INTELLIGENCE BRIEFING
Security Command Center
TLP:CLEAR
2026-06-20 06:52 UTC

ClickOnce Weaponization: Microsoft Deployment Tool Abused for Privilege-Free Malware Delivery and Persistence

SECURITY ANALYSIS | HIGH | CVSS 7.5

SCC Item ID	SCC-STY-2026-0227
Type	Security Analysis
Severity	HIGH
CVSS Base Score	7.5
Affected Products	Microsoft ClickOnce (Windows / .NET), Visual Studio (all versions supporting ClickOnce deployment)
Discovery Source	Rss:T1 Threatintel

Executive Summary

CrowdStrike researchers have detailed how Microsoft's ClickOnce deployment framework, a legitimate .NET application delivery mechanism, can be weaponized to install and persist malware on Windows systems without requiring administrative privileges or triggering UAC prompts. The technique exploits ClickOnce's by-design behavior: manifest-driven payloads stored in user-writable AppData directories with built-in auto-update functionality, enabling attackers to deliver second-stage tools post-compromise through a single malicious link or email attachment. For organizations running Windows environments, this represents a persistent delivery channel that most endpoint and email security tooling has historically undertreated, requiring an intentional detection and policy response.

Technical Analysis

ClickOnce is a Microsoft deployment framework built into .NET and Visual Studio that allows applications to be installed and updated from remote URLs without elevated privileges. Legitimate use cases include enterprise line-of-business application distribution. The abuse path documented by CrowdStrike exploits three structural properties of the technology: first, ClickOnce installs into %APPDATA%\Local\Apps, a directory writable by standard user accounts, meaning no UAC elevation prompt appears; second, the auto-update mechanism allows an attacker-controlled manifest to silently swap in replacement payloads after initial execution, supporting multi-stage campaigns; third, the host process for ClickOnce is dfsvc.exe, a Microsoft-signed binary, so initial execution can be proxied through a trusted Windows process (T1218, System Binary Proxy Execution).

Delivery vectors include phishing emails carrying malicious .application manifest files (T1566), hyperlinks embedded in web pages or documents that trigger single-click installation, and direct URL-based staging of remote payloads (T1105, Ingress Tool Transfer). Persistence is achieved through ClickOnce's native auto-update scheduling, which maps to T1547 (Boot or Logon Autostart Execution). Application command-and-control communication fits T1071.001 (Application Layer Protocol: Web Protocols), as update checks and payload retrieval occur over HTTP/HTTPS to attacker-controlled infrastructure.

The core defensive gap is detection asymmetry: security tooling and analyst workflows have historically applied greater scrutiny to .exe and .msi files than to .application manifest files, creating a coverage blind spot at both the email gateway and endpoint layers. CWE-829 (Inclusion of Functionality from Untrusted Control Sphere) and CWE-494 (Download of Code Without Integrity Check) apply because ClickOnce's design allows manifest-defined remote payloads to execute with only optional, not enforced, code signing. No specific threat actor has been attributed to active exploitation in available source material; CrowdStrike's reporting frames this as a technique disclosure rather than a named campaign post-mortem.

Action Checklist

1. Step 1: Assess exposure, audit whether Windows environments in your organization have ClickOnce enabled via dfsvc.exe and whether .application file execution is possible from user accounts without explicit policy restriction
2. Step 2: Review controls, verify that email security gateways inspect and quarantine .application attachments; confirm EDR coverage includes behavioral rules for dfsvc.exe spawning child processes or writing to AppData execution paths; cross-reference against CIS 2.1 (Inventory and Control of Software Assets) and CIS 8.3 (Address Unauthorized Software) to ensure application control policies block unsigned or non-allowlisted executables from AppData paths, and CIS 4.4 (Implement and Manage a Firewall on Servers) and CIS 4.5 (Implement and Manage a Firewall on End-User Devices) to ensure outbound connections from AppData-resident processes are blocked or alerted
3. Step 3: Audit logging for ClickOnce execution paths, enable process creation logging to capture dfsvc.exe invocations; review AU-2 (Event Logging) coverage to confirm .application execution events are captured in your SIEM; apply AU-12 (Audit Record Generation) to ensure endpoint agents generate records for AppData write and execution events
4. Step 4: Update threat model, add T1218 (System Binary Proxy Execution via dfsvc.exe), T1547 (Boot or Logon Autostart Execution via ClickOnce auto-update), and T1566 (.application phishing delivery) to your threat register; update detection use cases and hunting hypotheses accordingly
5. Step 5: Communicate findings, brief leadership on the specific risk: attackers can install and update malware on any standard Windows user workstation with a single click, no admin prompt, and no traditional installer artifact; frame this as a detection gap requiring tooling and policy remediation, not a patch-available vulnerability
6. Step 6: Monitor developments, track CrowdStrike's Part Two publication and any follow-on Microsoft guidance on ClickOnce policy controls; watch for CISA advisories referencing dfsvc.exe abuse or .application-based phishing campaigns

IR / Forensic Enrichment

Triage Priority	URGENT
Escalation Criteria	Escalate immediately to IR leadership and legal/compliance if Sysmon EventID 1 or Windows Security Event ID 4688 confirms dfsvc.exe has spawned an unexpected child process on any endpoint, or if any `%LOCALAPPDATA%\Apps\2.0\` directory contains executables not traceable to an authorized internal ClickOnce deployment, as this indicates active exploitation requiring breach notification assessment under applicable data protection regulations.
Recovery Notes	After containment of any confirmed ClickOnce-delivered malware, verify full removal by recursively deleting the attacker-controlled ClickOnce application directory under `%LOCALAPPDATA%\Apps\2.0\`, removing any associated registry Run keys (HKCU\Software\Microsoft\Windows\CurrentVersion\Run) pointing to AppData paths, and clearing ClickOnce deployment metadata from `%LOCALAPPDATA%\Microsoft\Windows\INetCache` and the ClickOnce application cache. Monitor the affected endpoint for 30 days post-eradication using Sysmon EventID 1 alerts on dfsvc.exe child process creation and Sysmon EventID 11 alerts on new file creation in `%LOCALAPPDATA%\Apps\2.0\`, as ClickOnce's built-in auto-update mechanism may attempt reinfection if the original manifest URL remains reachable. Confirm that the phishing delivery vector (email gateway quarantine of .application attachments or browser download blocking) is verified closed before returning the host to production.
Forensic Artifacts	ClickOnce application cache directory at `%LOCALAPPDATA%\Apps\2.0\` — contains the deployed malicious payload DLLs or EXEs organized by application identity GUID, manifest files (.manifest, .application) revealing the attacker-controlled deployment URL, and version metadata enabling attribution of the specific ClickOnce package delivered | Windows Security Event Log Event ID 4688 (Process Creation) records where `NewProcessName` is `C:\Windows\System32\dfsvc.exe` and `ParentProcessName` is a browser, email client (outlook.exe, thunderbird.exe), or file manager — establishing the initial .application file execution chain and the delivery vector | Sysmon EventID 13 (RegistryValue Set) records for HKCU\Software\Microsoft\Windows\CurrentVersion\Run or HKCU\Software\Microsoft\Windows NT\CurrentVersion\Windows\Load containing paths to `%LOCALAPPDATA%\Apps\2.0\` — confirming ClickOnce auto-update persistence registration without admin privileges | Network connection records (Sysmon EventID 3 or `Get-NetTCPConnection`) showing outbound HTTP/HTTPS connections from processes executing within `%LOCALAPPDATA%\Apps\2.0\` to the attacker-controlled ClickOnce deployment server URL embedded in the .application manifest — this URL is the command-and-control or update channel | ClickOnce deployment manifest files (*.application and *.manifest XML) recovered from `%LOCALAPPDATA%\Apps\2.0\` or from the user's Downloads, Temp, or email attachment cache directories — these contain the attacker's signing certificate thumbprint (or lack thereof if trust-not-required), the deployment provider URL, and the embedded payload identity, all of which are high-fidelity IOCs for threat intelligence sharing

Per-Action IR Details

Step 1: Assess exposure — audit whether Windows environments in your organization have ClickOnce enabled via dfsvc.exe and whether .application file execution is possible from user accounts without explicit policy restriction

NIST Phase: Preparation

Reference: NIST 800-61r3 §2 — Preparation: Establishing IR capability through asset and configuration baseline assessment

Controls: CIS 1.1 (Establish and Maintain Detailed Enterprise Asset Inventory), CIS 2.1 (Establish and Maintain a Software Inventory), CIS 2.2 (Ensure Authorized Software is Currently Supported), NIST AC-6 (Least Privilege)

Compensating: Run the following on each endpoint or via GPO startup script to enumerate dfsvc.exe presence and .application file associations: `Get-Item 'C:\Windows\System32\dfshim.dll' -ErrorAction SilentlyContinue` and `cmd /c assoc .application`. Use osquery with `SELECT * FROM file WHERE path = 'C:\Windows\System32\dfsvc.exe';` to confirm presence across a fleet. Document results in a shared spreadsheet grouped by OU.

Evidence: No live state is altered by this assessment step; volatile capture is not required. However, document the current state of .application file associations and dfsvc.exe presence before any policy changes, as this establishes the pre-remediation baseline: output of `assoc .application`, output of `ftype Application.Manifest`, and list of AppData\Local\Apps\2.0\ directories containing deployed ClickOnce payloads (`Get-ChildItem $env:LOCALAPPDATA\Apps\2.0 -Recurse -ErrorAction SilentlyContinue`).

Step 2: Review controls — verify that email security gateways inspect and quarantine .application attachments; confirm EDR coverage includes behavioral rules for dfsvc.exe spawning child processes or writing to AppData execution paths; cross-reference against CIS 4.4 (Implement and Manage a Firewall on Servers) and CIS 4.5 (Implement and Manage a Firewall on End-User Devices) to ensure outbound connections from AppData-resident processes are blocked or alerted

NIST Phase: Preparation

Reference: NIST 800-61r3 §2 — Preparation: Validating detective and preventive controls before an incident occurs

Controls: CIS 4.4 (Implement and Manage a Firewall on Servers), CIS 4.5 (Implement and Manage a Firewall on End-User Devices), NIST AC-4 (Information Flow Enforcement), NIST SI-3 — note: SI-3 (Malicious Code Protection) is not present in the knowledge base reference; omitting to avoid unverified citation, NIST AU-2 (Event Logging)

Compensating: Deploy Sysmon with a configuration that includes rules for dfsvc.exe as a ParentImage (EventID 1, ProcessCreate) and any write events to `%LOCALAPPDATA%\Apps\2.0\` (EventID 11, FileCreate). Use Windows Firewall with Advanced Security (`netsh advfirewall`) to create an outbound block rule for processes executing from `%LOCALAPPDATA%\Apps\2.0\*`. Verify email gateway quarantine of .application files by sending a benign test attachment with that extension and confirming interception.

Evidence: No live state is altered by this control-review step; volatile capture is not required. Before any firewall rule changes that would terminate active connections from AppData-resident processes, capture: `Get-NetTCPConnection | Where-Object {$_.OwningProcess -in (Get-Process | Where-Object {$_.Path -like '*AppData*'}).Id}` to document any currently active suspicious outbound connections originating from ClickOnce-deployed payloads in `%LOCALAPPDATA%\Apps\2.0\`.

Step 3: Audit logging for ClickOnce execution paths — enable process creation logging to capture dfsvc.exe invocations; review AU-2 (Event Logging) coverage to confirm .application execution events are captured in your SIEM; apply AU-12 (Audit Record Generation) to ensure endpoint agents generate records for AppData write and execution events

NIST Phase: Detection Analysis

Reference: NIST 800-61r3 §3.2 — Detection and Analysis: Ensuring log sources and detection coverage are sufficient to identify ClickOnce-based malware delivery

Controls: NIST AU-2 (Event Logging), NIST AU-12 (Audit Record Generation), NIST AU-3 (Content Of Audit Records), CIS 8.2 (Collect Audit Logs)

Compensating: Enable Windows Security Event ID 4688 (Process Creation) with command-line auditing via GPO (`Computer Configuration > Windows Settings > Security Settings > Advanced Audit Policy > Detailed Tracking > Audit Process Creation`). Deploy Sysmon EventID 1 with the following config entry: `C:\Windows\System32\dfsvc.exe` to capture all child processes spawned by the ClickOnce deployment service. Forward Sysmon logs to a central syslog server using NXLog (free edition) if no SIEM is available.

Evidence: Before enabling new logging policies that may alter system configuration state, capture the current audit policy baseline: `auditpol /get /category:* > auditpol_baseline.txt`. Document existing Sysmon configuration if deployed: `sysmon -c` output. Identify gaps by reviewing the last 72 hours of Security Event Log for any existing Event ID 4688 records where `ProcessName` contains `dfsvc.exe` or `NewProcessName` paths include `AppData\Local\Apps\2.0`,

establishing whether any ClickOnce invocations already occurred without detection.

Step 4: Update threat model — add T1218 (System Binary Proxy Execution via dfsvc.exe), T1547 (Boot or Logon Autostart Execution via ClickOnce auto-update), and T1566 (.application phishing delivery) to your threat register; update detection use cases and hunting hypotheses accordingly

NIST Phase: Preparation

Reference: NIST 800-61r3 §2 — Preparation: Maintaining an updated threat model and detection hypothesis library as a foundational IR capability

Controls: NIST AU-6 (Audit Record Review, Analysis, And Reporting)

Compensating: Translate the three ATT&CK techniques into concrete Sigma rules for free deployment: (1) detect `dfsvc.exe` spawning non-Microsoft child processes; (2) detect new registry Run key entries pointing to `%LOCALAPPDATA%\Apps\2.0\` paths (Sysmon EventID 13, RegistryEvent); (3) detect email delivery of `.application` attachments via mail server logs filtered on MIME type `application/x-ms-application`. Publish these as hunting hypotheses in a shared wiki or Git repository accessible to the 2-person team.

Evidence: This step does not alter live system state; volatile capture is not required. However, before updating detection rules that will suppress or reclassify existing alerts, export the current SIEM or event log detection baseline: document all existing rules referencing `dfsvc.exe`, `dfshim.dll`, or `AppData\Local\Apps` so that new detections can be validated against prior activity and true-positive/false-positive rates can be measured post-deployment.

Step 5: Communicate findings — brief leadership on the specific risk: attackers can install and update malware on any standard Windows user workstation with a single click, no admin prompt, and no traditional installer artifact; frame this as a detection gap requiring tooling and policy remediation, not a patch-available vulnerability

NIST Phase: Post Incident

Reference: NIST 800-61r3 §4 — Post-Incident Activity: Communicating lessons learned and identified capability gaps to leadership to drive policy and resource decisions

Controls: NIST AC-1 (Policy And Procedures)

Compensating: Prepare a one-page executive brief using concrete metrics: number of endpoints with dfsvc.exe present, count of user accounts capable of executing .application files, and the detection gap window (time since ClickOnce was available vs. time first detection rule was in place). Use a before/after table showing current detection coverage vs. proposed Sysmon/firewall compensating controls. Deliver verbally with the written brief as leave-behind; no specialized tooling required.

Evidence: This step does not alter live system state; no volatile capture required. Attach to the brief the output of the exposure audit from Step 1 (dfsvc.exe presence, .application association state, AppData\Local\Apps\2.0 directory inventory) as quantified evidence of blast radius, and include any Sysmon or Event ID 4688 findings from Step 3 showing whether ClickOnce invocations have already occurred in the environment.

Step 6: Monitor developments — track CrowdStrike's Part Two publication and any follow-on Microsoft guidance on ClickOnce policy controls; watch for CISA advisories referencing dfsvc.exe abuse or .application-based phishing campaigns

NIST Phase: Post Incident

Reference: NIST 800-61r3 §4 — Post-Incident Activity: Integrating external threat intelligence and vendor guidance into updated IR and detection posture

Controls: NIST AU-13 (Monitoring For Information Disclosure), CIS 7.1 (Establish and Maintain a Vulnerability Management Process), CIS 7.2 (Establish and Maintain a Remediation Process)

Compensating: Create a free RSS or email alert feed monitoring: (1) CrowdStrike blog via RSS for the ClickOnce Part Two follow-up; (2) CISA Known Exploited Vulnerabilities catalog (https://www.cisa.gov/known-exploited-vulnerabilities-catalog) filtered for 'ClickOnce' or 'dfsvc'; (3) Microsoft Security Response Center blog for any ClickOnce policy enforcement updates. Assign one team member to review these feeds weekly and update the Sigma rules and threat register within 48 hours of a new publication. Track in a shared

changelog.

Evidence: This step does not alter live system state; no volatile capture required. Maintain a running log of intelligence sources checked, dates reviewed, and any indicators extracted (new .application delivery domains, ClickOnce manifest hashes, C2 URLs embedded in observed malicious manifests) so that threat intelligence can be operationalized into detection rules without restarting the research process from scratch.

Detection Guidance

Primary hunting focus is dfsvc.exe process behavior. Analysts should alert on dfsvc.exe spawning child processes other than expected ClickOnce update workers, dfsvc.exe making outbound network connections to non-Microsoft or unclassified domains, and any executable written to %LOCALAPPDATA%\Apps\ paths that subsequently executes. Email gateway logs should be reviewed for .application file attachments or hyperlinks whose URLs end in .application, these are rarely legitimate in most enterprise environments and warrant immediate quarantine and investigation.

For SIEM-based detection, query process creation events where ParentImage contains dfsvc.exe and ChildImage is cmd.exe, powershell.exe, wscript.exe, or any binary outside system32. Combine with network telemetry: alert on HTTP/HTTPS requests from processes resident in %APPDATA% or %LOCALAPPDATA%\Apps\ to external IPs or domains not in your approved software update allow-list (supports AC-4, Information Flow Enforcement).

Persistence hunting: ClickOnce auto-update tasks register in the Windows Task Scheduler or AppData-local run keys. Query scheduled tasks and HKCU run key entries for executables under AppData paths. Cross-reference against CIS 1.1 (Establish and Maintain Detailed Enterprise Asset Inventory), any executable present in AppData that does not correspond to an inventoried ClickOnce-deployed application is a high-confidence hunt lead.

For environments using application allowlisting, verify that policies block unsigned or non-allowlisted executables from AppData execution paths entirely (supports AC-3, Access Enforcement and AC-6, Least Privilege). MITRE D3FEND countermeasures applicable: D3-SFA (System File Analysis) for monitoring AppData executable writes, D3-UAP (User Account Permissions) to restrict execution from user-writable directories, and D3-PAM (Process Analysis and Monitoring) to detect anomalous process behavior initiated through dfsvc.exe.

Indicators of Compromise

Type	Value	Context	Confidence
TOOL	`dfsvc.exe`	dfsvc.exe (Microsoft ClickOnce host process) leveraged via malicious .application manifest delivery to execute attacker-controlled payloads from user-writable AppData directories without administrative privileges	HIGH

Type	Value	Context	Confidence
URL	`Pending — refer to CrowdStrike blog posts (Part One and Part Two) for any published payload URLs or C2 indicators`	CrowdStrike's two-part technical disclosure may include specific malicious .application manifest URLs or staging infrastructure observed during research; values were not available in the provided source material	LOW

Framework Mappings

MITRE-ATTACK

- **T1547** — Boot or Logon Autostart Execution
- **T1204.002** — Malicious File
- **T1105** — Ingress Tool Transfer
- **T1218** — System Binary Proxy Execution
- **T1071.001** — Web Protocols
- **T1566** — Phishing

NIST-800-53R5

- **CM-7** — Least Functionality
- **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
- **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
- **14.2** — Train Workforce Members to Recognize Social Engineering Attacks
- **8.2** — Collect Audit Logs
- **5.4** — Restrict Administrator Privileges to Dedicated Administrator Accounts

HIPAA-SECURITY

- **164.308(a)(5)(i)** — Security Awareness and Training

ISO-27001-2022

- **A.5.34** — Privacy and protection of personal information

NIST-CSF-2

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

MITRE ATT&CK Mapping

Technique ID	Technique Name	Tactic
T1547	Boot or Logon Autostart Execution	Persistence
T1204.002	Malicious File	Execution
T1105	Ingress Tool Transfer	Command-And-Control
T1218	System Binary Proxy Execution	Defense-Evasion
T1071.001	Web Protocols	Command-And-Control
T1566	Phishing	Initial-Access

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/new-abuse-of-the-clickonce-t...	T3
	https://www.crowdstrike.com/en-us/blog/new-abuse-of-the-clickonce-t...	T3
	https://www.crowdstrike.com/en-us/blog/reasons-why-nonprofits-are-t...	T3
ClickOnce Deployment and Security - Visual Studio - Microsoft Learn	https://learn.microsoft.com/en-us/visualstudio/deployment/clickonce...	T1

Generated 2026-06-20 06:52 UTC by TJS Security Command Center