INTELLIGENCE BRIEFING
Security Command Center
TLP:CLEAR
2026-06-21 13:21 UTC

ClickOnce Weaponized: Microsoft's No-Admin Deployment Technology Becomes a Malware Delivery Channel

SECURITY ANALYSIS | MEDIUM | CVSS 5.0

SCC Item ID	SCC-STY-2026-0237
Type	Security Analysis
Severity	MEDIUM
CVSS Base Score	5.0
Affected Products	Microsoft ClickOnce, Windows, Visual Studio
Discovery Source	Rss:T1 Threatintel

Executive Summary

CrowdStrike researchers have documented a systematic method for weaponizing Microsoft ClickOnce, a legitimate Windows application deployment technology, to deliver malware without requiring administrative privileges. Because ClickOnce executes within user-space directories and bypasses the elevation prompts and installer telemetry that most endpoint defenses rely on, it represents a structural gap in many organizations' detection architectures rather than a patchable software flaw. This research signals a broader trend: attackers are increasingly targeting the seams between legitimate software deployment mechanisms and security tooling, making detection-coverage audits as important as patch management.

Technical Analysis

Microsoft ClickOnce was designed to simplify application deployment by allowing users to install and run Windows-based applications directly from a web browser or network share without elevated privileges. Execution occurs within %LocalAppData%\Apps\2.0, a user-writable directory that sits outside the path scrutiny most endpoint detection and response (EDR) platforms apply to traditional installer activity. CrowdStrike's research, beginning with Part 1 published on the CrowdStrike blog, documents how this architecture creates an abuse surface across several MITRE ATT&CK techniques.

The attack chain maps to a recognized set of TTPs. Initial access relies on user interaction with a malicious .application manifest file or URL, corresponding to T1204.002 (User Execution: Malicious File). The ClickOnce runtime, invoked through dfsvc.exe or a browser helper, fetches and deploys the payload, satisfying T1105 (Ingress Tool Transfer). Because the deployment binary can be signed or named to resemble legitimate software, T1036.005 (Masquerading: Match Legitimate Name or Location) applies. The runtime itself functions

as a living-off-the-land binary proxy, aligning with T1218 (System Binary Proxy Execution). Obfuscation of the embedded payload maps to T1027 (Obfuscated Files or Information), and the use of a trusted software distribution channel maps to T1072 (Software Deployment Tools).

The CWE-693 (Protection Mechanism Failure) classification is precise here: the issue is not a memory corruption bug or an authentication bypass. ClickOnce was designed to operate this way. Security controls that depend on elevation prompts, Windows Installer telemetry, or UAC events simply do not fire. Many EDR platforms lack behavioral rules tuned to ClickOnce-specific execution chains, and application allowlisting policies that permit dfsvc.exe by default provide no barrier.

Action Checklist

1. Step 1: Assess exposure, determine whether your environment permits ClickOnce application execution; query EDR telemetry for dfsvc.exe execution events and %LocalAppData%\Apps\2.0 write activity across endpoints
2. Step 2: Review controls, audit EDR and application control policies for coverage of user-space execution paths, specifically %LocalAppData%; verify whether your allowlisting solution (e.g., AppLocker, WDAC) has rules scoped to ClickOnce deployment directories; reference NIST CM-7 (Least Functionality) to restrict or prohibit ClickOnce where it serves no mission-essential purpose
3. Step 3: Validate logging, confirm audit logging captures process creation events for dfsvc.exe, msiexec.exe, and browser child processes; reference NIST AU-2 (Event Logging) and CIS 8.2 (Collect Audit Logs) to ensure user-space execution is within scope
4. Step 4: Update threat model, add ClickOnce-based delivery (T1218, T1204.002, T1072) to your threat register; update detection engineering backlogs to include behavioral rules for ClickOnce manifest fetches from external URLs and payload writes to %LocalAppData%\Apps\2.0
5. Step 5: Communicate findings, brief application owners and IT administrators on ClickOnce usage inventory; if no business-critical application requires ClickOnce, begin a controlled deprecation; reference NIST CM-7 for policy basis
6. Step 6: Monitor developments, track CrowdStrike's research series publication and incorporate any released detection logic or IOCs into your SIEM and EDR rule sets as they become available

IR / Forensic Enrichment

Triage Priority	STANDARD
Escalation Criteria	Escalate to urgent if EDR or Sysmon telemetry confirms dfsvc.exe executing a child process not signed by Microsoft, any PE file written to %LocalAppData%\Apps\2.0 that fails hash verification against known-good ClickOnce applications, or if a network connection from dfsvc.exe to an external non-Microsoft URL is observed — any of these conditions indicates active weaponized ClickOnce delivery and warrants immediate IR team engagement and potential breach notification assessment if user-accessible PII or PHI systems are in scope.

Recovery Notes	Because ClickOnce payloads execute entirely in user-space under %LocalAppData%\Apps\2.0 and require no administrative privileges, recovery must account for persistence mechanisms the delivered payload may have established in user-accessible locations: Run key entries under HKCU\Software\Microsoft\Windows\CurrentVersion\Run, scheduled tasks created without elevation, and additional binaries dropped in %AppData% or %Temp%. After removing confirmed malicious ClickOnce deployment directories and blocking dfsvc.exe lateral execution via AppLocker or WDAC, monitor Sysmon EID 1 and EID 11 for 30 days to detect reinfection via alternate delivery of a new .application manifest, particularly via phishing or compromised internal web servers. Verify integrity of the affected user profile by comparing %LocalAppData%\Apps\2.0 directory contents against the pre-incident inventory baseline captured in Step 1.
Forensic Artifacts	%LocalAppData%\Apps\2.0\ directory tree — contains the deployed ClickOnce application files including .exe, .dll, and .manifest files written by dfsvc.exe; timestamps and SHA-256 hashes establish when the weaponized payload was fetched and what was delivered | Sysmon Event ID 3 (Network Connection) logs for dfsvc.exe — captures the external URL from which the malicious .application manifest or payload was fetched, identifying attacker-controlled staging infrastructure specific to this ClickOnce campaign | Sysmon Event ID 1 (Process Create) logs showing dfsvc.exe as ParentImage — reveals any child processes spawned by the ClickOnce payload, which is the primary behavioral indicator distinguishing weaponized ClickOnce from legitimate deployment activity | HKCU\Software\Microsoft\Windows\CurrentVersion\Run and HKCU\Software\Microsoft\Windows\CurrentVersion\RunOnce registry keys — ClickOnce-delivered malware operating in user-space commonly establishes persistence here without requiring elevation, leaving a recoverable artifact even after the payload binary is removed | Browser history and download logs (Edge: %LocalAppData%\Microsoft\Edge\User Data\Default\History; Chrome: %LocalAppData%\Google\Chrome\User Data\Default\History) — ClickOnce delivery via phishing or malicious web page leaves a referrer URL and .application file download record that identifies the initial access vector and may link to the threat actor's infrastructure

Per-Action IR Details

Step 1: Assess exposure — determine whether your environment permits ClickOnce application execution; query EDR telemetry for dfsvc.exe execution events and %LocalAppData%\Apps\2.0 write activity across endpoints

NIST Phase: Preparation

Reference: NIST 800-61r3 §2 — Preparation: Assessing organizational posture and identifying exposure before an incident is declared

Controls: NIST CM-8 (System Component Inventory), 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 endpoint via PowerShell: `Get-ChildItem 'C:\Users\*\AppData\Local\Apps\2.0' -Recurse -ErrorAction SilentlyContinue | Select FullName, LastWriteTime` to identify existing ClickOnce deployment artifacts. Supplement with Sysmon Event ID 1 (Process Create) filtered on Image containing 'dfsvc.exe' using a Sigma rule targeting ParentImage and CommandLine fields. Use `wevtutil qe Security /q:"*[System[EventID=4688] and EventData[Data[@Name='NewProcessName'] and (Data='*dfsvc.exe')]]" /f:text` on hosts where Sysmon is unavailable.

Evidence: This step is read-only telemetry assessment and does not alter live host state; however, document baseline findings before any subsequent containment action. Capture: (1) list of all endpoints where dfsvc.exe has executed (Sysmon EID 1 or EDR process telemetry); (2) directory listings of %LocalAppData%\Apps\2.0 on affected hosts including file timestamps and SHA-256 hashes of any .exe, .dll, or .manifest files present; (3) network connection logs

associated with dfsvc.exe (Sysmon EID 3) showing external manifest fetch URLs — these are volatile and reflect the attacker-controlled staging infrastructure.

Step 2: Review controls — audit EDR and application control policies for coverage of user-space execution paths, specifically %LocalAppData%; verify whether your allowlisting solution (e.g., AppLocker, WDAC) has rules scoped to ClickOnce deployment directories; reference NIST CM-7 (Least Functionality) to restrict or prohibit ClickOnce where it serves no mission-essential purpose

NIST Phase: Preparation

Reference: NIST 800-61r3 §2 — Preparation: Evaluating and hardening preventive controls before or in parallel with an active detection posture

Controls: NIST CM-7 (Least Functionality), NIST CM-6 (Configuration Settings), CIS 4.6 (Securely Manage Enterprise Assets and Software), CIS 2.3 (Address Unauthorized Software)

Compensating: For teams without commercial application control: deploy AppLocker (built into Windows Enterprise/Education) with a Publisher rule that blocks any unsigned executable launching from `%LocalAppData%\Apps\2.0\*`. Export current AppLocker policy with `Get-AppLockerPolicy -Effective | ConvertTo-XML` to document baseline before changes. For WDAC, use the WDAC Wizard (free, Microsoft-provided) to create a policy that explicitly denies execution from user-writable ClickOnce deployment directories. Validate policy enforcement by attempting to launch a test ClickOnce app and confirming EID 8004 (AppLocker block) or EID 3076/3077 (WDAC audit/enforce) in the Application and Services Logs.

Evidence: Before modifying AppLocker or WDAC policies (which alter enforcement state), capture: (1) current effective AppLocker policy XML (`Get-AppLockerPolicy -Effective | ConvertTo-XML > applocker_baseline.xml`); (2) current WDAC policy in force (`CiTool.exe --list-policies`); (3) any existing EID 8003/8004 (AppLocker) or EID 3076/3077 (WDAC) events referencing dfsvc.exe or binaries under %LocalAppData%\Apps\2.0 — these pre-change events document what was already executing unchecked and constitute forensic proof of the detection gap.

Step 3: Validate logging — confirm audit logging captures process creation events for dfsvc.exe, msiexec.exe, and browser child processes; reference NIST AU-2 (Event Logging) and CIS 8.2 (Collect Audit Logs) to ensure user-space execution is within scope

NIST Phase: Preparation

Reference: NIST 800-61r3 §2 — Preparation: Ensuring log infrastructure is sufficient to support detection and forensic reconstruction of 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: Deploy Sysmon with SwiftOnSecurity's config (or a hardened derivative) and verify the following event types are generating: EID 1 (Process Create) for dfsvc.exe and any child processes; EID 3 (Network Connection) for dfsvc.exe fetching .application or .manifest files from external URLs; EID 11 (File Create) for writes to %LocalAppData%\Apps\2.0. Validate with: `wevtutil qe "Microsoft-Windows-Sysmon/Operational" /q:"*[System[EventID=1] and EventData[Data[@Name='Image'] and Data[contains(.,'dfsvc.exe')]]]" /c:5 /f:text`. If Sysmon is absent, enable Windows Audit Process Creation via GPO (Security Settings → Advanced Audit Policy → Detailed Tracking) and enable command-line auditing in the registry: `HKLM\SOFTWARE\Microsoft\Windows\CurrentVersion\Policies\System\Audit\ProcessCreationIncludeCmdLine_Enabled = 1`.

Evidence: This step validates logging coverage and does not alter live host state; no volatile capture prerequisite applies. Document: (1) current Sysmon configuration version and rule set hash to establish what was and was not captured prior to this review; (2) gaps identified — specifically whether EID 3 (network connection) was enabled for dfsvc.exe, since the manifest fetch to an attacker-controlled URL is the earliest detectable indicator of a ClickOnce weaponization attempt; (3) retention period and destination of existing Sysmon or Security logs to assess whether prior ClickOnce activity could be retroactively investigated.

Step 4: Update threat model — add ClickOnce-based delivery (T1218, T1204.002, T1072) to your threat register; update detection engineering backlogs to include behavioral rules for ClickOnce manifest fetches from external URLs and payload writes to %LocalAppData%\Apps\2.0

NIST Phase: Post Incident

Reference: NIST 800-61r3 §4 — Post-Incident Activity: Incorporating newly documented adversary technique into threat model and detection backlog to prevent recurrence

Controls: NIST IR-4 (Incident Handling), NIST IR-8 (Incident Response Plan), CIS 7.1 (Establish and Maintain a Vulnerability Management Process)

Compensating: Write Sigma rules (free, vendor-neutral) targeting: (1) dfsvc.exe spawning any child process (`ParentImage|endswith: 'dfsvc.exe'` AND `Image|endswith` any executable not signed by Microsoft); (2) file creation events under `%LocalAppData%\Apps\2.0` for PE files (.exe, .dll) written by dfsvc.exe. Publish rules to your SIEM or convert via `sigma convert` to Windows Event Log, Splunk, or Elastic format. Reference the CrowdStrike ClickOnce research series directly in rule metadata fields (Author, Reference, Description) so analysts have context when the rule fires. Use osquery to run a scheduled query: `SELECT name, path, pid FROM processes WHERE name = 'dfsvc.exe'` to catch runtime instances on hosts lacking EDR.

Evidence: No live host state is altered by this step; no volatile capture prerequisite applies. For threat model documentation, preserve: (1) the CrowdStrike research publication URL and publication date as the authoritative source for the technique; (2) any internal telemetry collected in Steps 1–3 showing historical dfsvc.exe execution — this serves as the empirical basis for threat register inclusion; (3) current detection rule set snapshot (Sigma rule files, SIEM query library export) dated before new rules are added, to enable a before/after coverage diff.

Step 5: Communicate findings — brief application owners and IT administrators on ClickOnce usage inventory; if no business-critical application requires ClickOnce, begin a controlled deprecation; reference NIST CM-7 for policy basis

NIST Phase: Post Incident

Reference: NIST 800-61r3 §4 — Post-Incident Activity: Communicating findings to stakeholders and driving policy-level remediation to eliminate the structural detection gap

Controls: NIST CM-7 (Least Functionality), NIST CM-3 (Configuration Change Control), CIS 2.2 (Ensure Authorized Software is Currently Supported), CIS 4.6 (Securely Manage Enterprise Assets and Software)

Compensating: Generate the ClickOnce usage inventory without enterprise tooling using: `Get-ChildItem 'C:\Users\*\AppData\Local\Apps\2.0' -Recurse -Include '*.application','*.manifest' | Select Directory, Name, LastWriteTime | Export-CSV clickonce_inventory.csv`. Share results with application owners via a structured briefing that maps each discovered ClickOnce application to a business owner, risk level, and deprecation feasibility. For the controlled deprecation, document the CM-3 change record before modifying AppLocker/WDAC policies or disabling the ClickOnce feature via Group Policy (User Configuration → Administrative Templates → Windows Components → Internet Explorer → Security Zones — or via WDAC publisher deny rules for dfsvc.exe).

Evidence: Before executing the controlled deprecation (which disables or blocks ClickOnce execution and constitutes a change to live system configuration), capture: (1) the full ClickOnce application inventory CSV as the change management baseline; (2) confirmation from each application owner that no production dependency on ClickOnce exists — document this in writing as the authorization artifact for the deprecation change record; (3) current AppLocker or WDAC policy export (see Step 2) as the rollback reference if a missed business-critical ClickOnce dependency surfaces post-deprecation.

Step 6: Monitor developments — CrowdStrike's Part 2 introduces a new undisclosed technique; track the full series publication and incorporate any released detection logic or IOCs into your SIEM and EDR rule sets as they publish

NIST Phase: Post Incident

Reference: NIST 800-61r3 §4 — Post-Incident Activity: Maintaining ongoing threat intelligence integration for an evolving adversary technique series with undisclosed follow-on capabilities

Controls: NIST IR-5 (Incident Monitoring), NIST IR-8 (Incident Response Plan), CIS 7.1 (Establish and Maintain a Vulnerability Management Process), CIS 7.2 (Establish and Maintain a Remediation Process)

Compensating: Set up a no-cost RSS or email alert for CrowdStrike's Adversary Intelligence blog filtered on 'ClickOnce' to receive Part 2 and subsequent publications automatically. Maintain a tracked ticket (Jira, GitHub Issue, or a plaintext change log) that links each published installment to the specific Sigma rule, SIEM query, or IOC list

derived from it, with an assigned analyst and a 5-business-day SLA for rule deployment after publication. Use MISP (free, open-source threat intelligence platform) or a shared STIX/TAXII feed to ingest and distribute any IOCs (file hashes of weaponized .application manifests, C2 URLs embedded in ClickOnce payloads, code-signing certificate thumbprints used in the campaign) once CrowdStrike releases them.

Evidence: This step is a monitoring and intelligence integration activity and does not alter live host state; no volatile capture prerequisite applies. Preserve for continuity: (1) a versioned log of each CrowdStrike ClickOnce series installment reviewed, date reviewed, and resulting detection rule or IOC changes made — this constitutes the audit trail for IR-5 (Incident Monitoring) compliance; (2) any IOCs released in Part 2 or later should be immediately cross-referenced against the historical dfsvc.exe telemetry and %LocalAppData%\Apps\2.0 artifacts collected in Step 1, in case a prior compromise is retroactively identifiable.

Detection Guidance

Detection for ClickOnce abuse requires shifting focus from traditional installer telemetry to user-space process and file activity. Key behavioral patterns to hunt:

1. Process execution: Monitor for dfsvc.exe (the ClickOnce deployment service) spawning child processes or being invoked from browser processes (chrome.exe, msedge.exe, iexplore.exe). Legitimate ClickOnce use is uncommon in most enterprise environments; any instance warrants review.

2. File system activity: Alert on new executable writes to %LocalAppData%\Apps\2.0 followed by immediate execution. This path is the canonical ClickOnce staging directory and should rarely see net-new executable files in managed environments.

3. Network activity: Hunt for dfsvc.exe or its child processes initiating outbound HTTP/HTTPS connections to external hosts, particularly to domains that do not match an internal ClickOnce deployment server. T1105 (Ingress Tool Transfer) activity from this process is a strong signal.

4. Manifest sourcing: Review proxy and DNS logs for .application file downloads or manifest URLs delivered via email links or web redirects. Spearphishing or drive-by delivery would likely precede the ClickOnce execution chain.

5. Masquerading indicators: Compare the signing certificate on any ClickOnce application against your internal CA and known-good vendor certificates. Unsigned or self-signed ClickOnce applications in production environments are high-confidence anomalies.

Relevant NIST controls to audit against: AU-2 (Event Logging) for process creation scope, AU-6 (Audit Record Review, Analysis, and Reporting) for periodic review of user-space execution events, SI-4 for system monitoring coverage gaps in non-admin execution paths. CIS 8.2 (Collect Audit Logs) provides the foundational logging requirement.

Indicators of Compromise

Type	Value	Context	Confidence
TOOL	`dfsvc.exe`	dfsvc.exe (ClickOnce deployment service) leveraged via malicious .application manifest delivered through browser or email link to fetch and execute malicious payload in %LocalAppData%\Apps\2.0 without administrative privileges	HIGH
TOOL	`Pending — refer to CrowdStrike blog series (Part 1 and Part 2) for published indicators`	CrowdStrike's multi-part technical series may include specific payload hashes, C2 infrastructure indicators, or malicious manifest samples; values were not available in the provided source material	LOW

Framework Mappings

MITRE-ATTACK

- **T1027** — Obfuscated Files or Information
- **T1204.002** — Malicious File
- **T1036.005** — Match Legitimate Resource Name or Location
- **T1072** — Software Deployment Tools
- **T1218** — System Binary Proxy Execution
- **T1105** — Ingress Tool Transfer

NIST-800-53R5

- **SI-3** — Malicious Code Protection
- **SI-4** — System Monitoring
- **CA-7** — Continuous Monitoring
- **SC-7** — Boundary Protection
- **AT-2** — Literacy Training and Awareness

CIS-V8

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

NIST-CSF-2

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

MITRE ATT&CK Mapping

Technique ID	Technique Name	Tactic
T1027	Obfuscated Files or Information	Defense-Evasion
T1204.002	Malicious File	Execution
T1036.005	Match Legitimate Resource Name or Location	Defense-Evasion
T1072	Software Deployment Tools	Execution
T1218	System Binary Proxy Execution	Defense-Evasion
T1105	Ingress Tool Transfer	Command-And-Control

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/why-small-businesses-choose-...	T3
	https://www.crowdstrike.com/en-us/blog/reasons-why-nonprofits-are-t...	T3
	https://www.crowdstrike.com/en-us/blog/how-the-infrastructure-inves...	T3
New Abuse of the ClickOnce Technology: Part 1 - CrowdStrike	https://www.crowdstrike.com/content/crowdstrike-www/locale-sites/us...	T3

Generated 2026-06-21 13:21 UTC by TJS Security Command Center