DFIR Final Project CS-07, MIAMI August 13, 2020

Overview:

The Final Project of the Digital Forensics and Incident Response course involves an examination of a captured network traffic log file, investigate_me.pcap. The log traffic will contain clues and evidence of malign activity. The log files must be viewed and parsed to find and provide evidentiary proof of the network activity and malign files.

Another file, investigate_me2.ova, which is an image of the suspect PC that was used by an attacker, will be examined to find the malicious code and explore how a vulnerability was exploited.

The original exploit allowed for another type of malware to be introduced and to also be exploited. The tasks of the Final Project are to examine the available files, document the various stages of the exploits, follow the clues to the next stages, and unlock the final file, Investigate_me3.bin.

Setup:

On a Kali VM in VirtualBox, download 2 files for examination.

investigate_me.pcap

Saved in directory: /home/kali/Documents/DFIR_Final/

Investigate_me3.bin

Saved in directory: /home/kali/Documents/DFIR_Final/

On a Windows 10 VM in VirtualBox, download the same 2 files for examination.

investigate_me.pcap

Saved in directory: C:\Users\Sudo\Downloads\DFIR_Final\

Investigate_me3.bin

Saved in directory: C:\Users\Sudo\Downloads\DFIR_Final\

On the Host PC, download 1 file for examination.

Investigate_me2.rar

Saved in directory: C:\Users\Admin\Downloads

Tasks

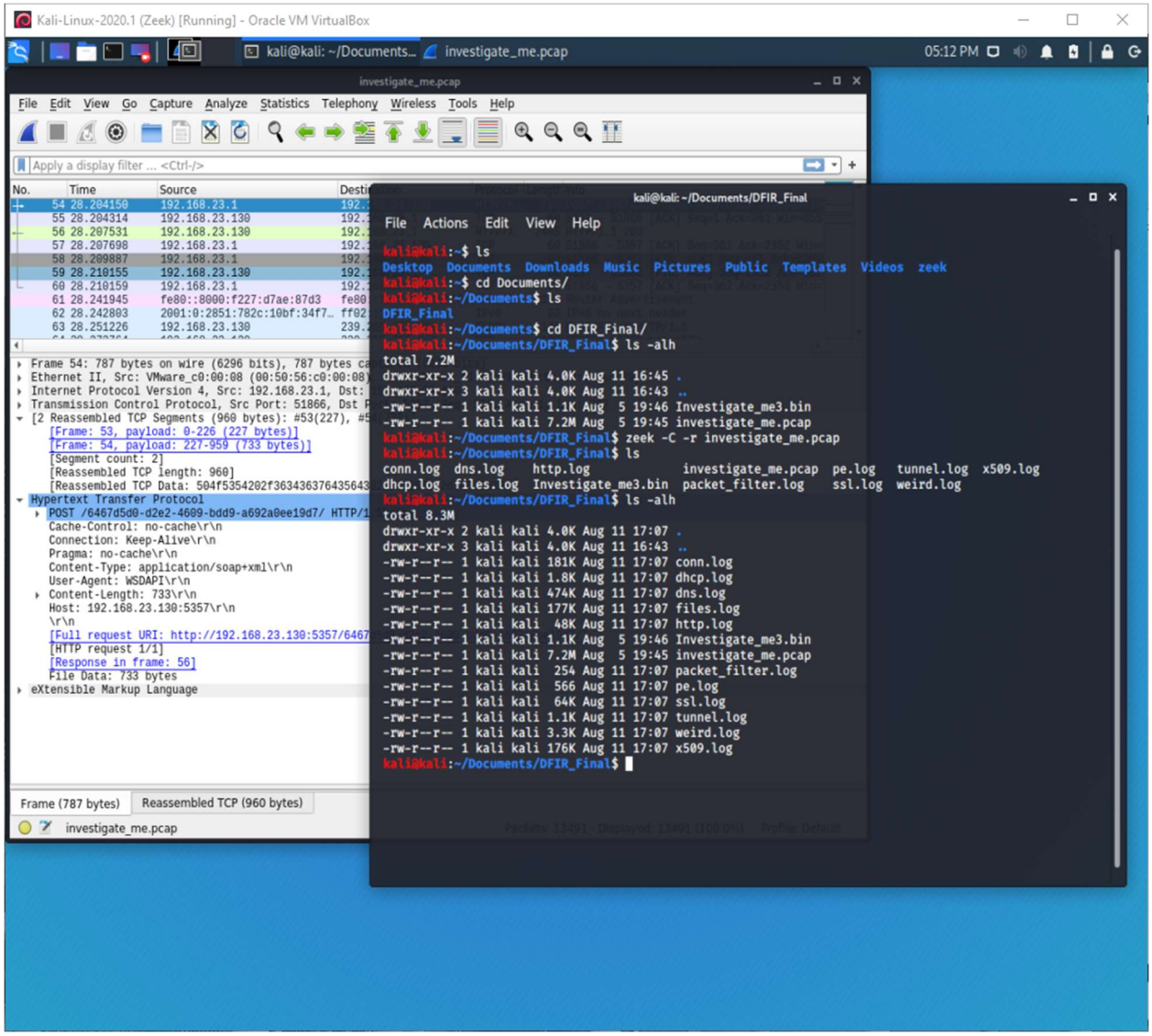
1) Study the .pcap file. Find any suspicious activity.

a. Run the network packet analyzer S/W, Wireshark. In the **File** menu, choose **Open**, then select the .pcap file to examine, **investigate_me.pcap**. Scan the output display for any anomalies.

b. Also, run the .pcap file through the Zeek parser software. Zeek will generate a series of log files that will be available for further analysis.

Syntax: zeek -C -r investigate_me.pcap

i. 10 log files were generated in the same directory as the original .pcap file.



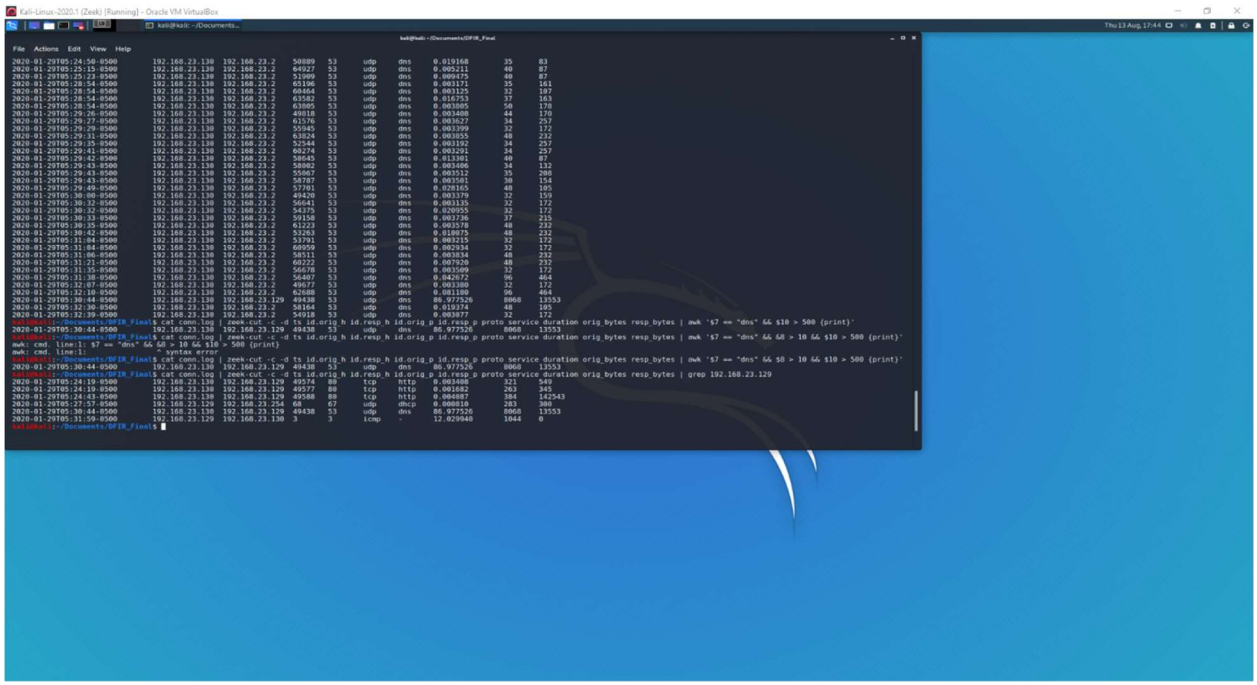
2) Use Zeek to extract valuable information from the .pcap file.

a. Looking through the **conn.log** file, there were many connections that used the UDP protocol for the DNS service. Typically, the responding IP Address appeared to be a DNS server or a host at address 192.168.23.2 with a very short connection duration (a fraction of a second) and a very small responding byte count (ranging 105 – 464 bytes). There was only 1 connection (ID of responding address is 192.168.23.129) using the DNS service for a long time (86.977526 seconds) and a very high responding byte count (13553 bytes) and a very high responding byte count (13553 bytes).

Syntax: cat conn.log | zeek-cut -c -d ts id.orig_h id.resp_h id.orig_p id.resp_p proto service duration orig_bytes resp_bytes | awk '$7 == "dns" && $8 > 10 && $10 > 500 {print}'

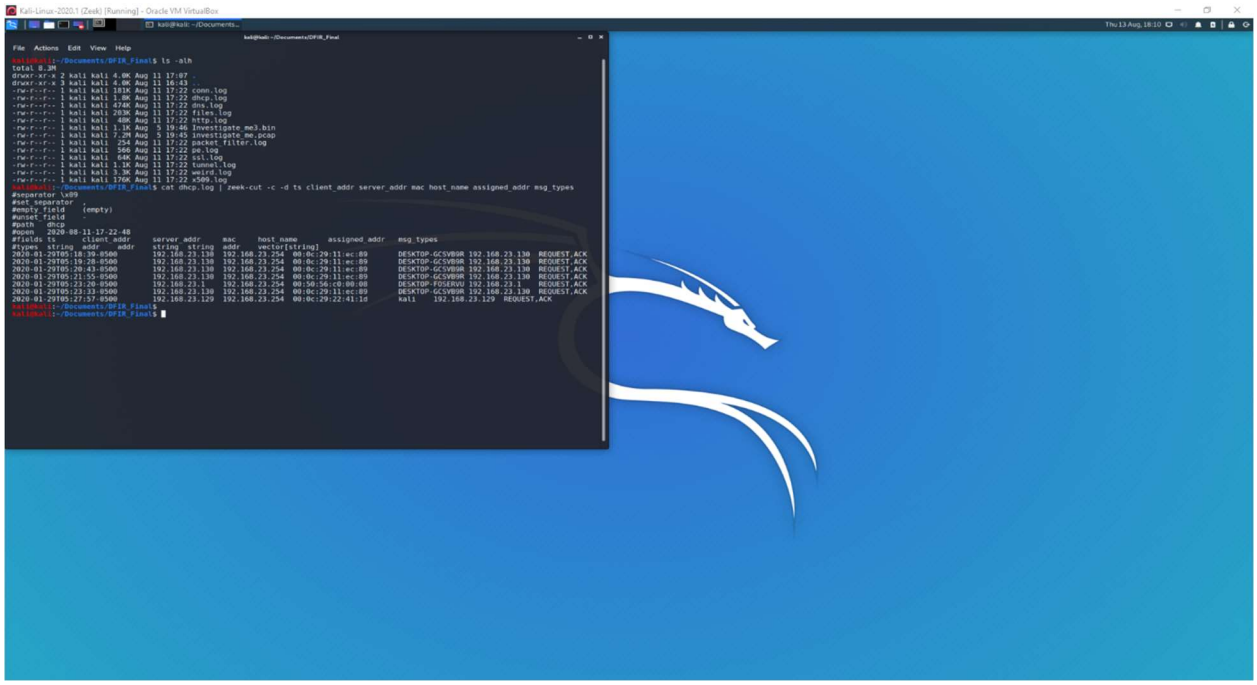
b. Further parsing using the grep expression to display any connections to the found IP Address of 192.168.23.129 revealed a total of 6 connections between the PC that captured the .pcap file (IP Addr 192.168.23.130) and the suspicious PC (192.168.23.129), including a DHCP request at 05:27:57, a DNS connection at 05:30:44, and an ICMP connection at 05:31:59.

cat conn.log | zeek-cut -c -d ts id.orig_h id.resp_h id.orig_p id.resp_p proto service duration orig_bytes resp_bytes | awk '$7 == "dns" && $8 > 10 && $10 > 500 {print}' | grep 192.168.23.129



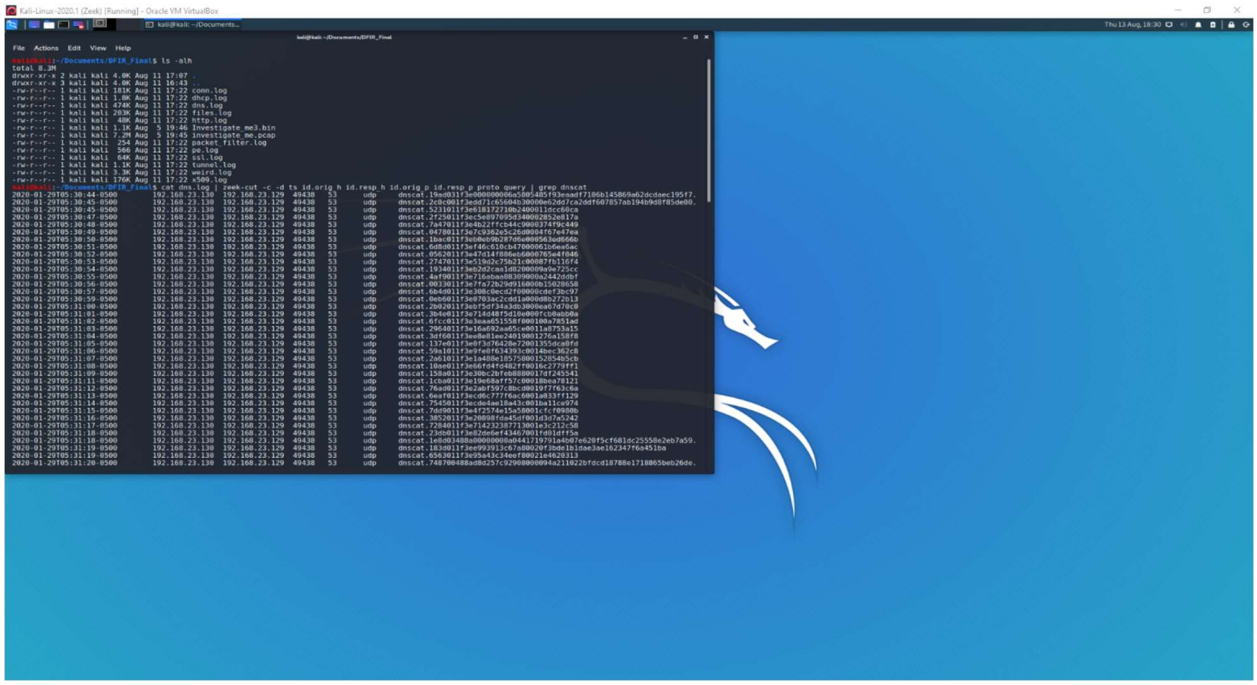
c. Looking through the **dhcp.log** revealed that the rouge PC (IP Addr 192.168.23.129) did request a dynamic IP address from DHCP server 192.168.23.254 at 05:27:57. This log reveals the PC's MAC address (00:0c:29:22:41:1d) and hostname (kali).

Syntax: cat dhcp.log | zeek_cut -c -d ts client_addr server_addr mac host_name assigned_addr msg_types



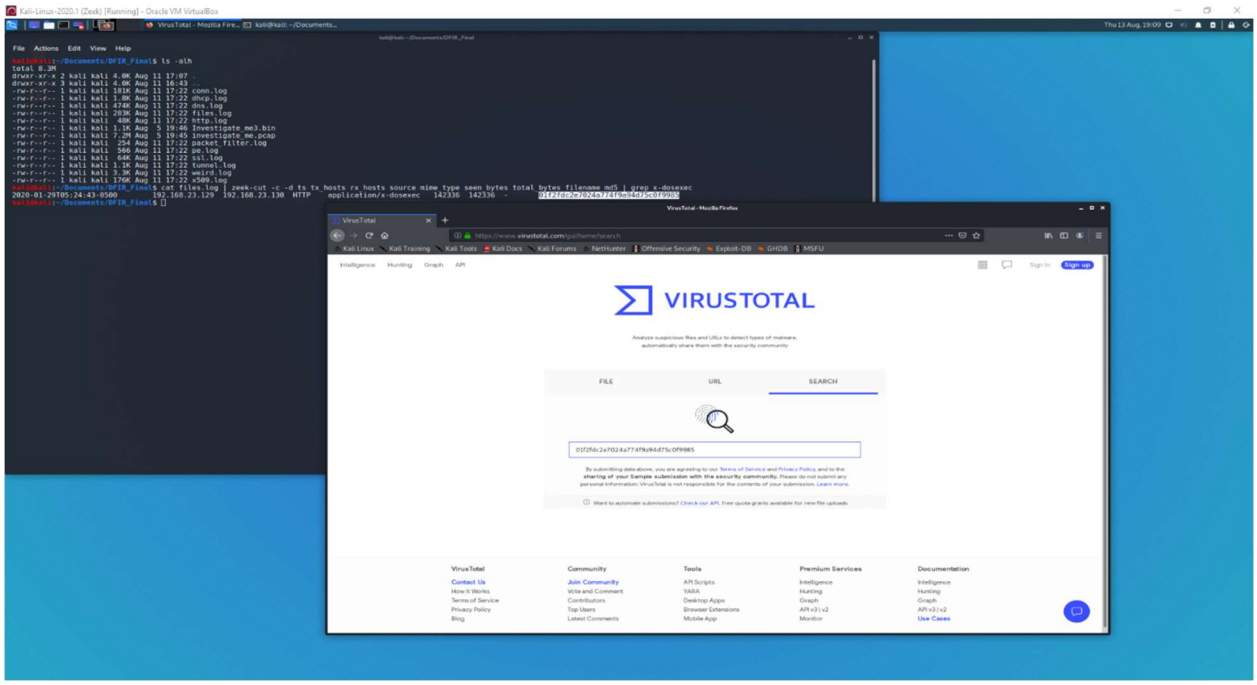
d. Looking through the **dns.log** revealed that there were numerous connections between 192.168.23.130 (port 49438) and 192.168.23.129 (port 53) using the UDP protocol. The first connection was at 05:30:44 and the final connection was at 05:32:06.

Syntax: cat dns.log | zeek-cut -c -d ts id.orig_h id.resp_h id.orig_p id.resp_p proto query | grep dnscat

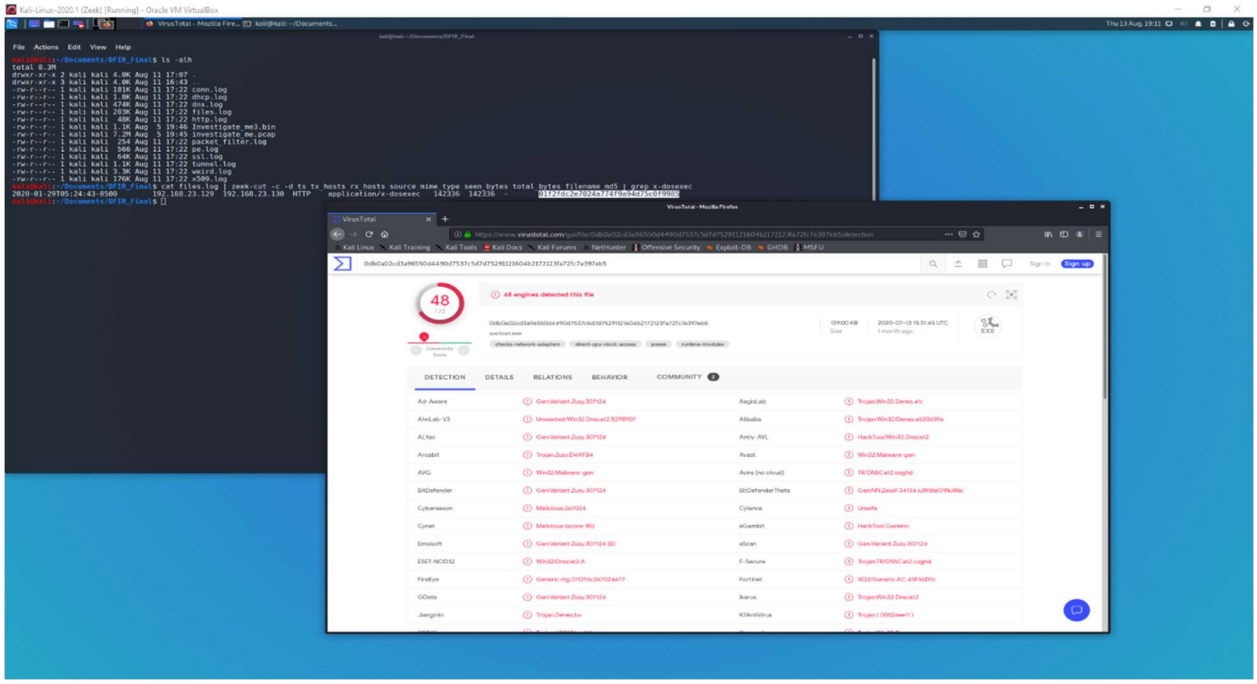


e. Looking through the **files.log** reveals that there is only 1 entry for an executable file of type “x-dosexec”. It was received from the suspicious PC (192.168.23.129) via HTTP protocol at 05:24:43. It is a relatively large file (142336 bytes).
f. The filename doesn’t display but using the md5 expression creates a hash of the file.
g. Copy the md5 hash of the file (01f2fdc2e7024a774f9a94d75c0f9985) in order to use it to compare the file with known viruses via an online virus lookup utility at www.VirusTotal.com.

Syntax: cat files.log | zeek-cut -c -d ts tx_hosts rx_hosts source mime_type seen_bytes total_bytes filename md5 | grep x-dosexec



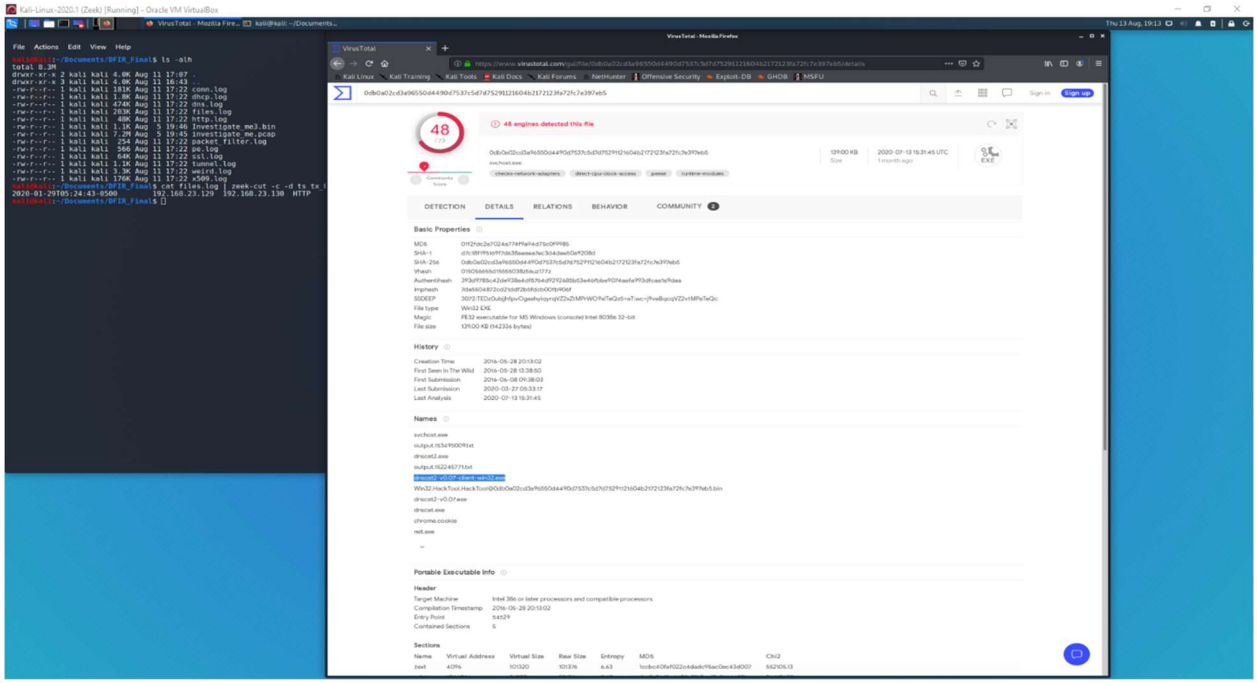
h. Paste the md5 hash into the VirusTotal search field and search for a virus match.



3) Locate the malicious file in the capture.

a. VirusTotal found a match of the md5 hash of the transferred file with a known virus.

b. There is a known virus called “dnscat2-v0.07-client-win32.exe”.



4) Identify the nature of the malicious file.

a. dnscat2 is an application that was written to allow tunneling into a network over the DNS protocol and permit an encrypted Command & Control channel. It allows remote control of an infected (client) PC from a remotely controlled (server) PC. Use of the DNS protocol effectively bypasses firewalls.

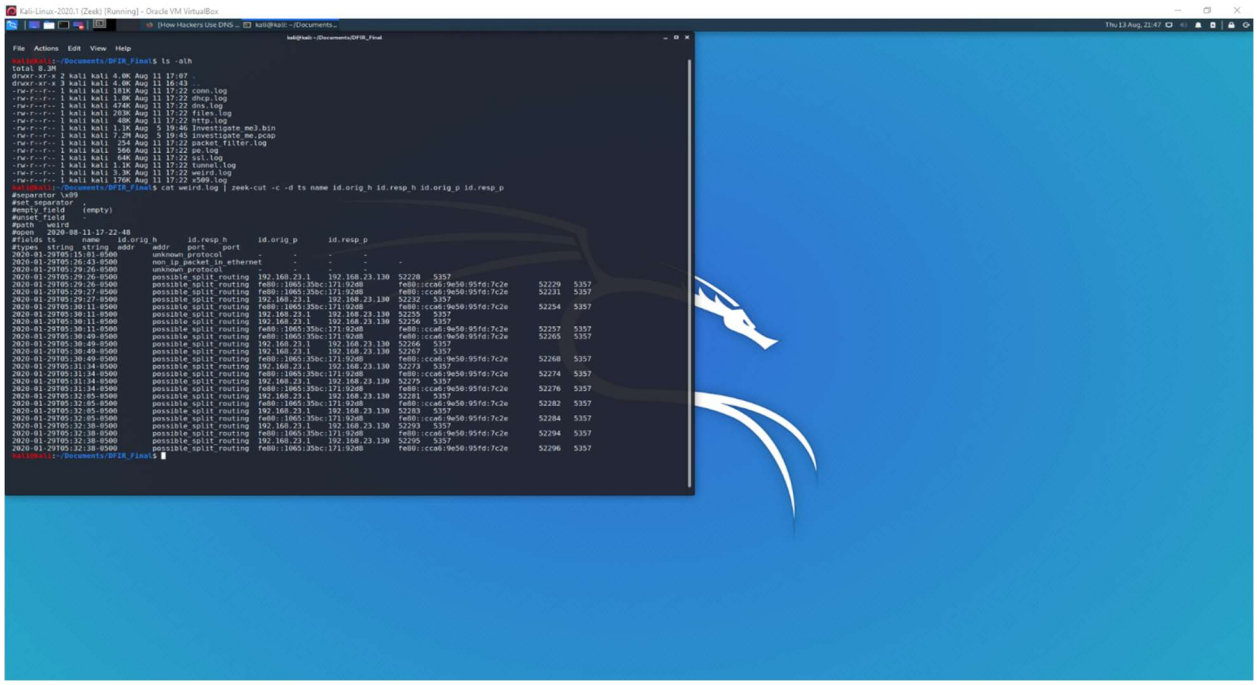
b. dnscat2 requires 2 parts: a client (victim) and a server (attacker). The compromised client typically will specify a domain name for DNS lookup, which the attacker has control over. When the victim PC searches for any DNS lookup, the attacker's server intercepts the lookup and re-directs it and establishes a logical connection. Other files may be embedded into the DNS lookup's reply to the victim PC. File transfers can be either inbound malware to the victim file or outbound files exfiltrating data from the victim PC.

Reference: https://github.com/iagox86/dnscat2

5) Find evidence of malicious activity in the .pcap file.

a. Looking through the **weird.log** file reveals a "non_ip_packet_in_ethernet" message at 05:26:43, then an "unknown_protocol" at 05:29:26, followed by 24 other notices of "possible_split_routing" occurring from 05:29:26 until 05:32:38.

Syntax: cat weird.log | zeek-cut -c -d ts name id.orig_h id.resp_h id.orig_p id.resp_p

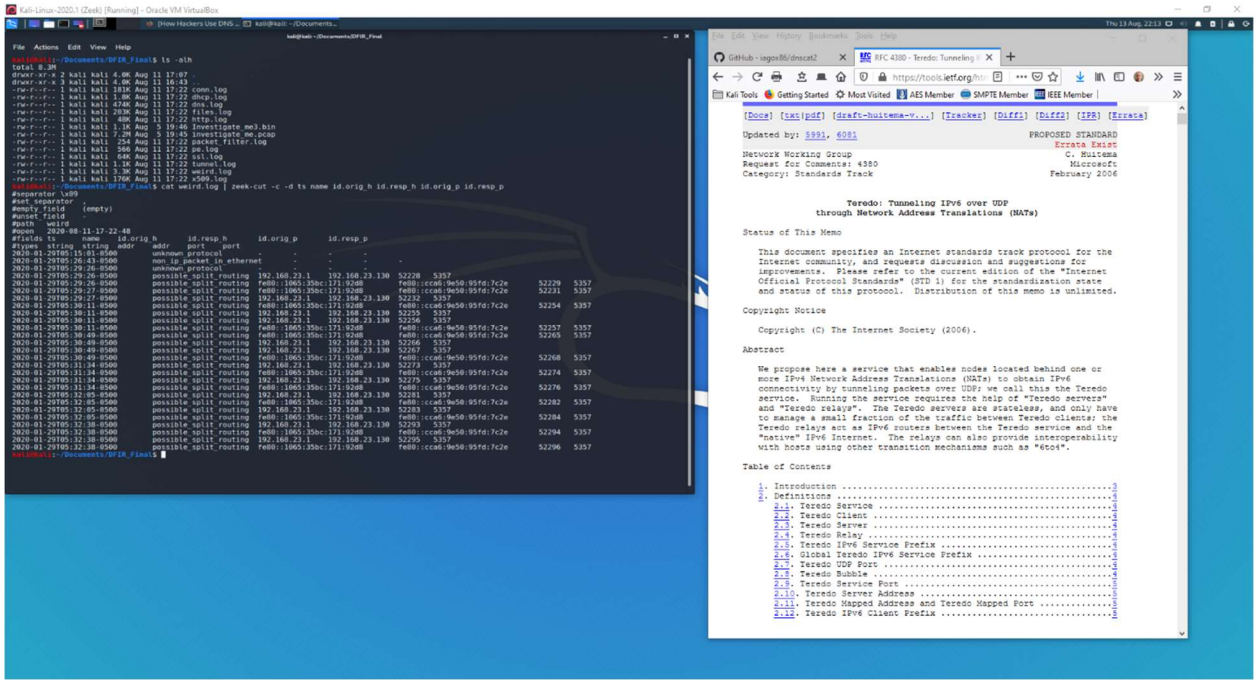


b. Looking through the **tunnel.log** reveals that a Teredo Tunnel was opened at 05:29:26 and then closed at 05:33:39.

c. As described in the IETF's RFC 4380, the Teredo service enables nodes on an IPv4 NAT network to obtain IPv6 connectivity by tunneling packets over UDP protocol on port 3544.

Reference: https://tools.ietf.org/html/rfc4380

Syntax: cat tunnel.log | zeek-cut -c -d ts id.orig_h id.resp_h id.orig_p id.resp_p tunnel_type action



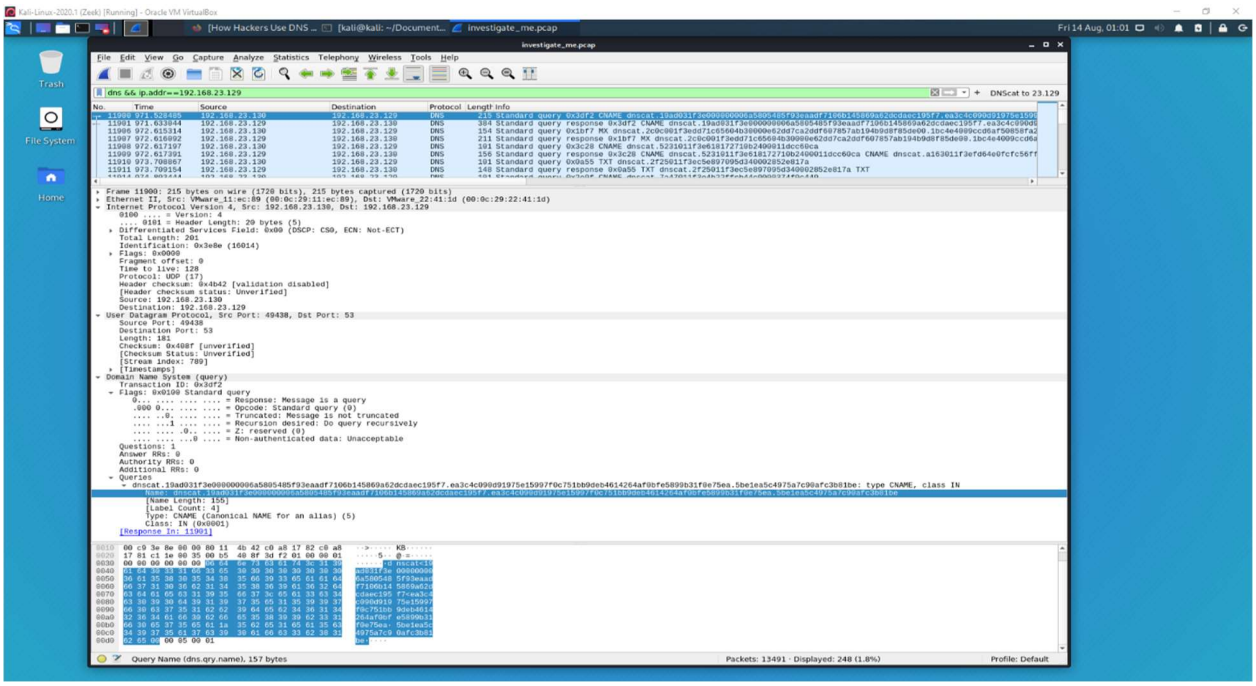
6) Investigate the infected PC (192.168.23.130). Use the .ova file and VMWare's Workstation 15 Player.

*** THE SEQUENCE OF THE FIRST MALICIOUS PACKET IS THE PASSWORD

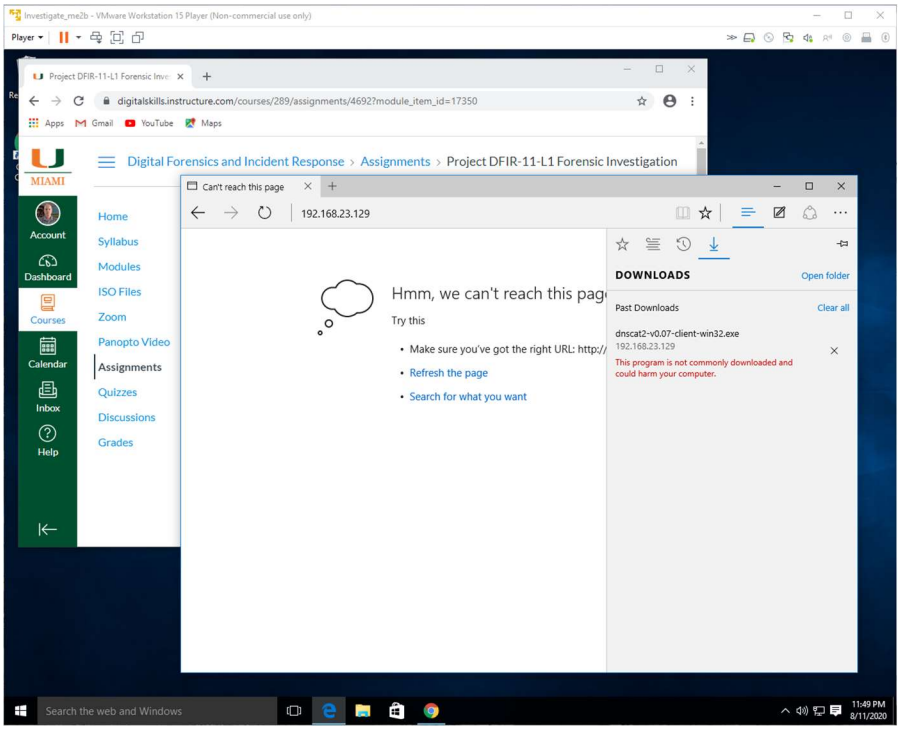
a. Download the provided compressed file, investigate_me2.rar, to the Host PC.
b. Download, install, and launch VMWare's Workstation 15 Player on the Host PC.
c. Uncompress the provided file, which will contain a .ova file that can be imported into VMWare Workstation 15 as a virtual machine that is an image of the infected PC.
    i. The .rar file is password protected. The password will be the sequence (packet #) of the first packet in Wireshark that indicates malicious software in the investigate_me.pcap file.
d. Looking at the file, **investigate_me.pcap**, in Wireshark, search for all connections that use the DNS protocol with 192.168.23.129 (suspected attacker DNS server).

Syntax: dns && ip.addr == 192.168.23.129

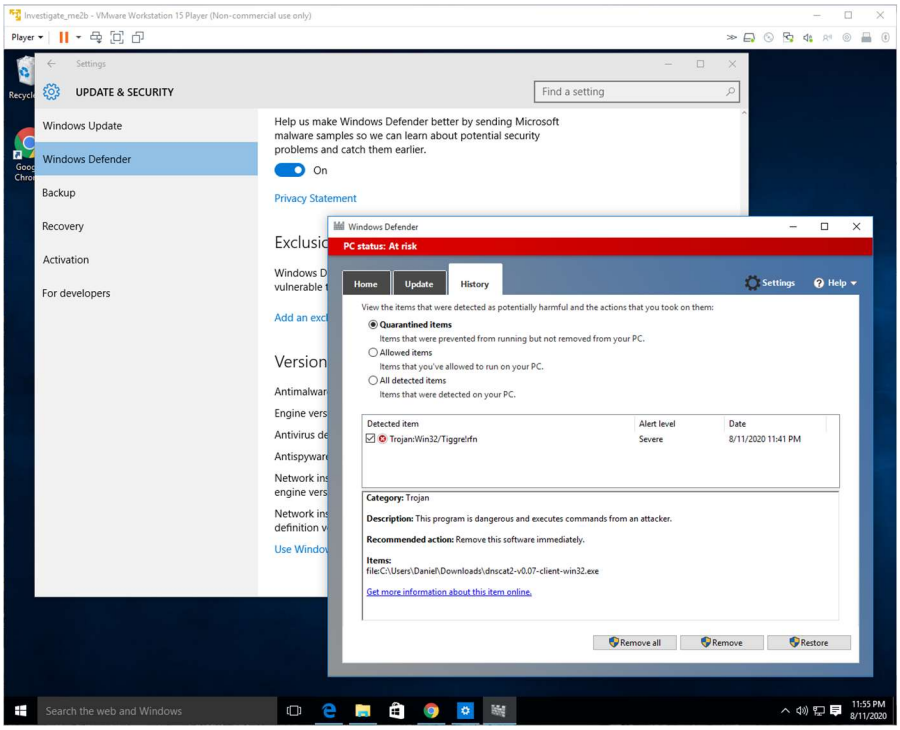
e. There are many packets indicating communication between 192.168.23.129 and 192.168.23.130. The DNS queries all include the prefix, dnscat.
f. The first packet (sequence) that includes the dnscat query is packet # **11900**.
    i. *** Use **11900** as the password to uncompress the file, **investigate_me2.rar**.



g. Uncompress the provided file, **investigate_me.rar**, using "**11900**" as the password.
h. The uncompressed file is an image of a PC in .ova format, named **Investigate_me2.ova**.
i. Import this .ova file into the VMWare Workstation 15 Player and launch the VM.
    i. This VM is a Windows 10 PC with Internet Explorer installed.
    ii. The IE history reveals a previous session with IP Addr 192.168.23.129, which is currently not reachable.
    iii. In the IE downloads can be found the executable well-known malicious file, **dnscat2-v0.07-client-win32.exe**



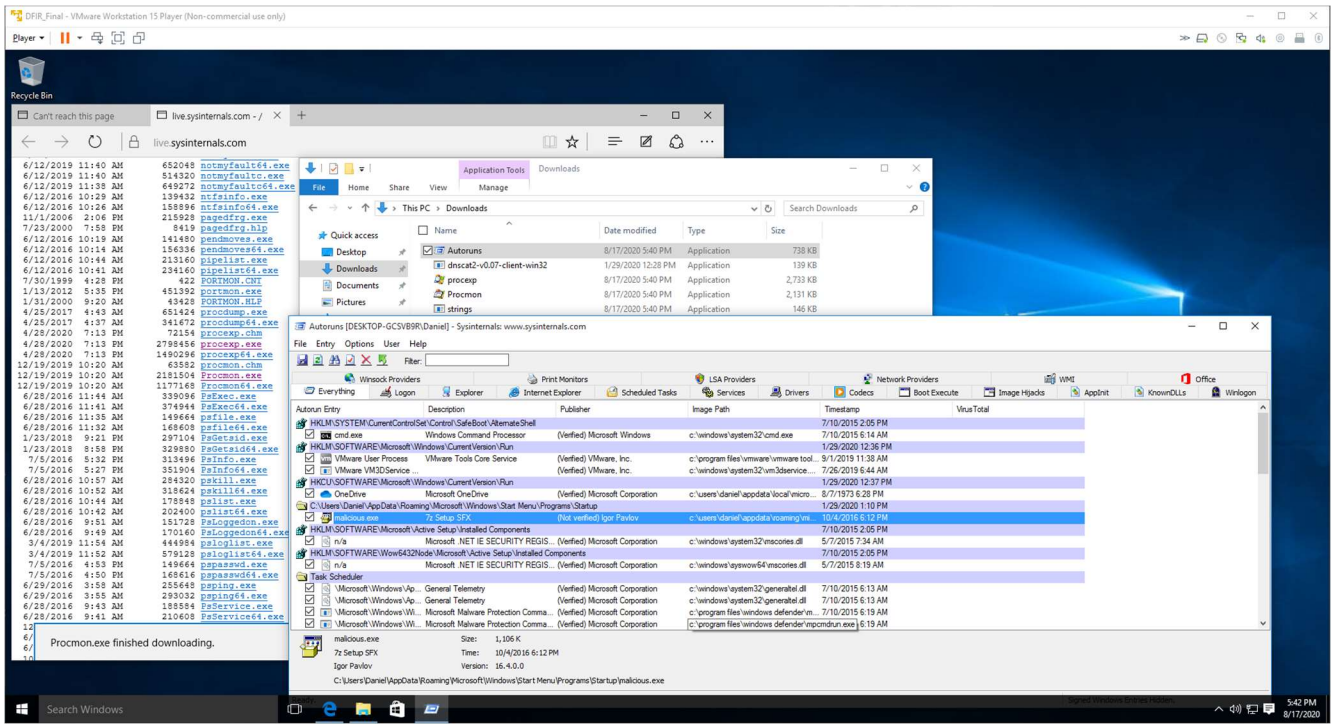
j. The VM's Windows Defender warned of the presence of known malware. The detected item is: **Trojan:Win32/Tiggre!rfn**.



7) Identify backdoors the attacker may have left.

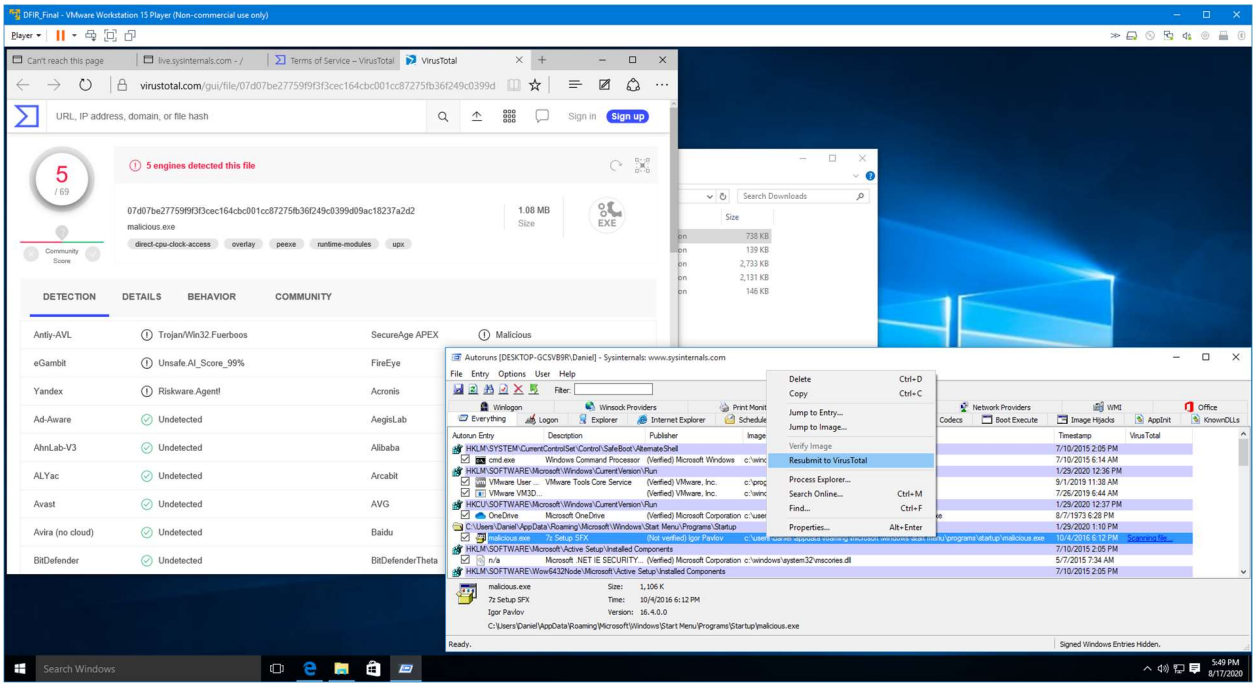
a. Download software utilities from live.sysinternals.com to assist in examining files on the Win 10 VM. Helpful sysinternals may include: autoruns.exe, strings.exe, procmon.exe, procexp.exe, sigcheck.exe.

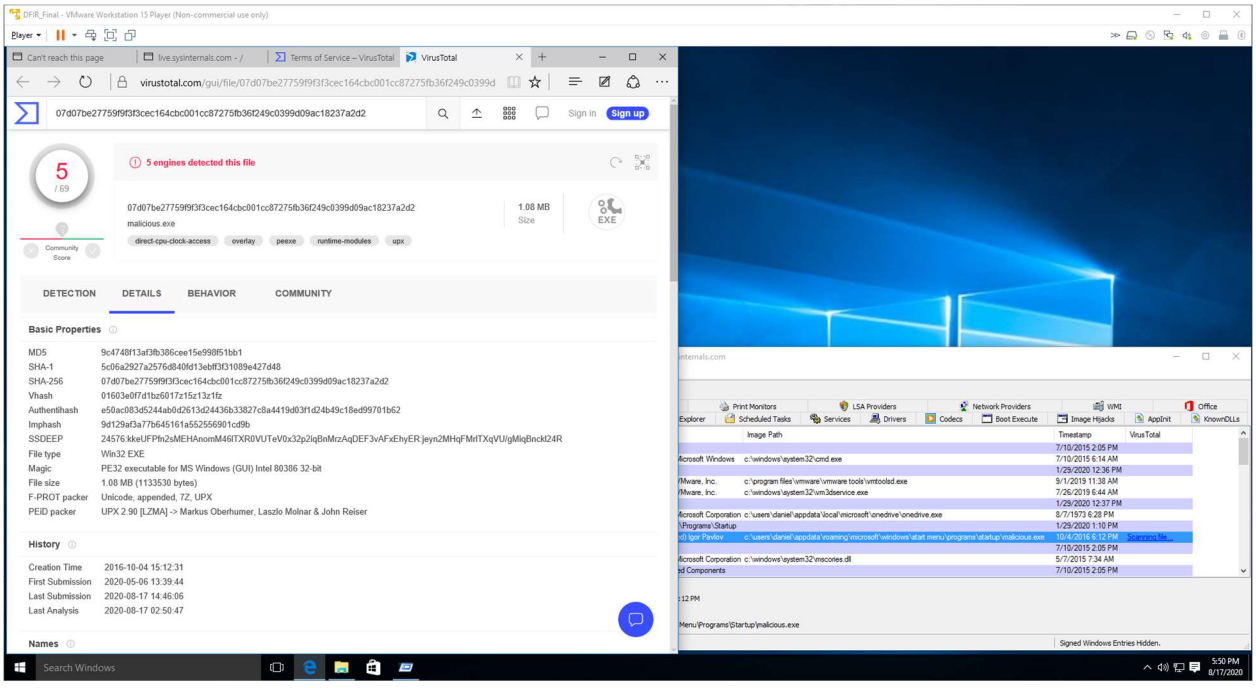
b. Using the sysinternal utility, **autoruns.exe**, reveals that there is an unusual executable file in the Startup directory of the Win 10 VM, called "**malicious.exe**". The file is from an unverified Publisher, therefore it is suspicious.

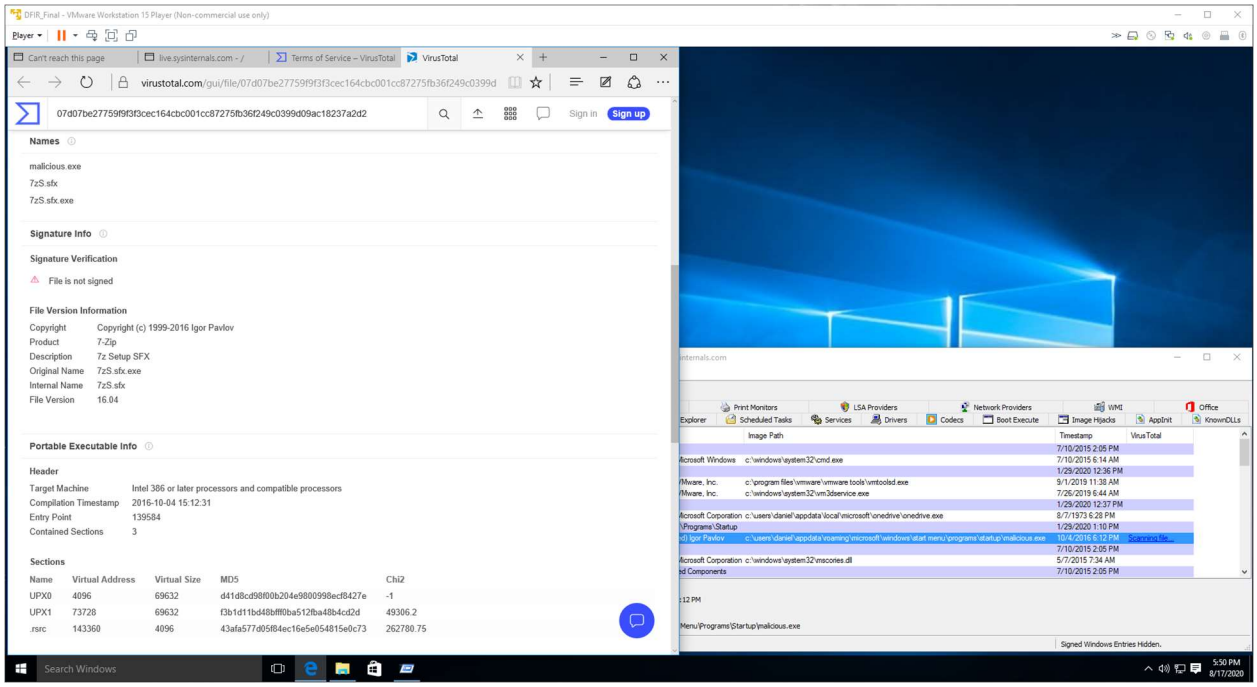


8) Investigate the second malware.

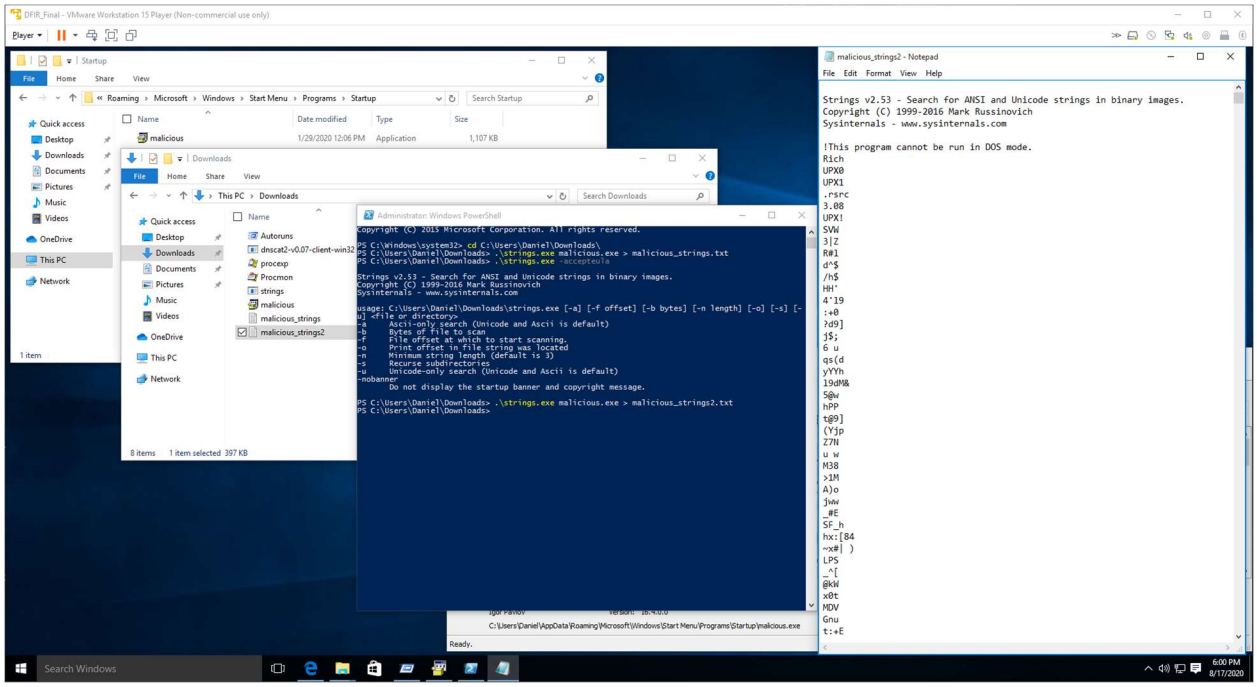
a. Select the file, **malicious.exe**, and right-click on it to have it scanned for virus by Virus Total.


DFIR_Final - VMware Workstation 15 Player (Non-commercial use only)
virustotal.com/gui/file/07d07be27759f9f3f3cec164cbc001cc87275fb36f249c0399d
5 engines detected this file
07d07be27759f9f3f3cec164cbc001cc87275fb36f249c0399d09ac18237a2d2
malicious.exe
1.08 MB
DETECTION
DETAILS
BEHAVIOR
COMMUNITY
Antiy-AVL
Trojan/Win32.Fuerboos
SecureAge APEX
Malicious
eGambit
Unsafe.AI_Score_99%
FireEye
Yandex
Riskware.Agent!
Acronis
Ad-Aware
Undetected
AegisLab
AhnLab-V3
Alibaba
ALYac
Arcabit
Avast
AVG
Avira (no cloud)
Baidu
BitDefender
BitDefenderTheta
Autoruns [DESKTOP-GCSVB9R\Daniel] - Sysinternals: www.sysinternals.com
Delete
Copy
Jump to Entry...
Jump to Image...
Verify Image
Resubmit to VirusTotal
Process Explorer...
Search Online...
Find...
Properties...
Search Windows

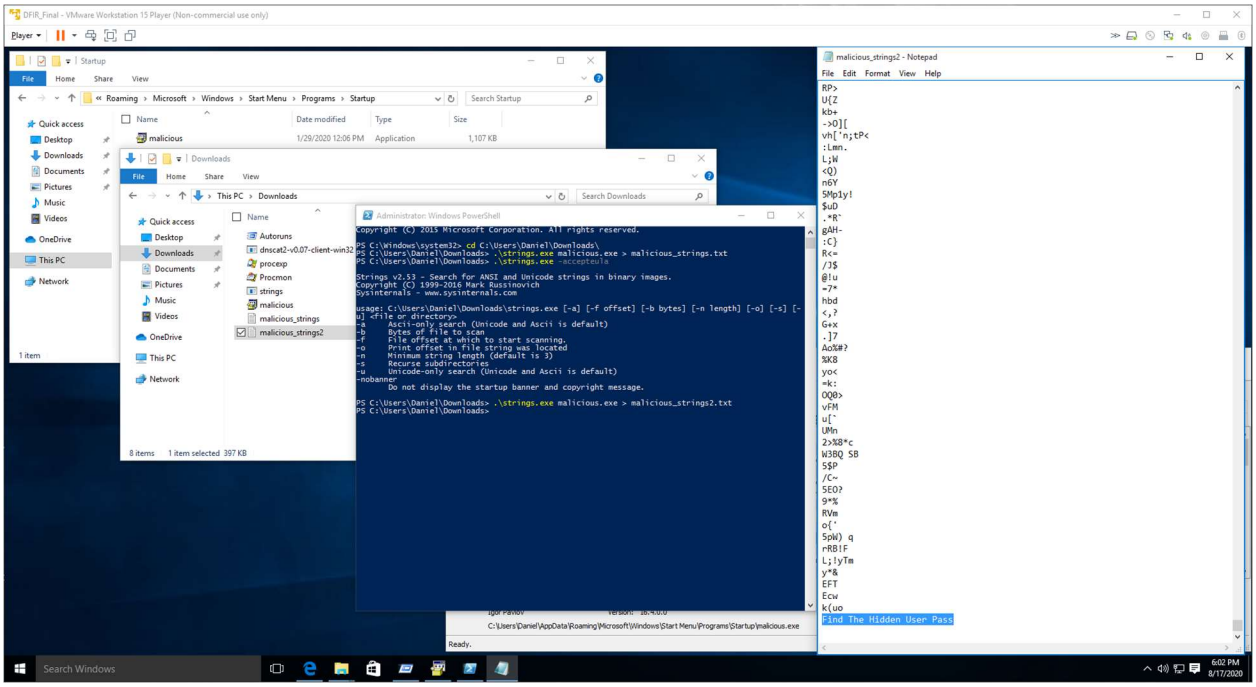

DFIR_Final - VMware Workstation 15 Player (Non-commercial use only)
virustotal.com/gui/file/07d07be27759f9f3f3cec164cbc001cc87275fb36f249c0399d
07d07be27759f9f3f3cec164cbc001cc87275fb36f249c0399d09ac18237a2d2
5 engines detected this file
07d07be27759f9f3f3cec164cbc001cc87275fb36f249c0399d09ac18237a2d2
malicious.exe
1.08 MB
DETECTION
DETAILS
BEHAVIOR
COMMUNITY
Basic Properties
MD5 9c4748f13af3fb386cee15e998f51bb1
SHA-1 5c06a2927a2576d840fd13ebff3f31089e427d48
SHA-256 07d07be27759f9f3f3cec164cbc001cc87275fb36f249c0399d09ac18237a2d2
Vhash 01603e0f7d1bz6017z15z13z1fz
Authentihash e50ac083d5244ab0d2613d24436b33827c8a4419d03f1d24b49c18ed99701b62
Imphash 9d129af3a77b645161a552556901cd9b
SSDEEP 24576:kkeUFPfn2sMEHAnomM46ITXR0VUTeV0x32p2iqBnMrzAqDEF3vAFxEhyER:jeyn2MHqFMrITXqVU/gMiqBnckI24R
File type Win32 EXE
Magic PE32 executable for MS Windows (GUI) Intel 80386 32-bit
File size 1.08 MB (1133530 bytes)
F-PROT packer Unicode, appended, 7Z, UPX
PEiD packer UPX 2.90 [LZMA] -> Markus Oberhumer, Laszlo Molnar & John Reiser
History
Creation Time 2016-10-04 15:12:31
First Submission 2020-05-06 13:39:44
Last Submission 2020-08-17 14:46:06
Last Analysis 2020-08-17 02:50:47
Names
Search Windows



b. Using the sysinternal utility, **strings.exe**, look for any hidden clues that may have been written into the executable code of the file, **malicious.exe**.



c. Scroll to the end of the executable file to reveal a clue.

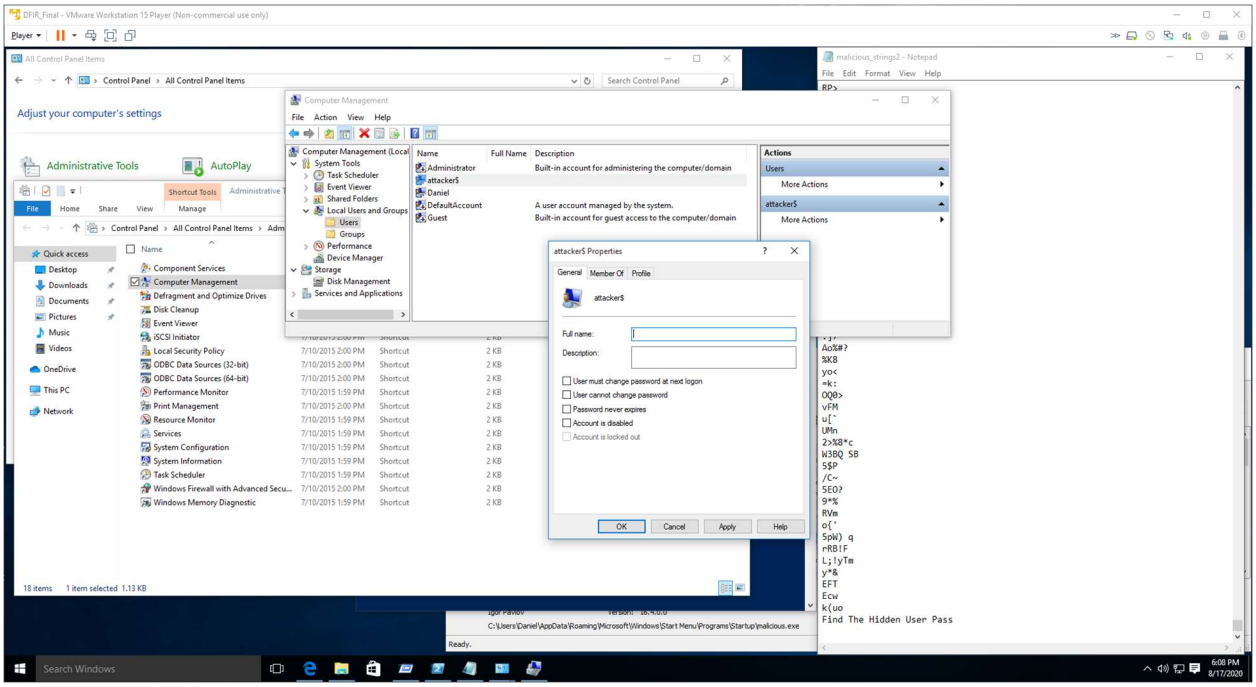


9) Follow the hint in the executable.

a. The hint in the executable file, **malicious.exe**, is revealed at the very last line of code. It says, “Find the hidden User Pass”.

b. The **malicious.exe** file could be dangerous to the victim PC by leaving it vulnerable to external connection at startup. If so, there would need to be a user to log in with. A malicious actor would make the logon user a hidden user. Check the Users on the Win10 VM.

c. From the Windows Control Panel, select \ All Control Panel Items \ Administrative Tools \ Computer Management \ System Tools \ Local Users and Groups



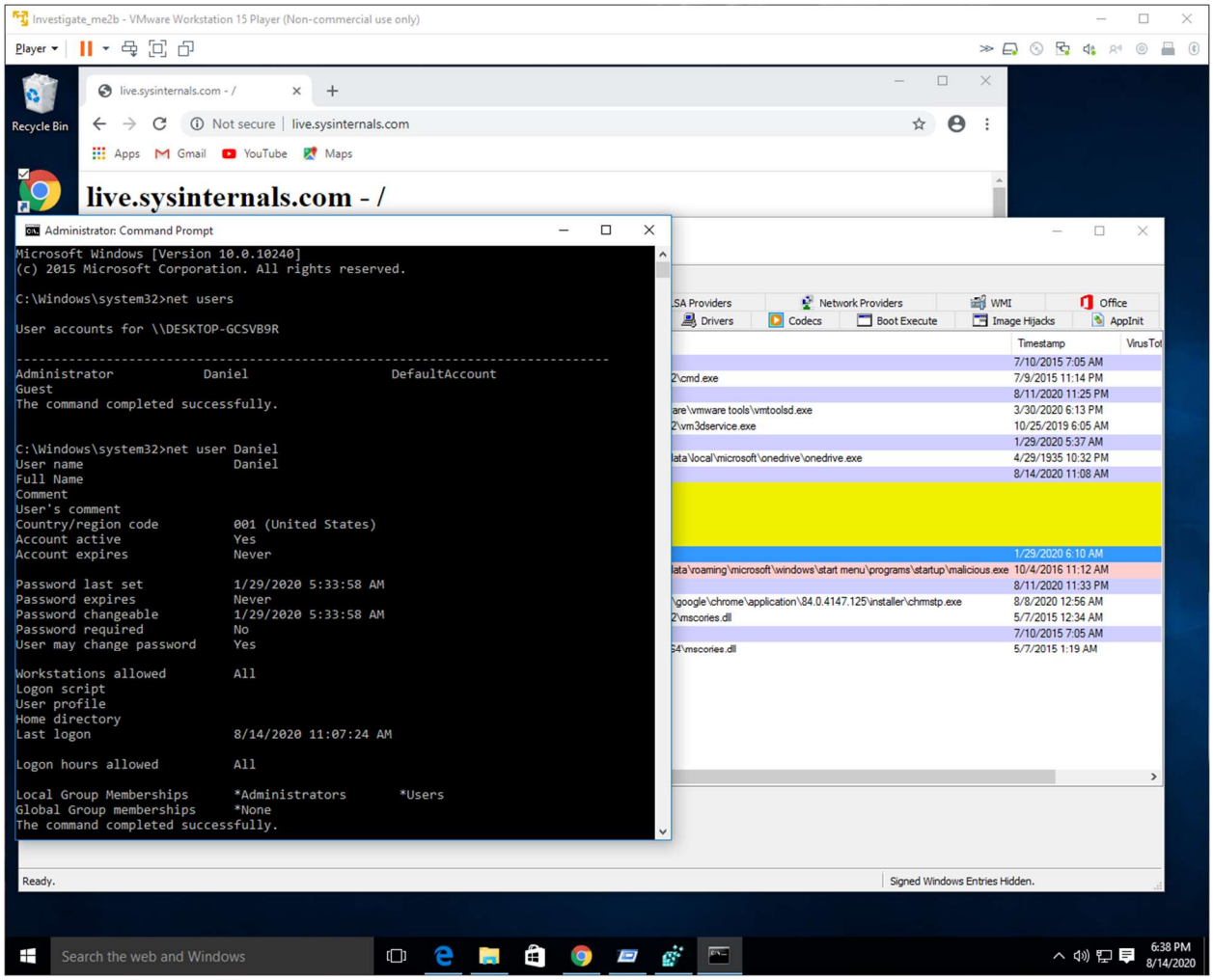
d. In the Users directory, there is a User called “attacker$”. This name does NOT appear as one of the logon options when the PC boots.

i. In cmd.exe, display users with command **net users**. This command reveals only known and authorized users, Administrator, DefaultAccount, Guest, and Daniel.

Syntax: net users

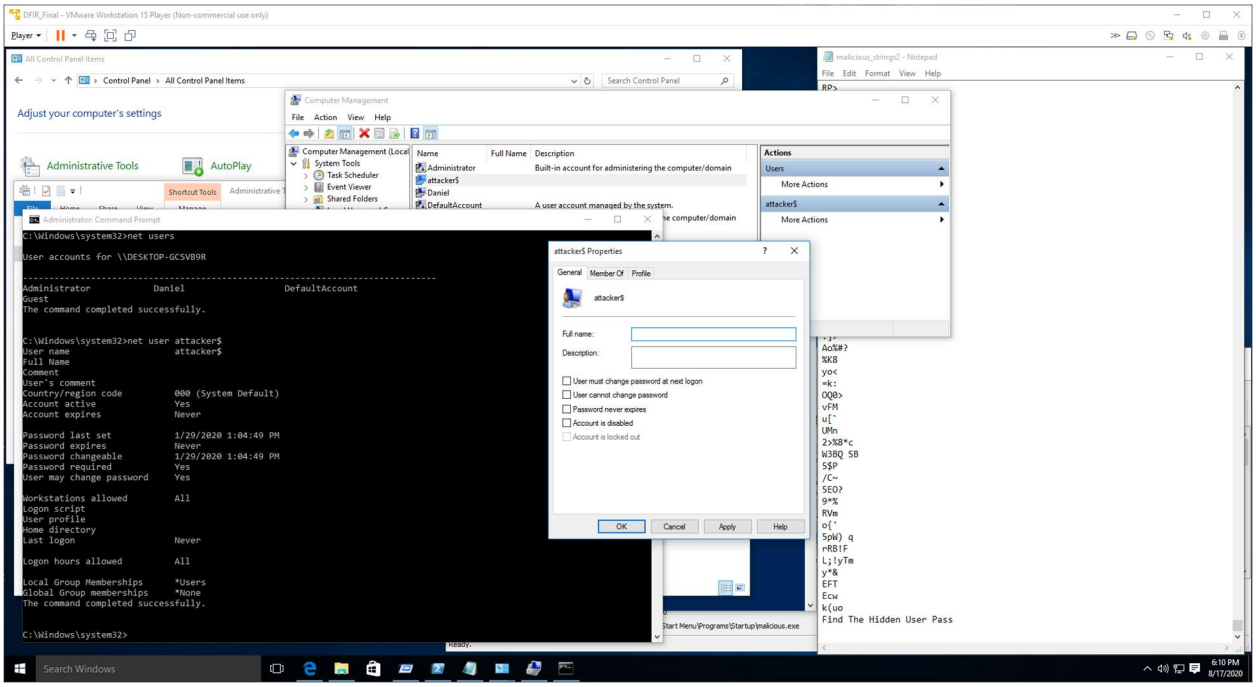
ii. To view the permissions and Group of a particular user, use the command **net user <username>**

Syntax: net user Daniel

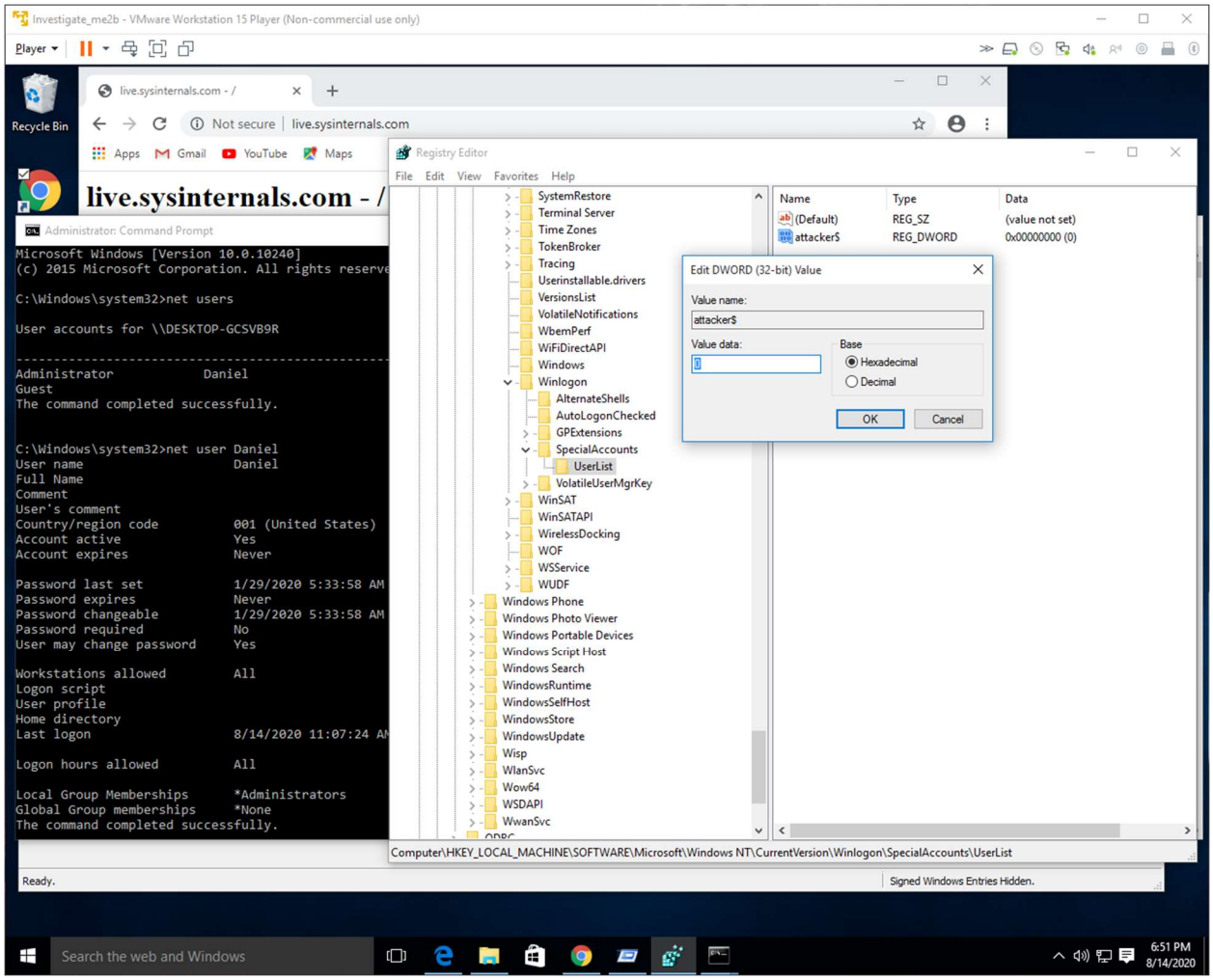


e. Even for a hidden user, the permissions and Group command will display the hidden user's permissions.

Syntax: net user attacker$

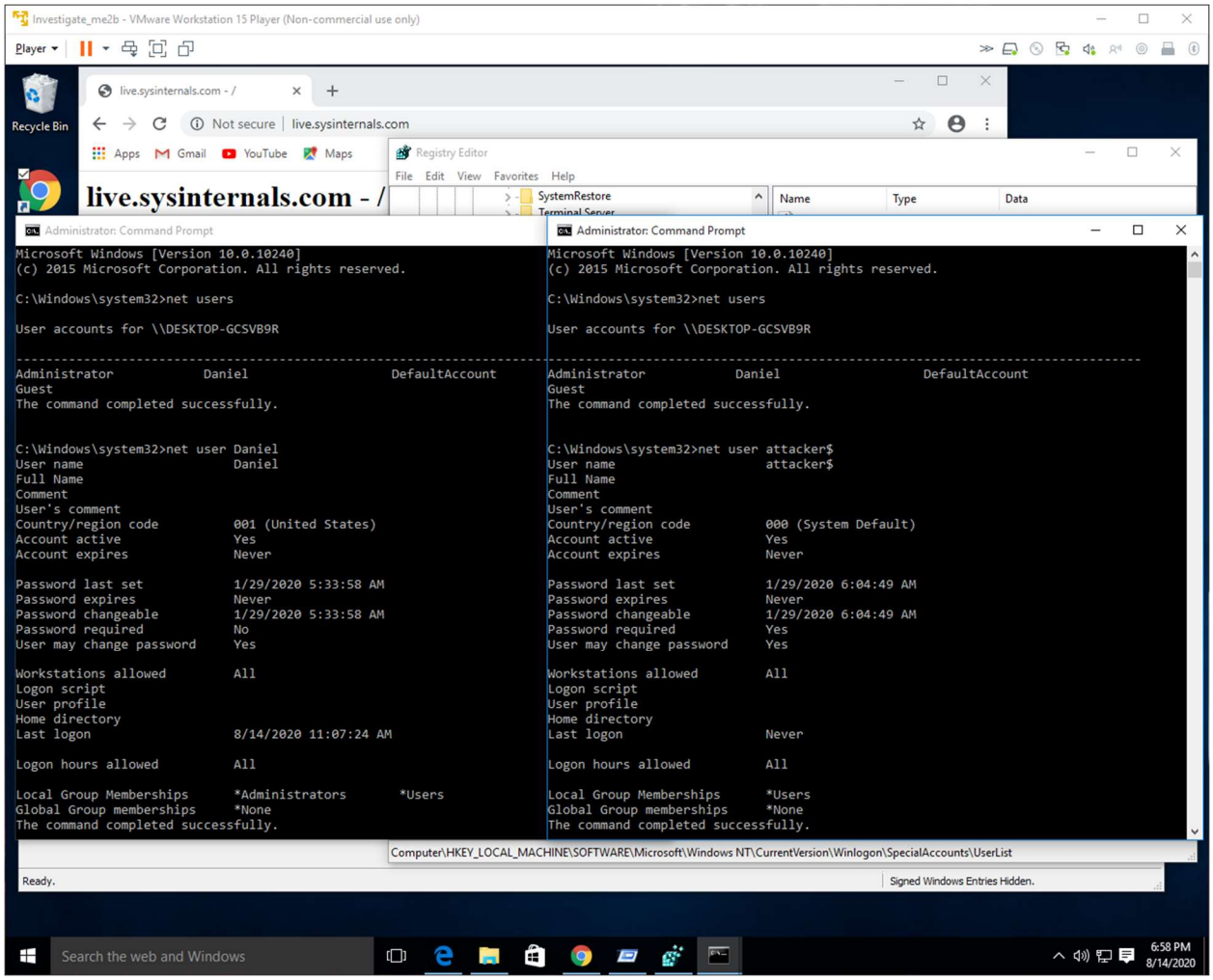


f. To see if there are any Special Accounts with hidden users, launch the Registry Editor, **RegEdit.exe**, and navigate to the WinLogon directory.
   i. A User in the Special Accounts has a $ character in its name.
   ii. Hiding special accounts from the logon screen is a built-in feature of Windows, but it isn't enabled by default. To enable it, use **RegEdit.exe** and add 2 new registry keys and a DWORD.
   iii. To hide the Special Account User from the Logon screen, in RegEdit.exe, change the Value Data of the Special Account User to 0.



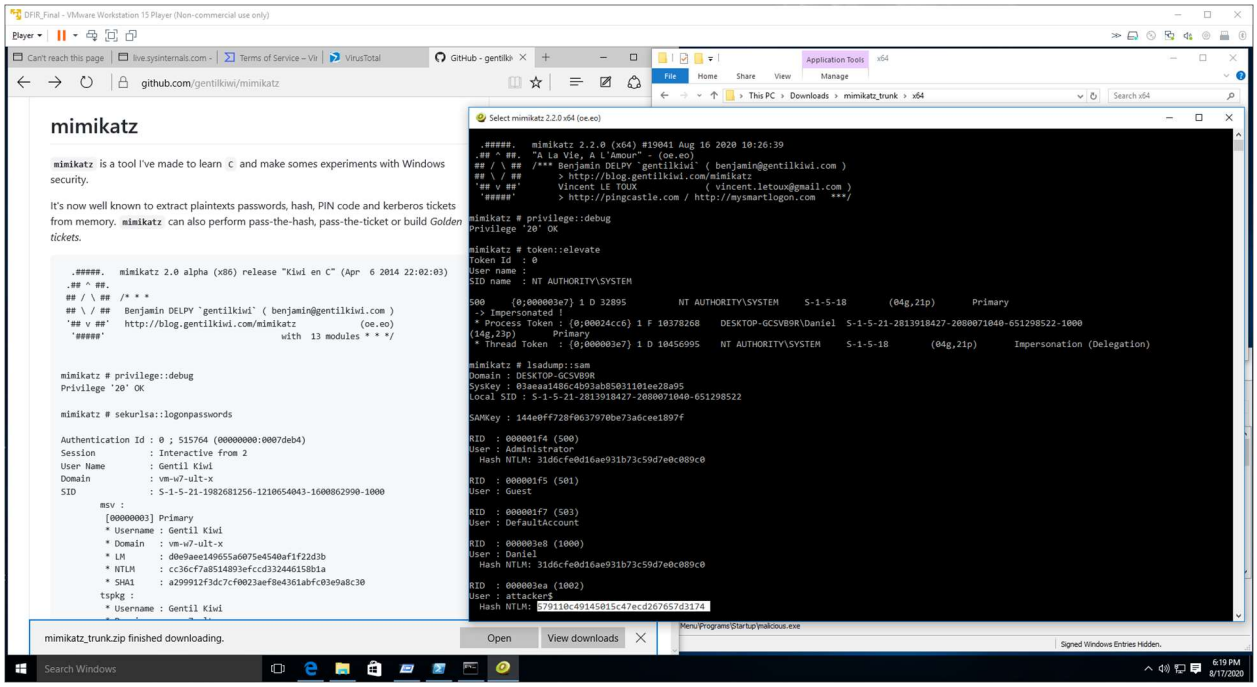
g. Now that we know there is a Special Account and a User in the UserList with a name of "attacker$" we can examine the permissions and groups of this hidden user. In cmd,exe, run the command **net user attacker$**.

Syntax: net user attacker$



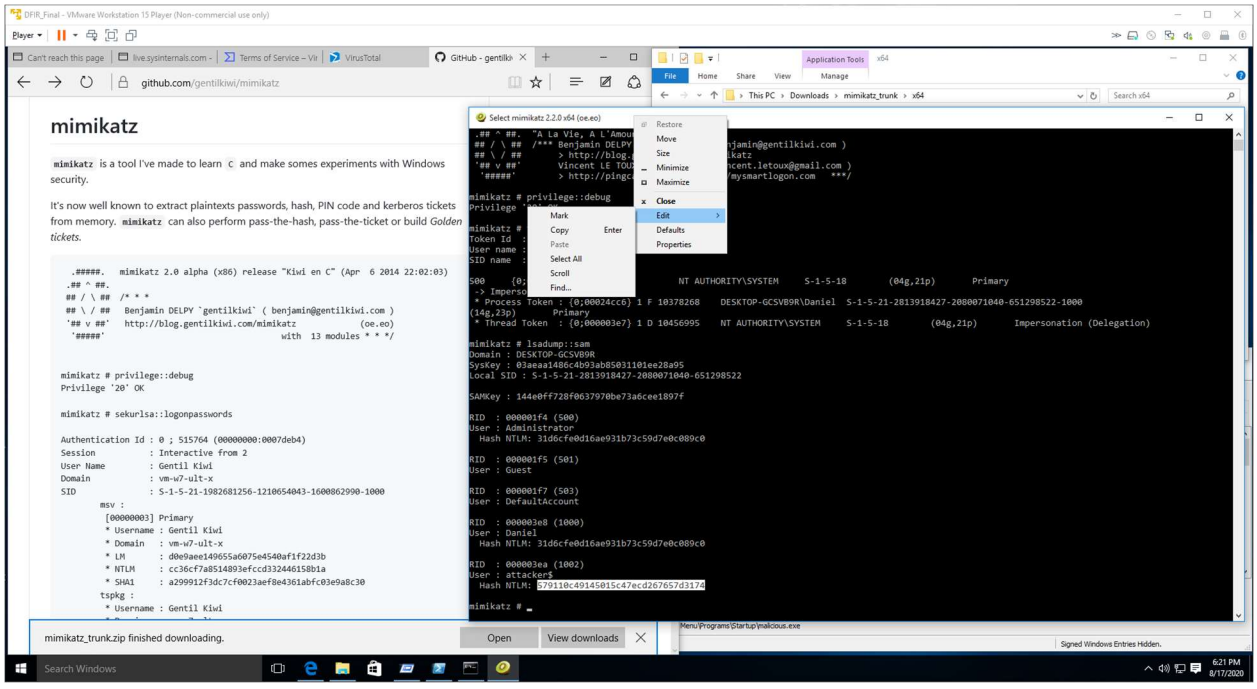
h. We can see that the hidden user, attacker$, requires a password. We don't know what the password is. We still need to find the password of this hidden user, attacker$.

i. Download and install Mimikatz onto the VM. Mimikatz will be able to scan the PC for all Users and display the User names and the Hashes of their passwords.

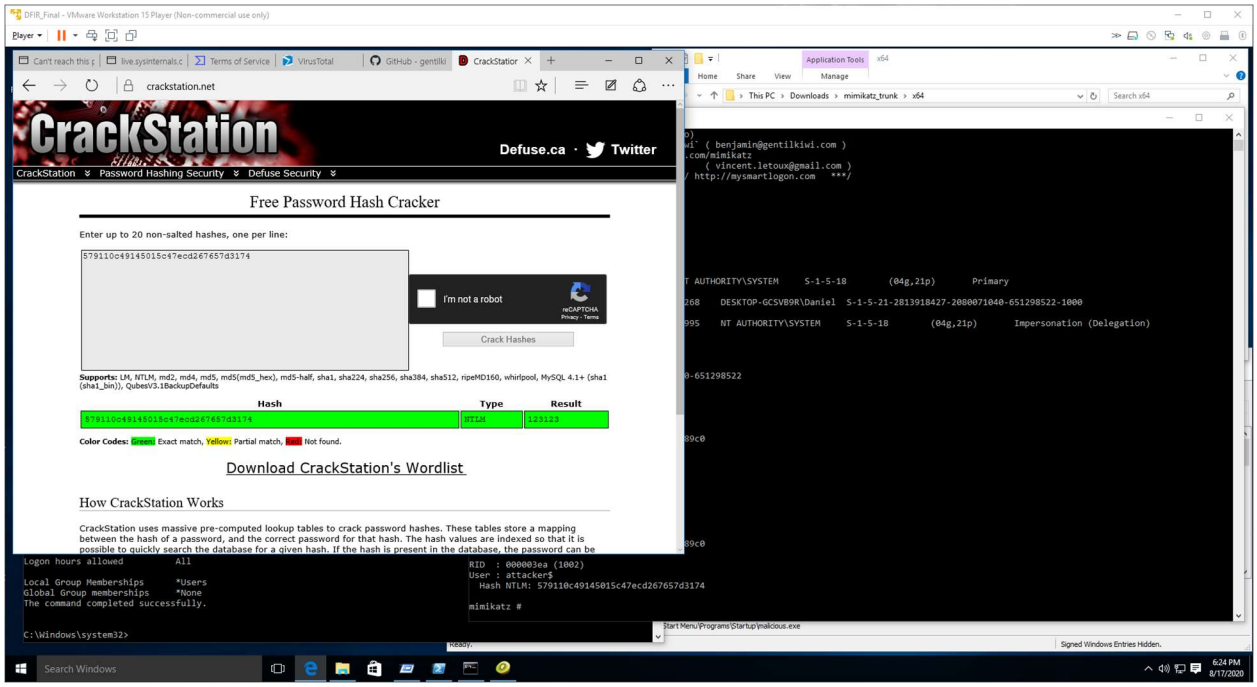


j. Find the entry for the User, attacker$, and select the hash of the password.

k. Copy the hash of the password.



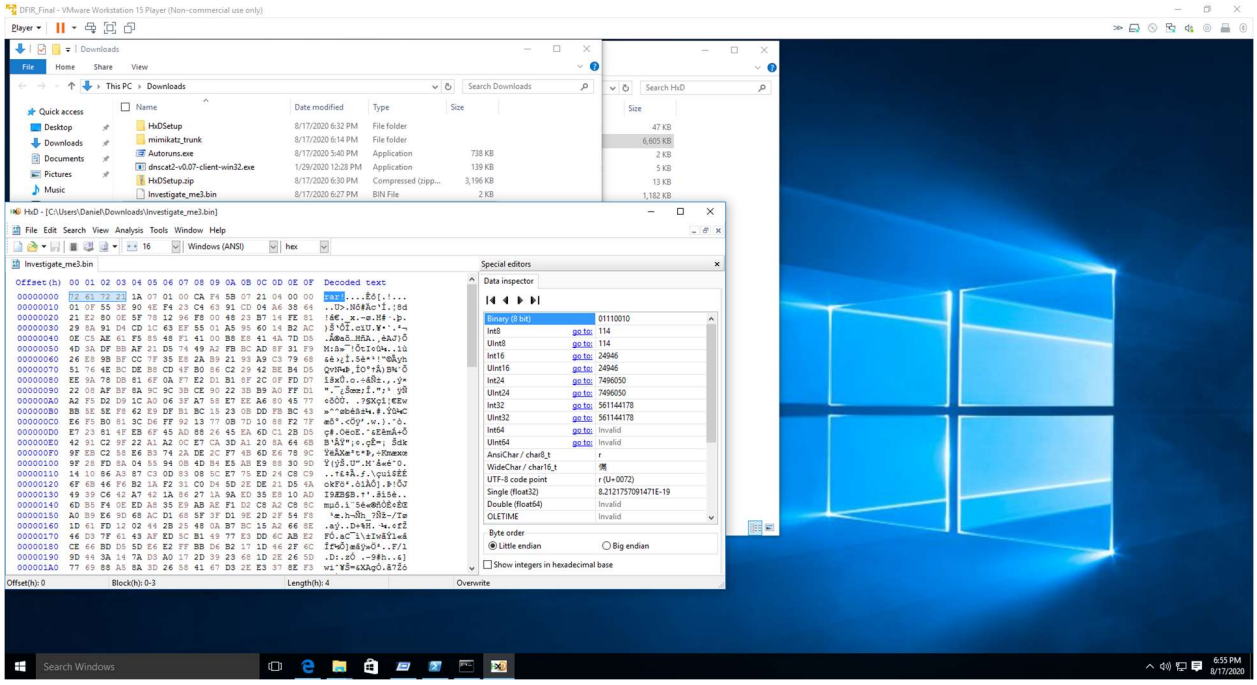
l. Using an online hash cracking utility, **CrackStation.net**, paste the hash into the lookup field and reveal the password in plain text.



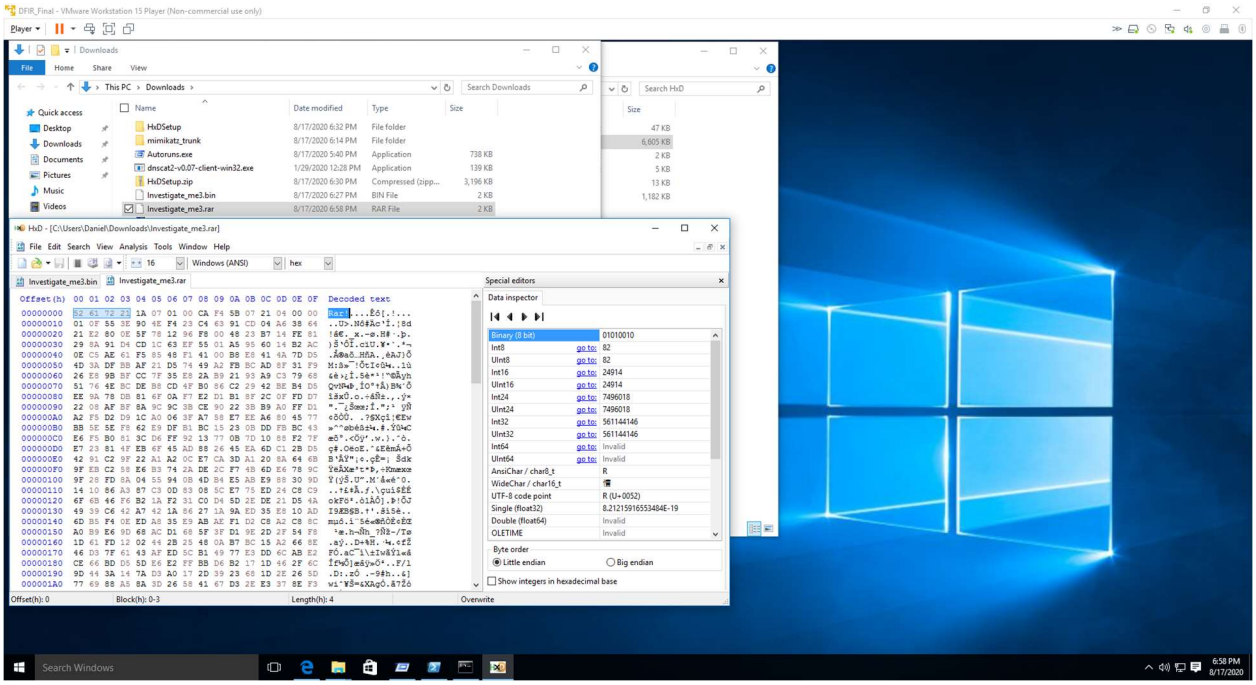
m. The Password is revealed to be: **123123**.

10) Complete the investigation of the second part

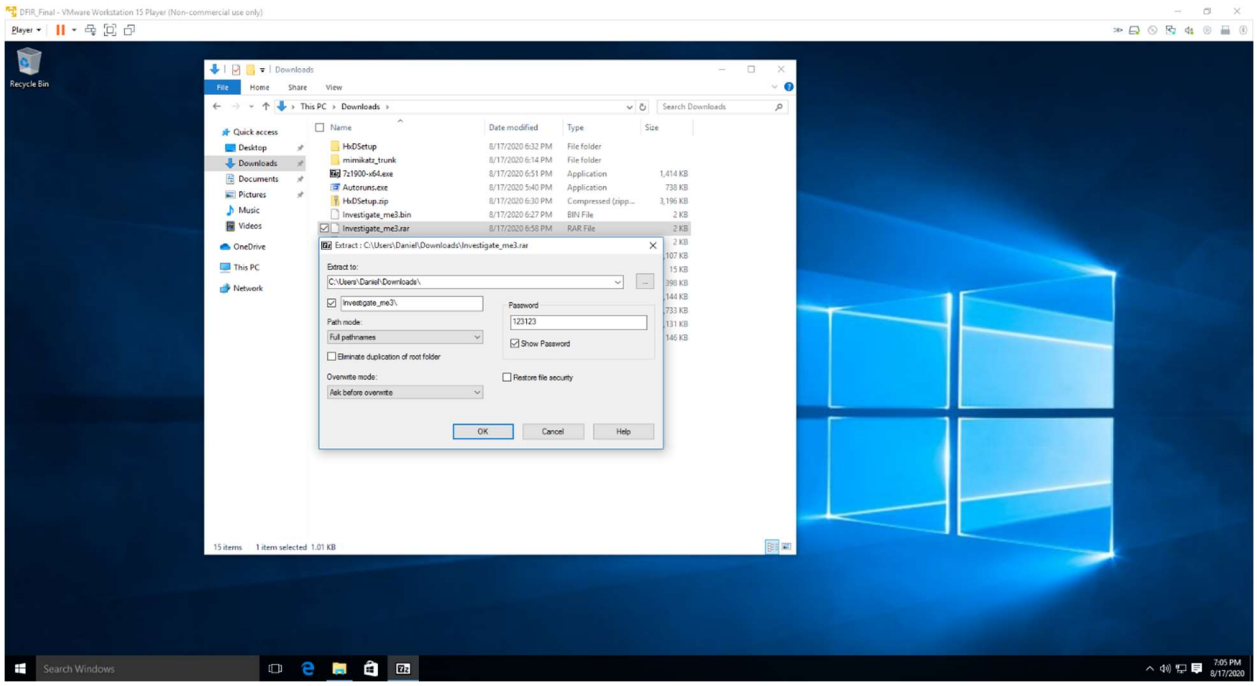
a. Examine the final file, **Investigate_me3.bin**.
b. In order to read a binary file, it may be helpful to examine it with a Hexadecimal to Binary editor, like HxD.
c. Download and install HxdD onto the VM. Open the file, **Investigate_me3.bin**.



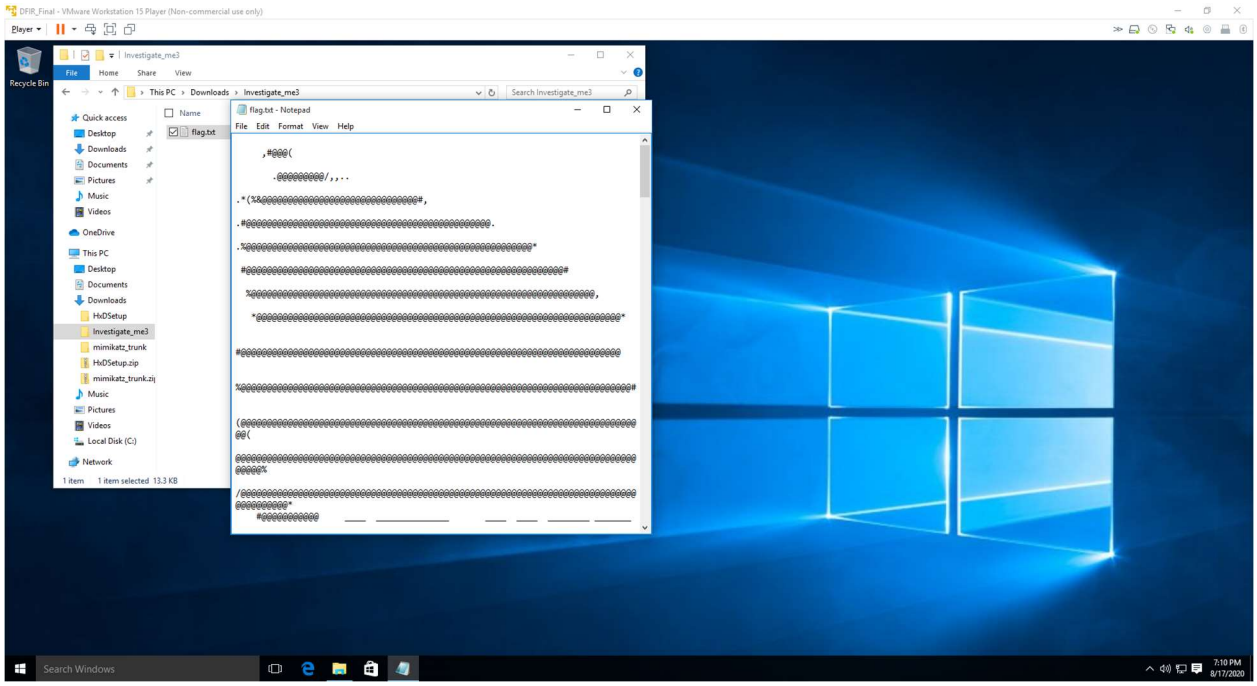
d. Compare the file header info against standard header info.
    i. The filename ends with the suffix .bin, however, the file header reads rar.
    ii. Change the filename from .bin to .rar
        1. I copied the file to the same directory but changed the suffix of the copied file. This preserves the original file and allows manipulation of the copied file.



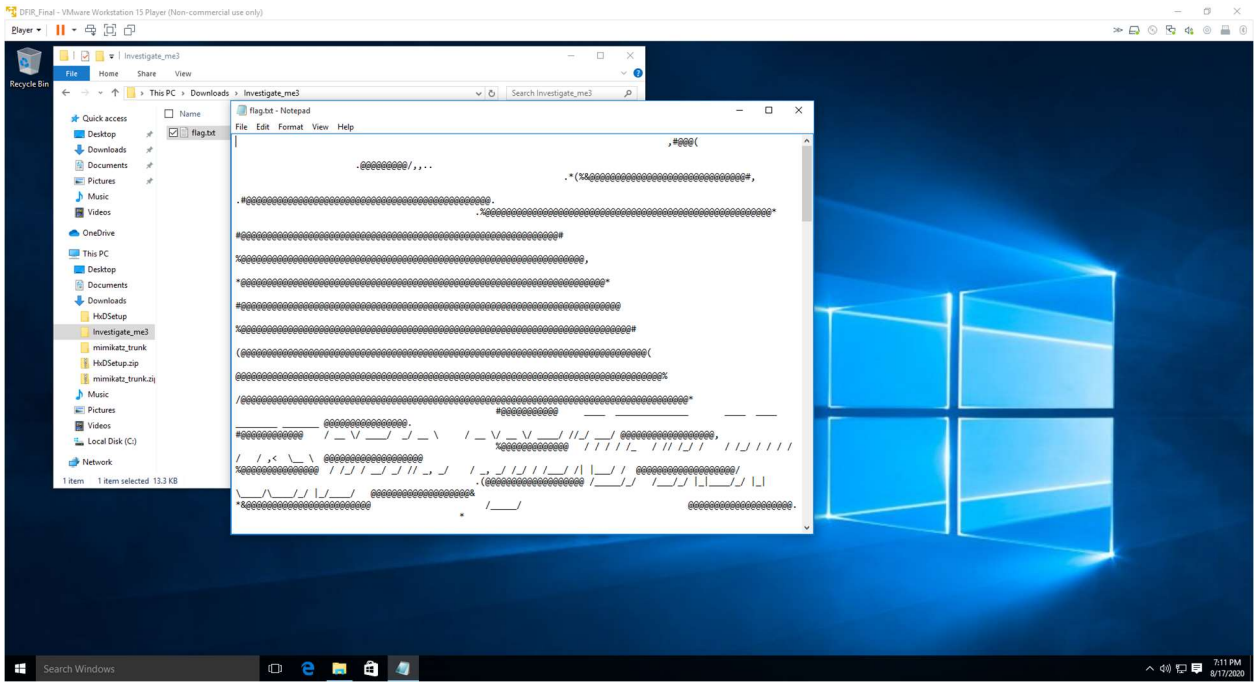
e. The copied file, **Investigate_me3.rar**, will not uncompress due to a file type error.
f. Further examination of the hexadecimal values of this file's header reveals that a normal .rar file would have a hexadecimal value of 52 61 72 21. However, this file's header value is 72 61 72 21. The difference is that 52 is a Capital R, and 72 is a lowercase r.
    i. Change the hexadecimal value to 52 61 72 21 and save the edited file.
g. Try to uncompress the newly-saved file, **Investigate_me3.rar**.



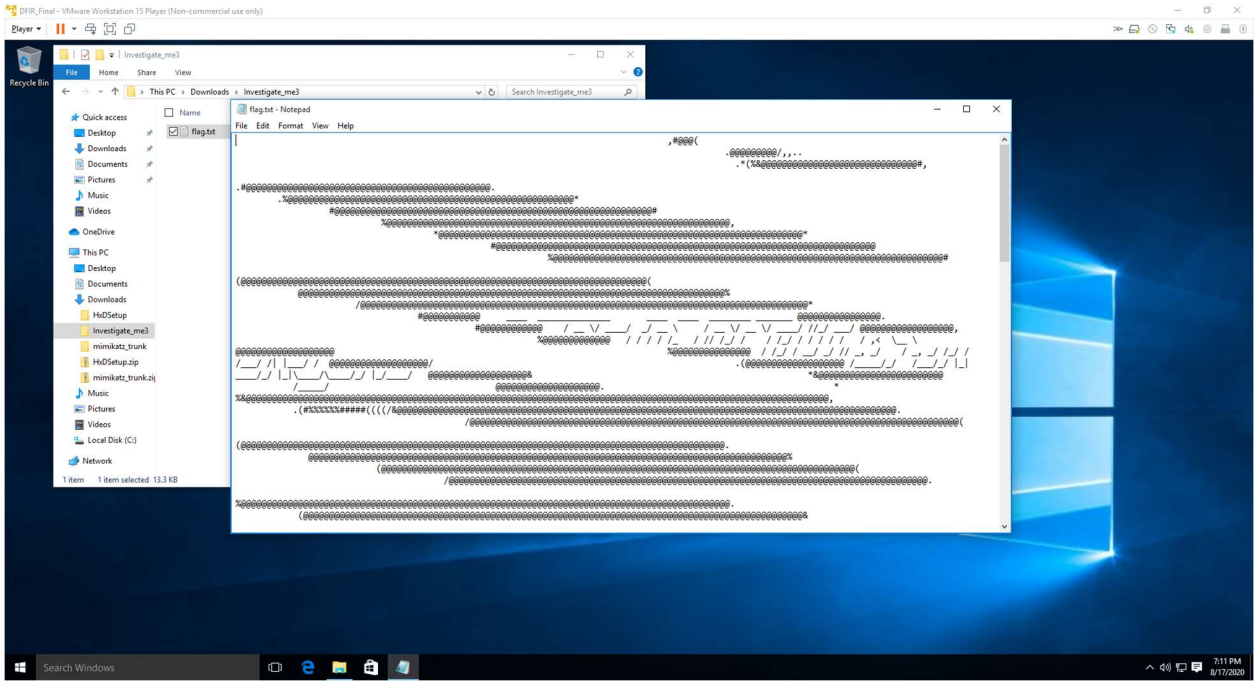
h. Using the password captured from the previous step, enter that password, **123123**, to uncompress this .rar file.
i. The uncompressed file reveals a folder, named **Investigate_me3**, with a text file inside it, called **flag.txt**.
j. Open the file, **flag.txt**, using a text editor like notepad.exe.



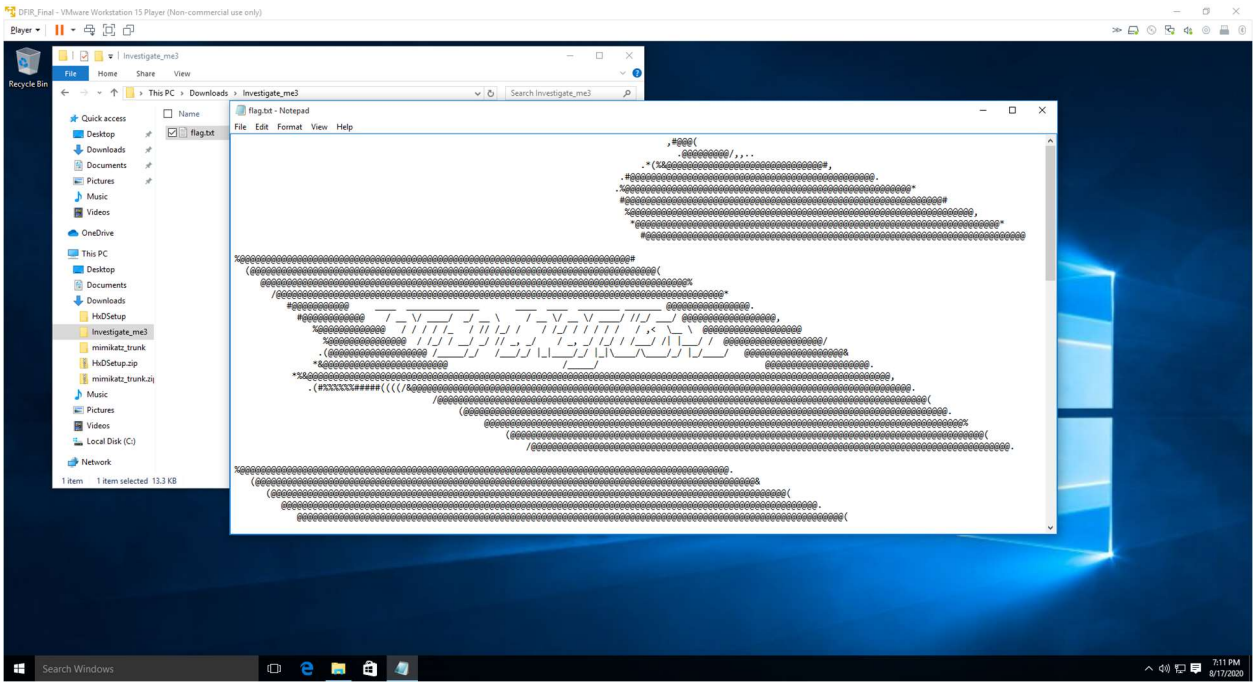
i. The contents of the text file seem like gibberish.
ii. With WordWrap enabled in notepad, expand the text window a little wider.



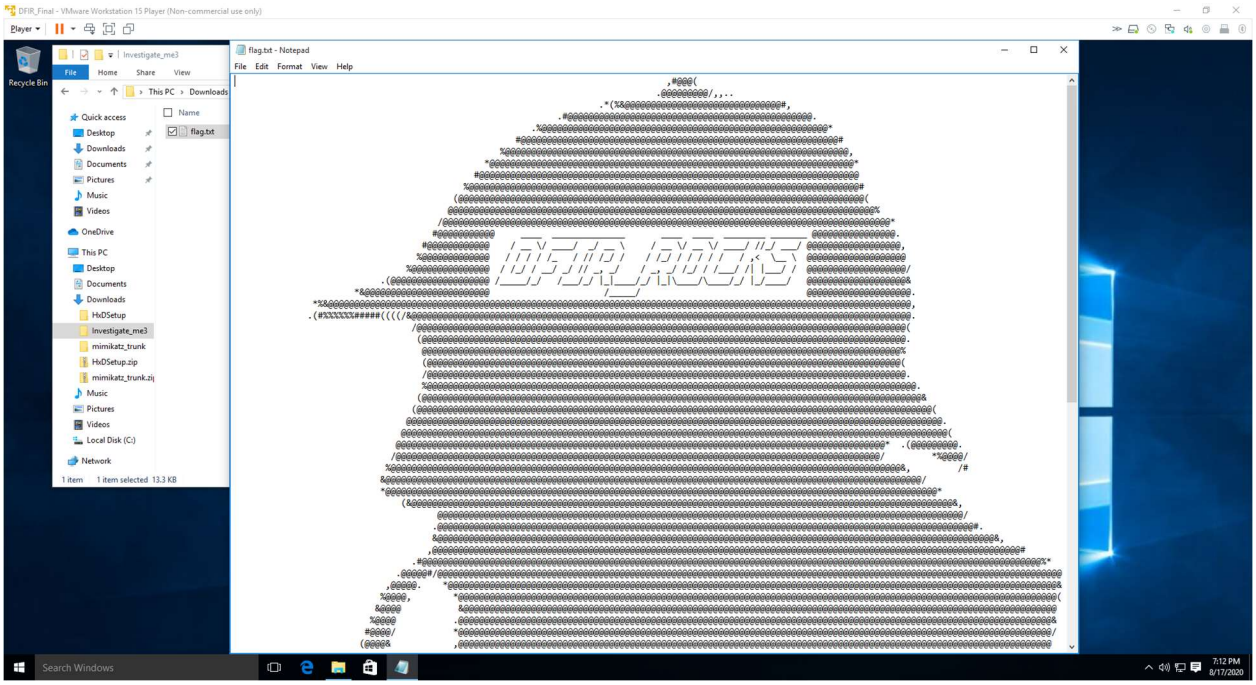
iii. Maybe a little wider.



iv. Maybe a little more.



v. It's starting to look like something. Even wider.



It appears Mr Holmes has led us to the final conclusion.

The End, Mr Watson.