

3XC-SUPPLY-CHAIN_allunell

Q1: Understanding the scope of the attack and identifying which versions exhibit malicious behavior is crucial for making informed decisions if these compromised versions are present in the organization. How many versions of 3CX running on Windows have been flagged as malware?

2

Source: https://www.splunk.com/en_us/blog/security/splunk-insights-investigating-the-3cxdesktopapp-supply-chain-compromise.html

Affected 3CX versions:

- 3CXDesktopApp-18.12.407.msi
- 3CXDesktopApp-18.12.416.msi

Q2: Determining the age of the malware can help assess the extent of the compromise and track the evolution of malware families and variants. What's the UTC creation time of the .msi malware?

2023-03-13 06:33

```
allun@kali: ~/Escritorio/Laboratorios/BlueTeam/temp_extract_dir
$ file 3CXDesktopApp-18.12.416.msi
3CXDesktopApp-18.12.416.msi: Composite Document File V2 Document, Little Endian, OS: Windows, Version 6.2, MSI Installer, Code page: 1252, Title: Installation Database, Subject: 3CX Desktop App, Author: 3CX Ltd., Keywords: Installer, Comments: Windows Installer Package, Template: x64!033, Revision Number: 998084FA-1803-4D8A-A4A6-65D94F4D288A, Create Time/Date: Mon Mar 13 06:33:26 2023, Last Saved Time/Date: Mon Mar 13 06:33:26 2023, Number of Pages: 465, Number of Words: 2, Name of Creating Application: Windows Installer XML Toolset (3.11.2.4516), Security: 2
```

Q3: Executable files (.exe) are frequently used as primary or secondary malware payloads, while dynamic link libraries (.dll) often load malicious code or enhance malware functionality. Analyzing files deposited by the Microsoft Software Installer (.msi) is crucial for identifying malicious files and investigating their full potential. Which malicious DLLs were dropped by the .msi file?

```
ffmpeg.dll,d3dcompiler_47.dll
```

Source: <https://www.virustotal.com/gui/file/59e1edf4d82fae4978e97512b0331b7eb21dd4b838b850ba46794d9c7a2c0983/relations>


Bundled Files (86)			
Scanned	Detections	File type	Name
2025-03-22	58 / 73	Win32 DLL	ffmpeg.dll
2025-03-24	49 / 73	Win32 DLL	d3dcompiler_47.dll
2025-03-19	40 / 62	CAB	product.cab
2025-02-25	28 / 72	Win32 EXE	_3CXDesktopApp.exe_7fcf4c12_fd40_46f7_8b99_2e20486670af
2025-03-11	24 / 71	Win32 EXE	3CXDesktopApp.exe

Q4: Recognizing the persistence techniques used in this incident is essential for current mitigation strategies and future defense improvements. What is the MITRE Technique ID employed by the .msi files to load the malicious DLL?

```
T1574
```

Source: <https://attack.mitre.org/techniques/T1574/001/>

Hijack Execution Flow: DLL Search Order Hijacking

Other sub-techniques of Hijack Execution Flow (13) 

Adversaries may execute their own malicious payloads by hijacking the search order used to load DLLs. Windows systems use a common method to look for required DLLs to load into a program. ^{[1][2]} Hijacking DLL loads may be for the purpose of establishing persistence as well as elevating privileges and/or evading restrictions on file execution.

There are many ways an adversary can hijack DLL loads. Adversaries may plant trojan dynamic-link library files (DLLs) in a directory that will be searched before the location of a legitimate library that will be requested by a program, causing Windows to load their malicious library when it is called for by the victim program. Adversaries may also perform DLL preloading, also called binary planting attacks, ^[3] by placing a malicious DLL with the same name as an ambiguously specified DLL in a location that Windows searches before the legitimate DLL. Often this location is the current working directory of the

ID: T1574.001

Ask AI

Sub-technique of: T1574

① Tactics: Persistence, Privilege Escalation, Defense Evasion

① Platforms: Windows

Contributors: Ami Hoston, CrowdStrike; Marina Liang; Stefan Kanthak; Travis Smith, Tripwire; Will Alexander, CrowdStrike

Version: 1.3

Created: 13 March 2020


Last Modified: 30 September 2024

Q5: Recognizing the malware type (threat category) is essential to your investigation, as it can offer valuable insight into the possible malicious actions you'll be examining. What is the threat category of the two malicious DLLs?

trojan

Source:
<https://www.virustotal.com/gui/file/59e1edf4d82fae4978e97512b0331b7eb21dd4b838b850ba46794d9c7a2c0983/relations>

Popular threat label

 trojan.samscissors/dllhijack

Threat categories

trojan

pua

Family labels

samscissors

dllh

Q6: As a threat intelligence analyst conducting dynamic analysis, it's vital to understand how malware can evade

detection in virtualized environments or analysis systems. This knowledge will help you effectively mitigate or address these evasive tactics. What is the MITRE ID for the virtualization/sandbox evasion techniques used by the two malicious DLLs?

T1497

Source: <https://attack.mitre.org/techniques/T1497/001/>

Virtualization/Sandbox Evasion: System Checks

Other sub-techniques of Virtualization/Sandbox Evasion (3)

Adversaries may employ various system checks to detect and avoid virtualization and analysis environments. This may include changing behaviors based on the results of checks for the presence of artifacts indicative of a virtual machine environment (VME) or sandbox. If the adversary detects a VME, they may alter their malware to disengage from the victim or conceal the core functions of the implant. They may also search for VME artifacts before dropping secondary or additional payloads. Adversaries may use the information learned from [Virtualization/Sandbox Evasion](#) during automated discovery to shape follow-on behaviors.^[1]

Specific checks will vary based on the target and/or adversary, but may involve behaviors such as [Windows Management Instrumentation](#), [PowerShell](#), [System Information Discovery](#), and [Query Registry](#) to obtain system information and search for VME artifacts. Adversaries may search for VME artifacts in memory,

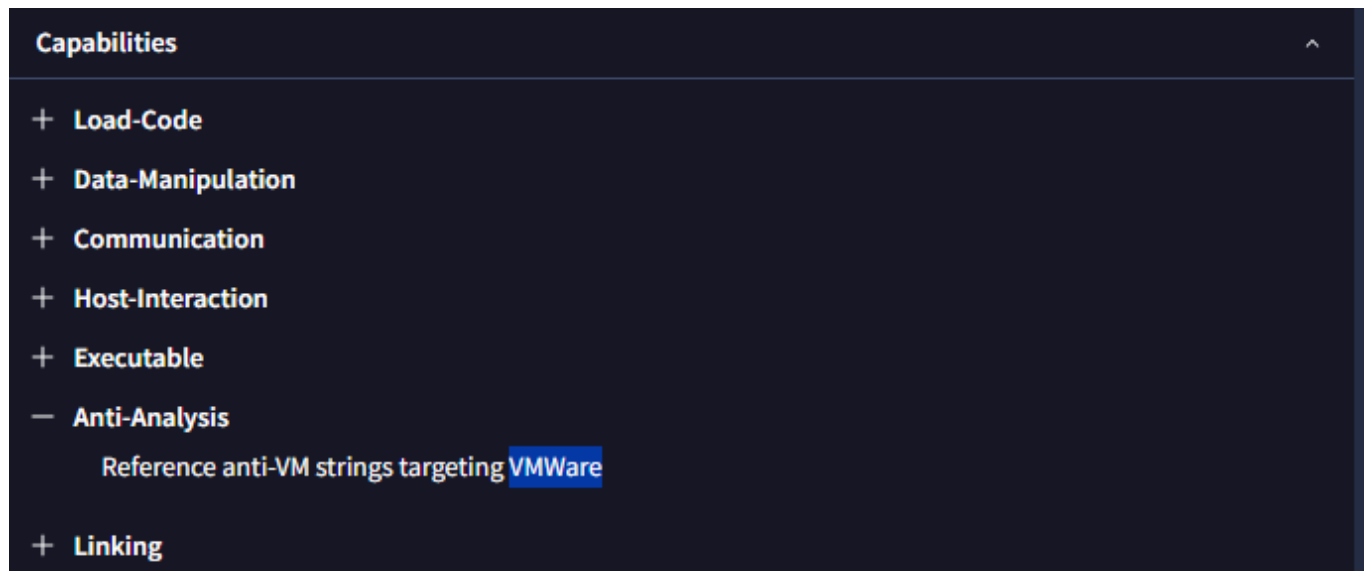
ID: **T1497**.001
Sub-technique of: T1497
① **Tactics:** [Defense Evasion](#), [Discovery](#)
① **Platforms:** Linux, Windows, macOS
① **Defense Bypassed:** Anti-virus, Host forensic analysis, Signature-based detection, Static File Analysis
Contributors: Deloitte Threat Library Team; Kostya Vasilkov
Version: 2.2
Created: 06 March 2020
Last Modified: 12 September 2024

Q7: When conducting malware analysis and reverse engineering, understanding anti-analysis techniques is vital to avoid wasting time. Which hypervisor is targeted by the anti-analysis techniques in the `ffmpeg.dll` file?

VMWare

Source:

<https://www.virustotal.com/gui/file/7986bbaee8940da11ce089383521ab420c443ab7b15ed42aed91fd31ce833896/behavior>



Q8: Identifying the cryptographic method used in malware is crucial for understanding the techniques employed to bypass defense mechanisms and execute its functions fully. What encryption algorithm is used by the `ffmpeg.dll` file?

RC4

Source:

<https://www.virustotal.com/gui/file/7986bbaee8940da11ce089383521ab420c443ab7b15ed42aed91fd31ce833896/behavior>



Q9: As an analyst, you've recognized some TTPs involved in the incident, but identifying the APT group responsible will help you search for their usual TTPs and uncover other potential malicious activities. Which group is responsible for this attack?

Lazarus

Source:

<https://www.virustotal.com/gui/file/7986bbaee8940da11ce089383521ab420c443ab7b15ed42aed91fd31ce833896/community>



patricksvgrapi

📅 2 years ago

This indicator was mentioned in a report.

- 🔍 Title: 3CXDesktopApp Backdoored in a Suspected **Lazarus** Campaign
- 📄 Reference: <https://blog.qualys.com/vulnerabilities-threat-research/2023/04/03/3cxdesktopapp-backdoored-in-a-suspected-lazarus-campaign>
- 📅 Report Publish Date: 2023-04-04
- 📁 Sample Upload Date: 2023-03-22
- 🔑 Reference ID: #6ee1e00ef (<https://www.virustotal.com/gui/search/6ee1e00ef/comments> for report's related indicators)