SPLUNK

Conclusion:

In this fun exercise, as a SOC Analyst, we have investigated a cyber-attack where the attacker had defaced a website 'imreallynotbatman.com' of the Wayne Enterprise. We mapped the attacker's activities into the 7 phases of the Cyber Kill Chain. Let us recap everything we have found so far:

Reconnaissance Phase:

We first looked at any reconnaissance activity from the attacker to identify the IP address and other details about the adversary.

Findings:

IP Address 40.80.148.42 was found to be scanning our webserver.

The attacker was using Acunetix as a web scanner.

Exploitation Phase:

We then looked into the traces of exploitation attempts and found brute-force attacks against our server, which were successful.

Findings:

Brute force attack originated from IP 23.22.63.114.

The IP address used to gain access: 40.80.148.42

142 unique brute force attempts were made against the server, out of which one attempt was successful

Installation Phase:

Next, we looked at the installation phase to see any executable from the attacker's IP Address uploaded to our server.

Findings:

A malicious executable file 3791.exe was observed to be uploaded by the attacker.

We looked at the sysmon logs and found the MD5 hash of the file.

Action on Objective:

After compromising the web server, the attacker defaced the website.

Findings:

We examined the logs and found the file name used to deface the webserver.

Weaponization Phase:

We used various threat Intel platforms to find the attacker's infrastructure based on the following information we saw in the above activities.

Information we had:

Domain: prankglassinebracket.jumpingcrab.com

IP Address: 23.22.63.114

Findings:

Multiple masquerading domains were found associated with the attacker's IPs.

An email of the user Lillian.rose@po1s0n1vy.com was also found associated with the attacker's IP address.

Deliver Phase:

In this phase, we again leveraged online Threat Intel sites to find malware associated with the adversary's IP address, which appeared to be a secondary attack vector if the initial compromise failed.

Findings:

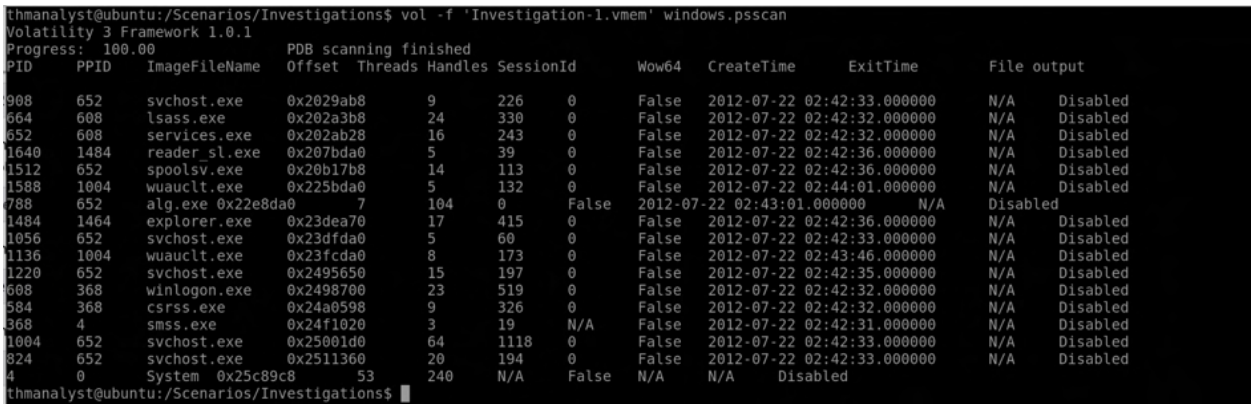
A malware name MirandaTateScreensaver.scr.exe was found associated with the adversary.

MD5 of the malware was c99131e0169171935c5ac32615ed6261

Volatility

Volatility is a free memory forensics tool developed and maintained by Volatility Foundation, commonly used by malware and SOC analysts within a blue team or as part of their detection and monitoring solutions. Volatility is written in Python and is made up of python plugins and modules designed as a plug-and-play way of analyzing memory dumps.

Simple example of using a plugin.

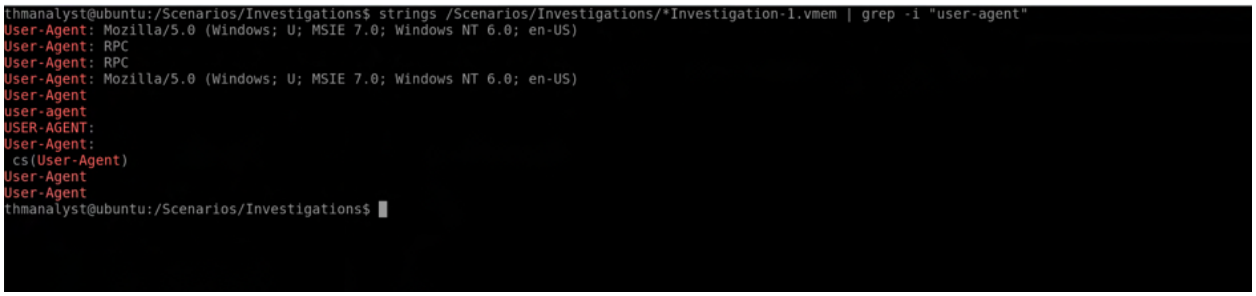


```
thmanalyst@ubuntu:/Scenarios/Investigations$ vol -f 'Investigation-1.vmem' windows.psscan
Volatility 3 Framework 1.0.1
Progress:  100.00               PDB scanning finished
PID     PPID    ImageFileName   Offset  Threads Handles SessionId       Wow64   CreateTime      ExitTime        File output

908     652     svchost.exe     0x2029ab8       9       226     0       False   2012-07-22 02:42:33.000000      N/A     Disabled
664     608     lsass.exe       0x202a3b8       24      330     0       False   2012-07-22 02:42:32.000000      N/A     Disabled
652     608     services.exe    0x202ab28       16      243     0       False   2012-07-22 02:42:32.000000      N/A     Disabled
1640    1484    reader_sl.exe   0x207bda0       5       39      0       False   2012-07-22 02:42:36.000000      N/A     Disabled
1512    652     spoolsv.exe     0x20b17b8       14      113     0       False   2012-07-22 02:42:36.000000      N/A     Disabled
1588    1004    wuauclt.exe     0x225bda0       5       132     0       False   2012-07-22 02:44:01.000000      N/A     Disabled
788     652     alg.exe 0x22e8da0       7       104     0       False   2012-07-22 02:43:01.000000      N/A     Disabled
1484    1464    explorer.exe    0x23dea70       17      415     0       False   2012-07-22 02:42:36.000000      N/A     Disabled
1056    652     svchost.exe     0x23dfda0       5       60      0       False   2012-07-22 02:42:33.000000      N/A     Disabled
1136    1004    wuauclt.exe     0x23fcda0       8       173     0       False   2012-07-22 02:43:46.000000      N/A     Disabled
1220    652     svchost.exe     0x2495650       15      197     0       False   2012-07-22 02:42:35.000000      N/A     Disabled
608     368     winlogon.exe    0x2498700       23      519     0       False   2012-07-22 02:42:32.000000      N/A     Disabled
584     368     csrss.exe       0x24a0598       9       326     0       False   2012-07-22 02:42:32.000000      N/A     Disabled
368     4       smss.exe        0x24f1020       3       19      N/A     False   2012-07-22 02:42:31.000000      N/A     Disabled
1004    652     svchost.exe     0x25001d0       64      1118    0       False   2012-07-22 02:42:33.000000      N/A     Disabled
824     652     svchost.exe     0x2511360       20      194     0       False   2012-07-22 02:42:33.000000      N/A     Disabled
4       0       System  0x25c89c8       53      240     N/A     False   N/A     N/A     Disabled
thmanalyst@ubuntu:/Scenarios/Investigations$
```

Exercise.

For case 1

Most things were straightforward in answering , we just had to use the required plugins however to find the user agent requires a special command which could be confusing.



Case2

In addition, there were outside OSINT that were required to find some of the solutions. Because we were dealing with a suspicious process called **@WanaDecryptor@**, OSINT was required. Some notable sources were virus total and blogs to find the correct mutex for the known indicator of the malware.
(**MsWinZonesCacheCounterMutexA0x0)**

One important thing to note is that this suspicious process is part of the famous malware **wannacry**. Finally to summarize, it's important to recognize the suspicious pid and parent process PID, so that we can trace and link things up.

Velociraptor

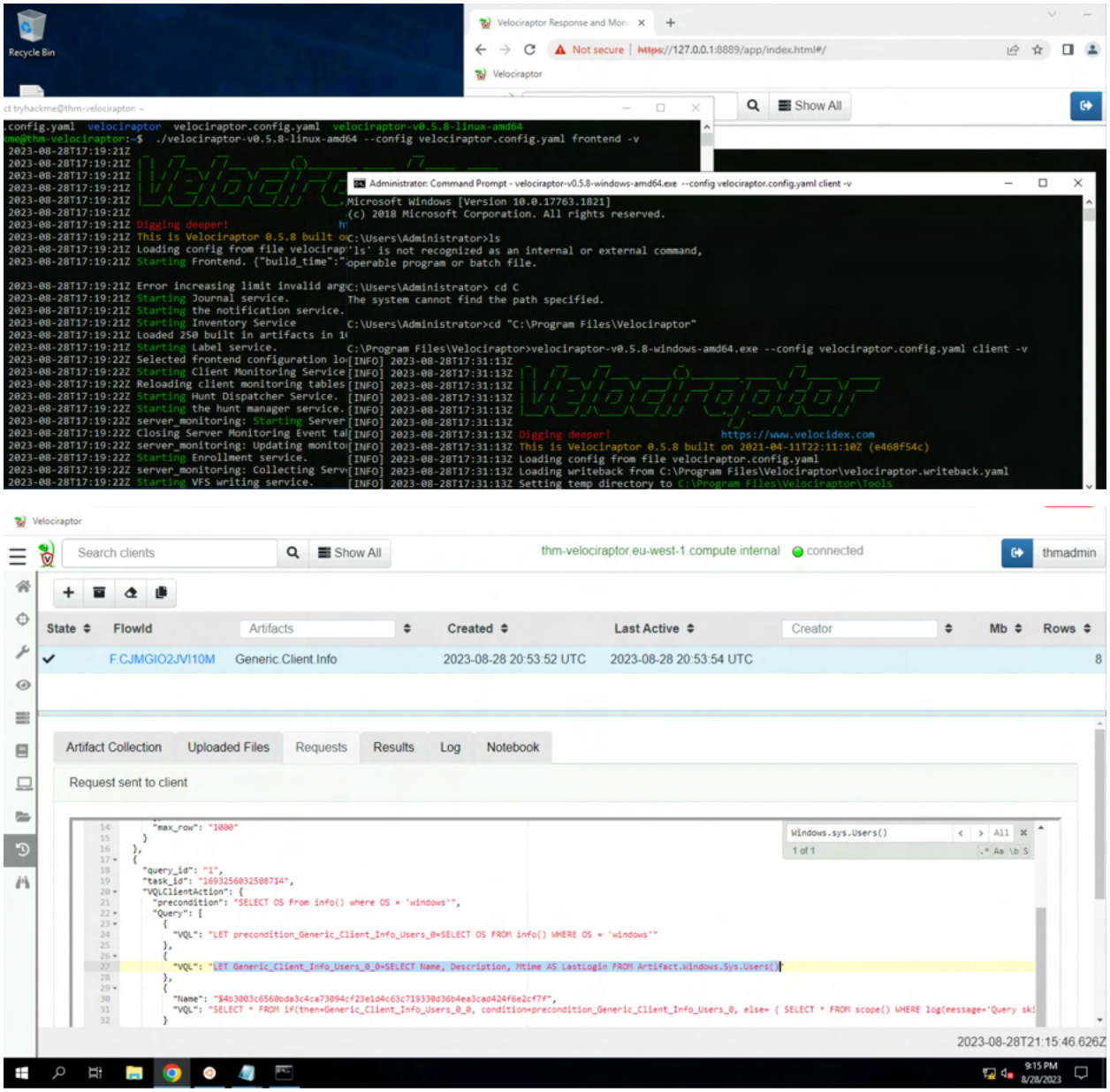
In this room, we will explore Rapid7's newly acquired tool known as Velociraptor.

Per the official Velociraptor documentation, "*Velociraptor is a unique, advanced open-source endpoint monitoring, digital forensic and cyber response platform. It was developed by Digital Forensic and Incident Response (DFIR) professionals who needed a powerful and efficient way to hunt for specific artifacts and monitor activities across fleets of endpoints. Velociraptor provides you with the ability to more effectively respond to a wide range of digital forensic and cyber incident response investigations and data breaches*".

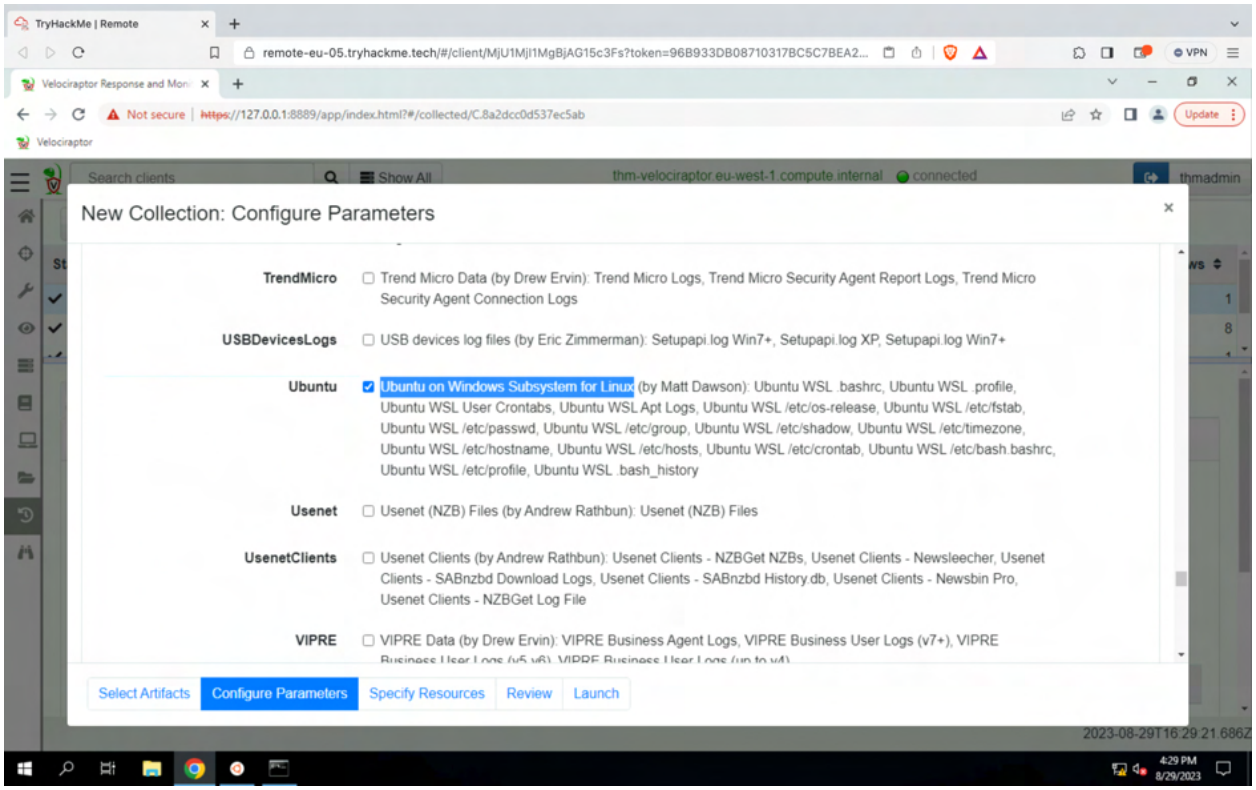
This tool was created by Mike Cohen, a former Google employee who worked on tools such as GRR (GRR Rapid Response) and Rekall (Rekall Memory Forensic Framework). Mike joined Rapid7's Detection and Response Team and continues to work on improving Velociraptor. At the date of this entry, the latest release for Velociraptor is 0.6.3.

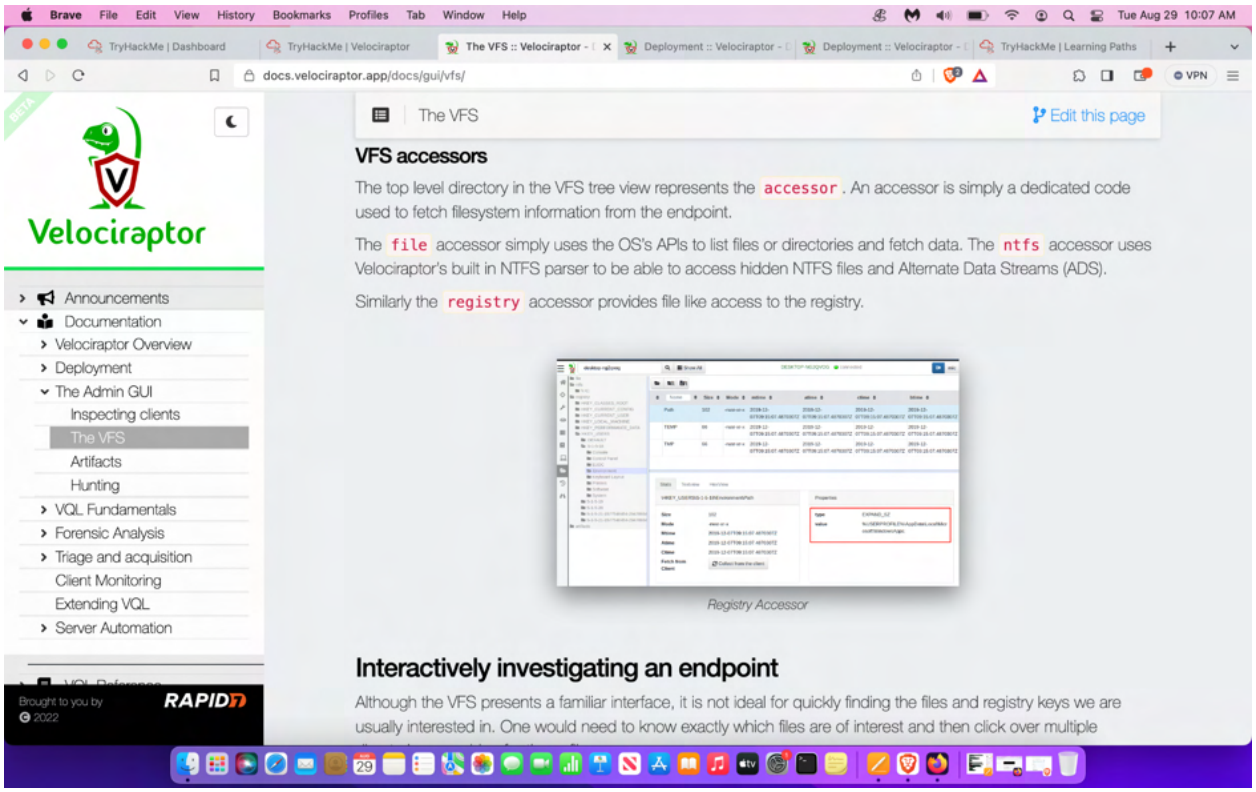
Learning Objectives

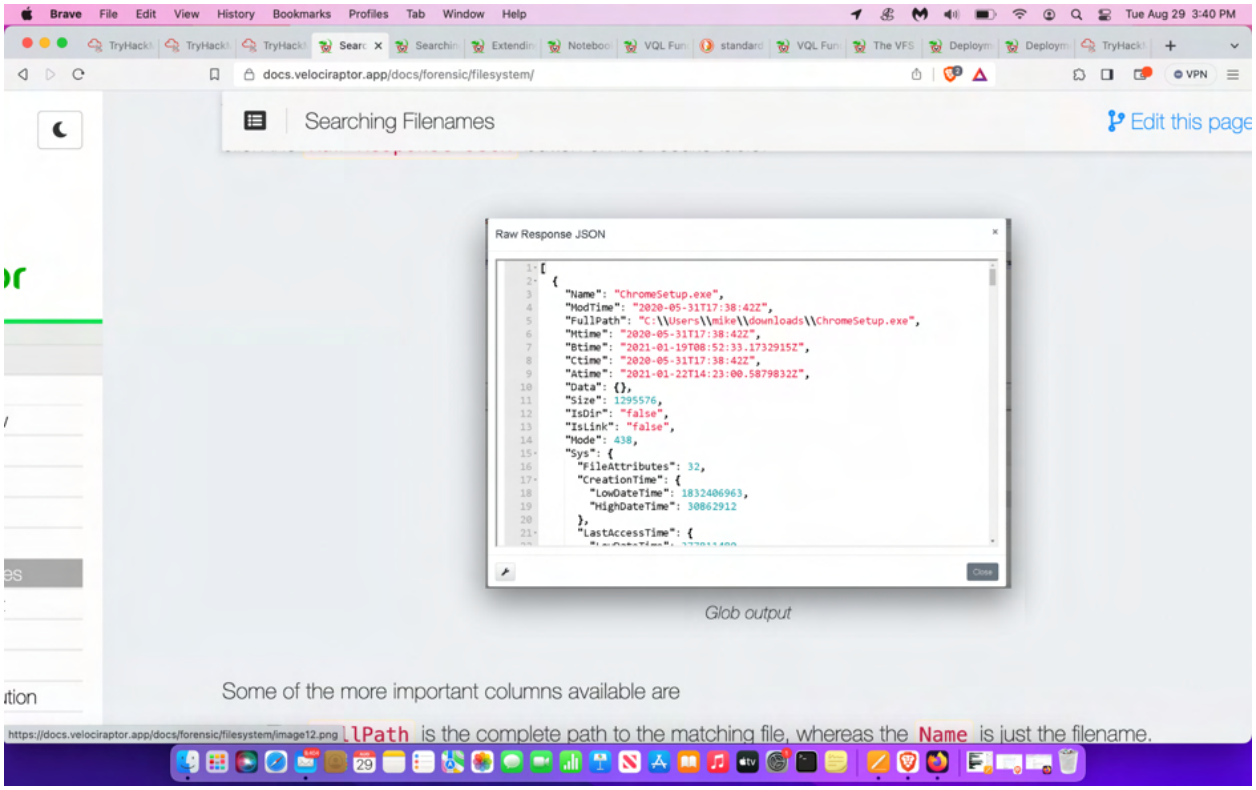
- Learn what is Velociraptor
- Learn how to interact with agents and create collections
- Learn how to interact with the virtual file system
- Learn what is VQL and how to create basic queries
- Use Velociraptor to perform a basic hunt

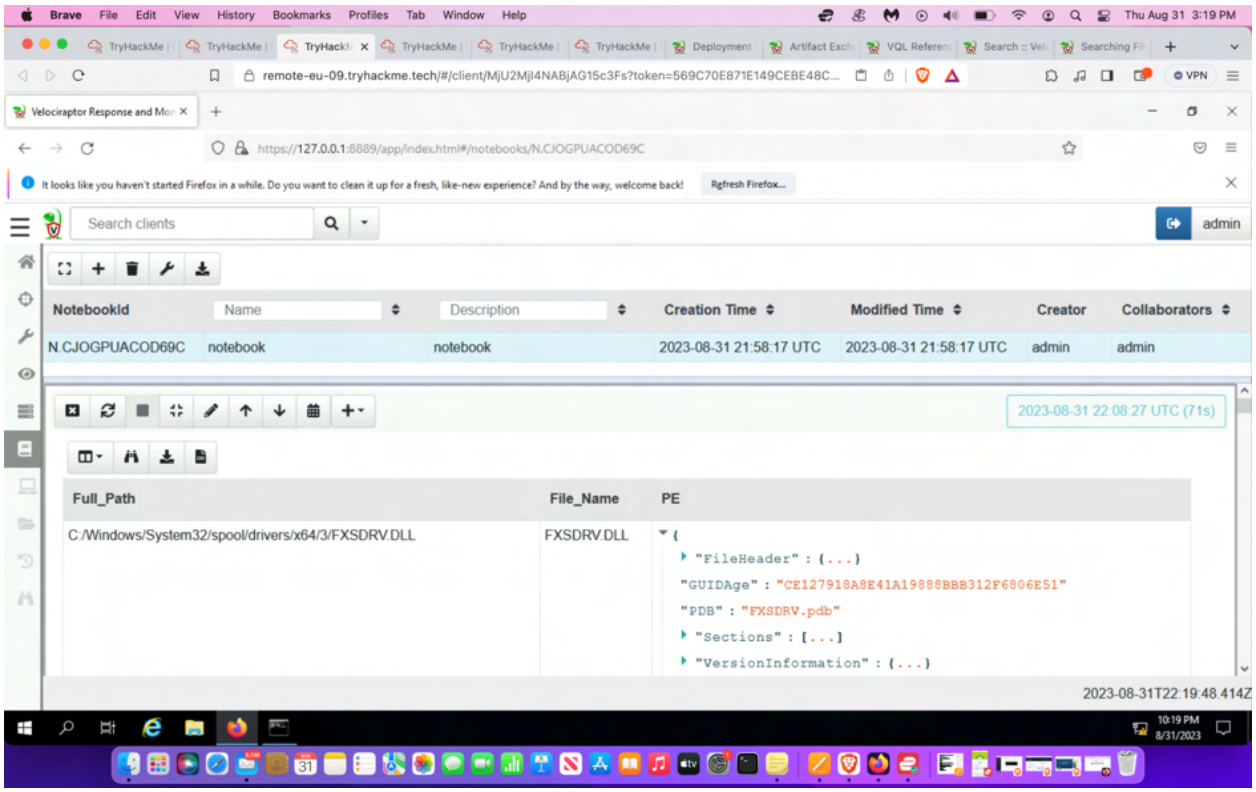

Recycle Bin
Velociraptor Response and Mon
Not secure
https://127.0.0.1:8889/app/index.html#/
Velociraptor
Show All
Administrator: Command Prompt - velociraptor-v0.5.8-windows-amd64.exe --config velociraptor.config.yaml client -v
Microsoft Windows [Version 10.0.17763.1821]
(c) 2018 Microsoft Corporation. All rights reserved.
C:\Users\Administrator>ls
'ls' is not recognized as an internal or external command,
operable program or batch file.
C:\Users\Administrator> cd C
The system cannot find the path specified.
C:\Users\Administrator>cd "C:\Program Files\Velociraptor"
C:\Program Files\Velociraptor>velociraptor-v0.5.8-windows-amd64.exe --config velociraptor.config.yaml client -v
[INFO] 2023-08-28T17:31:13Z
[INFO] 2023-08-28T17:31:13Z Digging deeper! https://www.velocidex.com
[INFO] 2023-08-28T17:31:13Z This is Velociraptor 0.5.8 built on 2021-04-11T22:11:10Z (e468f54c)
[INFO] 2023-08-28T17:31:13Z Loading config from file velociraptor.config.yaml
[INFO] 2023-08-28T17:31:13Z Loading writeback from C:\Program Files\Velociraptor\velociraptor.writeback.yaml
[INFO] 2023-08-28T17:31:13Z Setting temp directory to C:\Program Files\Velociraptor\Tools

Velociraptor
Search clients
Show All
thm-velociraptor.eu-west-1.compute.internal
connected
thmadmin
State
FlowId
Artifacts
Created
Last Active
Creator
Mb
Rows
F.CJMGIO2JVI10M
Generic.Client.Info
2023-08-28 20:53:52 UTC
2023-08-28 20:53:54 UTC
8
Artifact Collection
Uploaded Files
Requests
Results
Log
Notebook
Request sent to client
"max_row": "1000"
"query_id": "1",
"task_id": "1693256032508714",
"VQLClientAction": {
"precondition": "SELECT OS From info() where OS = 'windows'",
"Query": [
"VQL": "LET precondition_Generic_Client_Info_Users_0=SELECT OS FROM info() WHERE OS = 'windows'"
"VQL": "LET Generic_Client_Info_Users_0_0=SELECT Name, Description, Mtime AS LastLogin FROM Artifact.Windows.Sys.Users()"
"Name": "$4b3003c6560bda3c4ca73094cf23e1d4c63c719330d36b4ea3cad424f6e2cf7f",
"VQL": "SELECT * FROM if(then=Generic_Client_Info_Users_0_0, condition=precondition_Generic_Client_Info_Users_0, else= { SELECT * FROM scope() WHERE log(message='Query sk
Windows.sys.Users()
1 of 1
All
2023-08-28T21:15:46.626Z
9:15 PM
8/28/2023


TryHackMe | Remote
remote-eu-05.tryhackme.tech/#/client/MjU1MjI1MgBjAG15c3Fs?token=96B933DB08710317BC5C7BEA2...
Velociraptor Response and Mon
Not secure https://127.0.0.1:8889/app/index.html?#/collected/C.8a2dcc0d537ec5ab
Update
Velociraptor
Search clients
Show All
thm-velociraptor.eu-west-1.compute.internal connected
thmadmin
New Collection: Configure Parameters
TrendMicro
Trend Micro Data (by Drew Ervin): Trend Micro Logs, Trend Micro Security Agent Report Logs, Trend Micro Security Agent Connection Logs
USBDevicesLogs
USB devices log files (by Eric Zimmerman): Setupapi.log Win7+, Setupapi.log XP, Setupapi.log Win7+
Ubuntu
Ubuntu on Windows Subsystem for Linux (by Matt Dawson): Ubuntu WSL .bashrc, Ubuntu WSL .profile, Ubuntu WSL User Crontabs, Ubuntu WSL Apt Logs, Ubuntu WSL /etc/os-release, Ubuntu WSL /etc/fstab, Ubuntu WSL /etc/passwd, Ubuntu WSL /etc/group, Ubuntu WSL /etc/shadow, Ubuntu WSL /etc/timezone, Ubuntu WSL /etc/hostname, Ubuntu WSL /etc/hosts, Ubuntu WSL /etc/crontab, Ubuntu WSL /etc/bash.bashrc, Ubuntu WSL /etc/profile, Ubuntu WSL .bash_history
Usenet
Usenet (NZB) Files (by Andrew Rathbun): Usenet (NZB) Files
UsenetClients
Usenet Clients (by Andrew Rathbun): Usenet Clients - NZBGet NZBs, Usenet Clients - Newsleecher, Usenet Clients - SABnzbd Download Logs, Usenet Clients - SABnzbd History.db, Usenet Clients - Newsbin Pro, Usenet Clients - NZBGet Log File
VIPRE
VIPRE Data (by Drew Ervin): VIPRE Business Agent Logs, VIPRE Business User Logs (v7+), VIPRE
Select Artifacts
Configure Parameters
Specify Resources
Review
Launch
2023-08-29T16:29:21.686Z
4:29 PM
8/29/2023


Brave File Edit View History Bookmarks Profiles Tab Window Help
Tue Aug 29 10:07 AM
TryHackMe | Dashboard
TryHackMe | Velociraptor
The VFS :: Velociraptor -
Deployment :: Velociraptor -
Deployment :: Velociraptor -
TryHackMe | Learning Paths
docs.velociraptor.app/docs/gui/vfs/
Velociraptor
Announcements
Documentation
Velociraptor Overview
Deployment
The Admin GUI
Inspecting clients
The VFS
Artifacts
Hunting
VQL Fundamentals
Forensic Analysis
Triage and acquisition
Client Monitoring
Extending VQL
Server Automation
Brought to you by
RAPID7
The VFS
Edit this page
VFS accessors
The top level directory in the VFS tree view represents the accessor. An accessor is simply a dedicated code used to fetch filesystem information from the endpoint.
The file accessor simply uses the OS's APIs to list files or directories and fetch data. The ntfs accessor uses Velociraptor's built in NTFS parser to be able to access hidden NTFS files and Alternate Data Streams (ADS).
Similarly the registry accessor provides file like access to the registry.
Registry Accessor
Interactively investigating an endpoint
Although the VFS presents a familiar interface, it is not ideal for quickly finding the files and registry keys we are usually interested in. One would need to know exactly which files are of interest and then click over multiple


Searching Filenames
Edit this page
Raw Response JSON
Glob output
Some of the more important columns available are
lPath is the complete path to the matching file, whereas the Name is just the filename.


admin
NotebookId
Name
Description
Creation Time
Modified Time
Creator
Collaborators
N.CJOGPUACOD69C
notebook
notebook
2023-08-31 21:58:17 UTC
2023-08-31 21:58:17 UTC
admin
admin
2023-08-31 22:08:27 UTC (71s)
Full_Path
File_Name
PE
C:/Windows/System32/spool/drivers/x64/3/FXSDRV.DLL
FXSDRV.DLL
"FileHeader" : {...}
"GUIDAge" : "CE127918A8E41A19888BBB312F6806E51"
"PDB" : "FXSDRV.pdb"
"Sections" : [...]
"VersionInformation" : {...}
2023-08-31T22:19:48.414Z

```
SELECT "C:/" + FullPath AS Full_Path,FileName AS
File_Name,parse_pe(file="C:/" + FullPath) AS PE

FROM parse_mft(filename="C:/$MFT", accessor="ntfs")

WHERE NOT IsDir

AND FullPath =~ "Windows/System32/spool/drivers"

AND PE
```

Utilized VQL Query to find a suspicious DLL.

Malware Analysis

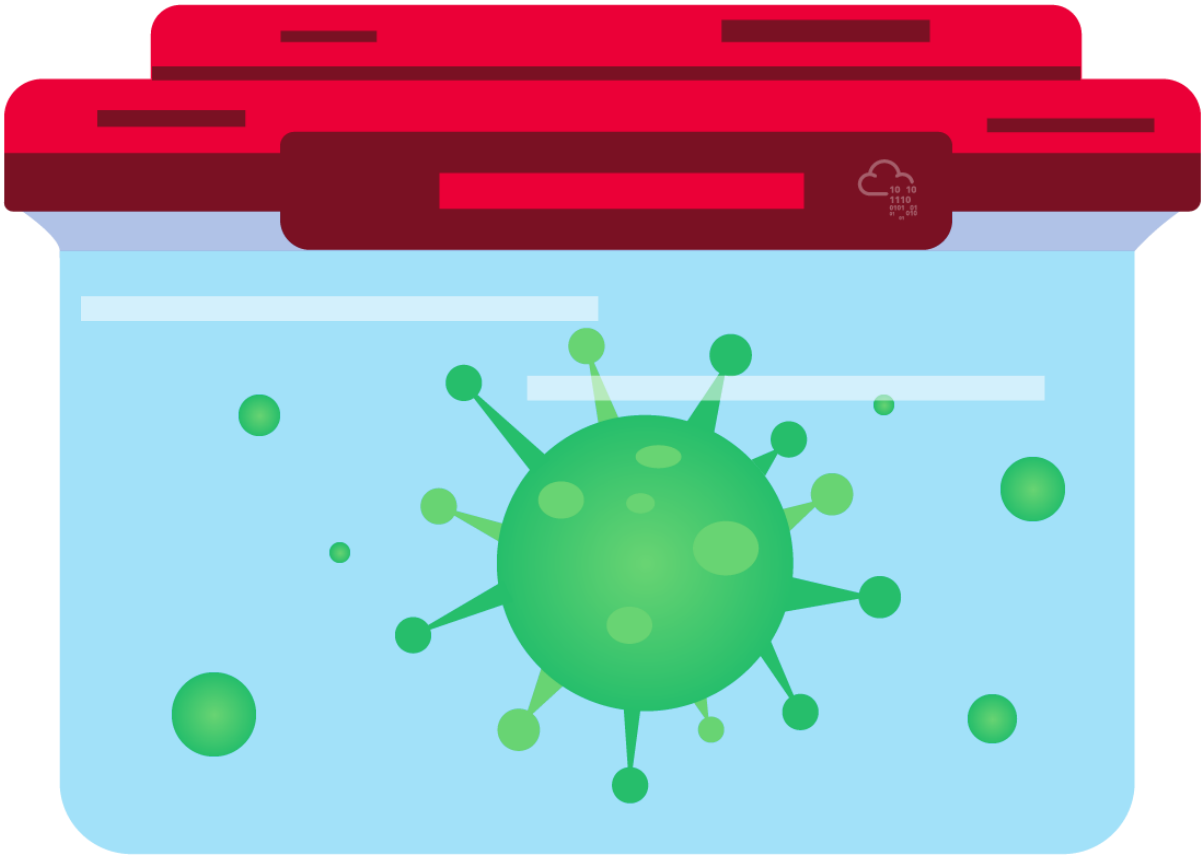
- Never analyze malware or suspected malware on a machine that does not have the sole purpose of analyzing malware.
- When not analyzing or moving malware samples around to different locations, always keep them in password-protected zip/rar or other archives so that we can avoid accidental detonation.
- Only extract the malware from this password-protected archive inside the isolated environment, and only when analyzing it.
- Create an isolated VM specifically for malware analysis, which has the capability of being reverted to a clean slate once you are done.
- Ensure that all internet connections are closed or at least monitored.
- Once you are done with malware analysis, revert the VM to its clean slate for the next malware analysis session to avoid residue from a previous malware execution corrupting the next one.

Malware Analysis is like solving a puzzle. Different tools and techniques are used to find the pieces of this puzzle, and joining those pieces gives us the complete picture of what the malware is trying o do. Most of the time, you will have an executable file (also called a binary or a PE file. PE stands for Portable Executable), a malicious document file, or a Network Packet Capture (Pcap). The Portable Executable is the most prevalent type of file analyzed while performing Malware Analysis.

To find the different puzzle pieces, you will often use various tools, tricks, and shortcuts. These techniques can be grouped into the following two categories:

- Static Analysis
- Dynamic Analysis

Static Analysis

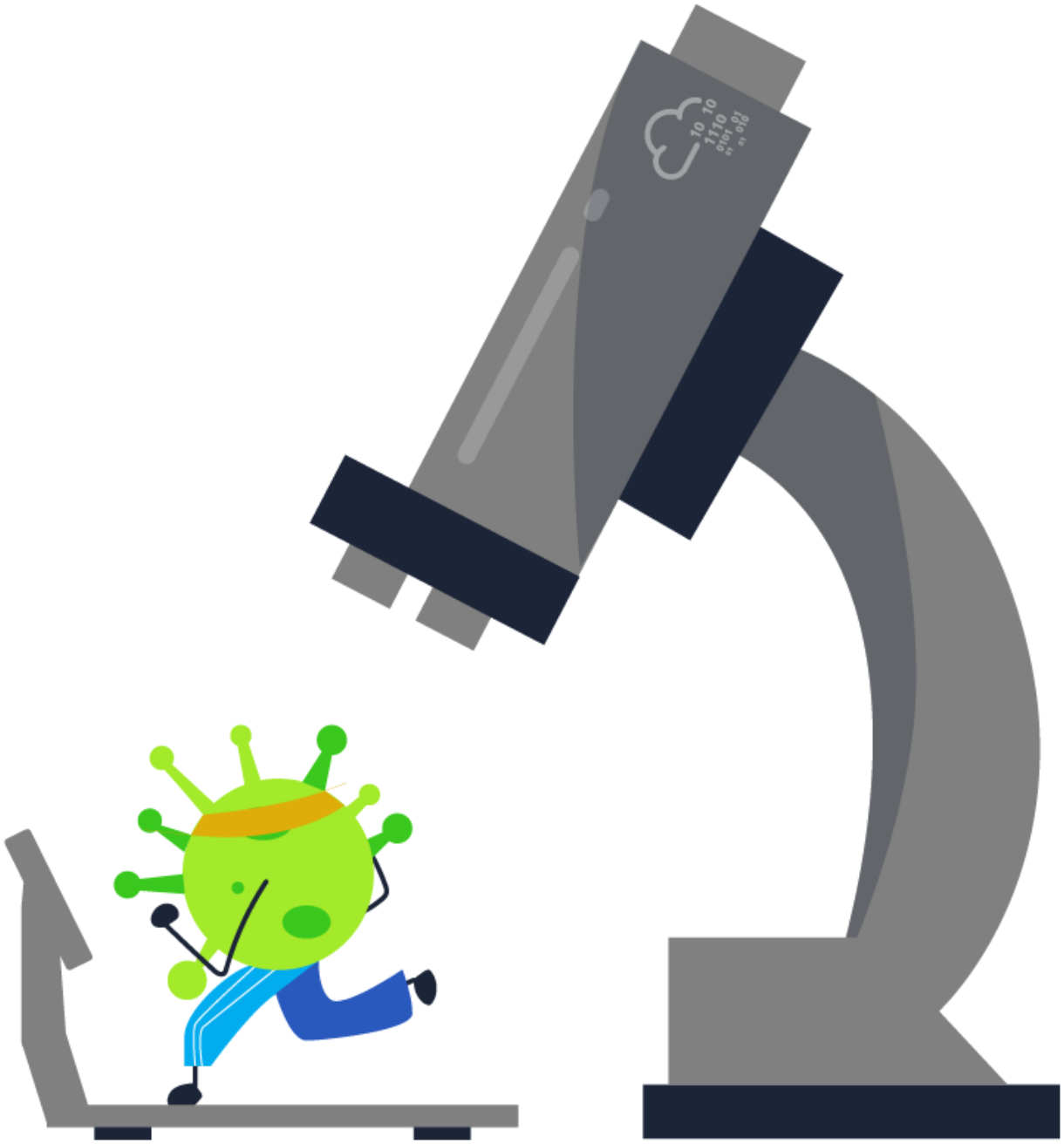


When malware is analyzed without being executed, it is called Static Analysis. In this case, the different properties of the PE file are analyzed without running it. Similarly, in the case of a malicious document, exploring the document's properties without analyzing it will be considered Static Analysis. Examples of static analysis include checking for strings in malware, checking the PE header for information related to different sections, or looking at the code using a disassemble. We will look at some of these techniques later in the room.

Malware often uses techniques to avoid static analysis. Some of these techniques use obfuscation, packing, or other means of hiding its properties. To circumvent these techniques, we often use dynamic analysis.

Dynamic Analysis

Malware faces a dilemma. It has to execute to fulfill its purpose, and no matter how much obfuscation is added to the code, it becomes an easy target for detection once it runs.



Static analysis might provide us with crucial information regarding malware, but sometimes that is not enough. We might need to run the malware in a controlled environment to observe what it does in these cases. Malware can often hide its properties to thwart Static Analysis. However, in most of those cases, Dynamic Analysis can prove fruitful. Dynamic analysis techniques include running the malware in a VM, either in a manual fashion with tools installed to monitor the

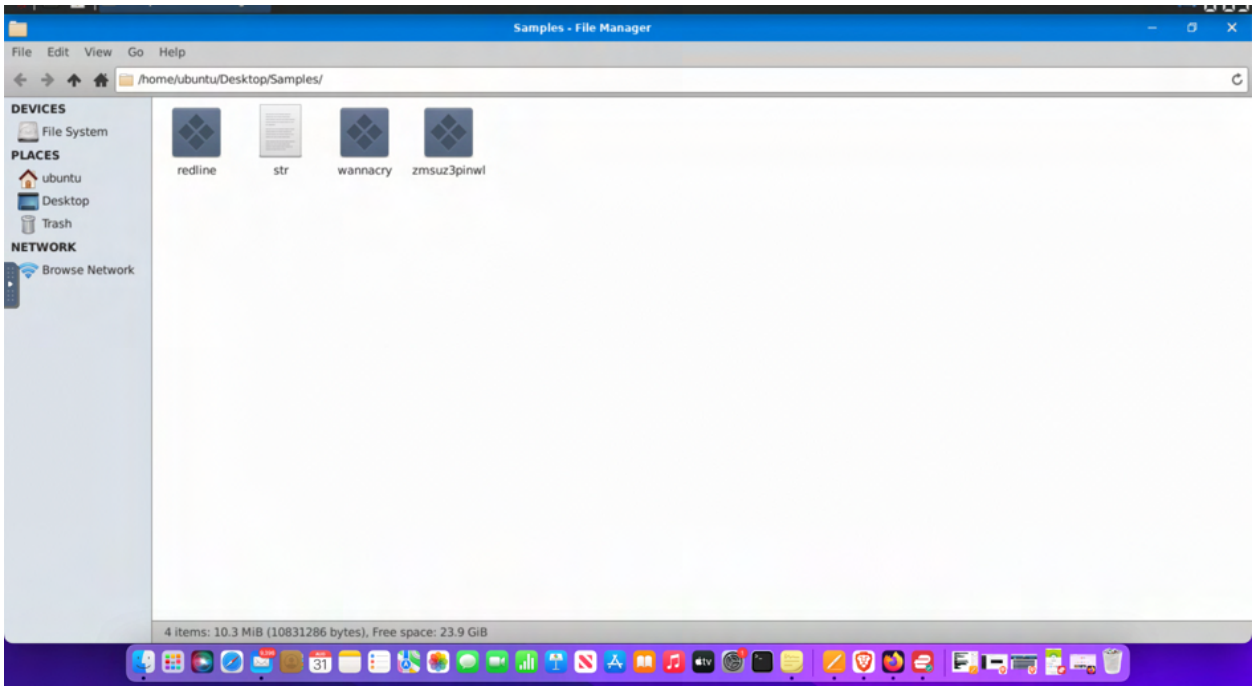
malware's activity or in the form of sandboxes that perform this task automatically. We will learn about some of these techniques later in this room. Once we run the malware in a controlled environment, we can use our knowledge from the Windows Forensics rooms to identify what it did in our environment. The advantage here is that since we control the environment, we can configure it to avoid noise, like activity from a legitimate user or Windows Services. Thus, everything we observe in such an environment points to malware activity, making it easier to identify what the malware did in this scenario.

Malware, however, often uses techniques to prevent an analyst from performing dynamic analysis. Since most dynamic analysis is performed in a controlled environment, most methods to bypass dynamic analysis include detecting the environment in which it is being run. Therefore, in these cases, the malware uses a different, benign code path if it identifies that it is being run in a controlled environment.

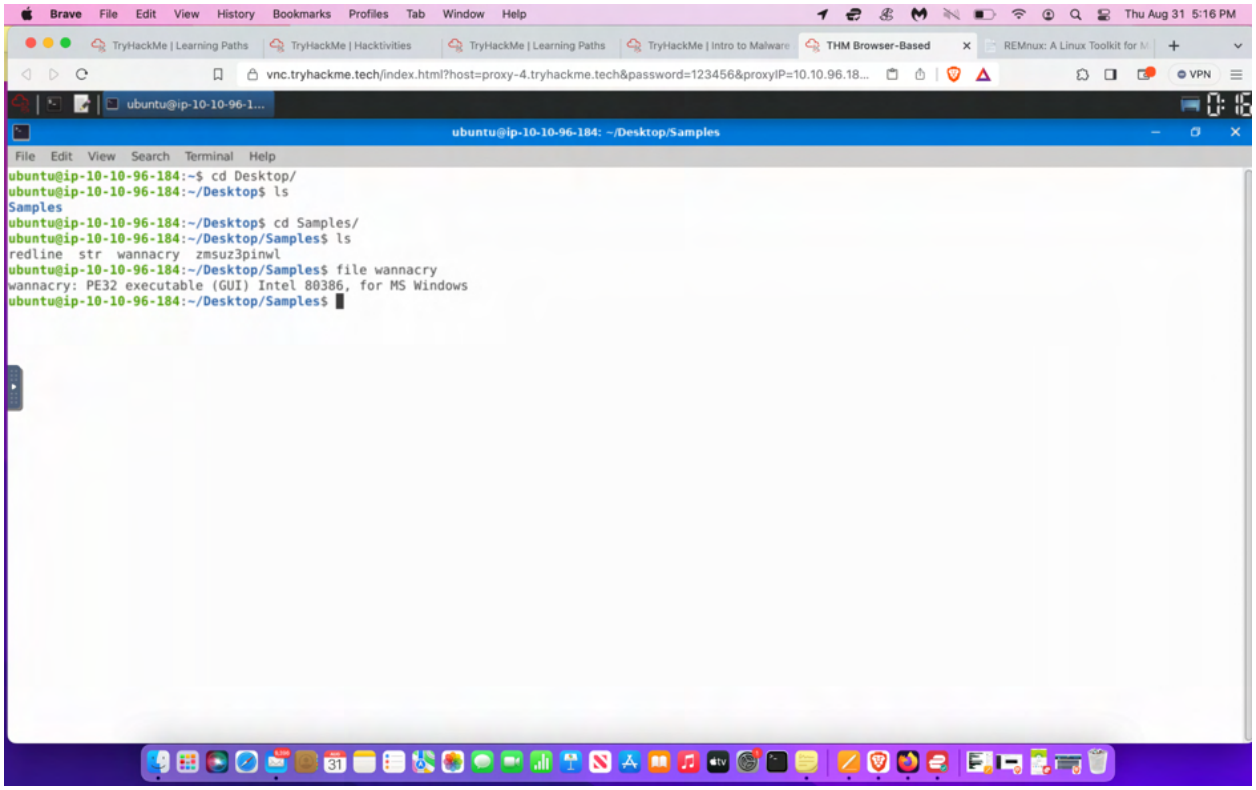
Advanced Malware Analysis

Advanced Malware Analysis Techniques

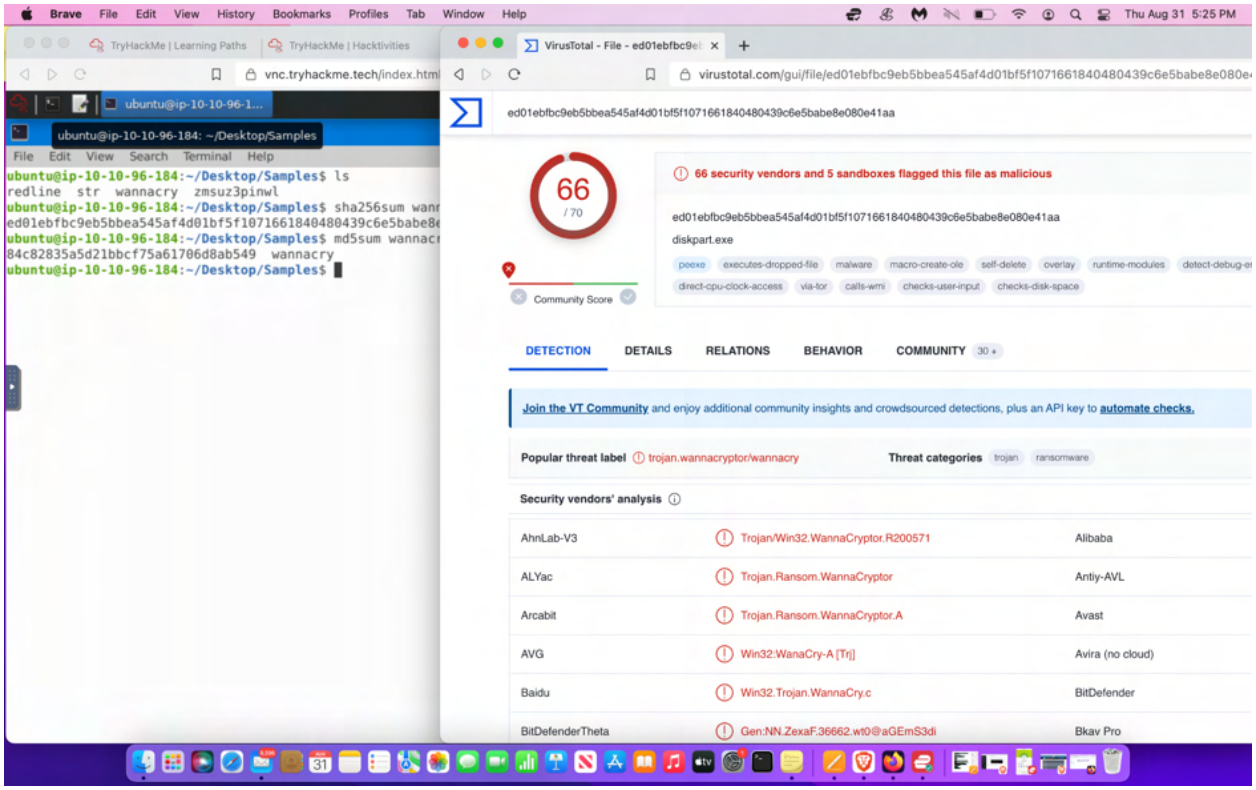
- Objective: Analyze malware that evades basic static and dynamic analysis.
- Tools: Disassemblers and debuggers.
- Disassemblers: Convert malware code from binary to assembly language.
    - Allows static examination of malware instructions.
- Debuggers: Attach to a program and monitor malware instructions during runtime.
    - Enables starting and stopping malware at different points.
    - Identifies crucial information.
    - Provides insights into system memory and CPU utilization.



Remnux the linux distribution used for malware analysis.

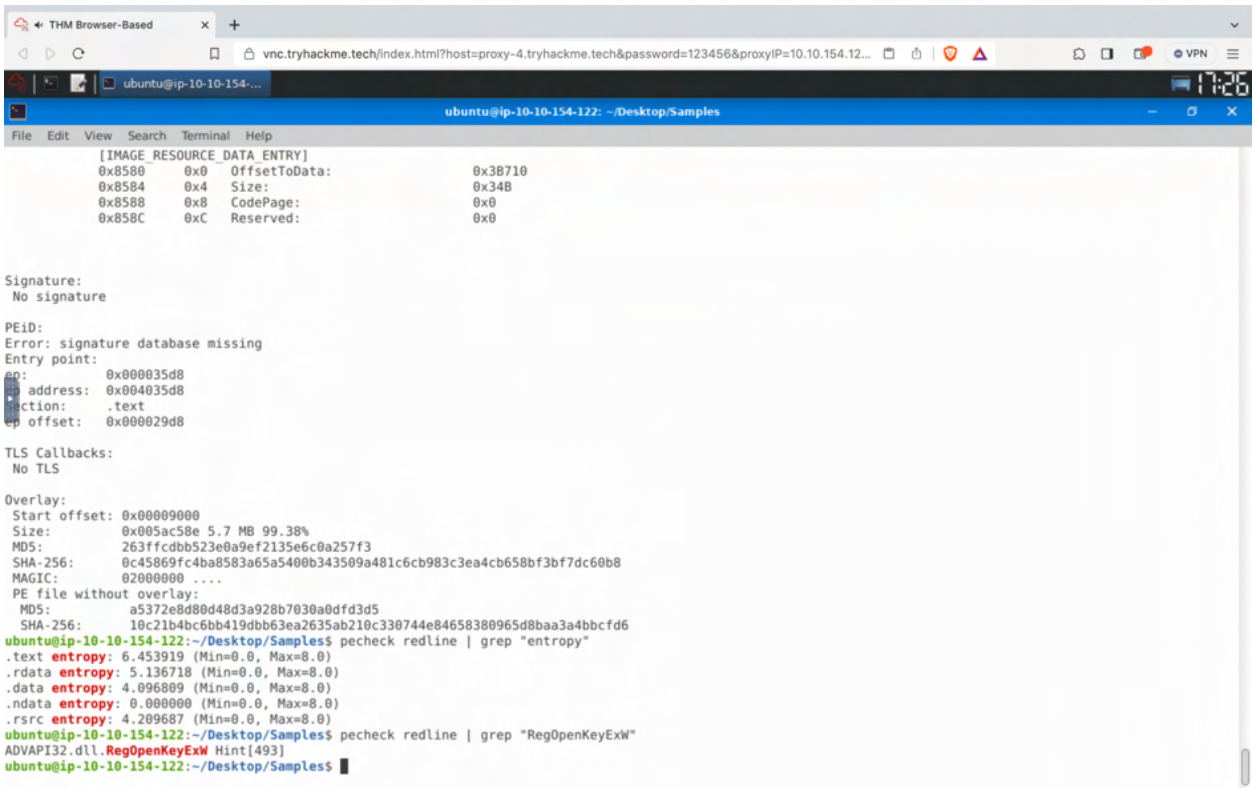


Details for the famous wannacry malware present in samples directory. uses the X-86. It is a PE32 Executable GUI for windows.



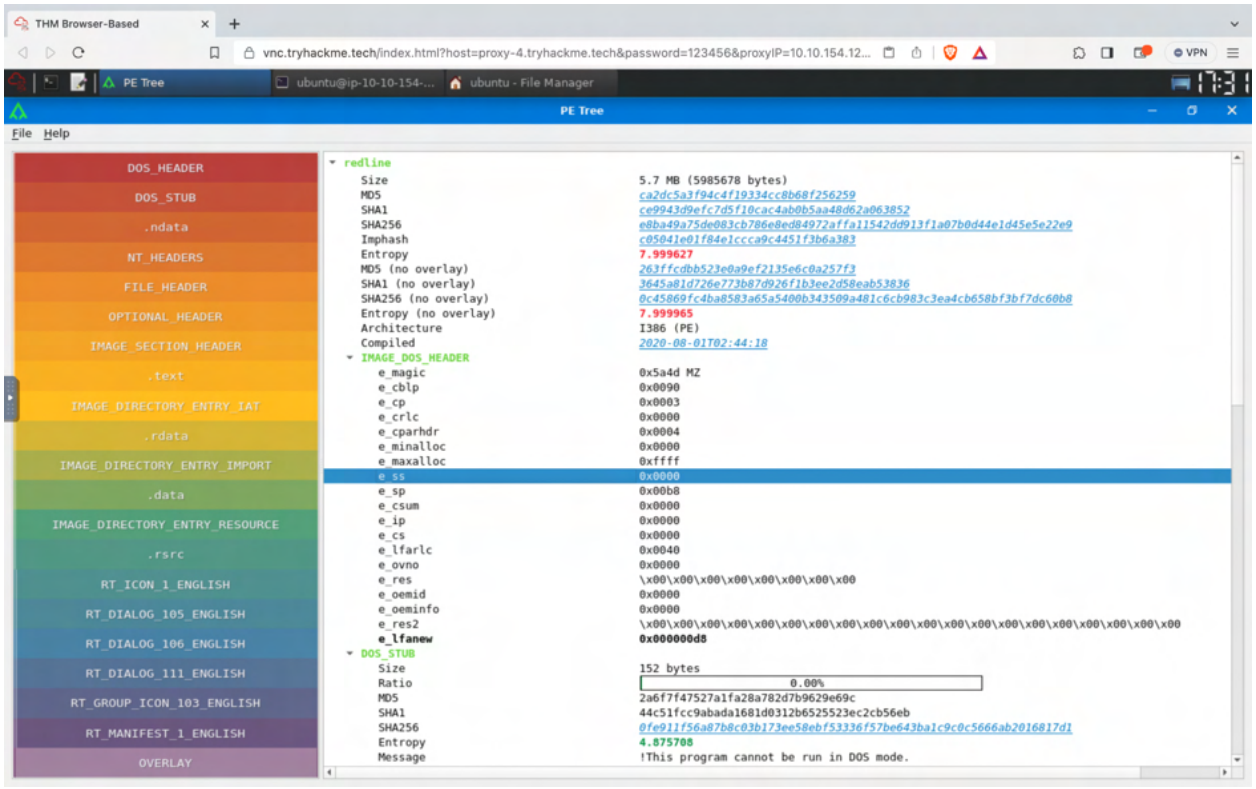
Calculate md5 or sha256 hash and look up the hash on virus total.

Analysed the PE header using the pecheck command to find various sections such as .text , .data, .rdata, .ndata, .rsrs



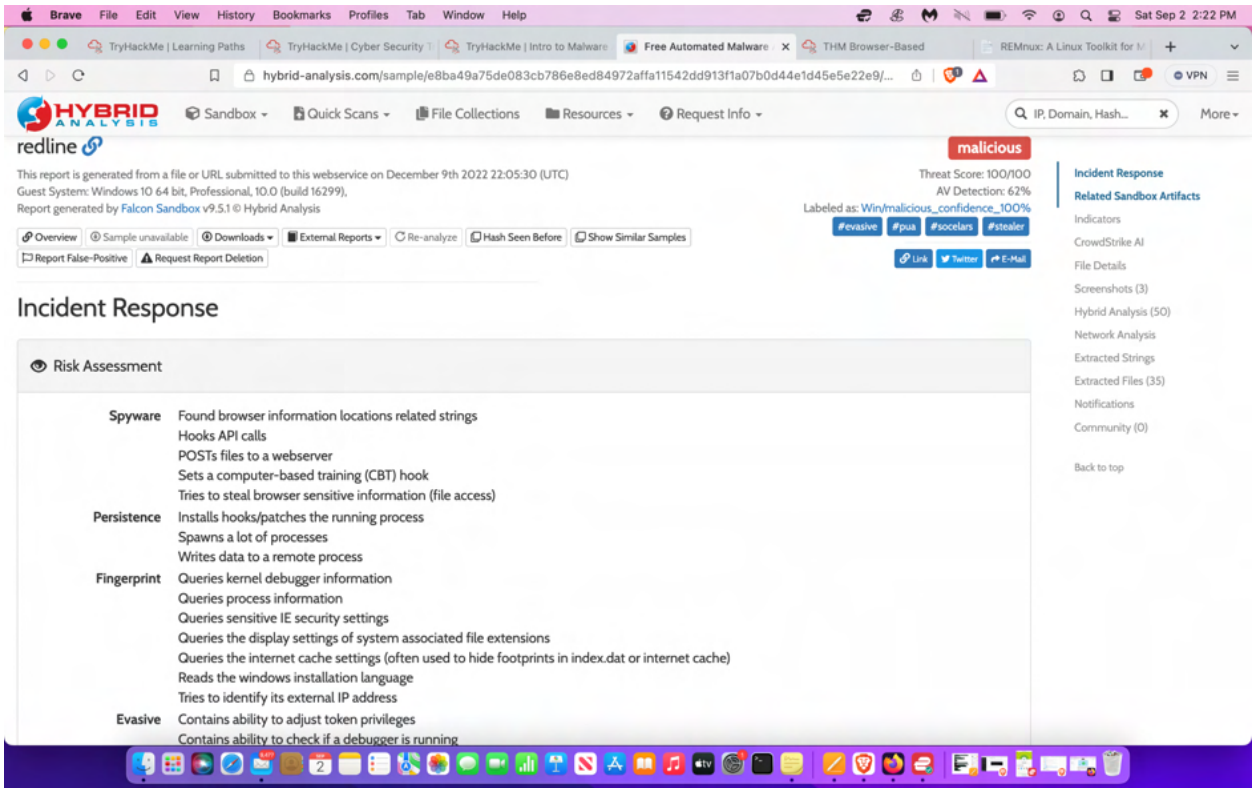
Executed the pe-tree command on the terminal `pe-tree redline`

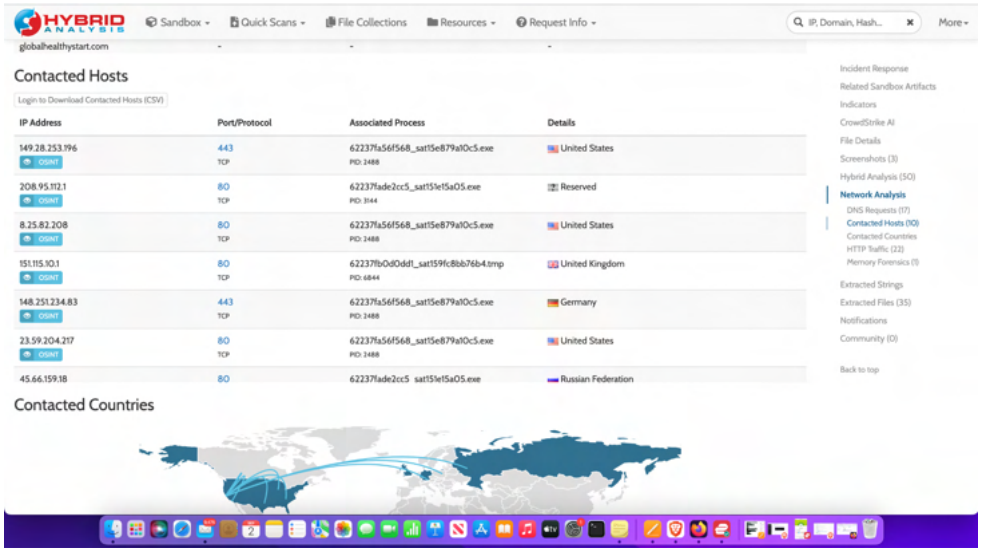
for the redline sample given to view a GUI based interface in a more readable format. Ultimately this contains metadata which can be helpful for our malware analysis.



Dynamic Analysis

Uploaded samples on hybrid analysis to analyze the report given for the redline sample. This was done my using the md5 hash as opposed to uploading an actual sample.





Contacted hosts findings in the network analysis of the incidence response.

Summary

Malware analysis. However, this was just scratching the surface.

- Static and Dynamic analysis of malware
- Finding strings, calculating hashes, and running AV scans on malware
- Introduction to the PE header and how to use information from it in malware analysis
- Sandboxing and different online sandboxes that we can use
- How malware evades the techniques we just discussed.

PHISHING

Email Protocols and Message Flow

Email Protocols:

- Email transactions involve specific protocols designed for network-related tasks.
- Three main protocols facilitate outgoing and incoming email messages.

SMTP (Simple Mail Transfer Protocol):

- Handles the sending of emails.

POP3 (Post Office Protocol):

- Responsible for transferring email between a client and a mail server.
- Emails are downloaded and stored on a single device.
- Sent messages are stored on the same device.
- Accessible only from the device where emails were downloaded.
- Enable "Keep email on server" to retain messages on the server.

IMAP (Internet Message Access Protocol):

- Also responsible for transferring email between a client and a mail server.
- Emails are stored on the server and can be downloaded to multiple devices.
- Sent messages are stored on the server.
- Messages can be synced and accessed across multiple devices.

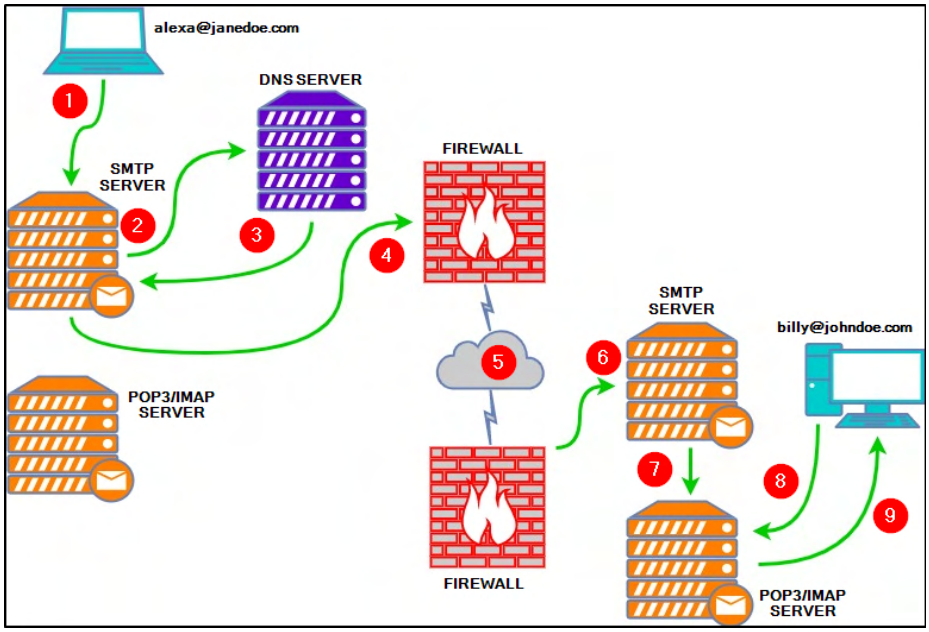
Email Transmission Steps:

Below is an explanation of each numbered point from the above diagram:

1. Alexa composes an email to Billy (billy@johndoe.com) in her favorite email client. After she's done, she hits the send button.

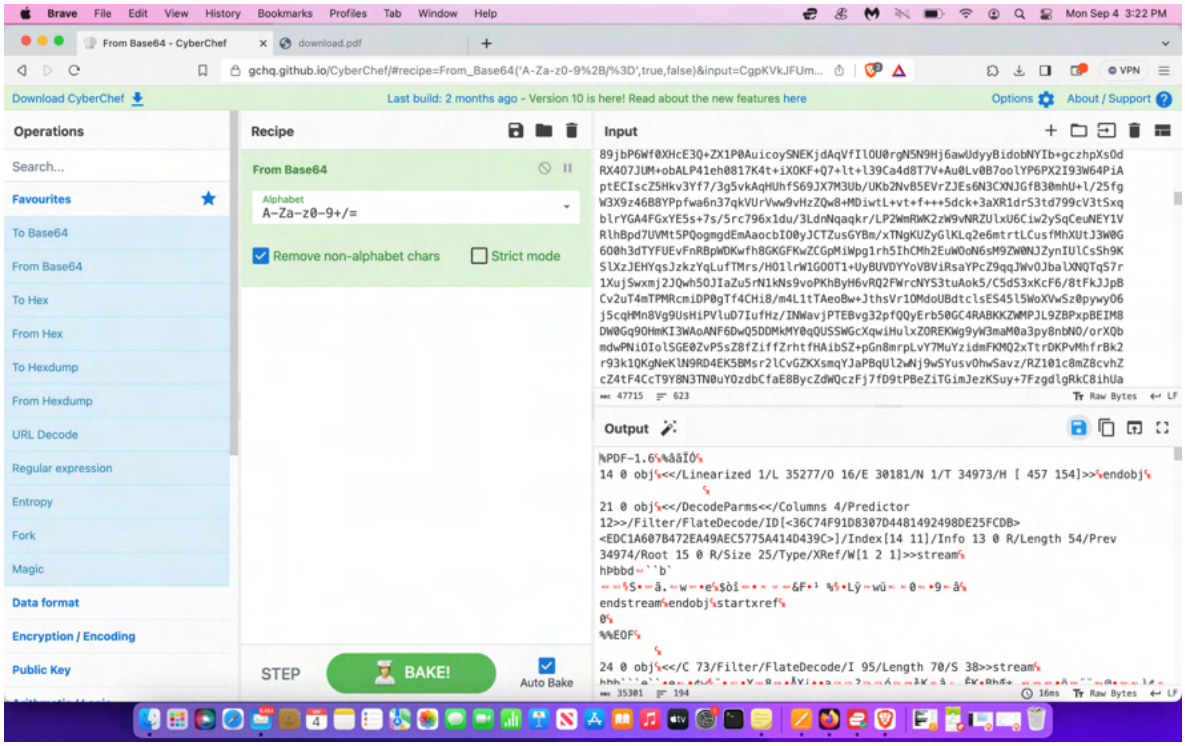
2. The **SMTP** server needs to determine where to send Alexa's email. It queries **DNS** for information associated with johndoe.com.
3. The **DNS** server obtains the information johndoe.com and sends that information to the **SMTP** server.
4. The **SMTP** server sends Alexa's email across the Internet to Billy's mailbox at johndoe.com.
5. In this stage, Alexa's email passes through various **SMTP** servers and is finally relayed to the destination **SMTP** server.
6. Alexa's email finally reached the destination **SMTP** server.
7. Alexa's email is forwarded and is now sitting in the local **POP3/IMAP** server waiting for Billy.
8. Billy logs into his email client, which queries the local **POP3/IMAP** server for new emails in his mailbox.
9. Alexa's email is copied (**IMAP**) or downloaded (**POP3**) to Billy's email client.

Lastly, each protocol has its associated default ports and recommended ports. For example, **SMTP** is port 25.

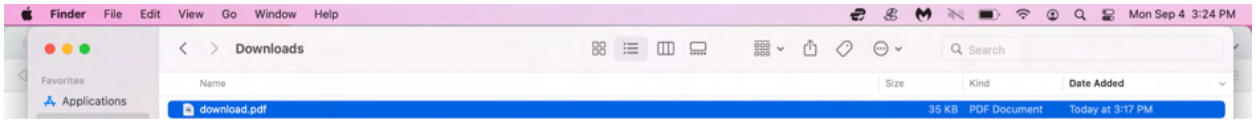


Email Body

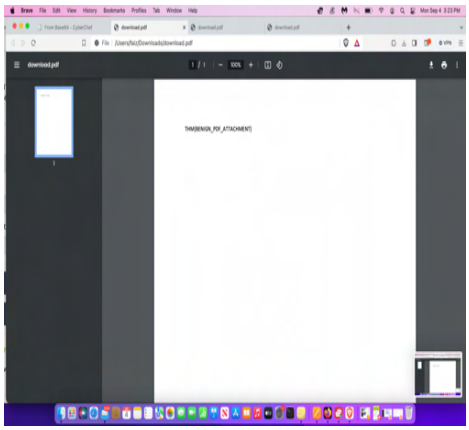
So far everything has been straightforward . One important thing to note is that contents within a pdf file that are encoded from base64 data have to be decoded in human readable format. One notable for example is a email , which consists of a pdf file. Since the base 64 data is very lengthy , we simply just copied the contents and pasted in in cyberchef.



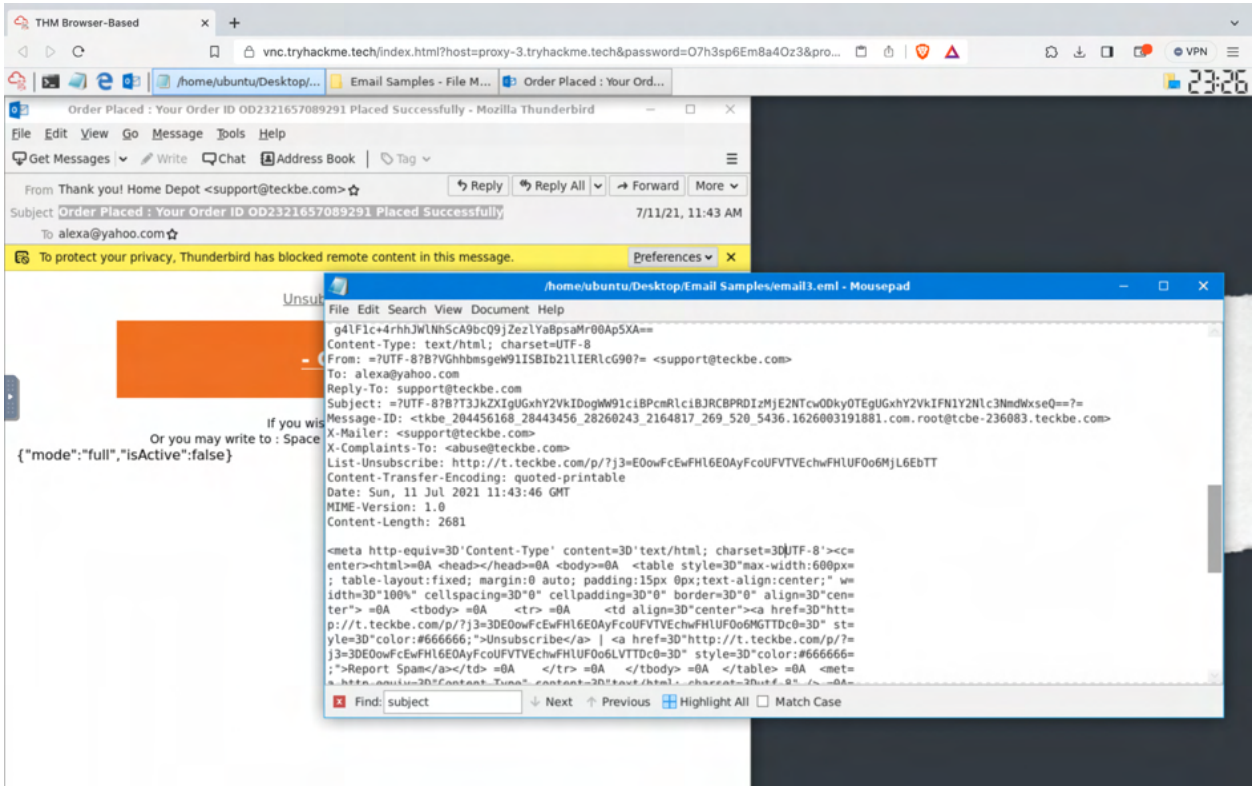
Then we selected the from base64 recipe and dragged it, and bake. We then saved the output to a pdf file .



Once we saved it to a pdf file, then we should be able to view the BASE64 contents decoded of the pdf file



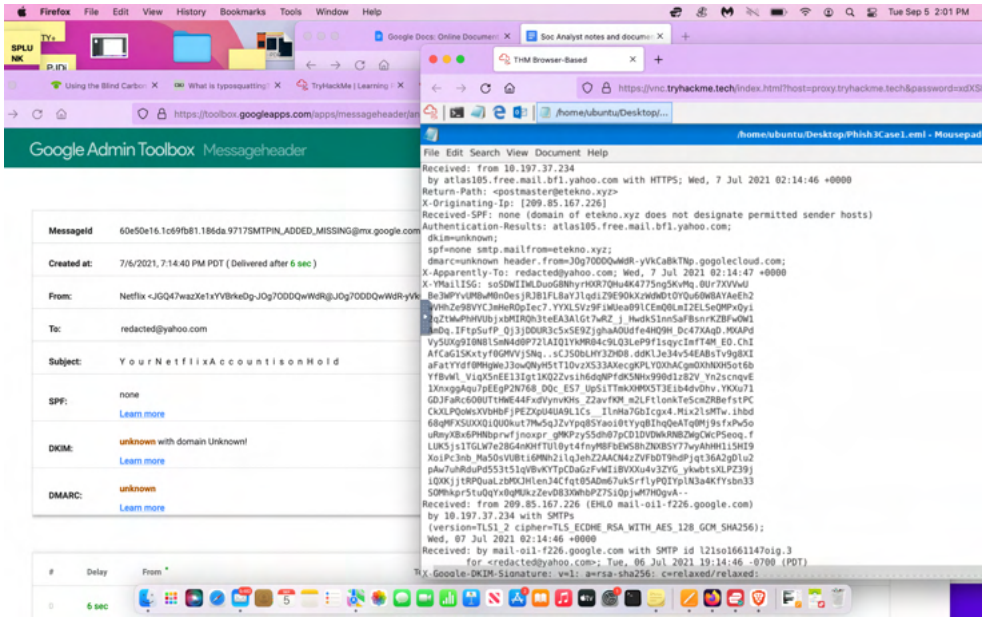
EMAIL ANALYSIS PHISING

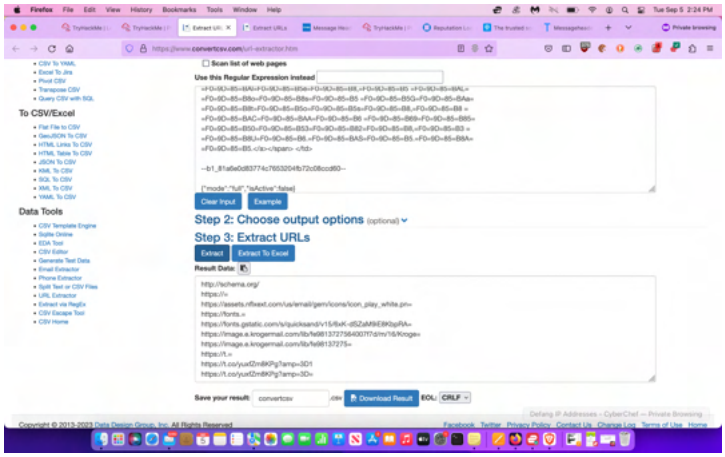


From the given samples we can find some useful information given the email headers. We can find the sender's address which is stated in the above screen shot. A way to find other valuable information such as which entity the sender has masqueraded as or in other words spoofed, we can use the given platform to open the email sample. From the analysis we can see in the email that the sender has spoofed using home depot. This information apparently was not visible within the sample file. Another factor was subject, that was encoded, we learned that the subject of the email could be retrieved by simply opening the email through the thunderbird application. A link was placed by the sender, which was also retrieved through the thunderbird application.

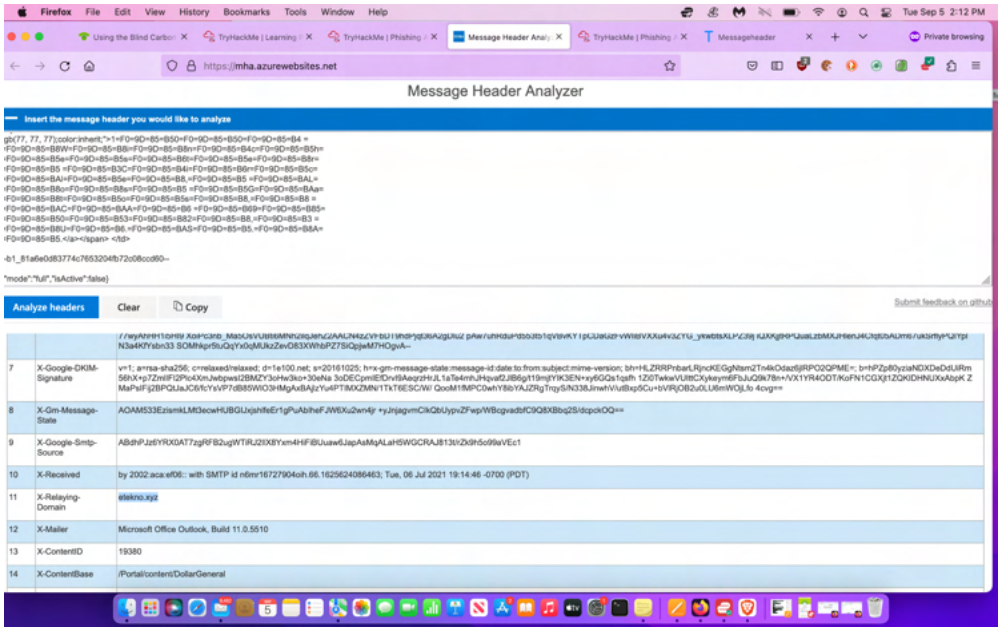
PHISHING CASE

Used email headers to find relevant information. There are many useful tools one can use to find useful information from a phishing email. The one listed below is from google, called message header. It appears to be user friendly and one can navigate through the findings to for analysis





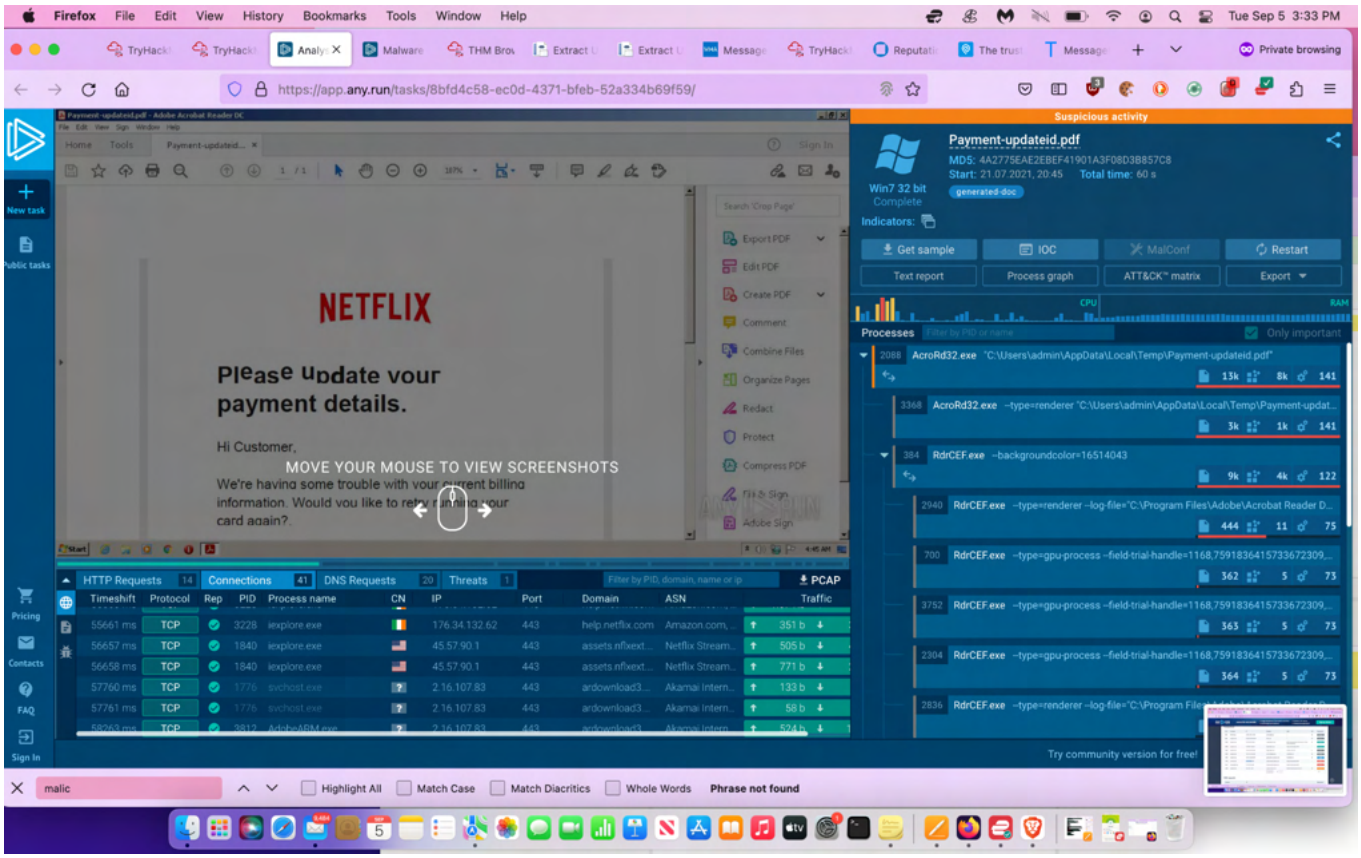
The url extractor is a pretty useful tool for finding all of the urls that are listed. Typically this could also be located in the e-mail body.

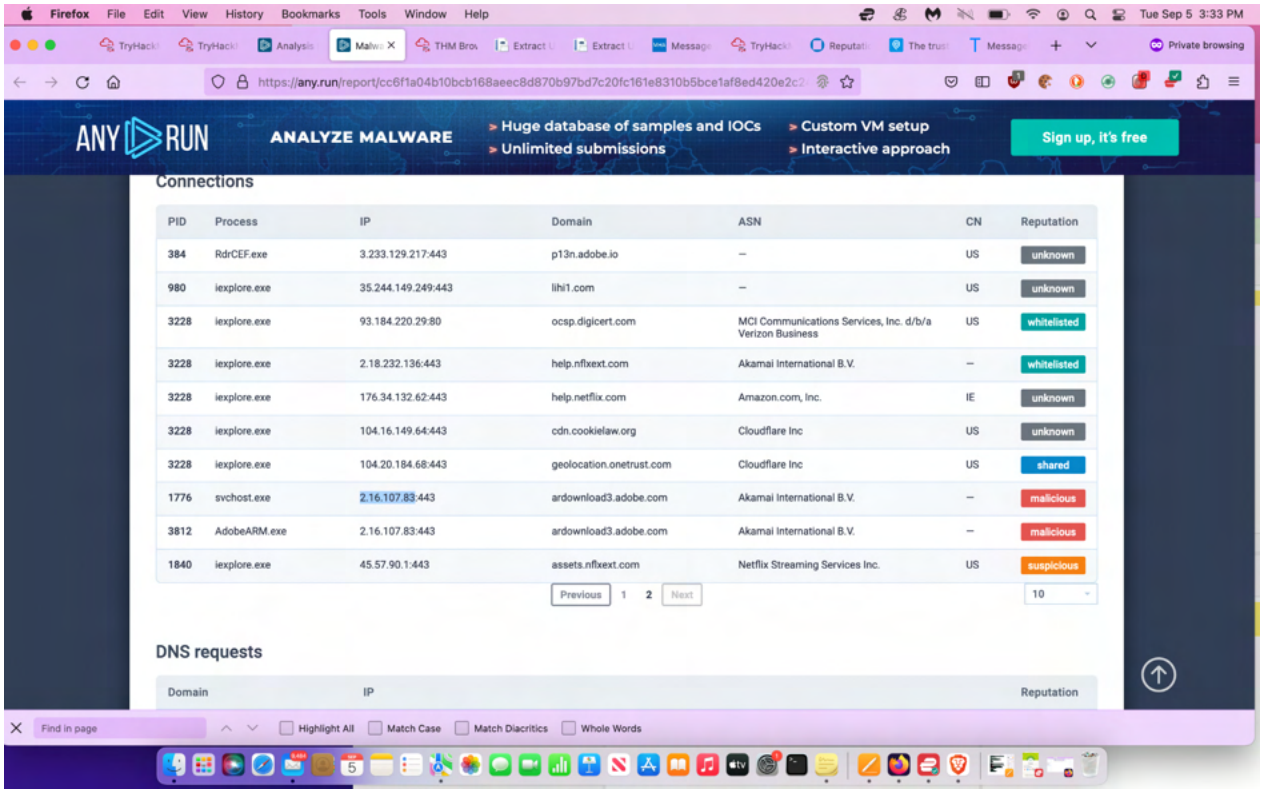


The message header analysis is also a useful tool to find important information such as domain of interest, submitting host, received host, protocols and IP addresses.

Phishing case 2 lab

In this lab we analyzed a malicious attachment from a phishing email through a online malware sandbox platform for analysis, we can analyze the network and discover other important details. In summary we discovered how the email was classified which was suspicious. We discovered a pdf file, SHA 256 of the pdf file was available. A windows process file was flagged as potentially bad traffic. In addition we can click and view screen shots of the live e-mail, in terms of how a potential victim could be a target.

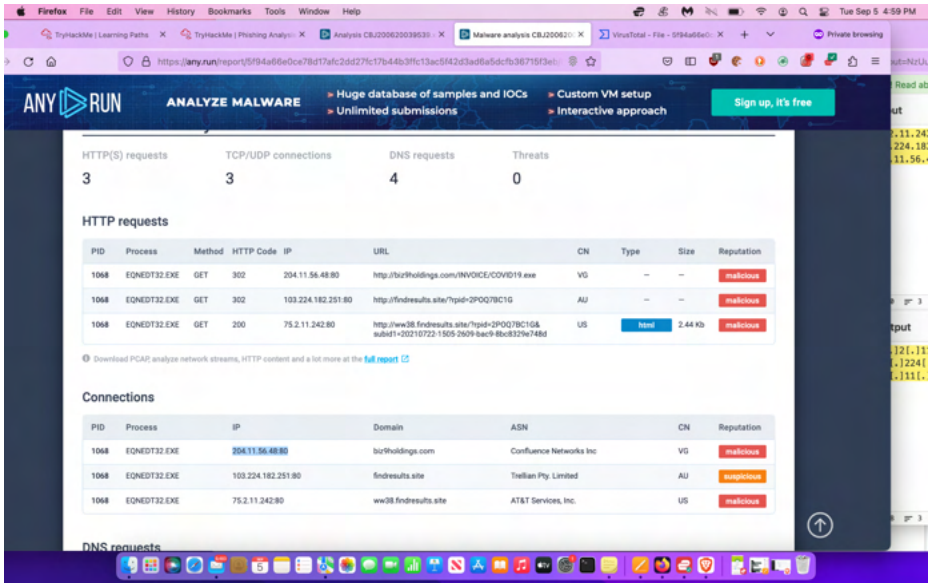


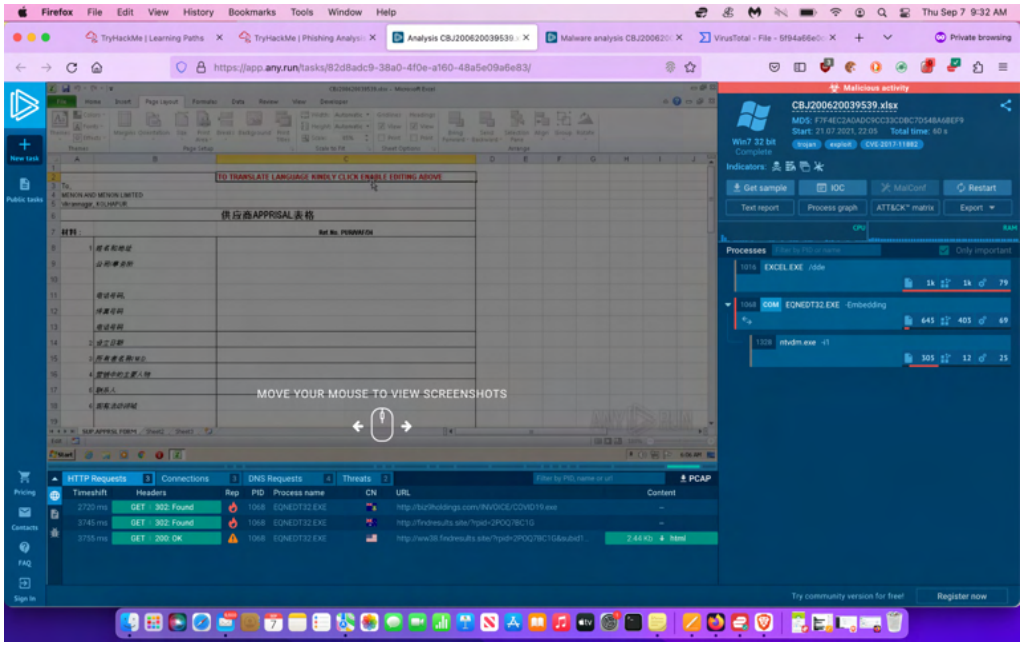


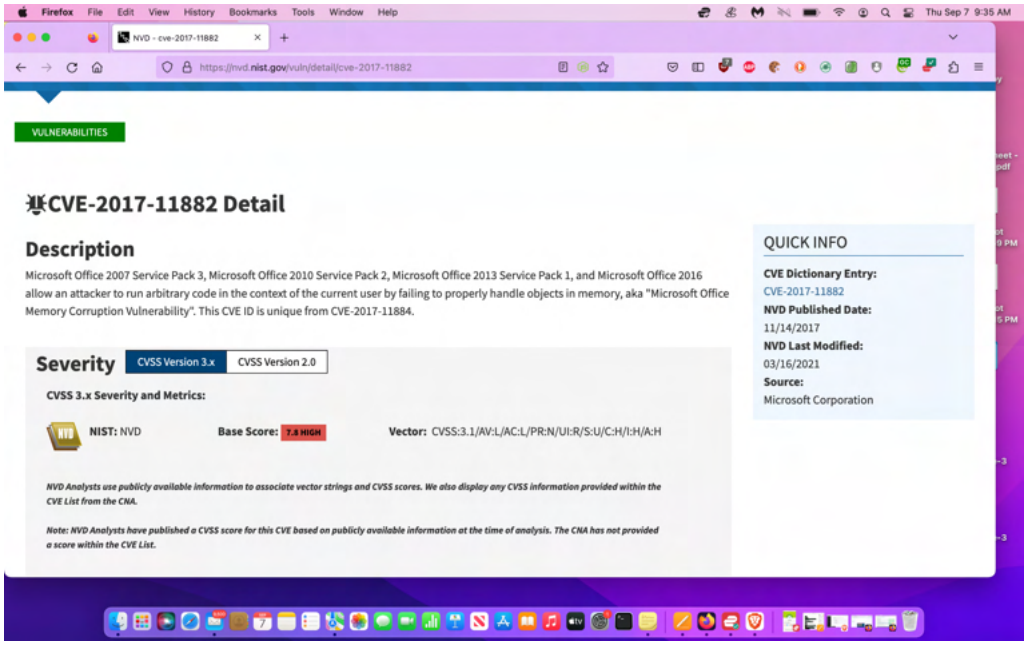
A helpful feature from the platform was a self generated report one could create . This lists important details such as PID, and ip address. Luckily it was readable and important labels such as malicious IP were available to analyze from the given report.

Phishing Case 3 lab.

For this case , a sandbox with the suspicious email was provided. We retrieved the details , and we discovered a Malicious activity had been classified. From the given report generated, We learned that this time there was an excel file present. We traced 3 malicious domains, along with 3 malicious IP addresses. In addition we discovered the vulnerability that the malicious attachment attempted to exploit. That vulnerability happens to be **CVE-2017-11882.**




MOVE YOUR MOUSE TO VIEW SCREENSHOTS


VULNERABILITIES
CVE-2017-11882 Detail
Description
Microsoft Office 2007 Service Pack 3, Microsoft Office 2010 Service Pack 2, Microsoft Office 2013 Service Pack 1, and Microsoft Office 2016 allow an attacker to run arbitrary code in the context of the current user by failing to properly handle objects in memory, aka "Microsoft Office Memory Corruption Vulnerability". This CVE ID is unique from CVE-2017-11884.
QUICK INFO
CVE Dictionary Entry:
CVE-2017-11882
NVD Published Date:
11/14/2017
NVD Last Modified:
03/16/2021
Source:
Microsoft Corporation
Severity
CVSS Version 3.x
CVSS Version 2.0
CVSS 3.x Severity and Metrics:
NIST: NVD
Base Score: 7.8 HIGH
Vector: CVSS:3.1/AV:L/AC:L/PR:N/UI:R/S:U/C:H/I:H/A:H
NVD Analysts use publicly available information to associate vector strings and CVSS scores. We also display any CVSS information provided within the CVE List from the CNA.
Note: NVD Analysts have published a CVSS score for this CVE based on publicly available information at the time of analysis. The CNA has not provided a score within the CVE List.

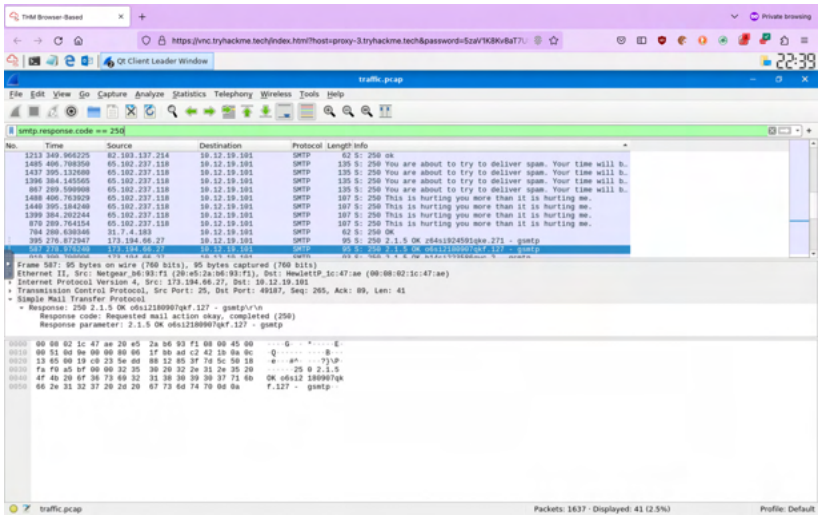
Phishing Prevention

In this lab we explored the varios the various methods defenders can take to protect users from falling as a victim. The main focus for the first portion was email security using SPF, DKIM, DMARC. We explored mitigations as per the MITRE ATTA&CK Framework in terms of tricking targets.

A notable concept we explored was S/MIME which is an important concept learned from our SECRITY+ studies. We briefly touched on digital signatures and encryption, not ignoring the importance of public key cryptography.

SMTP STATUS CODES

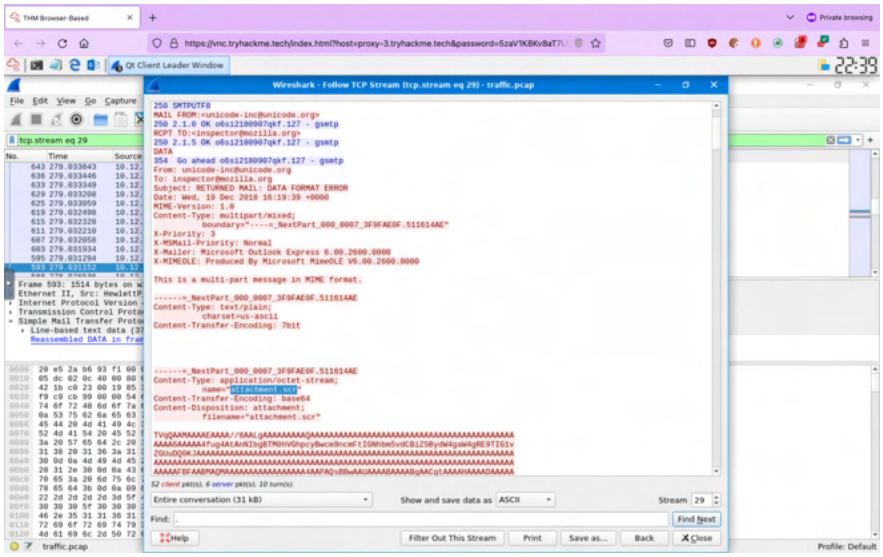
We proceeded on to the lab portion which involved SMTP status codes. We deployed wireshark , and completed the given scenario. Most of the portions were straightforward because we had our trusty wireshark documentation to guide us when filtering to analyze the given packet capture.



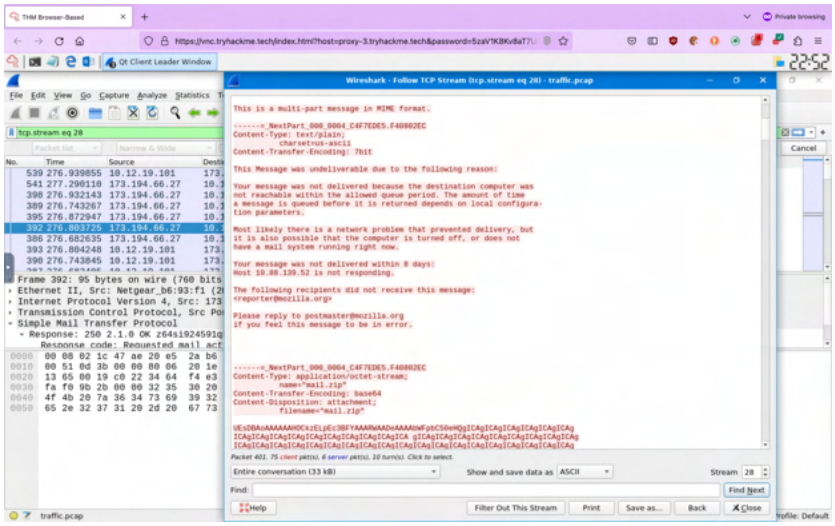
SMTP TRAFFIC ANALYSIS

In this portion of the lab we again explored packet capture analysis using wireshark on a virtual machine. Again most of it was straight forward using the given wire shark documentation. However, during our analysis we discovered some interesting findings which i will share.

During our analysis we discovered an attachment file



During our analysis we discovered an attachment file, we simply followed the tcp stream to retrieve the finding. We learned that the port the traffic was using was 25 catered to SMTP. The source ip address was 10.12.19.101 involved for all SMTP traffic. Finally relating to this we discovered the final attachment , which was mail.zip



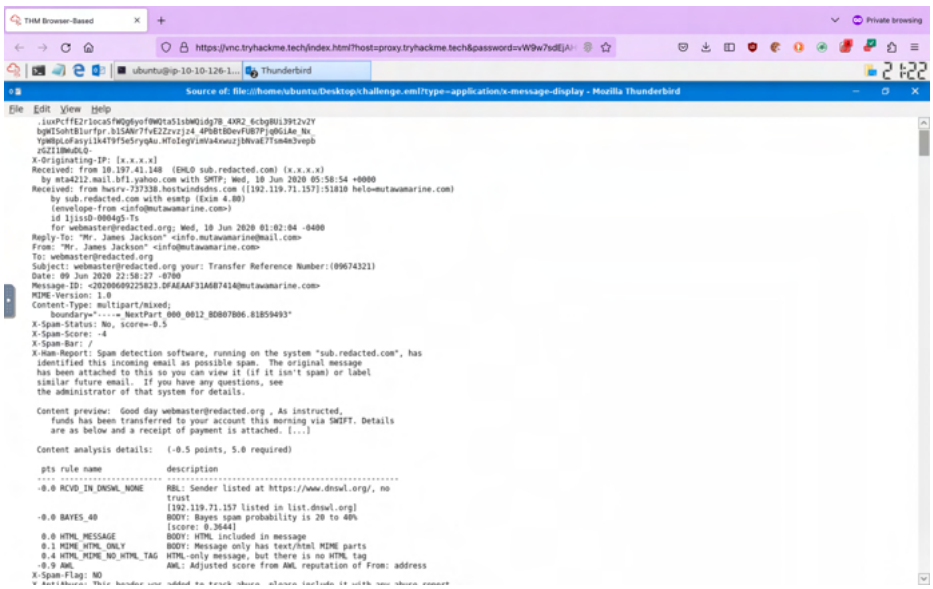
Conclusion

Finally in conclusion an important aspect for SOC analysts to study was a given phishing IR playbook. The phishing playbook https://www.incidentresponse.org/playbooks/phishing summarizes an incident response by the NIST incident process. It covers **Prepare, Detect , Analyze, Contain , Eradicate , Recovery, Post-Incident Handling.**

Lab Scenario:

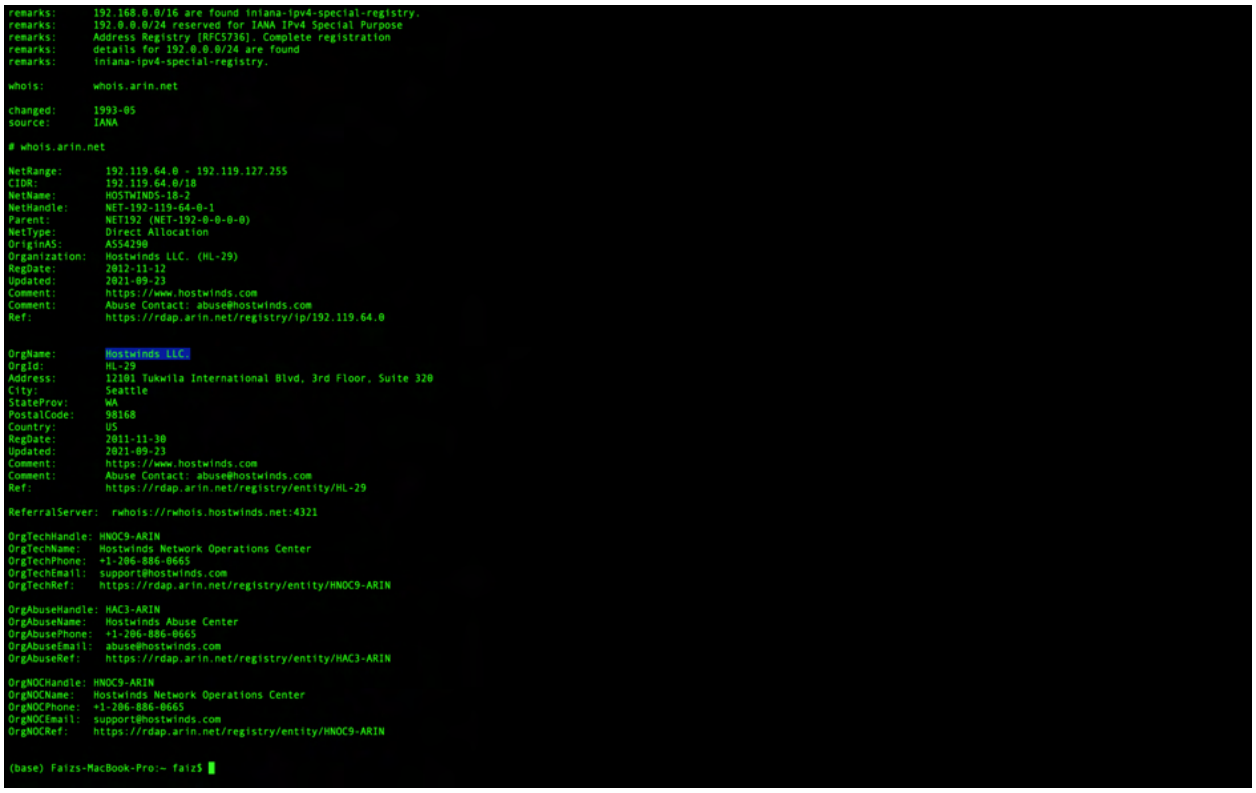
Here we investigated a E-mail sample via a virtual machine to determine if it was legitimate. The e-mail also contained an attachment . One important thing to note here is that it was a generic email with a generic good day greeting. We analyzed the email headers using thunderbird as the application.

From the analysis we notice the date the e-mail was received, which was 6/10/20.
Email was from Mr James. Jackson
Email address: info@mutawamarine.com
Email address to receive a reply of email: info@mutawamarine.com
Originating ip address: 192.119.71.157

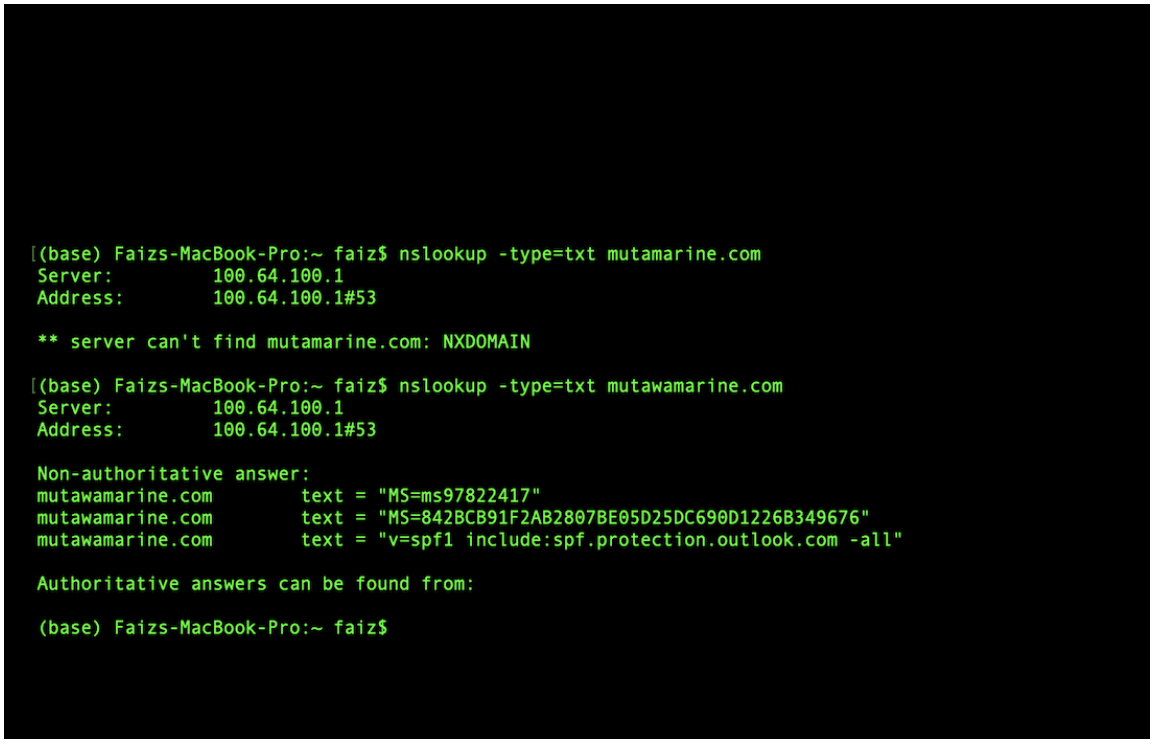


Next during our analysis we needed to run the who is command on our local terminal , so as we went ahead and did that .
Command : whois 192.119.71.157

From the terminal output , we notice that the owner of the originating IP address is Hostwinds LLC.



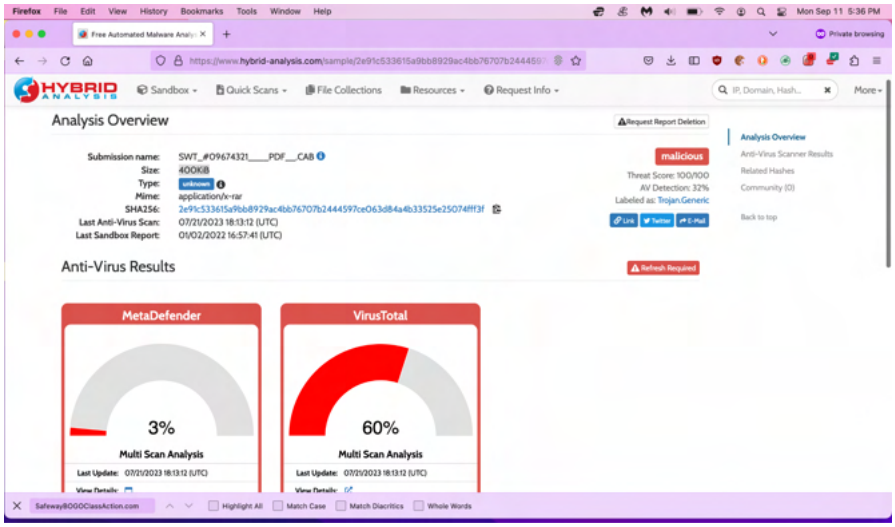
To find spf record for return path
Command: nslookup -type=txt mutamarine.com



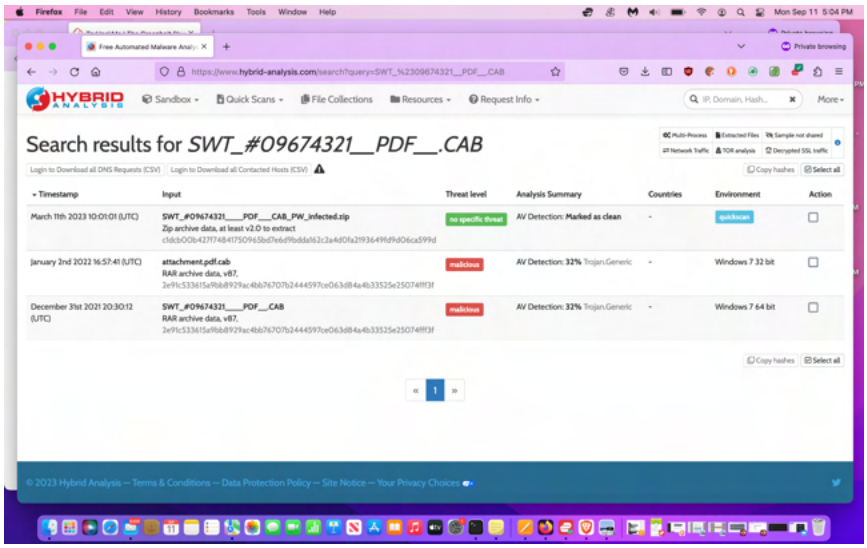
To find the DMARC record for the return path domain we used the dig command
Command: dig TXT _dmarc.mutawamarine.com
We notice that it is **v=DMARC1; p=quarantine; fo=1**

```
(base) Faizs-MacBook-Pro:~ faiz$ dig TXT _dmarc.mutawamarine.com

; <<>> DiG 9.10.6 <<>> TXT _dmarc.mutawamarine.com
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 62614
;; flags: qr rd ra; QUERY: 1, ANSWER: 1, AUTHORITY: 0, ADDITIONAL: 1

;; OPT PSEUDOSECTION:
; EDNS: version: 0, flags:; udp: 1232
;; QUESTION SECTION:
;_dmarc.mutawamarine.com.       IN      TXT

;; ANSWER SECTION:
_dmarc.mutawamarine.com. 3600   IN      TXT     "v=DMARC1; p=quarantine; fo=1"

;; Query time: 111 msec
;; SERVER: 100.64.100.1#53(100.64.100.1)
;; WHEN: Mon Sep 11 17:34:45 PDT 2023
;; MSG SIZE  rcvd: 93

(base) Faizs-MacBook-Pro:~ faiz$
```

Finally from our analysis we were found out that the attachment was named SWT_#09674321____PDF__.CAB. From there on we used the hybrid analysis to find out the
SHA256: 2e91c533615a9bb8929ac4bb76707b2444597ce063d84a4b33525e25074fff3f
File Extension: RAR



From our analysis we can infer that the file attachment was indeed malicious , marked as a Trojan.Generic



Finally we utilized virus total to find the exact file size of the pdf attachment (SWT_#09674321____PDF__.CAB) plugging in the SHA-256 we discovered on hybrid analysis.
We discovered that the file size of the attachment was actually 400.26 KB