BSC Cyber Security

F20F0 Coursework One

Fatima Hanifmohammed Patel

H00339652

Link to Demo: https://heriotwatt-my.sharepoint.com/:v:/g/personal/fp17_hw_ac_uk/EbJsfjBg77NDvde8Vwpn8_QBTfm-gqtndzACHcJl2RkEKQ?nav=eyJyZWZlcnJhbEluZm8iOnsicmVmZXJyYWxBcHAiOiJPbmVEcml2ZUZvckJ1c2luZXNzIiwicmVmZXJyYWxBcHBQbGF0Zm9ybSI6IldlYiIsInJlZmVycmFsTW9kZSI6InZpZXciLCJyZWZlcnJhbFZpZXciOiJNeUZpbGVzTGlua0NvcHkifX0&e=jFMVhb

Contents

Abbreviations

Abbreviation	Full-form
PC	Personal Computer
OS	Operating System
DF	Digital Forensics
i.e.	That is
SAM	Security Accounts Manager
etc	Et cetera
GUI	Graphical User Interface

1 Introduction

Twenty years ago, digital evidence played a minor role in criminal investigations. However, a recent study by IBM reveals that digital evidence now plays a crucial role in most legal cases, appearing in over 90% of crimes committed in the present day [1]. With the widespread adoption of technology, offenses like homicides, kidnappings, and domestic violence frequently involve digital evidence. Proper collection and analysis of this digital evidence can significantly enhance the effectiveness and success of both investigations and legal proceedings [2].

This report outlines the steps involved in establishing a prototype for a virtual digital forensics' laboratory on VirtualBox.

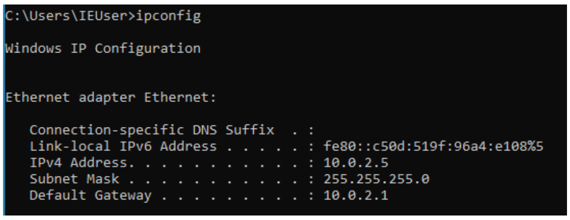
2 Setup

Digital forensics involves working with various malwares and viruses and therefore using virtual machines allows constraining security breaches from affecting the host machine. One can run various operating systems on one host machine using Virtual Box. This allows cross platform analysis of tools and malware. Malicious actors also use various operating systems so a good forensic analyst should be familiar with various systems.

Two virtual machines, Windows 10 and Ubuntu were installed and configured on VirtualBox to be part of the same NAT Network. Ubuntu and Windows are popular stable open-source OS. Having both OS means that all well-known digital forensic tools can be used.

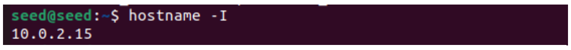
The IP addresses of the VM's are:

Win10 IP	*10.0.2.5*
Ubuntu IP	*10.0.2.15*



```
C:\Users\IEUser>ipconfig

Windows IP Configuration


Ethernet adapter Ethernet:

   Connection-specific DNS Suffix  . :
   Link-local IPv6 Address . . . . . : fe80::c50d:519f:96a4:e108%5
   IPv4 Address. . . . . . . . . . . : 10.0.2.5
   Subnet Mask . . . . . . . . . . . : 255.255.255.0
   Default Gateway . . . . . . . . . : 10.0.2.1
```

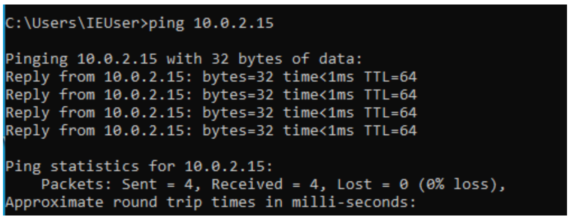
Figure 1: Screenshot showing Win10 IP



```
seed@seed:~$ hostname -I
10.0.2.15
```

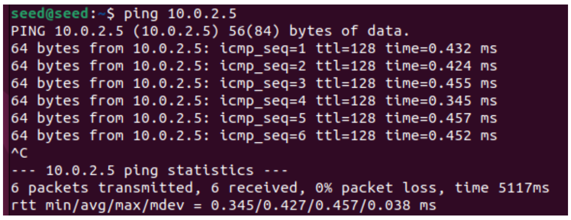
Figure 2: Screenshot showing Ubuntu IP

Since they are on the same network, they can directly communicate with eachother. This can be demonstrated by pinging:



```
C:\Users\IEUser>ping 10.0.2.15

Pinging 10.0.2.15 with 32 bytes of data:
Reply from 10.0.2.15: bytes=32 time<1ms TTL=64
Reply from 10.0.2.15: bytes=32 time<1ms TTL=64
Reply from 10.0.2.15: bytes=32 time<1ms TTL=64
Reply from 10.0.2.15: bytes=32 time<1ms TTL=64

Ping statistics for 10.0.2.15:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
```

Figure 3: Pinging Win10 vm from Ubuntu vm



```
seed@seed:~$ ping 10.0.2.5
PING 10.0.2.5 (10.0.2.5) 56(84) bytes of data.
64 bytes from 10.0.2.5: icmp_seq=1 ttl=128 time=0.432 ms
64 bytes from 10.0.2.5: icmp_seq=2 ttl=128 time=0.424 ms
64 bytes from 10.0.2.5: icmp_seq=3 ttl=128 time=0.455 ms
64 bytes from 10.0.2.5: icmp_seq=4 ttl=128 time=0.345 ms
64 bytes from 10.0.2.5: icmp_seq=5 ttl=128 time=0.457 ms
64 bytes from 10.0.2.5: icmp_seq=6 ttl=128 time=0.452 ms
^C
--- 10.0.2.5 ping statistics ---
6 packets transmitted, 6 received, 0% packet loss, time 5117ms
rtt min/avg/max/mdev = 0.345/0.427/0.457/0.038 ms
```

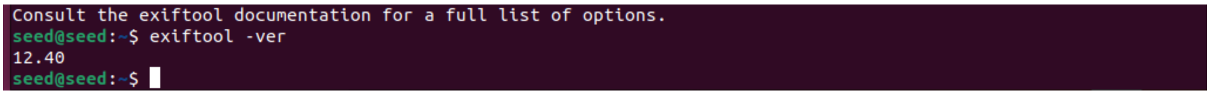
Figure 4: Pinging Ubuntu vm from Win10

3 Digital Forensics Tools

Tools are essential in digital forensics since they speed up and automate the investigation process which would otherwise be extremely tedious for humans to manually do. Tools can allow us to investigate evidence without tampering, reduce the risk of human error and easily detect patterns [16]. It is important for forensic analysts to be familiar with the functionality and drawbacks of using various tools to be able to efficiently analyze and withdraw evidence. Digital forensics involves using a range of tools with various uses from metadata analysis, password cracking, network analysis and data recovery [6][8][9][10]. This report covers a few tools that can used to carry these tasks out.

3.1 Tools installed on Linux Virtual Machine

3.1.1 ExifTool: Meta Data Analysis



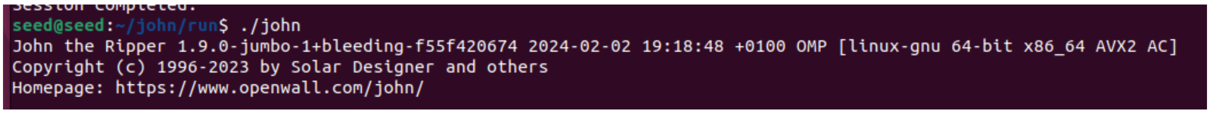
```
Consult the exiftool documentation for a full list of options.
seed@seed:~$ exiftool -ver
12.40
seed@seed:~$ 
```

Figure 5: Screenshot showing Exif installation

Reading metadata is essential in digital forensics to gain more insights on collected data. ExifTool is a command-line, platform-independent, free and open-source software program for reading, writing, and manipulating image, audio, video, and PDF metadata. It supports a vast array of file formats including EXIF, IPTC, XMP, etc [8].

ExifTool allows date and time stamp manipulation, adding or modifying GPS data and copying metadata between files. It also supports batch processing for large volumes of files. It can be used to extract thumbnail images from raw files, rename files based on metadata, and delete unwanted metadata to clean up files [8].

3.1.2 John the Ripper Jumbo: Password Cracker



```
Session completed.
seed@seed:~/john/run$ ./john
John the Ripper 1.9.0-jumbo-1+bleeding-f55f420674 2024-02-02 19:18:48 +0100 OMP [linux-gnu 64-bit x86_64 AVX2 AC]
Copyright (c) 1996-2023 by Solar Designer and others
Homepage: https://www.openwall.com/john/
```

Figure 6: Screenshot showing John the Ripper Installation

Cracking hashed or encrypted passwords is essential in digital forensics to access digital evidence. John the Ripper is a cross-platform, command-line tool for cracking hashed or encrypted passwords [6]. It was installed using sudo, but for advanced capabilities, **John The Ripper Jumbo** was set up by cloning their GitHub repository. **John the Ripper Jumbo** supports hundreds of hash and cipher types, including user passwords of Unix flavors, macOS, Windows, etc [7]. The main drawback of any password cracking tool is that there is the possibility that a password is not found if it is not in the specified wordlist or if the password is very long and random then bruteforce could take years. John the Ripper works much faster with higher-spec PCs.

3.2 Tools installed on Windows Virtual Machine

3.2.1 Autopsy: Digital Investigations

Autopsy is an open-source, cross-platform digital forensics platform and graphical interface to **The Sleuth Kit**. It is capable of analyzing hard drives, smartphones, and media cards, conducting data carving, keyword searches, and timeline analysis [9]. It has a user-friendly GUI and comprehensive case management. Its main drawback is its performance with very large datasets is slow.

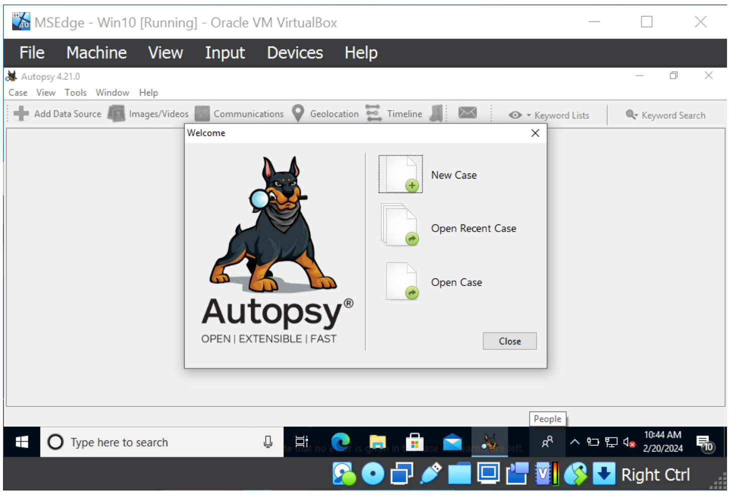
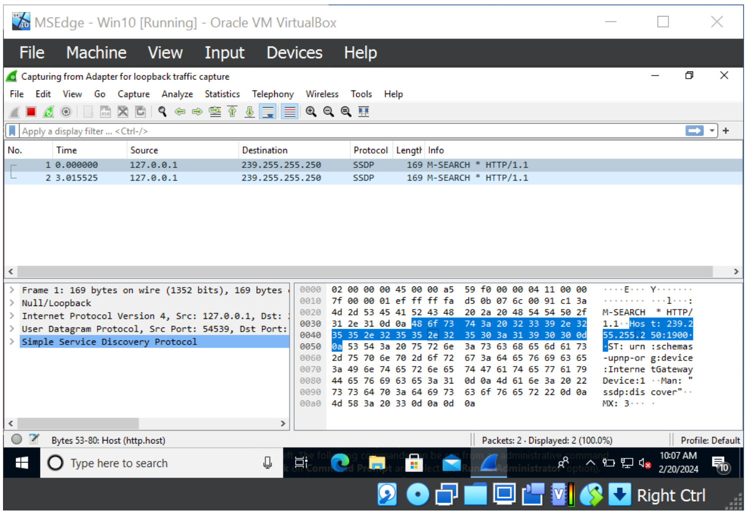


Figure 7: Autopsy GUI

3.2.2 Wireshark: Network Protocol Analyzer

Wireshark is an open-source, cross-platform network protocol analyzer widely used in digital forensics to capture and interactively browse the traffic running on a computer network [10]. It has a user-friendly GUI and powerful filtering capabilities. It could be difficult for new users since it requires understanding network protocols deeply to effectively analyze data.


MSEdge - Win10 [Running] - Oracle VM VirtualBox
File
Machine
View
Input
Devices
Help
Capturing from Adapter for loopback traffic capture
File
Edit
View
Go
Capture
Analyze
Statistics
Telephony
Wireless
Tools
Help
Apply a display filter ... <Ctrl-/>

```
No.   Time       Source      Destination       Protocol  Length Info
    1 0.000000   127.0.0.1   239.255.255.250   SSDP       169 M-SEARCH * HTTP/1.1
    2 3.015525   127.0.0.1   239.255.255.250   SSDP       169 M-SEARCH * HTTP/1.1
```

```
> Frame 1: 169 bytes on wire (1352 bits), 169 bytes
> Null/Loopback
> Internet Protocol Version 4, Src: 127.0.0.1, Dst:
> User Datagram Protocol, Src Port: 54539, Dst Port:
> Simple Service Discovery Protocol
```

```
0000  02 00 00 00 45 00 00 a5  59 f0 00 00 04 11 00 00   ····E··· Y·······
0010  7f 00 00 01 ef ff ff fa  d5 0b 07 6c 00 91 c1 3a   ········ ···l···:
0020  4d 2d 53 45 41 52 43 48  20 2a 20 48 54 54 50 2f   M-SEARCH  * HTTP/
0030  31 2e 31 0d 0a 48 6f 73  74 3a 20 32 33 39 2e 32   1.1··Hos t: 239.2
0040  35 35 2e 32 35 35 2e 32  35 30 3a 31 39 30 30 0d   55.255.2 50:1900·
0050  0a 53 54 3a 20 75 72 6e  3a 73 63 68 65 6d 61 73   ·ST: urn :schemas
0060  2d 75 70 6e 70 2d 6f 72  67 3a 64 65 76 69 63 65   -upnp-or g:device
0070  3a 49 6e 74 65 72 6e 65  74 47 61 74 65 77 61 79   :Interne tGateway
0080  44 65 76 69 63 65 3a 31  0d 0a 4d 61 6e 3a 20 22   Device:1 ··Man: "
0090  73 73 64 70 3a 64 69 73  63 6f 76 65 72 22 0d 0a   ssdp:dis cover"··
00a0  4d 58 3a 20 33 0d 0a 0d  0a                        MX: 3··· ·
```

Bytes 53-80: Host (http.host)
Packets: 2 · Displayed: 2 (100.0%)
Profile: Default
Type here to search
10:07 AM
2/20/2024
Right Ctrl

Figure 8: Wireshark GUI

4 Use Cases

The two digital forensic tools chosen to demonstrate use cases are:

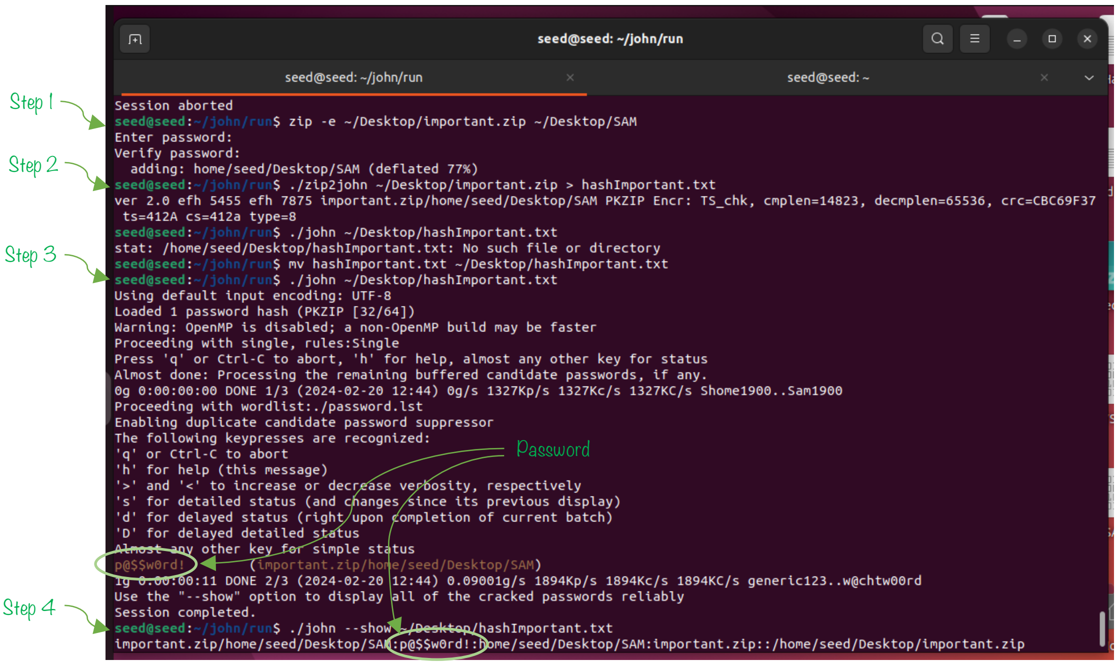
1. John the Ripper
2. Wireshark

Note: Screenshots are numbered with steps which are referenced to in the explanation.

4.1 Case One: Cracking Encrypted ZIP File's Password using John the Ripper

Cracking encrypted files is important in digital forensics if the evidence has been encrypted by the target. To demonstrate cracking an encrypted ZIP file, *important.zip* was created with a simple password: "p@$$w0rd!" (*step 1*).

To extract the password of an encrypted ZIP file using John the Ripper, dump the hash of the file using **zip2john** (a tool available in John the Ripper Jumbo) [4] into a text file called *hashImportant.txt* (*step 2*). Then attempt to crack the hash using John (*step 3*). You can specify that it is a ZIP file using the flag, `--format=zip` but John automatically does this. To view the password use the flag `--show` (*step 4*).



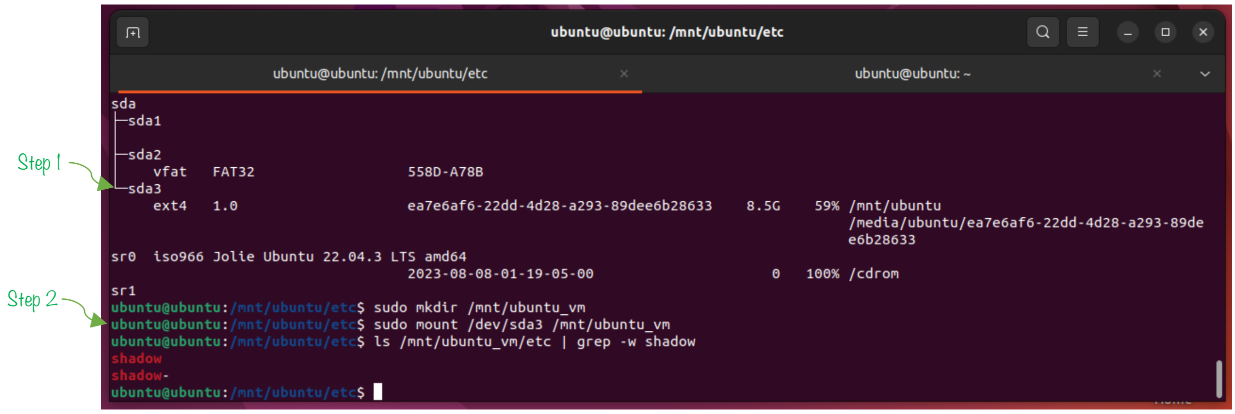

```
Session aborted
seed@seed:~/john/run$ zip -e ~/Desktop/important.zip ~/Desktop/SAM
Enter password:
Verify password:
  adding: home/seed/Desktop/SAM (deflated 77%)
seed@seed:~/john/run$ ./zip2john ~/Desktop/important.zip > hashImportant.txt
ver 2.0 efh 5455 efh 7875 important.zip/home/seed/Desktop/SAM PKZIP Encr: TS_chk, cmplen=14823, decmplen=65536, crc=CBC69F37
 ts=412A cs=412a type=8
seed@seed:~/john/run$ ./john ~/Desktop/hashImportant.txt
stat: /home/seed/Desktop/hashImportant.txt: No such file or directory
seed@seed:~/john/run$ mv hashImportant.txt ~/Desktop/hashImportant.txt
seed@seed:~/john/run$ ./john ~/Desktop/hashImportant.txt
Using default input encoding: UTF-8
Loaded 1 password hash (PKZIP [32/64])
Warning: OpenMP is disabled; a non-OpenMP build may be faster
Proceeding with single, rules:Single
Press 'q' or Ctrl-C to abort, 'h' for help, almost any other key for status
Almost done: Processing the remaining buffered candidate passwords, if any.
0g 0:00:00:00 DONE 1/3 (2024-02-20 12:44) 0g/s 1327Kp/s 1327Kc/s 1327KC/s Shome1900..Sam1900
Proceeding with wordlist:./password.lst
Enabling duplicate candidate password suppressor
The following keypresses are recognized:
'q' or Ctrl-C to abort
'h' for help (this message)
'>' and '<' to increase or decrease verbosity, respectively
's' for detailed status (and changes since its previous display)
'd' for delayed status (right upon completion of current batch)
'D' for delayed detailed status
Almost any other key for simple status
p@$$w0rd!        (important.zip/home/seed/Desktop/SAM)
1g 0:00:00:11 DONE 2/3 (2024-02-20 12:44) 0.09001g/s 1894Kp/s 1894Kc/s 1894KC/s generic123..w@chtw00rd
Use the "--show" option to display all of the cracked passwords reliably
Session completed.
seed@seed:~/john/run$ ./john --show ~/Desktop/hashImportant.txt
important.zip/home/seed/Desktop/SAM:p@$$w0rd!:home/seed/Desktop/SAM:important.zip::/home/seed/Desktop/important.zip
```

Figure 9: John the Ripper Jumbo being used to Crack an Encrypted Zip File

4.2 Case Two: Cracking Linux Password using John the Ripper

In digital forensics, being able to access the contents of a PC is essential to collecting evidence. Every OS has different mechanisms for storing usernames and passwords. To find the account names and passwords of a Linux OS, you need to access the *shadow* file which is limited to root access (which we don't have the password for).

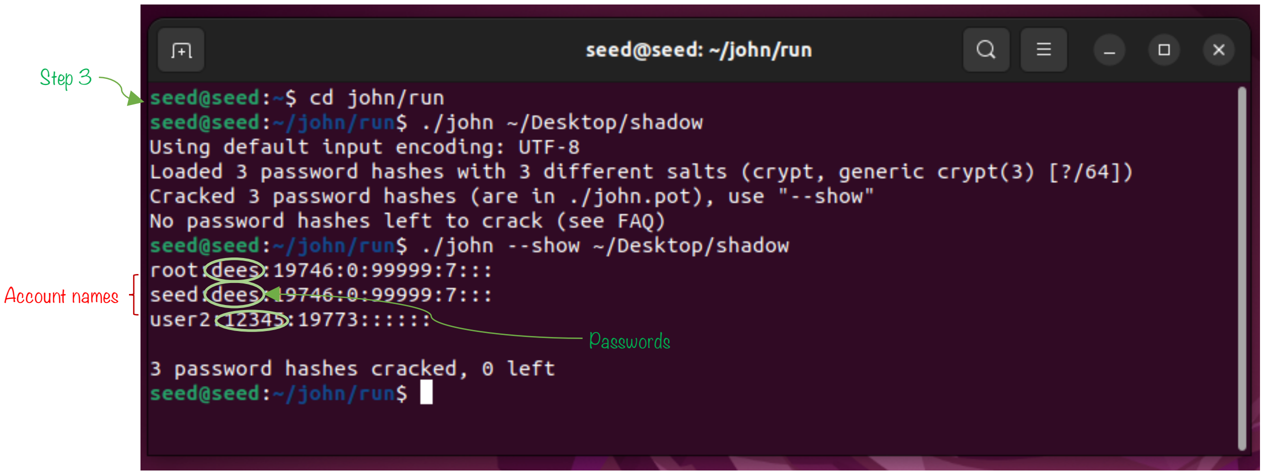
Therefore, to prevent the OS from blocking access, you can boot from another operating system on the same PC. To do this, add an ubuntu iso as a CD and change the boot order to boot from the CD before the hard drive. Then locate the target OS on the hard drive (step 1) and create a mount point to it (step 2) as shown in the screenshot below. The shadow file of the target OS is now accessible without root.


ubuntu@ubuntu: /mnt/ubuntu/etc
ubuntu@ubuntu: /mnt/ubuntu/etc
ubuntu@ubuntu: ~
Step 1
Step 2

```
sda
├─sda1
│
├─sda2
│   vfat   FAT32                   558D-A78B
└─sda3
    ext4   1.0                     ea7e6af6-22dd-4d28-a293-89dee6b28633    8.5G    59% /mnt/ubuntu
                                                                                    /media/ubuntu/ea7e6af6-22dd-4d28-a293-89de
                                                                                    e6b28633
sr0  iso966 Jolie Ubuntu 22.04.3 LTS amd64
                                   2023-08-08-01-19-05-00                     0   100% /cdrom
sr1
ubuntu@ubuntu:/mnt/ubuntu/etc$ sudo mkdir /mnt/ubuntu_vm
ubuntu@ubuntu:/mnt/ubuntu/etc$ sudo mount /dev/sda3 /mnt/ubuntu_vm
ubuntu@ubuntu:/mnt/ubuntu/etc$ ls /mnt/ubuntu_vm/etc | grep -w shadow
shadow
shadow-
ubuntu@ubuntu:/mnt/ubuntu/etc$ 
```

Figure 10: Finding out which Partition the Target OS Resides in and Making a Mount Point to it

Now run John on this *shadow* file (step 3) and the account names and passwords have been cracked as seen below:



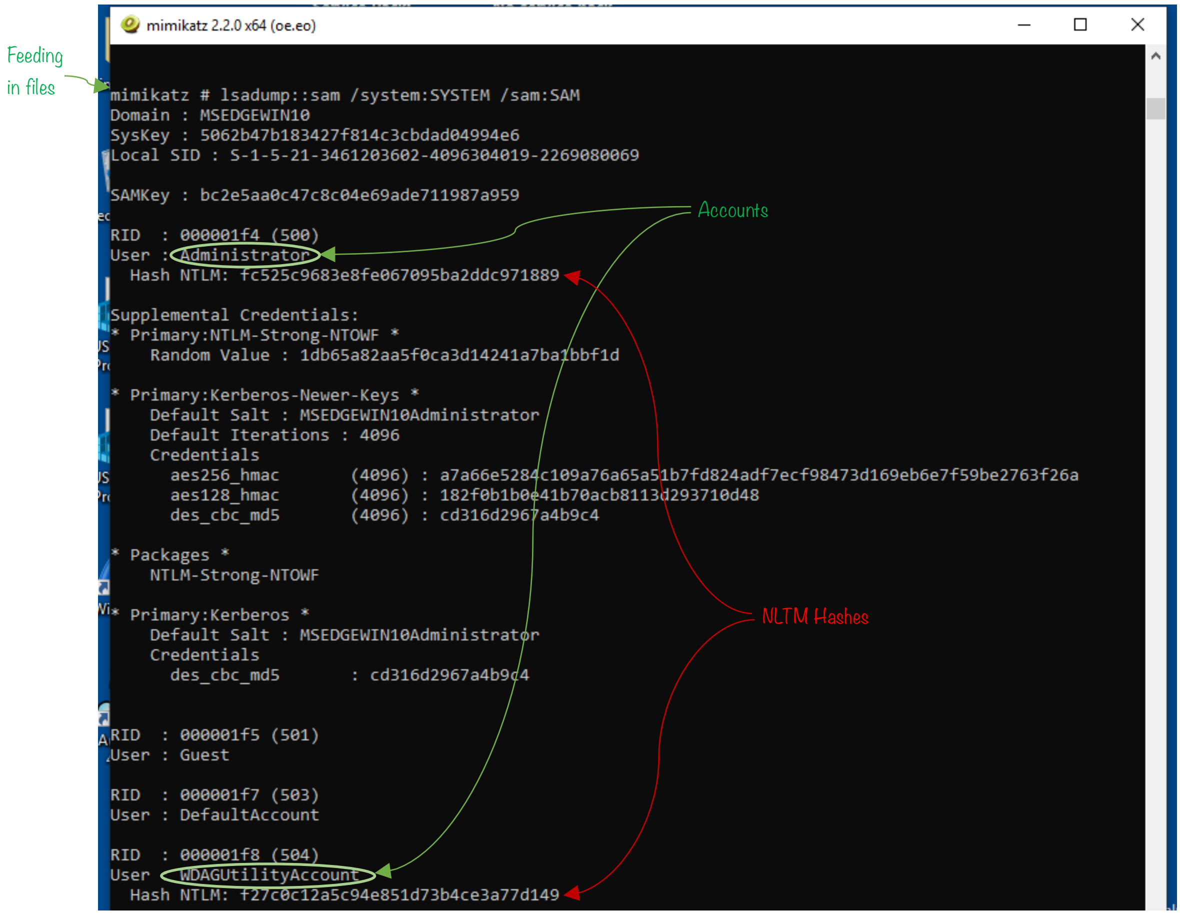
```
seed@seed:~$ cd john/run
seed@seed:~/john/run$ ./john ~/Desktop/shadow
Using default input encoding: UTF-8
Loaded 3 password hashes with 3 different salts (crypt, generic crypt(3) [?/64])
Cracked 3 password hashes (are in ./john.pot), use "--show"
No password hashes left to crack (see FAQ)
seed@seed:~/john/run$ ./john --show ~/Desktop/shadow
root:dees:19746:0:99999:7:::
seed:dees:19746:0:99999:7:::
user2:12345:19773::::::

3 password hashes cracked, 0 left
seed@seed:~/john/run$ 
```

Figure 11: Cracking the Shadow file of the Target using John

4.3 Case Three: Cracking Windows Password using John the Ripper

Windows utilizes a unique mechanism for storing user credentials. Usernames and associated password hashes are stored in a *SAM* file, which is locked while the OS is running. The *SAM* file alone does not contain all the information needed to decrypt these hashes, the *SYSTEM* file, which contains system-specific settings and configurations, includes the encryption keys necessary to decrypt the hashes stored in the *SAM* file.

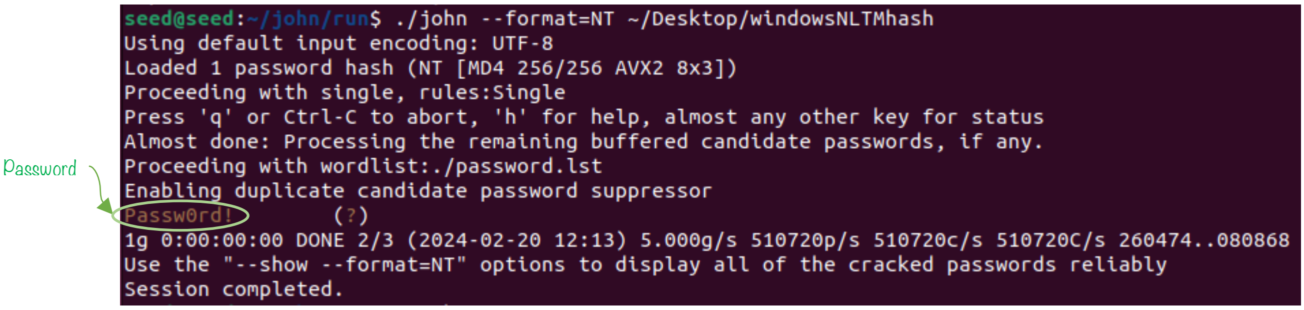
To access these files, follow a similar approach to the one used for the Linux OS, i.e., boot the system using another OS and mount the Windows filesystem. This allows you to copy the *SAM* and *SYSTEM* files to an external medium. These files are binary encoded and to convert them into a format that John the Ripper can attempt to crack, use **Mimikatz**, an open-source utility designed for Windows systems. **Mimikatz** is capable of extracting a variety of credential types, including the NTLM hashes required for password cracking by John the Ripper [11]. The screenshot below shows the NLTM hash dump by Mimikatz:



```
mimikatz # lsadump::sam /system:SYSTEM /sam:SAM
Domain : MSEDGEWIN10
SysKey : 5062b47b183427f814c3cbdad04994e6
Local SID : S-1-5-21-3461203602-4096304019-2269080069

SAMKey : bc2e5aa0c47c8c04e69ade711987a959

RID  : 000001f4 (500)
User : Administrator
  Hash NTLM: fc525c9683e8fe067095ba2ddc971889

Supplemental Credentials:
* Primary:NTLM-Strong-NTOWF *
    Random Value : 1db65a82aa5f0ca3d14241a7ba1bbf1d

* Primary:Kerberos-Newer-Keys *
    Default Salt : MSEDGEWIN10Administrator
    Default Iterations : 4096
    Credentials
      aes256_hmac       (4096) : a7a66e5284c109a76a65a51b7fd824adf7ecf98473d169eb6e7f59be2763f26a
      aes128_hmac       (4096) : 182f0b1b0e41b70acb8113d293710d48
      des_cbc_md5       (4096) : cd316d2967a4b9c4

* Packages *
    NTLM-Strong-NTOWF

* Primary:Kerberos *
    Default Salt : MSEDGEWIN10Administrator
    Credentials
      des_cbc_md5       : cd316d2967a4b9c4


RID  : 000001f5 (501)
User : Guest

RID  : 000001f7 (503)
User : DefaultAccount

RID  : 000001f8 (504)
User : WDAGUtilityAccount
  Hash NTLM: f27c0c12a5c94e851d73b4ce3a77d149
```

Figure 12: Extracting NTLM Hashes using Mimikatz

Now that the NLTM hashes have been extracted, copy them into a txt file and feed it into John the Ripper. To specify the hash format as NLTM, use the flag: `--format=NT`. As seen in the screenshot on the next page, the password has been cracked.

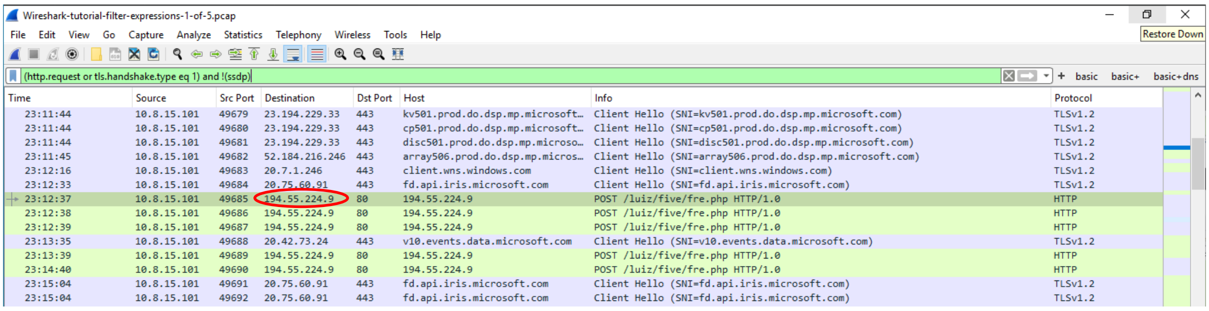


```
seed@seed:~/john/run$ ./john --format=NT ~/Desktop/windowsNLTMhash
Using default input encoding: UTF-8
Loaded 1 password hash (NT [MD4 256/256 AVX2 8x3])
Proceeding with single, rules:Single
Press 'q' or Ctrl-C to abort, 'h' for help, almost any other key for status
Almost done: Processing the remaining buffered candidate passwords, if any.
Proceeding with wordlist:./password.lst
Enabling duplicate candidate password suppressor
Passw0rd!        (?)
1g 0:00:00:00 DONE 2/3 (2024-02-20 12:13) 5.000g/s 510720p/s 510720c/s 510720C/s 260474..080868
Use the "--show --format=NT" options to display all of the cracked passwords reliably
Session completed.
```

Figure 13: John the Ripper Extracting Password from NLTM Hash

4.4 Case Four: Detecting Loki Bot Malware using Wireshark

In this scenario, there is a suspected security breach since they is an abnormal outflow of data on the company network. There is a network capture (pcap file) [13] of the incident which can be investigated using Wireshark. Upon inspecting the traffic filtered to show HTTP and TLS handshake protocols, they are multiple HTTP POST requests to a particular IP address, **194.55.224.9** which has been reported to Threatfox in 2023 [14]. This IP address has been associated with Loki.

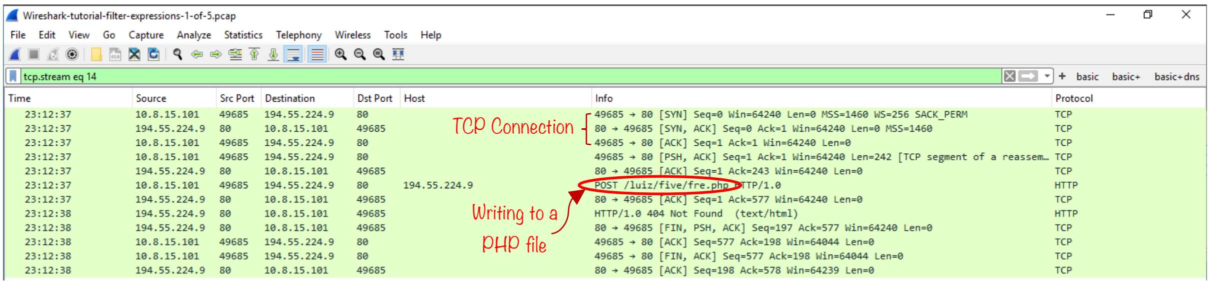


(http.request or tls.handshake.type eq 1) and !(ssdp)

Time	Source	Src Port	Destination	Dst Port	Host	Info	Protocol
23:11:44	10.8.15.101	49679	23.194.229.33	443	kv501.prod.do.dsp.mp.microsoft…	Client Hello (SNI=kv501.prod.do.dsp.mp.microsoft.com)	TLSv1.2
23:11:44	10.8.15.101	49680	23.194.229.33	443	cp501.prod.do.dsp.mp.microsoft…	Client Hello (SNI=cp501.prod.do.dsp.mp.microsoft.com)	TLSv1.2
23:11:44	10.8.15.101	49681	23.194.229.33	443	disc501.prod.do.dsp.mp.microso…	Client Hello (SNI=disc501.prod.do.dsp.mp.microsoft.com)	TLSv1.2
23:11:45	10.8.15.101	49682	52.184.216.246	443	array506.prod.do.dsp.mp.micros…	Client Hello (SNI=array506.prod.do.dsp.mp.microsoft.com)	TLSv1.2
23:12:16	10.8.15.101	49683	20.7.1.246	443	client.wns.windows.com	Client Hello (SNI=client.wns.windows.com)	TLSv1.2
23:12:33	10.8.15.101	49684	20.75.60.91	443	fd.api.iris.microsoft.com	Client Hello (SNI=fd.api.iris.microsoft.com)	TLSv1.2
23:12:37	10.8.15.101	49685	194.55.224.9	80	194.55.224.9	POST /luiz/five/fre.php HTTP/1.0	HTTP
23:12:38	10.8.15.101	49686	194.55.224.9	80	194.55.224.9	POST /luiz/five/fre.php HTTP/1.0	HTTP
23:12:39	10.8.15.101	49687	194.55.224.9	80	194.55.224.9	POST /luiz/five/fre.php HTTP/1.0	HTTP
23:13:35	10.8.15.101	49688	20.42.73.24	443	v10.events.data.microsoft.com	Client Hello (SNI=v10.events.data.microsoft.com)	TLSv1.2
23:13:39	10.8.15.101	49689	194.55.224.9	80	194.55.224.9	POST /luiz/five/fre.php HTTP/1.0	HTTP
23:14:40	10.8.15.101	49690	194.55.224.9	80	194.55.224.9	POST /luiz/five/fre.php HTTP/1.0	HTTP
23:15:04	10.8.15.101	49691	20.75.60.91	443	fd.api.iris.microsoft.com	Client Hello (SNI=fd.api.iris.microsoft.com)	TLSv1.2
23:15:04	10.8.15.101	49692	20.75.60.91	443	fd.api.iris.microsoft.com	Client Hello (SNI=fd.api.iris.microsoft.com)	TLSv1.2

Figure 14: Wireshark Capture Showing Suspicious HTTP POST Requests

Upon following the TCP stream, a TCP connection to this malicious domain can be seen. Therefore, these machines are probably affected with Loki Malware. Further analysis of the payload should be done to confirm these suspicions.



tcp.stream eq 14

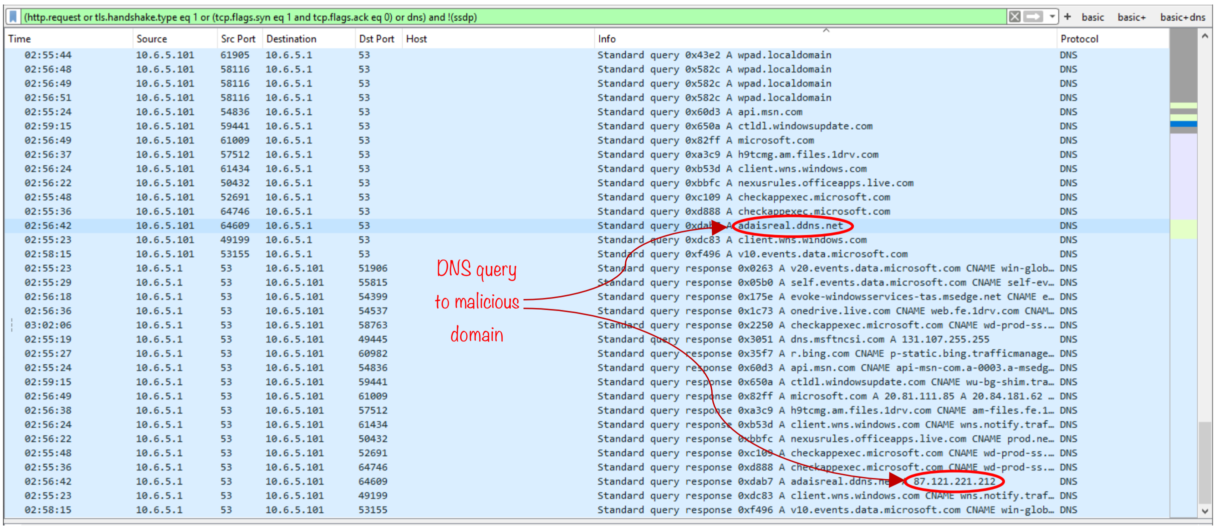
Time	Source	Src Port	Destination	Dst Port	Host	Info	Protocol
23:12:37	10.8.15.101	49685	194.55.224.9	80		49685 → 80 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 WS=256 SACK_PERM	TCP
23:12:37	194.55.224.9	80	10.8.15.101	49685		80 → 49685 [SYN, ACK] Seq=0 Ack=1 Win=64240 Len=0 MSS=1460	TCP
23:12:37	10.8.15.101	49685	194.55.224.9	80		49685 → 80 [ACK] Seq=1 Ack=1 Win=64240 Len=0	TCP
23:12:37	10.8.15.101	49685	194.55.224.9	80		49685 → 80 [PSH, ACK] Seq=1 Ack=1 Win=64240 Len=242 [TCP segment of a reassem…	TCP
23:12:37	194.55.224.9	80	10.8.15.101	49685		80 → 49685 [ACK] Seq=1 Ack=243 Win=64240 Len=0	TCP
23:12:37	10.8.15.101	49685	194.55.224.9	80	194.55.224.9	POST /luiz/five/fre.php HTTP/1.0	HTTP
23:12:37	194.55.224.9	80	10.8.15.101	49685		80 → 49685 [ACK] Seq=1 Ack=577 Win=64240 Len=0	TCP
23:12:38	194.55.224.9	80	10.8.15.101	49685		HTTP/1.0 404 Not Found (text/html)	HTTP
23:12:38	194.55.224.9	80	10.8.15.101	49685		80 → 49685 [FIN, PSH, ACK] Seq=197 Ack=577 Win=64240 Len=0	TCP
23:12:38	10.8.15.101	49685	194.55.224.9	80		49685 → 80 [ACK] Seq=577 Ack=198 Win=64044 Len=0	TCP
23:12:38	10.8.15.101	49685	194.55.224.9	80		49685 → 80 [FIN, ACK] Seq=577 Ack=198 Win=64044 Len=0	TCP
23:12:38	194.55.224.9	80	10.8.15.101	49685		80 → 49685 [ACK] Seq=198 Ack=578 Win=64239 Len=0	TCP

Figure 15: Wireshark Capture following TCP Stream

4.5 Case Five: Detecting Ave Maria Trojan using Wireshark

Ave Maria RAT (remote access trojan) has been suspected to have affected a few company machines because recent threat intelligence indicates a rise in Ave Maria RAT campaigns targeting similar organizations. This malware is spread through phishing emails and its malicious activity includes remote desktop control, stealing data, privilege escalation, browser credential parsing, email credential collections, and more [12]. To check if there has been a breach, they are pcap files [13] that have been captured over the company network.

The traffic has been filtered to show http traffic, attempts to start a TCP connection and DNS queries. In the capture, there is a DNS query for a known malicious domain, **adaisreal.ddns.net**, which resolves to the IP address **87.121.221.212** [15]. This domain has been associated with Ave Maria before.

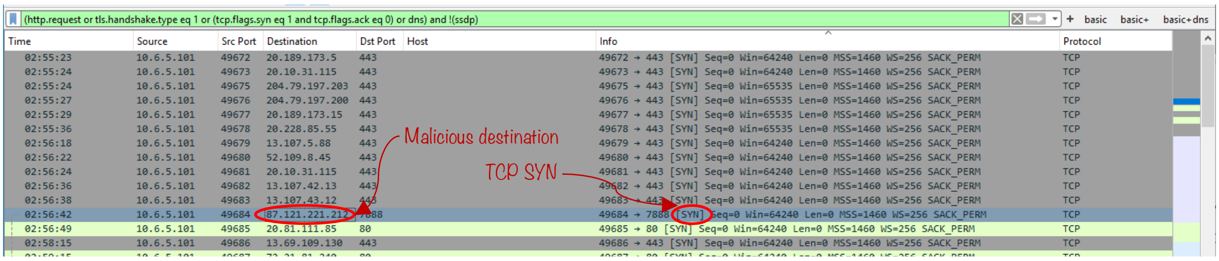


(http.request or tls.handshake.type eq 1 or (tcp.flags.syn eq 1 and tcp.flags.ack eq 0) or dns) and !(ssdp)

Time	Source	Src Port	Destination	Dst Port	Host	Info	Protocol
02:55:44	10.6.5.101	61905	10.6.5.1	53		Standard query 0x43e2 A wpad.localdomain	DNS
02:56:48	10.6.5.101	58116	10.6.5.1	53		Standard query 0x582c A wpad.localdomain	DNS
02:56:49	10.6.5.101	58116	10.6.5.1	53		Standard query 0x582c A wpad.localdomain	DNS
02:56:51	10.6.5.101	58116	10.6.5.1	53		Standard query 0x582c A wpad.localdomain	DNS
02:55:24	10.6.5.101	54836	10.6.5.1	53		Standard query 0x60d3 A api.msn.com	DNS
02:59:15	10.6.5.101	59441	10.6.5.1	53		Standard query 0x650a A ctldl.windowsupdate.com	DNS
02:56:49	10.6.5.101	61009	10.6.5.1	53		Standard query 0x82ff A microsoft.com	DNS
02:56:37	10.6.5.101	57512	10.6.5.1	53		Standard query 0xa3c9 A h9tcmg.am.files.1drv.com	DNS
02:56:24	10.6.5.101	61434	10.6.5.1	53		Standard query 0xb53d A client.wns.windows.com	DNS
02:56:22	10.6.5.101	50432	10.6.5.1	53		Standard query 0xbbfc A nexusrules.officeapps.live.com	DNS
02:55:48	10.6.5.101	52691	10.6.5.1	53		Standard query 0xc109 A checkappexec.microsoft.com	DNS
02:55:36	10.6.5.101	64746	10.6.5.1	53		Standard query 0xd888 A checkappexec.microsoft.com	DNS
02:56:42	10.6.5.101	64609	10.6.5.1	53		Standard query 0xdab7 A adaisreal.ddns.net	DNS
02:55:23	10.6.5.101	49199	10.6.5.1	53		Standard query 0xdc83 A client.wns.windows.com	DNS
02:58:15	10.6.5.101	53155	10.6.5.1	53		Standard query 0xf496 A v10.events.data.microsoft.com	DNS
02:55:23	10.6.5.1	53	10.6.5.101	51906		Standard query response 0x0263 A v20.events.data.microsoft.com CNAME win-glob…	DNS
02:55:29	10.6.5.1	53	10.6.5.101	55815		Standard query response 0x05b0 A self.events.data.microsoft.com CNAME self-ev…	DNS
02:56:18	10.6.5.1	53	10.6.5.101	54399		Standard query response 0x175e A evoke-windowsservices-tas.msedge.net CNAME e…	DNS
02:56:36	10.6.5.1	53	10.6.5.101	54537		Standard query response 0x1c73 A onedrive.live.com CNAME web.fe.1drv.com CNAM…	DNS
03:02:06	10.6.5.1	53	10.6.5.101	58763		Standard query response 0x2250 A checkappexec.microsoft.com CNAME wd-prod-ss.…	DNS
02:55:19	10.6.5.1	53	10.6.5.101	49445		Standard query response 0x3051 A dns.msftncsi.com A 131.107.255.255	DNS
02:55:27	10.6.5.1	53	10.6.5.101	60982		Standard query response 0x35f7 A r.bing.com CNAME p-static.bing.trafficmanage…	DNS
02:55:24	10.6.5.1	53	10.6.5.101	54836		Standard query response 0x60d3 A api.msn.com CNAME api-msn-com.a-0003.a-msedg…	DNS
02:59:15	10.6.5.1	53	10.6.5.101	59441		Standard query response 0x650a A ctldl.windowsupdate.com CNAME wu-bg-shim.tra…	DNS
02:56:49	10.6.5.1	53	10.6.5.101	61009		Standard query response 0x82ff A microsoft.com A 20.81.111.85 A 20.84.181.62 …	DNS
02:56:38	10.6.5.1	53	10.6.5.101	57512		Standard query response 0xa3c9 A h9tcmg.am.files.1drv.com CNAME am-files.fe.1…	DNS
02:56:24	10.6.5.1	53	10.6.5.101	61434		Standard query response 0xb53d A client.wns.windows.com CNAME wns.notify.traf…	DNS
02:56:22	10.6.5.1	53	10.6.5.101	50432		Standard query response 0xbbfc A nexusrules.officeapps.live.com CNAME prod.ne…	DNS
02:55:48	10.6.5.1	53	10.6.5.101	52691		Standard query response 0xc109 A checkappexec.microsoft.com CNAME wd-prod-ss.…	DNS
02:55:36	10.6.5.1	53	10.6.5.101	64746		Standard query response 0xd888 A checkappexec.microsoft.com CNAME wd-prod-ss.…	DNS
02:56:42	10.6.5.1	53	10.6.5.101	64609		Standard query response 0xdab7 A adaisreal.ddns.net A 87.121.221.212	DNS
02:55:23	10.6.5.1	53	10.6.5.101	49199		Standard query response 0xdc83 A client.wns.windows.com CNAME wns.notify.traf…	DNS
02:58:15	10.6.5.1	53	10.6.5.101	53155		Standard query response 0xf496 A v10.events.data.microsoft.com CNAME win-glob…	DNS

Figure 16: Wireshark Capture Showing DNS Query to Malicious Domain

Further analysis reveals a TCP segment directed to **87.121.221.212** over port **7888** with the SYN flag set. This port and communication pattern are characteristic of Ave Maria RAT's command and control (C2) communications [12].



(http.request or tls.handshake.type eq 1 or (tcp.flags.syn eq 1 and tcp.flags.ack eq 0) or dns) and !(ssdp)

Time	Source	Src Port	Destination	Dst Port	Host	Info	Protocol
02:55:23	10.6.5.101	49672	20.189.173.5	443		49672 → 443 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 WS=256 SACK_PERM	TCP
02:55:24	10.6.5.101	49673	20.10.31.115	443		49673 → 443 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 WS=256 SACK_PERM	TCP
02:55:24	10.6.5.101	49675	204.79.197.203	443		49675 → 443 [SYN] Seq=0 Win=65535 Len=0 MSS=1460 WS=256 SACK_PERM	TCP
02:55:27	10.6.5.101	49676	204.79.197.200	443		49676 → 443 [SYN] Seq=0 Win=65535 Len=0 MSS=1460 WS=256 SACK_PERM	TCP
02:55:29	10.6.5.101	49677	20.189.173.15	443		49677 → 443 [SYN] Seq=0 Win=65535 Len=0 MSS=1460 WS=256 SACK_PERM	TCP
02:55:36	10.6.5.101	49678	20.228.85.55	443		49678 → 443 [SYN] Seq=0 Win=65535 Len=0 MSS=1460 WS=256 SACK_PERM	TCP
02:56:18	10.6.5.101	49679	13.107.5.88	443		49679 → 443 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 WS=256 SACK_PERM	TCP
02:56:22	10.6.5.101	49680	52.109.8.45	443		49680 → 443 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 WS=256 SACK_PERM	TCP
02:56:24	10.6.5.101	49681	20.10.31.115	443		49681 → 443 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 WS=256 SACK_PERM	TCP
02:56:36	10.6.5.101	49682	13.107.42.13	443		49682 → 443 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 WS=256 SACK_PERM	TCP
02:56:38	10.6.5.101	49683	13.107.43.12	443		49683 → 443 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 WS=256 SACK_PERM	TCP
02:56:42	10.6.5.101	49684	87.121.221.212	7888		49684 → 7888 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 WS=256 SACK_PERM	TCP
02:56:49	10.6.5.101	49685	20.81.111.85	80		49685 → 80 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 WS=256 SACK_PERM	TCP
02:58:15	10.6.5.101	49686	13.69.109.130	443		49686 → 443 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 WS=256 SACK_PERM	TCP

Figure 17: Wireshark Capture Showing attempt to start TCP Connection to Malicious Domain

5 Conclusion

Digital forensics is a vast field that requires using and understanding various tools. From cracking passwords to access evidence and analyzing network traffic, any data can be useful evidence if correctly and carefully analyzed. John the Ripper is a proficient tool that can be used to crack various hashes and Wireshark allows for the detailed examination of network interactions. As the saying goes, "Everywhere you go, you leave a footprint behind."

6 References

[1] D. Williams, “Discover how technology helps manage the growth in digital evidence- Microsoft Industry Blogs,” Microsoft Industry Blogs, Sep. 20, 2022. Available: https://www.microsoft.com/en-us/industry/blog/government/2022/09/20/discover-how-technology-helps-manage-the-growth-in-digital-evidence/ (accessed Feb. 24, 2024).

[2] “DIGITAL EVIDENCE TASK FORCE EXECUTIVE PRIMER.” Available: https://www.theiacp.org/sites/default/files/2019-11/IACP_Digital_Evidence_Task_Force.pdf (accessed Feb. 24, 2024).

[3] L. Mathews, “Stealthy TrickBot Malware Has Compromised 250 Million Email Accounts And Is Still Going Strong,” *Forbes*. https://www.forbes.com/sites/leemathews/2019/07/14/stealthy-trickbot-malware-has-compromised-250-million-email-accounts-and-is-still-going-strong/?sh=432eb1864884 (accessed Feb. 24, 2024).

[4] “openwall/john,” *GitHub*, Sep. 25, 2020. https://github.com/openwall/john (accessed Feb. 26, 2024).

[5] M. Shivanandhan, “How to Crack Passwords Using John the Ripper – Pentesting Tutorial,” *freeCodeCamp.org*, Nov. 17, 2022. https://www.freecodecamp.org/news/crack-passwords-using-john-the-ripper-pentesting-tutorial/ (accessed Feb. 28, 2024).

[6] “John The Ripper,” Bugcrowd. https://www.bugcrowd.com/glossary/john-the-ripper/#:~:text=Ethical%20hackers%20and%20penetration%20testers

[7] Openwall, “John the Ripper password cracker,” *Openwall*, 2019. https://www.openwall.com/john/ (accessed Feb. 28, 2024).

[8] “ExifTool by Phil Harvey,” exiftool.org. https://exiftool.org/#:~:text=ExifTool%20can%20Read%2C%20Write%20and (accessed Feb. 28, 2024).

[9] “Autopsy (software),” Wikipedia, Nov. 13, 2023. https://en.wikipedia.org/wiki/Autopsy_(software)#:~:text=Autopsy%20analyzes%20major%20file%20systems (accessed Feb. 28, 2024)

[10] Wireshark, “Wireshark · About,” www.wireshark.org, 2023. https://www.wireshark.org/about.html (accessed Feb. 28, 2024).

[11] J. P. Updated: 12/21/2018, “What is Mimikatz: The Beginner’s Guide | Varonis,” Inside Out Security, Dec. 21, 2018. https://www.varonis.com/blog/what-is-mimikatz (accessed Feb. 28, 2024).

[12] “Defending the Gates: Understanding and Detecting Ave Maria (Warzone) RAT,” Splunk. https://www.splunk.com/en_us/blog/security/defending-the-gates-understanding-and-detecting-ave-maria-warzone-rat.html (accessed Feb. 29, 2024).

[13] B. Duncan, “Wireshark Tutorial: Display Filter Expressions,” Unit42, Jan. 11, 2019. https://unit42.paloaltonetworks.com/using-wireshark-display-filter-expressions/ (accessed Feb. 29, 2024).

[14] “ThreatFox | http://194.55.224.9/luiz/five/fre.php,” threatfox.abuse.ch. https://threatfox.abuse.ch/ioc/1149105/ (accessed Feb. 29, 2024).

[15] “VirusTotal,” www.virustotal.com. https://www.virustotal.com/gui/domain/adaisreal.ddns.net (accessed Feb. 29, 2024).

[16] “The Need For Digital Forensics: Why Digital Forensics Is Important?,” financialcrimeacademy.org, Aug. 15, 2023. https://financialcrimeacademy.org/need-for-digital-forensics/