2023 LONDON

4 - 6 October, 2023 / London, United Kingdom

SIDE LOADING IS NOT DEAD: THE CHINESE AND THE KOREAN WAY

Gabor Szappanos

Sophos, Hungary

gabor.szappanos@sophos.com

ABSTRACT

DLL side loading is an old technique, it used to be the favourite of Chinese APT groups and the most prolific payload was the infamous PlugX backdoor. Both are well covered by a lot of publications all over the industry. But just because it is an old technique, doesn't mean it is also abandoned.

One thing you learn if you stick around for long enough in this industry is that perpetrators don't just give up on techniques that work. And even if they do, the same tools and techniques tend to be resurrected a few years later. DLL side loading and the PlugX payload are very much alive and active these days.

This paper will cover incidents that used DLL side loading in 2022 and 2023.

PLUGX – THE CLASSIC METHOD

We observed localized outbreaks of a new variant of PlugX – a popular payload used by multiple Chinese APT groups. After first drawing attention to itself in Papua New Guinea in August 2022, additional infections appeared in Ghana, Mongolia, Zimbabwe and Nigeria.

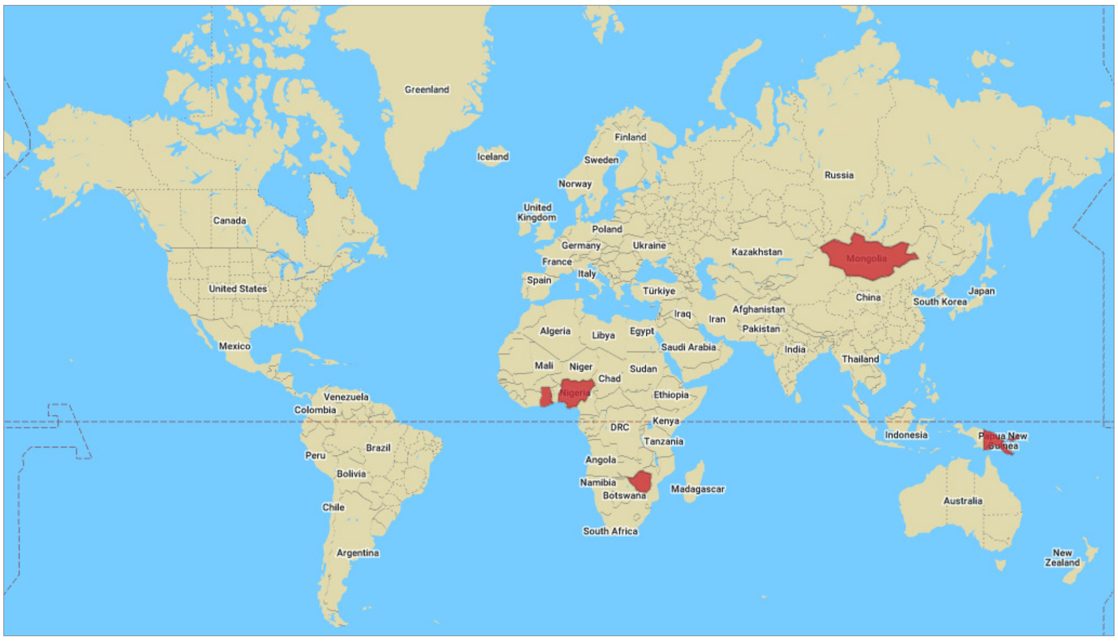


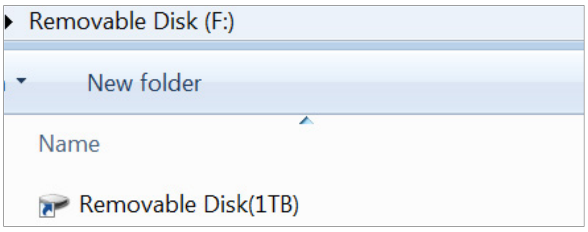
Figure 1: Distribution of infections of the new PlugX variant.

The new variant featured a couple of interesting developments not seen in the classic infection campaigns: it was able to spread via USB drives as a worm, and it automatically exfiltrated document files from the infected system.

PlugX used the following components in the side loading:

- Clean loader: *AvastSvc.exe, CEFHelper.exe (renamed)*
- Malicious loader: *wcs.dll*
- Encrypted payload: *avastauth.dat*

This particular version also showcased a couple of tricks to hide its malicious content from the casual observer. First, the infected removable media appears to be empty in *Windows Explorer* – as if the drive only contains another removable drive.



Removable Disk (F:)

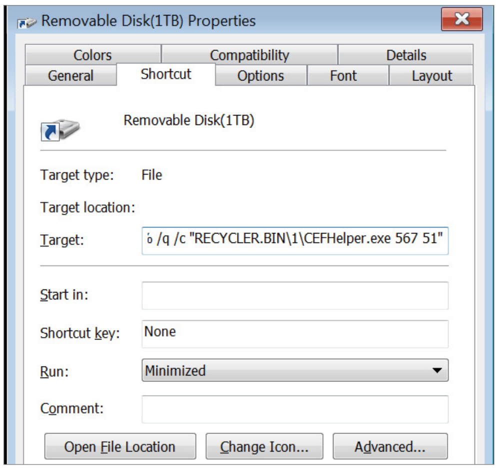
New folder

Name

Removable Disk(1TB)

Figure 2: Seemingly empty thumb drive.

In reality, the displayed item is not actually a drive but a *Windows* shortcut file using an icon resembling the one used for removable media. Should the victim click on this file, it executes the CEFHelper command:



Removable Disk(1TB)

Target type: File

Target location:

Target: ò /q /c "RECYCLER.BIN\1\CEFHelper.exe 567 51"

Start in:

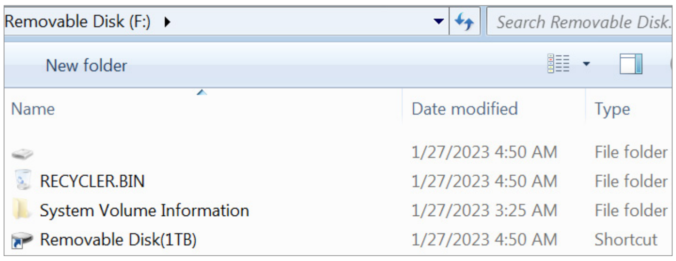
Shortcut key: None

Run: Minimized

Comment:

Figure 3: The shortcut command.

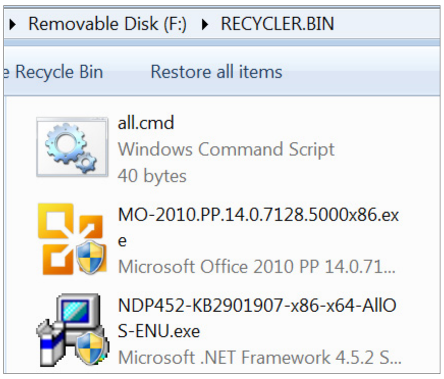
The other files and directories have the hidden and system attributes set so that they will not be visible by default in the file listing. After specifically enabling the display of hidden and system files, we can see the rest of the content:



Name	Date modified	Type
	1/27/2023 4:50 AM	File folder
RECYCLER.BIN	1/27/2023 4:50 AM	File folder
System Volume Information	1/27/2023 3:25 AM	File folder
Removable Disk(1TB)	1/27/2023 4:50 AM	Shortcut

Figure 4: The real content of the thumb drive.

The files copied by the backdoor are in the RECYCLER.BIN directory – for which the worm, in another obfuscation manoeuvre, drops a desktop.ini file that associates the directory with the actual Recycler function of the operating system. This causes *Windows* to treat the directory as if it really is a recycle bin, and files deleted by the user will be displayed there instead of the files physically in the directory.



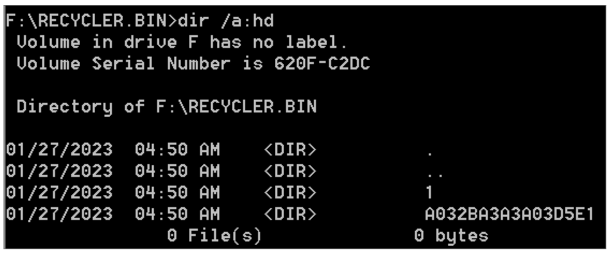
all.cmd
Windows Command Script
40 bytes

MO-2010.PP.14.0.7128.5000x86.ex
e
Microsoft Office 2010 PP 14.0.71...

NDP452-KB2901907-x86-x64-AllO
S-ENU.exe
Microsoft .NET Framework 4.5.2 S...

Figure 5: The files that are not there.

Sticking to the *Windows* command prompt reveals the real content of the RECYCLER.BIN directory:



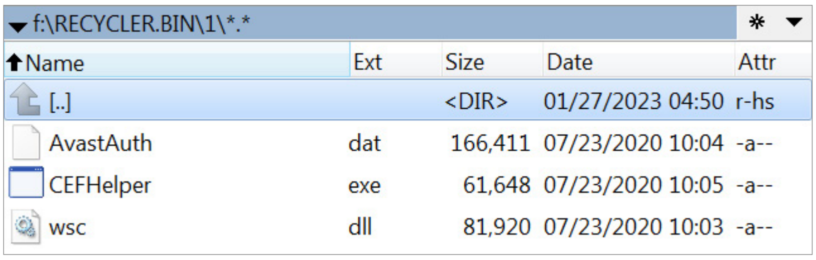
```
F:\RECYCLER.BIN>dir /a:hd
 Volume in drive F has no label.
 Volume Serial Number is 620F-C2DC

 Directory of F:\RECYCLER.BIN

01/27/2023  04:50 AM    <DIR>          .
01/27/2023  04:50 AM    <DIR>          ..
01/27/2023  04:50 AM    <DIR>          1
01/27/2023  04:50 AM    <DIR>          A032BA3A3A03D5E1
               0 File(s)              0 bytes
```

Figure 6: The real content of RECYCLER.BIN.

The directory named ‘1’ contains the DLL side-loading components:

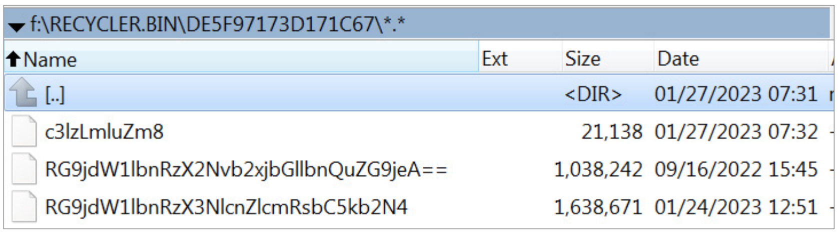


f:\RECYCLER.BIN\1*.*

Name	Ext	Size	Date	Attr
[..]		<DIR>	01/27/2023 04:50	r-hs
AvastAuth	dat	166,411	07/23/2020 10:04	-a--
CEFHelper	exe	61,648	07/23/2020 10:05	-a--
wsc	dll	81,920	07/23/2020 10:03	-a--

Figure 7: The side-loading components.

The other directory, which has a random name, contains the victim’s exfiltrated files.



f:\RECYCLER.BIN\DE5F97173D171C67*.*

Name	Ext	Size	Date
[..]		<DIR>	01/27/2023 07:31
c3lzLmluZm8		21,138	01/27/2023 07:32
RG9jdW1lbnRzX2Nvb2xjbGllbnQuZG9jeA==		1,038,242	09/16/2022 15:45
RG9jdW1lbnRzX3NlcnZlcmRsbC5kb2N4		1,638,671	01/24/2023 12:51

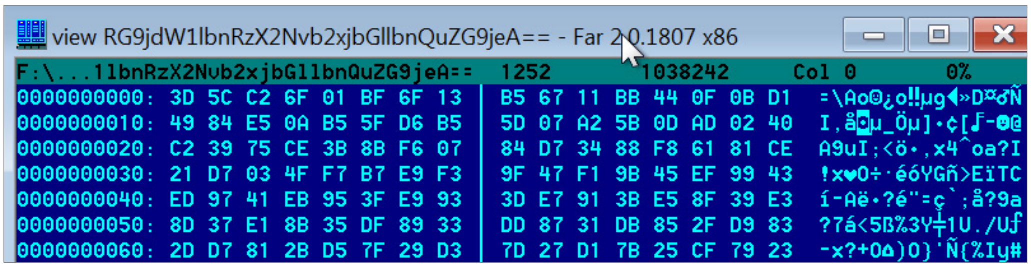
Figure 8: The exfiltrated documents.

This is another recent feature of PlugX: it collects .doc, .docx, .xls, .xlsx, .ppt, .pptx and .pdf files, likely for exfiltration. It saves them in encrypted form to RECYCLER.BIN, as shown above. The file names, including the path indicator, are converted to Base64 form. For example, in Figure 8, the two file names decode to:

```
Documents_coolclient.docx

Documents_serverdll.docx
```

The content of each file is also encrypted:



view RG9jdW1lbnRzX2Nvb2xjbGllbnQuZG9jeA== - Far 2.0.1807 x86

```
F:\...1lbnRzX2Nvb2xjbGllbnQuZG9jeA==  1252      1038242     Col 0        0%
0000000000: 3D 5C C2 6F 01 BF 6F 13 | B5 67 11 BB 44 0F 0B D1  =\Ao☺¿o‼µg◄»D☼♂Ñ
0000000010: 49 84 E5 0A B5 5F D6 B5 | 5D 07 A2 5B 0D AD 02 40  I,å◙µ_Öµ]•¢[♪-☻@
0000000020: C2 39 75 CE 3B 8B F6 07 | 84 D7 34 88 F8 61 81 CE  A9uI;<ö•,x4^oa?I
0000000030: 21 D7 03 4F F7 B7 E9 F3 | 9F 47 F1 9B 45 EF 99 43  !x♥O÷·éóYGñ>EïTC
0000000040: ED 97 41 EB 95 3F E9 93 | 3D E7 91 3B E5 8F 39 E3  í-Aë•?é"=ç`;å?9a
0000000050: 8D 37 E1 8B 35 DF 89 33 | DD 87 31 DB 85 2F D9 83  ?7á<5ß%3Y╪1U./Uƒ
0000000060: 2D D7 81 2B D5 7F 29 D3 | 7D 27 D1 7B 25 CF 79 23  -x?+O⌂)O}'Ñ{%Iy#
```

Figure 9: Document content is encrypted.

The C2 and exfiltration server is already down, but the USB worm functionality acts as a loose cannon and can initiate local outbreaks – just as we have seen in a few countries. The infected thumb drives can carry the exfiltrated documents to unwanted locations. And although the content is encrypted, sometimes even the file names can contain sensitive information.

DRAGON BREATH – TWO-STAGED LOADING

We have observed infections that were follow-ups to Operation Dragon Breath, a.k.a. APT-Q-27 (described in Chinese in [1], [2]), with some modifications. The original campaigns targeted people engaged in online gambling, and initial infection vectors were distributed via trojanized installers for popular applications like *Telegram*. The attackers behind these campaigns are believed to be Chinese groups.

We have seen victims in the Philippines, Japan, Taiwan, Singapore, Hong Kong and China.

Unlike the PlugX case, where only a skeleton side-loading set is installed on the infected system, here a fully functional application package is deployed – along with the malicious components. The original attacks used a classic side-loading scenario, with a clean application, a malicious loader and an encrypted payload. The new campaigns added a twist, using a two-staged approach. The first-stage clean application loads another clean application and a second-stage loader, and the latter side loads the malicious DLL, which executes the final payload.

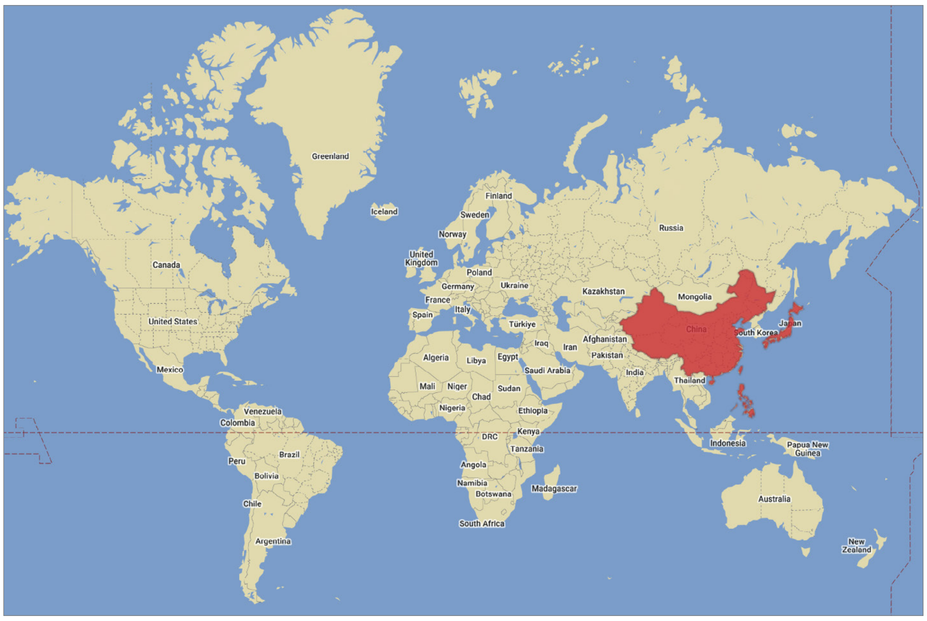
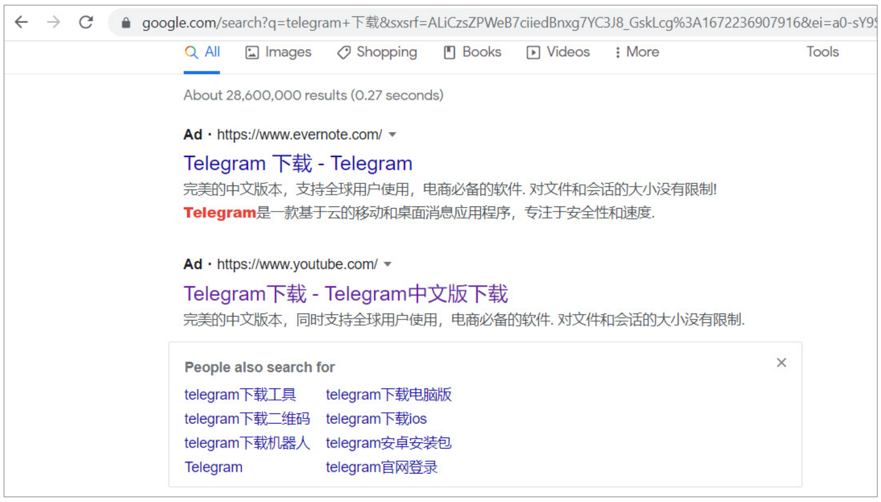


Figure 10: Distribution of infections of the new Dragon Breath variant.

The malicious installer is distributed using search engine poisoning. For example, searching for 'Telegram 下载' (which translates to '*Telegram* download') gave the following results:


google.com/search?q=telegram+下载&sxsrf=ALiCzsZPWeB7ciiedBnxg7YC3J8_GskLcg%3A1672236907916&ei=a0-sY9

About 28,600,000 results (0.27 seconds)

Ad · https://www.evernote.com/

Telegram 下载 - Telegram

完美的中文版本，支持全球用户使用，电商必备的软件. 对文件和会话的大小没有限制!
Telegram是一款基于云的移动和桌面消息应用程序，专注于安全性和速度.

Ad · https://www.youtube.com/

Telegram下载 - Telegram中文版下载

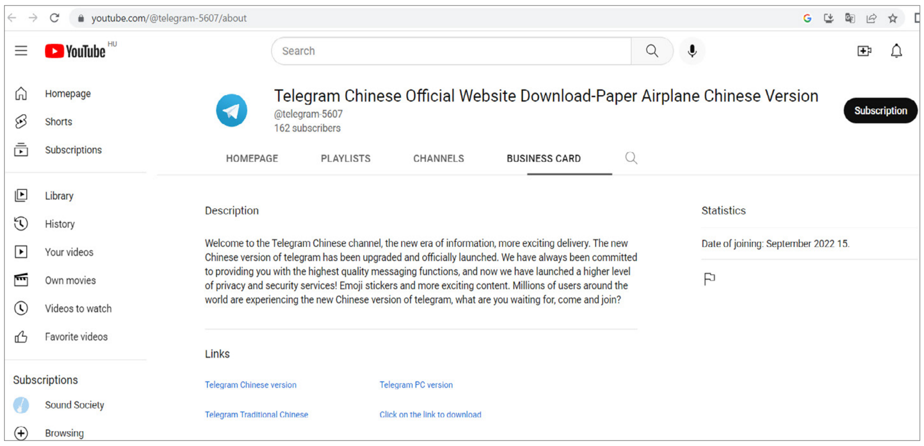
完美的中文版本，同时支持全球用户使用，电商必备的软件. 对文件和会话的大小没有限制.

People also search for

telegram下载工具 telegram下载电脑版
telegram下载二维码 telegram下载ios
telegram下载机器人 telegram安卓安装包
Telegram telegram官网登录

Figure 11: Web search leads to tainted content.

The second link led to a *YouTube* page offering download for the requested application installer:


youtube.com/@telegram-5607/about

Telegram Chinese Official Website Download-Paper Airplane Chinese Version
@telegram 5607
162 subscribers

Description

Welcome to the Telegram Chinese channel, the new era of information, more exciting delivery. The new Chinese version of telegram has been upgraded and officially launched. We have always been committed to providing you with the highest quality messaging functions, and now we have launched a higher level of privacy and security services! Emoji stickers and more exciting content. Millions of users around the world are experiencing the new Chinese version of telegram, what are you waiting for, come and join?

Statistics

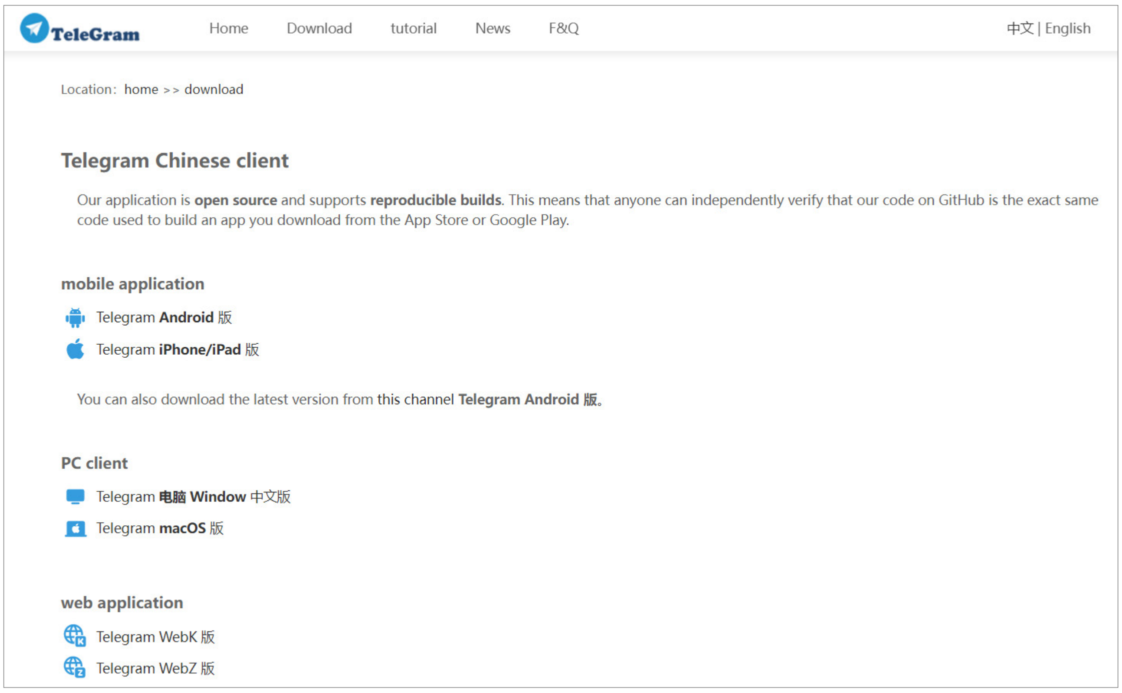
Date of joining: September 2022 15.

Links

Telegram Chinese version
Telegram PC version
Telegram Traditional Chinese
Click on the link to download

Figure 12: Malicious YouTube accounts promote fake sites.

The site to which the *YouTube* links redirected holds *Telegram* installer versions for all popular platforms:



Location: home >> download

Telegram Chinese client

Our application is **open source** and supports **reproducible builds**. This means that anyone can independently verify that our code on GitHub is the exact same code used to build an app you download from the App Store or Google Play.

mobile application

Telegram **Android** 版

Telegram **iPhone/iPad** 版

You can also download the latest version from this channel **Telegram Android 版**。

PC client

Telegram **电脑 Window** 中文版

Telegram **macOS** 版

web application

Telegram WebK 版

Telegram WebZ 版

Figure 13: Content for all platforms presented.

Of those, the *Android* and *Apple* links point to the legitimate locations of the installers. The *Windows* version, however, leads to content provided by the attackers.

Once the installer is executed, at the front the legitimate installer is displayed, which will install a full version of the *Telegram* application as well as a shortcut on the desktop. At this point, a fully working installation of *Telegram* is created on the victim's system.

However, some additional, unsolicited components are dropped during installation. First, the shortcut does not execute the *Telegram* program directly – instead, it invokes an unusual command line:

```
C:\Users\{redacted}\AppData\Roaming\Tg_B518c1A0Ff8C\appR.exe /s  /n  /u  /i:appR.dat
appR.dll
```

This command line refers to the following files:

- appR.exe: renamed regsvr32.exe
- appR.dll: renamed scrobj.dll
- appR.dat: installer script

The renamed script engine DLL will execute the JavaScript code stored in appR.dat. On the surface it displays the *Telegram Desktop* UI, as expected:

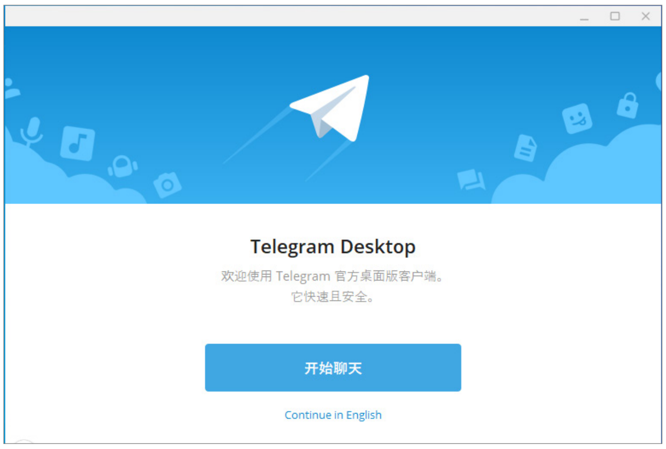
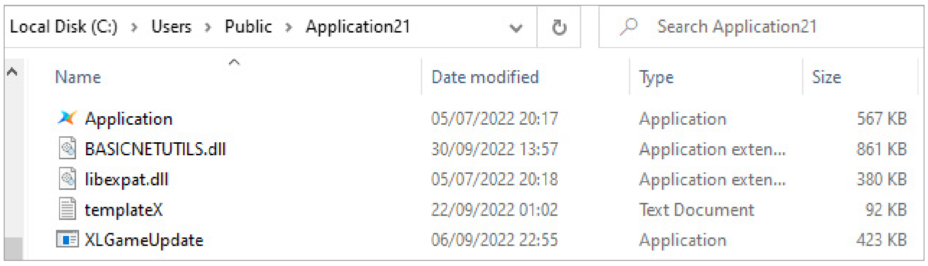


Figure 14: Telegram runs to cover malicious activities.

But behind the scenes it drops the side-loading components to another directory:



Name	Date modified	Type	Size
Application	05/07/2022 20:17	Application	567 KB
BASICNETUTILS.dll	30/09/2022 13:57	Application exten...	861 KB
libexpat.dll	05/07/2022 20:18	Application exten...	380 KB
templateX	22/09/2022 01:02	Text Document	92 KB
XLGameUpdate	06/09/2022 22:55	Application	423 KB

Figure 15: Side-loading components created.

The side-loading components and the startup link are only created when the desktop *Telegram* link is executed, not on the completion of the initial installer. This could be an anti-analysis trick – dynamic analysis sandboxes will not see the dropped side-loader files.

The side loading is not the usual, there are two stages: the first stage is executing a clean application, whilst the second stage does the usual side loading, as illustrated in Figure 16:

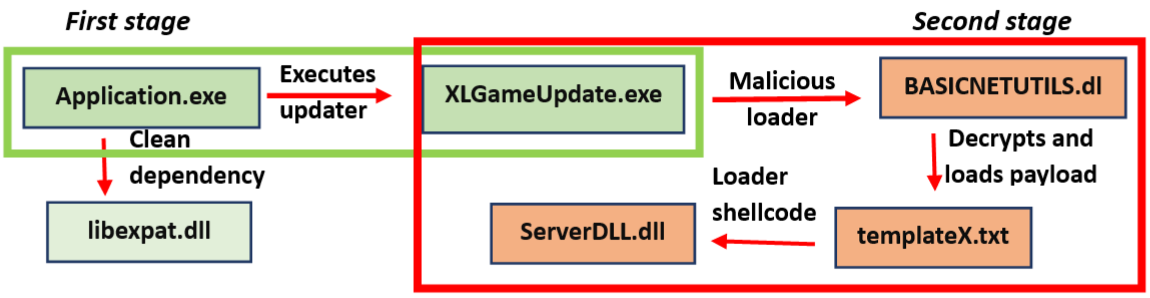


Figure 16: The side loading scheme.

The first-stage components are:

- Clean loader: XLGame.exe (迅雷游戏 by Shenzhen Thunder)
- Clean dependency: libexpat.dll
- Autoupdate EXE: XLGameUpdate.exe

The first-stage loader is the program XLGame.exe, signed by SHENZHEN THUNDER NETWORKING TECHNOLOGIES LTD and renamed to Application.exe. This has a clean dependency, libexpat.dll, which is part of the package.

XLGame will automatically perform an update if it finds a program named XLGameUpdate.exe in the same directory. The installation process makes use of it, as the malicious package contains an executable with this name.

However, this is not the original XLGameUpdate.exe, rather it is replaced by a clean EXE signed by Beijing Baidu Netcom Science and Technology Co., Ltd.

This second-stage loader is used for the usual side-loading scenario, its dependency, BASICNETUTILS.dll, is replaced with a malicious loader DLL.

The malicious loader will find templateX.txt in the same directory, load the content, decrypt the payload loader shellcode and execute it. Later versions used the file name ‘template.txt’.

The payload DLL contains one export:

```
dllname: ServerDll.dll
Ordinal: 1
file offset: 0x14780
RVA: 0x15380
export name: Fuck
```

Apart from the usual backdoor functionality, the payload also steals content from the *MetaMask Ethereum* wallet:

```
C:\Users\%s\AppData\Local\Google\Chrome\User Data\Default\Extensions\
nkbihfbeogaeaoehlefnkodbefgpgknn\
```

This application was reported in the earlier campaigns to steal the content of the wallet.

Operation Dragon Breath was not the only campaign that used search engine poisoning combined with trojanized installers of popular applications and featuring DLL side loading to execute the final payload. We have observed multiple other campaigns, seemingly by different threat actors. It seems that this scenario is gaining popularity among threat actors.

NITROGEN – EXPORT FORWARDING

Sophos X-Ops has observed the Nitrogen campaign targeting several organizations in the technology and non-profit sectors across multiple geographic regions. We think that the threat actors mean to stage compromised environments for ransomware deployment. This assessment is corroborated by recent research by *Trend Micro* [3], in which researchers observed a similar infection chain that led to a BlackCat (aka ALPHV) [4] ransomware infection.

While investigating this campaign, we uncovered a new initial access malware family called Nitrogen. The name derives from the components and debug information we found in the samples, which indicate that the developers refer to this project as Nitrogen or Nitronet.

The main components use the following class names:

```
NitrogenStager
MsfPythonStager
NitronetNativeStager
NitroInstaller
```

The following diagram illustrates our current understanding of the stages of the infection process.

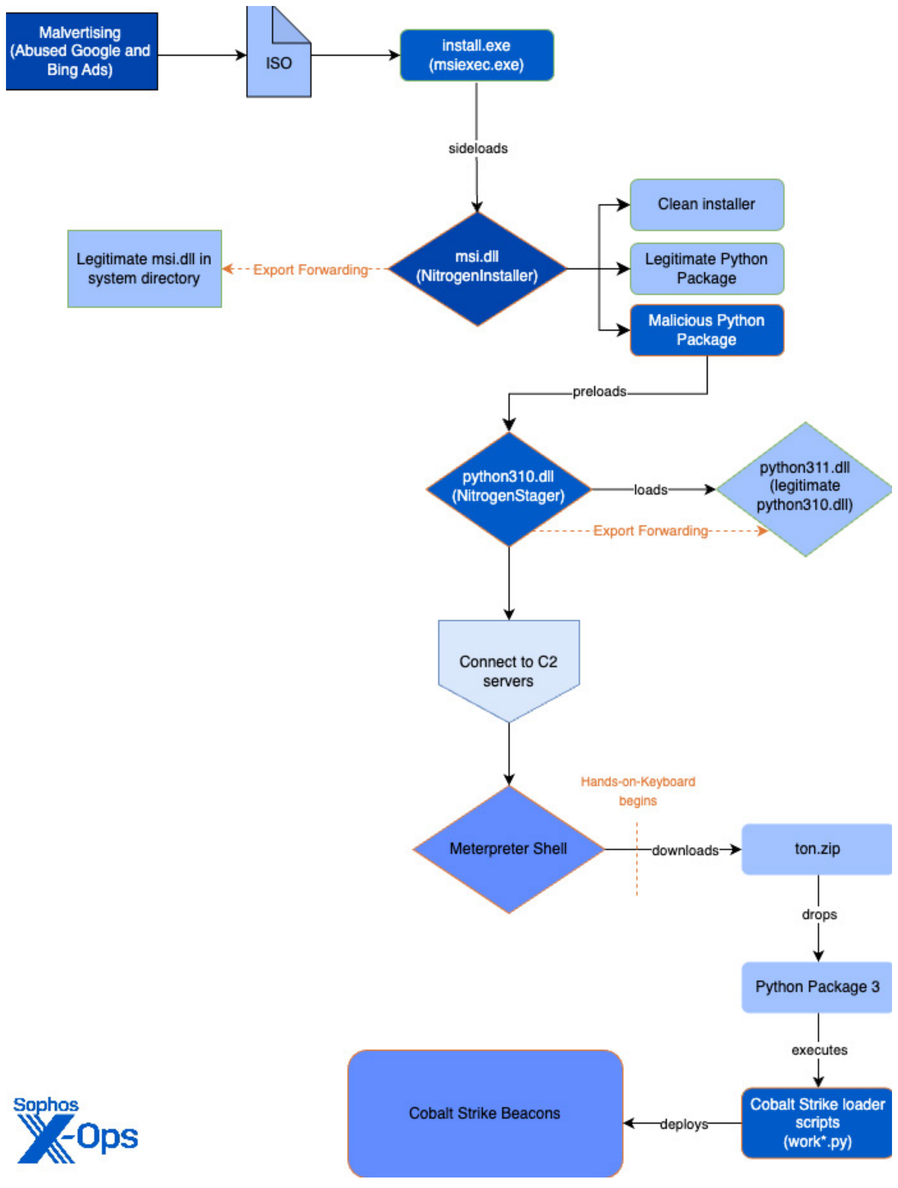

Malvertising (Abused Google and Bing Ads)
ISO
install.exe (msiexec.exe)
sideloads
Legitimate msi.dll in system directory
Export Forwarding
msi.dll (NitrogenInstaller)
Clean installer
Legitimate Python Package
Malicious Python Package
preloads
python310.dll (NitrogenStager)
loads
python311.dll (legitimate python310.dll)
Export Forwarding
Connect to C2 servers
Hands-on-Keyboard begins
Meterpreter Shell
downloads
ton.zip
drops
Python Package 3
executes
Cobalt Strike loader scripts (work*.py)
deploys
Cobalt Strike Beacons
Sophos X-Ops

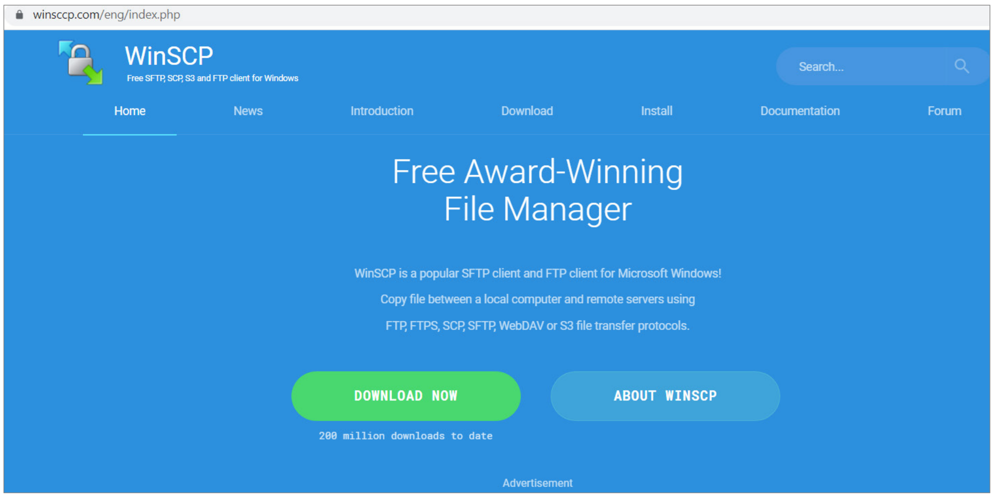
Figure 17: Nitrogen infection process.

The observed infection chain starts with malvertising via *Google* and *Bing Ads* to lure users to compromised *WordPress* sites and phishing pages impersonating popular software distribution sites, where they are tricked into downloading trojanized ISO installers.

When downloaded, the installers side load the malicious NitrogenInstaller DLL containing a legitimate software application bundled with a malicious Python execution environment. The Python package uses Dynamic Link Library (DLL) preloading to execute the malicious NitrogenStager file, which connects to the threat actor's command-and-control (C2) servers to drop both a Meterpreter shell and Cobalt Strike Beacons onto the targeted system. Throughout the infection chain, the threat actors use uncommon export forwarding and DLL preloading techniques to mask their malicious activity and hinder analysis.

As reported by @malwareinfosec [5], when a user searches *Google* for WinSCP, a *Google Ad* will pop up, referencing 'Secure File Transfer – For Windows' on the site `softwareinteractivo[.]com`, which is a phishing page that impersonates a guidance blog for system administrators.

When the advertisement on softwareinteractivo[.]com is clicked, it redirects the user to a fake download page for *WinSCP 6.1* (`winsccp[.]com`), which drops a malicious ISO file on the user's computer.

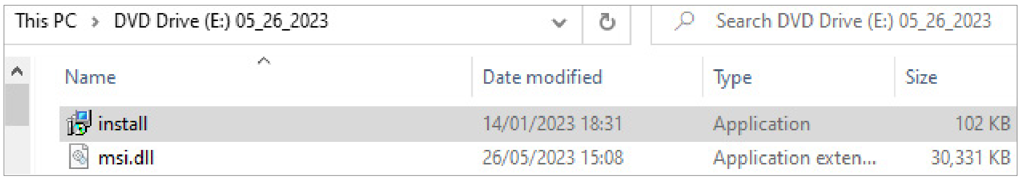

winsccp.com/eng/index.php
WinSCP
Free SFTP, SCP, S3 and FTP client for Windows
Search...
Home
News
Introduction
Download
Install
Documentation
Forum
Free Award-Winning File Manager
DOWNLOAD NOW
ABOUT WINSCP
200 million downloads to date
Advertisement

WinSCP is a popular SFTP client and FTP client for Microsoft Windows! Copy file between a local computer and remote servers using FTP, FTPS, SCP, SFTP, WebDAV or S3 file transfer protocols.

Figure 18: Winsccp[.]com is a malicious website mimicking the real WinSCP download page (winscp.net).

The downloaded trojanized installers are ISO disk images. These files then mount in *Windows Explorer* and can be mapped to a drive, where the content will be available.

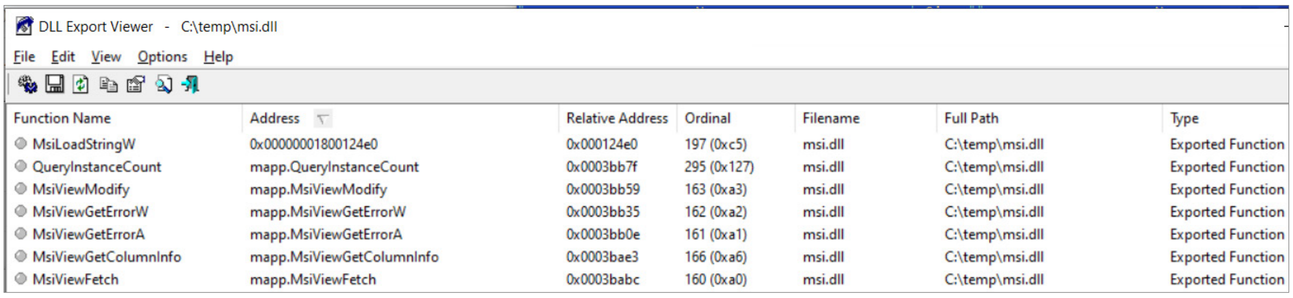
One of the files inside the ISO image is the `msiexec.exe` *Windows* tool, renamed to `install.exe` or `setup.exe`. When executed, the renamed `msiexec.exe` side loads the malicious `msi.dll` (NitrogenInstaller) file stored in the same image.


This PC > DVD Drive (E:) 05_26_2023
Search DVD Drive (E:) 05_26_2023

Name	Date modified	Type	Size
install	14/01/2023 18:31	Application	102 KB
msi.dll	26/05/2023 15:08	Application exten...	30,331 KB

Figure 19: Content of the trojanized installer image.

In this Nitrogen campaign, the threat actors use a tactic that is less commonly seen in side-loading attempts: using DLL proxying by forwarding exported functions (except for the main function `MsiLoadStringW`, which contains the malicious code) to the legitimate `msi.dll`, which resides in the system directory. Though DLL proxying is not a particularly novel technique, it typically occurs in DLL hijacking game hacks rather than in DLL side loading or preloading.


DLL Export Viewer - C:\temp\msi.dll
File Edit View Options Help

Function Name	Address	Relative Address	Ordinal	Filename	Full Path	Type
MsiLoadStringW	0x00000001800124e0	0x000124e0	197 (0xc5)	msi.dll	C:\temp\msi.dll	Exported Function
QueryInstanceCount	mapp.QueryInstanceCount	0x0003bb7f	295 (0x127)	msi.dll	C:\temp\msi.dll	Exported Function
MsiViewModify	mapp.MsiViewModify	0x0003bb59	163 (0xa3)	msi.dll	C:\temp\msi.dll	Exported Function
MsiViewGetErrorW	mapp.MsiViewGetErrorW	0x0003bb35	162 (0xa2)	msi.dll	C:\temp\msi.dll	Exported Function
MsiViewGetErrorA	mapp.MsiViewGetErrorA	0x0003bb0e	161 (0xa1)	msi.dll	C:\temp\msi.dll	Exported Function
MsiViewGetColumnInfo	mapp.MsiViewGetColumnInfo	0x0003bae3	166 (0xa6)	msi.dll	C:\temp\msi.dll	Exported Function
MsiViewFetch	mapp.MsiViewFetch	0x0003babc	160 (0xa0)	msi.dll	C:\temp\msi.dll	Exported Function

Figure 20: Exported functions of msi.dll.

The side-loaded `msi.dll` file – which the threat actors call NitrogenInstaller – proceeds to drop a clean installer for the legitimate decoy application (e.g. *Inno* installer for *WinSCP*) alongside two Python packages, both of which are about 8-10MB in size: a legitimate Python archive and a trojanized Python package containing the malicious `python310.dll` file (NitrogenStager).

Some of the NitrogenInstaller samples contained debug information, such as PDB paths, which gives an insight into the project structure:

```
Y:\nitronet\nitrogen\x64\Release - msi.dll\Nitrogen.pdb
Y:\x64\Release - msi.dll\Nitrogen.pdb
```

The two Python packages serve different roles, one of them is a legitimate package, the other is trojanized:

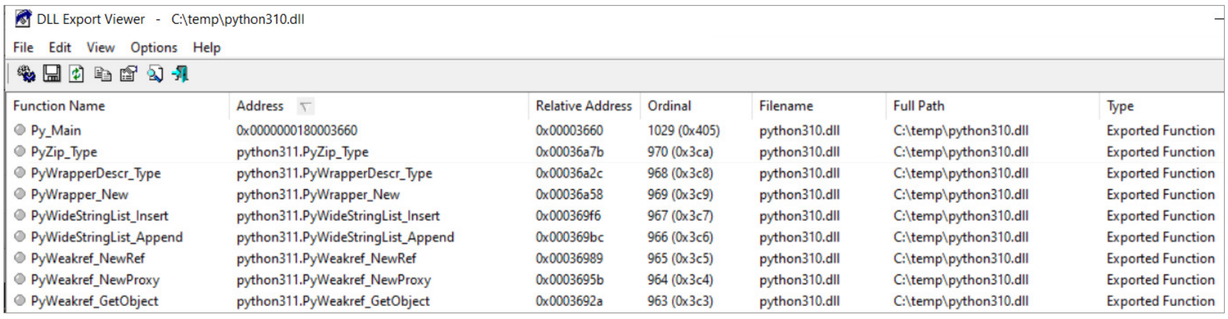
Python package	**Directory**	**Purpose**
BeaconPack Python package (legitimate)	The Videos directory within the Public folder	This package has the main component in `\bof\__pycache__\__init__.cpython-310.pyc`. It is based on the COFFLoader package [6] and uses this component to load Beacon Object Files. The main class of COFFLoader is called BeaconPack.
NitrogenStager Python package	My Music and Music directories within the Public and All Users folders	This package contains the trojanized `python310.dll` and is used to connect to the C2 servers and run the Meterpreter shell.

The NitrogenStager Python package is unviable; it cannot execute Python scripts.

However, for the threat actors to be able to conduct the later stages of the attack, such as the installation of Cobalt Strike Beacons, they need a working Python environment. This explains why the threat actors dropped the legitimate BeaconPack Python package: to execute Python code needed later in the infection chain.

To load the NitrogenStager DLL in the malicious Python package, the threat actors renamed the legitimate DLL (`python310.dll`) to `python311.dll` (which is stored in the same directory) and copied their own specially crafted malicious stager (NitrogenStager) into the directory under the name `python310.dll`.

The NitrogenStager Python package is unable to execute Python scripts, as its main function is replaced with malicious connect-back code and all other exports in the package are forwarded to the original legitimate Python DLL (`python311.dll`):


DLL Export Viewer - C:\temp\python310.dll
File Edit View Options Help

Function Name	Address	Relative Address	Ordinal	Filename	Full Path	Type
Py_Main	0x0000000180003660	0x00003660	1029 (0x405)	python310.dll	C:\temp\python310.dll	Exported Function
PyZip_Type	python311.PyZip_Type	0x00036a7b	970 (0x3ca)	python310.dll	C:\temp\python310.dll	Exported Function
PyWrapperDescr_Type	python311.PyWrapperDescr_Type	0x00036a2c	968 (0x3c8)	python310.dll	C:\temp\python310.dll	Exported Function
PyWrapper_New	python311.PyWrapper_New	0x00036a58	969 (0x3c9)	python310.dll	C:\temp\python310.dll	Exported Function
PyWideStringList_Insert	python311.PyWideStringList_Insert	0x000369f6	967 (0x3c7)	python310.dll	C:\temp\python310.dll	Exported Function
PyWideStringList_Append	python311.PyWideStringList_Append	0x000369bc	966 (0x3c6)	python310.dll	C:\temp\python310.dll	Exported Function
PyWeakref_NewRef	python311.PyWeakref_NewRef	0x00036989	965 (0x3c5)	python310.dll	C:\temp\python310.dll	Exported Function
PyWeakref_NewProxy	python311.PyWeakref_NewProxy	0x0003695b	964 (0x3c4)	python310.dll	C:\temp\python310.dll	Exported Function
PyWeakref_GetObject	python311.PyWeakref_GetObject	0x0003692a	963 (0x3c3)	python310.dll	C:\temp\python310.dll	Exported Function

Figure 21: Python310.dll refers to the export in python311.dll as python311.{exportname}.

This tactic is similar to the export forwarding technique used earlier when side loading `msi.dll`; however, in this case, the original clean DLL was part of the package as a renamed DLL instead of already residing on the system.

The malicious connect-back code in the `Py_Main` function runs automatically upon execution. *Sophos* detected NitrogenStager connecting to C2 servers using four different protocols (TCP, TCP over SSL, HTTP and HTTPS). The package contains a separate script for each protocol used, each of which has the functionality to connect to the C2 server, decode responses, and execute them. The stagers for the protocols are based on public domain tools likely generated by msfvenom [7], the payload-generating component of the Metasploit Framework.

Like the NitrogenInstaller sample, some of the NitrogenStager samples also contain debug information, including PDB paths:

```
Z:\projects\nitrogen_vs\x64\Release - python310emb\Nitrogen.pdb
Y:\x64\Release - python310emb\Nitrogen.pdb
Y:\nitronet\nitrogen\x64\Release - msi.dll\Nitrogen.pdb
```

We have not observed the endgame of these infections. But external reports draw connection between Nitrogen infections and ransomware deployment, so our current assumption is that these campaigns are run by either a ransomware group or an initial access broker.

3CXDESKTOP – ADDITIONS TO OPEN-SOURCE COMPONENT

Sophos MDR first identified malicious activity directed at its own customers and stemming from *3CXDesktopApp* on 29 March 2023. This is not a DLL side-loading scenario, but it deserves an honorary mention.

The attack uses a remarkable number of components. This is likely to ensure that customers were able to use the *3CX* desktop package without noticing anything unusual about the affected package. We have identified three crucial components:

- 3CXDesktopApp.exe, the clean loader
- ffmpeg.dll, a trojanized loader
- d3dcompiler_47.dll, a DLL with an appended encrypted payload

Figure 22 presents a high-level look at the attack flow as it works in *Windows*; there are some minor variations in the later steps with the *MacOS* version.

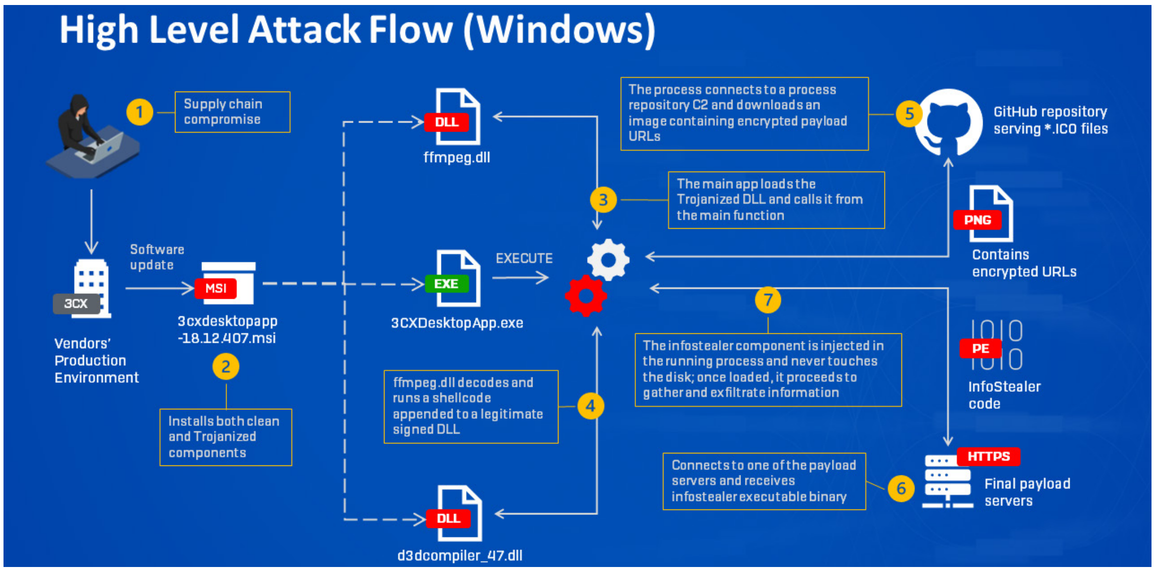

High Level Attack Flow (Windows)
1
Supply chain compromise
3CX
Vendors' Production Environment
Software update
MSI
3cxdesktopapp -18.12.407.msi
2
Installs both clean and Trojanized components
DLL
ffmpeg.dll
EXE
3CXDesktopApp.exe
EXECUTE
DLL
d3dcompiler_47.dll
ffmpeg.dll decodes and runs a shellcode appended to a legitimate signed DLL
4
3
The main app loads the Trojanized DLL and calls it from the main function
The process connects to a process repository C2 and downloads an image containing encrypted payload URLs
5
GitHub repository serving *.ICO files
PNG
Contains encrypted URLs
7
The infostealer component is injected in the running process and never touches the disk; once loaded, it proceeds to gather and exfiltrate information
PE
InfoStealer code
Connects to one of the payload servers and receives infostealer executable binary
6
HTTPS
Final payload servers

Figure 22: A high-level view of the attack flow.

In a normal DLL side-loading scenario, the malicious loader (ffmpeg.dll) would replace the clean dependency; its only function would be to queue up the payload. However, in this case, that loader is also entirely functional, as it normally would be in the *3CX* product – instead, there's an additional payload inserted at the DllMain function. Appending malicious code in this fashion adds bulk, but may have lowered suspicions – the abused *3CX* application functioned as expected, even as the trojan addresses reached out to the C2 beacon.

As part of our analysis, we also did a comparison of the ffmpeg.dll in *3CX* with the same file in other *Electron* apps. Our analysis has shown only the *3CX* ffmpeg.dll contains the malicious code. We conclude from this that this compromise does not affect other *Electron* apps, only the *3CX* ffmpeg.dll.

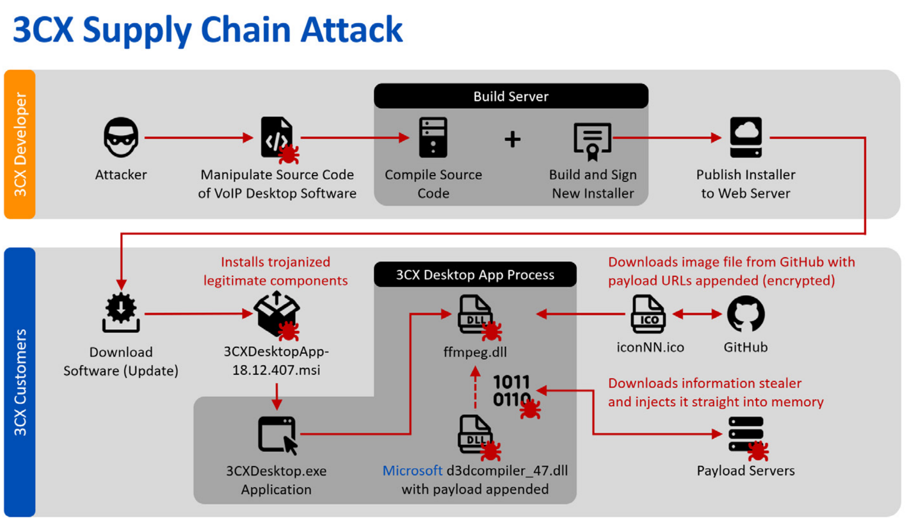

3CX Supply Chain Attack
3CX Developer
Attacker
Manipulate Source Code of VoIP Desktop Software
Build Server
Compile Source Code
+
Build and Sign New Installer
Publish Installer to Web Server
3CX Customers
Download Software (Update)
Installs trojanized legitimate components
3CXDesktopApp-18.12.407.msi
3CXDesktop.exe Application
3CX Desktop App Process
ffmpeg.dll
1011 0110
Microsoft d3dcompiler_47.dll with payload appended
Downloads image file from GitHub with payload URLs appended (encrypted)
iconNN.ico
GitHub
Downloads information stealer and injects it straight into memory
Payload Servers

Figure 23: What the affected 3CX developers and customers experienced.

Allowing the abused software to remain functional is not dissimilar to other DLL side-loading cases we've seen, but this campaign is slightly different even from the current rash of DLL side-loading cases. In particular, we've noted that the PE shellcode loader in use is unique in our experience. Previous to this, we've only seen it in incidents attributed to the Lazarus group; the code in this incident is a byte-to-byte match to those previous samples.

Conclusions

- Side loading is still alive.
- It may be old, but it still works fine.
- Localized attacks aim to stay under the radar of the security community.

REFERENCES

[1] Operation Dragon Breath – APT-Q-27. https://www.ctfiot.com/40522.html (in Chinese).

[2] Operation Dragon Breath – APT-Q-27. https://zhuanlan.zhihu.com/p/515150114 (in Chinese).

[3] Silva, L.; Caragay, RJ.; Dela Cruz, A.; Cardoso, G. Malvertising Used as Entry Vector for BlackCat, Actors Also Leverage SpyBoy Terminator. Trend Micro. 30 June 2023. https://www.trendmicro.com/en_us/research/23/f/malvertising-used-as-entry-vector-for-blackcat-actors-also-lever.html.

[4] Sophos. BlackCat. https://news.sophos.com/en-us/tag/blackcat/.

[5] https://infosec.exchange/@malwareinfosec/110539178322032927.

[6] COFFLoader. https://github.com/trustedsec/COFFLoader.

[7] Msfvenom. https://docs.metasploit.com/docs/using-metasploit/basics/how-to-use-msfvenom.html.