Computer Forensic Tool - Windows OS Artifact Finder

Prachiti Barge[1], Gauri Shewale[2], Meera Wadher[3], Prerna Jagesia[4]
[1,2,3,4]*Department of Computer Engineering and IT, Veermata Jijabai Technological Institute, Mumbai*

***Abstract*—This paper presents a Windows forensic artifact finder designed to assist investigators in extracting and analyzing critical system artifacts from memory, registry, disk, and network sources within the Windows OS environment. It leverages Volatility to extract crucial artifacts such as running processes, network connections, open files, and registry hives from memory dumps, uncovering hidden or tampered data. The integration of RegRipper allows the tool to extract valuable information from registry entries, providing insights into user behavior, system configuration, and potential malicious activities. This tool enables the investigator to extract crucial system information such as hidden directories, partition details and much more from the Windows file system. In addition to this, the tool also retrieves important network artifacts including DNS cache and routing tables. Along with this, the tool detects common network attacks and provides packet capture statistics and visualization. The tool also incorporates customizable artifact profiles and an intuitive UI, improving investigator efficiency by enabling rapid selection, analysis, and reporting of relevant artifacts. Overall, this work introduces a unified, customizable, and investigator-friendly forensic solution that enhances the reliability and completeness of digital investigations on Windows platforms.**

***Index Terms*—Disk Forensics, Memory Artifacts, Network Attacks, Operating System, Registry Forensics, Windows OS**

I. INTRODUCTION

The increasing sophistication and complexity of cyber threats demand robust and efficient methods for investigating compromised systems and identifying potential evidence. Windows 11, being one of the most widely used operating systems, presents a significant focus for forensic investigations. In such investigations, the identification and extraction of artifacts from various sources, including memory, disk, network, and registry, play a crucial role in understanding the attack vector, reconstructing events, and attributing actions to specific actors. There is not a single tool available in the market which is first of all free as well as integrating various technologies in one place to detect various artifacts. The paper will delve into various techniques and methodologies for artifact identification and extraction. It will discuss the utilization of memory forensics tools such as Volatility, which enables the analysis of volatile data to uncover hidden or tampered evidence. Additionally, disk forensics techniques will be explored, including partition details and hidden files listing. The network artifacts will be examined through the analysis of network logs, packet captures, and communication patterns to identify potential command and control channels or data exfiltration methods. Finally, registry analysis using tools like RegRipper will provide insights into user activity and system configurations.

II. LITERATURE REVIEW

This literature review is a thorough summary of studies on Windows OS, Windows OS Attacks and Computer Forensic Analysis. It provides a theoretical framework for the study that assists in clearly mentioning and defining the scope and objectives. By acknowledging the contributions of earlier researchers, we have designed the content of this section that basically highlights the Windows OS architecture, enlists the different types of attacks that can be done on the Windows OS and the computer forensic analysis based on the DFRWS Investigative Model.

Windows 10 architecture comprises several key components:

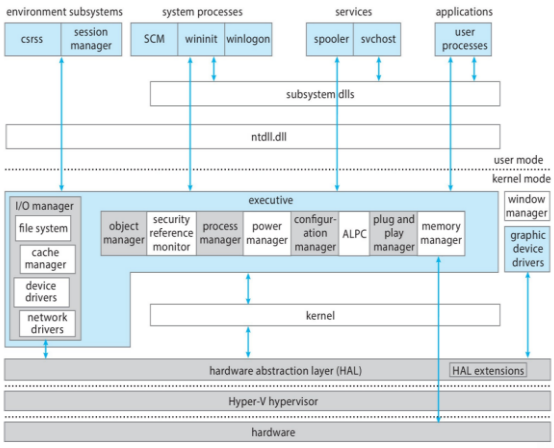


Fig 1 : Depiction of Windows 10 Architecture [1]

1. Kernel : The Windows NT kernel serves as the core, managing hardware and system operations, ensuring stability.

2. User and Kernel Modes : Windows 10 operates in user mode (applications) and kernel mode (OS), enhancing security and stability.

3. User Interface (UI) : The Windows Shell governs the graphical interface, encompassing the Start menu and window management.

4. Device Drivers : Drivers enable communication between hardware and the OS, supporting a wide range of devices.

5. Security Subsystems: Features like BitLocker, Windows Defender, and UAC ensure data protection and system security.

6. File System : NTFS offers encryption, compression, and journaling, maintaining data integrity.

7. Networking : Windows 10 supports various networking protocols, including Wi-Fi, VPN, and TCP/IP.

9. Application Support : Windows 10 runs Win32, UWP, and web-based applications, providing versatility.

10. Virtualization : Hyper-V enables multiple OS instances on the same hardware.

These elements create a versatile and secure environment for users, supporting a wide array of software and hardware while maintaining system stability and usability.

Despite security measures, there are always threats, tactics and techniques that can cause harm to digital systems. Our study mainly focuses on four key components of Windows OS - Memory, Disk, Registry & Network. The next subsection briefs us with some of the common attacks on these components.

1) Attacks on Windows 10

a) Memory Based Attacks

The following section lists the common attacks that can be performed on the memory module of the Windows subsystem.

1. Dump LSASS.exe Memory using direct system calls and API unhooking

Local Security Authority Subsystem Service (Lsass.exe) is the process on an Active Directory domain controller. It's responsible for providing Active Directory database lookups, authentication, and replication. It stores data, keys and critical information about the system. Because lsass.exe is a crucial system file, it is often faked by malware. The lsass.exe file used by Windows is located in the directory %\Windir%\System32. If it is running in any other location, the lsass.exe is most likely a virus, spyware, trojan or worm. The memory of lsass.exe is often dumped for offline credential theft attacks. This can be achieved using direct system calls and API unhooking in an effort to avoid detection[3].

2. DNS Cache Poisoning

DNS spoofing, also referred to as DNS cache poisoning, is a form of computer security hacking in which corrupt Domain Name System data is introduced into the DNS resolver's cache, causing the name server to return an incorrect result record, e.g. an IP address[4].

3. Process Memory Hollowing

Adversaries may inject malicious code into suspended and hollowed processes in order to evade process-based defenses. Process hollowing is a method of executing arbitrary code in the address space of a separate live process.

Process hollowing is commonly performed by creating a process in a suspended state then unmapping/hollowing its memory, which can then

be replaced with malicious code. A victim process can be created with native Windows API calls such as CreateProcess, which includes a flag to suspend the processes primary thread. At this point the process can be unmapped using APIs calls such as ZwUnmapViewOfSection or NtUnmapViewOfSection before being written to, realigned to the injected code, and resumed via VirtualAllocEx, WriteProcessMemory, SetThreadContext, then ResumeThread respectively[5].

4. Process Injection - Asynchronous Procedure Call
Adversaries may inject malicious code into processes via the asynchronous procedure call (APC) queue in order to evade process-based defenses as well as possibly elevate privileges. APC injection is a method of executing arbitrary code in the address space of a separate live process. APC injection is commonly performed by attaching malicious code to the APC Queue of a process's thread. Queued APC functions are executed when the thread enters an alterable state. A handle to an existing victim process is first created with native Windows API calls such as OpenThread[6]. At this point QueueUserAPC can be used to invoke a function (such as LoadLibraryA pointing to a malicious DLL)[7].

b) Disk Based Attacks
The following section lists the common attacks that can be performed on the disk module of the Windows subsystem.

1. Hide Artifacts: NTFS File Attributes
Adversaries may use NTFS file attributes to hide their malicious data in order to evade detection. Every New Technology File System (NTFS) formatted partition contains a Master File Table (MFT) that maintains a record for every file/directory on the partition. Within MFT entries are file attributes, such as Extended Attributes (EA) and Data [known as Alternate Data Streams (ADSs) when more than one Data attribute is present], that can be used to store arbitrary data (and even complete files). Adversaries may store malicious data or binaries in file attribute metadata instead of directly in files. This may be done to evade some defenses, such as static indicator scanning tools and anti-virus[8].

2. Masquerading: Rename System Utilities
Adversaries may rename legitimate system utilities to try to evade security mechanisms concerning the usage of those utilities. Security monitoring and control mechanisms may be in place for system utilities adversaries are capable of abusing. It may be possible to bypass those security mechanisms by renaming the utility prior to utilization (ex: rename rundll32.exe). An alternative case occurs when a legitimate utility is copied or moved to a different directory and renamed to avoid detections based on system utilities executing from non-standard paths[9].

c) Registry Based Attacks
The following section lists the common attacks that can be performed on the registry module of the Windows subsystem.

1. Registry Malware
Malicious software can modify or inject entries into the registry to achieve various objectives, such as persistence, privilege escalation, or executing malicious code. This can be done through techniques like registry key hijacking, where malware replaces or modifies legitimate registry entries to gain control over specific system functions.

2. Registry Poisoning
Attackers can manipulate registry values or keys to trick legitimate applications or the operating system into behaving unexpectedly. This can lead to system instability, crashes, or even privilege escalation, depending on the vulnerability being exploited.

3. Registry Denial-of-Service (DoS)
Attackers may flood the registry with a massive number of entries, overwhelming its capacity and causing performance issues or crashing the system. This can disrupt normal system operation and deny legitimate users access to the affected system.

4. Registry Access Control Attacks
If an attacker gains unauthorized access to modify the registry, they can alter critical settings or permissions, which can lead to the compromise of system security. For example, an attacker may modify registry permissions to grant themselves elevated privileges or to disable security mechanisms.

d) Network Based Attacks

The following section lists the common attacks that can be performed on the network module of the Windows subsystem.

1. DOS (Denial of Service)

A Denial-of-Service attack aims to overwhelm a system, network, or service to make it inaccessible to legitimate users. Attackers flood the target with an excessive amount of traffic or requests, consuming system resources or exploiting vulnerabilities. This can result in system crashes, slowdowns, or unresponsiveness, ultimately denying access to authorized users[10].

2. U2R (User to Root)

A User to Root attack involves an unauthorized user escalating their privileges from a normal user account to gain administrative or root-level access on a system. Attackers exploit vulnerabilities or weaknesses in software or operating systems to elevate their privileges and gain unauthorized control over the target system[11].

3. R2L (Remote to Local)

A Remote to Local attack refers to an attacker attempting to gain unauthorized access to a local system from a remote location. The attacker exploits vulnerabilities in network services or protocols to bypass security measures and gain unauthorized access to the target system. This type of attack typically targets weak points in remote access services or open network ports[12].

4. Probe

Probe attacks are part of the reconnaissance phase of an attack where the attacker scans a network or system to gather information about potential vulnerabilities and entry points. The attacker probes the target system by scanning for open ports, active services, or weaknesses in network configurations. The information collected during the probe phase helps the attacker identify potential attack vectors and plan subsequent steps of the attack[13].

B. Computer Forensic Analysis

Computer and network crimes such as attacks caused by computer virus or Trojan, network Infringement disputes, network frauds, online pornography and gambling are getting worse because of the wide penetration of computers and Internet which leads to tremendous economic loss and increasingly prominent social problems[14]. Computer forensics is an emerging research area that applies computer investigation and analysis techniques to help detection of these crimes and gathering of digital evidence suitable for presentation in courts. This new area combines the knowledge of information technology, forensics science, and law and gives rise to a number of interesting and challenging problems related to computer security and cryptography that are yet to be solved.

According to the Digital Forensic Research Workshop, here are the necessary steps that need to be

followed while performing android forensic analysis[15] :

1. Identification
2. Preservation
3. Collection
4. Examination
5. Analysis
6. Presentation

Few computer platform tools used in the process of forensic analysis are as follows:

1) Autopsy

Autopsy® is the premier end-to-end open source digital forensics platform. Built by Basis Technology with the core features one can expect in commercial forensic tools, Autopsy is a fast, thorough, and efficient hard drive investigation solution that evolves with one's needs. The Sleuth Kit is a library and a collection of command-line tools used to investigate disk images. Autopsy is the GUI program for TSK. It is used by law enforcement, military, and corporate examiners to investigate the actions taken place on the evidence computer, however, it can be used to recover deleted data from digital devices tools[16].

2) Volatility

Volatility is a tool used by digital memory forensic experts. The Volatility Framework is open source and written in Python. It can be run on Windows, Linux, and Mac systems. It can analyze raw dumps, crash dumps, VMware dumps (.vmem), virtual box dumps, and many others[17].

3) RegRipper

RegRipper is an open-source forensic software tool used for extracting and analyzing registry hive files in Windows operating systems. The tool is designed to assist in the investigation of various aspects of a Windows system, including user profiles, installed software, network settings, user activities, USB device history, browser history,

and more. It enables forensic investigators to quickly extract and analyze registry data, helping to uncover evidence and gain insights into system activities and user behavior[18].

4) NetworkMiner

NetworkMiner is a network forensic analysis tool used to capture and analyze network traffic. It is designed to extract various artifacts and information from network packets, providing insights into network activities, protocols, hosts, and files exchanged over the network[19].

Each tool has its own usage when it comes to forensic analysis. Given below, we have examples of Autopsy and Volatility usages.

- Forensics using Autopsy

1. Listing all the accounts which accessed the computer

Path : Results → Operating System User Account → Table → Date Accessed (descending) → Username first row value

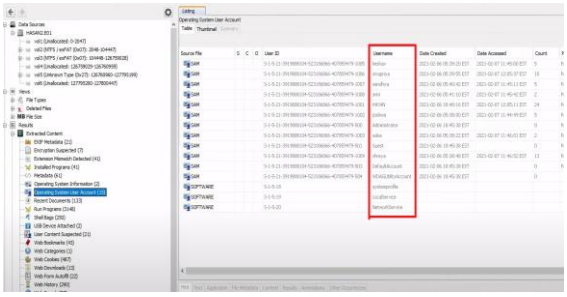


Fig 2: List of all accounts accessing computer

2. Powershell History for a user

Path : Data Sources → <Image name> → vol3 (NTFS / exFAT largest volume) → Users → <User name> → AppData → Roaming → Microsoft → Windows → Powershell → PSReadline → ConsoleHost_history.txt

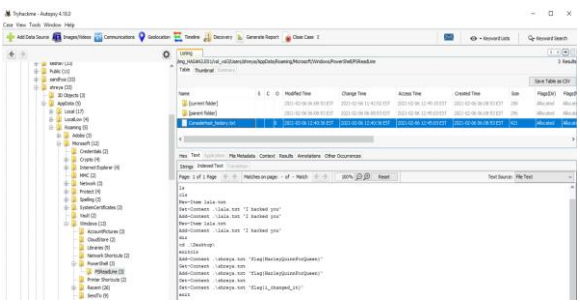
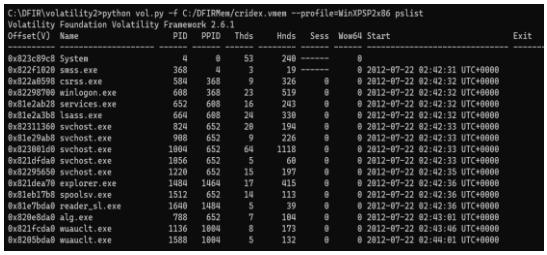


Fig 3: Powershell History of user

- Forensics using Volatility

1. Examining running processes from the memory dump

Command : python vol.py -f <memory_image_path> --profile=<profile> pslist

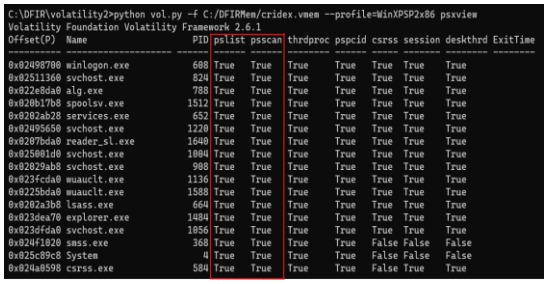


```
C:\DFIR\volatility2>python vol.py -f C:/DFIRMem/cridex.vmem --profile=WinXPSP2x86 pslist
Volatility Foundation Volatility Framework 2.6.1
Offset(V)  Name                    PID   PPID   Thds     Hnds   Sess  Wow64 Start                          Exit
---------- -------------------- ------ ------ ------ -------- ------ ------ ------------------------------ ------------------------------
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
```

Fig 4: List of Running Processes from memory dump

2. Checking for hidden processes

Command: python vol.py -f <memory_image_path> --profile=<profile> psxview



```
C:\DFIR\volatility2>python vol.py -f C:/DFIRMem/cridex.vmem --profile=WinXPSP2x86 psxview
Volatility Foundation Volatility Framework 2.6.1
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
```

Fig 5 : Checking for hidden processes

Our proposed tool has functionalities that can help us correlate attacks and their artifacts:

Component	Attack	Artifact
Network	Malware Traffic Analysis	The malicious IPs found acts as an important artifact for forensic investigation. One can check the abuse confidence score of the IP using the Check IP function in our tool.
Network	TCP SYN Flood	The Detect TCP SYN flood function in our tool reveals whether the attack has been performed and if yes, what are the IP addresses

		under suspicion.
Memory	Process Injection	The Process Scan function can help us find Processes that have occurred & originated from other processes.
Memory	DNS Cache Attack	The Cachedump function in our tool can help grab important cache hashes inside our registry.
Memory	Malware based Attacks	The Virus Total Scan function can help detect types of malware and malicious content using virus engines and website scanners.
Registry	Compromise of air gapped system using USB	The values of the BAM Key, Userassist key and USBSTOR key can be checked to get significant evidences of malicious activity
Registry	Run Key Persistence	The values of the Run and RunOnce keys can be checked to see if the attacker has used them for malware persistence
Disk	Hide Artifacts: NTFS File Attributes	The Alternate Data Streams which are hidden by nature can be detected using List Hidden Directories
Disk	Masquerading: Rename System Utilities	The Current Running Processes can be used to check which process is running currently with pids to disclose renamed system utilities

Table 1: Mapping of components, attacks and artifacts

III. PROPOSED TOOL

A. Problem statement

In the current era of advanced technology and pervasive digital presence, the need for computer forensic tools has become critical. These tools are essential for conducting comprehensive investigations, uncovering digital evidence, and analyzing cybercrimes. With the increasing sophistication of cyber threats and the widespread use of digital devices, computer forensic tools provide the means to identify, collect, preserve, and analyze digital artifacts, helping in the detection and prevention of cybercrimes. They assist in investigating data breaches, intellectual property theft, fraud, unauthorized access, and other digital crimes, ensuring the preservation of digital evidence and supporting legal proceedings in a technologically-driven world.

The objective is to develop an efficient and cost-effective Computer Forensic Tool by integrating functionalities from existing open-source tools. The tool should possess the capability to classify data as safe or malicious and present the findings through well-formatted reports. Additionally, it should feature an interactive and user-friendly interface, ensuring convenience and ease of use for investigators.

B. Proposed Windows Forensic Tool

Our tool is such that it extracts important artifacts from the registry, memory, network and disk. It combines the power of Volatility and RegRipper to enhance the efficiency and effectiveness of digital investigations. The tool aims to streamline the process of artifact identification and analysis in Windows systems, enabling investigators to extract valuable evidence for forensic examinations. The tool leverages Volatility, an open-source memory forensics framework, to perform in-depth analysis of volatile memory. It utilizes Volatility's comprehensive set of plugins to extract crucial artifacts such as running processes, network connections, open files, and registry hives from memory dumps. This enables the tool to uncover hidden or tampered data that might not be readily accessible through traditional file system analysis. Additionally, the tool integrates RegRipper, a renowned Windows registry analysis tool, to extract valuable information from registry hives. RegRipper's extensive set of plugins enables the tool to parse and interpret registry entries related to user activity, installed software, USB device history, and more. This allows investigators to uncover critical insights into user behavior, system configuration, and potential malicious activities.

The tool offers an intuitive user interface that facilitates artifact selection, analysis, and reporting, empowering investigators to efficiently navigate and interpret the extracted data. Furthermore, the tool supports customizable artifact profiles, enabling users to define specific artifacts of interest tailored to their investigation requirements.

Here's the GUI of our Proposed Tool.

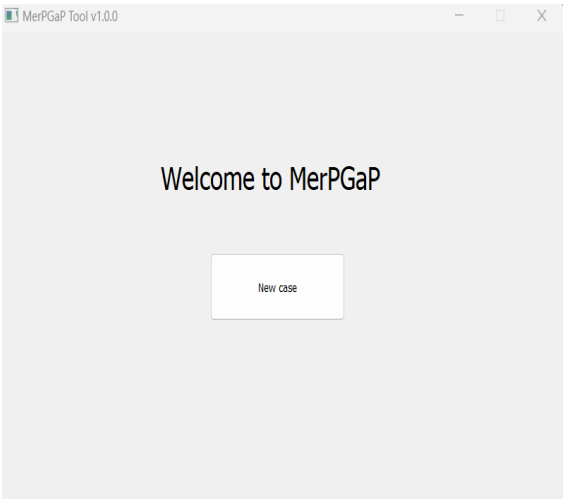


Fig 6 : Welcome Screen

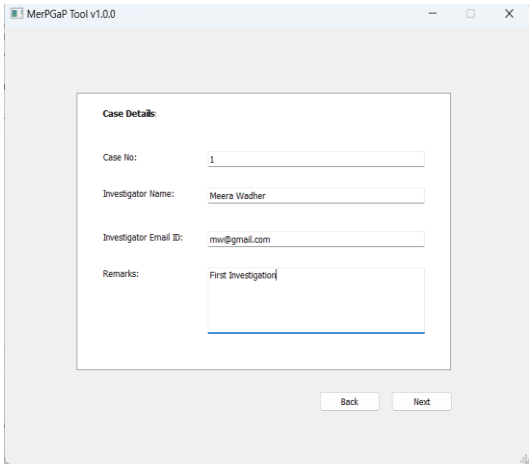


Fig 7 : Case Details Page

The tool consists of 4 modules for different types of forensics - Memory, Disk, Registry & Network. The Memory Module is applicable only for non-destructive data in the format of raw dump files and .mem files. The Virus Scan option in Memory Module gives an additional feature where uploaded file is checked on VirusTotal[20], a well-known website for suspicious file and website analysis. The Disk module runs on the live system to extract disk, filesystem related artifacts. The Registry Module can make hives of the current system and parse the generated hives to extract registry related artifacts. The Network Module displays the internet settings, ARP table, DNS cache,etc artifacts of the live system and provides an option for uploading a pcap file for analysis. The tool aids the investigator in detection of TCP SYN flood attack, UDP flood attack, ICMP flood attack in pcap file as well as provide visualization of stats. One additional feature is that it can also check Suspicious IP provided by the investigator with integration of AbuseIPDB[21].

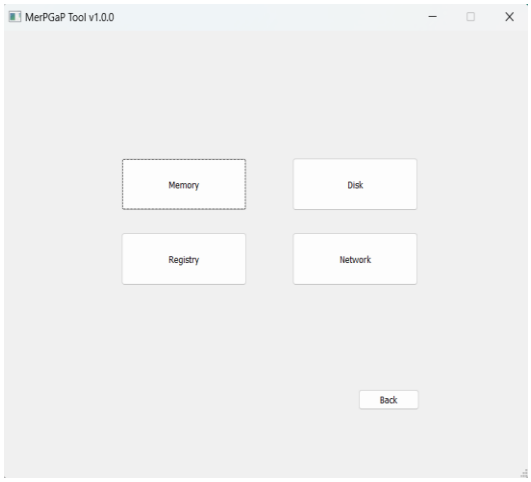


Fig 8 : Tool Menu

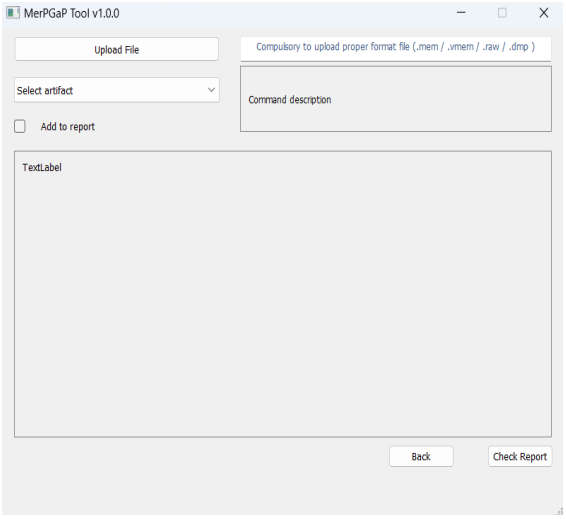


Fig 9 : Memory Module Interface

Here is a detailed view of the artifacts that are collected by our proposed tool:

Artifact Name	Description
System Artifacts	
PowerShell command history	Gathers PowerShell command history for all users
Windows Version	This command shows the Windows version present on the machine.
Basic Information	Displays basic information like Date and Time, OS Version, list of logged-in users and current user
Disk Artifacts	
Full File Listing	Shows the full file structure
Hidden Directories	Shows a list of all the hidden Directories and files
Hotfixes	Displays a list of hotfixes applied to the system
Driver Scan	Displays a list of installed device drivers
Shadow Copies	Displays existing shadow copies on the system. Shadow copies are point-in-time copies of files or volumes that are created by the Volume Shadow Copy Service (VSS).
User Accounts	Displays the user accounts on the system along with their last logon and status info
System Information	Shows live system information
Partition Details	Shows details about partitions present on the system
Event Logs	Gathers and parses Windows Event Logs. Shows details about windows events taking place on the system.
Memory Artifacts	
Windows Info	Shows OS & kernel details of the memory sample being analyzed
Memory Module Scan	Prints list of loaded modules
Kernel Module Scan	Prints list of kernel modules
File Scan	Scans for file objects present in a particular windows memory image
Driver Scan	Scans for drivers present in a particular windows memory image
Driver IRPs	List IRPs for drivers in a particular windows memory image
Mutexes	Scans for mutexes present in a particular windows memory image
Pool Scans	Pool tag scanning is a process commonly used in memory analysis in order to locate kernel

	object allocations, enabling investigators to discover evidence of artifacts that may have been freed or otherwise maliciously hidden from the operating system.
Process Scan	Scans for processes in a particular memory image
System Call Table	Table helps to check system call address from unexpected addresses
Symlink Scan	Finds links present in a particular windows memory image
Virtual Map	Lists virtual mapped sections
Page Pool	List big page pools
Service SIDs	Lists process token sids
SIDs	Print the SIDs owning each process
Token Privileges	Check for privileges tokens in unexpected services & list the processes using some privileged token
Environment Variables	Get the environment variables of each running process
Device Tree	Gives a tree based list on drivers and attached devices in a particular windows memory image
Driver Module	Gives driver module related information
Hash Dump	Grabs common windows hashes (SAM+SYSTEM)
Cache Dump	Grab domain cache hashes inside the registry
LSA Dump	Grabs LSASS secrets. LSASS secrets are sensitive security data that is stored in the Local Security Authority
	Subsystem Service (LSASS) process
Session Information	Prints Session information like session ID, session type, process ID, process user name, create time.
Service Scan	Lists services, offsets, order, PIDs, start, state, type, name, display, binary, etc.
Joblinks	Prints process job link information, like job session, process ID, joblink, process, PPID, active states etc
Callbacks	Lists kernel callbacks and notification routines.
DLL List	lists all DLLs for a given process in memory.
Network Artifacts	
Network information	Network configuration, routing tables, etc
Internet connections	Established network connections & settings
DNS cache entries	List of complete DNS cache contents
ARP table information	List of complete ARP cache contents
Routing Table	Used to show current routing table on live machine
Suspicious IPs	Lists of suspicious IPs on machines can be helpful for forensics
Detect Floods (TCP SYN, ICMP, UDP)	Used to identify network attacks
Display PCAP Statistics	Displays basic statistics of an input packet capture file

Display PCAP Visualization	Displays the packet count per protocol in the input packet capture file
Registry Artifacts	
Registry Key	Lists the registry keys under a hive or specific key value
UserAssist	Print userassist registry keys and information
Registry Certificates	Lists the certificates in the Registry's Certificate Store
Hivelist	Lists the registry hives present in a particular memory image, basically metadata of hives. Fields like offset, file full path, file output.
Hivescan	Similar to hivelist but hive scan command, as it can be used to filter the data and return only the rows that meet certain criteria.
Installed Apps	Lists all installed applications on the machine
User SIDs	Helps resolve SIDs to users on the machine
Autostart	Helps to determine autostart information for the machine
Typed Urls	Displays the most recently typed URLs in the Microsoft Edge Browser
System Environment Variables	Displays the environment variables set at the system level
User Environment Variables	Displays the environment variables set for the current logged on user
Hidden User Accounts	Displays the user accounts not visible during system start up
USB Devices	Displays the USB devices connected to the system

Table 2: Artifacts being collected by our tool

C. Algorithms

The algorithms used for developing the tool are listed below.

1) Autostart programs

Purpose : To extract data about programs set to autostart
Input : Software Hive
Output : Programs that run automatically on system boot
Algorithm
Step 1: Initialize list “keys” with Run and RunOnce as its items
Step 2: Initialize list “software_paths” with HKLM\Software\Microsoft\Windows\CurrentVersion\ and HKLM\Software\Wow6432Node\Microsoft\Windows\CurrentVersion\ as its items
Step 3: Iterate over software_paths list:
    Iterate over keys list:
        Append key to software_path
        Query the registry for the path generated
Step 4: Display the artifact details on the tool window
Step 5: If “Add to Report” is checked:
    Add artifacts to respective columns of report
  Else:
    Do not add
Step 6: Generate report

2) RecentDocs

Purpose : To extract data about documents
Input : NTUSER.dat Hive
Output : Documents recently accessed by the user sorted according to MRUList
Algorithm
  Step 1 : Provide the NTUSER.dat hive to the RegRipper tool to parse
  Step 2 : Use the recentdocs plugin to extract the relevant information
  Step 3 : Display the artifact details on the tool window
Step 4: If “Add to Report” is checked:
    Add artifacts to respective columns of report
  Else:
    Do not add

Step 5 : Generate report

3) Memory Module Scan
Purpose : To extract data about memory modules
Input : Memory dump file
Output : List of loaded memory modules with Base, Size, Name, Path, File Output details
Algorithm
Step 1 : Provide the memory dump file to Volatility to parse
Step 2 : Use the windows.modules plugin to extract the relevant information
Step 3 : Save the output to 'output.txt' file
Step 4 : Read this file to display the artifact details on the tool window
Step 5: If "Add to Report" is checked:
Add artifacts to respective columns of report
Else:
Do not add
Step 6: Generate report

4) VirusTotal Scan
Purpose : To extract data about the uploaded file using VirusTotal API
Input : File to be analyzed
Output : File categorized as malicious or harmless along with the engines that categorized them
Algorithm
Step 1 : Calculate the hash of the uploaded file
Step 2 : Query the /files endpoint of the VirusTotal API by using the hash calculated above
Step 3 : Extract the "last_analysis_results" from the response received into dictionary "dict_web"
Step 4 : Iterate over dict_web key-value pairs:
If dict_web[i][category] is malicious or suspicious:
Append "result" field to result_eng list
Append "engine_name" to engine_name list
Increment total_detect_count by 1
Step 5: If total_detect_count > 0:
File can be malicious
Else :
File is harmless and clean
Step 6: If "Add to Report" is checked:
Add artifacts to respective columns of report
Else:
Do not add
Step 7: Generate report

5) Check for suspicious IP
Purpose : To check if the given IP is malicious or not
Input : IP address to be checked
Output : IP categorized as malicious or safe along with the ConfidenceScore
Algorithm
Step 1 : Query the /check endpoint of abuseipdb api by providing the ipAddress to be checked as a query parameter
Step 2 : Extract the abuseConfidenceScore, countryCode, domain and ISP from the response received
Step 3 : If abuseConfidenceScore > 20 :
IP is malicious
Else:
IP seems to be safe
Step 4: If "Add to Report" is checked:
Add artifacts to respective columns of report
Else:
Do not add
Step 5: Generate report

6) Detect DOS attack
Purpose : To check if a DOS attack has been performed on the system
Input : pcap file of the traffic flowing in and out of the system
Output : Possibility of a DOS attack and suspicious IPs involved
Algorithm
Step 1: Extract packets from the pcap file using rdpcap
Step 2 : Filter for SYN/UDP/ICMP packets
Step 3 : Initialize empty source_ips dictionary
Step 4 : Iterate over the filtered packets:
If pkt.source_ip in source_ips dict:

source_ips[source_ip] += 1
Else:

source_ips[source_ip] = 1
Step 5 : Filter IPs with count > 100 as suspicious
Step 6 : If count of suspicious IPs > 0:
DOS attack detected
Else:
No attack detected

7) Disk Module Scan
Purpose : To extract data about system from disk
Input : Current Filesystem
Output : List of system related information such as system version, powershell history, application event logs, partition details, etc.
Algorithm
Step 1 : Select the desired artifact
Step 2 : Use the windows.api, plugin and python os library functions
to extract the relevant information
Step 3 : Save the output to 'output.txt' file
Step 4 : Read this file to display the artifact details on the tool window
Step 5: If "Add to Report" is checked:
Add artifacts to respective columns of report
Else:
Do not add
Step 6: Generate report

8) Report generation
Purpose : To extract generate report of all the desired artifacts
Input : Current Filesystem
Output : A text based report or HTML based report based on investigator's choice
Algorithm
Step 1 : Select the module for inspection
Step 2 : Upload file where necessity
Step 3 : Select the desired artifact
Step 4 : If "Add to Report" is checked:
Add artifacts to respective columns of report
Else:
Do not add
Step 5: Select Check Report option
Step 6: Preview the stored artifacts in the tool display window
Step 7: Click on "Save As" option to save text format report
Step 8: Click on "Generate HTML Report"
Iterate over artifact_name, artifact_value in report_data:
Add artifact_name in Sidebar of HTML report
Add artifact_value in MainContent tab of HTML report
Step 9: Save the HTML Report

D. Software Architecture

1) Component Diagram

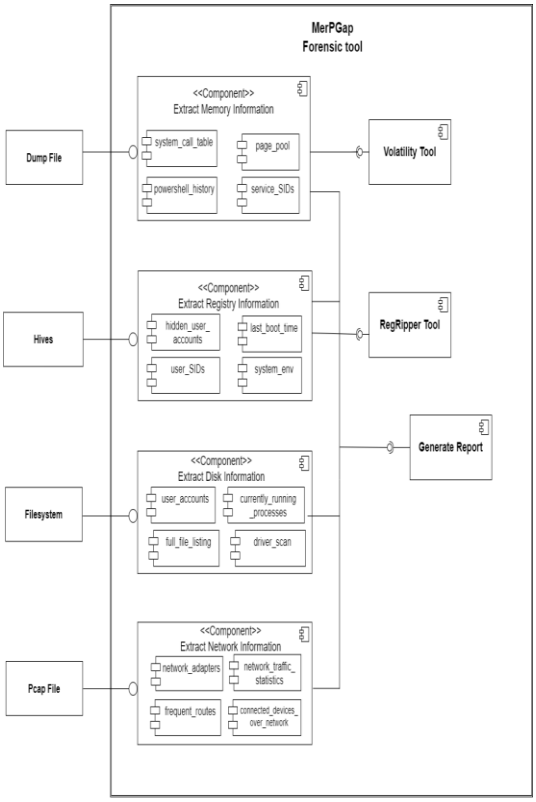


Fig 10 : Component Diagram

2) Deployment Diagram

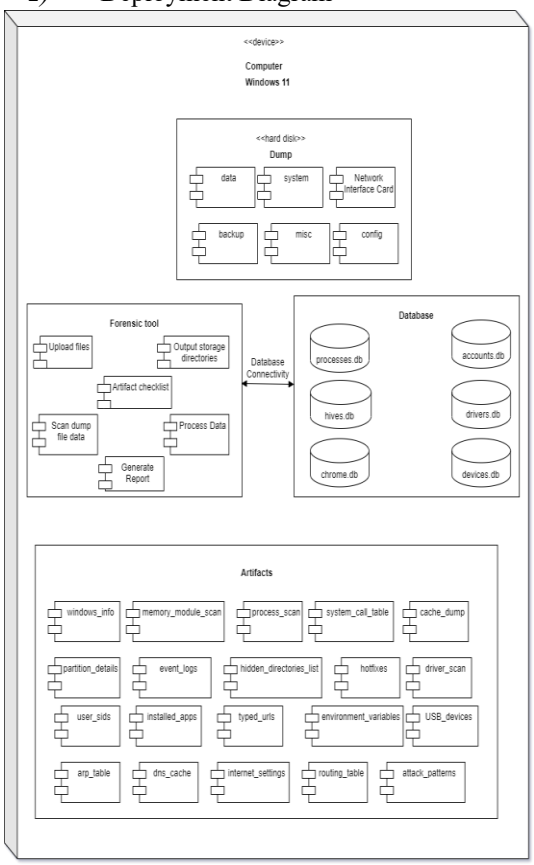


Fig 11 : Deployment Diagram

3) Data Structure Design

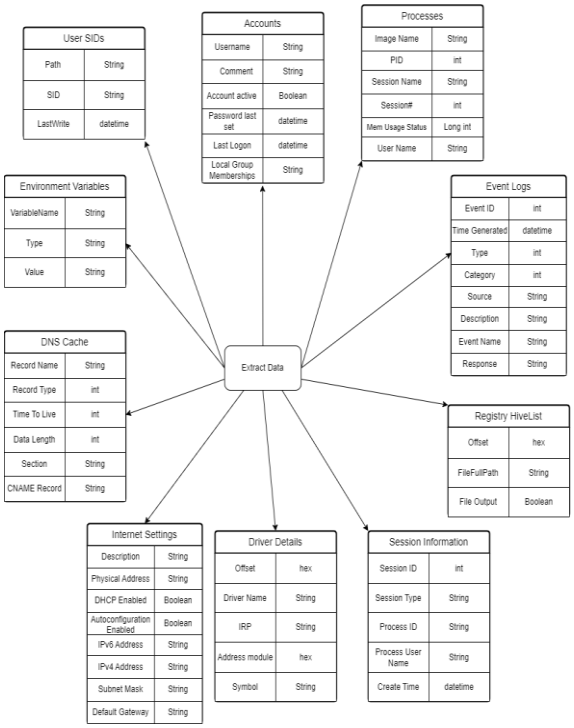


Fig 12 : Data Structure Design of the proposed tool

IV. IMPLEMENTATION

A. Experimental Setup

We performed the forensic analysis on the Windows 10 laptop where Forensic tool is running

1) Hardware Setup

TABLE I. HARDWARE REQUIREMENTS

RAM	8 GB (min)
Processor	Intel i5 8th gen, 2.3 GHz
OS	Windows 10, 11
Hard Disk	128 GB

2) Software Setup

TABLE II. SOFTWARE REQUIREMENTS

IDE	Visual Studio Code
Language	Python
Python Libraries	Pyqt5, Volatility, RegRipper
Command-Line Interpreter	Windows Command Prompt
Browser	Google Chrome, Edge, FireFox

B. Environmental Setup

Download the tool from the github repository[22] on the investigator's computer. Copy it to a USB device and transfer it to the victim's computer. This ensures the evidence on the victim's computer is not tampered.

C. Execution

1) Memory Module Snapshots

Our tool is used to do analysis of memory dumps. For this, we have provided a button to Upload File. The file can only be in the. mem, .vmem,. raw or .dmp formats.

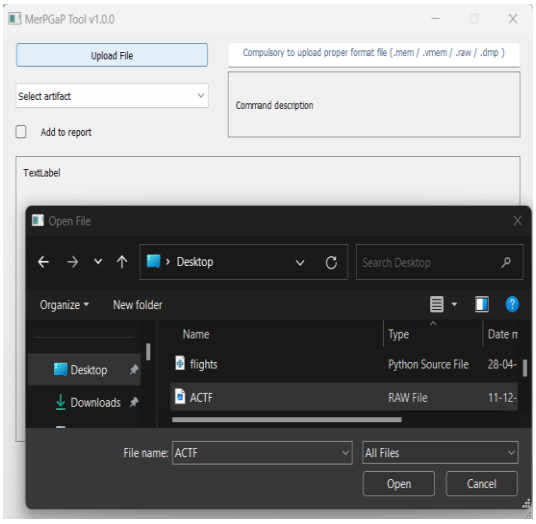


Fig 13 : File upload dialog box

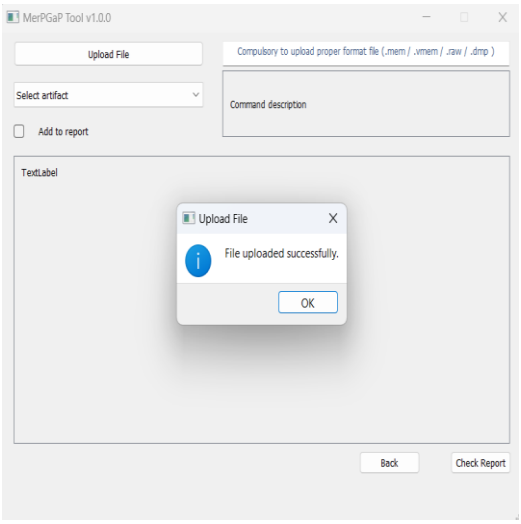


Fig 14 : Successful file upload message box

Running a few functions like Process List, Session Information & Driver Scan.

Session Information option gives us information about sessions, their time of creation, the type of session, process id, etc.

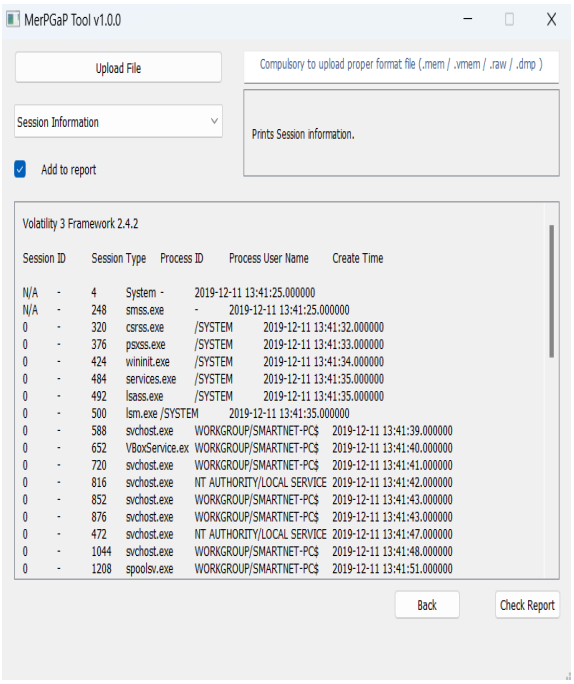


Fig 15 : Session Information Output

Driver IRP Scan option gives us information of IRPs for drivers in a particular windows memory image.

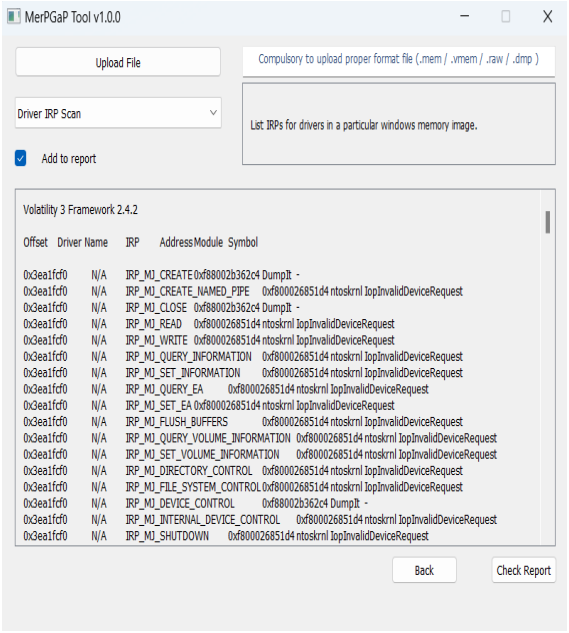


Fig 16 : Driver IRP Scan Output

Process List option gives us information about processes, their parent processes, create time, exit time, etc.

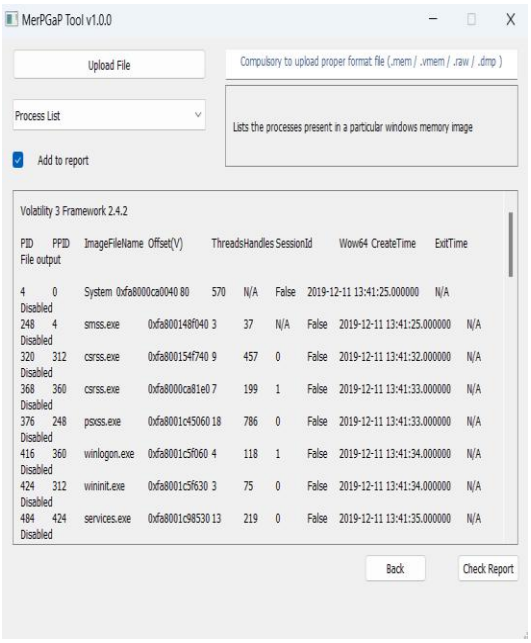


Fig 17 : Process List Output

We can add these artifact details to the report and view & save the report as a text file. We can choose to opt for a HTML type of report as well, which can be opened in our browser.

Generation of Reports

Text Report can be saved as a .txt file.

Fig 18 : Text format Report

HTML Report can be used to view our report in a formatted way, with tabs and sections. It can be opened in a browser.

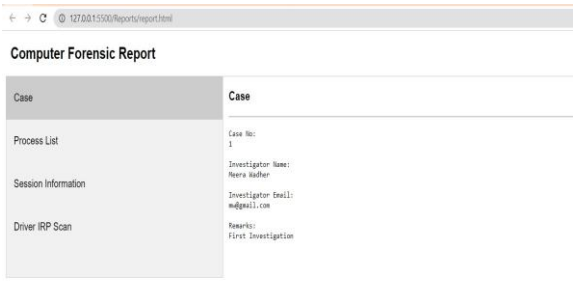


Fig 19 : HTML Report - Case Pane

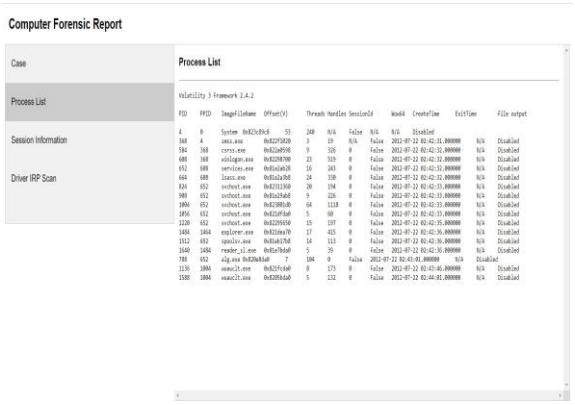


Fig 20 : HTML Report - Memory Module (Process List)

Additionally, we can also do Virustotal Scan using VirusTotalAPI, where multiple engines scan our memory dump file.

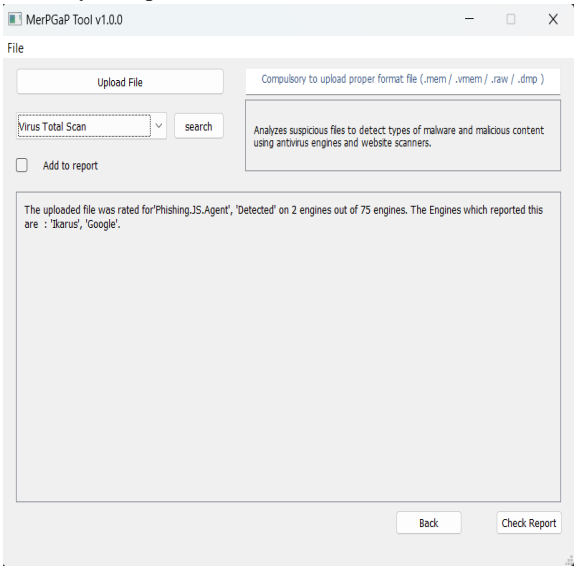


Fig 21 : Virustotal Scan Output

4.3.2 Disk Module Snapshots

1. All user accounts installed on the machine

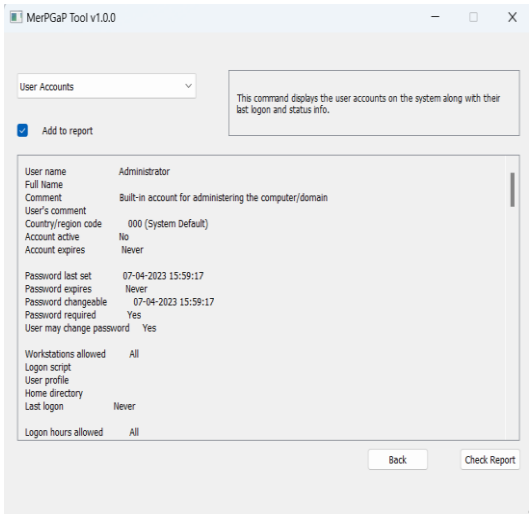


Fig 22 : User Accounts

2. Partition Details present on the system

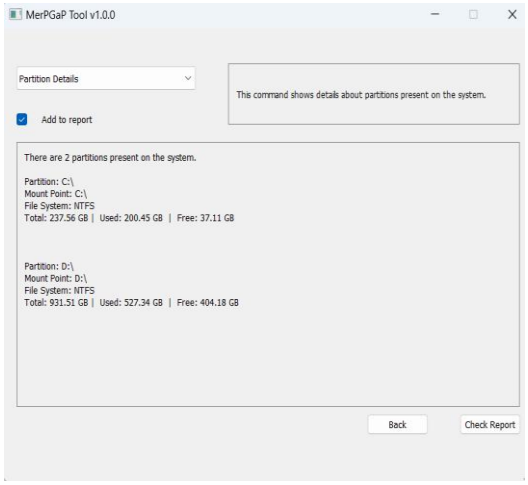


Fig 23 : Partition Details

3. Listing Hidden files and directories

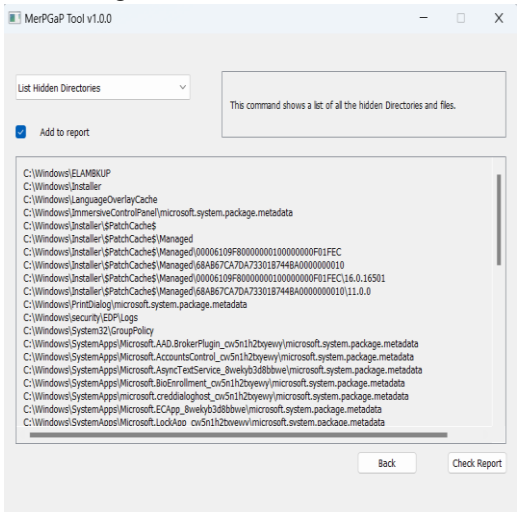


Fig 24 : Hidden Directories

4. Event Logs (Application only)

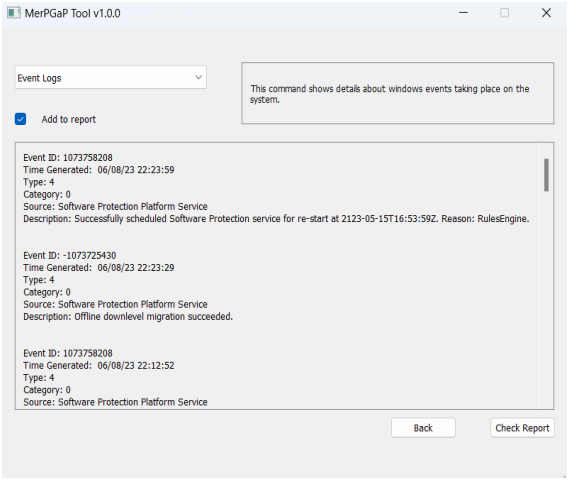


Fig 25 : Event Logs

5. HTML Report

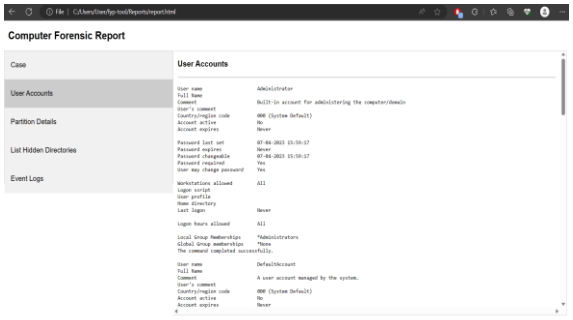


Fig 26 : HTML Report - Disk Module (User Accounts)

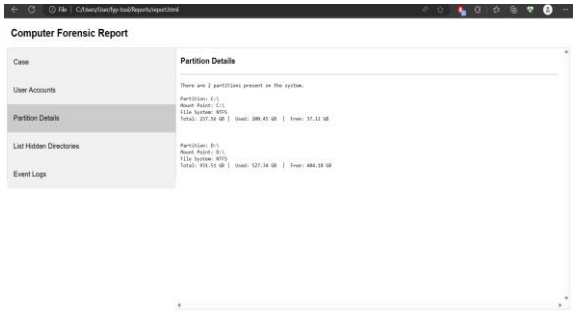


Fig 27 : HTML Report - Disk Module (Partition Details)

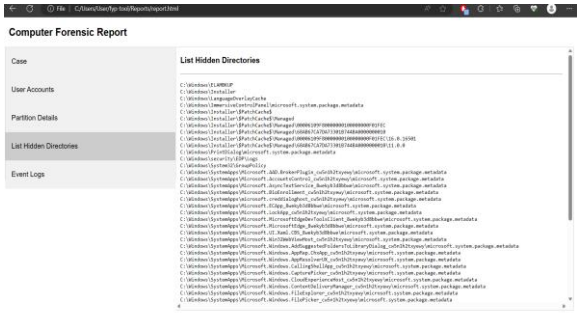


Fig 28 : HTML Report - Disk Module (List Hidden Directories)

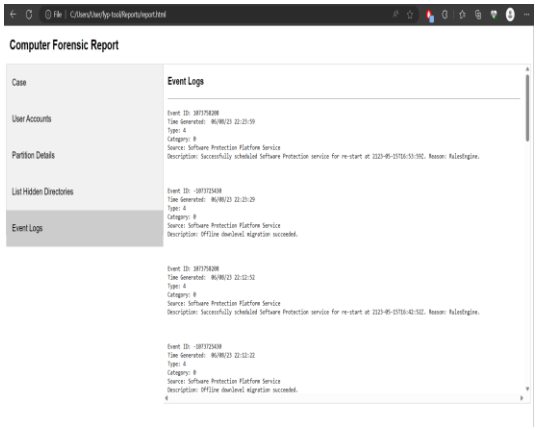


Fig 29 : HTML Report - Disk Module (Application Event Logs)

4.3.3 Registry Module Snapshots

1. Processes set to start automatically on system boot maintained in Run and RunOnce key

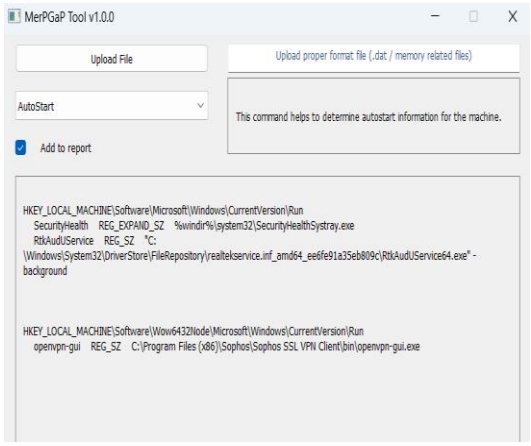


Fig 30 : List of autostart processes

2. Hidden user accounts

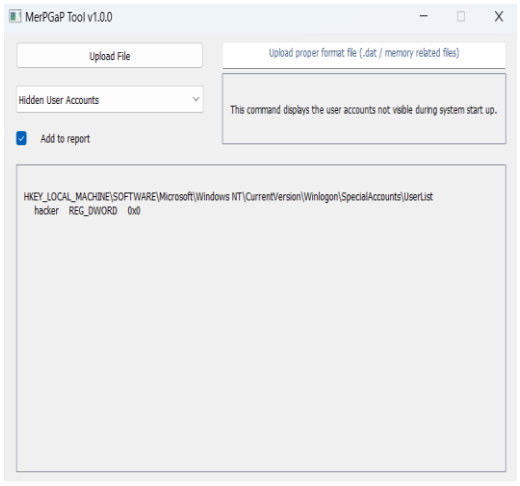


Fig 31 : Hidden user accounts on the system

3. List of processes recently executed by the user maintained in the UserAssist key

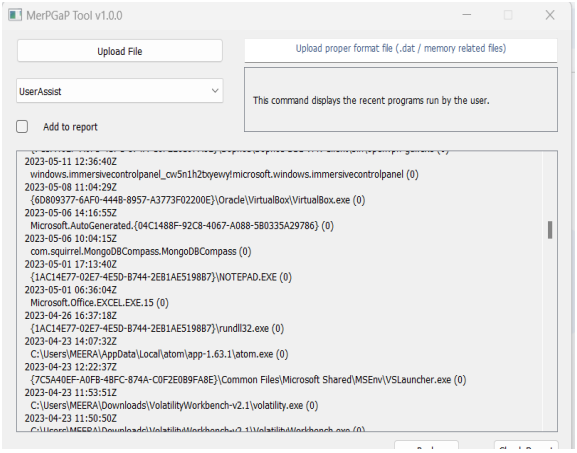


Fig 32: UserAssist key output

4. Files that were recently opened/saved and the folders that were opened are maintained in the RecentDocs registry key.

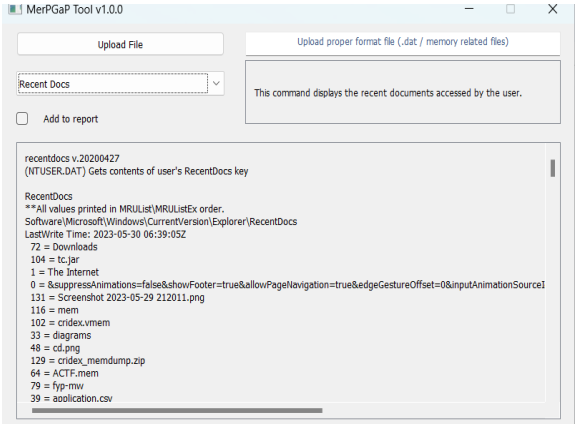


Fig 33 : RecentDocs key output

5. HTML Report gets generated after adding all the required artifacts in the report using the 'Add to report' button

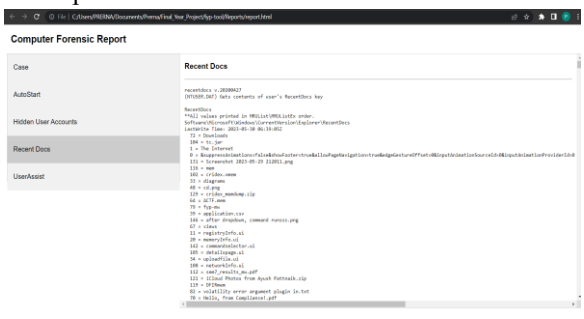


Fig 34 : HTML Report showing the Recent Docs tab

4.3.4 Network Module Snapshots

1. ARP Tab

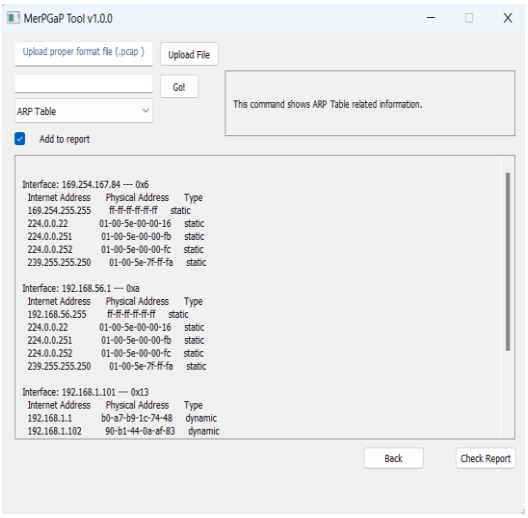


Fig 35 : ARP Table

2. Detecting suspicious IP addresses:

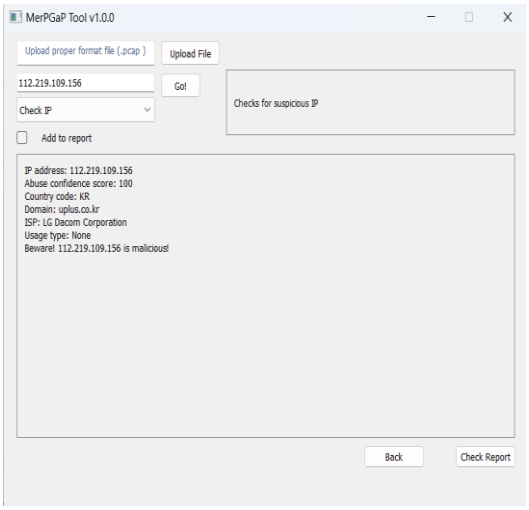


Fig 36: Result for malicious IP address

The IP address 112.219.109.156 is detected as malicious, as the abuse confidence score is a 100!

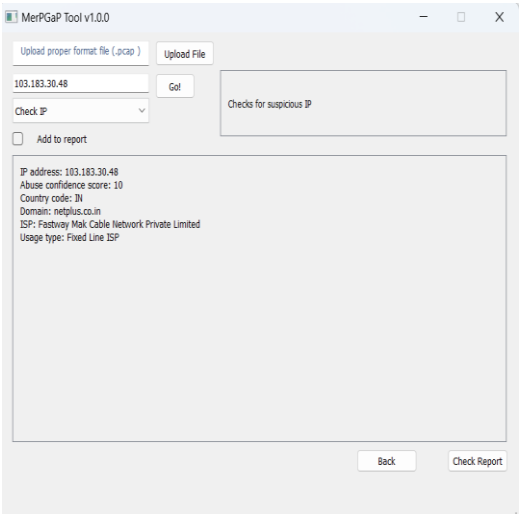


Fig 37: Result for non-malicious IP address

The IP address 103.183.30.48 is not flagged as malicious as its abuse confidence score is less than 20, but all its details are displayed to the user so that the user/investigator can decide for himself whether or not to consider the IP as safe.

3. Pcap file analytics

First we click on the Upload File button to select our desired pcap file to analyze.

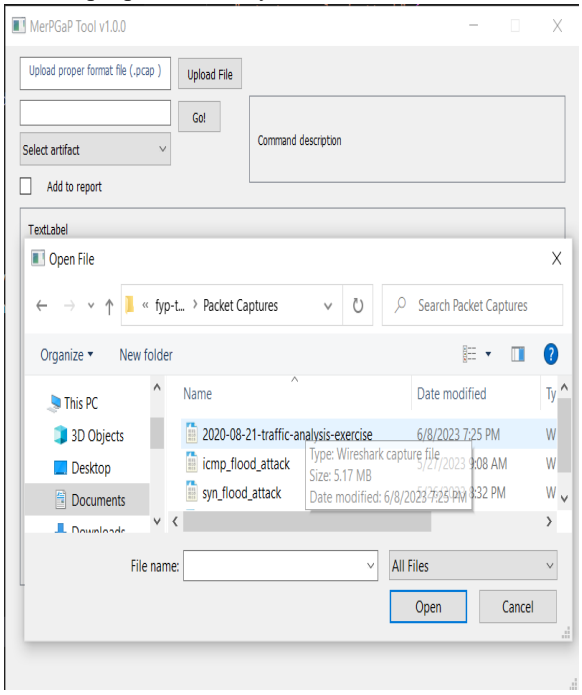


Fig 38 : Upload Pcap file Dialog box

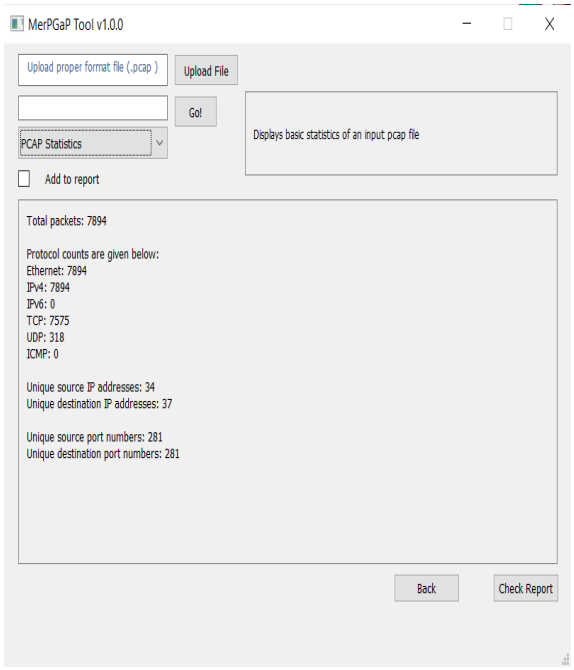


Fig 39 : Pcap File Statistics

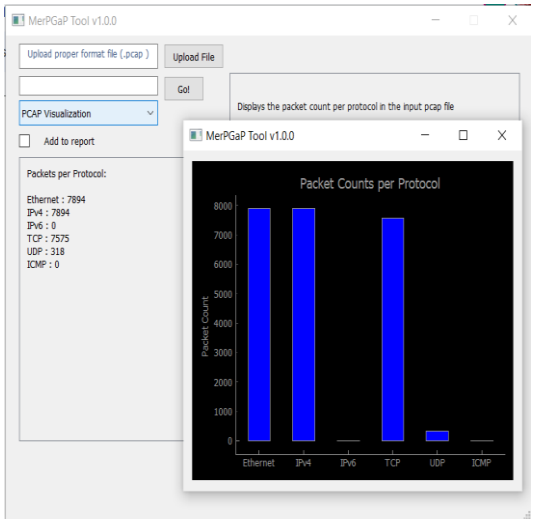


Fig 40 : Pcap Statistics and Visualization

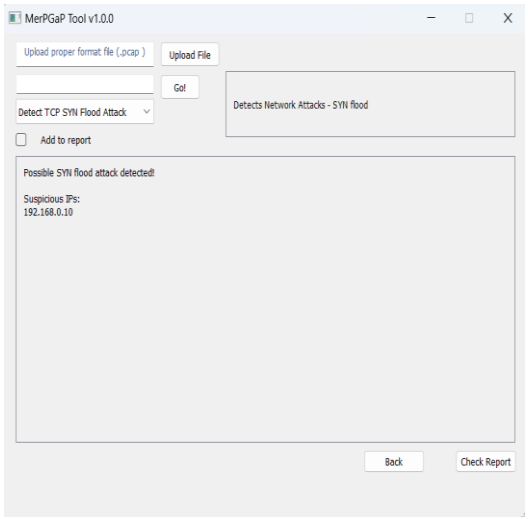


Fig 41 : Detecting SYN flood attack

4. HTML Report

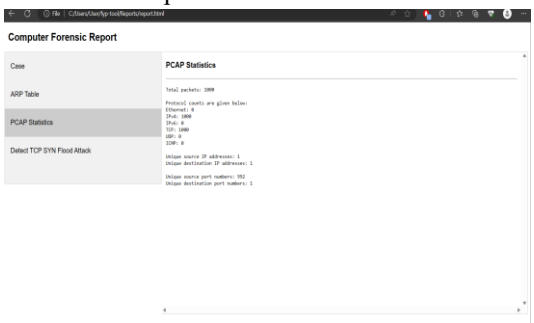


Fig 42 : HTML Report - Network Module

V. FUTURE SCOPE

Our Forensic data extraction tool can be extended functionally to extract data from computer devices based on other operating systems. The present implementation of the tool was focused by taking Windows 10 OS & Windows 11 OS into

consideration. However, the tool can be made feasible for data extraction from different operating systems, apart from Windows as well.

1. Automating the process of dump file creation and disk image creation and integrating it with the forensic tool is also a functionality that can be added. Currently, the two processes of file creation and data extraction from the input files are separate. In the future, the process of dump file creation can be automated and integrated within the tool itself.

2. The html report has a simple panel-based template for viewing artifacts. This can be re-structured to make it more standard & professional so that it can be presented in court.

3. Creating a tamper-proof & immutable forensic report is one of the main functions to be implemented so that artifact evidence can not be meddled with or misconfigured in any way.

4. Although data extraction for all major artifacts like Hidden Directories, Driver Scans, Network settings, Kernel scans etc. are provided, there is scope for further artifacts to be included.

VI. CONCLUSION

The paper initially gives a brief idea about Windows 10 Architecture. The vulnerabilities, threats owing to hardware and software architecture were looked upon. The Windows security model was briefly discussed followed by security w.r.t threats and attacks. The field of computer forensics is bound to attain great value as the ever precious data on PCs and laptops are becoming a target for cyber-attacks. Our paper includes use of open-source forensics and analysis tools, frameworks such as Autopsy, Volatility, RegRipper, NetworkMiner, FTK Imager, etc. Their results are documented as a guide for further studying and research purposes. Finally, a computer forensic tool for finding nondestructive data artifacts as well as live data artifacts was implemented and results were displayed. The tool was created with a user-friendly GUI. It requires a memory dump file for memory forensics and in other areas, it also operates on the live system. The user has the option to select an artifact to investigate and add to report as per preference. The extracted data file from memory can further be checked for any malicious or potential attack data. The tool generates the result as an interactive, well formatted HTML based report with artifacts. The reports can be viewed in any web browser. Thus, we have strived to make a basic tool for digital forensic analysis. There will always be more scope for research and enhancement of digital tools and technology, as long as we are living in a cyberworld.

REFERENCES

[1] UCEMA Executive Education, Depiction of Windows 10 Architecture, (2021), https://ucema.edu.ar/u/jmc/siop/U1/guia_cl01

[2] Ed Williams, Trustwave, SpiderLabs; Edward Millington, Article on OS Credential Dumping: LSASS Memory, (Feb. 11, 2020), https://attack.mitre.org/techniques/T1003/001/

[3] https://en.wikipedia.org/wiki/DNS_spoofing

[4] https://github.com/redcanaryco/atomic-red-team/blob/master/atomics/T1055.012/T1055.012.md

[5] https://learn.microsoft.com/en-us/windows/win32/sync/asynchronous-procedure-calls

[6] https://attack.mitre.org/techniques/T1055/003/

[7] https://github.com/redcanaryco/atomic-red-team/blob/master/atomics/T1055.012/T1055.012.md

[8] https://attack.mitre.org/techniques/T1036/003

[9] Wang, Y., & Lee, H. C. (2013). Research On Some Relevant Problems in Computer Forensics. Proceedings of the 2nd International Conference on Computer Science and Electronics Engineering (ICCSEE 2013). Atlantis Press. https://www.atlantis-press.com/article/4824.pdf

[10] https://www.cloudflare.com/en-gb/learning/ddos/glossary/denial-of-service/

[11] Pathak, A., & Pathak, S. (2020). Study on Decision Tree and KNN Algorithm for Intrusion Detection System. International Journal of Engineering Research & Technology (IJERT), 09(05), 376–381. doi:10.17577/IJERTV9IS050303

[12] Alkasassbeh, M., & Almseidin, M. (2018). Machine Learning Methods for Network Intrusion Detection. arXiv preprint

arXiv:1809.02610.
https://arxiv.org/pdf/1809.02610

[13] Ahmed, Md. Rayhan, Islam, Salekul, Shatabda, Swakkhar, Islam, A.K.M. Muzahidul, & Islam, Md. Towhidul (2021). Intrusion Detection System in Software-Defined Networks Using Machine Learning and Deep Learning Techniques – A Comprehensive Survey. TechRxiv Preprint. doi:10.36227/techrxiv. 17153213.v1
https://www.researchgate.net/publication/356947350_Intrusion_Detection_System_in_Software-Defined_Networks_Using_Machine_Learning_and_Deep_Learning_Techniques_-A_Comprehensive_Survey

[14] Wang, Y., & Lee, H. C. (2013). Research On Some Relevant Problems in Computer Forensics. Proceedings of the 2nd International Conference on Computer Science and Electronics Engineering (ICCSEE 2013). Atlantis Press. https://www.atlantis-press.com/article/4824.pdf

[15] Perumal, S. (2009). Digital Forensic Model Based On Malaysian Investigation Process. International Journal of Computer Science & Network Security, 9(8), 38–44. https://oarep.usim.edu.my/handle/123456789/11493

[16] Autopsy, Autopsy Tool Information, https://www.autopsy.com/

[17] Volatility https://github.com/volatilityfoundation/volatility

[18] RegRipper Tool https://github.com/keydet89/RegRipper3.0

[19] Netresec, Network security monitoring and network forensics Article on NetworkMiner https://www.netresec.com/?page=NetworkMiner

[20] Virus Total, https://www.virustotal.com/

[21] AbuseIPDB, https://www.abuseipdb.com/

[22] Github Link for the tool, https://github.com/gaurishewale20/fyp-tool