INTELLIGENCE BRIEFING
Security Command Center
TLP:CLEAR
2026-06-27 18:14 UTC

ClickOnce as a Weaponized Delivery and Persistence Platform: What Defenders Must Address Now

SECURITY ANALYSIS | HIGH | CVSS 5.0

SCC Item ID	SCC-STY-2026-0286
Type	Security Analysis
Severity	HIGH
CVSS Base Score	5.0
Affected Products	Microsoft Windows, ClickOnce deployment framework (dfsvc.exe, rundll32.exe, .appref-ms files); all Windows versions supporting ClickOnce; CrowdStrike Falcon (detection context)
Discovery Source	Rss:T1 Threatintel

Executive Summary

CrowdStrike researchers have documented a multi-stage attack chain that weaponizes Microsoft's ClickOnce deployment framework, a widely trusted enterprise technology, to deliver and persist malicious payloads without requiring elevated privileges or triggering standard email security filters. Because ClickOnce runs silently through trusted Windows binaries and its auto-update mechanism can pull attacker-controlled code without user interaction, this technique lowers the cost of entry for a broad range of threat actors. The research signals a maturing pattern of living-off-the-land abuse targeting legitimate deployment infrastructure, demanding immediate defensive attention from any organization that allows ClickOnce in its environment.

Technical Analysis

CrowdStrike researchers have documented a two-part analysis (June 2026) detailing how attackers can abuse Microsoft's ClickOnce deployment framework across three interlocking vectors. First, the .appref-ms file format, which Windows associates with ClickOnce application launches, can be weaponized as an initial access and persistence artifact. Delivered via spearphishing links (MITRE T1566.002) or malicious file execution (T1204.002), an .appref-ms file silently invokes the ClickOnce runtime without presenting a UAC prompt, satisfying CWE-276 (Incorrect Default Permissions) as a root cause. Second, ClickOnce's auto-update mechanism, if the application's deployment manifest points to an attacker-controlled or compromised server, will silently pull and execute updated payloads. This satisfies CWE-494 (Download of Code Without Integrity Check) because the integrity of the update source is not independently verified by the endpoint. Third, the execution

chain runs through dfsvc.exe and rundll32.exe, both signed Microsoft binaries, enabling living-off-the-land (LotL) behavior (T1218.011) that evades controls keyed to unsigned or unknown process execution. Persistence is achievable via .appref-ms file hijacking placed in startup locations (T1547, T1547.001) or scheduled task registration (T1053.005). The research also notes supply-chain relevance (T1195) in scenarios where a legitimate ClickOnce application's update infrastructure is compromised rather than deployed fresh by the attacker, turning a trusted software delivery channel into a lateral movement and payload staging conduit (T1072). CWE-693 (Protection Mechanism Failure) underlies the entire chain: ClickOnce was designed to bypass elevation requirements as a feature, and that design choice becomes a liability when the deployment source is adversary-controlled. No specific active campaign is attributed, but the technique's low privilege requirement and prevalence of ClickOnce in enterprise environments, particularly in organizations using legacy .NET desktop applications, make broad adoption by opportunistic and targeted threat actors technically plausible. CrowdStrike Falcon is cited in the research context as a detection reference, indicating the techniques were evaluated against modern EDR telemetry.

Action Checklist

1. Step 1: Assess exposure, audit whether your organization deploys or permits ClickOnce applications; query endpoints for the presence of .appref-ms files and dfsvc.exe execution history using EDR telemetry (CIS 1.1: Establish and Maintain Detailed Enterprise Asset Inventory; CIS 2.1: Establish and Maintain a Software Inventory)
2. Step 2: Review software inventory and authorization, verify that all ClickOnce applications in your environment are authorized, sourced from known-good deployment servers, and listed in your software inventory; flag any .appref-ms files not traceable to an approved application (CIS 2.1; CIS 2.3: Address Unauthorized Software)
3. Step 3: Audit update server integrity, for every authorized ClickOnce application, confirm the deployment manifest's update URL resolves to an organization-controlled or vendor-controlled server; any manifest pointing to an external or unrecognized host is a priority investigation item (NIST AC-20: Use Of External Systems)
4. Step 4: Configure application control and process monitoring, restrict or block dfsvc.exe and rundll32.exe from spawning unexpected child processes; alert on rundll32.exe executing ClickOnce payloads outside approved software paths; apply allowlisting where feasible (NIST AC-3: Access Enforcement; NIST SI-4: Information System Monitoring)
5. Step 5: Enforce least privilege and audit startup persistence locations, review user startup directories, registry run keys, and scheduled tasks for unauthorized .appref-ms entries; remove any not tied to approved applications (NIST AC-6: Least Privilege; T1547.001/T1053.005 mapped persistence locations)
6. Step 6: Update threat model and detection rules, add ClickOnce LotL abuse to your threat register, mapped to MITRE T1218.011, T1547, T1566.002, and T1105; tune EDR and SIEM rules to alert on dfsvc.exe network connections to non-approved hosts and .appref-ms file creation outside expected software directories
7. Step 7: Communicate findings, brief leadership on whether ClickOnce is present in the environment and what the silent-execution, no-UAC-prompt characteristic means for user-level attackers gaining a foothold without IT visibility
8. Step 8: Monitor for updates, watch for vendor mitigations from Microsoft or CISA advisories referencing this technique chain

IR / Forensic Enrichment

Triage Priority	URGENT
Escalation Criteria	Escalate to incident response immediately if any .appref-ms file is found whose deployment manifest URL resolves to a non-organization-controlled, non-vendor host, OR if Sysmon or EDR telemetry shows dfsvc.exe spawning child processes (cmd.exe, powershell.exe, mshta.exe) or initiating outbound connections to external hosts — either condition indicates active exploitation rather than mere exposure, and may trigger breach notification obligations if the payload accessed PII or PHI datastores.
Recovery Notes	After eradicating all unauthorized .appref-ms persistence entries and blocking unapproved dfsvc.exe network connections, reimage any host where dfsvc.exe was confirmed to have executed a payload from an attacker-controlled update URL, as the auto-update mechanism may have staged additional second-stage tooling that static file-system review will not surface. Monitor the ClickOnce application cache directory (`%LOCALAPPDATA%\Apps\2.0\`) and Windows Application event log for at least 30 days post-remediation for re-appearance of unauthorized .appref-ms files, which would indicate a missed persistence vector or re-infection via the original spearphish delivery channel. Validate recovery by confirming that dfsvc.exe produces zero outbound network connections to non-approved hosts across all monitored endpoints before closing the incident.
Forensic Artifacts	ClickOnce application cache directory contents: `%LOCALAPPDATA%\Apps\2.0\` — contains the downloaded payload DLLs and manifests staged by dfsvc.exe during auto-update pulls from attacker-controlled servers; hash all files present and compare against known-good vendor manifests | Windows Prefetch for dfsvc.exe: `C:\Windows\Prefetch\DFSVC.EXE-*.pf` — records the last eight execution timestamps and the file paths accessed during each execution, establishing a timeline of when ClickOnce was invoked and which payload directories it touched | Windows Application event log and Microsoft-Windows-ClickOnce-EventLog/Operational channel — contains deployment activation events, update check attempts, and manifest download records specific to dfsvc.exe activity, including the source URLs contacted during each update cycle | .appref-ms file metadata and contents from user startup directories (`%APPDATA%\Microsoft\Windows\Start Menu\Programs\Startup\`) and any registry Run key values — each file embeds the deployment manifest URL in plaintext, directly identifying the attacker-controlled update server | Sysmon Event ID 3 (NetworkConnect) records for dfsvc.exe and rundll32.exe — captures the destination IP, hostname, and port of every outbound connection made during ClickOnce payload delivery and update checks, providing the network-layer evidence of C2 or payload-staging activity

Per-Action IR Details

Step 1: Assess exposure — audit whether your organization deploys or permits ClickOnce applications; query endpoints for the presence of .appref-ms files and dfsvc.exe execution history using EDR telemetry (CIS 1.1: Establish and Maintain Detailed Enterprise Asset Inventory; CIS 2.1: Establish and Maintain a Software Inventory)

NIST Phase: Preparation

Reference: NIST 800-61r3 §2 — Preparation: establishing situational awareness of attack-surface exposure before an incident is declared

Controls: CIS 1.1 (Establish and Maintain Detailed Enterprise Asset Inventory), CIS 2.1 (Establish and Maintain a Software Inventory)

Compensating: Without EDR, run the following on each Windows endpoint via PowerShell remoting or a scheduled task: `Get-ChildItem -Path $env:APPDATA\Microsoft\Windows\Start Menu\Programs -Filter *.appref-ms -Recurse` and `Get-WinEvent -LogName 'Microsoft-Windows-Application-Server-Applications/Operational' | Where-Object {$_.Message -like '*dfsvc*'}`. Sysmon Event ID 1 (Process Create) filtered on Image containing 'dfsvc.exe' will surface execution history if Sysmon was deployed prior to this audit.

Evidence: This step is a passive audit and does not alter live system state. However, document the pre-audit baseline before any remediation: export the full list of .appref-ms file paths, their creation timestamps, and owning user accounts from every endpoint. Capture `dfsvc.exe` execution history from the Windows Application event log and, if available, Sysmon logs — this baseline is critical for distinguishing pre-existing legitimate ClickOnce apps from attacker-planted ones during any subsequent investigation.

Step 2: Review software inventory and authorization — verify that all ClickOnce applications in your environment are authorized, sourced from known-good deployment servers, and listed in your software inventory; flag any .appref-ms files not traceable to an approved application (CIS 2.1; CIS 2.3: Address Unauthorized Software)

NIST Phase: Preparation

Reference: NIST 800-61r3 §2 — Preparation: maintaining accurate software inventory as a prerequisite for identifying malicious ClickOnce artifacts during triage

Controls: CIS 2.1 (Establish and Maintain a Software Inventory), CIS 2.3 (Address Unauthorized Software), CIS 2.2 (Ensure Authorized Software is Currently Supported)

Compensating: Parse each .appref-ms file (they are plain text) to extract the deployment manifest URL: `Select-String -Path '*.appref-ms' -Pattern 'http'`. Cross-reference extracted URLs against an approved-server allowlist maintained in a simple CSV. Any URL resolving to a non-corporate, non-vendor domain should be treated as a priority investigation item. Use osquery with the query `SELECT name, path, install_date FROM apps WHERE name LIKE '%appref%'` on supported endpoints.

Evidence: Before flagging or removing any unauthorized .appref-ms file, capture: (1) the full file contents including the embedded deployment manifest URL, (2) file metadata — creation time, last-modified time, and owning user account via `Get-Item` — and (3) the Windows Prefetch entry for dfsvc.exe (`C:\Windows\Prefetch\DFSVC.EXE-*.pf`) which records the last eight execution timestamps. These artifacts establish whether a suspicious .appref-ms was recently executed, not merely present.

Step 3: Audit update server integrity — for every authorized ClickOnce application, confirm the deployment manifest's update URL resolves to an organization-controlled or vendor-controlled server; any manifest pointing to an external or unrecognized host is a priority investigation item (NIST AC-20: Use Of External Systems)

NIST Phase: Detection Analysis

Reference: NIST 800-61r3 §3.2 — Detection and Analysis: analyzing deployment manifests for attacker-controlled update URLs constitutes threat indicator analysis specific to ClickOnce's auto-update abuse vector

Controls: NIST AC-20 (Use Of External Systems)

Compensating: Extract and resolve every update URL from .appref-ms and .application manifest files using PowerShell: `[xml](Get-Content app.application) | Select-Xml -XPath '//deployment/@install'`. Pipe extracted hostnames through `nslookup` and compare against a blocklist of known-malicious hosting providers (GitHub raw, Pastebin, Discord CDN, and similar platforms documented in threat reporting as ClickOnce payload hosts). Capture DNS query logs from your router or Windows DNS debug log for historical resolution of these hostnames.

Evidence: Before any network-level blocking action: (1) capture full DNS resolution history for each manifest URL from Windows DNS Client event log (Event ID 3008 in Microsoft-Windows-DNS-Client/Operational) or from your perimeter DNS server query logs; (2) use Wireshark or `netstat -ano` correlated with `Get-Process` to capture any live outbound connections from dfsvc.exe to the update URL at the moment of discovery — this establishes C2 or payload-pull activity; (3) preserve the raw .application manifest file as a forensic artifact before any modification.

Step 4: Configure application control and process monitoring — restrict or block dfsvc.exe and rundll32.exe from spawning unexpected child processes; alert on rundll32.exe executing ClickOnce payloads outside approved software paths; apply allowlisting where feasible (NIST AC-3: Access Enforcement; NIST SI-4 equivalent behavioral monitoring — note: SI-4 is not in the provided knowledge base control set; reference NIST SP 800-53 Rev. 5 SI-4 directly for network/system monitoring guidance)

NIST Phase: Containment

Reference: NIST 800-61r3 §3.3 — Containment Strategy: restricting dfsvc.exe and rundll32.exe child-process spawning limits attacker ability to stage payloads via ClickOnce's trusted execution path without disrupting all ClickOnce functionality

Controls: NIST AC-3 (Access Enforcement), NIST AC-6 (Least Privilege), CIS 4.4 (Implement and Manage a Firewall on Servers), CIS 4.5 (Implement and Manage a Firewall on End-User Devices)

Compensating: Deploy Sysmon with a configuration rule alerting on Event ID 1 (Process Create) where ParentImage ends in 'dfsvc.exe' or 'rundll32.exe' and Image does not match your approved ClickOnce application paths. Use the free Sysmon config from SwiftOnSecurity as a baseline, adding: `dfsvc.exe`. On Windows 10/11 Pro and above, configure Windows Defender Application Control (WDAC) publisher rules to block unsigned binaries launched from `%LOCALAPPDATA%\Apps\2.0\` — the default ClickOnce installation directory — without an approved publisher certificate.

Evidence: Before applying any process-blocking rule that could terminate active dfsvc.exe or rundll32.exe instances: (1) acquire a full memory image of any currently running dfsvc.exe or rundll32.exe process using ProcDump (`procdump -ma dfsvc.exe dfsvc_mem.dmp`) to capture in-memory payload staging; (2) record all active network connections from these processes via `Get-NetTCPConnection | Where-Object {$_.OwningProcess -eq }`; (3) document the full process tree using `Get-WmiObject Win32_Process | Select-Object ProcessId, ParentProcessId, CommandLine` before any blocking rule disrupts the chain.

Step 5: Enforce least privilege and audit startup persistence locations — review user startup directories, registry run keys, and scheduled tasks for unauthorized .appref-ms entries; remove any not tied to approved applications (NIST AC-6: Least Privilege; T1547.001/T1053.005 mapped persistence locations)

NIST Phase: Eradication

Reference: NIST 800-61r3 §3.4 — Eradication: removing ClickOnce-based persistence mechanisms from startup directories, registry run keys, and scheduled tasks eliminates the attacker's ability to survive reboots via the auto-update pull mechanism

Controls: NIST AC-6 (Least Privilege), NIST AC-2 (Account Management), CIS 5.4 (Restrict Administrator Privileges to Dedicated Administrator Accounts)

Compensating: Query all three persistence vectors from an elevated PowerShell session: (1) Startup folder: `Get-ChildItem -Path 'C:\Users\*\AppData\Roaming\Microsoft\Windows\Start Menu\Programs\Startup\' -Filter *.appref-ms -Recurse`; (2) Registry run keys: `Get-ItemProperty 'HKCU:\Software\Microsoft\Windows\CurrentVersion\Run' | Select-Object * | Where-Object {$_ -like '*.appref-ms*'}`; (3) Scheduled tasks: `Get-ScheduledTask | Where-Object {$_.Actions.Execute -like '*appref-ms*' -or $_.Actions.Arguments -like '*dfsvc*'}`. Log all findings before deletion.

Evidence: Before removing any .appref-ms persistence entry: (1) capture the full registry hive containing the run key using `reg export HKCU\Software\Microsoft\Windows\CurrentVersion\Run runkeys_backup.reg`; (2) export scheduled task XML definitions via `Export-ScheduledTask -TaskName '' | Out-File task_backup.xml`; (3) hash and preserve any .appref-ms files slated for deletion using `Get-FileHash -Algorithm SHA256`; (4) check Windows Security Event Log for Event ID 4698 (Scheduled Task Created) and Event ID 4702 (Scheduled Task Updated) to identify when the persistence was established and under which account — this timestamps the initial compromise window.

Step 6: Update threat model and detection rules — add ClickOnce LotL abuse to your threat register, mapped to MITRE T1218.011, T1547, T1566.002, and T1105; tune EDR and SIEM rules to alert on dfsvc.exe network connections to non-approved hosts and .appref-ms file creation outside expected software directories

NIST Phase: Post Incident

Reference: NIST 800-61r3 §4 — Post-Incident Activity: updating the threat register and operationalizing new detection logic from a ClickOnce LotL campaign prevents recurrence and improves detection of future ClickOnce-based delivery chains

Controls: CIS 7.1 (Establish and Maintain a Vulnerability Management Process), CIS 7.2 (Establish and Maintain a Remediation Process), NIST AU-2 (Event Logging), NIST AU-6 (Audit Record Review, Analysis, And Reporting)

Compensating: Author Sigma rules targeting: (1) dfsvc.exe initiating outbound TCP connections (Sysmon Event ID 3, NetworkConnect, Image ends with 'dfsvc.exe', DestinationHostname not in approved allowlist); (2) .appref-ms file creation outside `%LOCALAPPDATA%\Apps\2.0\` (Sysmon Event ID 11, TargetFilename ends with '.appref-ms', path not matching approved directory). Submit both rules to the community Sigma repository for peer review. Use YARA to scan endpoint file systems for .appref-ms files containing URLs resolving to public hosting platforms (GitHub raw, Discord CDN, Pastebin).

Evidence: This step does not alter live system state; no volatile capture prerequisite applies. However, ground all new detection rules in the specific indicators documented during the active investigation: the exact update URLs extracted from malicious manifests, the child-process command lines spawned by dfsvc.exe, and the scheduled task names or registry value names used for persistence — generic rules that do not incorporate these campaign-specific IOCs will generate excessive false positives against legitimate ClickOnce deployments.

Step 7: Communicate findings — brief leadership on whether ClickOnce is present in the environment and what the silent-execution, no-UAC-prompt characteristic means for user-level attackers gaining a foothold without IT visibility

NIST Phase: Post Incident

Reference: NIST 800-61r3 §4 — Post-Incident Activity: communicating ClickOnce exposure findings to leadership fulfills the reporting and lessons-learned obligations of the IR lifecycle and enables risk-acceptance or remediation decisions at the appropriate authority level

Controls: NIST AC-1 (Policy And Procedures)

Compensating: Prepare a one-page executive summary quantifying: (1) number of endpoints with .appref-ms files present, (2) number of those files not traceable to an approved application, (3) whether dfsvc.exe was observed making outbound connections to non-approved hosts, and (4) the specific risk — that ClickOnce payloads execute without UAC prompts and bypass standard email attachment filters, meaning a user clicking a spearphish link is sufficient for persistence. Frame the decision as: permit ClickOnce with controls in place, restrict to an approved list, or block entirely via WDAC/AppLocker.

Evidence: No live system state is altered by this step; no volatile capture prerequisite applies. Attach the asset and software inventory outputs from Steps 1 and 2, the manifest URL audit from Step 3, and the persistence-location findings from Step 5 as supporting evidence appendices to the leadership briefing — these provide the factual basis for the risk quantification and prevent the briefing from relying on estimates.

Step 8: Monitor CrowdStrike's blog series — Part 1 and Part 2 are published; watch for additional installments, vendor mitigations from Microsoft, or CISA advisories referencing this technique chain

NIST Phase: Post Incident

Reference: NIST 800-61r3 §4 — Post-Incident Activity: continuous monitoring of threat intelligence sources for evolving ClickOnce weaponization techniques ensures detection and response posture keeps pace with attacker capability development

Controls: CIS 7.1 (Establish and Maintain a Vulnerability Management Process)

Compensating: Configure free RSS feeds or email alerts for: (1) CrowdStrike blog (adversary-intel tag), (2) CISA Known Exploited Vulnerabilities catalog updates, and (3) Microsoft Security Response Center advisories filtered on 'ClickOnce' or 'dfsvc'. Assign a named team member on a weekly rotation to review and triage new ClickOnce-related intelligence. Document any new IOCs (URLs, file hashes, infrastructure) from subsequent CrowdStrike installments directly into your Sigma rules and YARA signatures without waiting for a new incident to trigger the update cycle.

Evidence: No live system state is altered by this step; no volatile capture prerequisite applies. Maintain a running intelligence log that cross-references new CrowdStrike, Microsoft, or CISA reporting against the specific update URLs, payload hashes, and process-tree patterns documented during your own environment's audit — this enables rapid

determination of whether new published IOCs match artifacts already observed internally.

Detection Guidance

Detection centers on process lineage, network behavior of ClickOnce components, and file system artifacts. Key signals: (1) dfsvc.exe initiating outbound HTTP/HTTPS connections to hosts not matching your approved ClickOnce deployment servers, correlate against NIST AU-2 (Audit Events) and SI-4 (Information System Monitoring) to ensure network connection events are logged and reviewed; (2) rundll32.exe spawning child processes with ClickOnce-associated command-line patterns, particularly invocations referencing .appref-ms files or dfshim.dll; (3) .appref-ms file creation or modification in user profile directories, startup folders, or temp paths outside expected software installer activity via file system analysis through EDR or Sysmon; (4) scheduled tasks or registry run keys (HKCU\Software\Microsoft\Windows\CurrentVersion\Run) referencing dfsvc.exe or .appref-ms paths (T1547.001, T1053.005), hunt these via endpoint telemetry and Windows Event ID 4698 (scheduled task created); (5) process execution chains where a user-level process launches dfsvc.exe, which then performs file writes followed by outbound connections, consistent with a payload staging sequence (T1105: Ingress Tool Transfer). Log sources: Windows Security Event Log (process creation, scheduled task events), Sysmon EventIDs 1 (process create), 3 (network connect), 11 (file create), EDR process tree telemetry, proxy/DNS logs for dfsvc.exe outbound destinations. Hunt hypothesis: 'dfsvc.exe or rundll32.exe initiates a network connection to a host not in the approved ClickOnce server list within 60 seconds of a .appref-ms file being written to disk.' Apply NIST AC-2 (Account Management) and AU-6 (Audit Record Review, Analysis, and Reporting) to detect user-context ClickOnce persistence that does not require elevation, making it invisible to controls that watch only for privileged process behavior. Review AU-6 cadence to ensure ClickOnce-related process and network events are included in routine log review.

Indicators of Compromise

Type	Value	Context	Confidence
TOOL	`dfsvc.exe`	dfsvc.exe (Windows ClickOnce deployment service host) leveraged via malicious or hijacked .appref-ms file to silently execute attacker-controlled ClickOnce payloads without UAC elevation	HIGH
TOOL	`rundll32.exe`	rundll32.exe leveraged via ClickOnce application invocation to execute unsigned processes and establish persistence while evading security controls designed to detect unsigned binary execution.	HIGH
TOOL	`.appref-ms files`	.appref-ms ClickOnce application reference files used as the delivery artifact for initial access and persistence; written to startup locations or distributed via phishing to trigger silent payload execution	HIGH

Type	Value	Context	Confidence
URL	`Pending — refer to CrowdStrike blog Part 1 and Part 2 for published indicators`	CrowdStrike's two-part blog series may contain specific deployment server URLs, payload hashes, or additional file-based indicators not reproduced in the item data provided; consult source URLs directly	LOW

Framework Mappings

MITRE-ATTACK

- **T1547** — Boot or Logon Autostart Execution
- **T1566.002** — Spearphishing Link
- **T1195** — Supply Chain Compromise
- **T1053.005** — Scheduled Task
- **T1218.011** — Rundll32
- **T1036** — Masquerading
- **T1547.001** — Registry Run Keys / Startup Folder
- **T1105** — Ingress Tool Transfer
- **T1204.002** — Malicious File
- **T1072** — Software Deployment Tools

NIST-800-53R5

- **CM-7** — Least Functionality
- **SI-3** — Malicious Code Protection
- **SI-4** — System Monitoring
- **AT-2** — Literacy Training and Awareness
- **SC-7** — Boundary Protection
- **SI-8** — Spam Protection
- **SA-9** — External System Services
- **SR-2** — Supply Chain Risk Management Plan
- **SR-3** — Supply Chain Controls and Processes
- **SI-7** — Software, Firmware, and Information Integrity
- **CA-7** — Continuous Monitoring
- **CM-3** — Configuration Change Control
- **IR-5** — Incident Monitoring

OWASP-TOP10-2021

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

CIS-V8

- **2.5** — Allowlist Authorized Software
- **2.6** — Allowlist Authorized Libraries

NIST-CSF-2

- **DE.AE-08** — Incidents are declared when adverse events meet the defined incident criteria

MITRE ATT&CK Mapping

Technique ID	Technique Name	Tactic
T1547	Boot or Logon Autostart Execution	Persistence
T1566.002	Spearphishing Link	Initial-Access
T1195	Supply Chain Compromise	Initial-Access
T1053.005	Scheduled Task	Execution
T1218.011	Rundll32	Defense-Evasion
T1036	Masquerading	Defense-Evasion
T1547.001	Registry Run Keys / Startup Folder	Persistence
T1105	Ingress Tool Transfer	Command-And-Control
T1204.002	Malicious File	Execution
T1072	Software Deployment Tools	Execution

Sources

Source	URL	Tier
Blog	https://www.crowdstrike.com/en-us/blog/new-abuse-of-the-clickonce-t...	T3
	https://www.crowdstrike.com/en-us/blog/crowdstrike-falcon-scores-10...	T3
	https://www.crowdstrike.com/en-us/blog/consolidating-cybersecurity-...	T3
	https://www.crowdstrike.com/en-us/blog/reasons-why-nonprofits-are-t...	T3
New Abuse of the ClickOnce Technology: Part 1 - CrowdStrike	https://www.crowdstrike.com/en-us/blog/new-abuse-of-the-clickonce-t...	T3

Organizations should conduct their own validation and consult qualified security professionals before taking action based on this report. Tech Jacks Solutions is not liable for any damages resulting from the use of this information.

Generated 2026-06-27 18:14 UTC by TJS Security Command Center