INTELLIGENCE BRIEFING
Security Command Center
TLP:CLEAR
2026-06-20 18:48 UTC

ClickOnce Abused as Malware Delivery Channel: First Documented Analysis Reveals No-Admin-Required Attack Path

SECURITY ANALYSIS | **MEDIUM** | CVSS 5.0

SCC Item ID	SCC-STY-2026-0231
Type	Security Analysis
Severity	MEDIUM
CVSS Base Score	5.0
Affected Products	Microsoft ClickOnce, Windows (.NET environments), Visual Studio publishing pipeline
Discovery Source	Rss:T1 Threatintel

Executive Summary

CrowdStrike researchers have published the first documented analysis of Microsoft ClickOnce deployment technology being weaponized as a malware delivery channel, revealing that attackers can deliver malicious payloads to Windows systems without requiring administrative privileges. The no-admin requirement is significant: it means this vector bypasses one of the most common access controls organizations rely on to contain initial access operations, expanding the viable attack surface to virtually any Windows endpoint running .NET applications. This disclosure signals a broadening of living-off-the-land tradecraft into trusted developer tooling, a pattern that has historically outpaced detection coverage in enterprise environments.

Technical Analysis

CrowdStrike's two-part analysis documents how the ClickOnce deployment model, a Microsoft technology designed to let developers publish self-updating Windows applications via the Visual Studio publishing pipeline, can be turned into a malware staging mechanism. The attack centers on .application manifest files and XML-based deployment descriptors, which ClickOnce processes natively to fetch and execute code. Because the deployment model was designed to operate without elevation, an attacker who can deliver a malformed or malicious manifest to a target gains code execution under the user's existing privilege context. No UAC prompt. No administrative credential requirement.

The underlying weakness classes are CWE-494 (Download of Code Without Integrity Check) and CWE-346 (Origin Validation Error), meaning the deployment runtime does not enforce sufficient validation of where code originates or whether it has been tampered with. This maps directly to MITRE ATT&CK T1218 (System Binary

Proxy Execution), where adversaries leverage trusted system components to execute untrusted code, and T1105 (Ingress Tool Transfer), covering the remote payload staging that follows initial deployment. Delivery likely arrives via spearphishing (T1566, T1566.002) with a malicious .application file as the lure attachment or link, triggering the ClickOnce runtime through T1204.002 (Malicious File) when the user opens it. T1059.005 (Visual Basic Script) suggests scripting components may be involved in subsequent execution stages.

The no-admin execution path is the story's most operationally significant detail. Enterprise defenses that rely on privilege escalation as a detection tripwire, or that assume application installation requires elevation, have a blind spot here. Organizations running .NET-heavy development or business application stacks face broader exposure because ClickOnce is more likely to be present, trusted by endpoint controls, and familiar enough to users that a convincing manifest file could pass casual inspection. CrowdStrike has indicated a Part 2 publication will cover detection strategies; monitor their blog for updates.

Action Checklist

1. Step 1: Assess exposure, determine whether your organization uses ClickOnce-deployed applications in your .NET environment or Visual Studio publishing pipeline; inventory endpoints where the ClickOnce runtime is active (check for presence of dfsvc.exe and the ClickOnce application cache at %LocalAppData%\Apps\2.0\)
2. Step 2: Review controls, verify EDR coverage on dfsvc.exe (the ClickOnce deployment service host) and confirm behavioral detection rules cover unsigned or untrusted .application manifest execution; cross-reference against NIST AC-3 (Access Enforcement) to confirm application execution controls enforce origin validation; review CIS 2.3 (Address Unauthorized Software) to ensure the ClickOnce application cache is in scope for software inventory and unauthorized software detection
3. Step 3: Update threat model, incorporate T1218 (System Binary Proxy Execution via ClickOnce) and T1566.002 (Spearphishing Link delivering .application manifest) into your threat register; tag Windows .NET environments and developer workstations as elevated-risk assets for this vector; note CWE-494 and CWE-346 as the underlying weaknesses to guide detection engineering priorities
4. Step 4: Communicate findings, brief application security and endpoint detection teams on the no-admin-required execution path before broader leadership communication; frame leadership messaging around the bypass of privilege-based detection assumptions rather than a specific patch, since no CVE or vendor fix is assigned at this time
5. Step 5: Monitor developments, track CrowdStrike's Part 2 publication for specific detection rules and IOC releases; monitor MITRE ATT&CK sub-technique updates under T1218 for ClickOnce-specific additions; watch CISA advisories for any threat actor attribution or KEV designation if active exploitation is observed post-disclosure

IR / Forensic Enrichment

Triage Priority	STANDARD
Escalation Criteria	Escalate to urgent if Sysmon Event ID 1 logs show dfsvc.exe spawning unexpected child processes (cmd.exe, powershell.exe, mshta.exe) or initiating outbound network connections to non-organizational infrastructure, or if CISA issues a KEV designation for active ClickOnce exploitation, or if the organization hosts externally-accessible Visual Studio publishing pipelines that could serve as a delivery origin.

Recovery Notes	Because no patch or CVE remediation is available, recovery centers on detection hardening rather than system restoration: verify Sysmon rules and EDR behavioral policies for dfsvc.exe are deployed and generating telemetry across all .NET-capable endpoints before declaring the environment stabilized. For any endpoint where malicious ClickOnce execution is confirmed, purge the ClickOnce application cache at %LocalAppData%\Apps\2.0\ after capturing a forensic copy, revoke any tokens or credentials accessible to the user-context process spawned by dfsvc.exe, and monitor that user account and endpoint for 30 days post-remediation for re-infection or lateral movement indicators. Continue watchlisting dfsvc.exe parent-child process chains in your log pipeline until CrowdStrike's Part 2 IOC release allows rule refinement.
Forensic Artifacts	ClickOnce application cache directory (%LocalAppData%\Apps\2.0\): contains the downloaded .application manifest and deployed payload DLLs/EXEs written by dfsvc.exe; hash all files and inspect for unsigned or anomalously-named assemblies inconsistent with known-good ClickOnce deployments | Sysmon Event ID 1 (Process Create) logs for dfsvc.exe: captures command-line arguments, parent process (typically a browser or email client following a spearphishing link click), and child processes spawned — the no-admin execution path means child processes will run in user context, making this the primary behavioral indicator | Sysmon Event ID 3 (Network Connection) logs for dfsvc.exe: records outbound connections made during .application manifest retrieval and payload download, capturing the C2 or staging infrastructure URL that would not appear in standard Windows Security logs | Windows Security Event Log Event ID 4688 (Process Creation with command line) on endpoints with audit process creation enabled: secondary corroboration of dfsvc.exe execution lineage where Sysmon is not deployed, filtered on dfsvc.exe and its immediate child processes | Browser or email client download history and cache: since delivery relies on a spearphishing link (T1566.002) directing the user to a malicious .application manifest URL, the originating URL will appear in browser history (e.g., %LocalAppData%\Microsoft\Edge\User Data\Default\History or equivalent Chrome/Firefox paths) or Outlook attachment/link activity logs, establishing the initial access vector

Per-Action IR Details

Step 1: Assess exposure — determine whether your organization uses ClickOnce-deployed applications in your .NET environment or Visual Studio publishing pipeline; inventory endpoints where the ClickOnce runtime is active (check for presence of dfsvc.exe and the ClickOnce application cache at %LocalAppData%\Apps\2.0\)

NIST Phase: Preparation

Reference: NIST 800-61r3 §2 — Preparation: establishing visibility into the attack surface before an incident occurs, including asset inventory and identification of exposed components

Controls: CIS 1.1 (Establish and Maintain Detailed Enterprise Asset Inventory), CIS 2.1 (Establish and Maintain a Software Inventory), NIST AC-2 (Account Management)

Compensating: Run the following one-liner across endpoints via PowerShell remoting or a lightweight RMM tool to enumerate dfsvc.exe presence and cache population: `Get-ChildItem -Path "$env:LocalAppData\Apps\2.0\" -Recurse -ErrorAction SilentlyContinue | Select-Object FullName, LastWriteTime` and `Get-Process dfsvc -ErrorAction SilentlyContinue`. Pipe results to CSV for a 2-person team to triage. On Linux-adjacent pipelines, grep Visual Studio publish manifests for deploymentProvider URLs pointing to external infrastructure.

Evidence: This is a pre-incident exposure assessment step and does not alter live state. However, document the baseline state of %LocalAppData%\Apps\2.0\ on representative endpoints before any configuration changes — hash directory contents with `Get-FileHash` so post-incident changes to the cache are detectable. Record dfsvc.exe version, digital signature status, and parent process lineage via `Get-AuthenticodeSignature` for later comparison if malicious ClickOnce execution is later suspected.

Step 2: Review controls — verify EDR coverage on dfsvc.exe (the ClickOnce deployment service host) and confirm behavioral detection rules cover unsigned or untrusted .application manifest execution; cross-reference against NIST AC-3 (Access Enforcement) to confirm application execution controls enforce origin validation; review CIS 2.3 (Address Unauthorized Software) to ensure the ClickOnce application cache is in scope for software inventory and unauthorized software detection

NIST Phase: Preparation

Reference: NIST 800-61r3 §2 — Preparation: validating detection and prevention controls are in place before exploitation occurs, including confirming monitoring coverage of known execution paths

Controls: NIST AC-3 (Access Enforcement), CIS 2.3 (Address Unauthorized Software), NIST AU-2 (Event Logging)

Compensating: Deploy Sysmon with a configuration that explicitly includes dfsvc.exe as a monitored process for Event ID 1 (Process Create), Event ID 3 (Network Connection), and Event ID 11 (File Create). Use SwiftOnSecurity's Sysmon config as a base and add a dedicated rule: `C:\Windows\SysWOW64\dfsvc.exe` and `C:\Windows\System32\dfsvc.exe` across all relevant event types. Audit the ClickOnce cache path with a scheduled Task Scheduler job running `Get-ChildItem %LocalAppData%\Apps\2.0\ -Recurse | Where-Object {$_.LastWriteTime -gt (Get-Date).AddDays(-1)}` to flag new entries daily.

Evidence: Before modifying any detection rules or EDR policies (which alter the monitoring surface), capture the current Sysmon or EDR rule set as a versioned baseline. Record whether dfsvc.exe is currently in any exclusion list — EDR exclusions for legitimate ClickOnce deployments are a common gap that attackers exploit. No volatile host state is altered by this step, but the control-gap documentation itself is evidentiary for post-incident review.

Step 3: Update threat model — incorporate T1218 (System Binary Proxy Execution via ClickOnce) and T1566.002 (Spearphishing Link delivering .application manifest) into your threat register; tag Windows .NET environments and developer workstations as elevated-risk assets for this vector; note CWE-494 and CWE-346 as the underlying weaknesses to guide detection engineering priorities

NIST Phase: Preparation

Reference: NIST 800-61r3 §2 — Preparation: incorporating new threat intelligence into risk posture and detection engineering before active exploitation is observed in the environment

Controls: CIS 7.1 (Establish and Maintain a Vulnerability Management Process), NIST AU-6 (Audit Record Review, Analysis, and Reporting)

Compensating: Create a YARA rule targeting malicious .application manifest files: scan for the `` XML element referencing external or non-organizational URLs, combined with absence of a trusted Authenticode publisher element. Run this YARA rule against the ClickOnce cache directory (`%LocalAppData%\Apps\2.0\`) on developer and endpoint assets using YARA's command-line scanner. Publish the threat model update as a one-page internal advisory to application security and SOC leads with specific dfsvc.exe behavioral indicators drawn from the CrowdStrike research.

Evidence: This is a threat model update step and does not alter live system state. However, capture the current state of your threat register and asset risk tags before updating — the delta between old and new tagging documents the gap that existed prior to this disclosure, which is relevant for GRC and post-incident review. No volatile evidence collection is required for this step.

Step 4: Communicate findings — brief application security and endpoint detection teams on the no-admin-required execution path before broader leadership communication; frame leadership messaging around the bypass of privilege-based detection assumptions rather than a specific patch, since no CVE or vendor fix is assigned at this time

NIST Phase: Preparation

Reference: NIST 800-61r3 §2 — Preparation: establishing communication plans and ensuring internal stakeholders are informed of emerging threat vectors as part of IR readiness

Controls: NIST AC-1 (Policy and Procedures)

Compensating: Use a structured internal advisory template that includes: (1) the no-admin execution path via dfsvc.exe, (2) the specific .application manifest delivery mechanism via spearphishing link, (3) affected asset classes

(developer workstations, .NET environments), (4) current detection gap status, and (5) interim compensating actions from Steps 1-3. Distribute via encrypted email to named stakeholders; do not post to open channels until EDR coverage is confirmed. A 2-person team can draft and deliver this briefing within one business day using the CrowdStrike research as the primary source.

Evidence: No live system state is altered by this communication step. Retain all internal advisories, briefing records, and acknowledgment receipts as documentation artifacts — these establish the timeline of organizational awareness, which is relevant if active exploitation is later discovered and post-incident review must determine whether reasonable preparatory steps were taken in the window between disclosure and exploitation.

Step 5: Monitor developments — track CrowdStrike's Part 2 publication for specific detection rules and IOC releases; monitor MITRE ATT&CK sub-technique updates under T1218 for ClickOnce-specific additions; watch CISA advisories for any threat actor attribution or KEV designation if active exploitation is observed post-disclosure

NIST Phase: Post Incident

Reference: NIST 800-61r3 §4 — Post-Incident Activity: using threat intelligence gathered from incident analysis and external sources to improve detection posture and update IR preparedness for future events

Controls: NIST AU-6 (Audit Record Review, Analysis, and Reporting), CIS 7.1 (Establish and Maintain a Vulnerability Management Process)

Compensating: Set up an RSS or email alert feed for CISA Known Exploited Vulnerabilities (KEV) catalog updates filtered on 'ClickOnce' or 'dfsvc'. Subscribe to CrowdStrike's adversary intelligence blog via RSS. Create a recurring 2-week calendar reminder to check the MITRE ATT&CK changelog (https://attack.mitre.org/resources/changelog/) for updates to T1218 sub-techniques. When IOCs are published (URLs hosting malicious .application manifests, dfsvc.exe child process hashes), immediately import them into osquery scheduled queries targeting process creation and network connection tables.

Evidence: This is an ongoing intelligence monitoring step with no live state modification. Maintain a dated threat intelligence log that records each external update reviewed (CrowdStrike Part 2, ATT&CK changelog, CISA KEV), the IOCs or detection rules extracted, and the internal action taken in response. This log serves as both an operational record and evidence of continuous monitoring posture for compliance and post-incident review purposes.

Detection Guidance

The primary behavioral signal is the ClickOnce deployment service host process, dfsvc.exe, spawning unexpected child processes or initiating outbound network connections to non-organizational infrastructure. Security teams should audit process creation events where dfsvc.exe is the parent and the child is a scripting host (wscript.exe, cscript.exe, mshta.exe) or a command shell. This directly reflects the T1059.005 and T1218 technique mapping.

Log sources to prioritize: Windows Security event logs for process creation (Event ID 4688 with command-line logging enabled), Sysmon Event ID 1 (process creation) and Event ID 3 (network connections) filtered on dfsvc.exe, and application event logs for ClickOnce deployment activity. Per NIST AU-2 (Event Logging) and NIST AU-3 (Content of Audit Records), confirm that process creation logging captures parent-child relationships and full command-line arguments, as these are required to distinguish legitimate ClickOnce deployments from malicious manifest execution.

For threat hunting, develop hypotheses around: (1) .application files delivered via email or downloaded from external URLs outside of known software distribution domains; (2) ClickOnce application cache directories (%LocalAppData%\Apps\2.0\) containing executables that do not correspond to any inventoried or authorized application per CIS 2.1 (Establish and Maintain a Software Inventory); (3) network connections from dfsvc.exe to IP addresses or domains absent from your approved application delivery infrastructure.

D3FEND countermeasures to consider (from the MITRE D3FEND project, https://d3fend.mitre.org/): D3-SFA (System File Analysis) applied to the ClickOnce application cache to detect unauthorized or modified executables; D3-UAP (User Account Permissions) to restrict which user accounts or system contexts are permitted to invoke the ClickOnce runtime where it is not operationally required; D3-FMBV (File Magic Byte Verification) to validate that files staged through ClickOnce manifest delivery match their declared types before execution is permitted by endpoint controls.

Policy gap to audit: verify whether your application allowlisting or EDR policy treats the ClickOnce deployment cache as a trusted execution path. If it does, that trust may need to be scoped or conditioned on code-signing validation to close the CWE-494 gap.

Indicators of Compromise

Type	Value	Context	Confidence
TOOL	`dfsvc.exe`	ClickOnce deployment service host (dfsvc.exe) leveraged via malicious .application manifest delivery to execute untrusted code under user-level privileges without administrative elevation	HIGH
URL	`Pending — refer to CrowdStrike blog Part 2 (https://www.crowdstrike.com/en-us/blog/) for published indicators`	CrowdStrike's Part 2 analysis covers detection strategies and is expected to include specific indicators; Part 1 source material does not publish discrete hash, domain, or IP values	LOW

Framework Mappings

MITRE-ATTACK

- **T1218** — System Binary Proxy Execution
- **T1059.005** — Visual Basic
- **T1566** — Phishing
- **T1105** — Ingress Tool Transfer
- **T1204.002** — Malicious File
- **T1566.002** — Spearphishing Link

NIST-800-53R5

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

NIST-CSF-2

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

MITRE ATT&CK Mapping

Technique ID	Technique Name	Tactic
T1218	System Binary Proxy Execution	Defense-Evasion
T1059.005	Visual Basic	Execution
T1566	Phishing	Initial-Access
T1105	Ingress Tool Transfer	Command-And-Control
T1204.002	Malicious File	Execution
T1566.002	Spearphishing Link	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/why-small-businesses-choose-...	T3
	https://www.crowdstrike.com/en-us/blog/how-the-infrastructure-inves...	T3
	https://www.crowdstrike.com/en-us/blog/reasons-why-nonprofits-are-t...	T3
ClickOnce Deployment and Security - Visual Studio (Windows)	https://learn.microsoft.com/en-us/visualstudio/deployment/clickonce...	T1

Generated 2026-06-20 18:48 UTC by TJS Security Command Center