code phishing activity.
Traffic Redirection	The forced rerouting of user or crawler traffic away from the expected destination to an attacker-controlled or illicit destination. This was a core BadIIS capability.
Trusted Directory	A file path location treated by software as safer or approved for execution. The Notepad++ bypass occurred because trust was checked before the path was normalized correctly.
Trusted Directory Check	A safety control that treats certain file paths as allowed or less risky. In the Notepad++ advisory, a bypass occurred because path normalization was not done before this trust check.
Vishing	Voice phishing. A phone or voice-based social engineering approach that persuades a victim to trust an attacker or perform an unsafe action.
VSIX	The package format used for Visual Studio Code extensions. In the trojanized extension case, a malicious extension package served as the trigger for the larger credential-stealing payload chain.
WAF	Web Application Firewall. A protective control used to inspect and filter malicious or malformed web requests. It was recommended as a mitigation for exposed NGINX environments.
Worker Process	A service process that handles active traffic or application work. Several vulnerabilities in the newsletter described crafted input causing the worker process to restart or crash.
XSS	Cross-Site Scripting. A web vulnerability where malicious script can run in a user's browser context. The Atlassian bulletin and related dependency references included this issue type.
Zero-Day	A vulnerability that becomes known before an official fix is available or before defenders are broadly prepared. The newsletter referenced newly disclosed and still-developing high-risk exposures in this category.