INTELLIGENCE BRIEFING
Security Command Center
TLP:CLEAR
2026-05-12 14:08 UTC

JDownloader Official Website Compromised to Distribute Python RAT via Trojanized Installers

THREAT CAMPAIGN | **CRITICAL** | CVSS 9.0

SCC Item ID	SCC-CAM-2026-0305
Type	Threat Campaign
Severity	CRITICAL
CVSS Base Score	9.0
Affected Products	JDownloader Windows and Linux installers downloaded between May 6-7, 2026
Published	2026-05-11
Discovery Source	Gemini

Executive Summary

The official JDownloader website was compromised on May 6-7, 2026, with legitimate Windows and Linux installers replaced by trojanized versions containing a Python-based Remote Access Trojan. Any user or organization that downloaded JDownloader during that window received malware capable of full system compromise, credential theft, and persistent remote access. This is a supply chain attack against a trusted distribution channel, meaning standard defenses like HTTPS verification and trusted-source policies provided no protection.

Technical Analysis

Threat actors compromised the official JDownloader distribution infrastructure and replaced legitimate installer binaries for Windows and Linux with trojanized versions embedding a Python-based RAT. The specific RAT family and attribution have not been publicly confirmed at time of writing. The attack window is confirmed as May 6-7, 2026. No CVE has been assigned; this is a website and distribution infrastructure compromise, not a vulnerability in JDownloader's codebase. Relevant CWEs: CWE-494 (Download of Code Without Integrity Check) and CWE-506 (Embedded Malicious Code). MITRE ATT&CK techniques observed or inferred: T1195.002 (Compromise Software Supply Chain), T1059.006 (Python command execution), T1071.001 (C2 over HTTP/S), T1027 (Obfuscated Files or Information), T1543 (Create or Modify System Process for persistence). No vendor-issued patch applies; remediation requires removal of the trojanized installer and any installed instance obtained during the compromise window. Installer integrity verification mechanisms (hash checking, code signing validation) were bypassed or absent at the distribution level. Source quality score is 0.64 with all current sources at T3; treat specific technical claims as pending confirmation from the JDownloader project or a vetted malware analysis report.

Action Checklist

1. Containment, Identify all systems where JDownloader was downloaded or installed between May 6-7, 2026. Isolate those endpoints from the network immediately pending investigation. Block outbound connections from those hosts until triage is complete.
2. Detection, Search endpoint logs, EDR telemetry, and proxy/DNS logs for JDownloader installer execution during May 6-7, 2026. Hunt for Python interpreter processes spawned by or associated with the JDownloader installer process tree (T1059.006). Look for unexpected outbound HTTP/S connections from systems where JDownloader was recently installed (T1071.001). Check for new services, scheduled tasks, or startup entries created around installation time (T1543).
3. Eradication, Remove any JDownloader installation obtained during the May 6-7 window. Do not attempt to repair or reinstall from the same source until JDownloader officially confirms the distribution infrastructure is clean and publishes verified hashes. Conduct full malware triage on affected hosts; the RAT may have dropped secondary payloads or established persistence independently of the installer.
4. Recovery, Reimage compromised hosts where possible, or perform thorough forensic verification before returning them to production. Reset credentials for any accounts accessed from affected systems. Verify no lateral movement occurred from isolated hosts before reconnecting to the network. Monitor affected user accounts for anomalous activity post-remediation.
5. Post-Incident, Review software procurement controls: enforce installer hash verification against published checksums before execution. Evaluate whether open-source or freeware tools in your environment are subject to equivalent integrity checking. Map this incident to T1195.002 and assess whether your threat model adequately covers supply chain compromise of non-enterprise tools. Consider blocking direct installer downloads from unofficial or unverified mirrors as a standing policy.

IR / Forensic Enrichment

Triage Priority	IMMEDIATE
Escalation Criteria	Escalate to senior IR leadership, legal counsel, and executive notification if any of the following are confirmed: (1) credential theft evidence found (Event ID 4648/4776 showing post-compromise authentication), triggering potential breach notification obligations under GDPR, CCPA, or HIPAA if affected systems processed PII or PHI; (2) lateral movement detected from isolated hosts to domain controllers or cloud infrastructure; (3) the organization lacks EDR visibility on more than 20% of endpoints, creating an unquantifiable blast radius; or (4) the Python RAT is confirmed to have exfiltrated data, shifting the incident classification from compromise to data breach.

Recovery Notes	Prioritize reimaging over in-place remediation for all confirmed compromised hosts — the Python RAT in this supply chain attack may have established persistence mechanisms independent of the JDownloader installation path, and manual removal risks leaving residual backdoors. Before returning any host to production, verify the SHA-256 hash of any replacement JDownloader installer against hashes published by the JDownloader project via an independent channel (GitHub, signed release notes) — not from jdownloader.org until the project formally confirms distribution infrastructure integrity. Monitor all user accounts that authenticated from affected hosts for a minimum of 30 days post-remediation, specifically tracking anomalous logon times, geographic anomalies, and new OAuth token grants or API key creation events that could indicate the threat actor retained credential-based access after endpoint remediation.
Forensic Artifacts	Python RAT process tree: Sysmon Event ID 1 (Process Create) showing python.exe or pythonw.exe with ParentImage referencing the JDownloader install directory (C:\Users\\AppData\Local\JDownloader 2.0\ on Windows or ~/.jdownloader2/ on Linux) — anomalous because legitimate JDownloader runs on JVM and has no reason to spawn a Python interpreter | Trojanized installer binary: the JDownloader installer file downloaded between May 6–7, 2026, preserved with SHA-256 hash and file metadata intact — located in browser download directories (C:\Users\\Downloads\ or ~/Downloads/), Windows Prefetch at C:\Windows\Prefetch\JDOWNLOADER*.pf, and MFT ($MFT) entries confirming creation timestamp within the compromise window | Python RAT persistence artifacts: Windows Registry autorun keys (HKCU\Software\Microsoft\Windows\CurrentVersion\Run), scheduled tasks (C:\Windows\System32\Tasks\ directory), or Windows Services (HKLM\SYSTEM\CurrentControlSet\Services\) created between May 6–8, 2026; on Linux, new systemd unit files in /etc/systemd/system/ or crontab entries containing python3 or references to scripts in ~/.config/ or /tmp/ | C2 network artifacts: Sysmon Event ID 3 (Network Connection) logs showing python.exe establishing outbound TCP connections to non-JDownloader infrastructure, plus proxy/DNS logs capturing the destination domains or IPs contacted — HTTP POST request bodies in proxy logs may contain encoded system information consistent with RAT beacon or exfiltration traffic (T1071.001) | Credential exposure evidence: Windows Security Event Log Event IDs 4624 (Successful Logon), 4648 (Explicit Credential Use), and 4776 (NTLM Credential Validation) from affected hosts covering May 6 through isolation date, plus browser credential store files (C:\Users\\AppData\Local\Google\Chrome\User Data\Default\Login Data, equivalent Firefox logins.json) which Python RATs commonly target for credential harvesting

Per-Action IR Details

Containment — Identify all systems where JDownloader was downloaded or installed between May 6–7, 2026. Isolate those endpoints from the network immediately pending investigation. Block outbound connections from those hosts until triage is complete.

NIST Phase: Containment

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

Controls: NIST IR-4 (Incident Handling), NIST IR-5 (Incident Monitoring), CIS 1.1 (Establish and Maintain Detailed Enterprise Asset Inventory), CIS 4.4 (Implement and Manage a Firewall on Servers), CIS 4.5 (Implement and Manage a Firewall on End-User Devices)

Compensating: Query proxy or DNS logs for requests to jdownloader.org or any CDN mirror domains between 2026-05-06T00:00Z and 2026-05-07T23:59Z. On Windows, run: Get-WinEvent -LogName Security | Where-Object {$_.Id -eq 4688 -and $_.Message -match 'JDownloader'} to find installer executions. Use Windows Firewall (netsh advfirewall firewall add rule name='Block JD RAT Outbound' dir=out action=block) or iptables (iptables -I OUTPUT -m owner --uid-owner -j DROP) to block outbound traffic from confirmed hosts while preserving disk state for forensics.

Evidence: Before isolating, capture: (1) active network connections from the host using netstat -anob (Windows) or ss -tulpn (Linux) — the Python RAT will show established TCP sessions to C2 infrastructure; (2) running process list with full command lines via tasklist /v (Windows) or ps auxf (Linux), specifically looking for python.exe or python3 processes not launched by a known application; (3) volatile memory image using WinPmem or LiME kernel module before network isolation destroys C2 session artifacts; (4) Windows Prefetch files at C:\Windows\Prefetch\JDOWNLOADER*.pf showing execution timestamps; (5) browser download history confirming the May 6–7 download window from jdownloader.org.

Detection — Search endpoint logs, EDR telemetry, and proxy/DNS logs for JDownloader installer execution during May 6–7, 2026. Hunt for Python interpreter processes spawned by or associated with the JDownloader installer process tree (T1059.006). Look for unexpected outbound HTTP/S connections from systems where JDownloader was recently installed (T1071.001). Check for new services, scheduled tasks, or startup entries created around installation time (T1543).

NIST Phase: Detection Analysis

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

Controls: NIST SI-4 (System Monitoring), NIST AU-6 (Audit Record Review, Analysis, and Reporting), NIST AU-12 (Audit Record Generation), CIS 8.2 (Collect Audit Logs), CIS 7.1 (Establish and Maintain a Vulnerability Management Process)

Compensating: Deploy Sysmon (config with SwiftOnSecurity baseline) and query Event ID 1 (Process Create) for any python.exe or pythonw.exe whose ParentImage path contains 'JDownloader' or whose ParentCommandLine references the installer. Query Sysmon Event ID 3 (Network Connect) for outbound connections from python.exe processes to non-local IPs. For scheduled task persistence (T1543.005), run: Get-ScheduledTask | Where-Object {$_.Date -ge '2026-05-06' -and $_.Date -le '2026-05-08'} | Select TaskName,TaskPath,Date. On Linux, check crontab -l for all users and examine /etc/systemd/system/ and /etc/init.d/ for new unit files created during the May 6–7 window using: find /etc/systemd/system -newer /tmp/ref_date -type f. Use a Sigma rule targeting python.exe spawned by a Java-based parent (JDownloader is Java-based, making python.exe a highly anomalous child process).

Evidence: Capture before analysis: (1) Windows Security Event Log Event ID 4688 (Process Creation) with command-line auditing enabled, filtering for python.exe or pythonw.exe with parent process referencing JDownloader's install path (default: C:\Users\\AppData\Local\JDownloader 2.0\); (2) Sysmon Event ID 7 (Image Loaded) showing DLLs or Python modules loaded by the RAT process; (3) Proxy/DNS logs showing DNS queries and HTTP POSTs or GETs from affected hosts to domains/IPs not associated with the legitimate JDownloader update infrastructure (app.jdownloader.org, api.jdownloader.org); (4) Windows Registry run keys at HKCU\Software\Microsoft\Windows\CurrentVersion\Run and HKLM\SOFTWARE\Microsoft\Windows\CurrentVersion\Run for entries created May 6–7, 2026; (5) Linux: ~/.bashrc, ~/.profile, /etc/rc.local, and /etc/cron.d/ for new persistence entries timestamped within the compromise window.

Eradication — Remove any JDownloader installation obtained during the May 6–7 window. Do not attempt to repair or reinstall from the same source until JDownloader officially confirms the distribution infrastructure is clean and publishes verified hashes. Conduct full malware triage on affected hosts; the RAT may have dropped secondary payloads or established persistence independently of the installer.

NIST Phase: Eradication

Reference: NIST 800-61r3 §3.4 — Eradication

Controls: NIST SI-2 (Flaw Remediation), NIST SI-3 (Malicious Code Protection), NIST SI-7 (Software, Firmware, and Information Integrity), NIST CM-7 (Least Functionality — implied via removal of unauthorized software), CIS 2.3 (Address Unauthorized Software), CIS 2.2 (Ensure Authorized Software is Currently Supported)

Compensating: Use ClamAV with a freshly updated signature database to scan the full JDownloader install directory and any user profile temp directories (C:\Users\\AppData\Local\Temp\ on Windows, /tmp/ and ~/.cache/ on Linux) where the RAT may have staged payloads. Write a YARA rule targeting Python-compiled bytecode (.pyc files) or embedded Python scripts in unexpected locations: scan with yara -r C:\Users\\AppData\ . Enumerate all files created or modified in the JDownloader install path between May 5–8, 2026 using: Get-ChildItem -Recurse 'C:\Users\\AppData\Local\JDownloader 2.0\' | Where-Object {$_.LastWriteTime -ge '2026-05-05'} | Select

FullName,LastWriteTime,Length. Do not reinstall JDownloader until the project publishes SHA-256 hashes for clean installers via a channel independent of jdownloader.org (e.g., their GitHub release page).

Evidence: Capture before eradication: (1) Full forensic disk image (using FTK Imager or dc3dd) of affected hosts before any removal — the trojanized installer and any dropped RAT components constitute evidence; (2) File hashes (SHA-256) of the trojanized installer file and all files in the JDownloader install directory, compared against any vendor-published clean hashes; (3) Contents of the JDownloader install directory including any unexpected .py, .pyc, .pyd, or .dll files not present in a clean installation; (4) Python RAT persistence artifacts: Windows services (sc query state=all output), scheduled tasks (schtasks /query /fo LIST /v), and registry autorun keys exported via reg export HKLM\SOFTWARE\Microsoft\Windows\CurrentVersion\Run C:\evidence\run_keys.reg; (5) Linux: output of systemctl list-units --type=service --state=running and all files in ~/.local/share/applications/ or /usr/local/bin/ created in the compromise window.

Recovery — Reimage compromised hosts where possible, or perform thorough forensic verification before returning them to production. Reset credentials for any accounts accessed from affected systems. Verify no lateral movement occurred from isolated hosts before reconnecting to the network. Monitor affected user accounts for anomalous activity post-remediation.

NIST Phase: Recovery

Reference: NIST 800-61r3 §3.5 — Recovery

Controls: NIST IR-4 (Incident Handling), NIST IR-8 (Incident Response Plan), NIST AC-2 (Account Management), NIST IA-5 (Authenticator Management — implied via credential reset), CIS 5.3 (Disable Dormant Accounts), CIS 5.4 (Restrict Administrator Privileges to Dedicated Administrator Accounts), CIS 6.2 (Establish an Access Revoking Process)

Compensating: For hosts that cannot be reimaged, run a manual integrity check: compare all files in the OS and user profile directories against a known-good baseline using Get-FileHash on Windows or sha256sum on Linux, cross-referenced against a clean system snapshot. For credential reset scope, pull Windows Security Event Log Event ID 4624 (Successful Logon) and 4648 (Explicit Credential Logon) from affected hosts covering May 6 through isolation date to enumerate all accounts that authenticated — every account on that list requires password reset and session token revocation. To check for lateral movement, query network flow logs or Windows Security Event ID 4648 and 4776 (Credential Validation) on domain controllers for authentications originating from the isolated hosts' IP addresses after May 6, 2026. Use osquery on surviving infrastructure: SELECT * FROM logged_in_users; and SELECT * FROM last; to identify sessions that may have propagated from compromised hosts.

Evidence: Before reconnecting hosts to production: (1) Windows Event ID 4624/4648 logs exported from the affected host covering the full exposure window (May 6 through isolation), identifying every credential used from that host — these drive the credential reset scope; (2) SMB connection logs (Event ID 5140 — Network Share Object Accessed) from adjacent file servers showing whether the compromised host accessed shared resources during the compromise window; (3) Domain controller logs for Kerberos TGT requests (Event ID 4768) originating from the affected host's IP, indicating potential pass-the-hash or credential reuse for lateral movement; (4) VPN/remote access logs for any sessions authenticated from the affected user's credentials after May 6, 2026, indicating credential theft and external use; (5) Cloud service access logs (e.g., Azure AD Sign-In Logs, AWS CloudTrail) if affected users had cloud credentials stored on the compromised host.

Post-Incident — Review software procurement controls: enforce installer hash verification against published checksums before execution. Evaluate whether open-source or freeware tools in your environment are subject to equivalent integrity checking. Map this incident to T1195.002 and assess whether your threat model adequately covers supply chain compromise of non-enterprise tools. Consider blocking direct installer downloads from unofficial or unverified mirrors as a standing policy.

NIST Phase: Post Incident

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

Controls: NIST SI-7 (Software, Firmware, and Information Integrity), NIST SI-2 (Flaw Remediation), NIST RA-3 (Risk Assessment), NIST SA-9 (External System Services — supply chain context), NIST IR-8 (Incident Response Plan), CIS 2.1 (Establish and Maintain a Software Inventory), CIS 2.2 (Ensure Authorized Software is Currently Supported),

CIS 7.1 (Establish and Maintain a Vulnerability Management Process), CIS 7.2 (Establish and Maintain a Remediation Process)

Compensating: Build a hash verification gate into your software deployment workflow using PowerShell: (Get-FileHash -Algorithm SHA256 -Path .\JDownloader-Setup.exe).Hash -eq '' before any installer execution — automate this as a pre-execution check script. Conduct an inventory audit using: Get-Package -Name '*JDownloader*' across all managed endpoints via PSRemoting, or deploy an osquery query (SELECT name,version,install_date FROM programs WHERE name LIKE '%JDownloader%';) to enumerate the full blast radius. For standing supply chain detection, implement a Sigma rule alerting on any installer EXE downloaded from a freeware/open-source project domain that spawns an interpreter process (python.exe, perl.exe, ruby.exe) within 60 seconds of first execution — this would have detected this attack pattern. Add jdownloader.org installer download events as a canary in your proxy log alerting until JDownloader confirms clean infrastructure.

Evidence: Collect for lessons-learned and threat model update: (1) Complete software inventory of all freeware, open-source, and unsanctioned tools installed across the environment (from SCCM, Intune, or osquery) to identify the full population of similarly-at-risk software lacking code-signing or hash verification controls; (2) Proxy log summary of all external installer downloads (files matching *.exe, *.msi, *.sh, *.deb, *.rpm patterns) from the past 90 days, to quantify exposure from unverified installer procurement; (3) ATT&CK T1195.002 (Compromise Software Supply Chain) mapping documentation, including whether existing detection rules would have flagged the Python RAT process tree spawned from a Java-based parent (JDownloader runs on JVM) — this parent-child anomaly is a high-fidelity detection opportunity that should be codified; (4) The incident timeline from first download (May 6, 2026) to detection, measuring mean time to detect (MTTD) against the NIST 800-61r3 §4 lessons-learned framework to identify detection gaps in endpoint and proxy monitoring; (5) Hash comparison report: SHA-256 of recovered trojanized installer vs. any vendor-published hash for the clean version, preserved as permanent incident evidence.

Detection Guidance

Primary hunt: query EDR or SIEM for process creation events where the parent process is a JDownloader installer (filename patterns: jdownloader*.exe, jdownloader*.sh, jd-*.jar with install-related naming) executed between May 6-7, 2026. Secondary hunt: identify python.exe or python3 processes spawned within the JDownloader process tree or from the JDownloader installation directory. Watch for outbound network connections (T1071.001) from JDownloader install paths to non-JDownloader infrastructure. Review Windows Event Log ID 4688 (process creation) and Linux auditd execve records for Python interpreter invocations originating from JDownloader directories. Check for new scheduled tasks (Windows: Event ID 4698; Linux: cron entries) or services created at or after installer execution. No confirmed IOC hashes or C2 infrastructure have been publicly released at time of writing; flag this detection guidance as preliminary pending a vetted malware analysis report.

Indicators of Compromise

Type	Value	Context	Confidence
URL	`https://jdownloader.org (installers downloaded May 6–7, 2026)`	Official distribution site confirmed to have served trojanized installers during this window; treat any installer obtained in this window as malicious	HIGH
HASH	`[not yet publicly confirmed]`	No verified file hashes for the trojanized installers have been published in available T3 sources at time of writing; monitor threat intel feeds for updates	LOW

Framework Mappings

MITRE-ATTACK

- **T1071.001** — Web Protocols
- **T1027** — Obfuscated Files or Information
- **T1543** — Create or Modify System Process
- **T1195.002** — Compromise Software Supply Chain
- **T1059.006** — Python

NIST-800-53R5

- **SI-3** — Malicious Code Protection
- **SI-4** — System Monitoring
- **CM-7** — Least Functionality
- **SA-9** — External System Services
- **SR-3** — Supply Chain Controls and Processes
- **SI-7** — Software, Firmware, and Information Integrity
- **CM-3** — Configuration Change Control
- **SR-2** — Supply Chain Risk Management Plan

OWASP-TOP10-2021

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

CIS-V8

- **2.5** — Allowlist Authorized Software
- **2.6** — Allowlist Authorized Libraries
- **15.1** — Establish and Maintain an Inventory of Service Providers

ISO-27001-2022

- **A.8.8** — Management of technical vulnerabilities
- **A.5.21** — Managing information security in the ICT supply chain

NIST-CSF-2

- **GV.SC-01** — Cybersecurity supply chain risk management program

SOC2-TSC

- **CC9.2** — Manages risks associated with vendors and business partners

MITRE ATT&CK Mapping

Technique ID	Technique Name	Tactic
T1071.001	Web Protocols	Command-And-Control

Technique ID	Technique Name	Tactic
T1027	Obfuscated Files or Information	Defense-Evasion
T1543	Create or Modify System Process	Persistence
T1195.002	Compromise Software Supply Chain	Initial-Access
T1059.006	Python	Execution

Sources

Source	URL	Tier
Official JDownloader site served malware to Windows and Linux ...	https://securityaffairs.com/191920/malware/official-jdownloader-sit...	T3
New cPanel vulnerabilities, JDownloader delivers malware ...	https://www.youtube.com/watch?v=DaE-F7KAyLs	T3
JDownloader site hacked to replace installers with Python RAT ...	https://www.reddit.com/r/DataHoarder/comments/1t8nnkn/jdownloader_s...	T3
Top download manager JDownloader hacked — installers replaced ...	https://www.techradar.com/pro/security/top-download-manager-jdownlo...	T3
New cPanel vulnerabilities, JDownloader delivers malware ...	https://www.linkedin.com/pulse/new-cpanel-vulnerabilities-jdownload...	T3

Generated 2026-05-12 14:08 UTC by TJS Security Command Center