holmesglen

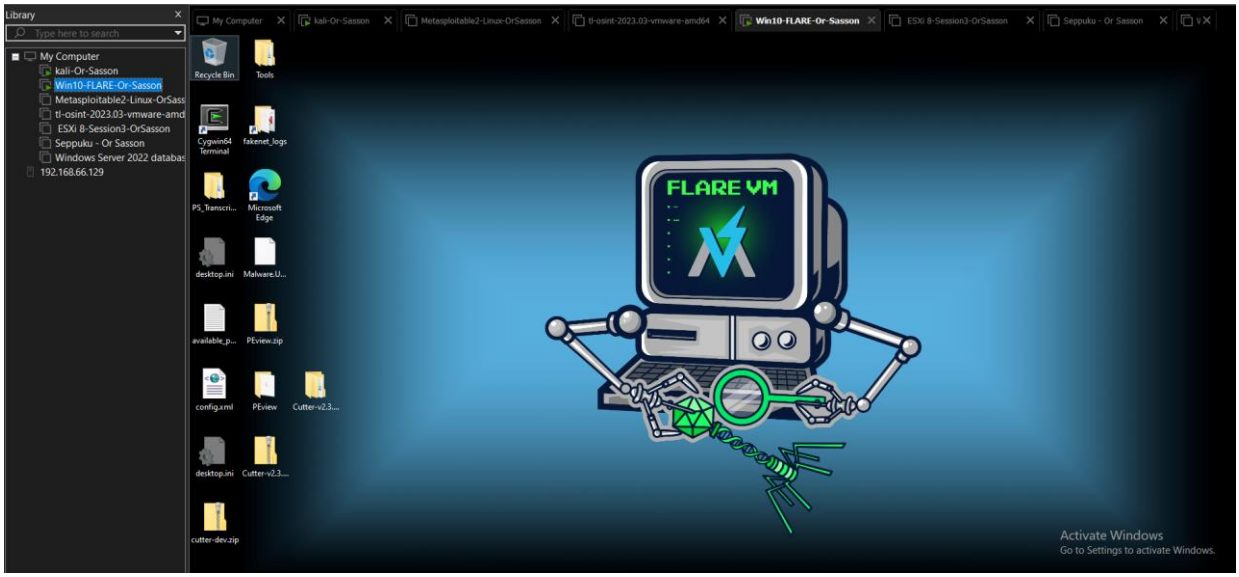
Assessment Task 2: Portfolio of Software Skills

Student answer sheet / Marking sheet

Unit title:	Develop software skills for the cyber security practitioner	**Unit code:**	VU23291

Part A: Prepare virtual environment and provide screenshots of the running virtual machines.

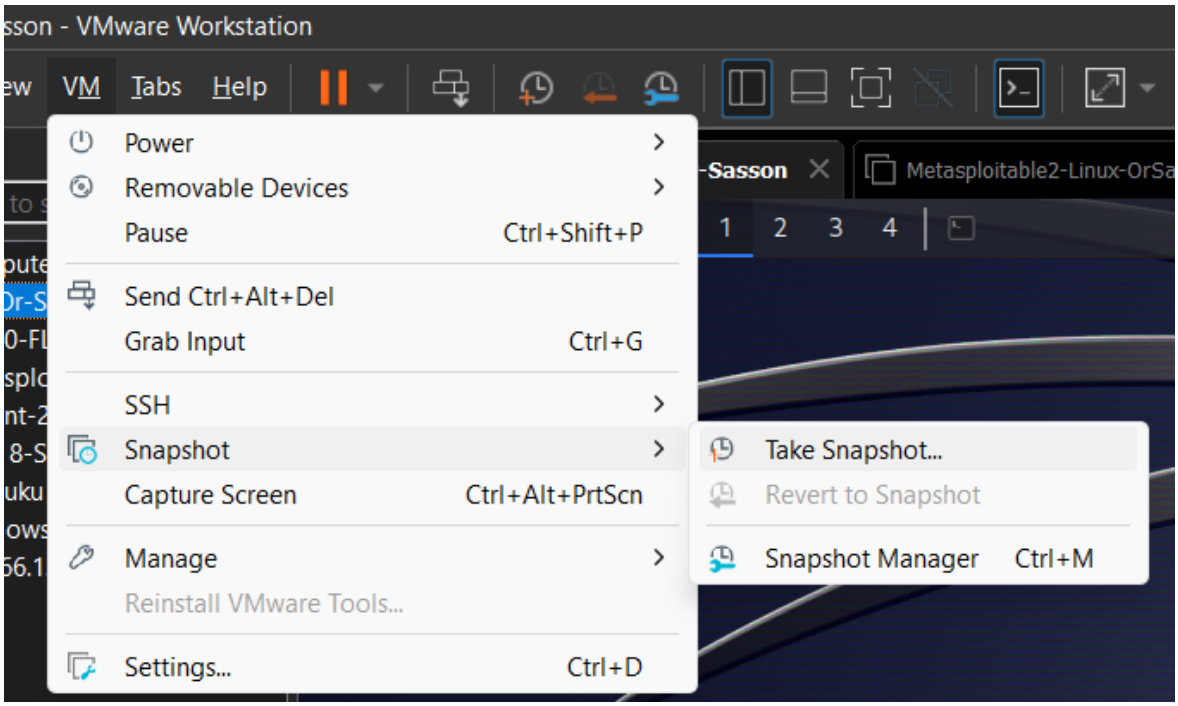
1. Screen shot of Windows 10 VM

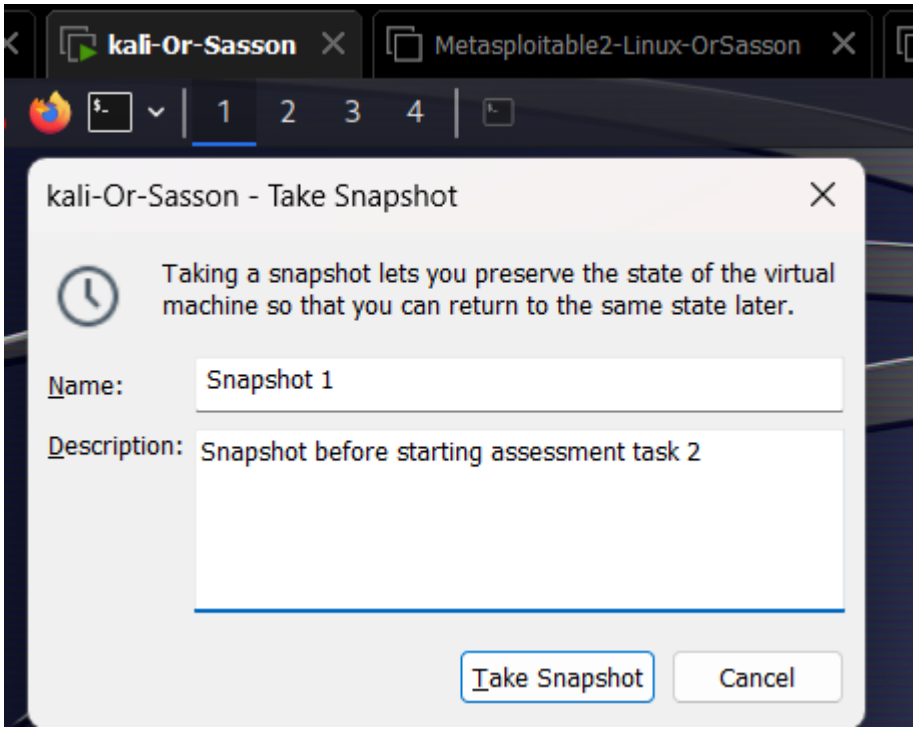


2. Screen shot Kali Linux VM

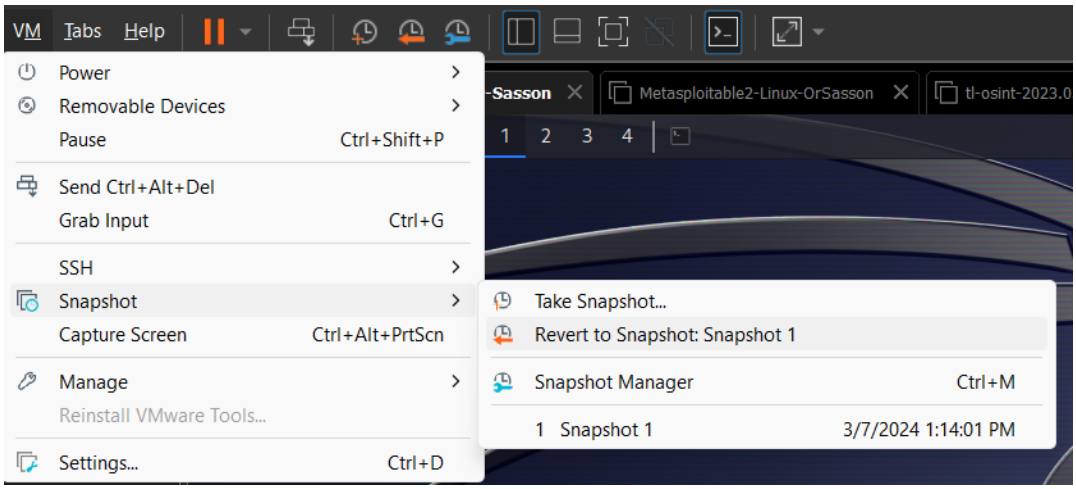
3. Provide process of taking snapshot of virtual machine.

Assessment Task 2: Portfolio of Software Skills



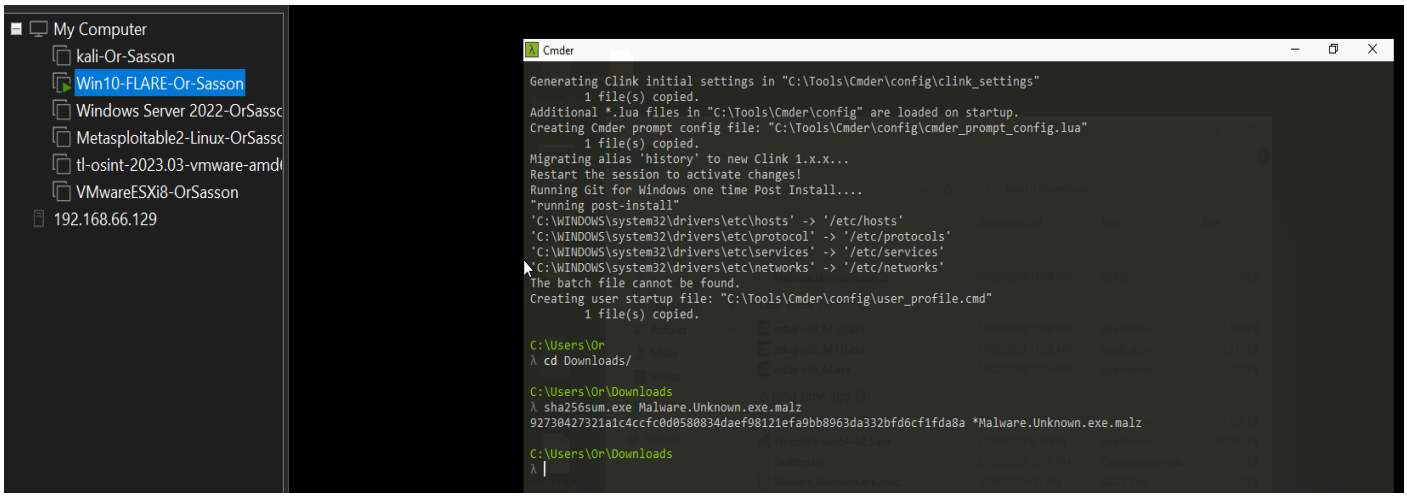


Assessment Task 2: Portfolio of Software Skills

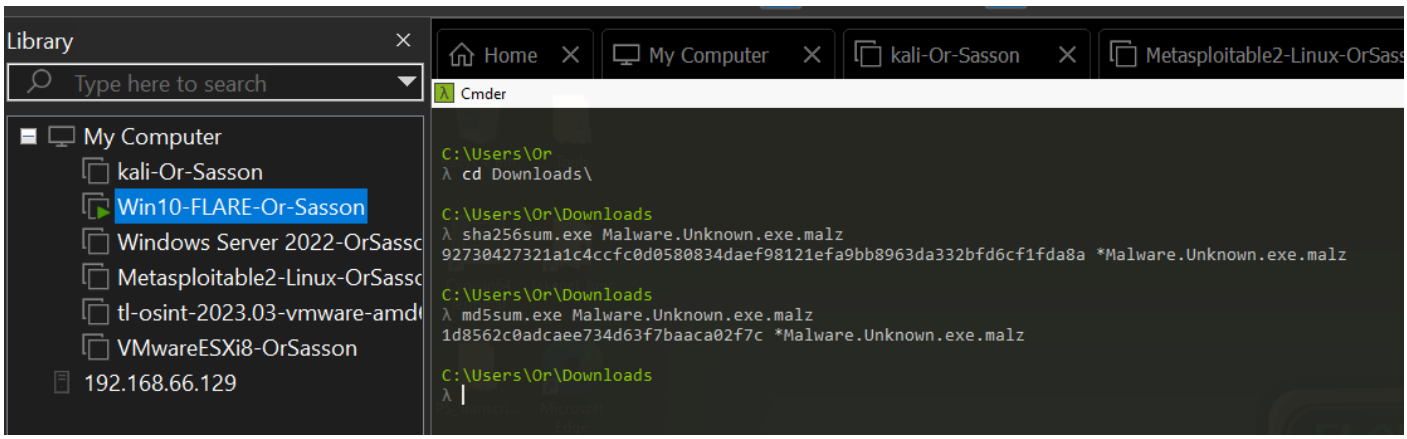


Part B Malware Analysis
(please make sure that you work in a safe environment to complete this task.)
SAMPLE MALEWARE FILE CAN BE DOWNLOADED FROM:
https://drive.google.com/drive/folders/1LjpNm5orbzgrrwfGfVRH608hmi5YRKEh?usp=sharing

1. Find Hashes of Infected file

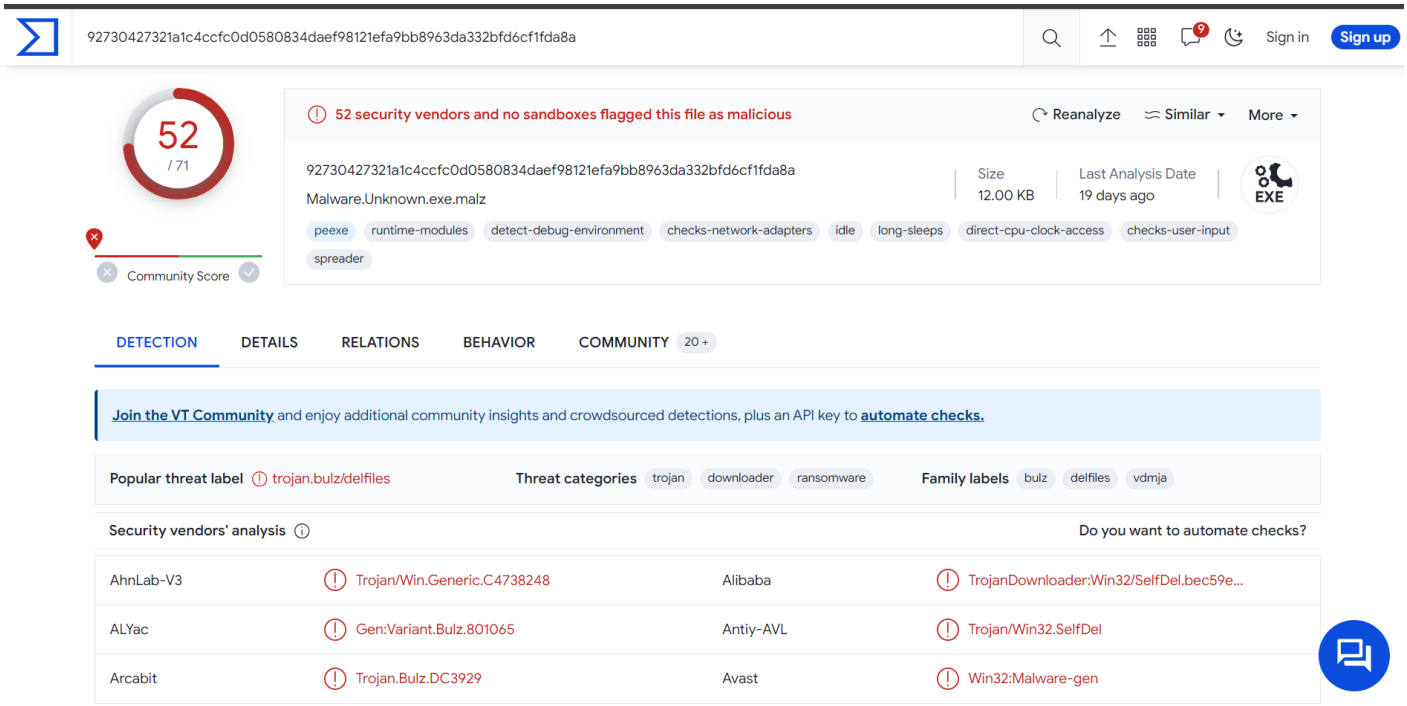


2. Find MD5 of infected file



Assessment Task 2: Portfolio of Software Skills

3. Provide Virustotal.com result of Hash/ md5 sum of infected file



holmesglen

Assessment Task 2: Portfolio of Software Skills

AVG	Win32:Malware-gen	Avira (no cloud)	TR/DelFiles.vdmja
BitDefender	Gen:Variant.Bulz.801065	CrowdStrike Falcon	Win/malicious_confidence_100% (W)
Cybereason	Malicious.0e7243	Cylance	Unsafe
Cynet	Malicious (score: 100)	DeepInstinct	MALICIOUS
DrWeb	Trojan.MulDrop19.15754	Elastic	Malicious (high Confidence)
Emsisoft	Gen:Variant.Bulz.801065 (B)	eScan	Gen:Variant.Bulz.801065
ESET-NOD32	Win32/TrojanDownloader.Small.BKM	Fortinet	W32/PossibleThreat
GData	Gen:Variant.Bulz.801065	Google	Detected
Gridinsoft (no cloud)	Ransom.Win32.Sabsik.oa!s1	Ikarus	Trojan-Downloader.Win32.Small
Jiangmin	Trojan.Jobutyve.i	K7AntiVirus	Trojan-Downloader (0058a8611)
K7GW	Trojan-Downloader (0058a8611)	Kaspersky	HEUR:Trojan.Win32.SelfDel.gen
Lionic	Trojan.Win32.DelFiles.4!c	Malwarebytes	Trojan.SelfDelete
MAX	Malware (ai Score=100)	MaxSecure	Trojan.Malware.73875556.susgen
McAfee	RDN/Ransom	Microsoft	Ransom:Win32/Cobra!mclg
Panda	Trj/GdSda.A	Rising	Downloader.Small!8.B41 (TFE:5:L7ehqQid...
Sangfor Engine Zero	Downloader.Win32.Small.Vbg5	SentinelOne (Static ML)	Static AI - Malicious PE
Skyhigh (SWG)	BehavesLike.Win32.Generic.lm	Sophos	Mal/Generic-S
Tencent	Malware.Win32.Gencirc.11b8bdf8	Trellix (FireEye)	Generic.mg.1d8562c0adcaee73
TrendMicro	TROJ_GEN.R002C0PH923	TrendMicro-HouseCall	TROJ_GEN.R002C0PH923
Varist	W32/ABRisk.WXPJ-7017	VBA32	Trojan.SelfDel
VIPRE	Gen:Variant.Bulz.801065	ViRobot	Trojan.Win32.Z.Agent.12288.EBS
Webroot	W32.Trojan.TR.DelFiles.vdmja	WithSecure	Trojan.TR/DelFiles.vdmja

Assessment Task 2: Portfolio of Software Skills

Zillya	Downloader.Small.Win32.140841	ZoneAlarm by Check Point	HEUR:Trojan.Win32.SelfDel.gen
Acronis (Static ML)	Undetected	Baidu	Undetected
BitDefenderTheta	Undetected	Bkav Pro	Undetected
ClamAV	Undetected	CMC	Undetected
Kingsoft	Undetected	NANO-Antivirus	Undetected
Palo Alto Networks	Undetected	QuickHeal	Undetected
SecureAge	Undetected	SUPERAntiSpyware	Undetected
TACHYON	Undetected	TEHTRIS	Undetected
Trapmine	Undetected	VirIT	Undetected
Xcitium	Undetected	Yandex	Undetected
Zoner	Undetected	Avast-Mobile	Unable to process file type
BitDefenderFalx	Unable to process file type	Symantec Mobile Insight	Unable to process file type
Trustlook	Unable to process file type	Symantec	—

4. Use FLOSS to extract strings from binary of infected file, provided detailed analysis
screenshot + Explanation

In those two screenshots after running floss on the infected file (Malware.Unknown.exe), I couldn't find anything which could help me to analyse this file to find its purpose.

Assessment Task 2: Portfolio of Software Skills

Metasploitable2-Linux-OrSasson | tl-osint-2023.03-vmware-amd64 | VMwareESXi8-OrSasson | Win10-FLARE-Or-Sasson

Administrator: FLOSS

```
FLARE-VM Wed 21/02/2024 22:05:42.79
C:\Users\Or\Desktop>floss Malware.Unknown.exe.malz
INFO: floss: extracting static strings
finding decoding function features: 100%|| 69/69 [00:00<00:00, 1389.47 functions/s, skipped 25 library functions (36%)
INFO: floss.stackstrings: extracting stackstrings from 22 functions
INFO: floss.results: ineIGenu
extracting stackstrings: 100%|| 22/22 [00:00<00:00, 56.43 functions/s]
INFO: floss.tightstrings: extracting tightstrings from 0 functions...
extracting tightstrings: 0 functions [00:00, ? functions/s]
INFO: floss.string_decoder: decoding strings
emulating function 0x401be2 (call 1/1): 100%|| 20/20 [00:00<00:00, 20.65 functions/s]
INFO: floss: finished execution after 16.22 seconds
INFO: floss: rendering results


FLARE FLOSS RESULTS (version v3.0.1-0-g3782dc9)

+-----------------------+-----------------------------------------------------------------------------+
| file path             | Malware.Unknown.exe.malz                                                    |
| identified language   | unknown                                                                     |
| extracted strings     |                                                                             |
|  static strings       | 177 (2521 characters)                                                       |
|   language strings    |   0 (   0 characters)                                                       |
|  stack strings        | 1                                                                           |
|  tight strings        | 0                                                                           |
|  decoded strings      | 0                                                                           |
+-----------------------+-----------------------------------------------------------------------------+
```

Select Administrator: FLOSS

```
+-----------------------------------+

!This program cannot be run in DOS mode.
r&cgr
rRich
.text
`.rdata
@.data
.rsrc
@.reloc
[_^]
[_^]
h02@
u_Ph
@PPh
h81@
hl3@
D$`Ph@1@
SSVRP
SSVRP
|\t;u
jdRP
Y_^[
h01@
=MC@
u"hPC@
h\C@
Y_^[
=LC@
```

Assessment Task 2: Portfolio of Software Skills

In that screenshot I could find this path (in red) which looks like a file which gets downloaded from the internet with URL.

```
%h0@
%d0@
%|0@
WVS3
WVU3
v\tN+D$
RSDSI
C:\Users\Matt\source\repos\HuskyHacks\PMAT-maldev\src\DownloadFromURL\Release\DownloadFromURL.pdb
GCTL
.text$mn
.idata$5
.00cfg
.CRT$XCA
.CRT$XCAA
.CRT$XCZ
```

In the next screenshot I could see the functions I pointed out in red (the pathway), and I did some research to know what they do.

"ShellExecuteW" operates on a specified URL/file.
"Thrd_sleep" which is a function on C++ which delays an execution (might be a malicious execution of a payload)
"URLDownloadToFile" is a function which Downloads bits from the Internet and saves them to a file.
"InternetOpenUrlW" opens a resource specified by a complete FTP or HTTP URL.
"InternetOpenW" is a function which initializes an application's use of the WinINet.
"WinINet" Is API which gives the application the ability to interact with FTP and HTTP protocols so it can access to the internet and using resources.

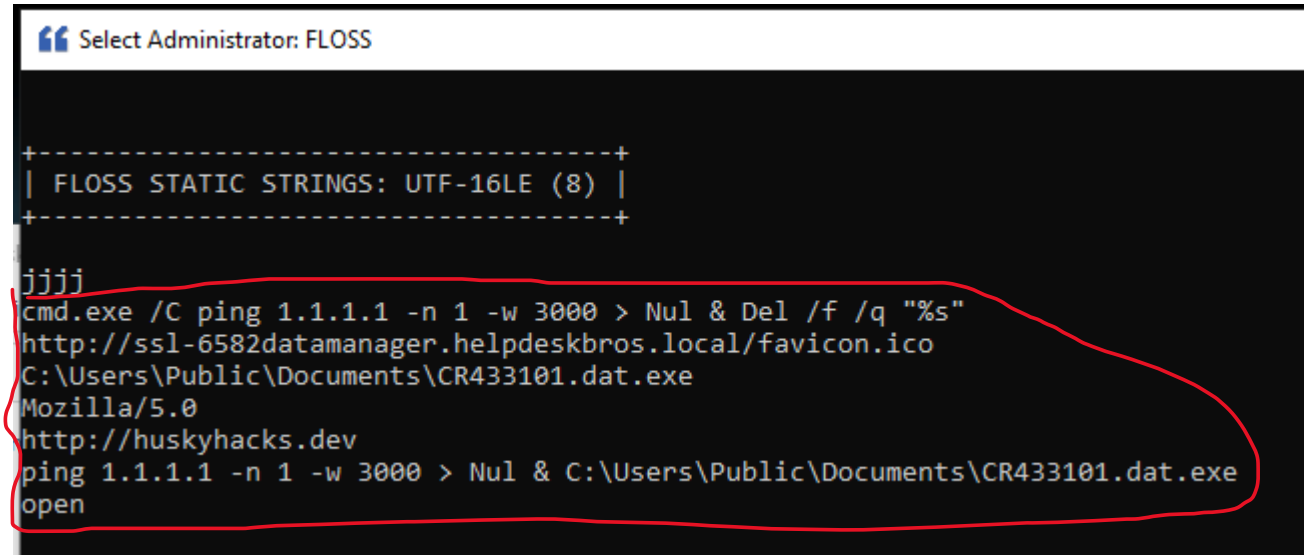
Resources for the research:
https://learn.microsoft.com/en-us/windows/win32/api/wininet/nf-wininet-internetopenurlw

```
KERNEL32.dll
ShellExecuteW
SHELL32.dll
_Query_perf_frequency
_Thrd_sleep
_Query_perf_counter
_Xtime_get_ticks
MSVCP140.dll
URLDownloadToFileW
urlmon.dll
InternetOpenUrlW
InternetOpenW
WININET.dll
__current_exception
__current_exception_context
```

Then, I had a look in the static Unicode strings.
I could see a ping command which is sending one ping to a specific IP address and giving some instructions of discarding and deleting a file with Nul & Del.
There is also a domain name: "helpdeskbros" with an address which can lead somewhere.

Assessment Task 2: Portfolio of Software Skills

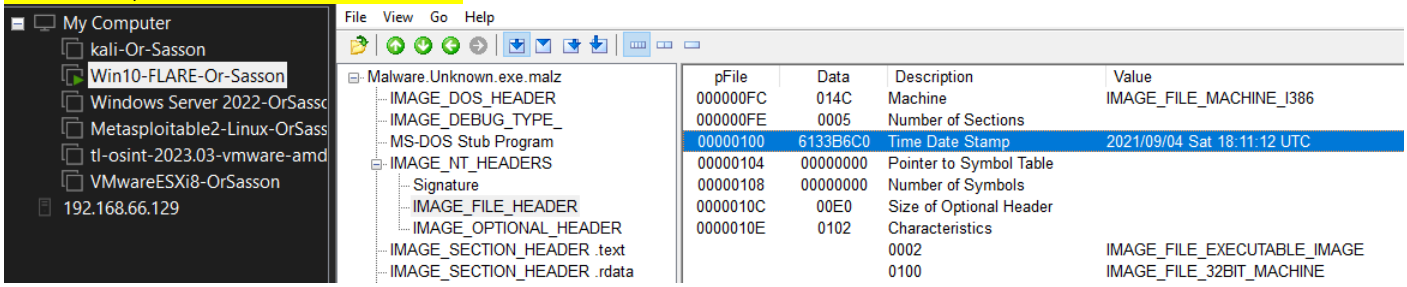
In addition, I could see in the Users/Public/Documents directory an exe file.
There is also a browser name (Muzailla/5.0) with an address that supposed to get opend- http://huskyhacks.dev.
At the end there is another ping command to a specific port number (3000) With the command "Nul" following to discard the exe file in the directory Users/Public/Documents.



5. Find the structure of the binary using PEVIEW utility and answer following questions:

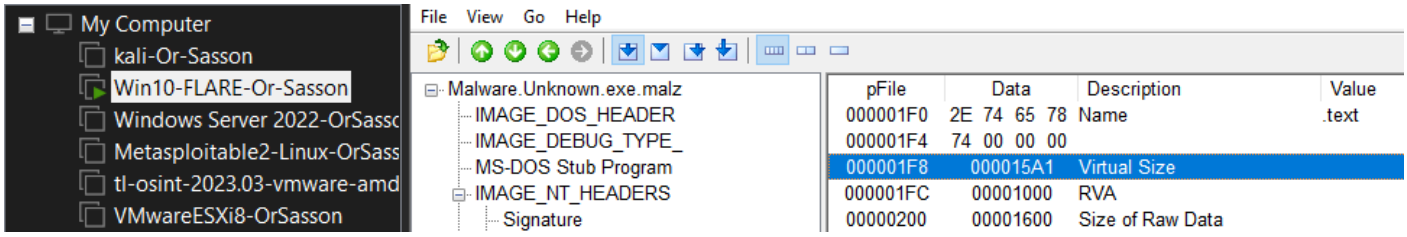
I. When Binary was compiled

It was compiled on 04/09/201 at 18:11



II. Is it Packed or unpacked binary

It seems like it is an unpacked binary, because when I analysed the Virtual Size of the file and the size of the Size of Raw data, it was similar.



The decimal of The Size of Raw Data was:

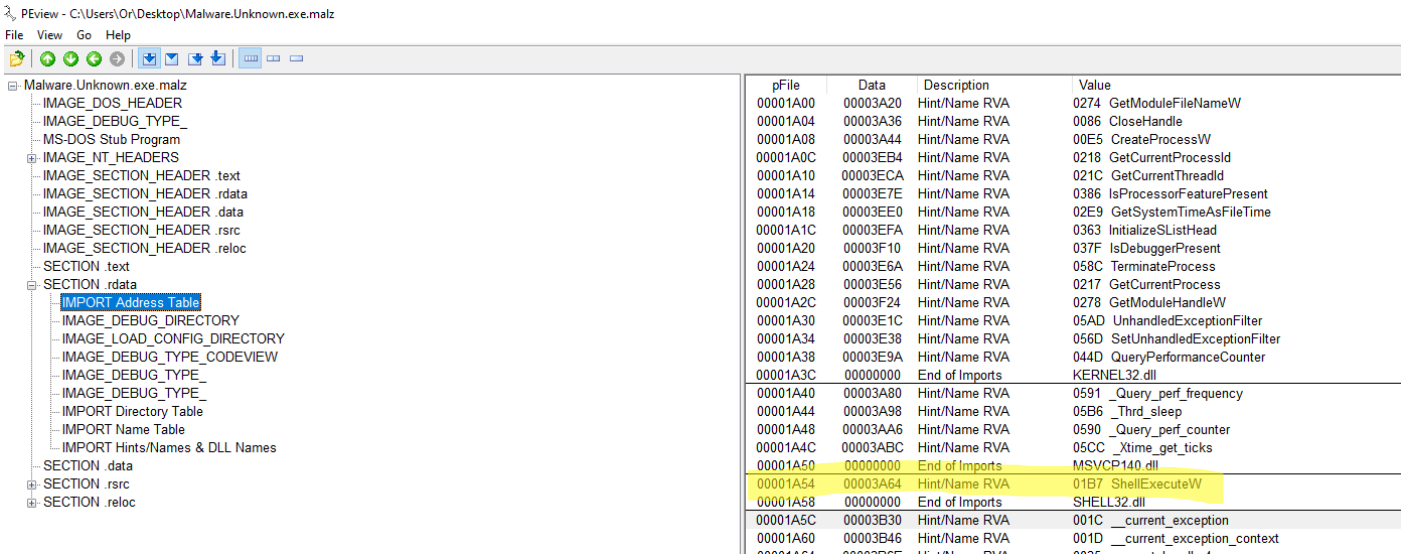
Assessment Task 2: Portfolio of Software Skills

The decimal of the Virtual Size:

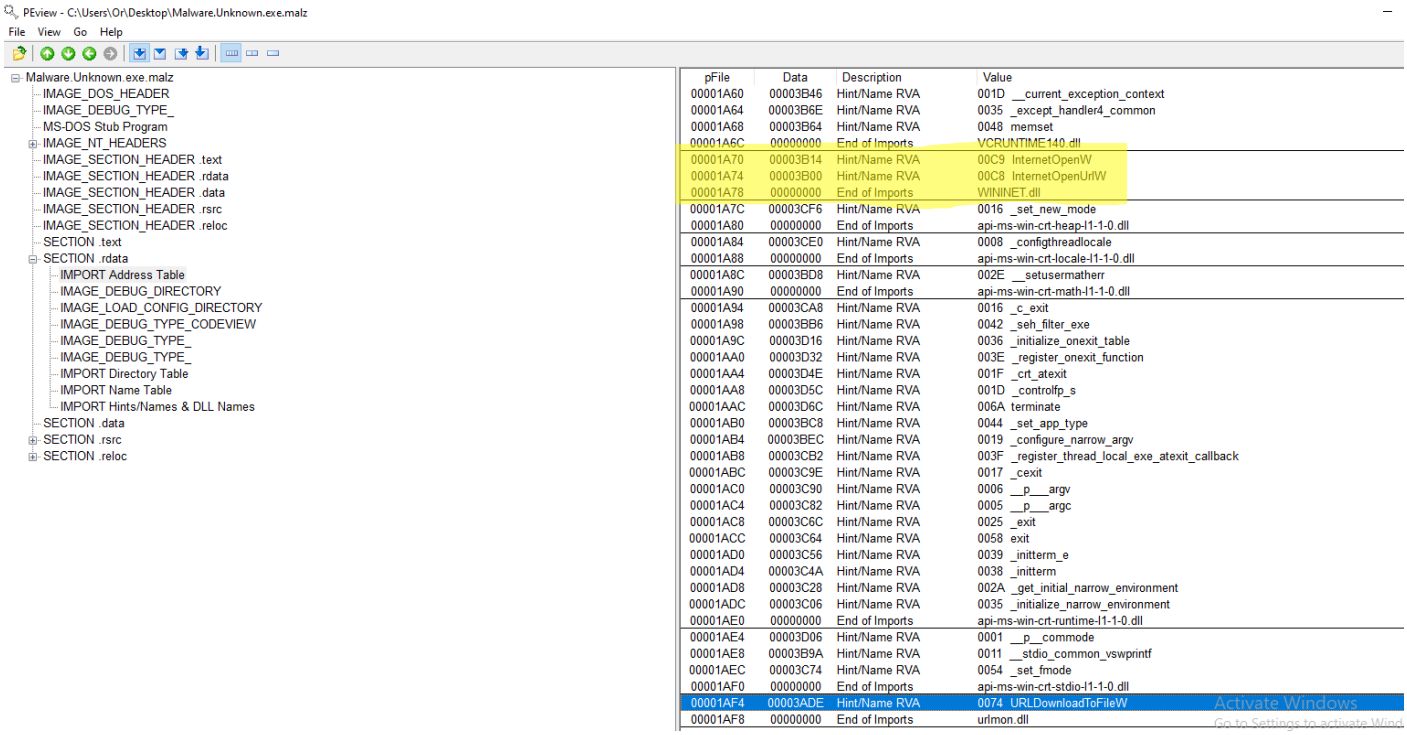
Those two are very similar in size so it indicates us that this is an unpacked binary.

III. Identify some indicators in the API Calls

"ShellExecuteW" operates on a specified URL/file.
"URLDownloadToFile" is a function which Downloads bits from the Internet and saves them to a file.
ÏnternetOpenUrlW" opens a resource specified by a complete FTP or HTTP URL.
"InternetOpenW" is a function which initializes an application's use of the WinINet.
"WinINet" Is API which gives the application the ability to interact with FTP and HTTP protocols so it can access to the internet and using resources.



Assessment Task 2: Portfolio of Software Skills

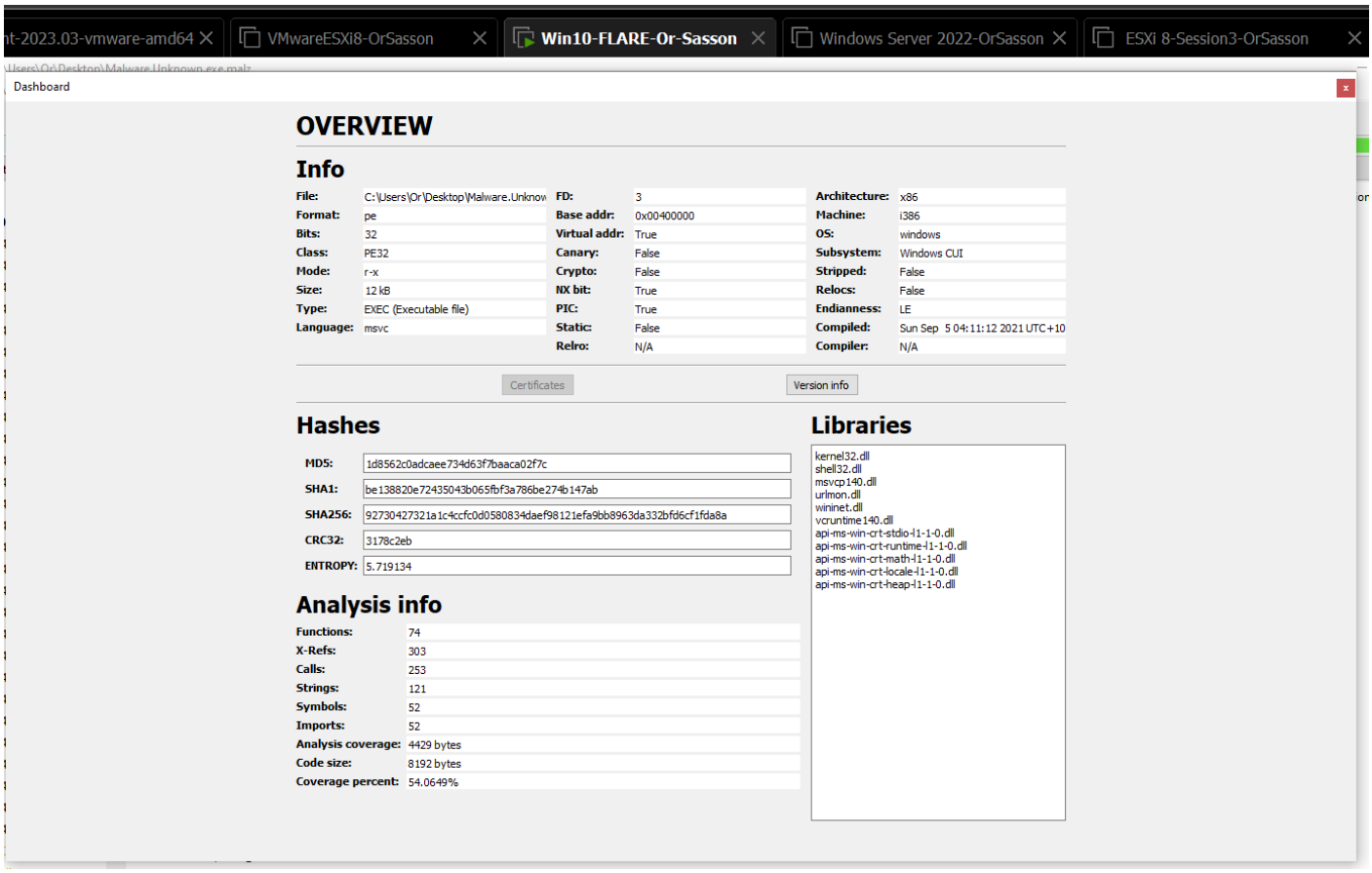


pFile	Data	Description	Value
00001A60	00003B46	Hint/Name RVA	001D __current_exception_context
00001A64	00003B6E	Hint/Name RVA	0035 _except_handler4_common
00001A68	00003B64	Hint/Name RVA	0048 memset
00001A6C	00000000	End of Imports	VCRUNTIME140.dll
00001A70	00003B14	Hint/Name RVA	00C9 InternetOpenW
00001A74	00003B00	Hint/Name RVA	00C8 InternetOpenUrlW
00001A78	00000000	End of Imports	WININET.dll
00001A7C	00003CF6	Hint/Name RVA	0016 _set_new_mode
00001A80	00000000	End of Imports	api-ms-win-crt-heap-l1-1-0.dll
00001A84	00003CE0	Hint/Name RVA	0008 _configthreadlocale
00001A88	00000000	End of Imports	api-ms-win-crt-locale-l1-1-0.dll
00001A8C	00003BD8	Hint/Name RVA	002E __setusermatherr
00001A90	00000000	End of Imports	api-ms-win-crt-math-l1-1-0.dll
00001A94	00003CA8	Hint/Name RVA	0016 _c_exit
00001A98	00003BB6	Hint/Name RVA	0042 _seh_filter_exe
00001A9C	00003D16	Hint/Name RVA	0036 _initialize_onexit_table
00001AA0	00003D32	Hint/Name RVA	003E _register_onexit_function
00001AA4	00003D4E	Hint/Name RVA	001F _crt_atexit
00001AA8	00003D5C	Hint/Name RVA	001D _controlfp_s
00001AAC	00003D6C	Hint/Name RVA	006A terminate
00001AB0	00003BC8	Hint/Name RVA	0044 _set_app_type
00001AB4	00003BEC	Hint/Name RVA	0019 _configure_narrow_argv
00001AB8	00003CB2	Hint/Name RVA	003F _register_thread_local_exe_atexit_callback
00001ABC	00003C9E	Hint/Name RVA	0017 _cexit
00001AC0	00003C90	Hint/Name RVA	0006 __p___argv
00001AC4	00003C82	Hint/Name RVA	0005 __p___argc
00001AC8	00003C6C	Hint/Name RVA	0025 _exit
00001ACC	00003C64	Hint/Name RVA	0058 exit
00001AD0	00003C56	Hint/Name RVA	0039 _initterm_e
00001AD4	00003C4A	Hint/Name RVA	0038 _initterm
00001AD8	00003C28	Hint/Name RVA	002A _get_initial_narrow_environment
00001ADC	00003C06	Hint/Name RVA	0035 _initialize_narrow_environment
00001AE0	00000000	End of Imports	api-ms-win-crt-runtime-l1-1-0.dll
00001AE4	00003D06	Hint/Name RVA	0001 __p__commode
00001AE8	00003B9A	Hint/Name RVA	0011 __stdio_common_vswprintf
00001AEC	00003C74	Hint/Name RVA	0054 _set_fmode
00001AF0	00000000	End of Imports	api-ms-win-crt-stdio-l1-1-0.dll
00001AF4	00003ADE	Hint/Name RVA	0074 URLDownloadToFileW
00001AF8	00000000	End of Imports	urlmon.dll

Part C Reverse Engineering ---Provide screen shots of each step.
Use Cutter Utility to disassemble the sample file
(please make sure that you work in a safe environment to complete this task.)
SAMPLE MALEWARE FILE CAN BE DOWNLOADED FROM:
https://drive.google.com/drive/folders/1LjpNm5orbzgrrwfGfVRH608hmi5YRKEh?usp=sharing

1. Provide screenshot of Overview page and list the interesting detail.

Assessment Task 2: Portfolio of Software Skills



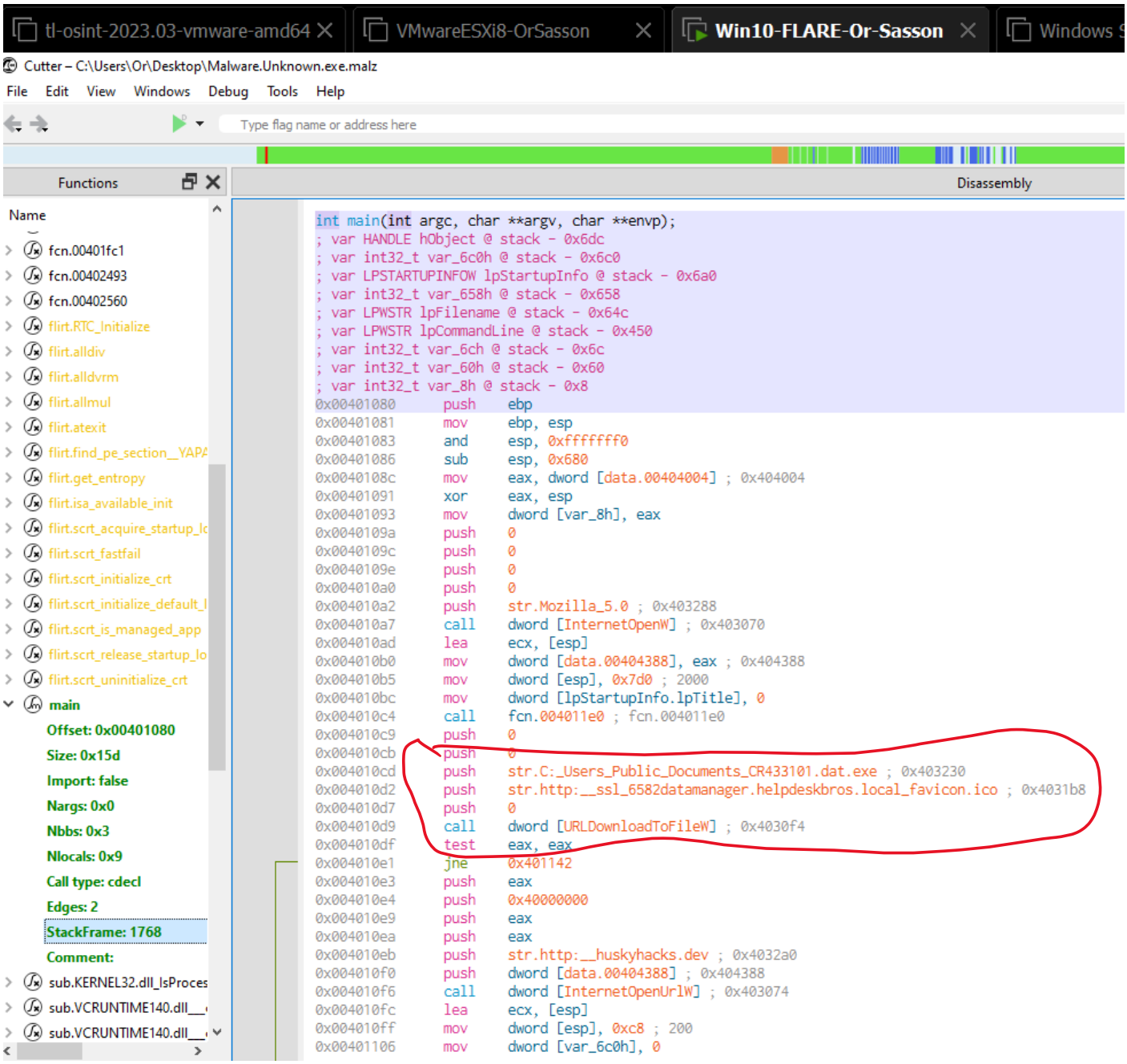
Interesting details are:

- Type of the file
- the compiled date and time
- the Hashes (especially the MD5)
- the size of the code
- libraries

2. Provide the screenshot of assembly code of main function and identify any evil indictor in the code.

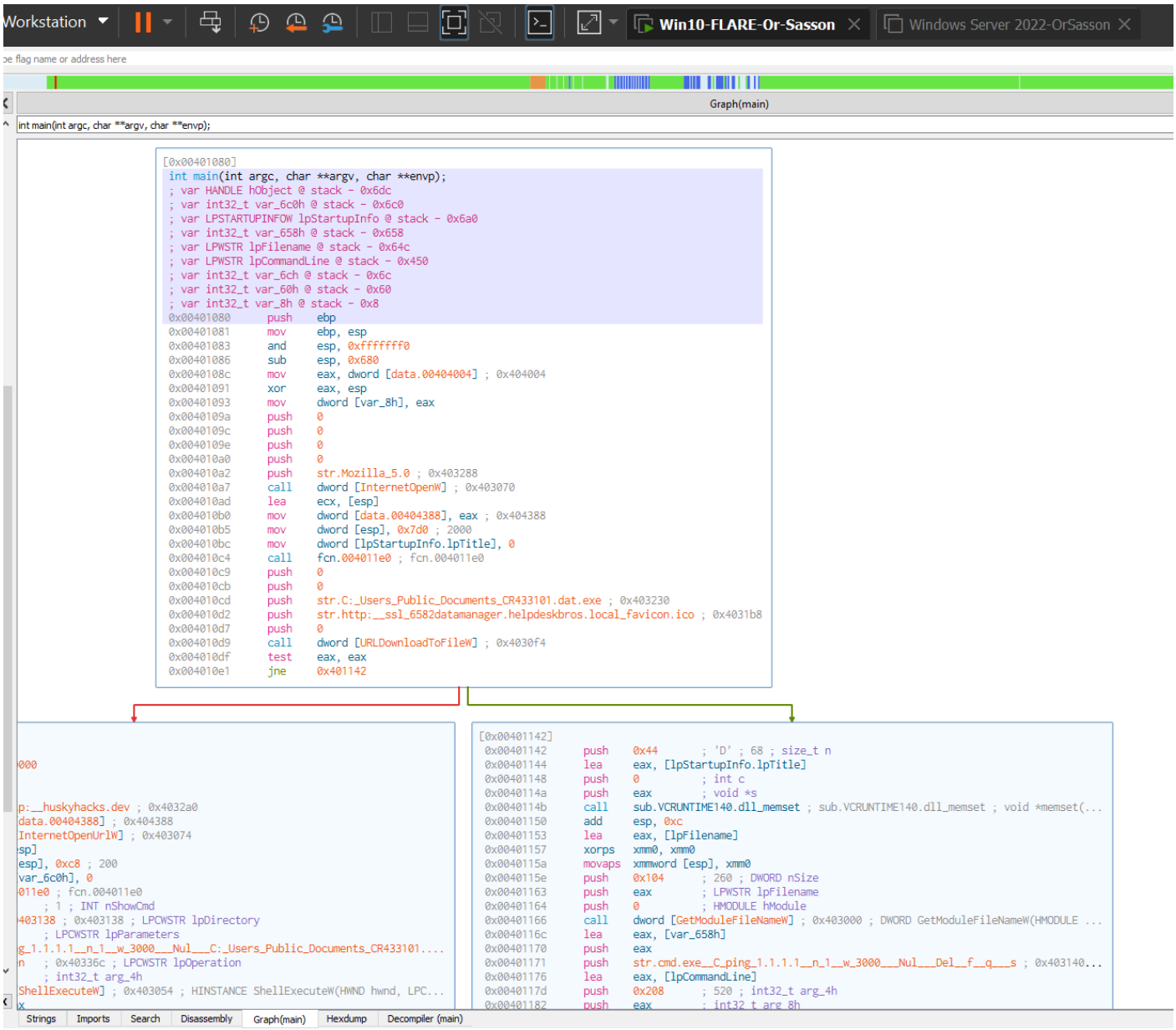
As we can see, the main code got a PUSH operation which led to an exe file which needs to be installed and then direct to a URL which downloading something from the internet which could be an evil indicator as you can view in the screenshot below.

Assessment Task 2: Portfolio of Software Skills



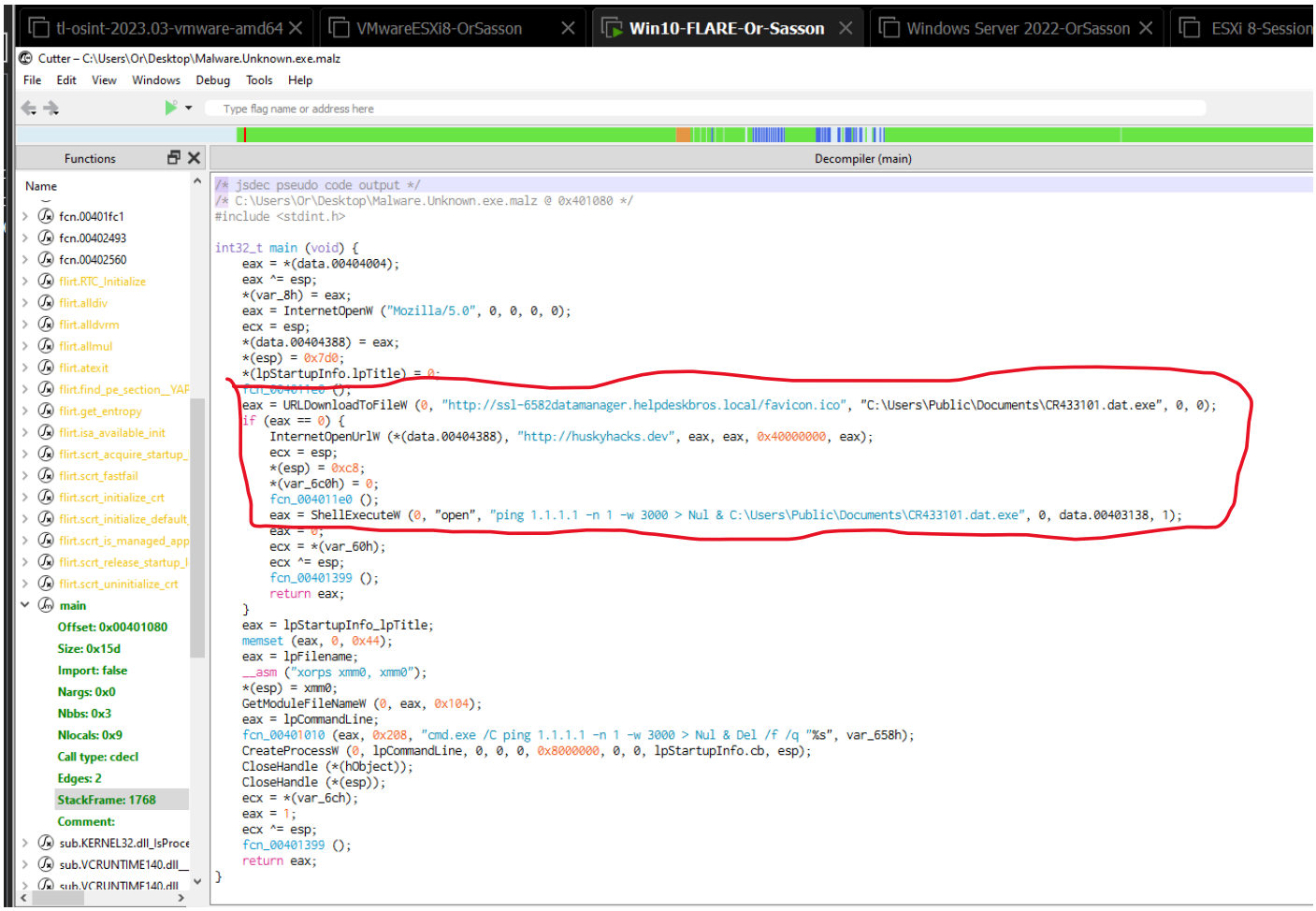
3. Provide graphical representation of main function.

Assessment Task 2: Portfolio of Software Skills



4. Provide screenshot of decompiler and identify the indicator.

Assessment Task 2: Portfolio of Software Skills

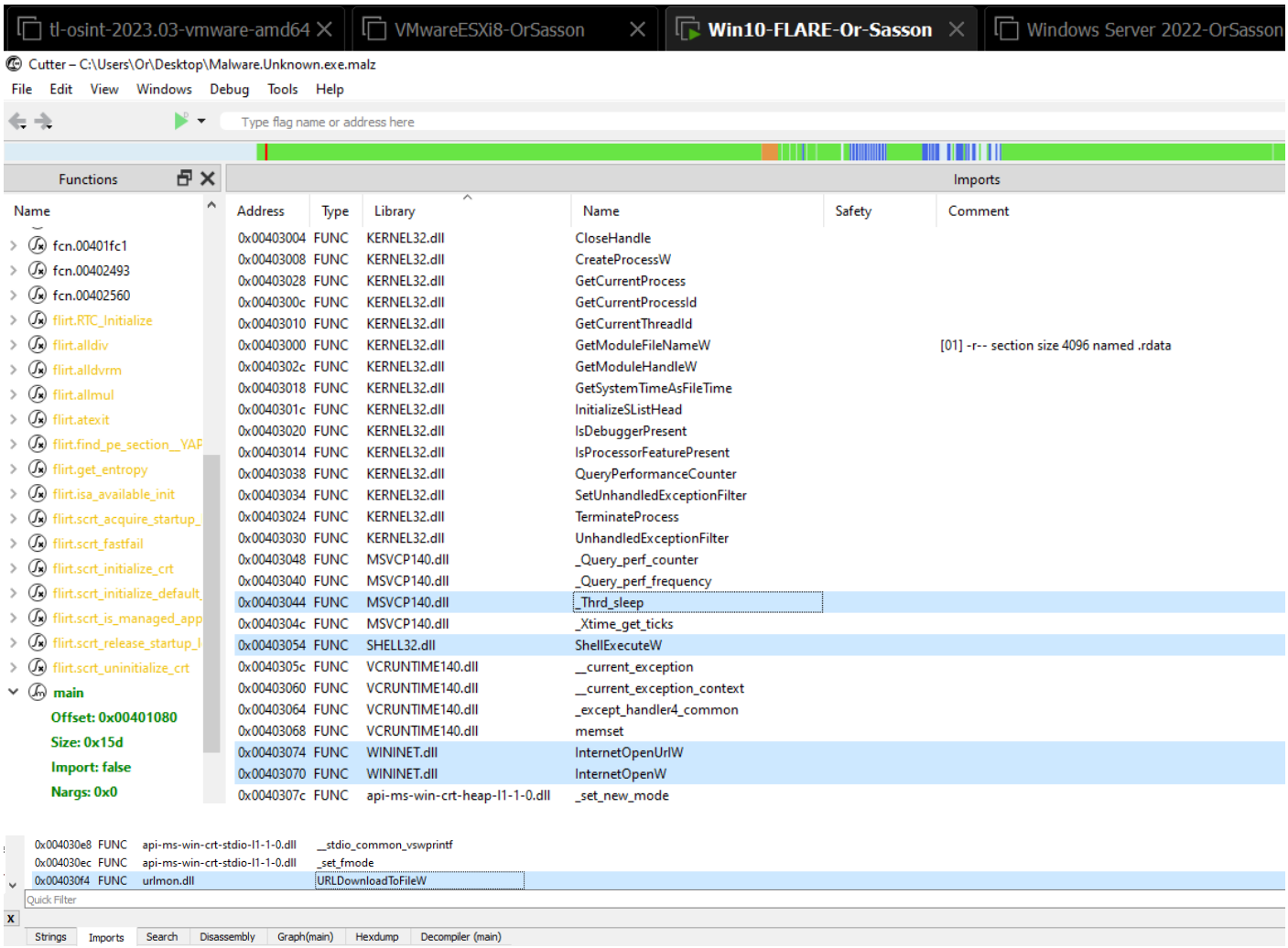


In the screenshot I could see the functions I pointed out in red (the pathway), and I did some research to know what they do.

"URLDownloadToFile" is a function which Downloads a file from the URL and saves them to a file.
ÏnternetOpenUrlW" opens a resource specified by a complete FTP or HTTP URL.
"ShellExecuteW" - I could see a ping command which is sending one ping to a specific IP address and giving some instructions of discarding and deleting a file with Nul & Del.

5. Provide screenshot of Imports, can you identify any interesting imports library function, if yes, provide more details about it.

Assessment Task 2: Portfolio of Software Skills



"Thrd_sleep" which is a function on C++ which delays an execution (might be a malicious execution of a payload)
"ShellExecuteW" operates on a specified URL/file.
ÏnternetOpenUrlW" opens a resource specified by a complete FTP or HTTP URL.
"InternetOpenW" is a function which initializes an application's use of the WinINet.
URLDownloadToFile" is a function which Downloads bits from the Internet and saves them to a file.

Resources for the research:
https://learn.microsoft.com/en-us/windows/win32/api/wininet/nf-wininet-internetopenurlw

Part-D Perform Buffer Overflow exploitation- Justify your answers by providing screenshots

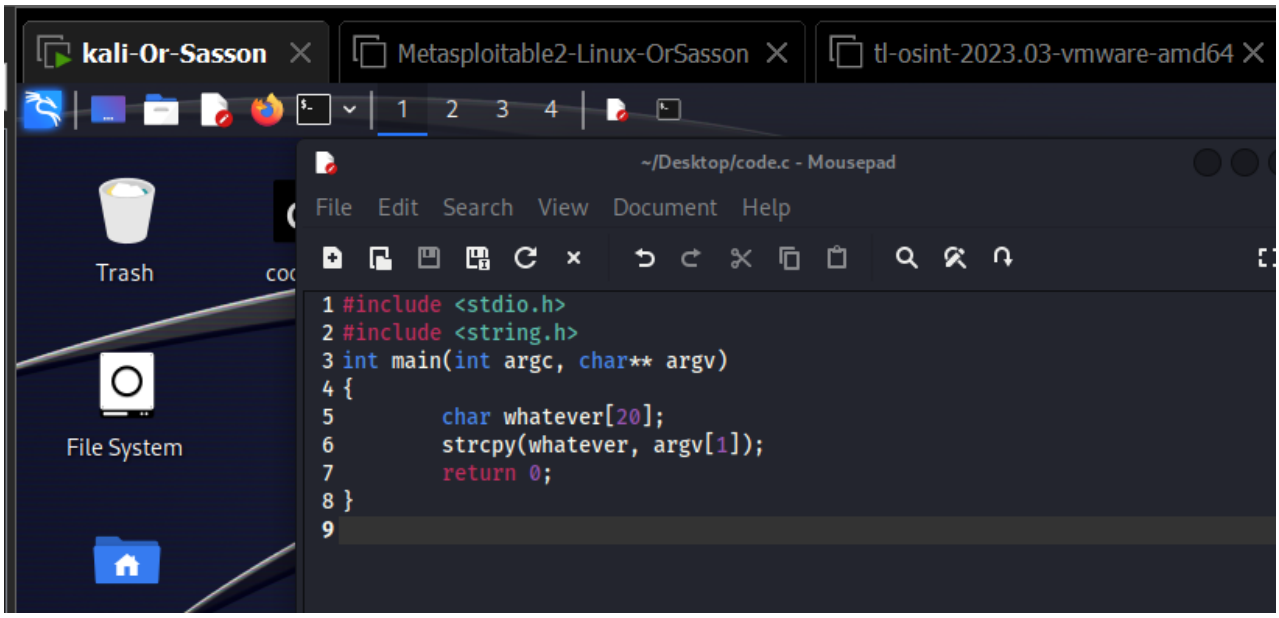
1. Consider the following c source code and save it.

```
#include <stdio.h>
#include <string.h>
int main(int argc, char** argv)
{
        char whatever[20];
        strcpy(whatever, argv[1]);
        return 0;
}
}
```

Assessment Task 2: Portfolio of Software Skills

#provide steps / screenshot

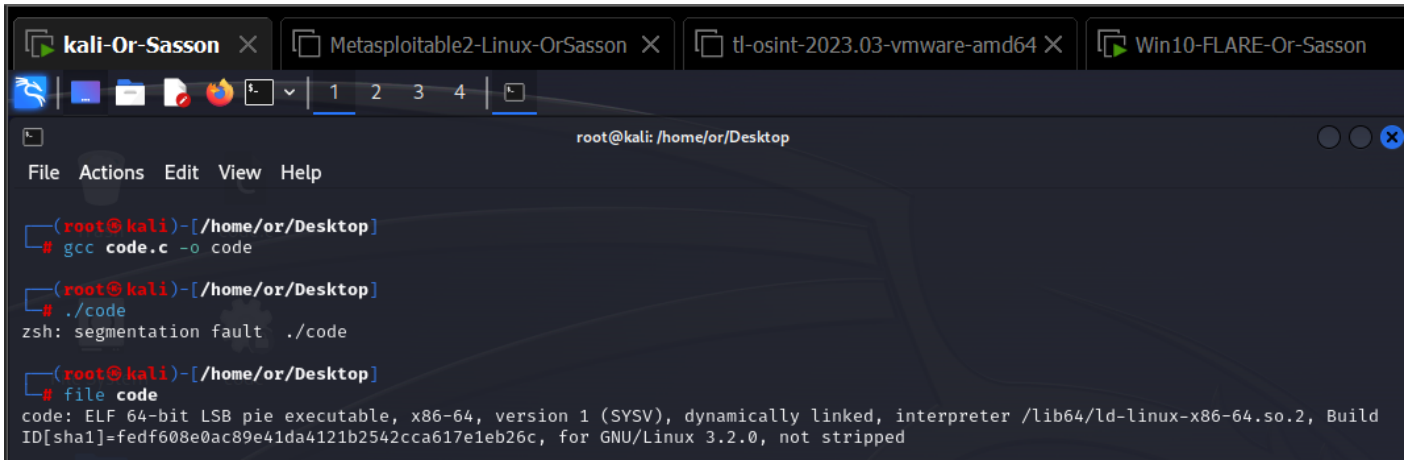
I copied the source code into a text editor on Linux and save it as “code.c”.



2. Compile and run the program with gcc complier, provide screenshot of the compilation process and the output of the program

#provide steps / screenshot

I run the code file and get a fault which indicates about a buffer which been overwritten. The type of the file is ELF based on 64-bit.



3. Examine the Buffer Overflows with gdb.

#provide steps / screenshot

Assessment Task 2: Portfolio of Software Skills

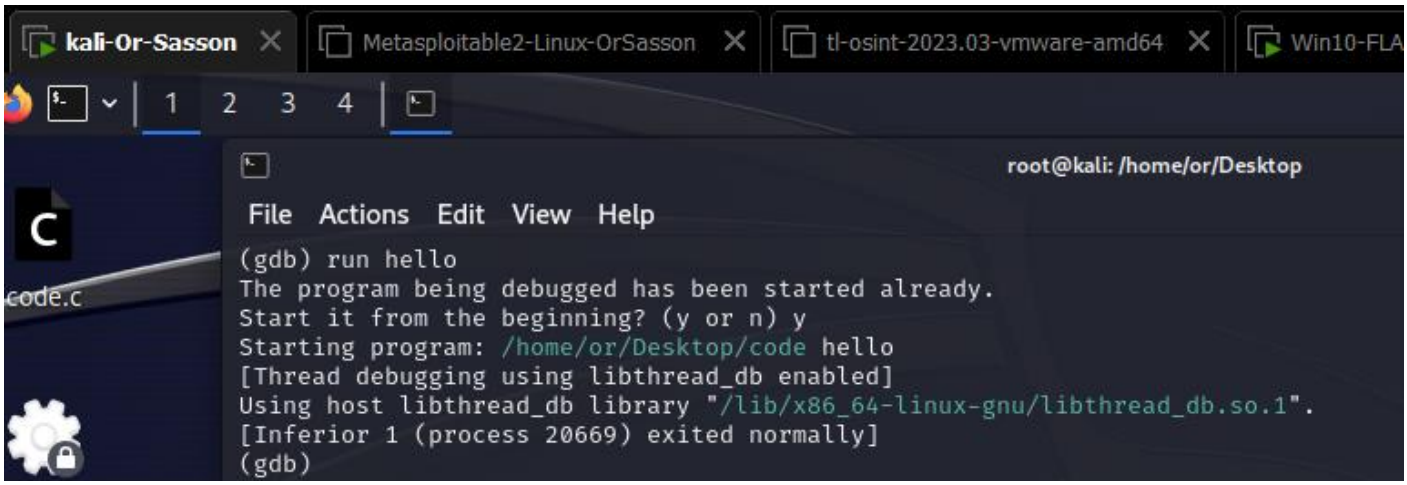
```
kali-Or-Sasson   Metasploitable2-Linux-OrSasson   tl-osint-2023.(
1 2 3 4
root@kali: /home/or/Desktop
File Actions Edit View Help

┌──(root㉿kali)-[/home/or/Desktop]
└─# gdb ./code
GNU gdb (Debian 13.2-1) 13.2
Copyright (C) 2023 Free Software Foundation, Inc.
License GPLv3+: GNU GPL version 3 or later <http://gnu.org/licenses/gpl.html>
This is free software: you are free to change and redistribute it.
There is NO WARRANTY, to the extent permitted by law.
Type "show copying" and "show warranty" for details.
This GDB was configured as "x86_64-linux-gnu".
Type "show configuration" for configuration details.
For bug reporting instructions, please see:
<https://www.gnu.org/software/gdb/bugs/>.
Find the GDB manual and other documentation resources online at:
    <http://www.gnu.org/software/gdb/documentation/>.

For help, type "help".
Type "apropos word" to search for commands related to "word"...
Reading symbols from ./code...
(No debugging symbols found in ./code)
(gdb) info functions
All defined functions:

Non-debugging symbols:
0x0000000000001000  _init
0x0000000000001030  strcpy@plt
0x0000000000001040  __cxa_finalize@plt
0x0000000000001050  _start
0x0000000000001080  deregister_tm_clones
0x00000000000010b0  register_tm_clones
0x00000000000010f0  __do_global_dtors_aux
0x0000000000001130  frame_dummy
0x0000000000001139  main
0x000000000000116c  _fini
```

```
kali-Or-Sasson   Metasploitable2-Linux-OrSasson
1 2 3 4
root@kali: /hor
File Actions Edit View Help
(gdb) disas main
Dump of assembler code for function main:
   0x0000000000001139 <+0>:     push   %rbp
   0x000000000000113a <+1>:     mov    %rsp,%rbp
   0x000000000000113d <+4>:     sub    $0x30,%rsp
   0x0000000000001141 <+8>:     mov    %edi,-0x24(%rbp)
   0x0000000000001144 <+11>:    mov    %rsi,-0x30(%rbp)
   0x0000000000001148 <+15>:    mov    -0x30(%rbp),%rax
   0x000000000000114c <+19>:    add    $0x8,%rax
   0x0000000000001150 <+23>:    mov    (%rax),%rdx
   0x0000000000001153 <+26>:    lea    -0x20(%rbp),%rax
   0x0000000000001157 <+30>:    mov    %rdx,%rsi
   0x000000000000115a <+33>:    mov    %rax,%rdi
   0x000000000000115d <+36>:    call   0x1030 <strcpy@plt>
   0x0000000000001162 <+41>:    mov    $0x0,%eax
   0x0000000000001167 <+46>:    leave
   0x0000000000001168 <+47>:    ret
End of assembler dump.
```

Assessment Task 2: Portfolio of Software Skills

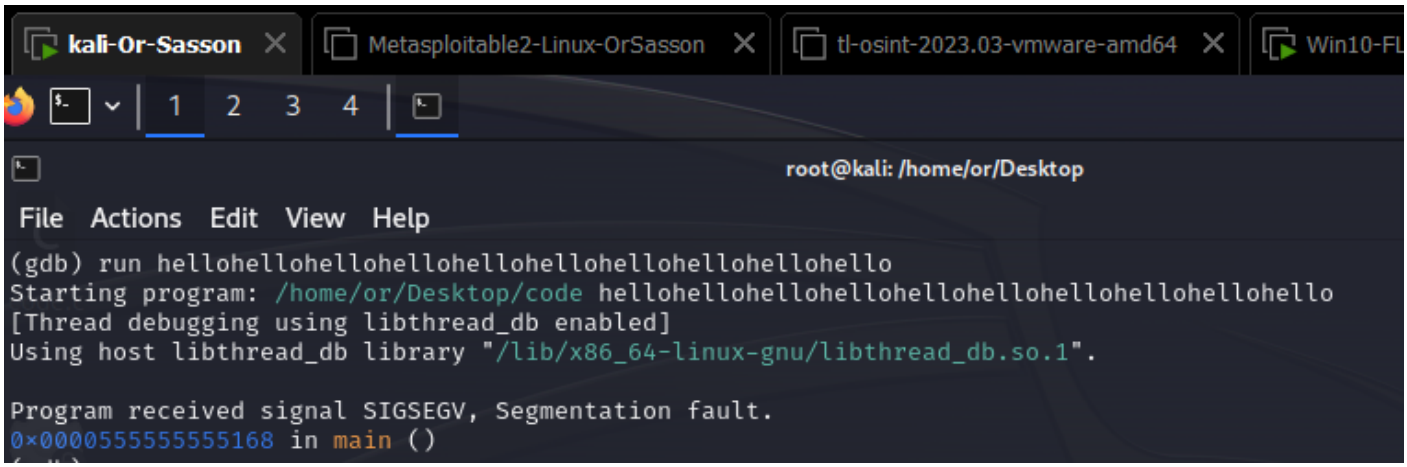
4. In gdb, provide input less than 20 char see what happens, give your comments by providing the screenshot. #provide steps / screenshot

The program processed normally without any fault.



5. Now let's throw an input more than 20 chars, explain your observation with the screenshot #provide steps / screenshot

I received a segmentation fault which indicates that the program is trying to overwrite the buffer which is not allowed.

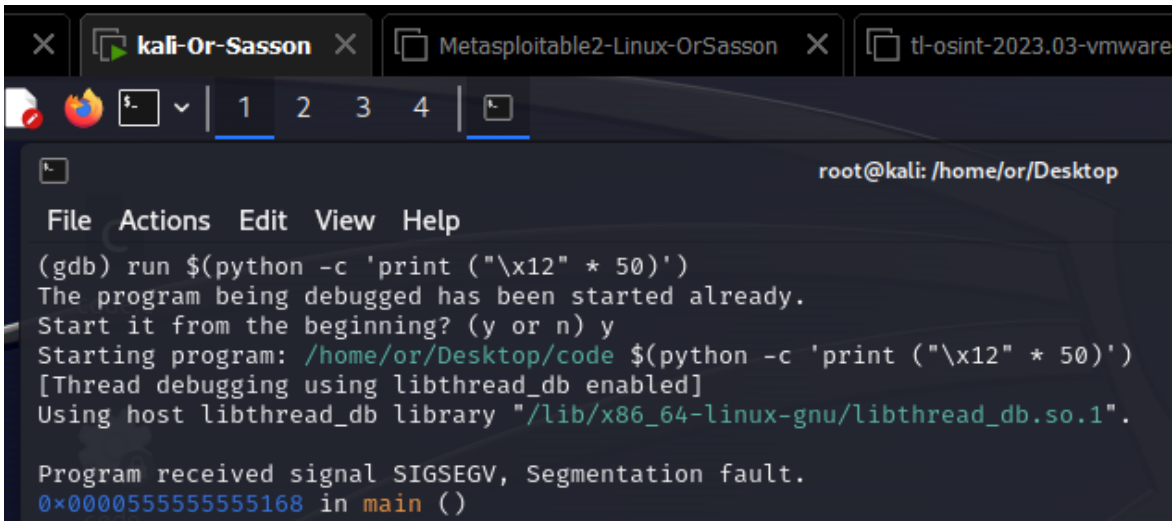


6. For the proof of concept, the python script $(python -c "print('\x12' * 50)") which will overwrites the registers Run this script in gdb and see what happen.

#provide steps / screenshot

It couldn't run this code and I got a segmentation fault with a register address because it tried to overwrite the buffer which isn't allowed.

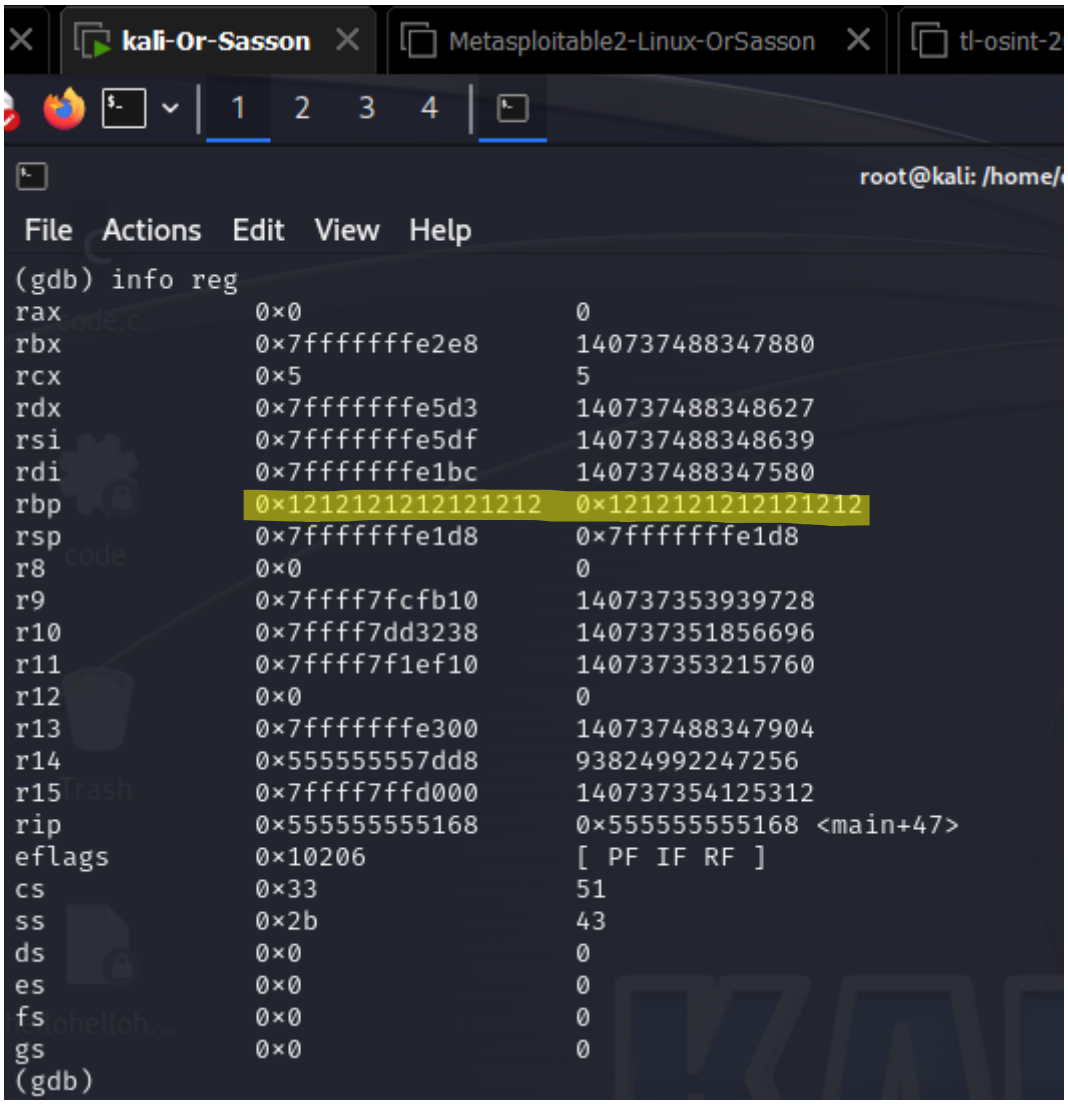
Assessment Task 2: Portfolio of Software Skills



7. In previous step, segmentation fault will occur, when you run the script. Now type command to see the info about registers. Provide the screen shot and highlight the register with address \x12. Give your observation on output #your observation

I could see that the RBP was overwritten with many of the ”/x12” values I inserted with the python script above.

Assessment Task 2: Portfolio of Software Skills



Part-E: Research, install & deploy operating system tools to secure code

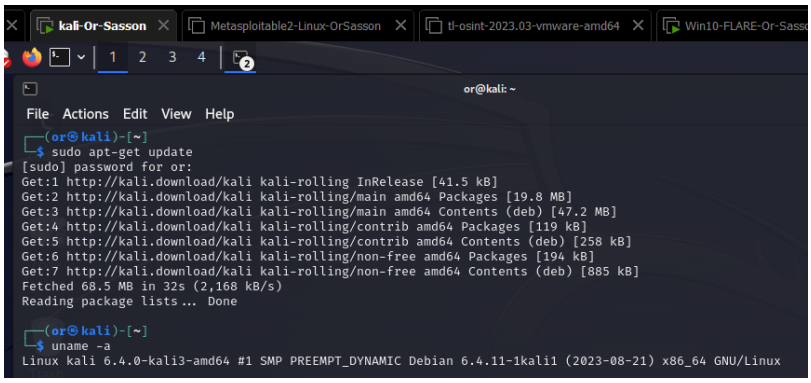
Research antivirus software for your version of Linux.
Install antivirus software in your Linux virtual machine.
Submit a document that describes the software, explains why you chose this one and includes screen captures of the installation and usage of the software.

After my research I choose to install Clamav antivirus which were suitable for my Kali Linux Debian version.
One of the reasons I have chosen ClamAV is because it is open source as Linux and its virus database getting updated regularly by users. Also, it has a huge virus directory, and all its commands can run through the command-line on Linux which gives an advanced control which antivirus with GUI only wouldn't offer. In addition, beginners could use ClamTk which could run with a friendly interface (GUI). Furthermore, ClamAV is owned by cisco, and it is free and considered one of the most popular antiviruses for Linux.

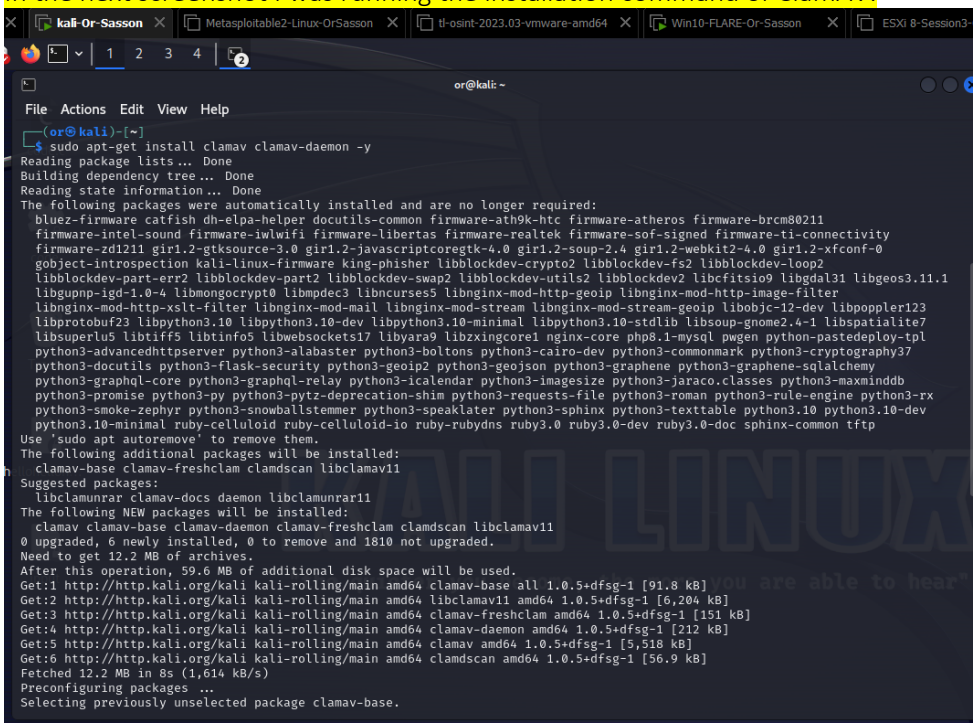
ClamAV can detect and remove many types of malwares, such as viruses, trojans and worms, and got a signature-based detection to identify threats. Also, it has real-time Filesystem detection, hoc file scanning and it can scan mail servers to identify malware. All scans can be scheduled and run regularly.

In the screenshot below I updated my Linux and checked my version to be able to research the right antivirus:

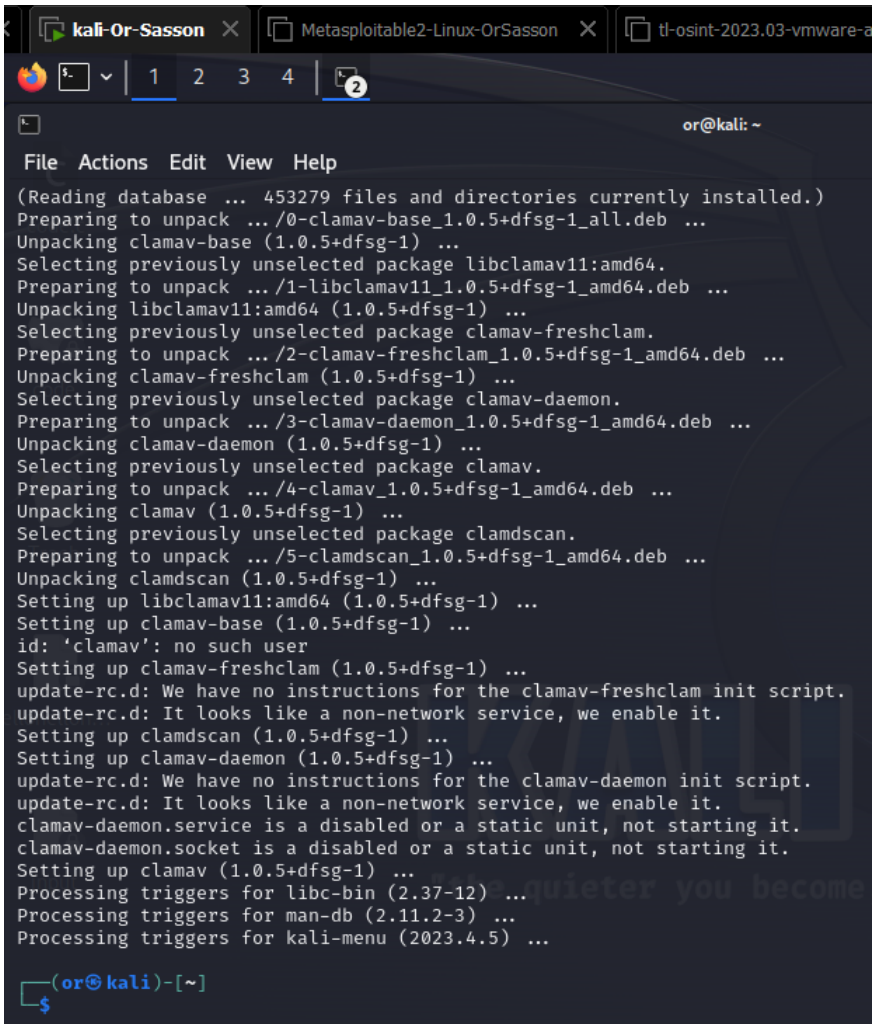
Assessment Task 2: Portfolio of Software Skills



In the next screenshot I was running the installation command of ClamAV:

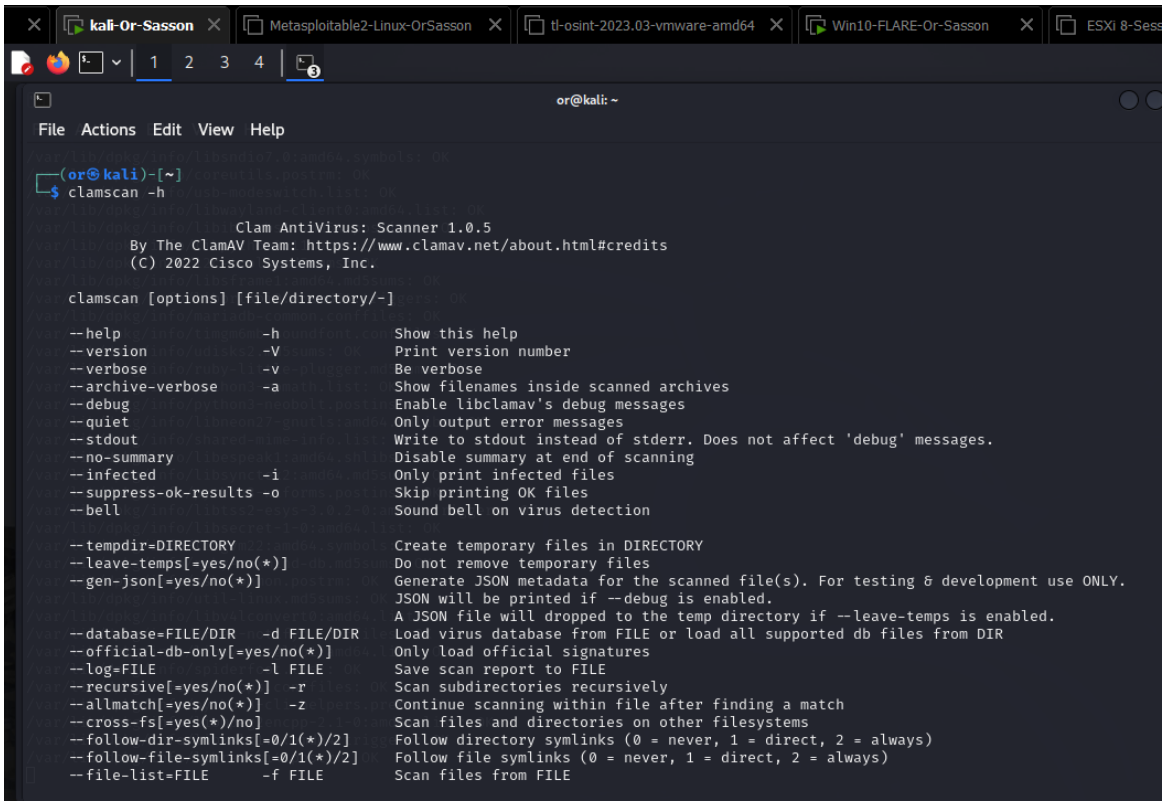


Assessment Task 2: Portfolio of Software Skills

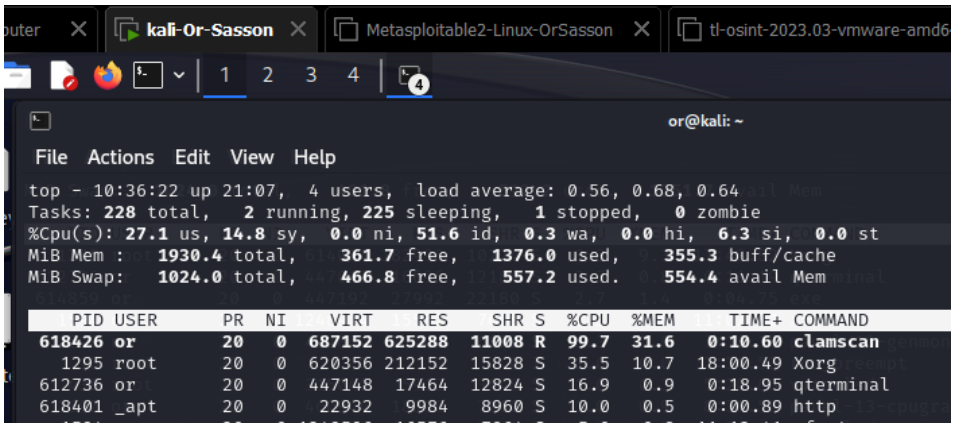


With the clamscan –help I could view the options of the commands I can use plus the version of the software:

Assessment Task 2: Portfolio of Software Skills



In this screenshot I checked the usage of the software while I was running a scan to monitor the CPU usage:



In this screenshot I was running a scan against my “home“ directory:

Assessment Task 2: Portfolio of Software Skills

```
┌──(or㉿kali)-[~]
└─$ clamscan
Loading:    33s, ETA:   0s [=========================>]    8.69M/8.69M sigs
Compiling:  16s, ETA:   0s [=========================>]       41/41 tasks

/home/or/.bash_history: OK
/home/or/scan.xml: OK
/home/or/.xsession-errors: OK
/home/or/.viminfo: OK
/home/or/.wget-hsts: OK
/home/or/.face: OK
/home/or/scan.html: OK
/home/or/.Xauthority: OK
/home/or/.profile: OK
/home/or/.xorgxrdp.10.log: OK
/home/or/.face.icon: Symbolic link
/home/or/pass.txt: OK
/home/or/.ICEauthority: Empty file
/home/or/googlelogo_light_color_272×92dp.png: OK
/home/or/.bashrc: OK
/home/or/user.txt: OK
/home/or/.xsession-errors.old: OK
/home/or/.zshrc: OK
/home/or/.bash_logout: OK
/home/or/chmod 600 sshkey: OK
/home/or/file2.txt: OK
/home/or/.bashrc.original: OK
/home/or/.sudo_as_admin_successful: Empty file
/home/or/crackpass.txt: OK
/home/or/.dmrc: OK
/home/or/file1.txt: OK

---------- SCAN SUMMARY ----------
Known viruses: 8686452
Engine version: 1.0.5
Scanned directories: 1
Scanned files: 23
Infected files: 0
Data scanned: 0.18 MB
Data read: 0.09 MB (ratio 2.05:1)
Time: 51.240 sec (0 m 51 s)
Start Date: 2024:03:09 10:36:11
End Date:   2024:03:09 10:37:02
```

I used the following command to scan the whole system and remove infected files:

```
┌──(or㉿kali)-[~]
└─$ clamscan -r --remove /
Loading:    53s, ETA:   0s [=========================>]    8.69M/8.69M sigs
Compiling:  22s, ETA:   0s [=========================>]       41/41 tasks ks

/tmp/vmware-root_717616-1938583192: Can't open directory.
/tmp/systemd-private-250a519137a7490793c84d9d7cc7bd6c-systemd-logind.service-5CtLpD: Can't open directory.
/tmp/.xfsm-ICE-6Y8CK2: OK
/tmp/vmware-root_438-557881233: Can't open directory.
/tmp/systemd-private-250a519137a7490793c84d9d7cc7bd6c-ModemManager.service-2IU3LR: Can't open directory.
/tmp/systemd-private-250a519137a7490793c84d9d7cc7bd6c-colord.service-2WsnjP: Can't open directory.
/tmp/.X0-lock: OK
/tmp/systemd-private-250a519137a7490793c84d9d7cc7bd6c-upower.service-HQ4EFN: Can't open directory.
/tmp/systemd-private-250a519137a7490793c84d9d7cc7bd6c-haveged.service-MckSkt: Can't open directory.
/tmp/systemd-private-250a519137a7490793c84d9d7cc7bd6c-polkit.service-6lVeKT: Can't open directory.
/0: Empty file
```

I also installed the Clamtk for a friendly GUI:

Assessment Task 2: Portfolio of Software Skills