

# Malware Analysis Proof of Concept (PoC)

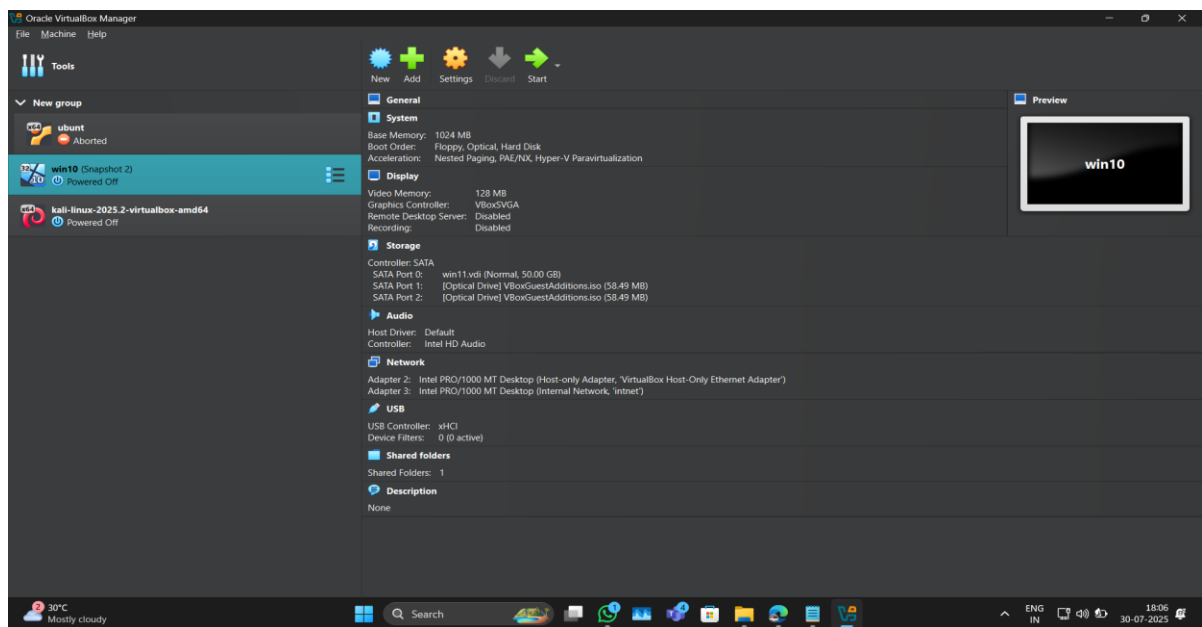
## Lab Setup

### Isolated Lab Environment:

1. **Host OS:** Windows 10 64-bit (for virtualization and sharing tools)
2. **VM Software:** Oracle VirtualBox
3. **VM1 (Attacker):** Kali Linux
4. **VM2 (Victim):** Windows 10 32-bit (Isolated and without internet)

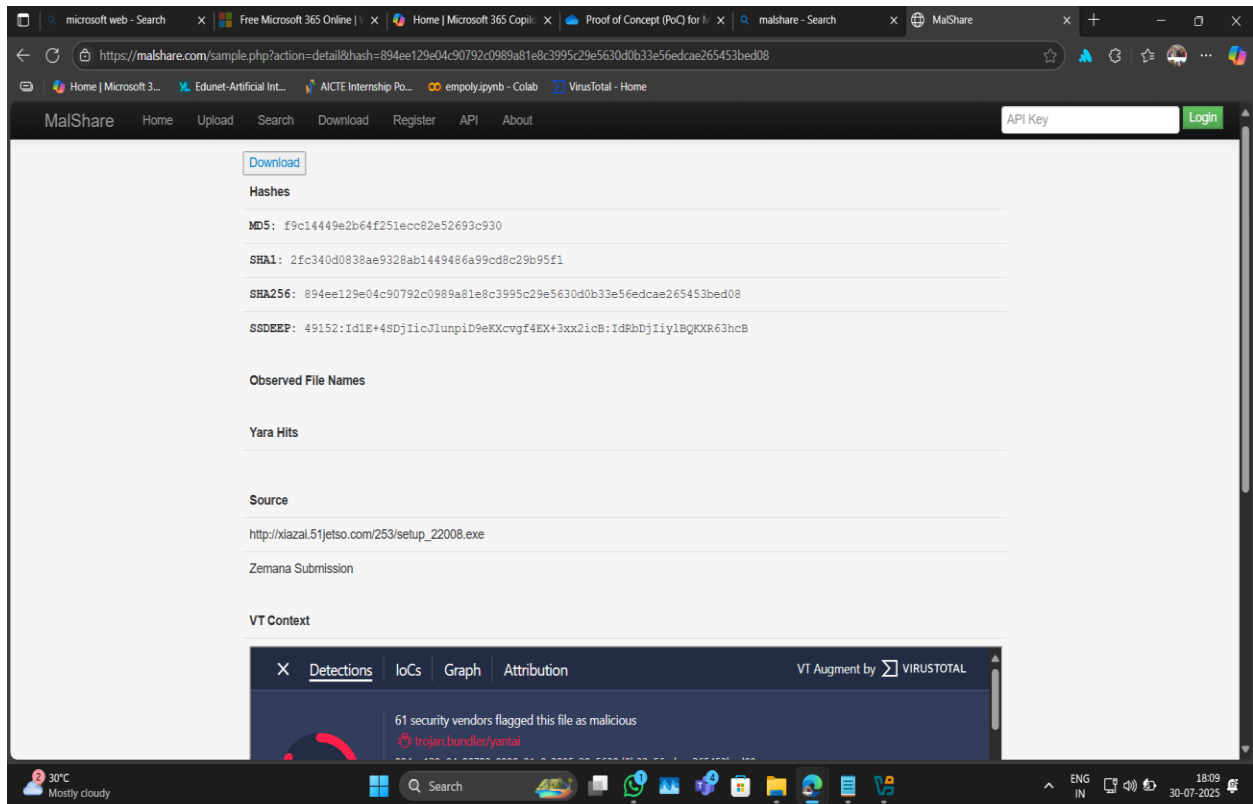
### Steps:

- Install Oracle VirtualBox
- Create VM for Kali Linux
- Create VM for Windows 10 32-bit
- Setup **Host-Only Adapter** on both machines for internal communication



# Malware Sample Acquisition

1. Go to [malshare.com](https://malshare.com)
2. Search for hash:  
894ee129e04c90792c0989a81e8c3995c29e5630d0b33e56edcae265453bed08
3. Download file (saved as .vir extension)



4. Start Python server in Kali:

```
python3 -m http.server 8888
```

5. On Win10 VM:
  - a. Open browser > <http://<Kali-IP>:8888>
  - b. Download malware sample

# Tools Setup

## Tools to Use (All downloaded in Host Windows):

- Autoruns
- Process Explorer
- Process Monitor
- RegShot
- DIE (Detect It Easy)
- Resource Hacker
- Strings (Sysinternals)
- TDSSKiller
- WinHex

Note: all tools zip file - [https://github.com/Nikhil2604Saini/Digisuraksha-parhari-foundation/tree/main/Cybersecurity-Internship-Program-2025/Week-2\\_Malware-IOC-APT28/Malware-Reports/malware%20analysis%20tool](https://github.com/Nikhil2604Saini/Digisuraksha-parhari-foundation/tree/main/Cybersecurity-Internship-Program-2025/Week-2_Malware-IOC-APT28/Malware-Reports/malware%20analysis%20tool)

## Share Tools:

- Go to VirtualBox settings
- Devices > Shared Folder > Add tools folder as shared
- Access from Z: drive or \\VBOXSVR\\<folder> in VM

# Static Analysis

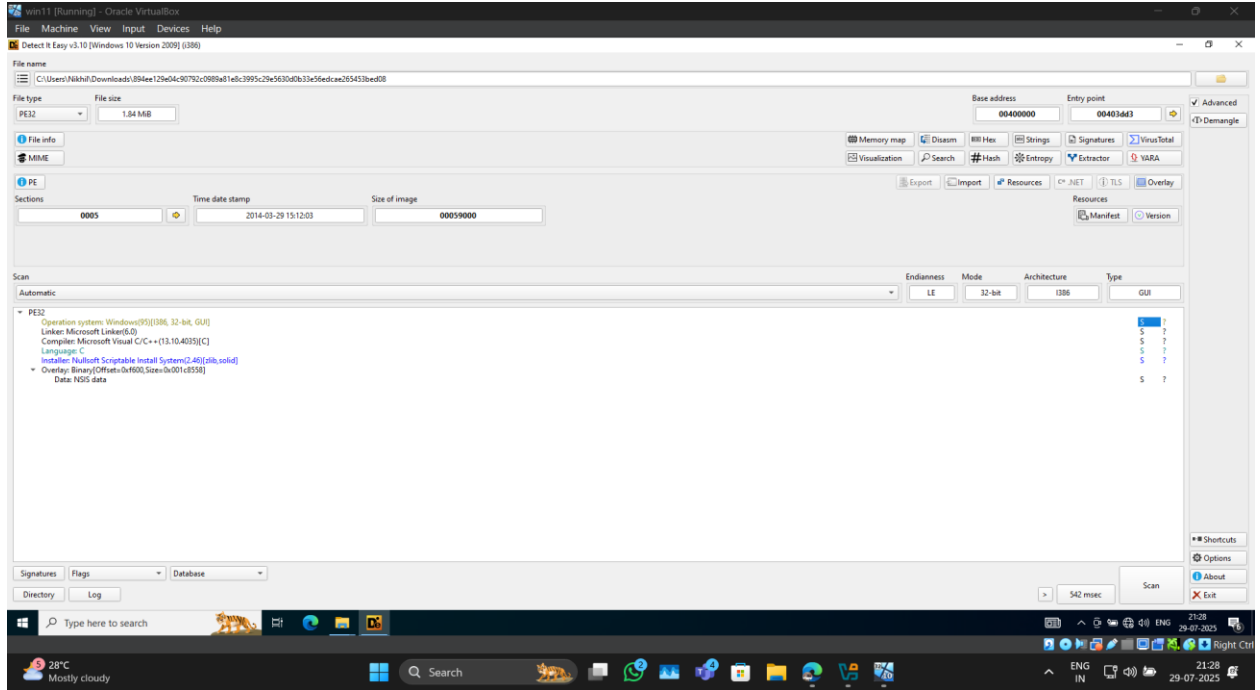
## Tools Used, Explanation and Steps:

### 1. DIE (Detect It Easy):

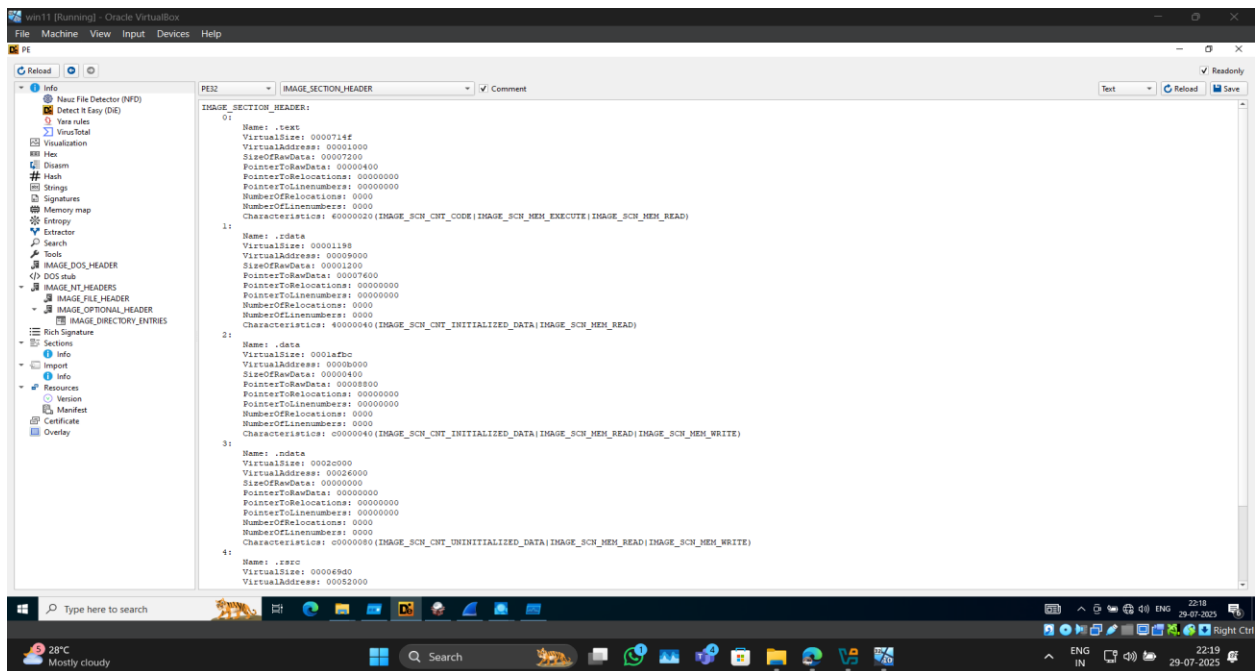
- a. Purpose: Identifies if a binary is packed or obfuscated, shows compiler info.
- b. Steps:
  - i. Open DIE.
  - ii. Click 'File' > 'Open' and select malware .exe.
  - iii. Review file type, entropy, packer and compiler info.

- c. What We Learn: Detects packers (e.g., UPX), entropy (high = encrypted), compiler used.

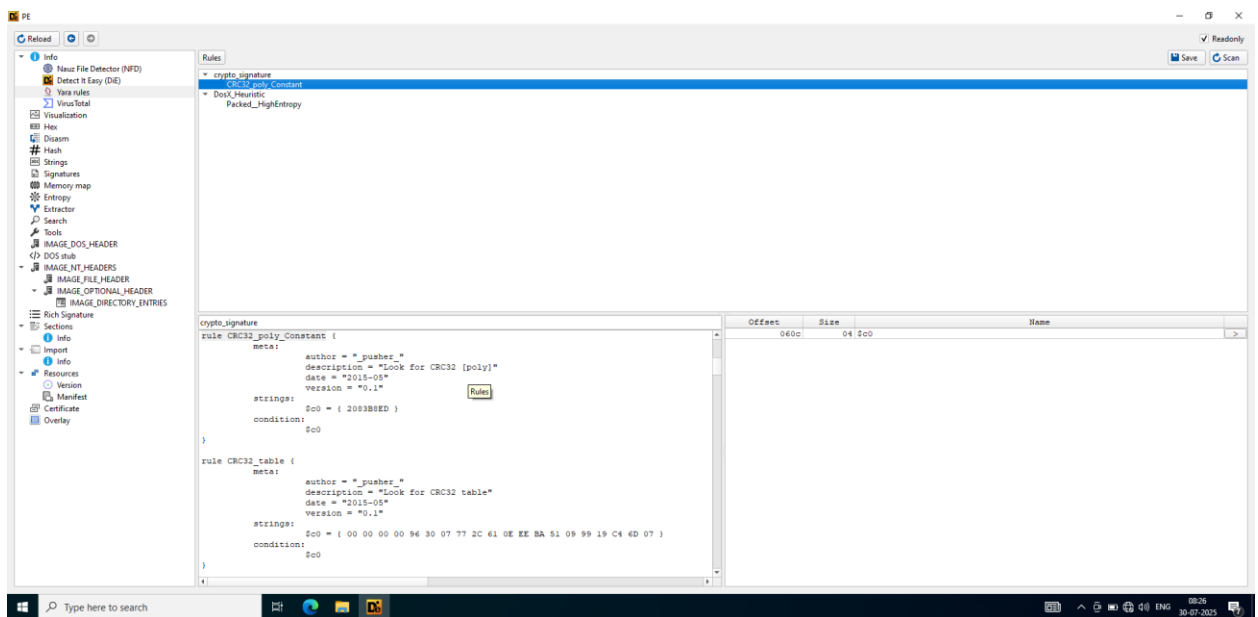
## 1. File Information( file type: PE32, Bass add.:00400000 , Entry point: 00403dd3)



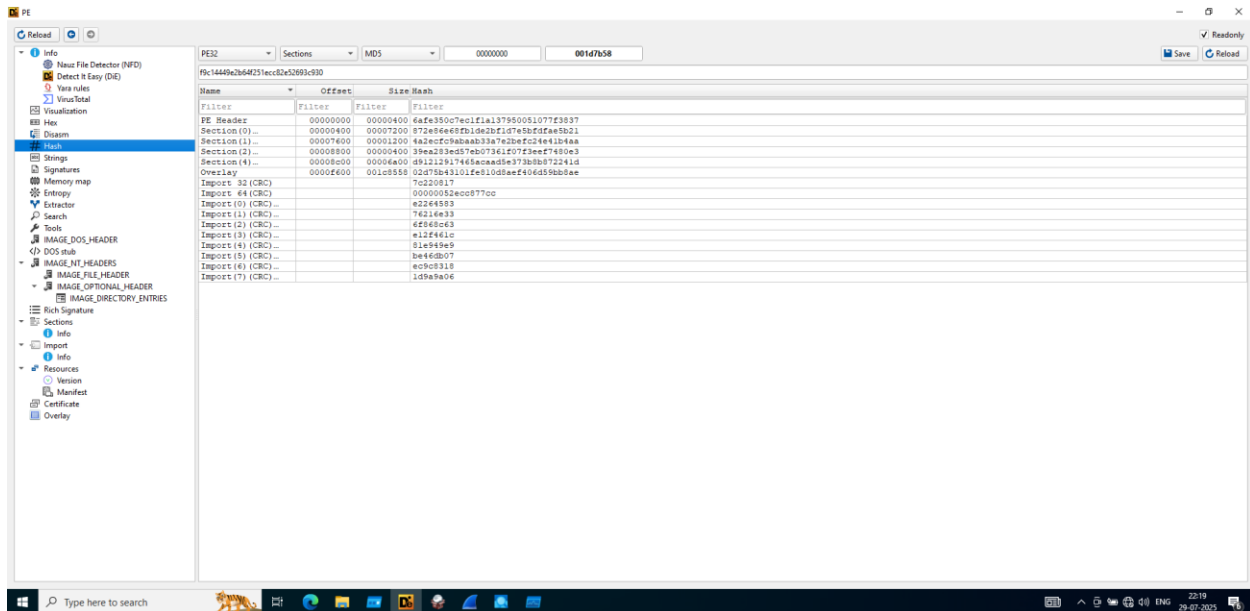
## 2. Section Header (.text, .rdata, .data, .ndata, .rsrc).



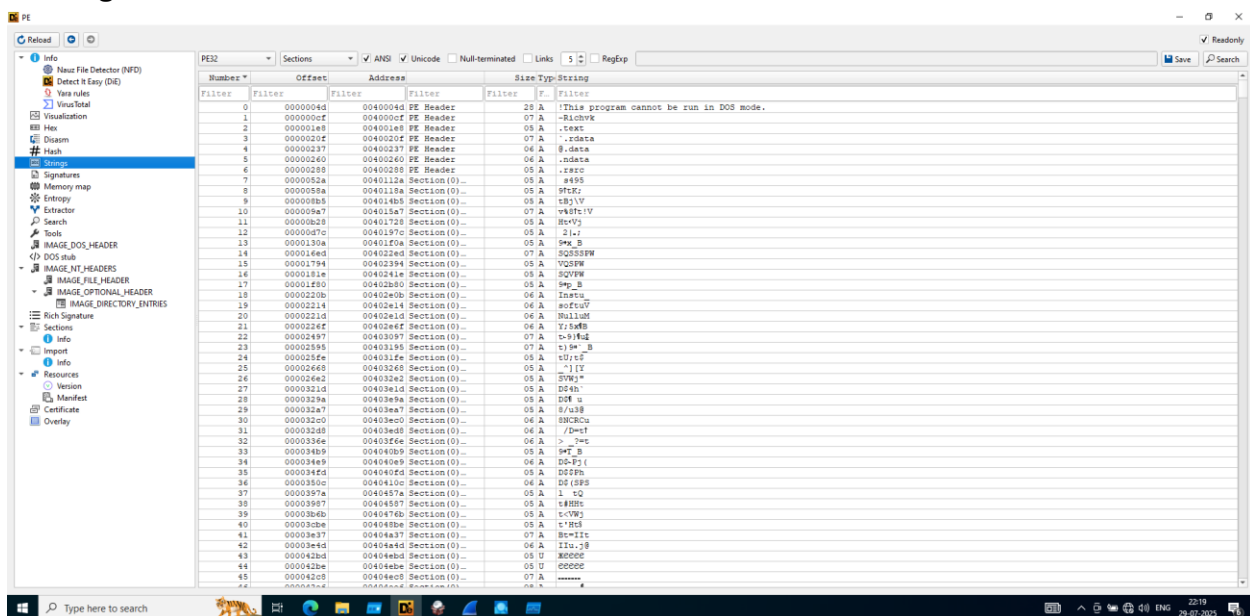
## 3. Yara rule



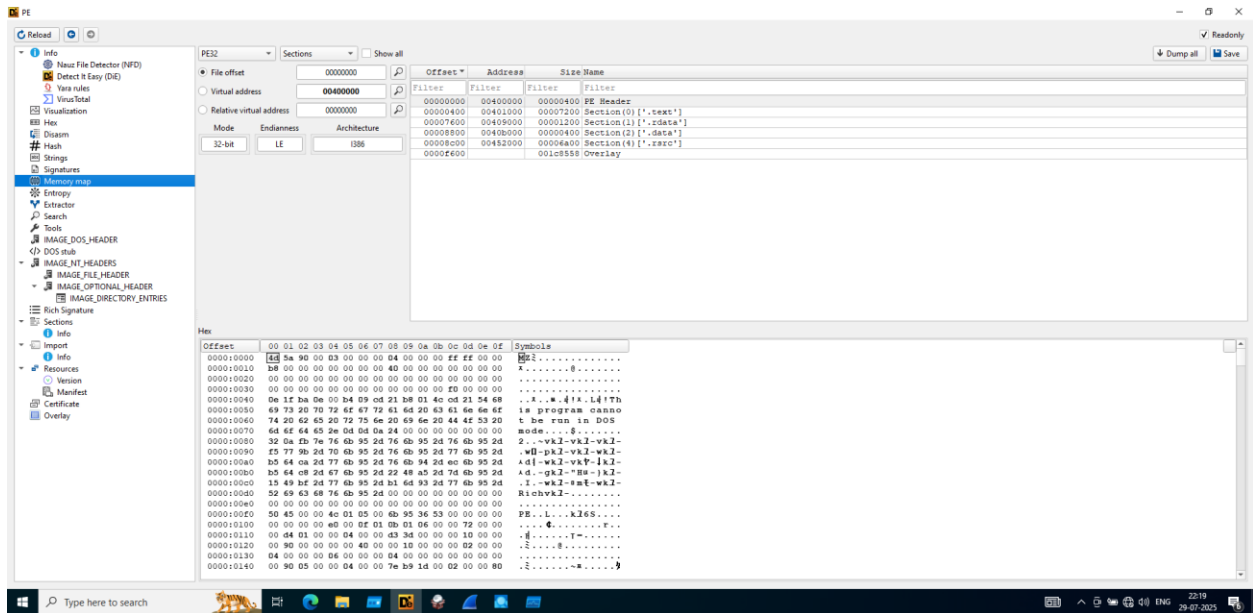
## 4. Hash



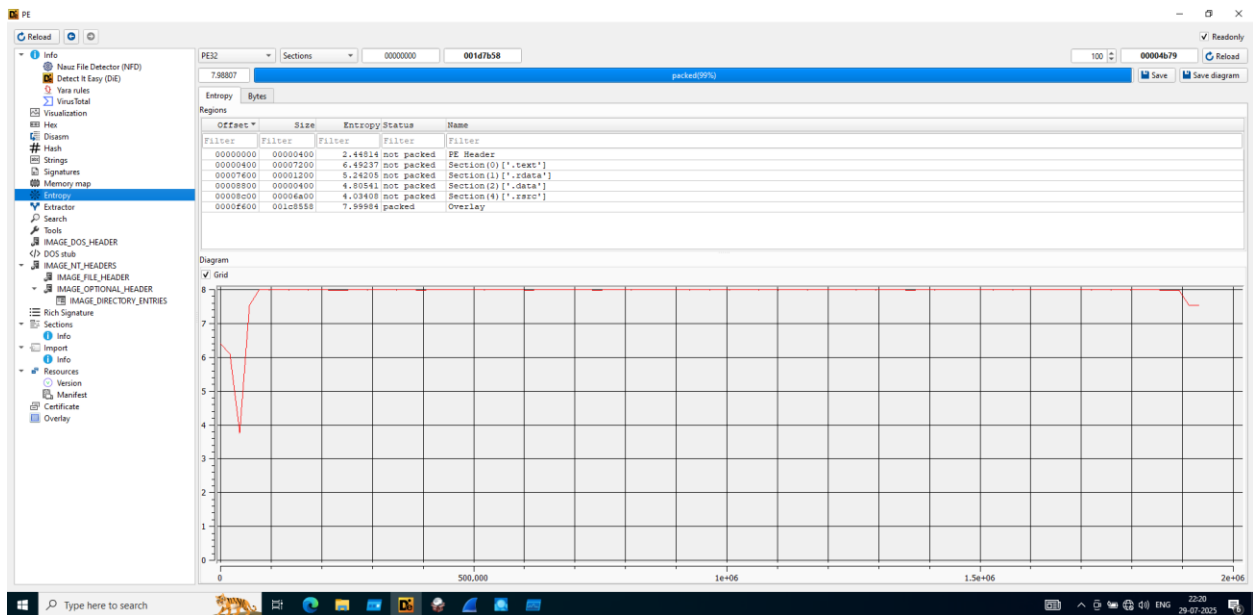
## 5.Strings



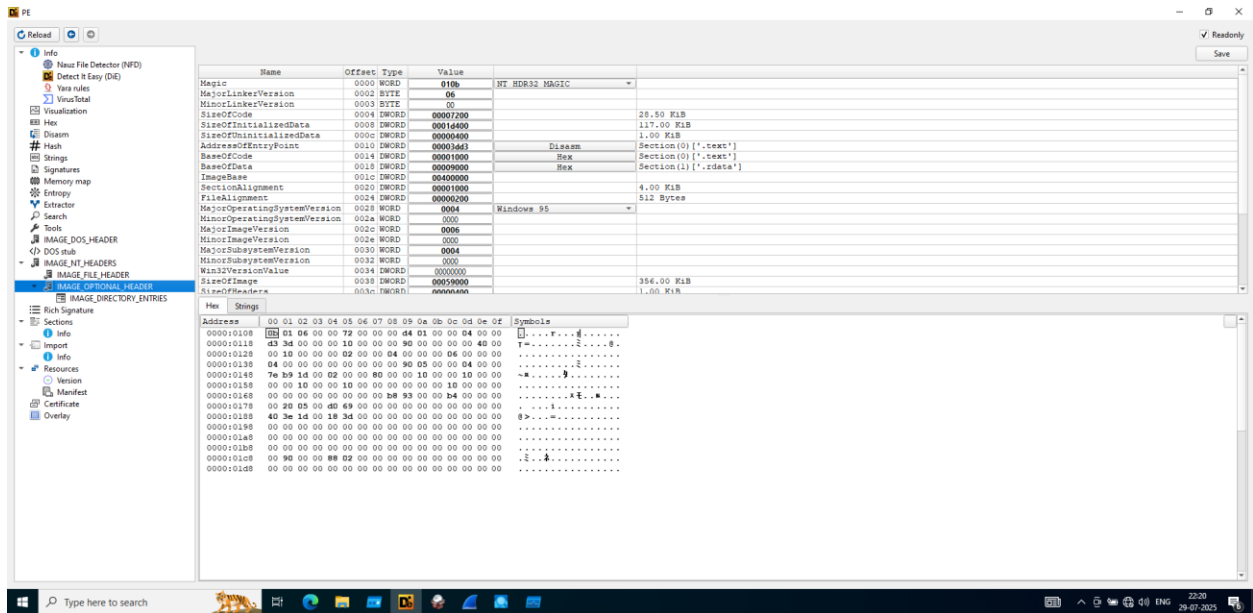
## 6.Memory mapping(Section address ,size ,name)



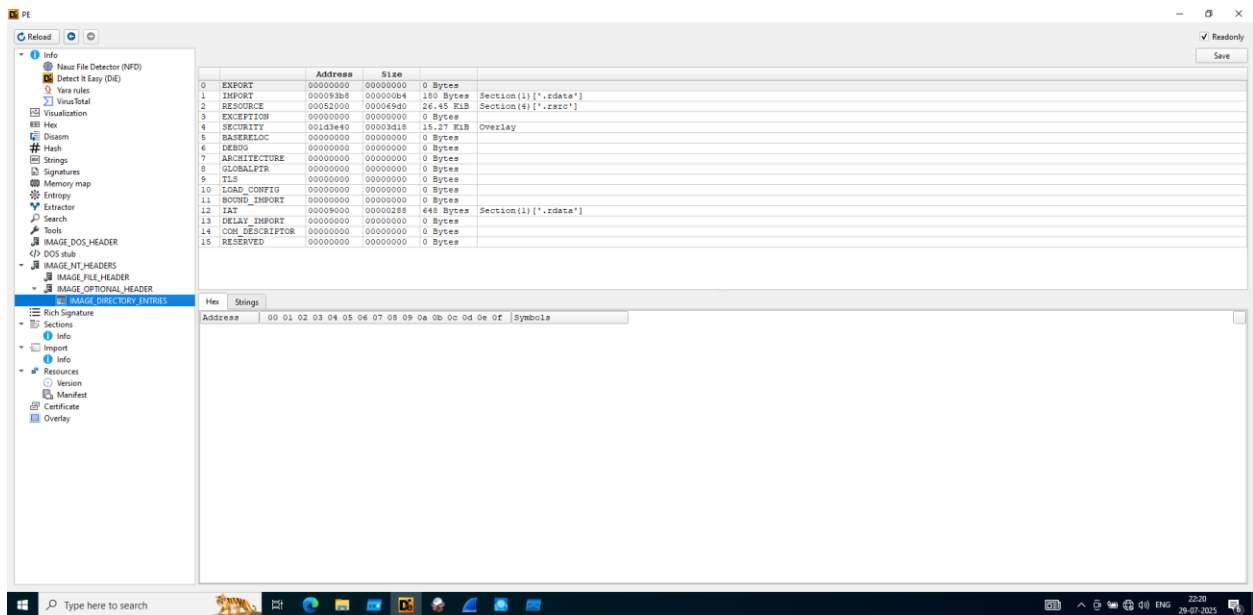
## 7.Entropy( high entropy)



## 8.Image optional Header

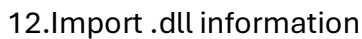


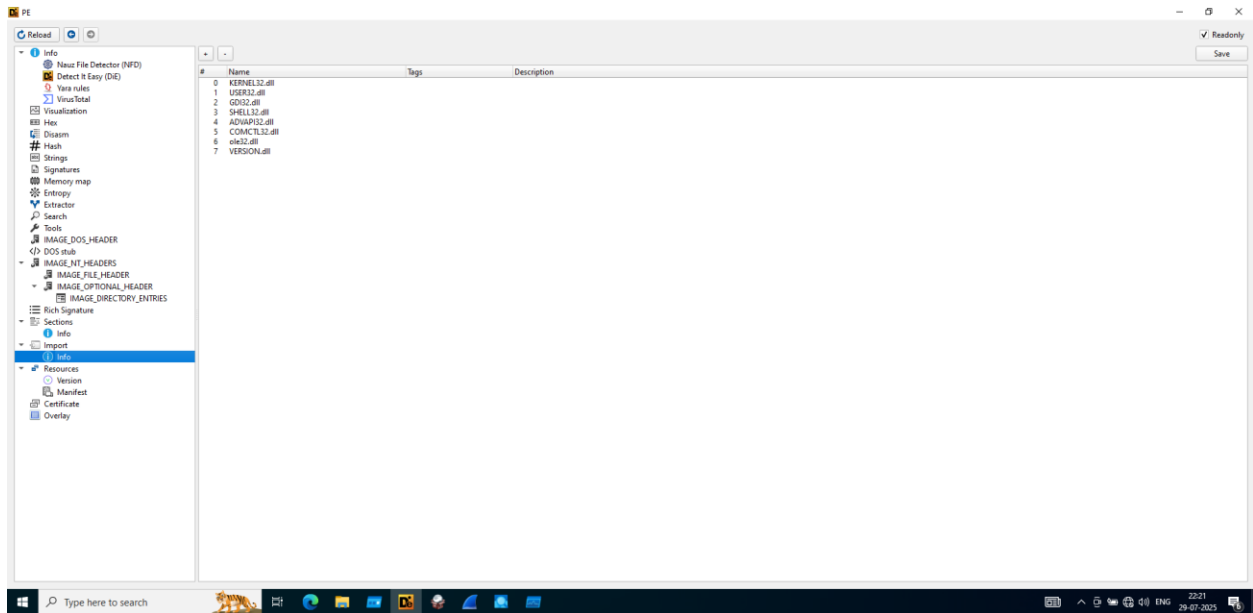
## 9. Image directory entry



## 10. Rich Signature





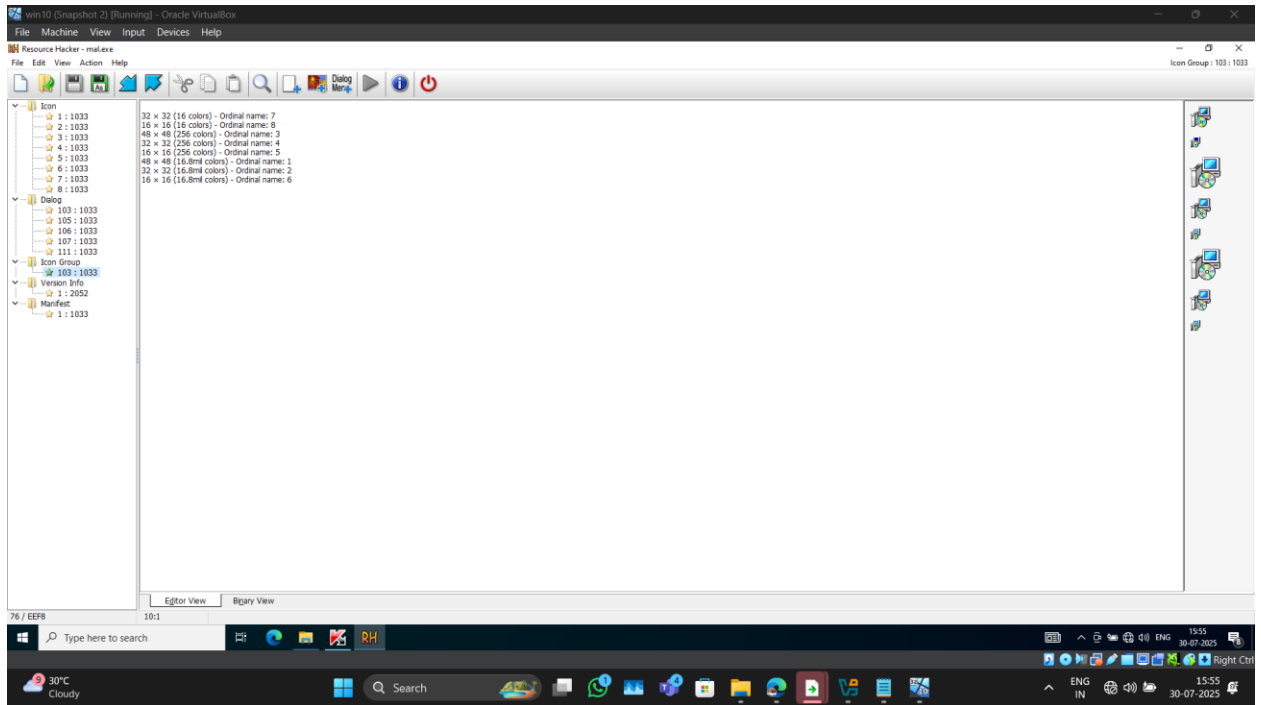


## 2. Strings (Sysinternals):

- Purpose: Extracts readable strings from binary files.
- Steps:
  - Copy `strings.exe` and malware sample in same folder.
  - Run command: `strings malware_sample.exe > output.txt`
  - Open `output.txt` to inspect extracted data.
- What We Learn: Reveals IPs, URLs, filenames, commands, suspicious keywords.

## 3. Resource Hacker:

- Purpose: Views and edits executable resources (icons, metadata).
- Steps:
  - Open Resource Hacker.
  - Load the malware file.
  - Browse through Version Info, Manifest, Icon, String Table.
- What We Learn: Fake company details, misleading info, embedded images or messages.



#### 4. WinHex:

- a. Purpose: Examines raw hexadecimal data of binary files.
- b. Steps:
  - i. Launch WinHex.
  - ii. Click File > Open > Load malware sample.
  - iii. Use Find (Ctrl+F) to search keywords like “http”, “cmd”, “exe”.
- c. What We Learn: Raw byte-level structure, signatures, and indicators not visible in high-level tools.

#### 5. Online Repositories (VirusTotal, MalwareBazaar, Vx-underground, Virus.exchange, Anu.run):

- a. Purpose: Online services for malware intelligence.
- b. Steps:
  - i. Search using malware hash.
  - ii. Check detection rate, behavior reports, sandbox results.
  - iii. Compare reports across platforms.
- c. What We Learn: Classification, threat level, network behavior, historical context.

61 / 72  
Community Score -107

61/72 security vendors flagged this file as malicious

Reanalyze Similar More

894ee129e04c90792c0989a81e8c3995c29e5630d0b33e56edcae265453bed08

Size 1.84 MB Last Analysis Date 47 minutes ago

peexe direct-cpu-clock-access detect-debug-environment long-sleeps signed nsis runtime-modules overlay

DETECTION DETAILS RELATIONS BEHAVIOR COMMUNITY

Display grouped sandbox reports

Sandbox	Verdict	Confidence	Score	Reasons
CAPA	Malicious	7	0	
CAPE Sandbox	Malicious	2	23	1
Microsoft Sysinternals	Malicious	0	50	12
Tencent HABO	Malicious	0	0	1
VirusTotal Jujubox	Malicious	0	3	2
Rising MOVES	Malicious	0	0	10
VirusTotal Cuckoofork	Malicious	0	0	5
Zenbox	Malicious	5	0	4

Activity Summary

Download Artifacts Full Reports Help

Activity Summary

Download Artifacts Full Reports Help

Memory Pattern Urls

- http://113.107.111.148:5051/Calendar/addDateAction.action?&mac=
- http://113.107.111.148:5051/Calendar/addDateAction.action?status=xz&address=www.
- http://113.107.111.148:8080/KuaiPing/addCalendarAction.action?&mac=%d.%d.%d.GetNativeSystemInfoe
- http://WebXml.com.cn/
- http://WebXml.com.cn/getWeather
- http://WebXml.com.cn/http://WebXml.com.cn/getSupportCityDatasetResponsegetSupportCityDatasetgetSuppo
- http://crl.thawte.com/ThawteTimestampingCA.crl0
- http://curl.haxx.se/docs/http-cookies.html
- http://hao.360.cn/?src=lm&sn=3f17941795
- http://hao.360.cn/?src=lm&sn=3f17941795Software

Behavior Similarity Hashes

Sandbox	Hash
CAPA	17255d8079115a792aa04b375466178
CAPE Sandbox	4d8db05b69762e0e7e199b502aa7207b
Microsoft Sysinternals	156888cab79cfb2657047c639ff9603
Tencent HABO	5f6a7f2abef0bcac2b8fcb7699150646
VirusTotal Jujubox	7be5b5f0f01cda55085d877c6d7a21f
Zenbox	2e10e55ef3f05fc7532922793ea67071

File system actions

Note: for more info. Refer [VirusTotal - File - 894ee129e04c90792c0989a81e8c3995c29e5630d0b33e56edcae265453bed08](https://www.virustotal.com/gui/file/894ee129e04c90792c0989a81e8c3995c29e5630d0b33e56edcae265453bed08/behavior)

# Dynamic Analysis

## Tools Used, Explanation and Steps:

### 1. Process Explorer (Sysinternals):

- a. Purpose: Lists active processes with detailed info.
- b. Steps:
  - i. Run as administrator.
  - ii. Go to Options > Enable 'Verify Image Signatures'.
  - iii. Sort by Company Name to highlight unsigned processes.
  - iv. Look for suspicious, unknown, or unsigned entries.
- c. What We Learn: Hidden processes, parent-child relations, memory usage.

### 2. Autoruns (Sysinternals):

- a. Purpose: Displays all auto-starting programs.
- b. Steps:
  - i. Launch Autoruns.
  - ii. Review 'Logon', 'Scheduled Tasks', 'Services', and 'Drivers' tabs.
  - iii. Highlight unfamiliar entries, check their paths and signatures.
- c. What We Learn: Persistence techniques used by malware.

### 3. Process Monitor (Sysinternals):

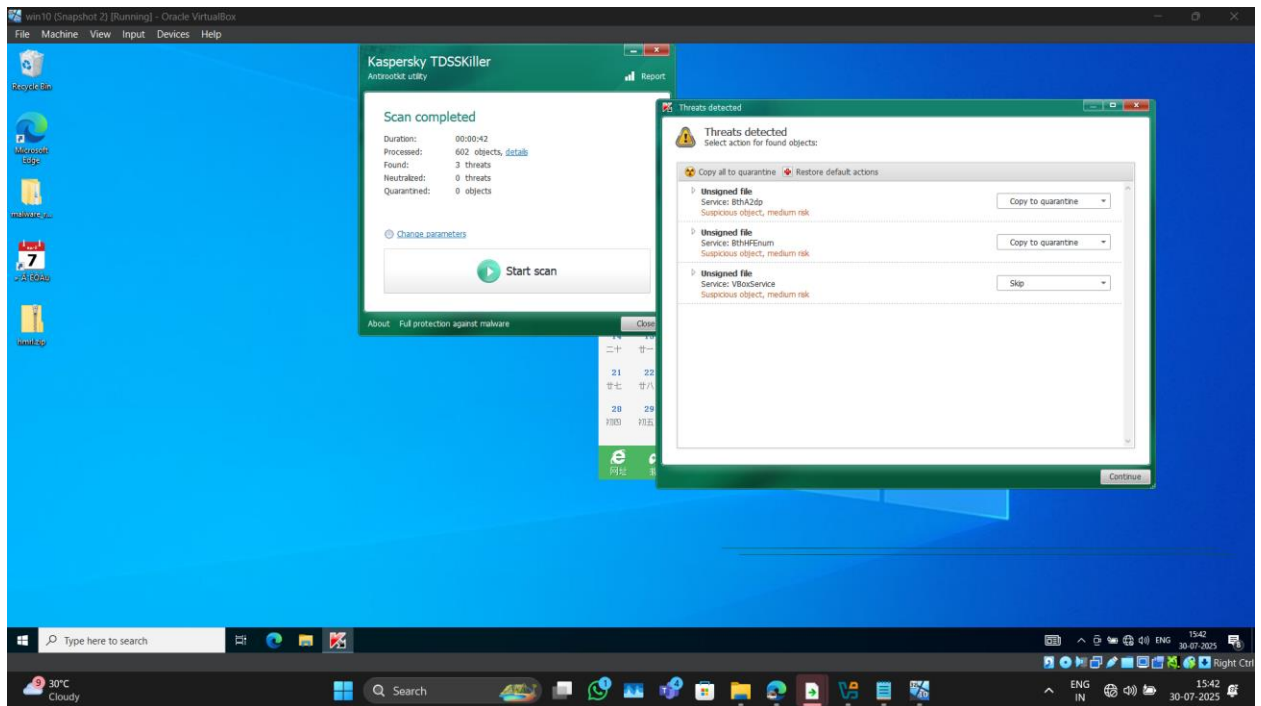
- a. Purpose: Captures real-time system activity.
- b. Steps:
  - i. Launch ProcMon, press Ctrl+E to start logging.
  - ii. Run the malware.
  - iii. Press Ctrl+E again to stop.
  - iv. Use filters (e.g., Process Name contains malware\_sample.exe).
- c. What We Learn: Registry, file, and process events triggered by malware.

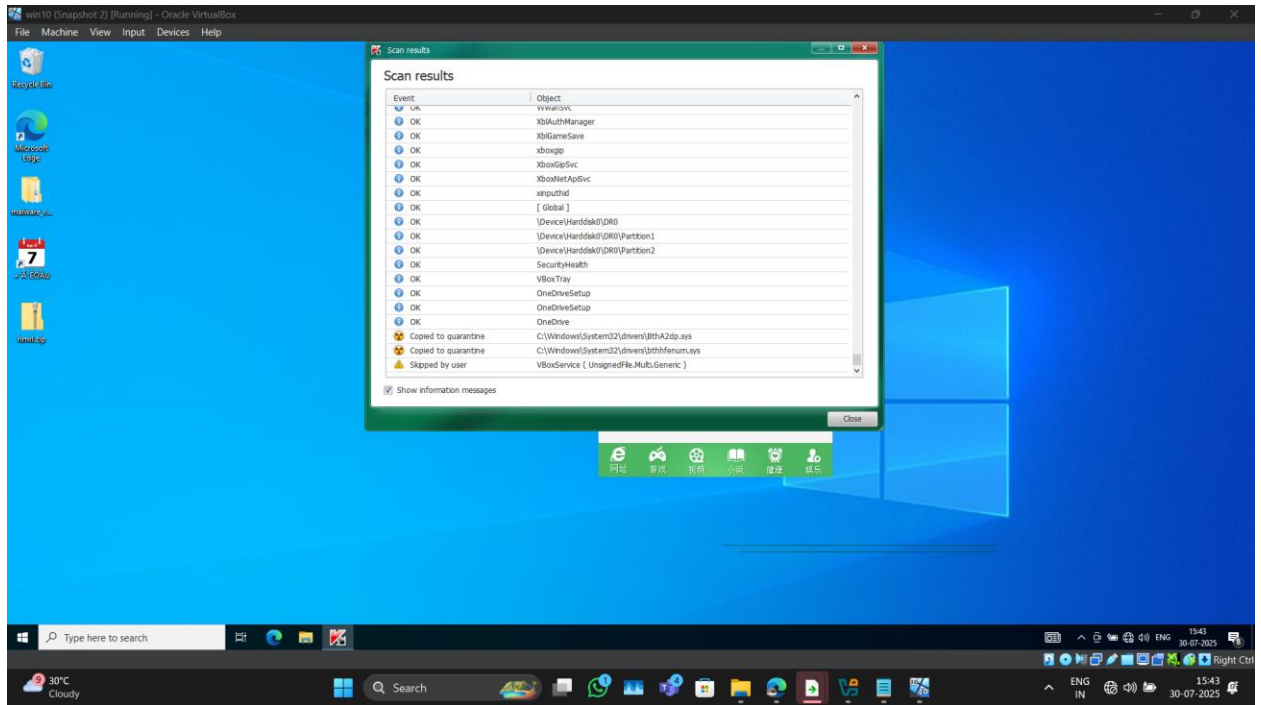
### 4. RegShot:

- a. Purpose: Compares registry before and after execution.
- b. Steps:
  - i. Open RegShot.
  - ii. Take Snapshot 1 before running malware.
  - iii. Run malware and wait a few minutes.
  - iv. Take Snapshot 2.
  - v. Click 'Compare' and analyze differences.
- c. What We Learn: New/modified registry keys indicating persistence or configuration changes.

## 5. TDSSKiller (Kaspersky):

- a. Purpose: Rootkit detection and removal.
- b. Steps:
  - i. Launch TDSSKiller.
  - ii. Accept terms and click 'Start Scan'.
  - iii. Wait for results.
  - iv. Quarantine or delete detected files.
- c. What We Learn: Finds hidden malicious drivers, kernel-level threats.





## File Transfer Recap

- Kali to Win10 VM:
  - Start server: `python3 -m http.server 8888`
  - Open <http://<Kali-IP>:8888> in Win10 VM browser
  - Download .vir file safely

Out files on :[https://github.com/Nikhil2604Saini/Digisuraksha-parhari-foundation/tree/main/Cybersecurity-Internship-Program-2025/Week-2\\_Malware-IOC-APT28/Malware-Reports/malware\\_tools\\_report](https://github.com/Nikhil2604Saini/Digisuraksha-parhari-foundation/tree/main/Cybersecurity-Internship-Program-2025/Week-2_Malware-IOC-APT28/Malware-Reports/malware_tools_report)