NFSU JOURNAL OF CYBER SECURITY & DIGITAL FORENSICS

NFSU – Journal of Cyber Security and Digital Forensics
Volume – 2, Issue – 1, June 2023
E-ISSN - 2583-7559

Automated Windows Triaging & Malware Analysis

Triaging Windows system for hunting malicious process

Dr. Parag Shukla,
Assistant Professor,
School of Cyber Security and Digital Forensics,
National Forensic Sciences University,
Gandhinagar, Gujarat, India
parag.shukla@nfsu.ac.in

Aditya Pratap
Student, M.Sc. Digital Forensics & Information Security,
School of Cyber Security and Digital Forensics,
National Forensic Sciences University,
Gandhinagar, Gujarat, India
Aditya.pratap9557@gmail.com

Dr. Jay Teraiya,
Assistant Professor,
School of Cyber Security and Digital Forensics,
National Forensic Sciences University,
Gandhinagar, Gujarat, India
jay.teraiya@nfsu.ac.in

***Abstract*—**In order to detect and triage malicious processes on Windows machines, efficient and automated techniques need to be developed. Using a comprehensive suite of open-source tools and available python libraries, this research project presents an innovative approach for automating the analysis of Windows machines. The primary objective is to enhance the capabilities of analysts by providing them with a program that can swiftly identify and analyze various security parameters, such as process PID-PPID relationships, network activities, and process command lines. By leveraging open-source tools, the proposed solution aims to streamline the triaging and analysis process, thereby addressing the challenge faced by analysts when encountering machines infected with malicious programs.

Keywords—Automated analysis, Windows triaging, Malware analysis, Open-source tools, Security parameters

I. INTRODUCTION

It has become increasingly difficult for organizations and individuals to cope with advanced cyber threats in today's digital world. A malicious process on a Windows machine can detoriate operations, compromise sensitive data, and jeopardize entire networks' security. Detecting and analyzing these malicious processes is a critical task for cyber security analysts. However, the process often proves to be time-consuming, hindering the prompt response required in the face of evolving threats.

Using an automated approach, this research project aims to detect malicious processes on Windows machines through machine triaging and analysis. Using open-source tools and python libraries, the project seeks to enhance analysts' capabilities by providing them with an efficient solution for evaluating various security parameters.

II. METHODOLOGY

A. Windows Triaging Process

In forensics, Volatile Data or data-in-motion gets deleted once system is turned off. Non-volatile data, on the other hand, resides even when the system is shut down. As defined by *National Institute of Standards & Techniques (NIST)*, "The Digital Forensics process consists of four main process areas, namely,

- the Acquisition stage where artifacts are collected from the affected systems;
- the Examination and Analysis stages related to parsing and understanding the evidence collected from the acquisition stage;
- and finally Reporting on the findings observed."

Windows triaging is process of analyzing the system for volatile and non-volatile information of the infected machine, to look for certain attributes which tells whether the machine has been infected/user to perform attack.
It also includes identifying different types of potentially important evidence on the computer being investigated

B. Volatile & Non-Volatile Information

It is vital for any analyst to capture volatile data from the target machine because it contains complex data and is highly sensitive. According to *RFC 3227* provided by *Internet Engineering Task Force (IETF)*, the order of volatility of evidences is as:

- Registers
- Cache
- Routing Table, Arp Cache, Process Table, Kernel statsitcs
- Temporary file system
- Disk

- Remote logging and monitoring data
- Physical configuration and network topology
- Archival media

Non-Volatile data or we can say data at rest, that mostly contains the user and system r elated data, which remains even if the system is powered down. For example, hard drives, flash memory, and ROM contains non-volatile data. This data includes the following:

- User information
- File timestamp
- Documents/Images/Videos/Audio etc.
- List of installed applications
- Browser details and history
- Email
- Recent Network folders

C. Suspicious Network & Process Analysis

Infected machines present a critical challenge for analysts to detect and analyzing malicious processes. In this section, we examine suspicious processes and networks, emphasizing how malware will typically run processes and establish remote or local network connections.

During process analyses, analysts look for a variety of indicators that could indicate malicious activity.

- Process tree analysis
- Command line analysis
- Dynamic behavior monitoring

In order to communicate with remote servers or spread within a network, malware often establishes network connections. Here, the key aspects to consider are

- Network traffic monitoring
- Connection analysis
- Local network propagation

D. Tools and Techniques

- Eric Zimmerman
  - This is a collection of open-source digital forensic tools for analyzing Windows systems. Infected machines can be investigated and analyzed with these tools to identify malicious content within them.
  - Some of the notable tools include:
    - Registry explorer
    - Shell Bags explorer
    - Prefetch parser
    - Amcache parser
    - LNK files parser
    - Jump list parser

- VirusTotal
  - An online service for detecting and analyzing suspicious files and URLs.
  - A variety of antivirus engines and analysis techniques are utilised to provide insight into the nature of malicious content
  - A file or URL can be submitted to virustotal, which performs scans and generates detailed reports, including malware and indicators of compromise (IOCs).

- AbuseIPDB
  - The platform collects and shares information about IP addresses involved in malicious activities.
  - The database can be queried by analysts to determine if IP addresses have been reported for abusive behavior, such as hosting malware, participating in botnets, or conducting malicious activities.

- Yara
  - The tool can be used to create and share rules for identifying and classifying malware based on specific features or patterns
  - Develop signatures enabling efficient and automated detection of malware families and behaviors

- OPSWAT Metadefender
  - A combination of antivirus engines, threat intelligence, and various methods for detecting and analyzing malicious content is used
  - Provide capabilities for scanning files, URLs and email attachments to identify potential malware and security risks
  - Also offer features like file hash lookups, threat intelligence integration and advanced behavioral analysis to gain deeper insights

E. Python Libraries

Python is widely used for interacting with the system, retrieving process and network information, connecting with security services by calling API, etc. Python supports a huge variety of libraries. Some of the major libraries that are used in this project

- Psutil
  - Cross-platform library providing an interface to retrieve information about running processes and system utilization.
  - It is used for process monitoring and analysis and gather detailed information on their resource

utilization, network connections, parent-child relationships and command-line arguments

- IPWhois
  - Facilitates IP address information retrieval by querying various public WHOIS records
  - Used to investigate ownership and geolocation of IP addresses involved in security incidents
  - Can also obtain details about IP address allocations, contact information and organization responsible for IP range

- Socket
  - Core module in python that provides low-level networking capabilities
  - Used for developing custom network tools for port scanning, network traffic analysis
  - Also offers control to examine network behavior, identify potential threats and analyze network-based attacks

- Subprocess
  - Used to manage system processes, allowing to interact with command line tools and execute external programs
  - In this project, it is used to automate the execution of various security tools and scanners, capture their output and integrate them into larger analysis workflow, enhancing efficiency and scalability

- Streamlit
  - Simplifies the creation of interactive web applications and data dashboards
  - Develop user friendly interfaces for visualizing and exploring security-related data enabling efficient data exploration and decision-making.

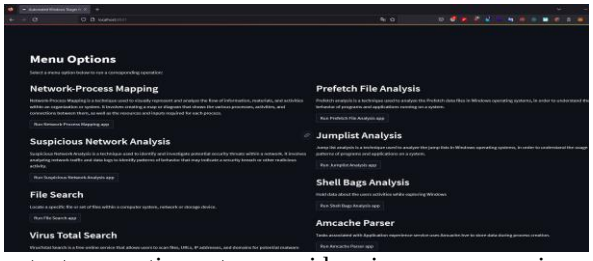
III. IMPLEMENTATION

A. Application Dashboard

The dashboard provides multiple scan options for analysts to choose at time of triaging the infected machine.

- Network-Process Mapping
- Suspicious Network Analysis
- File Search
- Virus Total Search
- Prefetch File Analysis
- Shell Bags Analysis
- Amache Parser ….

These scanning options will allow analyst to perform



important operations to consider in any scenario while investigating any Windows machine.

All the operations are executed in the background via python subprocess module. When analyst selects any scanning option, it shows new window showing options for new scan. The GUI is built on top of python using streamlit module that performs the css and html automatically without any front end code required.

B. Network-Process Mapping

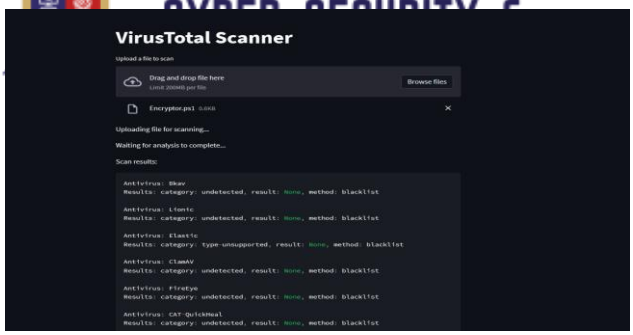
The output panel shows the PID's of running process belonging to same parent ID (PPID) along with any network activity associated with the process.

This allows analyst to view the running processes and check the command line arguments if any encrypted command or any remote network activity is associated with the running processes.

C. Virus Total Scanner

The panel allows to upload any file type up to size of 200MB. The file is scanned via multiple anti-virus solutions to check for malicious pattern of the file. VirusTotal is used in the background via python module vt. VirusTotal provides private API key also for personal usage with limitations on daily usage.

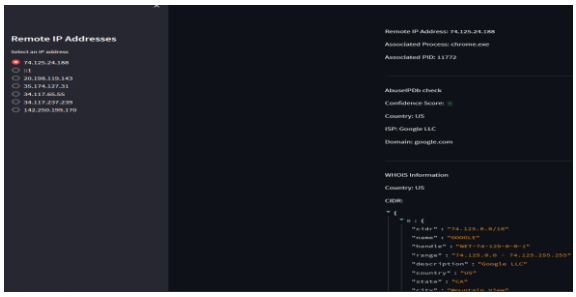
The result contains the list of Antivirus scanned with and the result of the uploaded file.



D. *Suspicious Network Activity Analysis*

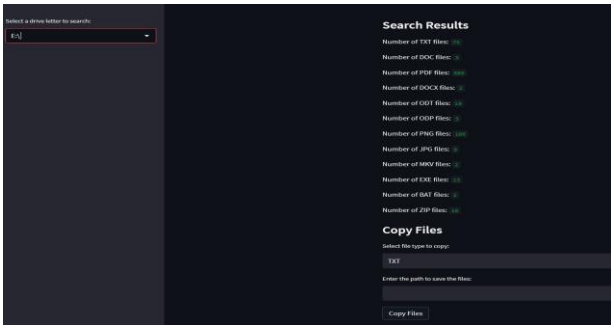
This scan option lists out the remote connections that are established in the system. Windows cmdline command netstat is used to list out the ESTABLISHED remote connections.

The obtained remote connections are then analyzed using AbuseIPdb API key to scan if the IP address is associated with any malware hosting activities, C2 server or any other offensive behavior. Along with the AbuseIPdb check, the associated PID is listed to know the process information as well and lastly WHOIs information required to know the details of the remote IP address such as CIDR, Description, Owner and Registrar information



E. *File Searching*

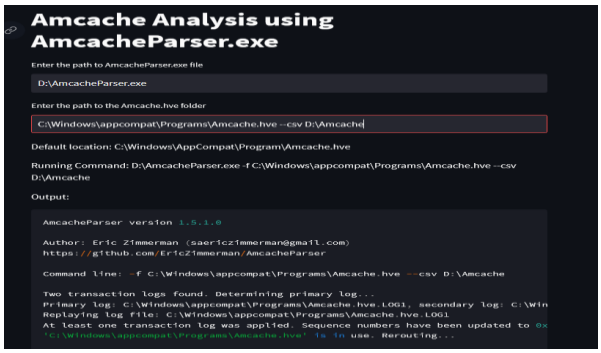
While triaging, analyst has to know what are the characteristics of files present within the system to come up with quick decision that these file types may contain important information and has to analyze them further.



Therefore, analyst can use this scan and also copy the searched files to any external media providing the folder path to save the searched files.

F. *Amcache Parser*

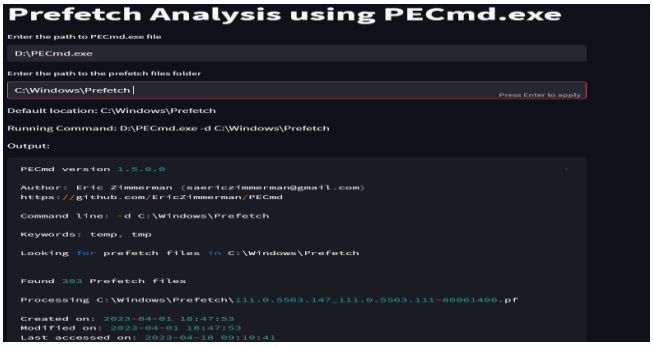
Amcache records the task associated with Application experience service uses registry file Amcache.hve to store data during process creation. Amcache by Eric Zimmerman is to



used analyse the Amcache files from the system and also create CSV for the output.

G. *Prefetch analysis*

Prefetch Files are used to speed up the program starting process. Store list of all files and DLLs used by rpogram when started in order to preload these file into memory when program starts.



For analysts it is important to determine which applications were recently executed, the number of times it was executed etc. PECmd by Eric Zimmerman is used to analyze the stored prefetch files within the system or analyst can analyze the collected prefetch files from the infected machine in forensic lab.

IV. RESULTS & DISCUSSIONS

From the analysis conducted, there are common Windows path where malware attributes are created, the traces of malware execution can be tracked and know about the execution methodology as well.

The solution covers a broader scope for the analyst to perform these operations within limited time and providing deep insight on the system.

- Prefetch files
  - Located within %SystemRoot%/Prefetch
  - Naming schema consists of application name followed by eight character hash of application
  - Information for prefetch stored in registry HKLM\SYSTEM\CurrentControlSet\Control\SessionManager\MemoryManagement\PrefetchParameters
  - Contains the following information
    - Executable name
    - Path to executable
    - Number of time program ran within system
    - Last run time
    - List of DLLs used by program

- Amcache
  - Includes the task associated with application experience service
  - Within Windows machine, Amcache found in C:\Windows\AppCompat\Program\Amcache.hve
  - The structure of Amcache.hve is as
    - Amcache.hve\Root]File\{Volume GUID}\##
- Jumplists
  - These are access items that are recently used
  - It does not include recent media files accessed by the user
  - By default, recent application data is stored in %UserProfile$\AppData\Roaming\Microsoft\Windows\Recent\AutomaticDestinations
- LNK Files
  - These are referred to shortcut files that points to application or files on Windows OS, usually generated when user installs any application and select option to access it from desktop
  - Reveals the following information
    - MAC time attributes
    - Previous activities on computer
    - Linked file size
    - Original file path
    - Serial number and name of volume held the linked file

The built solution included a wide range of tasks, including process analysis, network activity scanning, security risk assessment using VirusTotal and AbuseIPDB, and examination of various system artefacts such prefetch files, jumplist files, and amcache files, among others. The project also emphasised how crucial a user-friendly GUI is for triaging that is simple to use and effective.

By scanning these processes using VirusTotal, the solution effectively identifies security risks associated with specific executables. This made it possible for analysts to focus on the procedures that posed the greatest security dangers and to order their inquiries. The usage of AbuseIPDB along with the network activity analysis was crucial in spotting malicious connections and probable command and control servers. The approach immediately identified connections with well-known malicious sites and highlighted them using AbuseIPDB, enabling quick detection and action.

The solution's user-friendly GUI features allows analysts able to move through the triaging process with ease thanks to its organised workflow. The GUI made it simple to access the numerous analytical choices quickly, resulting in a speedy and effective user experience.

V. FUTURE SCOPE OF WORK

The future scope of this project include potential enhancements to expand its capabilities. Implementing an agent-server model, incorporating security scoring, providing security hardening recommendations and improving interactive reporting and visualization are key areas for future development. These enhancements will empower analysts with comprehensive insights, proactively security measures and efficient incident response capabilities.

Agent -Server model involves developing agent that will be pushed to victim machine to collect data directly from the target system. Agent can gather information such as process details, network activities, running handles and other relevant data. It improves the efficiency and accuracy of data collection, eliminates the need for remote data retrieval methods.

A comprehensive security score indicates the overall vulnerability level of the machine. This information enables analysts to prioritize and implement security hardening measure effectively. Based on the identified weakness, solution can suggest specific measures to improve the security posture of the machine. Include recommendations like disabling unnecessary services, applying security patches, configuring firewall rules or implementing secure network protocols.

Also incorporating machine learning techniques to enhance malware detection and analysis capabilities of the system. This can enable real-time detection and mitigation of emerging threats, minimizing the time between detection and response.

ACKNOWLEDGMENT

I would like to express my sincere gratitude to my mentor Dr. Parag C. Shukla sir for his continuous support and motivation that enabled to diversify the road-map for the project and providing the golden opportunity to do this wonderful project on the topic Automated Windows Triaging & Malware analysis, which also helped me in doing a lot of research and I came to know about so many new things related to Windows malware analysis, incident response, threat intelligence, process & network analysis, etc.

NFSU – Journal of Cyber Security and Digital Forensics
Volume – 2, Issue – 1, June 2023
E-ISSN - 2583-7559

Secondly, I would also like to thank my parents and friends who helped me a lot in finishing this project within the limited time.

REFERENCES

[1] Eric Zimmerman's tools. Available at: https://ericzimmerman.github.io/#!index.md

[2] ABUSEIPDB - IP address abuse reports - making the internet safer, one ... Available at: https://www.abuseipdb.com

[3] Virustotal. Available at: https://www.virustotal.com/gui/

[4] Chad Tilbury (2023) SANS Poster. Available at: https://www.sans.org/posters/windows-forensic-analysis/

[5] Malin, C.H., Casey, E. and Aquilina, J.M. (2012) Malware forensics field guide for windows systems: Digital Forensics Field Guides. Waltham, MA: Syngress.

[6] Miroshnikov, A. (2018) Windows Security Monitoring: Scenarios and patterns. Indianapolis, IN: John Wiley & Sons Inc.

[7] Nagar, R. (2006) Windows NT File System internals. Osr Press.

[8] Opswat, Advanced threat prevention and detection, MetaDefender Cloud. Available at: https://metadefender.opswat.com/

[9] Shaaban, A. and Sapronov, K. (2016) Practical windows forensics: Leverage the power of Digital Forensics for Windows Systems. Birmingham: Packt Publishing.