INTELLIGENCE BRIEFING
Security Command Center
TLP:CLEAR
2026-05-29 14:00 UTC

Kimsuky Expands Operational Toolkit: LLM-Assisted Malware, VS Code Tunneling, and Real-Time Infection Verification Mark Tactical Shift

THREAT CAMPAIGN | **HIGH** | CVSS 7.5

SCC Item ID	SCC-CAM-2026-0378
Type	Threat Campaign
Severity	HIGH
CVSS Base Score	7.5
Affected Products	Microsoft Visual Studio Code (Remote Tunneling feature), Cloudflare Quick Tunnels, Cisco Webex (spoofed installer), nProtect Online Security (spoofed), AhnLab Safe Transaction (spoofed), DWAgent (legitimate RAT abused), South Korean B2B messaging platforms
Published	2026-05-29T01:57:41
Discovery Source	Rss

Executive Summary

Kimsuky, a North Korean state-sponsored threat group, conducted targeted intrusion campaigns against South Korean military, corporate, and government organizations between March and April 2026. The group introduced two new malware families and adopted tactics that hide malicious traffic inside legitimate tools, Microsoft VS Code, Cloudflare tunnels, and spoofed installers for trusted South Korean software, making detection significantly harder with standard controls. Organizations with South Korean business ties, defense-industrial supply chains, or operations in the region face elevated risk of undetected, long-dwell intrusions.

Technical Analysis

Kimsuky (MITRE G0094, aka Velvet Chollima) deployed two new malware families, HelloDoor and HttpMalice, via updated HTTPSpy infection chains during March-April 2026. Initial access vectors include spearphishing links (T1566.002) and malicious attachments (T1566.001) delivering spoofed installers mimicking Cisco Webex, nProtect Online Security, and AhnLab Safe Transaction (T1036.005, T1204.002). Post-exploitation activity leverages Microsoft VS Code Remote Tunneling and Cloudflare Quick Tunnels as covert C2 channels (T1572, T1090.003, T1102), effectively routing adversary traffic through trusted infrastructure. A novel 'JSONPing'

technique (T1082 adjacent) provides operators real-time confirmation of successful host compromise before progressing the intrusion. DWAgent, a legitimate commercial RAT, is deployed to blend with administrative traffic (T1219). New malware variants written in Rust carry code structure patterns consistent with LLM-assisted development, suspected based on code entropy and function naming patterns; technical IOC confirmation pending reverse engineering and primary vendor review. Additional TTPs include keylogging (T1056.001), screen capture (T1113), scheduled task persistence (T1053.005), system information discovery (T1016, T1082), and reflective code loading (T1620). Relevant CWEs: CWE-912 (Hidden Functionality), CWE-494 (Download of Code Without Integrity Check), CWE-506 (Embedded Malicious Code), CWE-356 (Product UI does not Warn User of Unsafe Actions). No CVE is assigned to this campaign. Source confidence (0.56) reflects T3 source distribution and pending IOC confirmation from primary threat intelligence vendors.

Action Checklist

1. Step 1: Containment, Identify any VS Code Remote Tunneling activity in your environment. Audit VS Code installations across endpoints (CIS 2.1) and block outbound connections to vscode.dev, tunnel.azurefd.net, and Cloudflare Quick Tunnel domains (*.trycloudflare.com) at the perimeter firewall (CIS 4.4, CIS 4.5, NIST AC-4). If developers require legitimate tunnel access, implement allowlisting for authorized user accounts and hosts. Suspend use of DWAgent unless operationally required, and revoke any active DWAgent sessions pending review.
2. Step 2: Detection, Query endpoint logs for VS Code processes spawning network connections to tunnel endpoints (T1572). Search email gateway logs for spoofed Webex, nProtect, or AhnLab installer attachments (T1566.001/T1566.002). Review SIEM for scheduled task creation by non-standard processes (T1053.005) and outbound HTTP POST activity with JSON-formatted beacons to unknown external hosts. Hunt for repeated, low-volume outbound HTTP POST requests with JSON-formatted bodies (e.g., {"status":"alive"}) typically occurring shortly after installer execution or at predictable intervals (JSONPing pattern, T1082). Enable audit logging across all endpoints per NIST AU-2 and CIS 8.2 if not already active. Hunt for Rust-compiled executables with high entropy in user-writable directories (T1027).
3. Step 3: Eradication, Remove any identified HelloDoor, HttpMalice, or HTTPSpy artifacts from affected hosts. Revoke and rotate credentials for all accounts active on compromised or suspected hosts (D3-CRO). Disable VS Code Remote Tunneling via Group Policy or endpoint management where not required for operations. Block DWAgent installation packages via application allowlisting (NIST CM-7, CIS 2.3). Remove unauthorized scheduled tasks and confirm no persistence mechanisms remain (T1053.005).
4. Step 4: Recovery, Re-image confirmed compromised hosts before returning to production. Validate endpoint integrity using system file analysis (D3-SFA) post-remediation. Confirm tunnel-blocking rules are enforced and logging is active (NIST AU-12, CIS 8.2). Monitor for re-infection attempts targeting the same user accounts or asset classes for a minimum of 30 days post-containment. Require MFA for all remote access and administrative accounts before restoring access (CIS 6.4, CIS 6.5, D3-MFA).
5. Step 5: Post-Incident, Conduct a tabletop or after-action review focused on detection gaps for living-off-the-land and legitimate-tool abuse. Assess whether existing detection rules cover commercial RAT traffic (DWAgent) and developer tunnel protocols. Implement software installation controls to prevent unapproved RAT and tunnel tool deployment (NIST CM-7, CIS 4.6). Review and harden user account permissions to limit blast radius of credential theft (NIST AC-6, D3-UAP). Brief threat intelligence team on Kimsuky TTPs and update hunt hypotheses to include LLM-assisted malware indicators.

IR / Forensic Enrichment

Triage Priority	IMMEDIATE
Escalation Criteria	Escalate immediately to senior IR leadership and legal/compliance if any confirmed HelloDoor, HttpMalice, or HTTPSpy execution is identified on hosts with access to defense, government contract, or sensitive corporate data, or if VS Code tunnel sessions show interactive operator activity (keystrokes, lateral movement commands) indicating hands-on-keyboard intrusion by Kimsuky rather than automated beacon activity, as this constitutes a nation-state intrusion with potential regulatory notification obligations and counterintelligence equities.
Recovery Notes	Re-image all hosts with confirmed HelloDoor, HttpMalice, or DWAgent unauthorized deployment — do not attempt in-place remediation given Kimsuky's demonstrated use of multiple persistence mechanisms (scheduled tasks, VS Code tunnel tokens, potential in-memory implants) that may survive partial cleanup. Before restoring any host to production, confirm VS Code tunnel tokens have been revoked via the vscode.dev management console (not just local file deletion), all scheduled tasks have been audited against a clean baseline, and MFA is enforced on the restored accounts. Maintain enhanced monitoring for 30 days minimum targeting the same user accounts and asset classes that were compromised, with specific focus on DNS queries to Cloudflare tunnel infrastructure and new executable drops in user-writable directories, as Kimsuky campaigns have historically involved re-targeting of the same organizations with modified lures after initial detection.
Forensic Artifacts	VS Code tunnel registration token files at %APPDATA%\Code\User\globalStorage\ms-vscode.remote-server\token and associated tunnel configuration JSON — these persist the Kimsuky operator's tunnel registration identity and are required evidence for revoking access via vscode.dev and for attributing the specific tunnel ID to the intrusion timeline | DWAgent session and connection logs at C:\Program Files\DWAgent\log\ and C:\ProgramData\DWAgent\ — these record operator-controlled remote session timestamps, remote IP addresses, and command sequences executed through the RAT, providing direct evidence of hands-on-keyboard activity distinct from automated malware beacon traffic | Rust PE binary artifacts (HelloDoor, HttpMalice, HTTPSpy) in user-writable directories (%APPDATA%\Roaming\, %LOCALAPPDATA%\Temp\, %PUBLIC\) — identifiable pre-hash by high-entropy .rdata sections and 'rustc' compiler version strings embedded in the PE, with file creation timestamps correlated against the intrusion timeline to establish initial access timing | Windows Task Scheduler XML task definitions at C:\Windows\System32\Tasks\ for any tasks created by non-SYSTEM accounts during the intrusion window — these record the exact persistence command line, trigger schedule, and RunAs user context used by Kimsuky and constitute primary evidence of T1053.005 activity | Email gateway logs and quarantined attachment copies of spoofed Cisco-WebexSetup.exe, nProtect, or AhnLab installer lures — including full SMTP headers (source IP, relay chain, envelope sender) and attachment hashes (SHA256) to establish the initial delivery vector, support phishing infrastructure attribution, and enable blocklisting across the organization's email security controls

Per-Action IR Details

Step 1: Containment — Identify any VS Code Remote Tunneling activity in your environment. Audit VS Code installations across endpoints (CIS 2.1) and block outbound connections to vscode.dev, tunnel.azurefd.net, and Cloudflare Quick Tunnel domains (*.trycloudflare.com) at the perimeter firewall (CIS 4.4, CIS 4.5, NIST AC-4). Suspend use of DWAgent unless operationally required, and revoke any active DWAgent sessions

pending review.

NIST Phase: Containment

Reference: NIST 800-61r3 §3.3 — Containment Strategy

Controls: NIST AC-4 (Information Flow Enforcement), NIST IR-4 (Incident Handling), CIS 2.1 (Establish and Maintain a Software Inventory), CIS 4.4 (Implement and Manage a Firewall on Servers), CIS 4.5 (Implement and Manage a Firewall on End-User Devices)

Compensating: Run 'Get-ChildItem -Path C:\Users -Recurse -Filter code.exe -ErrorAction SilentlyContinue' across endpoints via PowerShell remoting or a domain GPO startup script to enumerate VS Code installations without a SIEM. For DWAgent, run 'Get-Process dwagent -ErrorAction SilentlyContinue' and 'Get-Service | Where-Object {$_.DisplayName -like "*DWAgent*"}' to identify active sessions. Use Windows Firewall (netsh advfirewall) or pfSense/OPNsense outbound rules to block *.trycloudflare.com and tunnel.azurefd.net by FQDN; add a DNS sinkhole entry for these domains using Pi-hole or bind RPZ if perimeter firewall lacks FQDN-based blocking.

Evidence: Before blocking tunnel endpoints, capture: (1) NetFlow or firewall session logs showing established connections from workstations to vscode.dev, tunnel.azurefd.net, or *.trycloudflare.com — record source IP, destination IP, duration, and bytes transferred to establish dwell time; (2) DWAgent connection logs at C:\Program Files\DWAgent\log\ or C:\ProgramData\DWAgent\ showing remote session initiation timestamps and remote IP addresses used by the operator; (3) Windows Security Event Log Event ID 7045 (Service Installed) and Event ID 4697 (Service installed in the system) on hosts where DWAgent was deployed without authorization; (4) VS Code tunnel configuration files at %APPDATA%\Code\User\globalStorage\ms-vscode.remote-server\ which persist tunnel registration tokens even after the tunnel session ends.

Step 2: Detection — Query endpoint logs for VS Code processes spawning network connections to tunnel endpoints (T1572). Search email gateway logs for spoofed Webex, nProtect, or AhnLab installer attachments (T1566.001/T1566.002). Review SIEM for scheduled task creation by non-standard processes (T1053.005) and outbound HTTP POST activity with JSON-formatted beacons to unknown external hosts ('JSONPing' pattern, T1082). Enable audit logging across all endpoints per NIST AU-2 and CIS 8.2 if not already active. Hunt for Rust-compiled executables with high entropy in user-writable directories (T1027).

NIST Phase: Detection Analysis

Reference: NIST 800-61r3 §3.2 — Detection and Analysis

Controls: NIST AU-2 (Event Logging), NIST AU-6 (Audit Record Review, Analysis, and Reporting), NIST SI-3 (Malware Protection), CIS 8.2 (Collect Audit Logs)

Compensating: Deploy Sysmon with SwiftOnSecurity's config (minimum) augmented with EventID 3 (Network Connection) filtering on code.exe, node.exe (VS Code's Node runtime), or dwagent.exe connecting to tunnel.azurefd.net or *.trycloudflare.com. Use this PowerShell one-liner to find high-entropy Rust binaries in user-writable paths: 'Get-ChildItem -Path C:\Users,C:\ProgramData -Recurse -Include *.exe,*.dll -ErrorAction SilentlyContinue | Where-Object { $_.Length -gt 100KB }' then pipe to a Get-FileHash check against known-good hashes. For the JSONPing beacon pattern (T1082), capture traffic on the perimeter with Wireshark/tcpdump filtering 'tcp port 80 or 443 and (http.request.method == POST)' and inspect payloads for JSON-formatted system enumeration data (hostname, username, OS version). For scheduled task hunting, run 'Get-ScheduledTask | Where-Object {$_.TaskPath -notlike "\Microsoft*"} | Select-Object TaskName, TaskPath, @{N="Actions";E={$_.Actions.Execute}}' and flag tasks with executables in %TEMP%, %APPDATA%, or %PUBLIC%.

Evidence: Before concluding detection scope: (1) Email gateway logs (MTA headers, attachment hashes) for messages delivering spoofed Cisco-WebexSetup.exe, nProtect_setup.exe, or AhnLab installers — Kimsuky has historically used ISO and ZIP wrappers, so inspect container file types not just attachment extensions; (2) Sysmon Event ID 1 (Process Create) for parent-child chains where code.exe or a spoofed installer spawns cmd.exe, powershell.exe, or mshta.exe — this indicates the VS Code tunnel was used as a command execution channel; (3) Windows Task Scheduler operational log at 'Microsoft-Windows-TaskScheduler/Operational' Event IDs 106 (Task Registered) and 200 (Task Executed) for tasks created by non-SYSTEM, non-administrative accounts; (4) Browser or application download history for delivery of spoofed South Korean B2B platform installer packages, stored in %LOCALAPPDATA%\Microsoft\Windows\INetCache\ or browser profile download directories; (5) File system artifacts of HelloDoor or HttpMalice staging — look for Rust PE executables (identifiable by 'rustc' version strings in the binary

or 'std::' symbol traces) dropped in %APPDATA%\Roaming\, %LOCALAPPDATA%\Temp\, or subdirectories mimicking legitimate software vendor paths.

Step 3: Eradication — Remove any identified HelloDoor, HttpMalice, or HTTPSpy artifacts from affected hosts. Revoke and rotate credentials for all accounts active on compromised or suspected hosts (D3-CRO). Disable VS Code Remote Tunneling via Group Policy or endpoint management where not required for operations. Block DWAgent installation packages via application allowlisting (NIST CM-7, CIS 2.3). Remove unauthorized scheduled tasks and confirm no persistence mechanisms remain (T1053.005).

NIST Phase: Eradication

Reference: NIST 800-61r3 §3.4 — Eradication

Controls: NIST CM-7 (Least Functionality), NIST IA-5 (Authenticator Management), NIST SI-2 (Flaw Remediation), CIS 2.3 (Address Unauthorized Software), CIS 5.1 (Establish and Maintain an Inventory of Accounts)

Compensating: To disable VS Code tunneling without an MDM platform, deploy a GPO that sets the registry value 'HKLM\SOFTWARE\Policies\Microsoft\VSCode\remote.tunnels.access' to 'off', or push a settings.json policy via GPO file preferences to %APPDATA%\Code\User\settings.json adding '"remote.tunnels.access": "off"'. For DWAgent application blocking without a commercial allowlisting tool, use AppLocker (Windows 7+/Server 2008 R2+) with a Publisher rule denying execution of DWAgent-signed binaries, or a Path rule blocking execution from DWAgent's default install directory 'C:\Program Files\DWAgent\'. To verify scheduled task eradication, run 'schtasks /query /fo LIST /v | findstr /i "task name\|run as user\|task to run"' and cross-reference against a known-good baseline captured pre-incident. For credential rotation without a PAM tool, prioritize accounts with any logon event (Event ID 4624) on confirmed compromised hosts within the intrusion window.

Evidence: Before eradication, preserve forensic copies of: (1) Full memory images from hosts where HelloDoor or HttpMalice was active — Kimsuky malware has demonstrated in-memory execution capability and Rust-based loaders may inject into legitimate processes, making disk-only analysis insufficient; use WinPmem or DumpIt for acquisition; (2) Copies of all identified scheduled task XML definitions from C:\Windows\System32\Tasks\ — these record the exact command line, user context, and creation timestamp used by Kimsuky for persistence; (3) Registry export of HKCU\Software\Microsoft\Windows\CurrentVersion\Run and HKLM\SOFTWARE\Microsoft\Windows\CurrentVersion\Run to document any autorun entries added by the malware families; (4) VS Code tunnel registration tokens from %APPDATA%\Code\User\globalStorage\ms-vscode.remote-server\token — these tokens allow Kimsuky to re-establish tunnel access even if the VS Code binary is removed, so token revocation through the vscode.dev dashboard must accompany local file deletion; (5) DWAgent session logs and any operator-configured remote access credentials stored by DWAgent in its local configuration database before service termination.

Step 4: Recovery — Re-image confirmed compromised hosts before returning to production. Validate endpoint integrity using system file analysis (D3-SFA) post-remediation. Confirm tunnel-blocking rules are enforced and logging is active (NIST AU-12, CIS 8.2). Monitor for re-infection attempts targeting the same user accounts or asset classes for a minimum of 30 days post-containment. Require MFA for all remote access and administrative accounts before restoring access (CIS 6.4, CIS 6.5, D3-MFA).

NIST Phase: Recovery

Reference: NIST 800-61r3 §3.5 — Recovery

Controls: NIST AU-12 (Audit Record Generation), NIST CP-10 (System Recovery and Reconstitution), NIST IA-2 (Identification and Authentication — Organizational Users), CIS 6.4 (Require MFA for Remote Network Access), CIS 6.5 (Require MFA for Administrative Access), CIS 8.2 (Collect Audit Logs)

Compensating: For system file integrity validation without a commercial tool, run 'sfc /scannow' on Windows hosts post-reimaging and cross-reference installed software hashes against vendor-published checksums using Get-FileHash (SHA256). To confirm tunnel-blocking is enforced without a SIEM, configure Windows Event Forwarding (WEF) to collect Sysmon Event ID 3 (Network Connection Blocked) from all endpoints to a central WEC server — this requires only Windows infrastructure and provides visibility into blocked tunnel connection attempts. For 30-day re-infection monitoring without EDR, deploy a scheduled PowerShell task on recovered hosts that runs daily, checks for new executables in user-writable directories (compared against a hash baseline), and emails results to the SOC

mailbox. Enforce MFA for RDP and VPN using Windows Hello for Business or Duo's free tier before restoring any account access.

Evidence: Before returning hosts to production, verify: (1) Firewall rule hit counters on the blocks for vscode.dev, tunnel.azurefd.net, and *.trycloudflare.com — a non-zero hit count post-reimaging on a supposedly clean host indicates re-infection or a missed compromised system attempting callback; (2) Active Directory logon audit (Event ID 4624, logon type 3 and 10) for accounts that were active on compromised hosts, reviewed for any logon events from unexpected source IPs or at unusual hours during the recovery window, indicating credential reuse by Kimsuky operators; (3) DNS query logs from the internal resolver for queries to Cloudflare tunnel subdomains or vscode.dev originating post-remediation — Kimsuky's use of legitimate infrastructure means malware callbacks are indistinguishable from legitimate developer traffic without DNS-layer visibility; (4) Email gateway quarantine logs for any redelivery attempts of spoofed Webex, nProtect, or AhnLab installer lures targeting the same users who were originally phished, indicating continued targeting of the organization.

Step 5: Post-Incident — Conduct a tabletop or after-action review focused on detection gaps for living-off-the-land and legitimate-tool abuse. Assess whether existing detection rules cover commercial RAT traffic (DWAgent) and developer tunnel protocols. Implement software installation controls to prevent unapproved RAT and tunnel tool deployment (NIST CM-7, CIS 4.6). Review and harden user account permissions to limit blast radius of credential theft (NIST AC-6, D3-UAP). Brief threat intelligence team on Kimsuky TTPs and update hunt hypotheses to include LLM-assisted malware indicators.

NIST Phase: Post Incident

Reference: NIST 800-61r3 §4 — Post-Incident Activity

Controls: NIST AC-6 (Least Privilege), NIST CM-7 (Least Functionality), NIST IR-4 (Incident Handling), NIST RA-3 (Risk Assessment), CIS 4.6 (Securely Manage Enterprise Assets and Software)

Compensating: Write Sigma rules targeting the specific Kimsuky detection gaps identified: (1) a rule matching Sysmon EID 3 for code.exe or node.exe connecting to tunnel.azurefd.net or *.trycloudflare.com; (2) a rule matching Sysmon EID 1 for DWAgent.exe spawning child processes; (3) a rule matching Windows EID 4698 (Scheduled Task Created) where the task action path contains %APPDATA%, %TEMP%, or %PUBLIC%. Publish these to your internal Sigma repository and test against the logs collected during this incident. For LLM-assisted malware hunting, add YARA rules scanning for Rust PE artifacts with obfuscated string tables (high entropy .rdata sections) combined with network-capable API imports (WinINet, WinHTTP) — this targets the HelloDoor/HttpMalice profile without requiring known-bad hashes. Use osquery to enforce a scheduled query detecting new services or scheduled tasks created in the last 24 hours that run from non-standard paths.

Evidence: For the after-action record, preserve and document: (1) The complete set of Kimsuky TTPs observed in this intrusion mapped to MITRE ATT&CK — specifically T1572 (Protocol Tunneling via VS Code), T1566.001/T1566.002 (Spearphishing with spoofed South Korean software lures), T1053.005 (Scheduled Task persistence), T1082 (System Information Discovery via JSONPing beacons), and T1027 (Obfuscated Rust-compiled payloads) — as the structured threat intelligence input for updating hunt hypotheses; (2) Samples of HelloDoor, HttpMalice, or HTTPSpy binaries (if recovered) submitted to an internal or trusted sandbox for behavioral analysis, with the resulting IOC set (C2 IPs, URI patterns, mutex names, registry artifacts) formalized as detection rules; (3) The full timeline of initial access through discovery through C2 establishment, reconstructed from the log evidence collected in Steps 1–4, to identify the precise detection gap duration and inform the tabletop scenario design; (4) Documentation of which user accounts and asset classes were targeted, to inform the threat intelligence team's assessment of whether Kimsuky's targeting of South Korean military, corporate, and government entities extends to this organization's specific sector and whether re-targeting is likely.

Detection Guidance

Priority detection signals for this campaign: (1) VS Code Remote Tunnel abuse, alert on VS Code processes (code.exe, code-tunnel.exe) establishing outbound connections to vscode.dev, *.vscode-cdn.net, or *.azurefd.net from non-developer endpoints; cross-reference with asset inventory (CIS 1.1) to flag anomalous

hosts. (2) Cloudflare Quick Tunnel C2, block and alert on DNS queries or outbound TLS to *.trycloudflare.com from non-developer endpoints; organizations with authorized developer tunnel use should implement allowlisting for approved users/hosts and monitor for anomalies (T1090.003). (3) JSONPing verification, hunt for repeated, low-volume outbound HTTP POST requests with JSON-formatted bodies (e.g., {"status":"alive"} or similar status indicators) to unrecognized external IPs, typically occurring shortly after installer execution or at regular intervals. (4) Spoofed installer delivery, inspect email gateway logs for ZIP or executable attachments with names containing 'Webex', 'nProtect', or 'AhnLab' from external senders; validate file hash against vendor-published checksums (D3-FMBV). (5) DWAgent deployment, alert on dwagsvc.exe or dwagent.exe processes not present in authorized software inventory (CIS 2.1); treat as presumptively malicious absent documented business justification. (6) Scheduled task persistence, audit Windows Task Scheduler for tasks created by user-context processes with encoded or obfuscated command lines (Event ID 4698). (7) Rust binary hunting, flag high-entropy executables in %APPDATA%, %TEMP%, and user-writable paths lacking valid code signatures. Local account monitoring (D3-LAM) should flag new accounts or privilege changes on affected hosts. Logging must meet NIST AU-2 and AU-3 standards for these signals to be actionable.

Indicators of Compromise

Type	Value	Context	Confidence
DOMAIN	`*.trycloudflare.com`	Cloudflare Quick Tunnel domains abused as covert C2 channels by Kimsuky in this campaign	MEDIUM
DOMAIN	`vscode.dev`	VS Code Remote Tunneling infrastructure abused for C2; legitimate Microsoft domain repurposed — block tunnel-specific subpaths rather than the root domain where operationally feasible	MEDIUM
URL	`https://thehackernews.com/2026/05/kimsuky-deploys-httpspy-expands-arsenal.html`	Primary campaign reporting source — T3, human validation recommended	LOW
URL	`https://www.darktrace.com/blog/darktrace-identifies-campaign-targeting-south-korea-leveraging-vs-code-for-remote-access`	Darktrace campaign analysis covering VS Code tunnel abuse — T3, human validation recommended	LOW

Framework Mappings

MITRE-ATTACK

- **T1218.010** — Regsvr32
- **T1027** — Obfuscated Files or Information
- **T1027.010** — Command Obfuscation
- **T1082** — System Information Discovery
- **T1572** — Protocol Tunneling

- **T1105** — Ingress Tool Transfer
- **T1113** — Screen Capture
- **T1566.002** — Spearphishing Link
- **T1218** — System Binary Proxy Execution
- **T1497.001** — System Checks
- **T1059.005** — Visual Basic
- **T1219** — Remote Access Tools
- **T1566.001** — Spearphishing Attachment
- **T1053.005** — Scheduled Task
- **T1059** — Command and Scripting Interpreter
- **T1552.004** — Private Keys
- **T1102** — Web Service
- **T1090.002** — External Proxy
- **T1041** — Exfiltration Over C2 Channel
- **T1059.001** — PowerShell
- **T1056.001** — Keylogging
- **T1497** — Virtualization/Sandbox Evasion
- **T1016** — System Network Configuration Discovery
- **T1071.001** — Web Protocols
- **T1090.003** — Multi-hop Proxy
- **T1620** — Reflective Code Loading
- **T1036.005** — Match Legitimate Resource Name or Location
- **T1204.002** — Malicious File

NIST-800-53R5

- **SI-3** — Malicious Code Protection
- **SI-4** — System Monitoring
- **CA-7** — Continuous Monitoring
- **SC-7** — Boundary Protection
- **AT-2** — Literacy Training and Awareness
- **SI-8** — Spam Protection
- **CM-7** — Least Functionality
- **SI-7** — Software, Firmware, and Information Integrity
- **CM-3** — Configuration Change Control

OWASP-TOP10-2021

- **A08:2021** — Software and Data Integrity Failures

CIS-V8

- **2.5** — Allowlist Authorized Software

- **2.6** — Allowlist Authorized Libraries
- **8.2** — Collect Audit Logs

NIST-CSF-2

- **DE.CM-01** — Networks and network services are monitored

ISO-27001-2022

- **A.5.23** — Information security for use of cloud services

MITRE ATT&CK Mapping

Technique ID	Technique Name	Tactic
T1218.010	Regsvr32	Defense-Evasion
T1027	Obfuscated Files or Information	Defense-Evasion
T1027.010	Command Obfuscation	Defense-Evasion
T1082	System Information Discovery	Discovery
T1572	Protocol Tunneling	Command-And-Control
T1105	Ingress Tool Transfer	Command-And-Control
T1113	Screen Capture	Collection
T1566.002	Spearphishing Link	Initial-Access
T1218	System Binary Proxy Execution	Defense-Evasion
T1497.001	System Checks	Defense-Evasion
T1059.005	Visual Basic	Execution
T1219	Remote Access Tools	Command-And-Control
T1566.001	Spearphishing Attachment	Initial-Access
T1053.005	Scheduled Task	Execution
T1059	Command and Scripting Interpreter	Execution
T1552.004	Private Keys	Credential-Access
T1102	Web Service	Command-And-Control
T1090.002	External Proxy	Command-And-Control
T1041	Exfiltration Over C2 Channel	Exfiltration
T1059.001	PowerShell	Execution

Technique ID	Technique Name	Tactic
T1056.001	Keylogging	Collection
T1497	Virtualization/Sandbox Evasion	Defense-Evasion
T1016	System Network Configuration Discovery	Discovery
T1071.001	Web Protocols	Command-And-Control
T1090.003	Multi-hop Proxy	Command-And-Control
T1620	Reflective Code Loading	Defense-Evasion
T1036.005	Match Legitimate Resource Name or Location	Defense-Evasion
T1204.002	Malicious File	Execution

Sources

Source	URL	Tier
Security News	https://thehackernews.com/2026/05/kimsuky-deploys-httpspy-expands-a...	T3
Darktrace Identifies Campaign Targeting South Korea Leveraging ...	https://www.darktrace.com/blog/darktrace-identifies-campaign-target...	T3
Cisco Webex App Client-Side Remote Code Execution Vulnerability	https://sec.cloudapps.cisco.com/security/center/content/CiscoSecuri...	T3
Zero Day Initiative — The January 2025 Security Update Review	https://www.thezdi.com/blog/2025/1/14/the-january-2025-security-upd...	T3
Patch Tuesday Update - October 2024 | Vulnerability Research Team	https://www.fortra.com/blog/patch-tuesday-1-update-october-2024	T3

Generated 2026-05-29 14:00 UTC by TJS Security Command Center