INTELLIGENCE BRIEFING
Security Command Center
TLP:CLEAR
2026-04-24 13:42 UTC

Tropic Trooper Shifts to AdaptixC2, Abuses GitHub and VS Code Tunnels to Evade Enterprise Detection

THREAT CAMPAIGN | **HIGH** | CVSS 7.5

SCC Item ID	SCC-CAM-2026-0215
Type	Threat Campaign
Severity	HIGH
CVSS Base Score	7.5
Affected Products	SumatraPDF (trojanized installer), Microsoft Visual Studio Code (tunnel feature), GitHub (as C2 relay infrastructure)
Published	2026-04-24T05:29:00
Discovery Source	Rss

Executive Summary

Tropic Trooper (APT23), a long-established state-aligned threat group, has upgraded its attack toolkit and is now delivering malware through trojanized SumatraPDF installers, routing command-and-control traffic through GitHub and Microsoft VS Code tunnels, developer services that most enterprise firewalls treat as trusted. The campaign targets Chinese-speaking individuals in Taiwan, South Korea, and Japan, with lures themed around military content. The deliberate use of legitimate developer infrastructure makes this campaign significantly harder to detect through conventional network monitoring, increasing the risk of sustained, undetected access to compromised environments.

Technical Analysis

Tropic Trooper has retired Cobalt Strike and Mythic Merlin in favor of AdaptixC2 Beacon as its primary post-exploitation framework. Initial access is achieved via trojanized SumatraPDF installers (CWE-506: Embedded Malicious Code) distributed inside military-themed ZIP archives (T1566.001 spearphishing attachment, T1204.002 malicious file execution). The installer does not verify code integrity before execution (CWE-494). Post-compromise, the implant establishes C2 via GitHub repositories (T1102, T1102.002 bidirectional communication through web services) and Microsoft VS Code remote tunnel feature (T1021.005, T1572 protocol tunneling), both of which are commonly allowlisted in enterprise environments. Additional techniques include obfuscation (T1027), scripting interpreter execution (T1059), ingress tool transfer (T1105), masquerading (T1036.005), and supply chain-adjacent installer trojanization (T1195.002). No CVE identifier is

confirmed for this campaign. A SumatraPDF buffer overflow (CVE-2026-25920, referenced in source metadata but not independently verified as exploited in this campaign) may be relevant to installer trojanization mechanisms; treat this reference as preliminary pending primary-source confirmation. CVSS base score of 7.5 is campaign-assigned, not CVE-derived. Source quality score is 0.64 (T3 sources); treat technical specifics as preliminary pending primary-source confirmation.

Action Checklist

1. Step 1: Containment, Identify any SumatraPDF installations across the environment; flag versions not deployed via verified internal software distribution. Isolate hosts running unsigned or unverified SumatraPDF binaries. Block outbound connections to GitHub raw content endpoints and VS Code tunnel relay domains (*.vscode-cdn.net, tunnels.api.visualstudio.com) at the perimeter for hosts that have no legitimate developer use case. Prioritize endpoints used by employees in Taiwan, South Korea, or Japan given the campaign's targeting profile.
2. Step 2: Detection, Query EDR telemetry for SumatraPDF processes spawning child processes (cmd.exe, powershell.exe, wscript.exe). Search proxy and firewall logs for sustained outbound sessions to github.com or VS Code tunnel relay domains from non-developer hosts. Look for AdaptixC2 Beacon network signatures if your NDR vendor has released them. Review Windows Event ID 4688 (process creation) and Sysmon Event ID 1 for sumatrapdf.exe with anomalous parent/child relationships. Alert on VS Code CLI invocations (code tunnel) on hosts where VS Code is not an approved application.
3. Step 3: Eradication, Remove any trojanized SumatraPDF installer or binary. Deploy only the official SumatraPDF build from sumatrapdfreader.org, verified by file hash against the publisher's published checksums. Revoke any VS Code tunnel sessions not authorized by IT. Rotate credentials for any account that authenticated from a suspected compromised host. If AdaptixC2 Beacon persistence mechanisms are identified (registry run keys, scheduled tasks, service installation), remove each persistence artifact and confirm removal via EDR.
4. Step 4: Recovery, After remediation, monitor previously affected hosts for 30 days for re-beaconing to GitHub or VS Code tunnel endpoints. Validate that no new unauthorized scheduled tasks, services, or registry run entries have appeared. Confirm SumatraPDF binary hashes match the official publisher release on all endpoints. Re-scan affected hosts with updated threat signatures once EDR vendors publish AdaptixC2 Beacon detections.
5. Step 5: Post-Incident, Review allowlist policies for developer services (GitHub, VS Code tunnels, similar living-off-trusted-sites infrastructure); implement context-aware controls that restrict these services to hosts with a documented developer role. Evaluate whether software installation policies prevent users from executing unsigned or ZIP-delivered installers. Map the gap against MITRE ATT&CK T1102 (Web Service C2) and T1572 (Protocol Tunneling) and assess whether existing detection coverage addresses both techniques. Update threat hunt hypotheses to include living-off-trusted-sites (LoTS) C2 patterns.

IR / Forensic Enrichment

Triage Priority	URGENT

Escalation Criteria	Escalate immediately to senior IR leadership and legal/compliance if AdaptixC2 Beacon activity is confirmed on hosts processing PII, PHI, or regulated data, or if evidence indicates lateral movement beyond the initially identified host — given Tropic Trooper's state-aligned attribution, national security or export control reporting obligations may also apply depending on the organization's sector and jurisdiction.
Recovery Notes	After eradication, maintain elevated monitoring on all recovered hosts for a minimum of 30 days, with specific focus on outbound HTTPS sessions to github.com and *.vscode-cdn.net, as AdaptixC2 Beacon operators may attempt to re-establish persistence using a different delivery lure or a new trojanized binary once they detect their primary foothold has been removed. Re-validate SumatraPDF binary integrity weekly against publisher-published SHA-256 checksums and confirm no new VS Code tunnel OAuth authorizations appear on accounts associated with the incident. Conduct a full credential reset and review for all accounts confirmed to have been active on compromised hosts before returning those endpoints to production use.
Forensic Artifacts	Trojanized SumatraPDF binary: full disk image or quarantined copy with SHA-256 hash for comparison against official sumatrapdfreader.org publisher checksums — the malicious installer will fail signature validation via Get-AuthenticodeSignature or sigcheck.exe (Sysinternals) | AdaptixC2 Beacon in-memory artifacts: WinPmem or Magnet RAM Capture memory dump from suspected host, to be analyzed with Volatility3 for injected PE segments, hollowed processes, or reflectively loaded DLLs associated with the Beacon payload dropped by the trojanized SumatraPDF installer | VS Code tunnel OAuth authorization records: GitHub account settings at github.com/settings/connections showing VS Code tunnel app authorizations from non-developer accounts, along with GitHub audit log entries (for organization accounts) documenting when the tunnel OAuth token was issued and from which IP | Network PCAP or proxy logs showing AdaptixC2 Beacon polling pattern: sustained HTTPS sessions with regular polling intervals to raw.githubusercontent.com or tunnels.api.visualstudio.com from non-developer hosts, with TLS SNI captured — Tropic Trooper's use of these trusted services means the C2 channel will blend into normal enterprise traffic and must be differentiated by session duration, polling regularity, and originating process | Windows Registry persistence keys and scheduled task XML exports: specifically HKCU\Software\Microsoft\Windows\CurrentVersion\Run and HKLM\SOFTWARE\Microsoft\Windows\CurrentVersion\Run for entries created at or after SumatraPDF execution time, plus schtasks /query /xml output for any task whose action path references the SumatraPDF installation directory or a user-writable path — these represent the Beacon's persistence mechanism installed by the trojanized installer

Per-Action IR Details

Step 1: Containment — Identify any SumatraPDF installations across the environment; flag versions not deployed via verified internal software distribution. Isolate hosts running unsigned or unverified SumatraPDF binaries. Block outbound connections to GitHub raw content endpoints and VS Code tunnel relay domains (*.vscode-cdn.net, tunnels.api.visualstudio.com) at the perimeter for hosts that have no legitimate developer use case. Prioritize endpoints used by employees in Taiwan, South Korea, or Japan given the campaign's targeting profile.

NIST Phase: Containment

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

Controls: NIST IR-4 (Incident Handling), NIST CM-7 (Least Functionality), NIST SC-7 (Boundary Protection), CIS 2.3 (Address Unauthorized Software), CIS 4.4 (Implement and Manage a Firewall on Servers), CIS 4.5 (Implement and Manage a Firewall on End-User Devices)

Compensating: Use osquery to enumerate SumatraPDF installations: `SELECT name, version, install_location FROM programs WHERE name LIKE '%Sumatra%';` — cross-reference install paths against known-good internal deployment paths (e.g., SCCM package locations). On Windows hosts without EDR, run `Get-FileHash -Algorithm SHA256 -Path 'C:\path\to\SumatraPDF.exe'` and compare against official checksums published at sumatrapdfreader.org/download-free-pdf-viewer. For network blocking without a commercial firewall, push Windows Firewall rules via GPO to deny outbound TCP 443 to resolved IPs for tunnels.api.visualstudio.com and *.vscode-cdn.net on all non-developer OU members.

Evidence: Before isolating any host, capture: (1) full memory dump using WinPmem or Magnet RAM Capture to preserve any in-memory AdaptixC2 Beacon artifacts; (2) running process list with parent PIDs via `wmic process get name,processid,parentprocessid,executablepath`; (3) active network connections via `netstat -anob` to record any live sessions to GitHub or VS Code tunnel relay IPs; (4) SHA-256 hash and digital signature of the installed SumatraPDF binary via `Get-AuthenticodeSignature`; (5) prefetch files for SumatraPDF.exe at `C:\Windows\Prefetch\SUMATRAPDF*.pf` to establish execution history.

Step 2: Detection — Query EDR telemetry for SumatraPDF processes spawning child processes (cmd.exe, powershell.exe, wscript.exe). Search proxy and firewall logs for sustained outbound sessions to github.com or VS Code tunnel relay domains from non-developer hosts. Look for AdaptixC2 Beacon network signatures if your NDR vendor has released them. Review Windows Event ID 4688 (process creation) and Sysmon Event ID 1 for sumatrapdf.exe with anomalous parent/child relationships. Alert on VS Code CLI invocations (code tunnel) on hosts where VS Code is not an approved application.

NIST Phase: Detection Analysis

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

Controls: NIST IR-5 (Incident Monitoring), NIST AU-6 (Audit Record Review, Analysis, and Reporting), NIST AU-12 (Audit Record Generation), NIST SI-4 (System Monitoring), CIS 8.2 (Collect Audit Logs)

Compensating: Deploy Sysmon with SwiftOnSecurity's config (github.com/SwiftOnSecurity/sysmon-config) to capture Event ID 1 (process create) and Event ID 3 (network connection). Hunt with this PowerShell one-liner against the Security event log: `Get-WinEvent -LogName Security | Where-Object {$_.Id -eq 4688 -and $_.Message -match 'SumatraPDF' -and ($_.Message -match 'cmd.exe' -or $_.Message -match 'powershell.exe' -or $_.Message -match 'wscript.exe')}`. For network detection without NDR, run Wireshark or tcpdump capturing TLS SNI fields on perimeter taps and filter for SNI containing 'vscode-cdn.net' or 'tunnels.api.visualstudio.com' from non-developer VLANs. Deploy the public Sigma rule for VS Code tunnel abuse (search Sigma repo for 'vscode tunnel') converted to Windows Event Log queries.

Evidence: Before alerting or tuning, preserve: (1) Windows Security Event Log exports (EVTX) from suspect hosts for Event IDs 4688 and 4624/4625 covering the suspected intrusion window; (2) Sysmon Event ID 1 logs showing the full SumatraPDF.exe process tree including command-line arguments; (3) Sysmon Event ID 3 (network connection) logs showing outbound connections from sumatrapdf.exe or any child process to github.com, raw.githubusercontent.com, or *.vscode-cdn.net; (4) proxy/firewall logs filtered for sustained HTTPS sessions (>5 minutes duration or high byte counts) to github.com from non-developer hosts, which would indicate AdaptixC2 Beacon polling over GitHub as a relay; (5) DNS query logs for tunnels.api.visualstudio.com resolutions from endpoints not in the approved developer asset group.

Step 3: Eradication — Remove any trojanized SumatraPDF installer or binary. Deploy only the official SumatraPDF build from sumatrapdfreader.org, verified by file hash against the publisher's published checksums. Revoke any VS Code tunnel sessions not authorized by IT. Rotate credentials for any account that authenticated from a suspected compromised host. If AdaptixC2 Beacon persistence mechanisms are identified (registry run keys, scheduled tasks, service installation), remove each persistence artifact and confirm removal via EDR.

NIST Phase: Eradication

Reference: NIST 800-61r3 §3.4 — Eradication

Controls: NIST IR-4 (Incident Handling), NIST SI-2 (Flaw Remediation), NIST SI-3 (Malicious Code Protection), NIST SI-7 (Software, Firmware, and Information Integrity), CIS 2.1 (Establish and Maintain a Software Inventory), CIS 2.2

(Ensure Authorized Software is Currently Supported), CIS 5.3 (Disable Dormant Accounts)

Compensating: Enumerate and remove AdaptixC2 Beacon persistence without EDR using these targeted checks: (1) Registry run keys — `reg query HKCU\Software\Microsoft\Windows\CurrentVersion\Run` and `HKLM\SOFTWARE\Microsoft\Windows\CurrentVersion\Run` for any entry referencing SumatraPDF paths or dropped DLL names; (2) Scheduled tasks — `schtasks /query /fo LIST /v | findstr /i 'sumatra'`; (3) Services — `sc query type= all state= all | findstr /i 'sumatra'`. Revoke VS Code tunnel sessions by navigating to https://github.com/settings/connections and revoking the VS Code tunnel OAuth app authorization for any account that authenticated from a compromised host. For credential rotation, prioritize accounts whose NTLM hashes or Kerberos tickets could have been captured by an in-memory Beacon running under the user's session.

Evidence: Before removing any artifact, capture: (1) full registry export of `HKCU\Software\Microsoft\Windows\CurrentVersion\Run`, `HKLM\SOFTWARE\Microsoft\Windows\CurrentVersion\Run`, and `HKLM\SYSTEM\CurrentControlSet\Services` to document Beacon persistence mechanism; (2) binary copy of the trojanized SumatraPDF installer or EXE (quarantine, do not delete before hashing); (3) contents of any dropped DLLs or staged payloads found in the SumatraPDF installation directory or `%APPDATA%` paths associated with the trojanized installer; (4) scheduled task XML exports via `schtasks /query /xml` for any task referencing the malicious binary; (5) Windows Security Event ID 4648 (explicit credential logon) and ID 4672 (special privileges assigned) logs to identify accounts that may have been used or targeted during the Beacon's active period.

Step 4: Recovery — After remediation, monitor previously affected hosts for 30 days for re-beaconing to GitHub or VS Code tunnel endpoints. Validate that no new unauthorized scheduled tasks, services, or registry run entries have appeared. Confirm SumatraPDF binary hashes match the official publisher release on all endpoints. Re-scan affected hosts with updated threat signatures once EDR vendors publish AdaptixC2 Beacon detections.

NIST Phase: Recovery

Reference: NIST 800-61r3 §3.5 — Recovery

Controls: NIST IR-4 (Incident Handling), NIST SI-7 (Software, Firmware, and Information Integrity), NIST AU-6 (Audit Record Review, Analysis, and Reporting), NIST SI-4 (System Monitoring), CIS 7.1 (Establish and Maintain a Vulnerability Management Process), CIS 7.2 (Establish and Maintain a Remediation Process)

Compensating: Without EDR re-scan capability, implement a daily scheduled task on each recovered host running: `Get-FileHash 'C:\path\to\SumatraPDF.exe' -Algorithm SHA256 | Export-Csv C:\logs\sumatra_hash_audit.csv -Append` — compare output daily against the official hash. For re-beaconing detection, configure Windows Firewall audit logging (`auditpol /set /subcategory:'Filtering Platform Connection' /success:enable /failure:enable`) and parse Event ID 5156 (network connection allowed) for any connection to github.com or vscode-cdn.net from previously affected hosts. Use a YARA rule targeting known AdaptixC2 Beacon signatures (check VirusTotal community and GitHub threat intel repos for published rules) run weekly via `yara64.exe rule.yar C:\` on recovered endpoints.

Evidence: During the 30-day monitoring window, continuously collect: (1) Sysmon Event ID 3 logs from recovered hosts filtered for outbound connections to github.com, raw.githubusercontent.com, and *.vscode-cdn.net to detect any re-established Beacon channel; (2) Windows Event ID 7045 (new service installed) and Task Scheduler operational log (Event ID 106 — task registered) to detect re-persistence attempts; (3) weekly hash verification results for the SumatraPDF binary against sumatrapdfreader.org published SHA-256 checksums; (4) any new entries under `HKCU\Software\Microsoft\Windows\CurrentVersion\Run` or `HKLM\SOFTWARE\Microsoft\Windows\CurrentVersion\Run` that were not present after eradication.

Step 5: Post-Incident — Review allowlist policies for developer services (GitHub, VS Code tunnels, similar LoTS infrastructure); implement context-aware controls that restrict these services to hosts with a documented developer role. Evaluate whether software installation policies prevent users from executing unsigned or ZIP-delivered installers. Map the gap against MITRE ATT&CK T1102 (Web Service C2) and T1572 (Protocol Tunneling) and assess whether existing detection coverage addresses both techniques. Update threat hunt hypotheses to include living-off-trusted-sites (LoTS) C2 patterns.

NIST Phase: Post Incident

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

Controls: NIST IR-4 (Incident Handling), NIST IR-8 (Incident Response Plan), NIST SI-5 (Security Alerts, Advisories, and Directives), NIST CM-7 (Least Functionality), CIS 2.3 (Address Unauthorized Software), CIS 7.1 (Establish and Maintain a Vulnerability Management Process), CIS 4.6 (Securely Manage Enterprise Assets and Software)

Compensating: For LoTS C2 detection without a commercial SIEM, write a Sigma rule targeting VS Code tunnel CLI invocations: process name `code.exe` with command-line argument `tunnel` on hosts not in a developer OU — convert to Windows Event Log or Sysmon format using sigma-cli. For software installation policy enforcement without MDM, push a GPO Software Restriction Policy (SRP) or AppLocker rule denying execution of unsigned binaries and binaries run from `%TEMP%`, `%APPDATA%`, or user-writable paths — this directly blocks trojanized ZIP-delivered SumatraPDF installers. Document the ATT&CK T1102 and T1572 gap in the IR lessons-learned report and submit detection logic to the MITRE ATT&CK Defender (MAD) community or internal threat hunt backlog.

Evidence: For lessons-learned documentation, compile: (1) timeline reconstruction from Sysmon and Windows Event logs showing the full attack chain from trojanized SumatraPDF execution through AdaptixC2 Beacon establishment and C2 communication via GitHub or VS Code tunnel relay; (2) all IOCs collected during the incident (SumatraPDF binary hashes, C2 relay domains/IPs, dropped payload paths, registry persistence keys) formatted for STIX/TAXII sharing or internal threat intel platform ingestion; (3) proxy/firewall log excerpts documenting the LoTS C2 session patterns (connection duration, bytes transferred, polling interval) to baseline future anomaly detection for T1102 and T1572; (4) gap analysis documenting which existing detection rules fired, which did not, and why GitHub and VS Code tunnel traffic bypassed perimeter controls.

Detection Guidance

Primary behavioral indicators: SumatraPDF processes spawning shells or network connections; VS Code tunnel CLI invocations on non-developer hosts; sustained HTTPS sessions from endpoint processes to github.com or *.vscode-cdn.net where the initiating process is not a recognized developer tool. Log sources to query: EDR process tree telemetry, Sysmon Event IDs 1 (process create), 3 (network connect), and 11 (file create); proxy/firewall logs filtered for VS Code tunnel relay FQDNs; Windows Event ID 4688 with command-line logging enabled. Specific patterns: sumatrapdf.exe initiating outbound connections; code.exe with --tunnel argument on hosts outside developer populations; periodic low-volume HTTPS beaconing to GitHub repositories not associated with known internal projects. Network-based detection is degraded by design, behavioral and host-based telemetry should be the primary detection surface for this campaign. AdaptixC2 Beacon-specific network signatures should be obtained from EDR/NDR vendors once published. Source quality for IOCs in this item is T3 (0.64); no confirmed hashes, IPs, or domains have been independently verified, treat any published IOCs as leads requiring validation before blocking.

Indicators of Compromise

Type	Value	Context	Confidence
URL	`github.com (repositories used as C2 relay)`	AdaptixC2 Beacon routes C2 traffic through GitHub repositories; specific repository URLs not confirmed in verified sources	LOW
DOMAIN	`*.vscode-cdn.net`	VS Code tunnel relay infrastructure abused for C2 communication; pattern-based, not a specific IOC from a verified primary source	LOW

Type	Value	Context	Confidence
DOMAIN	`tunnels.api.visualstudio.com`	VS Code remote tunnel API endpoint; used by the threat actor to blend C2 with legitimate developer traffic	LOW

Framework Mappings

MITRE-ATTACK

- **T1021.005** — VNC
- **T1102.002** — Bidirectional Communication
- **T1071.001** — Web Protocols
- **T1027** — Obfuscated Files or Information
- **T1572** — Protocol Tunneling
- **T1059** — Command and Scripting Interpreter
- **T1566.001** — Spearphishing Attachment
- **T1105** — Ingress Tool Transfer
- **T1036.005** — Match Legitimate Resource Name or Location
- **T1102** — Web Service
- **T1204.002** — Malicious File
- **T1195.002** — Compromise Software Supply Chain

NIST-800-53R5

- **SI-3** — Malicious Code Protection
- **SI-4** — System Monitoring
- **CM-7** — Least Functionality
- **SI-7** — Software, Firmware, and Information Integrity
- **AT-2** — Literacy Training and Awareness
- **SC-7** — Boundary Protection
- **SI-8** — Spam Protection
- **CA-7** — Continuous Monitoring
- **SA-9** — External System Services
- **SR-3** — Supply Chain Controls and Processes
- **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

MITRE ATT&CK Mapping

Technique ID	Technique Name	Tactic
T1021.005	VNC	Lateral-Movement
T1102.002	Bidirectional Communication	Command-And-Control
T1071.001	Web Protocols	Command-And-Control
T1027	Obfuscated Files or Information	Defense-Evasion
T1572	Protocol Tunneling	Command-And-Control
T1059	Command and Scripting Interpreter	Execution
T1566.001	Spearphishing Attachment	Initial-Access
T1105	Ingress Tool Transfer	Command-And-Control
T1036.005	Match Legitimate Resource Name or Location	Defense-Evasion
T1102	Web Service	Command-And-Control
T1204.002	Malicious File	Execution
T1195.002	Compromise Software Supply Chain	Initial-Access

Sources

Source	URL	Tier
Security News	https://thehackernews.com/2026/04/tropic-trooper-uses-trojanized.html	T3
Tropic Trooper Uses Trojanized SumatraPDF and GitHub to Deploy ...	https://www.socdefenders.ai/item/a18de338-2ae0-44b3-8335-d98b46c77dd1	T3
Untrusted Search Path in SumatraPDF Reader (explorer.exe on ...	https://github.com/sumatrapdfreader/sumatrapdf/security/advisories/...	T3
CVE-2026-25920: SumatraPDF Buffer Overflow Vulnerability	https://www.sentinelone.com/vulnerability-database/cve-2026-25920/	T3

Source	URL	Tier
Security vulnerability reporting · Issue #5254 - GitHub	https://github.com/sumatrapdfreader/sumatrapdf/issues/5254	T3

Generated 2026-04-24 13:42 UTC by TJS Security Command Center