# **CCleaner Supply Chain Attack (2017)**

### **1. Core Issue**

The CCleaner attack was a **compromise of a trusted software distribution platform**. Attackers managed to slip a malicious version of CCleaner, a popular Windows optimization tool, into the official distribution channel. The Trojanized installer contained a backdoor that enabled remote command execution. The core issue lay in the **infiltration of the vendor’s build environment** and the failure to detect unauthorized modifications before release.

### **2. Who Was Attacked**

The direct victim was **Piriform**, the company behind CCleaner (later acquired by Avast). Its development and release pipeline was compromised, leading to malicious versions being pushed through legitimate update mechanisms.

### **3. Who Was Affected**

* Over **2.3 million users** worldwide downloaded and installed the compromised version.
* While the majority of victims were everyday users, telemetry showed that the attackers specifically targeted **large corporations and tech firms**, including Intel, Microsoft, Samsung, and Cisco, attempting to deliver secondary payloads.

### **4. Exploit Chain Details**

1. **Initial Compromise** – Attackers gained access to Piriform’s build environment.
2. **Malicious Code Injection** – The CCleaner installer was modified to include a backdoor (Floxif malware).
3. **Distribution to Users** – The Trojanized version (v5.33) was digitally signed and hosted on official CCleaner servers.
4. **Installation by Victims** – Users trusted the signed installer and unknowingly deployed malware.
5. **Command & Control (C2)** – The malware contacted remote servers, enabling attackers to issue commands.
6. **Targeted Delivery** – In select high-value environments, additional malware (Stage 2 payload) was installed for espionage.

### **5. Prevention / Protection Steps**

* **Code Signing with Verification**: Validate code integrity not only at build time but post-build as well.
* **Build Environment Security**: Isolate critical signing infrastructure and enforce multi-factor authentication.
* **Supply Chain Auditing**: Conduct regular third-party audits of build and release pipelines.
* **Network Monitoring**: Deploy IDS/IPS to detect unusual outbound connections to known malicious IPs.

### **6. Fixes & Vendor Response**

* Piriform quickly released a **clean version (v5.34)** to remove the backdoor.
* Avast conducted a forensic investigation and coordinated with law enforcement.
* C2 infrastructure associated with the attack was taken down.
* Security vendors issued detection signatures for the Floxif malware.

### 

### **7. If No Fix Available**

* Users should uninstall the compromised version immediately.
* Rotate credentials and scan for persistence mechanisms.
* Rebuild systems if secondary payloads are suspected.
* Employ endpoint detection tools to identify any residual activity.

### **8. Reference Material**

* Talos Intelligence – CCleaner Supply Chain Attack Analysis:  
   https://blog.talosintelligence.com/ccleaner-backdoor/
* Avast Security Advisory – CCleaner Compromise Details:  
   https://blog.avast.com/ccleaner-incident-analysis
* Piriform / Avast Official Statement:  
   https://www.ccleaner.com/ccleaner-incident
* CISA Alert – CCleaner Compromise:  
   https://www.cisa.gov/news-events/analysis-reports/ar17-260a
* US-CERT Technical Alert (TA17-260A):  
   https://www.us-cert.gov/ncas/analysis-reports/AR17-260A
* Securelist – CCleaner Backdoor Follow-Up Investigation:  
   https://securelist.com/ccleaner-backdoor-follow-up/82637/

### **9. Further Reading**

* ENISA Threat Landscape for Supply Chain Attacks (2021):  
   https://www.enisa.europa.eu/publications/threat-landscape-for-supply-chain-attacks
* MITRE ATT&CK – Supply Chain Compromise (T1195):  
   https://attack.mitre.org/techniques/T1195/
* OWASP Software Supply Chain Security Guide:  
   https://owasp.org/www-project-software-supply-chain-security/
* OpenSSF Best Practices for Securing Software Supply Chains:  
   https://openssf.org/working-groups/supply-chain-integrity/
* Avast’s detailed post-incident report:  
   https://blog.avast.com/ccleaner-incident-analysis

### **10. Tooling**

* Sigstore / Cosign – Open-source artifact signing and verification:  
   https://sigstore.dev/
* in-toto – Build provenance and supply chain integrity framework:  
   https://in-toto.io/
* YARA – Pattern matching tool for malware detection:  
   https://virustotal.github.io/yara/
* Zeek – Network analysis platform for detecting suspicious traffic:  
   https://zeek.org/
* Microsoft Defender for Endpoint – Threat detection and IOC monitoring:  
  <https://www.microsoft.com/en-us/security/business/threat-protection/microsoft-defender-endpoint>
* VirusTotal – Malware scanning and signature matching:  
   https://www.virustotal.com/