A Forensic Analysis Tool for Windows File System Artifacts for Security Incident Response

Eric Kimani Mwangi

066455

A Dissertation Submitted in Partial Fulfilment of the Requirements for the Degree of Master of Science in Information System Security at Strathmore University

School of Computing and Engineering Sciences

Strathmore University

Nairobi, Kenya

June, 2025

Declaration and Approval

Declaration

I declare that this work has not been previously submitted and approved for the award of a degree by this or any other University. To the best of my knowledge and belief, the dissertation contains no material previously published or written by another person except where due reference is made in the dissertation itself.

Eric Kimani Mwangi

Signature: 

Date: 22nd May 2025

Approval

The dissertation of Eric Kimani Mwangi was reviewed and approved by the following:

Dr. Victor Rop
Lecturer, School of Computing and Engineering Sciences,
Strathmore University

Dr. Julius Butime,
Dean, School of Computing and Engineering Sciences,
Strathmore University

Prof. Bernard Shibwabo,
Director of Graduate Studies,
Strathmore University

Abstract

Effective incident response relies heavily on timely access to actionable insights, and forensic tools play a vital role in equipping security teams with the information needed to investigate and mitigate threats. One of the major challenges during the analysis phase of incident response is the intentional hiding or deletion of data by cybercriminals. Threat actors often erase evidence such as scripts and executables used in reconnaissance, exploitation, command and control, and data exfiltration to avoid detection. While there are commercial forensic tools available to recover such data, these solutions are frequently complex and demand significant system resources, making them impractical for use on compromised or resource-constrained systems.

To address this gap, this dissertation presents the development of a lightweight, Python-based console tool designed for Microsoft Windows environments. The tool leverages native Windows artifacts including event logs, prefetch files, LNK files, registry hives, network connections, scheduled tasks, and browser data to support forensic investigators and incident responders in recovering and analyzing deleted evidence. Using the Rapid Application Development (RAD) methodology, the project focused on creating an efficient and accessible solution that minimizes resource usage while maximizing forensic value.

Evaluation of the tool demonstrated its ability to successfully recover key Windows artifacts and, crucially, retrieve deleted executable (.exe) files. These capabilities are essential for identifying malicious activity and understanding the scope of an incident. The results affirm the value of lightweight forensic tools in improving the speed and effectiveness of incident response, offering a practical alternative to more resource-intensive commercial solutions.

Keywords: Deleted file recovery, Indicator of Compromise, Windows Artifacts, Windows Operating System, incident response.

Table of Contents

Declaration and Approval ..... **ii**
Abstract ..... **iii**
Table of Contents ..... **iv**
List of Figures ..... **viii**
List of Tables ..... **x**
List of Abbreviations ..... **xi**
Definition of Terms ..... **xii**
Acknowledgements ..... **xiii**
Chapter 1 : Introduction **1**
1.1Background to the Study 1
1.2 Problem Statement 2
1.3 Aim 2
1.4 Research Objectives 3
1.5 Research Questions 3
1.6 Research Justification 3
1.6.1 Contribution to Computing 3
1.6.2 Contribution to Society 4
1.7 Scope and Limitation 4
Chapter 2 : Literature Review **5**
2.1 Introduction 5
2.2 Overview of Detection and Analysis 5
2.3 Windows Operating System 6
2.4 Windows Artifacts 7
2.4.1 Prefetch File 8
2.4.2 Browser Artifacts 9
2.4.3 Volume Shadow Copy Artifacts 10
2.4.4 Network Artifacts 12
2.4.5 Link File Artifacts 13
2.4.6 User Assist Keys 14
2.4.7 Scheduled Task 15

2.4.8 Hidden Files 16
2.4.9 Database Files 17
2.5 Deleted File Recovery in Incident Response 18
2.5.1 File Record on NTFS and FAT32 19
2.5.2 Deleted File Recovery on NTFS and FAT32 19
2.6 Recent Works in Extracting File Artifacts and Recovering Deleted Files 20
2.6.1 File Artifact Extraction 20
2.6.2 Deleted File Recovery 20
2.7 Gaps in the Existing Works 21
2.8 Conceptual framework 22
2.9 Summary of the Literature Review 23
Chapter 3 : Methodology 24
3.1 Introduction 24
3.2 Research Approach 24
3.3.1 Requirement Planning 24
3.3.2 System Design 25
3.3.3 Development 25
3.3.4 Implementation and Testing 26
3.4 Sampling 27
3.5 Ethical Considerations and Approval 27
3.6 Chapter Summary 28
Chapter 4 : System Analysis, Design and Architecture 29
4.1 Introduction 29
4.2 Requirement Analysis 29
4.2.1 Functional Requirements 29
4.2.2 Non-Functional Requirements 29
4.3 System Architecture 30
4.3.1 Client Layer 30
4.3.2 Disk Layer 30
4.4 System Design 33
4.4.1 Use Case Diagram 33

4.4.2 Sequence Diagram 35
4.4.3 Data Flow Diagram 37
4.5 Application Wireframe 38
4.6 Chapter Summary 39
Chapter 5 : System Implemenation and Testing 40
5.1 Introduction 40
5.2 System Implemenation 40
5.2.1 Hardware Requirements 40
5.2.2 Software Requirements 40
5.3 Incident Response Tool 41
5.3.1 Import Modules 41
5.3.2 Executing the Tool 42
5.3.3 Tool Dashboard 42
5.3.3 Artifacts Extraction 43
5.3.4 Recovery of Deleted Files 54
5.4 System Testing 58
5.4.1 Functional Testing 58
5.4.2 Integration Testing 60
5.4.3 User Friendliness 61
5.4.4 Performance 61
5.5 System Evaluation and Validation 62
Chapter 6 : Discussion 64
6.1 Introduction 64
6.2 Review of Current Technologies Used in Extracting File Artifacts 64
6.3 Review of Research Objectives 65
6.4 Validation of the Forensic Tool for Windows Artifact Extraction in Incident Response 66
Chapter 7 : Conclusions and Recommendations 68
7.1 Conclusions 68
7.2 Recommendations 68
7.3 Future Works 69
References 70

Appendices **74**
Appendix A: Similarity Report 74
Appendix B: Ethical Clearance Confirmation 75
Appendix C: Survey Questionnaire 76
Appendix D: Participant Information Sheet and Consent Form 80

List of Figures

Figure 2.1: Incident Response Life Cycle (Cichonski et al., 2012) 5
Figure 2.2: Operating System Version Market Share (Statcounter, 2024) 6
Figure 2.3: Prefetch Files created from script execution 9
Figure 2.4: Browser history artifact 10
Figure 2.5: Restore point creation 11
Figure 2.6: Volume shadow copy created from the restore point 11
Figure 2.7: Restored file from the shadow copy 12
Figure 2.8: Telnet C2 13
Figure 2.9: Shortcut file added to the startup folder 14
Figure 2.10: User Assist registry data 15
Figure 2.11: Startup scheduled task 16
Figure 2.12: File created with hidden attribute 17
Figure 2.13: Staging database for harvested credentials 18
Figure 2.14: Conceptual framework 22
Figure 3.1: RAD Phases 24
Figure 4.1: System Architecture 32
Figure 4.2: Use Case Diagram 34
Figure 4.3: Sequence Diagram 36
Figure 4.4: Data Flow Diagram 38
Figure 4.5: Wireframe Showing the Dashboard 39
Figure 5.1: Tool dashboard 43
Figure 5.2: Browser history artifacts 44
Figure 5.3: Prefetch Test 45
Figure 5.4: Prefetch logs 45
Figure 5.5: Modify Registry Test 46
Figure 5.6: Registry dump 46
Figure 5.7: Telnet C2 47
Figure 5.8: Telnet service 48
Figure 5.9: Shadow Copy Artifact 49
Figure 5.10: Hidden file artifact 50

Figure 5.11: Database Artifact 51
Figure 5.12: Recovered lnk file 52
Figure 5.13: Recovered scheduled task 53
Figure 5.14: User assist data artifact 54
Figure 5.15: Assessment and penetration testing tools: 56
Figure 5.16: Selection of Recovery disk 56
Figure 5.17: Recovered Files 57
Figure 5.18: Recovered files validation 58
Figure 5.19:User friendliness Testing 61
Figure 5.20: Performance testing 62
Figure 5.21: System Evaluation and Validation testing 63

List of Tables

Table 5.1: Hardware Environment 40
Table 5.2: Software Environment 40
Table 5.3: Results from Functional Testing 59
Table 5.4: Results from Integration Testing 61

List of Abbreviations

AI	Artificial Intelligence
ATT&CK	Adversarial Tactics, Techniques, and Common Knowledge
API	Application Physical Interface
CTI	Cyber Threat Intelligence
CSV	Comma-Separated Values
C2	Command-and-control
FTP	File Transfer Protocol
GUI	Graphical User Interface
GUID	Globally Unique Identifier
IOC	Indicator of Compromise
IR	Incident Response
MFT	Master File Table
ML	Machine Learning
OS	Operating System
SOC	Security Operation Centre
SQL	Structured Query Language
TTP	Tactics, Techniques and Procedures
VBR	Volume Boot Record
VCS	Volume Shadow Copy
RAD	Rapid Application Development

Definition of Terms

Indicator of Compromise: An Indicator of Compromise (IOC) is a piece of forensic data that identifies potentially malicious activity on a system or network. These indicators help cybersecurity professionals detect data breaches, malware infections, or other threat activity by providing evidence that an attack may have occurred or is in progress. Common types of IOCs include file hashes, IP addresses, domain names, URLs, registry keys, and unusual network traffic patterns (Haber & Rolls, 2024).

Tactics, Techniques, and Procedures: Tactics, Techniques, and Procedures (TTPs) are structured descriptions of the methods adversaries use to exploit cybersecurity vulnerabilities, forming a core component of Cyber Threat Intelligence (CTI). TTPs encompass an attacker's overarching goals (tactics), the means of achieving those goals (techniques), and the specific tools or steps used (procedures) (Fayyazi et al., 2024).

Command and Control: Command and Control (C2) refers to the infrastructure and methodologies that adversaries employ to maintain communication with compromised systems within a target network. This communication enables attackers to issue commands, exfiltrate data, deploy additional malware, and orchestrate other malicious activities remotely (Haider et al., 2024).

Acknowledgements

This research project would not have been possible without the support and encouragement of many individuals. I extend my heartfelt gratitude to my supervisor, Dr. Victor Rop, for his guidance, insightful feedback, and invaluable suggestions throughout every stage of this work. I also wish to thank the university faculty and my colleagues for their continued support during my Master's studies. My deepest appreciation goes to my beloved family and countless friends, whose patience, encouragement, and unwavering love sustained me through this journey. Above all, I am profoundly grateful to God Almighty for His grace and the opportunity to complete this endeavour.

Chapter 1 : Introduction

1.1Background to the Study

In the ever-evolving landscape of cybersecurity, the persistent and escalating threat of malicious cyber activities poses a significant challenge to organizations worldwide. The need for advanced techniques in incident response is evident, particularly in the realm of identifying, recovering, and analyzing Indicators of Compromise (IOCs) within Windows environments. IOCs serve as critical traces left behind by cyber adversaries, reflecting their intent to compromise systems and exfiltrate sensitive data. Windows file systems, being the foundation of data storage and retrieval in these environments, contain valuable artifacts that can potentially unveil the tactics, techniques, and procedures (TTPs) employed by threat actors.

Artifacts refer to elements or zones within a computer system that house crucial information pertinent to user-initiated activities on the computer (Liew & Ikeda, 2019). The nature and location of information stored in these artifacts vary from one operating system to another. Proper identification, processing, and analysis of these artifacts are essential to validate or refute observations made during forensic analysis. It is important to note that the absence of information in a specific artifact does not necessarily indicate the non-occurrence of the activity within the computer system. Within the Windows environment, numerous artifacts play a significant role as key evidence in the forensic analysis of digital media. The types and locations of these artifacts may exhibit variations among different versions of the Windows operating system (Kondapally, 2015).

Traditional data recovery methodologies often fall short in effectively retrieving data associated with IOCs, emphasizing the necessity for a specialized forensic analysis framework tailored explicitly for Windows file system artifacts. Understanding the complexities of Microsoft Windows file systems becomes paramount for the development of a tool that can not only recover compromised data but also provide insights into the source, impact, and potential attribution of cyber threats.

This research lays the groundwork for the research on the forensic analysis of Windows file system artifacts for enhanced data recovery of indicators of compromise. The understanding of the Windows file system architecture, IOCs, forensic analysis techniques, challenges in data recovery,

and the broader incident response context establishes a solid foundation for the development of a specialized tool to address the persistent challenges in contemporary cybersecurity landscapes.

1.2 Problem Statement

The increasing sophistication and rapid evolution of cyber threats demand a proactive and effective approach to incident response. Cybersecurity professionals face significant challenges in accurately identifying, recovering, and analyzing IOCs within the Windows operating system. Threat actors use advanced tactics, techniques, and procedures such as data exfiltration, malware attacks, and phishing campaigns to exploit system vulnerabilities, leaving behind critical forensic artifacts that are essential for thorough investigation.

Existing forensic analysis and IOC data retrieval tools, however, face several key limitations that hinder their effectiveness. Many tools struggle to recover deleted or partially overwritten artifacts within the Windows file system, where evidence may be fragmented or obscured by system operations or attacker countermeasures. This results in incomplete data recovery, which compromises the accuracy and completeness of investigations. Furthermore, these tools often lack advanced automation and intelligent parsing capabilities to interpret complex and evolving IOC patterns across diverse artifact types such as registry entries, event logs, prefetch files, and shadow copies. Consequently, analysts must manually sift through large volumes of data, increasing investigation time and the risk of missing critical evidence. In addition, most tools rely heavily on command-line interfaces, which require specialized expertise and increase the chance of human error, especially under the pressure of active incident response. Lastly, many of the comprehensive forensic solutions available are commercial products with high licensing costs and modular pricing, limiting their accessibility to organizations with constrained budgets (maheswari & Shobana, 2021).

1.3 Aim

The purpose of this study is to create a tool designed for cybersecurity professionals that facilitates the identification of Indicators of Compromise from Windows file system artifacts and enables the recovery of deleted malicious files left by threat actors, thereby supporting the analysis of potential intrusions or security incidents.

1.4 Research Objectives

i. To analyze Windows file system artifacts for identifying indicators of compromise associated with malicious activities.
ii. To review tools and techniques available in the Windows environment used in fetching Windows file system artifacts and reconstruction of deleted files.
iii. To design and develop a forensic analysis tool tailored for the Windows Operating system aimed at extracting file system artifacts and recovering deleted indicators of compromise from Memory by threat actors.
iv. To evaluate the tools accuracy in recovery, reconstruction of deleted IOCs and generating an output report to assist in the Incident Response Process.

1.5 Research Questions

i. What file system artifacts can be identified and leveraged to identify IOCs?
ii. What tools are used in recovering deleted files from memory and fetching Windows files artifacts?
iii. How will the developed tool leverage python libraries to efficiently fetch critical windows artifacts enabling the recovery and reconstruction of files flagged as IOCs?
iv. Is the tool developed effective in reporting and providing all the required inputs to assist in the Incident response process?

1.6 Research Justification

1.6.1 Contribution to Computing

Cyber incidents are increasing in frequency and sophistication, prompting a necessary shift towards proactive cybersecurity measures. One critical approach involves the collection and analysis of Windows artifacts digital traces left by user and system activity which serve as a valuable source of threat intelligence. This practice contributes significantly to the field of computing by enhancing the ability to detect and analyze potential threats before they escalate into major incidents. Through the analysis of network and browser artifacts, cybersecurity professionals can identify suspicious connections between endpoints and command-and-control (C2) hosts. For example, browser history and cache files may reveal beaconing communication patterns, which are characteristic of malware attempting to maintain contact with C2 infrastructure.

Furthermore, endpoint packet captures can offer detailed information about the nature of these communications, including the specific port being used. This, in turn, helps identify the service running on that port such as an unauthorized FTP connection on Port 22 providing crucial insights for threat mitigation. These technical processes not only push the boundaries of digital forensics and behavioural analytics within computing but also drive innovation in network monitoring and incident response tools.

1.6.2 Contribution to Society

Beyond technical contributions, these cybersecurity practices have broader societal impacts. By identifying and neutralizing threats, organizations can protect their digital assets, maintain operational continuity, and uphold the privacy and trust of their stakeholders. This fosters a more secure digital ecosystem that benefits individuals, businesses, and critical infrastructure. Additionally, as organizations adopt such proactive strategies, they contribute to a wider culture of cyber awareness and readiness, which is essential in an era of increasingly complex cyber threats. This evolution also highlights the growing demand for skilled cybersecurity professionals, emphasizing the need for education and training in digital forensics and incident response to support both computing advancements and societal resilience.

1.7 Scope and Limitation

The research aimed to conduct an in-depth forensic analysis of Windows file system artifacts to enhance the recovery of data compromised by threat actors, focusing on the identification and analysis of IOCs. The scope included a comprehensive examination of various file system elements such as Master File Table entries, timestamps and registry data. The tool extracts, interprets, and leverages identified artifacts to improve the precision and efficiency of data recovery processes during incident response. The study addressed challenges associated with IOCs and data recovery techniques, with the ultimate objective of contributing to a more understanding of cyber threats and strengthening incident response capabilities within the dynamic and evolving cybersecurity landscape. The research focused on the exploration of Windows artifacts and is limited to the Windows operating system.

Chapter 2 : Literature Review

2.1 Introduction

This chapter presents a review on Windows artifacts and existing tools used to fetch and analyze the artifacts by incident responders, their limitations and challenges experienced. The chapter exhaustively discusses artifacts generated by Windows OS, such as prefetch files, browser history, jump lists, scheduled tasks and examines the tools both commercial and open-source used to recover such artifacts. The chapter concludes by identifying gaps in the current tools and proposes a research solution aimed at addressing the limitations enhancing the efficiency and reliability of artifact and deleted file recovery by incident responders.

2.2 Overview of Detection and Analysis

The field of security operations and incident response is a crucial foundation in safeguarding organizations from the constantly changing landscape of cyber threats. Security teams rely on the Incident Response Plan to effectively respond and manage security incidents through the orchestration of numerous security tools to identify potential indicators of compromise. Digital forensic involves gathering of data and comprehensive analysis of the indicators surrounding a security incident (Gorecki, 2020). This process precedes the incident response actions and falls under the detection and analysis phase within the NIST incident response lifecycle as illustrated in Figure 2.1.

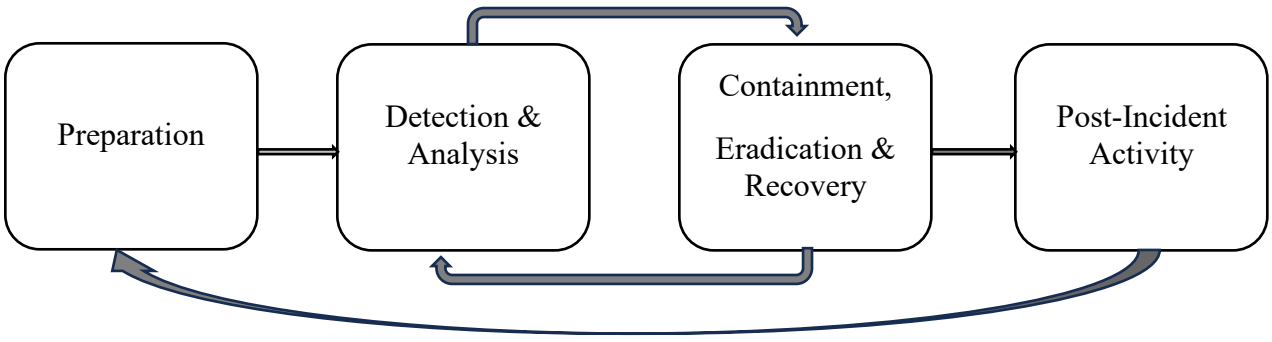


Figure 2.1: Incident Response Life Cycle (Cichonski et al., 2012)

The detection and analysis phase are triggered by an alert or event flagged by the Security Information and Event Management (SIEM), endpoint detection and response solution, anomaly detected by user in a device or any security system implemented in an organization to monitor threats. External sources such as Cyber Threat Intelligence (CTI) feeds can also be used to trigger the second stage of Incident Response. The Detection and analysis phase rely on windows artifacts which hold digital traces left on the device after a security incident. The security analyst is required to identify and analyze the artifacts which will be used as forensic evidence to classify the alert as true positive and identify valuable IOCs. Windows Operating system has multiple artifacts that play a pivot role in the forensic examination of digital media.

2.3 Windows Operating System

An operating system is a collection of software designed to manage computer resources and deliver essential services to tools. From 1985 to 2024, Microsoft developed and released 28 versions of the Windows OS. The different versions have an average global market share of 72% and provide distinct methods of storing data in disk that is of forensic value to an examiner as illustrated in Figure 2.2.

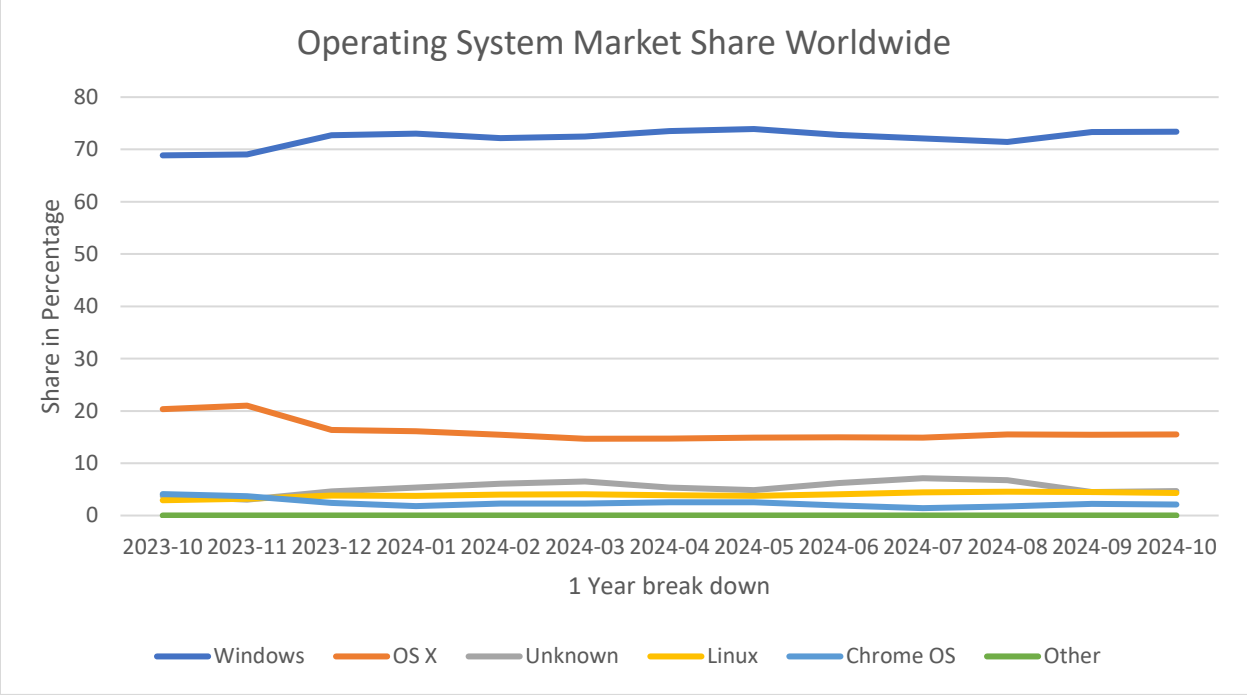


Figure 2.2: Operating System Version Market Share (Statcounter, 2024)

Windows versions from 7 to Windows 11 on PC and Server 2012R2 to Server 2022 generate data and artifacts that are often overlooked and underexamined as digital evidence which provide valuable insights in incident response and forensic examination.

2.4 Windows Artifacts

Windows artifacts are objects created by the OS which contain crucial information about the activities and actions made by users and tools on a device. The artifacts vary in type, location stored and they are invaluable in incident investigation and forensic analysis as they provide evidence of user and tool actions in an end device (Duranec et al., 2019). In recent years, enterprise level breaches have increased from Advanced Persistent Threats (APT) highlighting the need for a comprehensive understanding of incidents and the importance of evaluating the impact to enable effective recovery and preventive measures within an organization (Budhrani et al., 2022a). A simple review of anomalies for instance malicious executables and policy breaches are no longer sufficient as attackers continuously adapt and improve their tactics and techniques to evade detection.

Threat actors hide their trail by leveraging benign files (living of the land tools) or commands which allow them to bypass signature-based detection tools. Endpoints with the different Windows OS versions and configurations, have essential logging enabled which is stored in memory. Disk memory contains the most volatile, cyber artifacts such as active processes, network connections, registry changes, scheduled tasks which are essential for incident response as it provides insights into recent activities and potential intrusions and in some cases the task becomes challenging if the threat actor has erased or deleted evidence related to the breach (Barakat & Hadi, 2016).

Mainstream forensics tools struggle to keep up with the changes in artifact data formats and storage location between different Windows OS versions. This means manual analysis is often employed to examine the digital artifacts. This section will discuss the windows artifacts in scope for the research, tools used to extract the artifacts and their shortcomings which are addressed by the proposed solution. Powershell scripting and the Atomic red Team library of tests mapped to the MITRE ATT&CK framework was used to showcase the invaluable artifacts left behind in Memory from emulated attacks (Red Canary, 2024).

2.4.1 Prefetch File

Prefetch file is a Windows OS artifact that enhances a tool's start-up time by preloading the necessary data required to run the tool stored in the prefetch folder on the same disk volume as the OS. Prefetch loading is automatically enabled in the Windows OS ensuring performance optimization. From an incident response perspective, prefetch files are a valuable source of digital evidence by tracking which programs were run and when which aids in reconstructing threat actors activity on a system (Budhrani et al., 2022b).

The prefetch folder is located on the same volume as the OS (C:\Windows\Prefetch) for systems with the OS on the C drive. Prefetch files have the .pf extension and store data such as file name, file size, creation and modification time, executable file location, run counter and last run time. This information is useful in profiling a systems activity through the identification of frequently run tools which is helpful in building a timeline of the frequently accessed tools (S et al., 2020).

Figure 2.3 shows the execution of Technique T1059.001: AutoIt Script Execution, a scripting language tool used for automating the Windows GUI which threat actors can use to execute commands, automate tasks and deliver malware through the manipulation of Windows tools and system functions. The researcher executed a benign script which simulates the launch of a calculator tool which created prefetch data for both AUTOIT3.EXE and CALC.EXE. The simulation helps in validating the detection around prefetch file creation and automated tool launches.

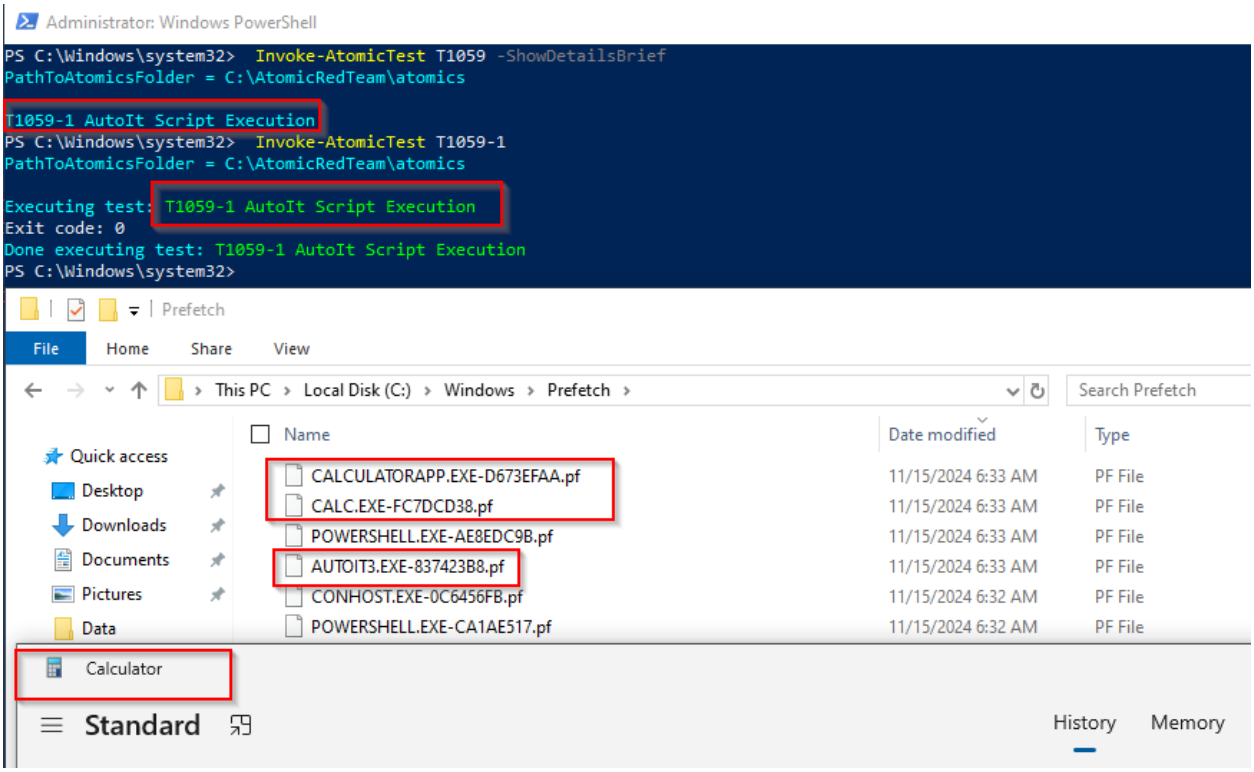


Figure 2.3: Prefetch Files created from script execution

2.4.2 Browser Artifacts

Browser artifacts are created based on user activities on systems using various web browsers like Google Chrome, Microsoft Edge, Mozilla Firefox and Internet Explorer. Each browsers stores data in unique formats, however the artifacts generally record key user browsing activity, such as: web page visited, date and time, content accessed and search queries. A substantial amount of information remains in memory even if the user attempts to delete browsing data using in-browser options or external cleaners (Javed et al., 2024). The scope of this research will focus on Google chrome history which stores the browser history and cache in directory C:

C:\Users\Username\AppData\Local\Google\Chrome\User Data\Default).

Figure 2.4 showcases the browsing history of a threat actor navigating to a credential stealer GitHub repository which can be used in an organization to compromise credentials. The digital artifatcs evidence can be used to build a timeline of events when execution of the Mimikatz tool is noted in the same device where the web page was accessed.

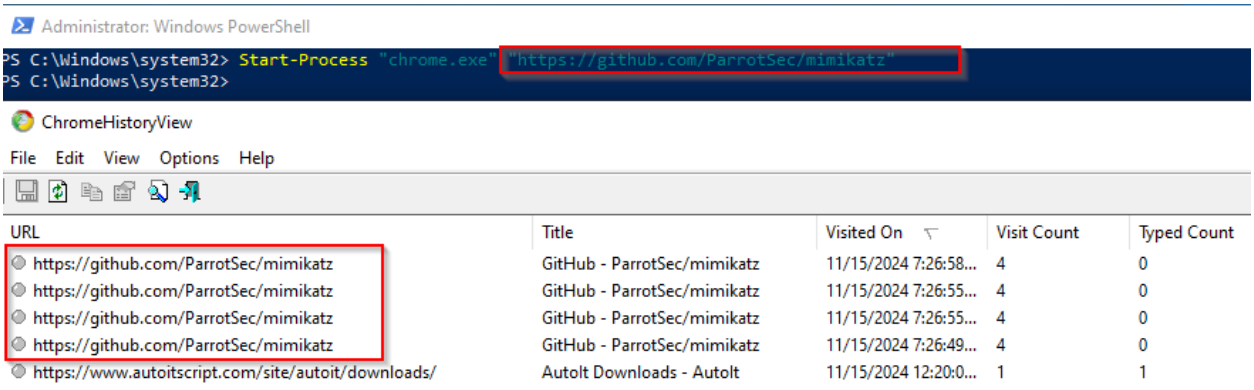


Figure 2.4: Browser history artifact

2.4.3 Volume Shadow Copy Artifacts

Volume shadow copy (VCS) is a key resource for extracting evidence even after previous data has already been erased from the disk. Windows OS creates snapshots of disk volumes that may be recoverable by decoding and analyzing the snapshots stored in volume shadow copies. The VCS is a feature in Windows that supports restore point functionalities allowing files to be reverted or rolled back to an earlier state. VCS operates at the block level in memory and stores changed snapshot data in logical blocks. When a change is made, the original block is copied to the shadow copy file before the new data is written to disk which when combined with the modified data on the original volume can restore the volume to its previous state before any changes occurred. Since only the modified data is stored, recovering data from a VCS without the original volume is not possible (Sreeja & Balan, 2016).

The VSC are invaluable to an incident responder as they provide snapshots of a systems state at different points in time by identifying, parsing and examining the VSC. The snapshots allow incident responders to recover deleted and altered data, track unauthorised changes and understand the timeline of events leading to an incident. Figure 2.5 – 2.7 shows the process of creating a shadow copy from a restore point and recovering a file used for credential dumping by threat actors from the shadow copy to the parent folder.

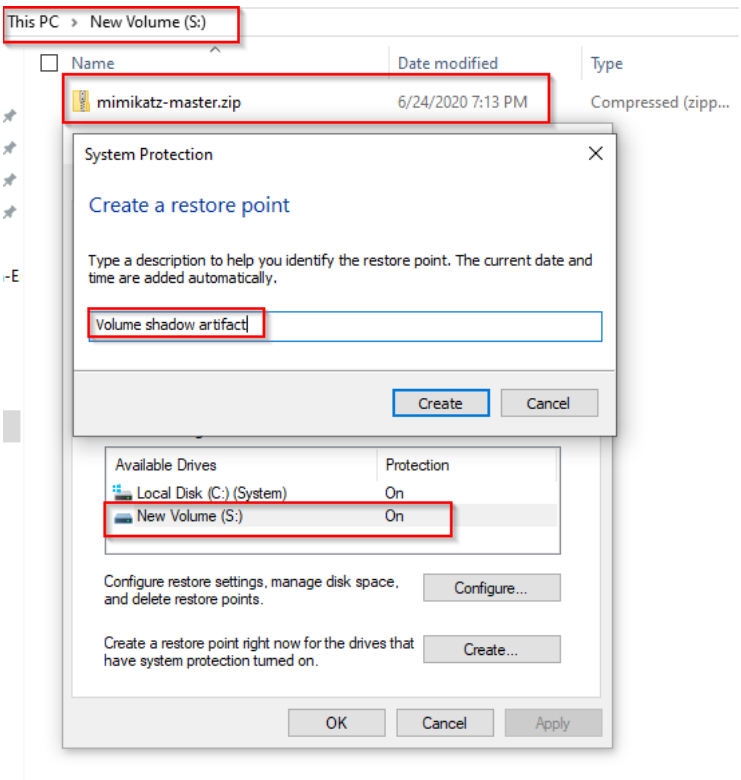


Figure 2.5: Restore point creation

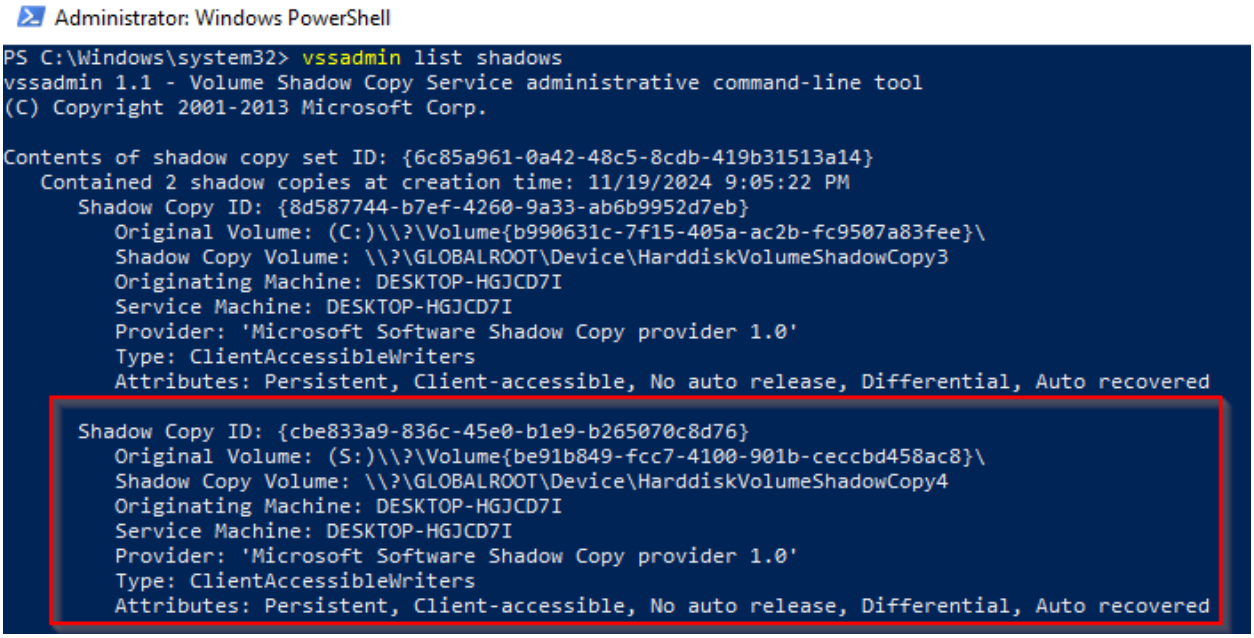


Figure 2.6: Volume shadow copy created from the restore point

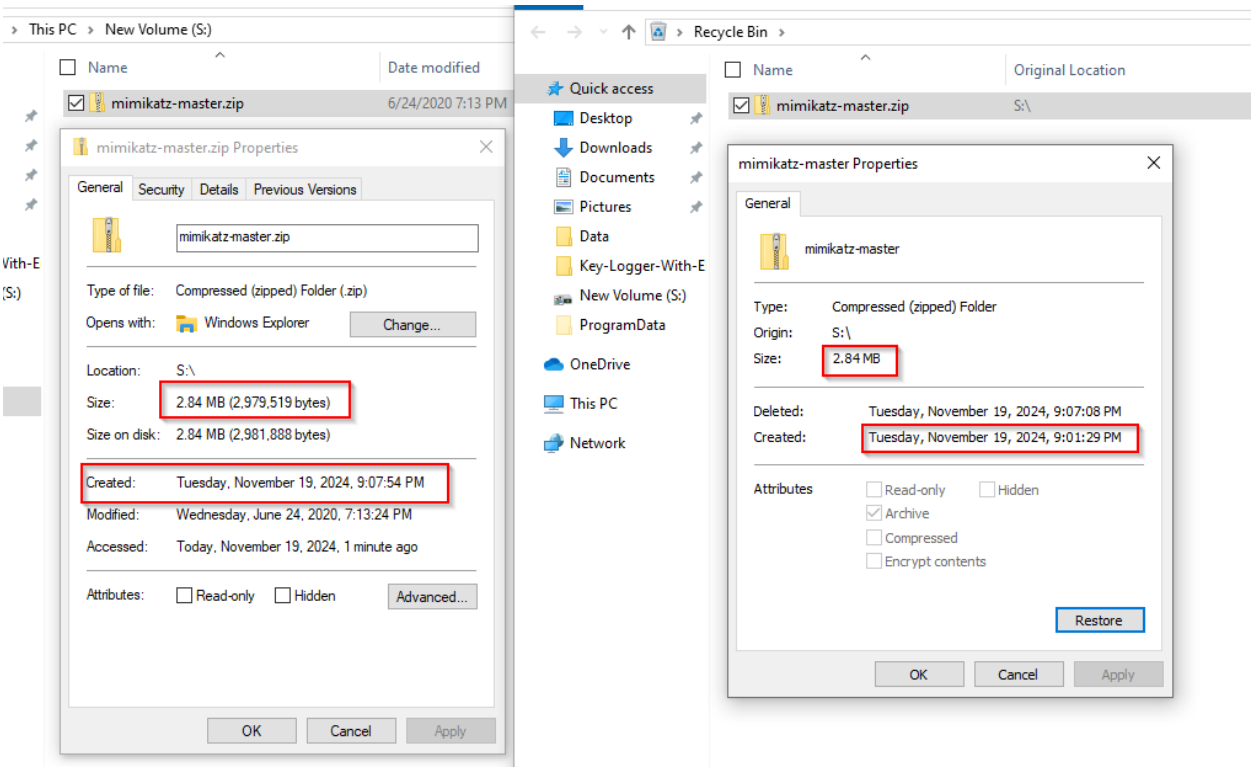


Figure 2.7: Restored file from the shadow copy

2.4.4 Network Artifacts

Windows network artifacts are invaluable to an incident responder as they provide detailed insights into network activities enabling the detection of malicious behavior and tracking a threat actors movement. Network artifacts such as ARP tables and DNS cache reveal IP addresses, port connections, domain names and communicating endpoints which help analysts to identify C2 servers, lateral movement and data exfiltration. Figure 2.8 provides a demonstration of how adversaries might establish a C2 server using telnet. This simulation is valuable for improving detection capabilities for network activities creating effective response procedures to terminate potential C2 channels during actual incidents (Alsmadi & Alazab, 2017).

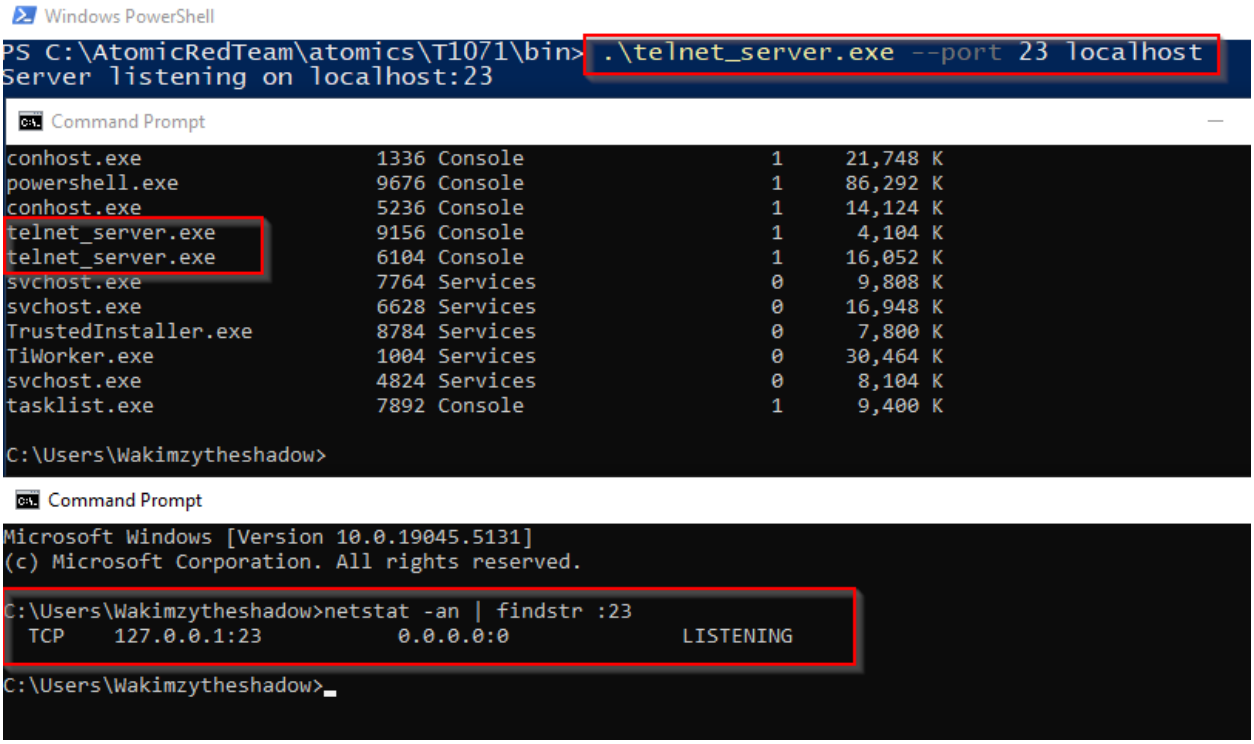


Figure 2.8: Telnet C2

2.4.5 Link File Artifacts

Link Files (.lnk) also known as windows shortcut files are shortcuts created by both users and the OS with an aim of providing access to frequently used files, folders or tools. The shortcut files include local files and files accessed from external sources such as attached storage, file shares or remote systems. Link files hold critical forensic value to an incident responder as they help establish the existence and usage of file on a system even if the original files have been deleted or moved (Lee et al., 2023). The artifacts contain metadata such as the original file path, creation, access and modification timestamps. During a breach, the metadata allows responders reconstruct user activity, identify accessed or executed files, reveal evidence of access of sensitive files and exfiltrated data which makes link files a key source of understanding the scope and timeline of an incident.

Figure 2.9 shows MITRE technique that adversaries can use to establish persistence on a compromised system by adding malicious files to the startup folder which will execute every time the user logs in. The shortcut file can be used to point to a malicious script providing the attacker with an opportunity to execute arbitrary commands or payloads. Detecting unauthorised shortcuts

in the startup locations is critical for identifying potential persistence mechanism during incident response.

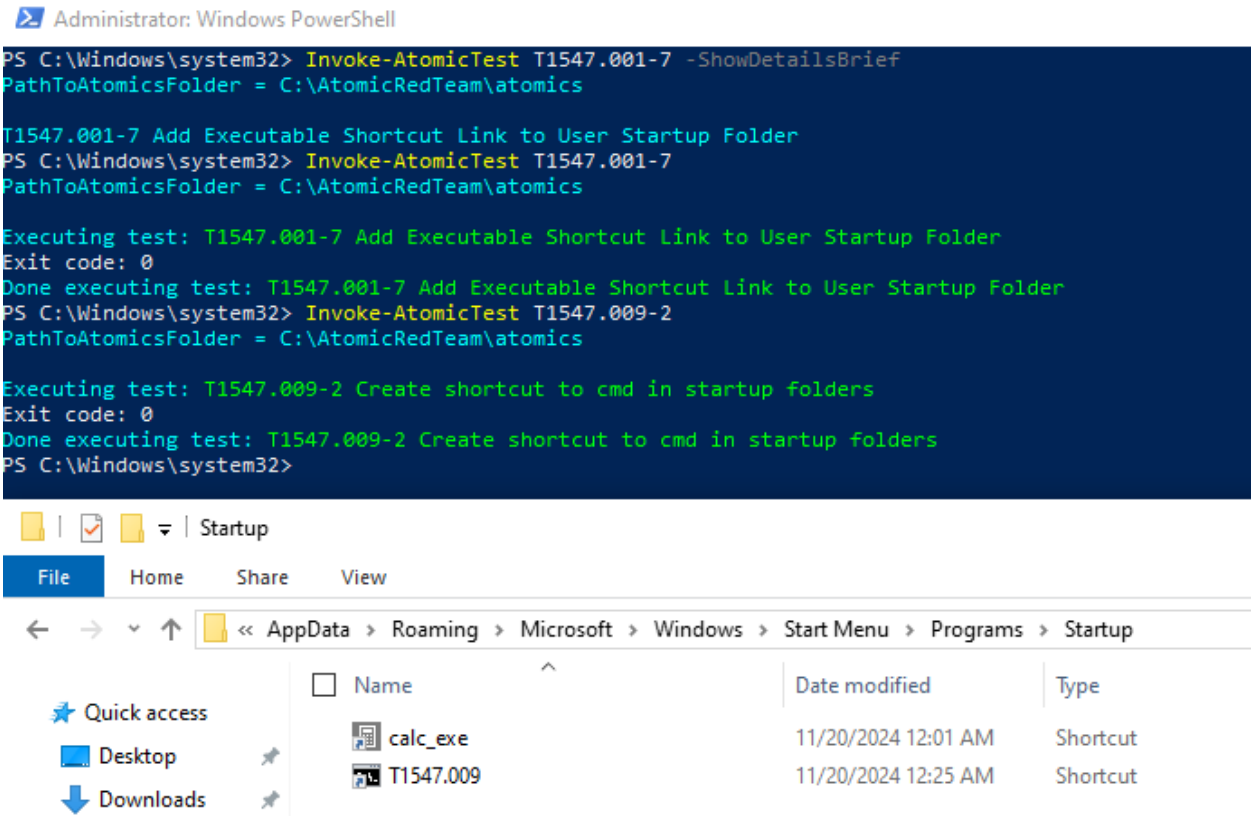


Figure 2.9: Shortcut file added to the startup folder

2.4.6 User Assist Keys

User assist data is crucial for incident responders as it provides detailed information about the programs executed by user on a Windows system. User Assist data contains a set of registry keys located in the windows registry and it tracks the tools a user has launched including the timestamp, file path, execution count and date of the last execution which aids incident responders in identifying potentially malicious or unauthorised programs. The examination of user assist data provides incident responders with means of reconstructing a timeline of user activity, uncover signs of compromise and detect signs of compromise and cross reference it with other system logs to detect suspicious behaviour and other attack vectors within the network (Smith et al., 2018). Figure 2.10 illustrates the initialization of process phisery.exe used by threat actors for credential harvesting which is added to the user registry and a Powershell script is used to decode the user assist data to extract the recently run tools.

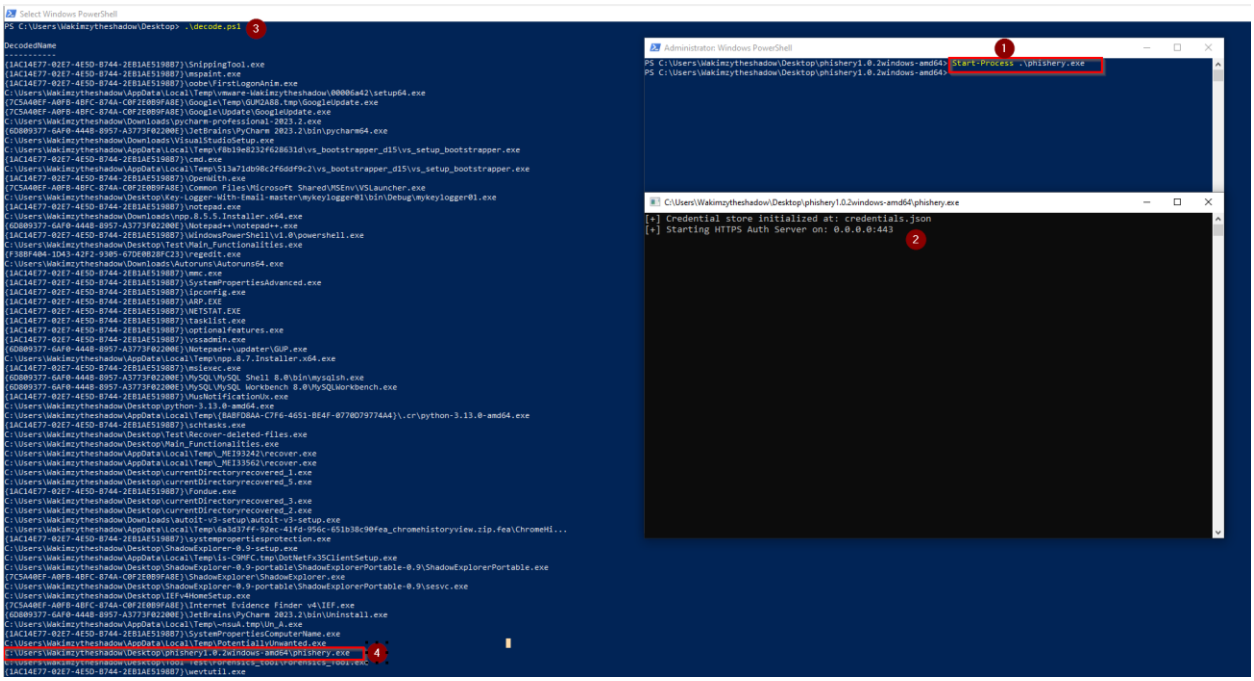


Figure 2.10: User Assist registry data

2.4.7 Scheduled Task

Scheduled tasks are automated processes setup in Windows task scheduler to execute at specific times, intervals or in response to a certain system event. They are important to incident responders because threat actors exploit them to maintain persistence, execute malicious scripts and run unauthorised tasks such as data exfiltration. Analysing scheduled tasks allows incident responders to identify anomalous entries, suspicious scripts, unexpected triggers and executables. Investigating these tasks helps identify persistence mechanism and understand how threat attackers maintain access in an endpoint (Kim et al., 2024). Figure 2.11 highlights how a threat actor can inject a startup scheduled task which creates a trigger to execute calc.exe when a user logs on to a system.

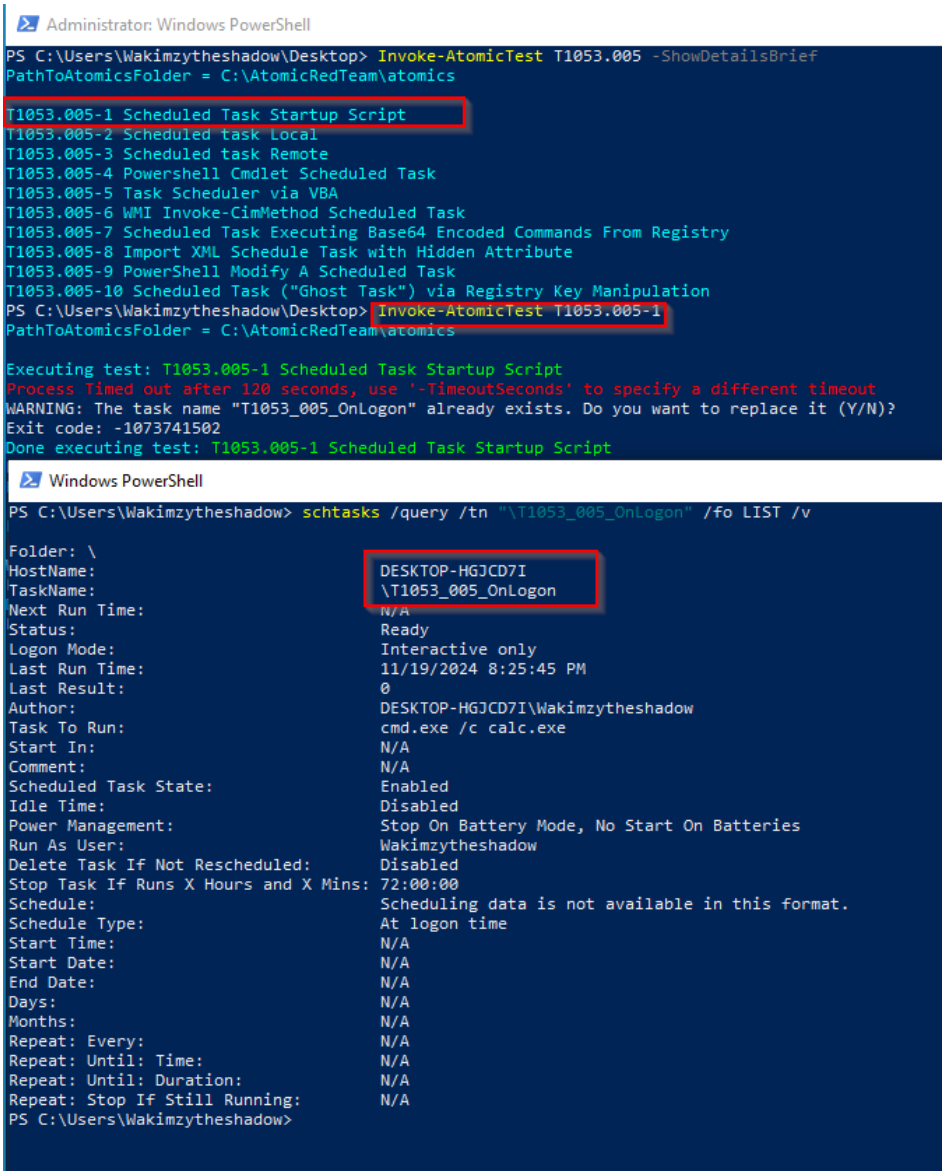


Figure 2.11: Startup scheduled task

2.4.8 Hidden Files

Hidden files are files deliberately concealed from standard user's views by using attributes that make them invisible to file explorer and common system utilities. Threat actors often create hidden files to store malicious payloads, tools or log of their activities ensuring they remain undetected by users and security controls. Uncovering and analysing these files is critical for identifying anomalous or malicious activity, understating the attackers TTP and assessing the scope of the compromise. The hidden files provide key evidence such as malware, malicious files or exfiltrated data making them vital focus during forensic investigations (Kävrestad et al., 2024). Figure 2.12

illustrates how a threat actor can create a file with a hidden attribute to hide files on the Windows system to conceal their activities and evade detection by users and basic security scans.

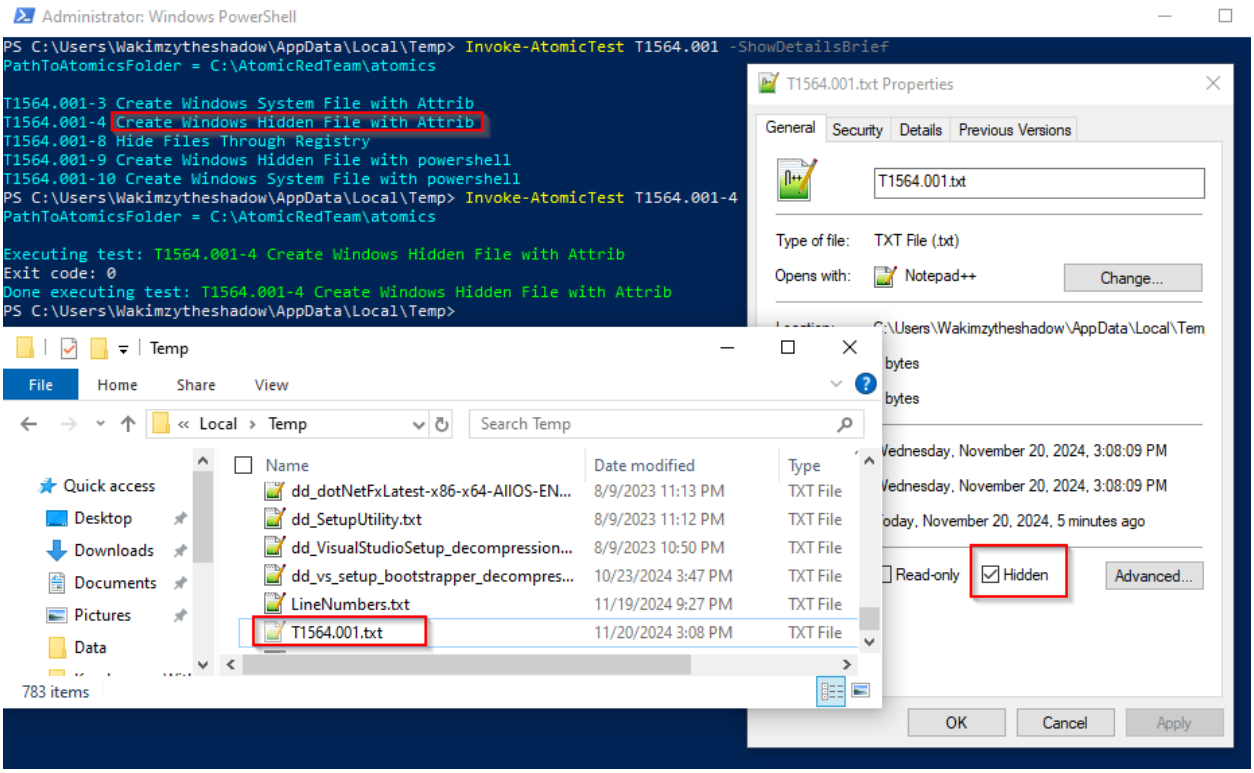


Figure 2.12: File created with hidden attribute

2.4.9 Database Files

Databases created by threat actors to capture and exfiltrate data are vital artifacts for incident response as they often contain sensitive and confidential information such as harvested credentials, exfiltrated files and logged keystrokes. Identifying the malicious databases enables incident responders assess the scope of data captured and exfiltrated, understand the attackers TTP and evaluate the impact of the exploit. Figure 2.13 demonstrates how to simulate and analyze locally staged data; the first script creates a simple database with harvested credentials and the second script scans for the database file analysing its structure showing records and columns. This emulates a threat actor logging of compromised credentials to be used for lateral movement.

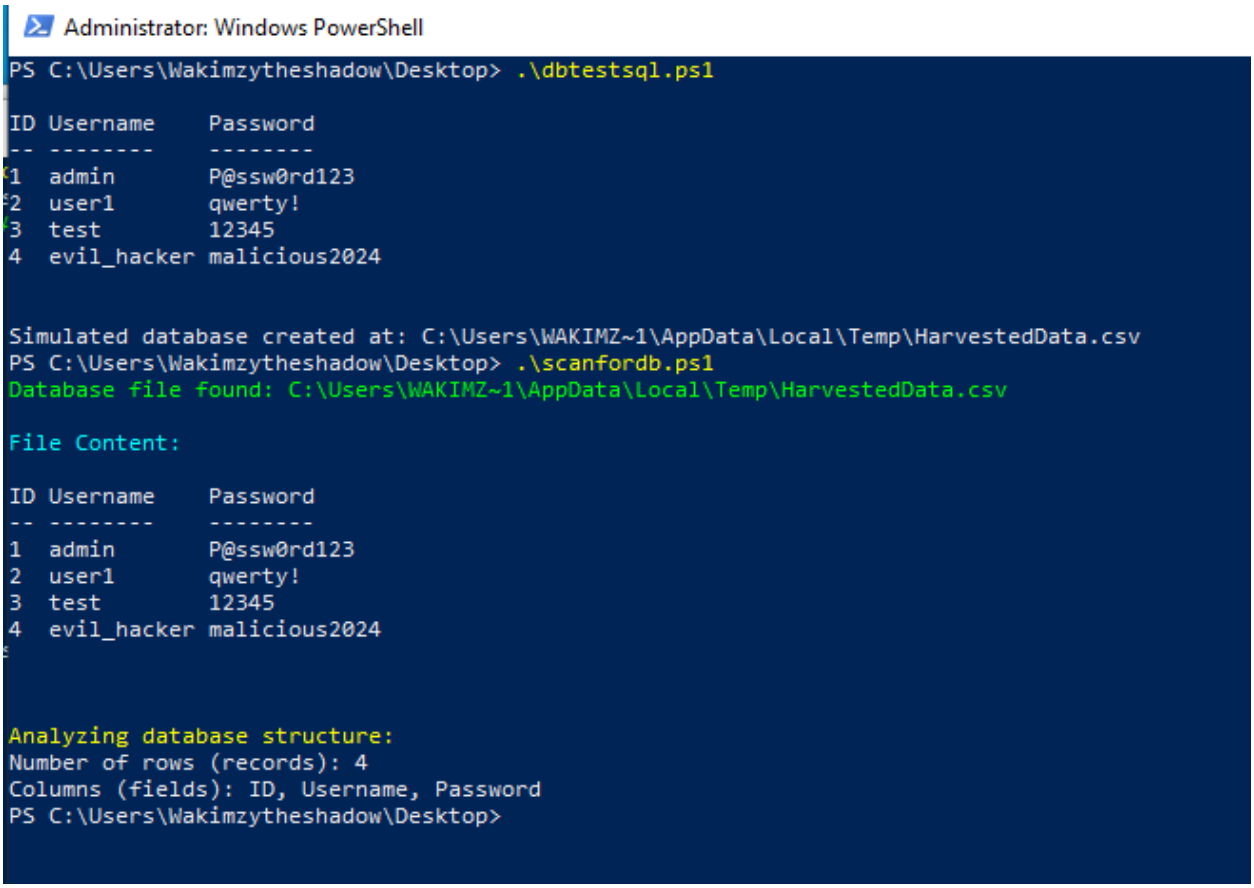


Figure 2.13: Staging database for harvested credentials

2.5 Deleted File Recovery in Incident Response

Recovering deleted files from disk is a critical aspect of the incident response process as it can provide valuable evidence and insights into the TTPs and scope of a security breach. Deleted files contain crucial indicators such as malware payloads, exfiltrated data or scripts used by attackers to evade detections from security solutions. Through the reconstruction of the deleted files, incident responders can reconstruct the sequence of events, identify vulnerabilities exploited by attackers and attribute the TTPs to a specific APT group. The recovered files inform remediation strategies and the evidence can be crucial for attribution and legal action and the capability strengthens an organizations capability to respond and recover from security incident effectively (Oh et al., 2022).

The process of recovering the deleted files requires a deep understanding of the underlying disk structure in order to locate and restore lost data effectively. File systems manage and allocate storage in file formats such as NTFS, FAT32, ext4 and APFS which translate to different ways in

which each file system handles the deletion and recovery of files. This section is going to highlight how the NTFS and FAT32 file system delete and recover data.

2.5.1 File Record on NTFS and FAT32

In NTFS, a file record consists of a 48-byte record header and a property list. The header includes key metadata, such as a "FILE" start flag, the offset of the first attribute, file status such as deleted, regular file, or directory, and the actual record length. File-related data is managed as attributes, which are categorized into resident and non-resident attributes. Resident attributes store their values directly in the file record, while non-resident attributes, if too large, are stored externally in a region called a Data Run. For fragmented non-resident attributes, NTFS uses multiple Data Runs, tracked in the Run List, to manage discontinuous data efficiently (Zhang et al., 2020).

FAT32 supports long file names by recording them in directory entries, with each entry storing 13 characters encoded in Unicode (2 bytes per character). Long file names span multiple directory entries, arranged in reverse order to avoid conflicts. These entries contain metadata like a serial number and checksum but omit details like the start cluster or file size. To maintain compatibility with older systems, FAT32 also generates a short file name for each long file name, stored alongside its directory entry, allowing access via either name format(Zhang et al., 2020).

2.5.2 Deleted File Recovery on NTFS and FAT32

When a file is deleted in FAT32 file system, the first byte of its directory entry is replaced with "E5." If the file spans a large cluster, the high cluster number in the directory entry and the FAT table entry for the file are cleared. This makes recovery impossible if the file is fragmented, overwritten, or its high cluster bits are erased. Otherwise, recovery involves locating the deleted file’s directory entry, extracting the file name, starting cluster, and file size, and then reading the data from the specified cluster in the data area into a new file (Meisheng & Huang, 2008).

When a file is deleted on NTFS, its file flag is set to 00H, but other attributes remain unchanged, simplifying recovery. The process involves scanning the MFT to identify records with a 00H flag, extracting the file name from the file name attribute, and analysing the data attributes. For resident attributes, the file data is stored directly in the attribute value. For non-resident attributes, the Run List identifies the logical clusters containing the data, which are checked against the bitmap file to confirm they have not been overwritten. If intact, the data is read and the file is recovered.

2.6 Recent Works in Extracting File Artifacts and Recovering Deleted Files

2.6.1 File Artifact Extraction

FTK Imager is a forensic imaging tool that captures exact copies of drives, including deleted files and slack space, useful for in-depth file system analysis. While powerful, it can be complex for non-forensic users and may require substantial system resources during large-scale extractions (Himanshu et al., 2021).

EnCase Forensic offers comprehensive file artifact extraction, including examining the Windows registry, deleted files, and file system structures. It is highly effective for forensic investigations, but it is a commercial product, meaning it comes with a high cost (McCluskey et al., 2022).

Autopsy, an open-source digital forensics platform, is often paired with Sleuth Kit, a collection of command-line tools designed for analyzing disk images. Autopsy excels in file system analysis and recovery but may have a steeper learning curve and limited support for newer file systems (Farnan et al., 2024).

X1 Search is a powerful indexing tool designed to quickly search and recover artifacts within large volumes of data, though its main limitation is that it is not as deep in file recovery compared to specialized recovery tools (McCluskey et al., 2022).

2.6.2 Deleted File Recovery

Recuva, a consumer-grade tool, can recover files deleted from both FAT and NTFS file systems. It is simple to use and free for basic recovery, but its recovery capabilities are limited to less complex scenarios and may struggle with deeply corrupted or overwritten files (Salman et al., 2023).

R-Studio is a professional-grade data recovery tool that works across various file systems such as NTFS, FAT, exFAT and can recover files from damaged or corrupted partitions. It offers powerful recovery capabilities, but its user interface can be intimidating, and it may require a paid license for full features (K. A.-P. Angamutu & Selvarajah, 2023).

Disk Drill supports both Windows and macOS systems and is known for its ability to recover a wide range of file types. While it provides an intuitive interface, its free version has limited

features, and deep recovery of very fragmented or overwritten files may not always be successful (K. A. Angamutu et al., 2020).

Windows File Recovery is a free command-line tool by Microsoft designed for recovering deleted files from NTFS, FAT, exFAT, and ReFS file systems. While effective, it lacks a graphical interface, which could deter casual users, and the command-line interface limits flexibility compared to GUI-based tools (Microsoft, 2024).

2.7 Gaps in the Existing Works

While these tools are crucial for digital forensic investigators and data recovery professionals, they have certain limitations. Tools such as FTK Imager and EnCase Forensic demand significant system resources and are associated with high costs, making them less suitable for most security environments. Additionally, these tools may not offer the ability to both extract artifacts and recover deleted files in one workflow, requiring investigators to rely on multiple separate tools. Autopsy and Sleuth Kit, while powerful, can be difficult for beginners due to their command-line interfaces. Tools such as Recuva and Disk Drill, while user-friendly, often face challenges in deep recovery scenarios, particularly when the file system is fragmented or overwritten. Furthermore, R-Studio and Windows File Recovery are highly effective for data recovery but require specialized knowledge, and their paid versions can be expensive. Another challenge faced by these tools is the difficulty in extracting artifacts in a timely manner, as the process can be slow, particularly with large or complex datasets (Carvajal et al., 2013).

Popular tools like Autopsy and EnCase, while widely used, often struggle with certain limitations, such as the inability to handle fragmented or partially overwritten files effectively, slower processing speeds for large datasets, and challenges in parsing complex file systems like NTFS or FAT32 with precision. These gaps can hinder the efficiency and accuracy of investigations, especially in time-sensitive scenarios.

A dedicated tool tailored to these needs can provide faster and more accurate extraction of file artifacts, enabling responders to recover essential evidence that might otherwise remain inaccessible. Additionally, such a tool could integrate seamlessly into forensic workflows, offering features like advanced metadata analysis, detection of hidden or obfuscated files, and better handling of fragmented data. These capabilities would empower responders to reconstruct events

more comprehensively, identify key indicators of compromise, and respond swiftly to mitigate damage. In high-pressure environments, where even small delays or missed evidence can have significant consequences, the availability of a tool designed to address these challenges could dramatically enhance the overall effectiveness and reliability of incident response efforts.

2.8 Conceptual framework

The design of the tool consisted of two main architecture layers: client and the drive layer. Figure 2.14 shows the conceptual framework of the incident response tool.

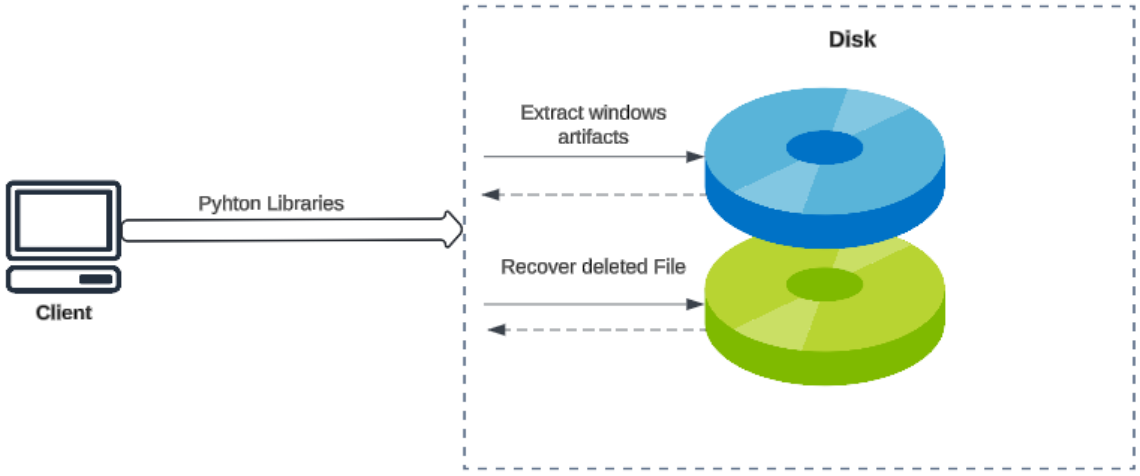


Figure 2.14: Conceptual framework

The client layer acts as the GUI that the user interacts with the tool. The layer is designed to provide the user with the capabilities to extract multiple windows artifacts and export as report and recover deleted .exe file from different disk drives.

Windows artifacts are remnants of data stored on disk during an actor's normal operation of a system. During Incident response, the incident response tool extracts the artifacts from disk layer which contains the actor's activities or malware traces. The extracted artifacts provide insights for blue team operators which include indicators of compromise, which are the timeline of events that are crucial in identifying, containing and remediating an incident.

2.9 Summary of the Literature Review

In summary, the reviewed literature underscores the critical role of incident response in minimizing the impact of cyber threats through timely detection, analysis, and recovery. Given the dominance of the Windows operating system in both enterprise and personal computing environments accounting for over 72% of the global desktop OS market, cyberattacks are predominantly tailored to exploit Windows based vulnerabilities. As a result, Windows-specific artifacts such as event logs, registry entries, prefetch files, and shortcut links have become essential sources of forensic evidence during investigations.

The use of frameworks like Atomic Red Team, combined with PowerShell scripting, allows security analysts to simulate real-world attack behaviours and observe how different artifacts are generated during each phase of an intrusion. This has proven invaluable in refining the detection of TTPs used by threat actors. Furthermore, it highlights the value of detailed system artifacts in constructing a reliable forensic timeline and identifying IOCs.

Despite these advances, the literature reveals several limitations in existing forensic tools. Most are either commercial and prohibitively expensive or require advanced technical knowledge to operate effectively, often relying heavily on command-line interfaces. Furthermore, these tools are rarely comprehensive, often focusing on either artifact extraction or file recovery but not both. A significant gap exists in tools that can efficiently extract Windows artifacts and recover deleted .exe files within a single, user-friendly platform. This is especially problematic when investigating incidents involving stealthy threat actors who attempt to erase their presence by deleting malicious executables or modifying system traces.

To address these gaps, the conceptual framework for the proposed tool emphasized integration, usability, and practical forensic depth. It was designed as a Microsoft Windows application that features a graphical user interface for intuitive use, along with a robust disk-level engine capable of extracting key artifacts and recovering deleted files from logical drives. By aligning with the phases of the incident response lifecycle and responding to the limitations of existing tools, this research aimed to contribute a practical solution for incident responders working in Windows environments.

Chapter 3 : Methodology

3.1 Introduction

Research methodology refers to a set of procedures and a systematic approach used to identify, select and analyze data to a resolution for an existing problem by providing a theoretical understanding into the methods that can be applied to particular cases to achieve specific results (Ishak & Alias, 2005) . This chapter covers the methodology to be employed in the study to address the research questions and meet research objectives outlined in Chapter 1.

3.2 Research Approach

The projects objectives were addressed by designing, developing and testing a system for extracting Windows file system artifatcs, recovering deleted artifacts from memory. The process was accomplished by implementing the Rapid Application Development methodology. RAD is a strategy of Agile development which focuses on software development with efficiency, it emphasizes prototyping and iterative development (Fauzi et al., 2023).

Rapid Application Development Methodology consists of four phases: Requirement Planning, System Design, development and implementation as illustrated in Figure 3.1.

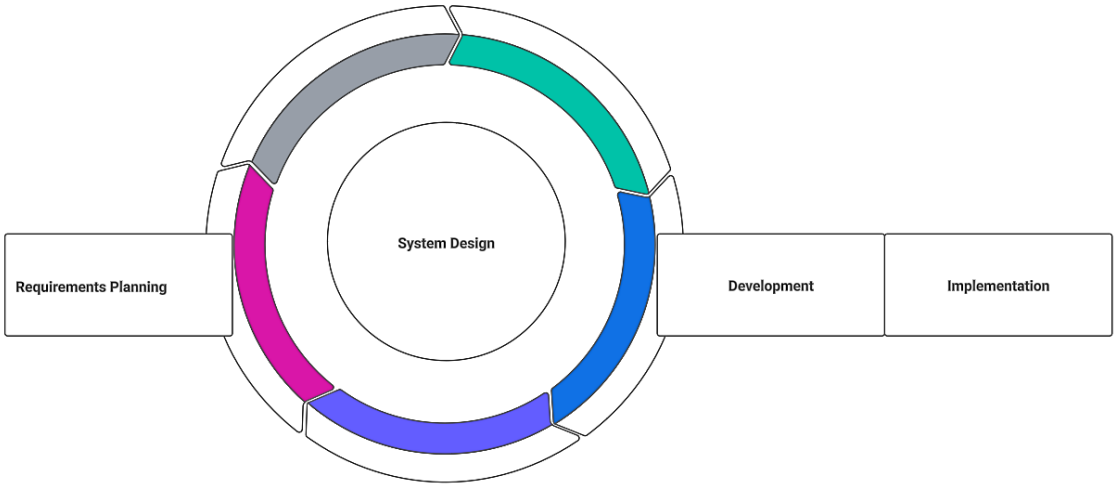


Figure 3.1: RAD Phases

3.3.1 Requirement Planning

This phase involved discovering the study's requirements through the definition of the research problem, implementation planning, analysis of risks and the procedures required to achieve the

study's objectives. This involved review meetings with the supervisor, security personnels, incident handlers who benefit from the objectives of the study. The consultations were necessary to outline the features of the proposed solution and receive analytical reviews and improvements which were included in the final solution. Implemenation planning involved understanding the dependencies and integrations of the proposed tool, documentation and review of existing tools to identify gaps which were achieved in Chapter 2 of the study. The next phase of requirement planning was risk analysis to have an understanding of the feasibility of the study, the scope, limitation and the milestones required to achieve the proposed solution in every phase.

3.3.2 System Design

This phase provided a roadmap of how the research questions presented in Chapter 1 were answered. The research employed an experimental system design. This involved setting up a virtual Windows Lab environment with various components where a mock cyber-attack was executed using the Atomic red team library of tests mapped to the MITRE ATT&CK framework to facilitate the extraction of the System artifacts outlined in Chapter 2 of the study. The aim of the cyber-attack was to dump all the artifacts discussed within the study to test the effectiveness of the proposed research tool.

Deliverables in the system design phase were contextual diagrams, system architecture which defined the scope of the study. Use Cases were presented in textual description and modelling outlining the Attack vector that threat actors employ to compromise the lab environment and the incident responder role in extracting the system artifacts for analysis. Sequence and data flow diagrams were employed to provide a high-level hierarchical view of the system processes, functions and actors interactions with the subject tool.

3.3.3 Development

Building on the requirements gathered and the system design diagrams, this phase focused on iterative development where the tool was built in rapid cycles. The implementation of the core features was based on the system requirements. Python programming language was used to develop the functionalities required to extract the windows artifatcs, C programming language to develop the functionality required to recover deleted files and the system was bundled into an exe

using py2xe. The tool employed multiple python libraries required to gain access, read and extract artifacts from disk.

The tools and framework used for the tool included:

i. Atomic Red Team - This is a library of tests mapped to the MITRE ATT&CK used to carry out the attack simulation.
ii. Idle - Idle was used as a development environment for the extraction of the Windows artifacts using python libraries.
iii. Visual studio - Visual Studio was used in developing the Recovery of deleted File source Code.
iv. VMware Workstation Pro - The hypervisor was used to run the windows lab environment where the research tool was tested and the attack simulation executed.
v. MySQL Server - The MySQL client was required for installing a DB on the test VM.
vi. Powershell - Powershell was required to create artifacts which were not bundled in the Atomic Red Team framework.
vii. Windows 10 Enterprise .iso and Windows Server 2016 .iso - The Virtual Image was used in developing the tool, executing attacks and testing the functionalities of the tool.

3.3.4 Implementation and Testing

The research developed a tool that constitutes of a software tool that provides the incident responder with a graphical user interface to interact with the configured functionalities. The incident responder has the option to extract artifacts concurrently via the GUI and output the results to a report. The tool went through a number of tests to identify whether it was in tandem with the set research objectives:

i. Functional and non-functional tests – The test was conducted to determine if the system functionalities meet the set objectives. The tool was able to extract the windows artifacts and recover deleted files used by incident responder as IOCs.
ii. Performance test – This test was done to ensure that the tool is able to extract different artifacts concurrently based on the incident responder selection.
iii. Integration test – This test was done to ascertain the tool can run on both the Windows PC and server operating system versions.

iv. User tests – This test ascertained the ease at which the incident responder was able to navigate through the tool satisfactory and rate the look and feel of the tool.

3.4 Sampling

This research study employed the purposive sampling technique which involved intentionally selecting participants in a study based on the qualities and characteristics that the participant possesses (Etikan, 2016). This sampling technique required that the chosen individuals have the expertise and familiarity with the subject matter, in this case, the researcher required the participants to have substantial knowledge in security operations and incident response. The total sample space for this study was five incident responders.

3.5 Ethical Considerations and Approval

The Incident response tool was developed and tested solely to evaluate its functionality in recovering Windows artifacts and deleted files. Testing was conducted exclusively on the virtual machines specified on this research. The research required ethical approval from Strathmore University which was essential to validate the study and authenticate the results. The study carefully considered key ethical principles, including confidentiality, data validity, voluntary participation with informed consent, risk minimization, and the integrity of research methods. To uphold these ethical standards, the researcher:

i. Established a trust agreement between the researcher and the participants, ensuring that both parties provided informed and explicit consent to the study's requirements.
ii. Ensured the data collection process adhered to the regulations outlined in the Data Protection Act of Kenya.
iii. Adhered to informed consent guidelines, which are established and safeguarded in Kenya's laws through National Commission for Science, Technology, and Innovation.
iv. Applied the third ethical principle, which was derived from the Economic and Social Research Council and It emphasizes that "the confidentiality of information provided by research subjects and the anonymity of respondents must be respected"(Jerrim & Vries, 2023). In situations where confidentiality was restricted, anonymity was promoted

3.6 Chapter Summary

Chapter Three presented the research methodology used in the development and evaluation of the tool. The study adopted the Rapid Application Development (RAD) model, which emphasized iterative prototyping and regular user feedback to facilitate rapid system development and refinement. This approach enabled continuous improvements to the tool based on practical testing results and stakeholder input.

The chapter described the tools and frameworks that were utilized during development and testing. The implementation phase included comprehensive testing, which was categorized into functional and non-functional testing, performance testing, and user testing. These testing phases were essential in verifying the tool's capability to extract Windows artifacts and recover deleted executable files effectively.

A purposive sampling method was employed, selecting participants based on their expertise in cybersecurity and relevance to the study. A total of five participants took part in the evaluation. Additionally, ethical approval was obtained from Strathmore University, ensuring that all procedures complied with institutional ethical standards and that participant rights and confidentiality were protected throughout the research.

Chapter 4 : System Analysis, Design and Architecture

4.1 Introduction

This chapter presents and describes the various functionalities of the system based on the requirements and methodology discussed under Section 3.3 of chapter three. The functionalities of the proposed tool were achieved through use case diagrams, sequence diagrams and data flow diagrams.

4.2 Requirement Analysis

This section discussed the functional and non-functional requirements identified through the information gathered during the requirement planning and system design phase and its operational constraints.

4.2.1 Functional Requirements

The functional requirements are statements of service explaining the process flow and activities that are carried out within the tool for its intended users. The functional requirements were identified from literature gathered in chapter two of this research. The functions were defined as follows:

i. File system artifacts acquisition – The tool shall enable the users to extract the relevant and required Windows artifatcs from a specified drive.
ii. Reporting and Exporting data – The tool will generate a .csv report with all the indicators for Incident response.
iii. Input capabilities – The tool will provide a query feature to select the intended drive for querying and recovering artifacts.
iv. Concurrent Extraction of artifacts - The tool should allow the user to extract artifacts in parallel to expedite the collection process.

4.2.2 Non-Functional Requirements

The Non-functional requirements are not essential to core of the tool but they play a role in ensuring the system is operational. These include:

i. Usability – The tool should have an interface that is easy to interactive and should be intuitive.

ii. Performance - The tool should extract and process the artifacts to the user without any interruptions and be able to handle multiple requests.
iii. Compatibility – The tool should be able to extract data from Windows 8, 10 and 11 versions.

4.3 System Architecture

System architecture acted as a conceptual representation and alignment of the system design and functional requirements. The design of the tool consisted of two main architecture layers: client and the drive layer. Figure 4.1 shows the architectural design of the incident response tool.

4.3.1 Client Layer

The client layer of the incident response tool provides a user-friendly graphical interface designed to support efficient analysis within a Windows environment. Developed using Python's Tkinter library, the interface allows incident responders to interact with the tool without relying on command-line operations. The GUI is organized into multiple tabs, each corresponding to a specific function such as artifact extraction, log viewing, or file recovery enabling users to switch between tasks seamlessly. This tabbed layout allows simultaneous navigation and operation, so the responder can extract different types of Windows artifacts in parallel without restarting or interrupting ongoing processes. The interface also includes scrollable output areas for real-time feedback and integrates background threading to ensure the system remains responsive during long-running tasks. Overall, the design emphasizes ease of use, task separation, and multitasking capabilities, making it practical for real-time incident response and forensic workflows.

4.3.2 Disk Layer

The disk layer of the incident response tool is the core component responsible for interfacing directly with the Windows operating system to extract forensic artifacts and recover deleted data. Built using a combination of powerful Python libraries, this layer performs low-level operations such as accessing registry entries, reading system logs, analyzing file metadata, and interacting with the file system. It enables the tool to gather essential evidence from system directories, databases, and registry hives, supporting tasks like timeline reconstruction and behaviour analysis.

A key feature of this layer is its ability to recover deleted .exe files potentially malicious executables removed by threat actors to cover their tracks. Through the selection of a logical drive,

the tool initiates a scan of the disk, including unallocated space, to locate and restore these deleted files. This recovery process is crucial for identifying traces of unauthorized activity or malware presence. The use of modules like winreg, win32file, shutil, and ctypes ensures access to protected system areas, while sqlite3, os, and Path support file and database operations. Timestamp interpretation using calendar, datetime, and time further enhances the artifact analysis process. Together, these capabilities make the disk layer a robust foundation for comprehensive investigation within a Windows environment.

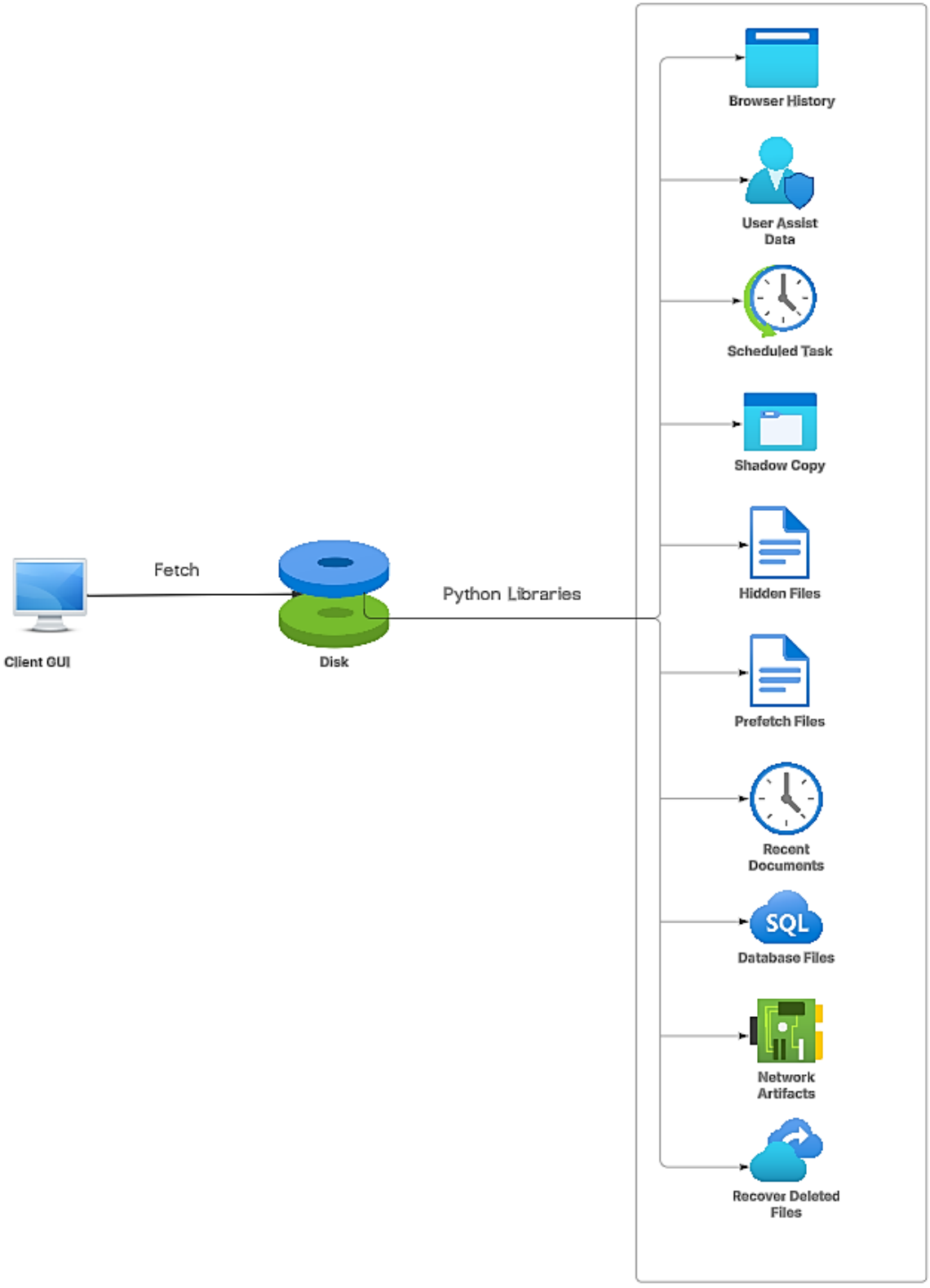


Figure 4.1: System Architecture

4.4 System Design

The system diagrams were used to model the conceptual structure and illustrate the behaviour of the artifact extraction tool. These diagrams provide a logical overview of the system and offer detailed insights into its implementation. Aligned with the RAD Methodology, the diagrams developed include the use case diagram, system sequence diagram, data flow diagram (DFD), and wireframes each contributing to the planning and understanding of how the tool extracts and processes Windows artifacts during investigations.

4.4.1 Use Case Diagram

Use case Diagram showcases and summarizes the interactions between the actor and the different functionalities in the tool. The main actor in this research is the incident responder as illustrated in Figure 4.2.

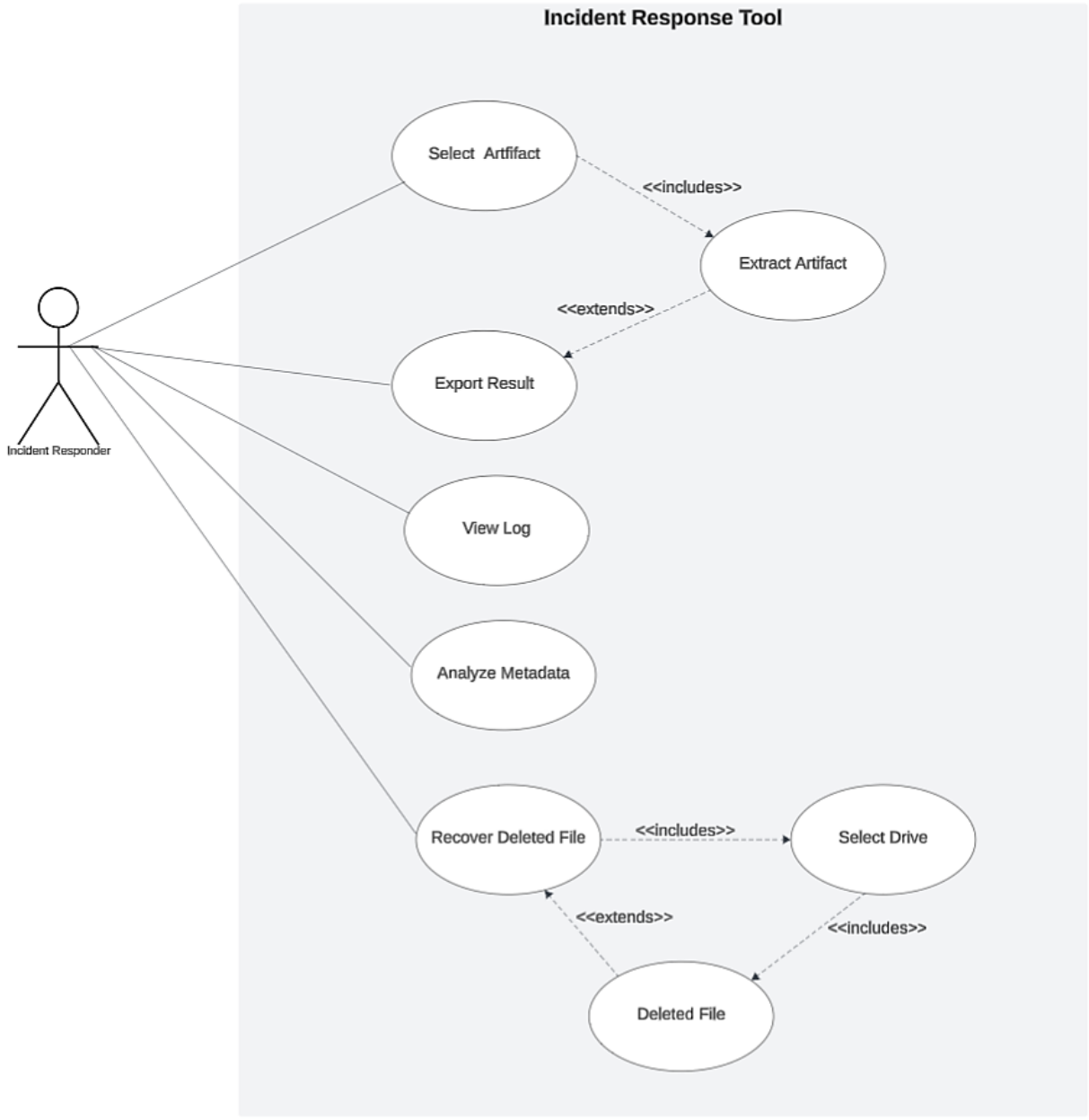


Figure 4.2: Use Case Diagram

Use Case Descriptions

i. Select Artifact – The responder selects the specific type of Windows artifact to extract, such as registry hives, event logs, or prefetch files, based on the focus of the investigation.

ii. Extract Artifacts – The system scans the file system and relevant directories to retrieve the selected artifacts. Data is parsed into readable formats while preserving key metadata.

iii. Export Results – Extracted data can be exported in CSV formats for reporting, documentation, or further investigation.

iv. View Log – A detailed log records all actions performed by the system during artifact extraction and analysis, supporting traceability.

v. Analyze Metadata – The analyst reviews metadata (timestamps, file paths, file sizes) to help reconstruct user actions and system events.

vi. Recover Deleted EXE Files – The user is given the option to recover deleted .exe files from a specified drive or location, allowing targeted file recovery as part of the forensic process.

4.4.2 Sequence Diagram

Sequence diagram illustrates all the events initiated by an actor within a tool. Figure 4.3 illustrates the order in which the user interacts with the tool.

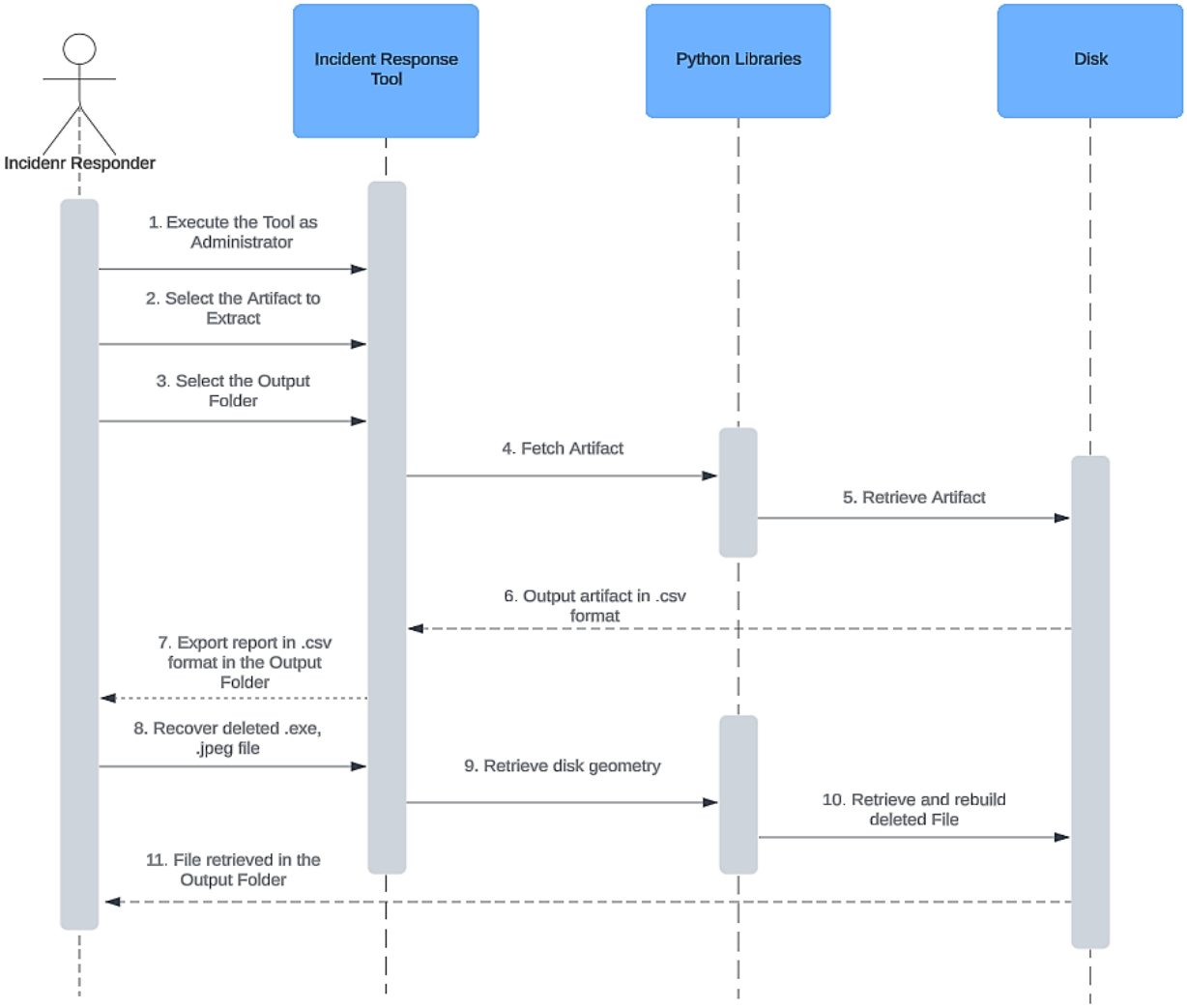


Figure 4.3: Sequence Diagram

The incident responder, acting as the primary user, runs the tool as an administrator to extract a selected Windows artifact. Upon execution, the tool presents a graphical interface that allows the responder to navigate through various Windows artifacts and choose the required one. Additionally, the responder must specify an output folder, which must be located within the same directory as the tool. Once selected, the chosen Windows artifact is extracted from the disk. The incident response tool, when run as an administrator, extracts the selected Windows artifacts from the disk. Utilizing Python libraries, the tool retrieves the artifacts from memory and outputs the extracted data in a CSV format.

The incident response tool extracts selected Windows artifacts from the disk and saves the output in a .csv format. Before execution, the output folder must be defined within the tool and created in the same directory as the tool. Once the extraction is complete, the retrieved artifacts are stored in a .csv file. The incident response tool currently supports the extraction of deleted .exe files from a selected drive, with a size limitation of 2MB. This restriction is primarily due to limited computational resources available in the virtual lab environment used during development and testing. Larger files may require more memory and processing power, which could not be fully accommodated under the current setup. To overcome this limitation, future improvements could involve deploying the tool in a more robust hardware environment or optimizing the tool's memory management to handle larger files efficiently. Additionally, integrating external storage or cloud-based processing resources could further enhance the tool's capability to extract and analyze larger executable files without compromising performance. The incident responder selects the drive for extraction and specifies an output folder, which must be located in the same directory as the tool. Once the process is complete, the extracted artifacts are saved in a .csv format.

4.4.3 Data Flow Diagram

The flow of data in the incident response tool can be showcased through a data flow diagram in Figure 4.4. The DFD categorizes the input and output data and other processes withing the tool. The main inputs in Figure are selection of artifacts, output file, drive to be scanned for deleted files, artifacts extraction and the outputs are a csv report and the files recovered.

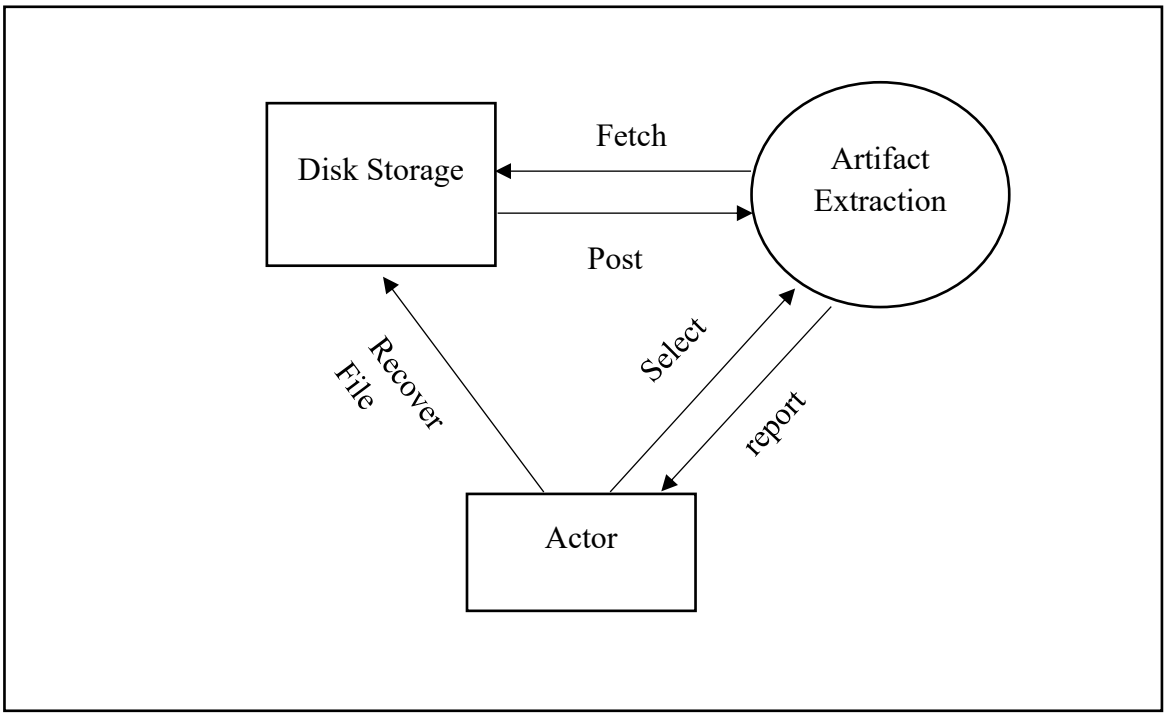


Figure 4.4: Data Flow Diagram

4.5 Application Wireframe

The IR tool will have a single dashboard as shown in Figure 4.5 where the incident responder can switch between different tabs to select the artifact which they want to extract. This allows simultaneous extraction of artifacts in the same dashboard.

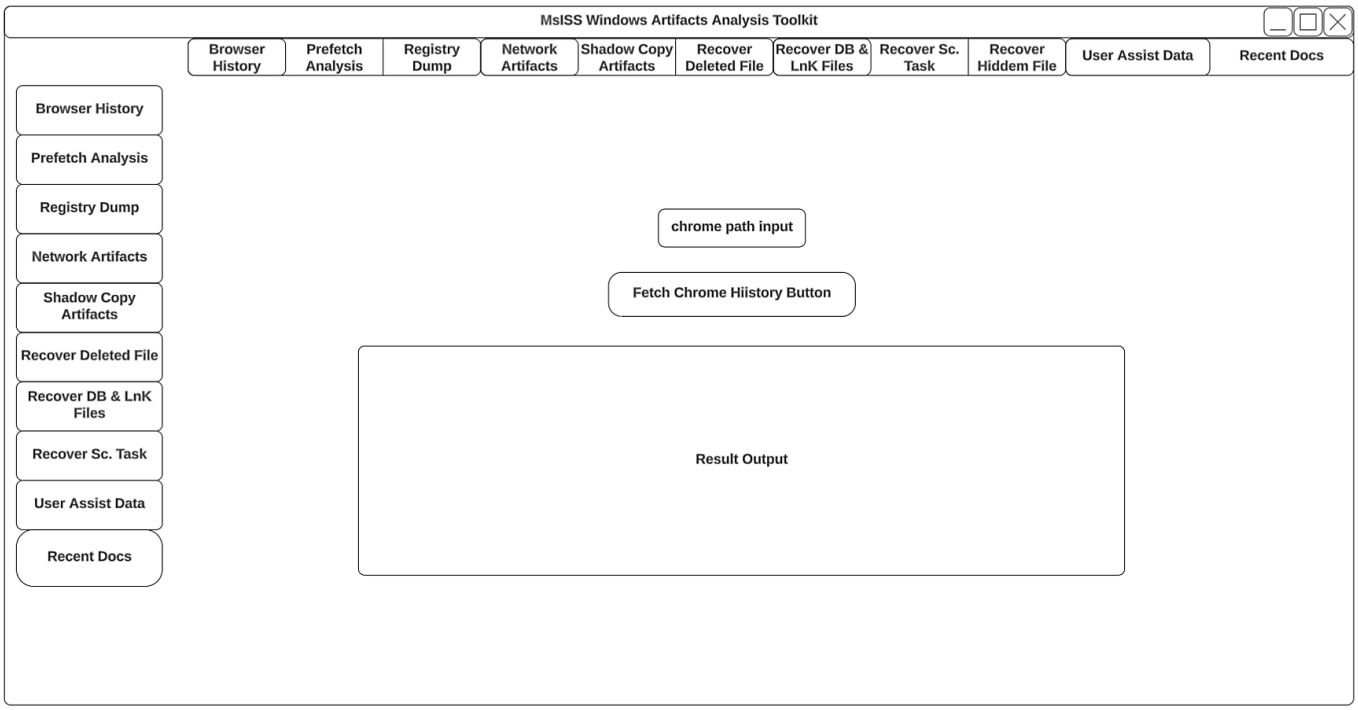


Figure 4.5: Wireframe Showing the Dashboard

4.6 Chapter Summary

This chapter detailed the system design process, starting with gathering requirements from key stakeholders and prospective users. The information collected informed the definition of the system's functional and non-functional specifications. A comprehensive model was created to outline the system's core components. Use case modelling was performed to produce descriptive use case scenarios and corresponding diagrams. User interactions with the system were depicted through sequence diagrams. Lastly, a wireframe was developed specifically for the Microsoft Windows application to visualize the user interface layout.

Chapter 5 : System Implemenation and Testing

5.1 Introduction

This chapter covered the implementation of the Incident response tool, the software components and the virtual lab environment that was used to test the effectiveness of the tool. The chapter focuses on outlining the requirements and steps that were used during the study.

5.2 System Implemenation

5.2.1 Hardware Requirements

The tool was developed on a 64-bit computer running the Windows operating system and tested on a Windows OS virtual environment. Additional specifications of the computing environment used for system development are captured in Table 5.1.

Table 5.1: Hardware Environment

Hardware	Justification
64-bit Windows Operating system .iso, core i7 Processor with 2.5GHz speed, 10GB RAM, 150 GB Hard Drive, and Network Adapter bridged.	The Virtual Image was used in developing the tool, executing attacks and testing the functionalities of the tool.
64-bit Windows Operating system .iso, core i5 Processor with 2.5GHz speed, 16GB RAM, 500 GB Hard Drive.	The laptop was used to host the Virtual environment, online research and dissertation writing.

5.2.2 Software Requirements

The development process was divided into frontend and backend. For backend development, python and C were utilized. The tools used to build the application are captured in Table 5.2.

Table 5.2: Software Environment

Software	Justification
Windows 10 Enterprise .iso and Windows Server 2016 .iso	This was the operating system used in the Virtual Lab to run the tool.

Atomic Red Team	This is a library of tests mapped to the MITRE ATT&CK used to carry out the attack simulation.
Visual Studio	Visual Studio was used in developing the Recovery of deleted File source Code.
Idle	Idle was used as a development environment for the extraction of the Windows artifacts using python libraries.
Py2xe	This python extension was used to convert the python scripts into an executable windows tool.
VMware Workstation Pro	The hypervisor was used to run the windows lab environment where the research tool was tested and the attack simulation was executed.
MySQL Server	The MySQL client was required for installing a DB on the test VM.
Powershell	Powershell was required to create artifacts which were not bundled in the Atomic Red Team framework.

5.3 Incident Response Tool

The tool was developed using Python and C programming languages and is compatible with both Windows PCs and servers. It utilizes import modules to access various libraries and functions, enabling efficient execution of common tasks.

5.3.1 Import Modules

i. import argparse - The module facilitated easier writing of a user-friendly command line tool by providing options to automatically generate help and usage messages.
ii. import sqlite3 as lite – The module was used to define SQL queries used to fetch SQL databases in the Disk.

iii. import calendar – The module provided the functions required to generate and display the dates for the extracted windows artifacts.
iv. import os – The module provided a way to interact with the OS: read and write operations and navigating through directories.
v. import time – This module was used to fetch the current time.
vi. import csv – The module was used to write the specified windows artifacts logs to a CSV file format.
vii. import operator – The module was used to sort the timeline of prefetch files.
viii. import struct – The module provided the functionality with working with C libraries used in developing the deleted file code.
ix. import winreg – The module was used to gain access and read the windows registry.
x. from pathlib import Path – pathlib module provided an approach to work with the file system, interact with the file paths and access the specified artifacts.
xi. import subprocess – The module was used to span new processes and run shell commands used in dumping scheduled processes and network information.
xii. import shutil – The shutil module was used to copy db files to the output db.
xiii. import ctypes – The module was used to call functions id DLL required in fetching hidden files from disk.
xiv. from tkinter import messagebox, simpledialog – The module was used to create and display input dialogs for the incident responder.

5.3.2 Executing the Tool

The incident responder is required to execute the tool as Administrator which is necessary for system-level resource access that is not permitted for non-admin users. This includes registry keys, network configurations and protected directories. The elevated privilege provides the tool with read and write access which bypass User Account Control.

5.3.3 Tool Dashboard

Once the incident responder has executed the tool as an administrator, they are presented with a dashboard containing all the system functionalities for selecting and extracting artifacts and recovering deleted files. The main dashboard has different tabs which allow the actor to switch and extract the artifatcs simultaneously. Figure 5.1 Shows the dashboard presented to the user.

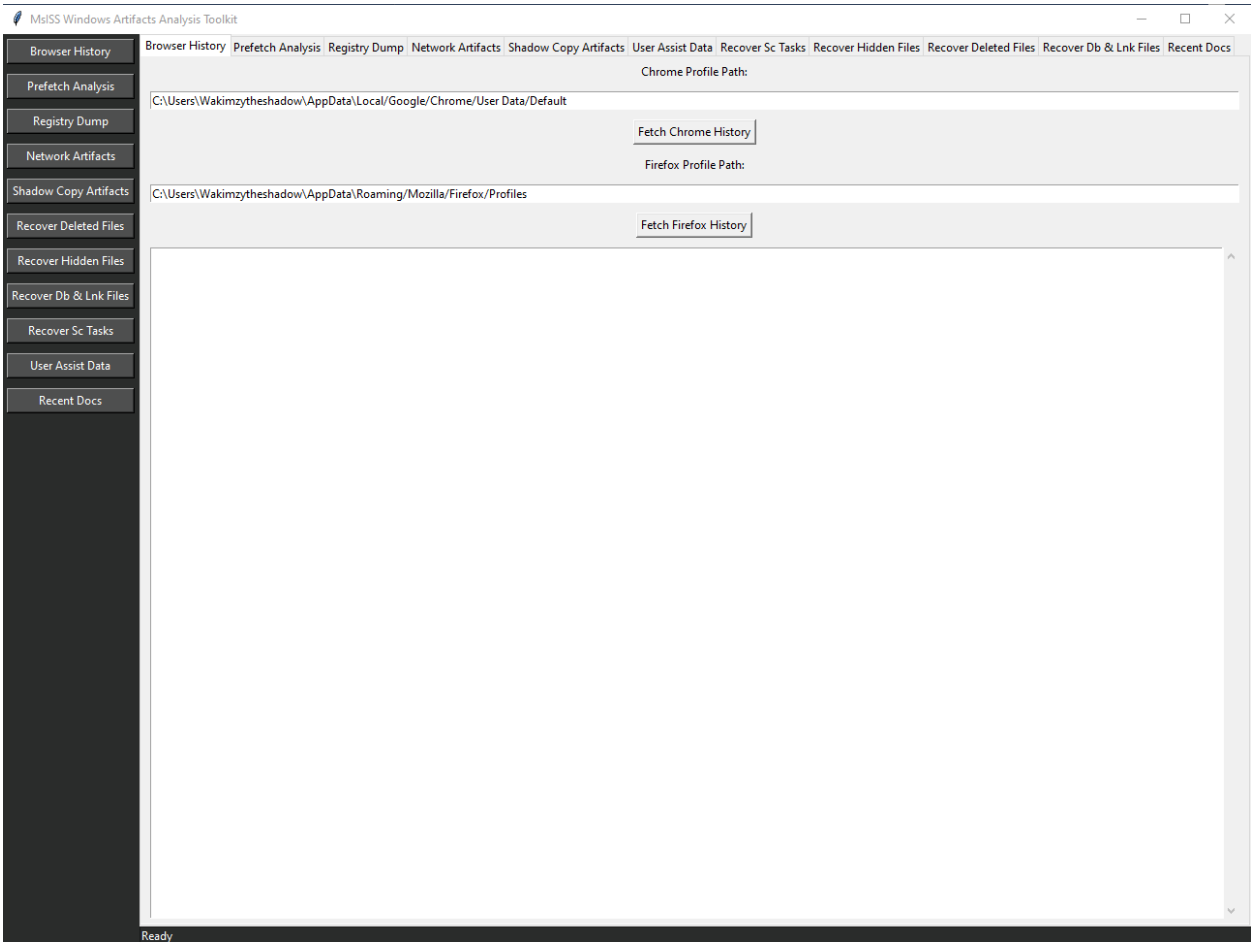


Figure 5.1: Tool dashboard

5.3.3 Artifacts Extraction

Atomic red team library tests were used to execute attacks on the virtual environment leaving traces of the windows artifacts for extraction. The section will include the tests run and the output displayed from the tool.

5.3.3.1 Fetch Chrome Browser History

The browser history tab allows the incident responder to extract chrome browser history logs from the default chrome path. The incident responder can choose to select a different path to fetch the chrome history logs if the default path does not exist. The function connects to the database, retrieves the browsing history, converts timestamps to Unix epoch time and optionally formats the data for timeline analysis. Figure 5.2 shows the browsing history captured from malicious GitHub repositories.

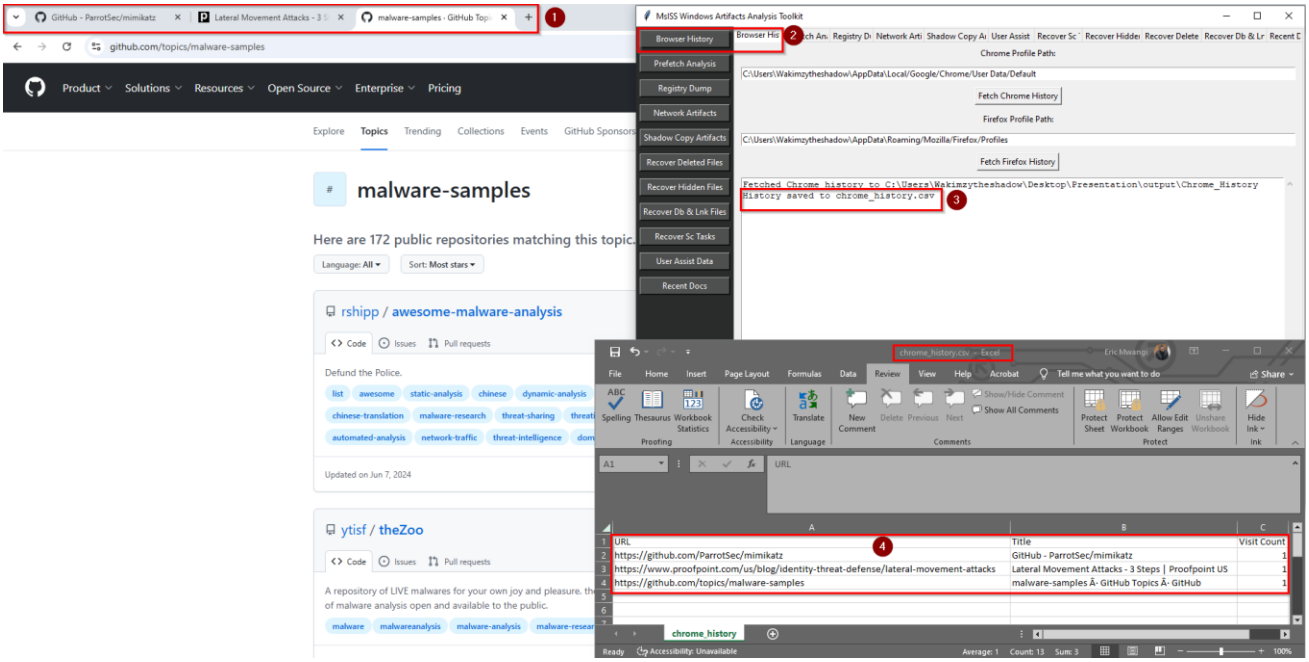


Figure 5.2: Browser history artifacts

5.3.3.2 Extract prefetch files

The prefetch files are stored in .pf extension and the init constructor sets the path to the Windows Prefetch directory, generate timeline scans for .pf files, records the first and last execution times, writes this information to a CSV file, and sorts the entries by timestamp. Figure 5.3 showcases an execution of process injection MITRE technique T1055 which is a Read-Write-Execute process Injection. Figure 5.4 shows the prefetch log captured from the incident response tool.

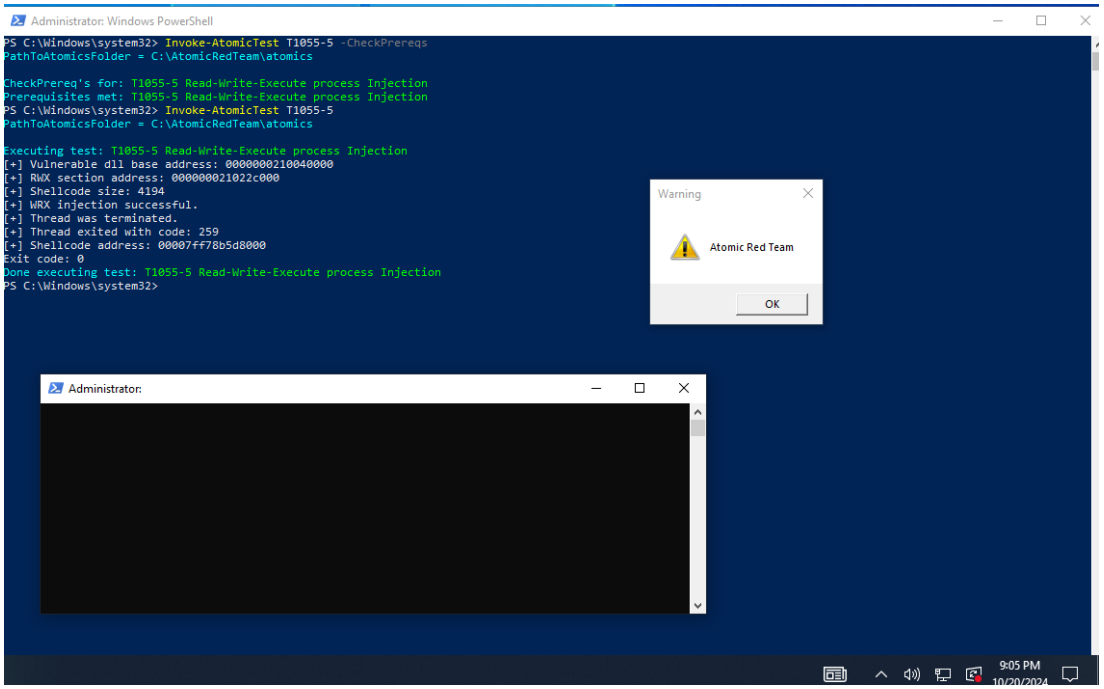


Figure 5.3: Prefetch Test

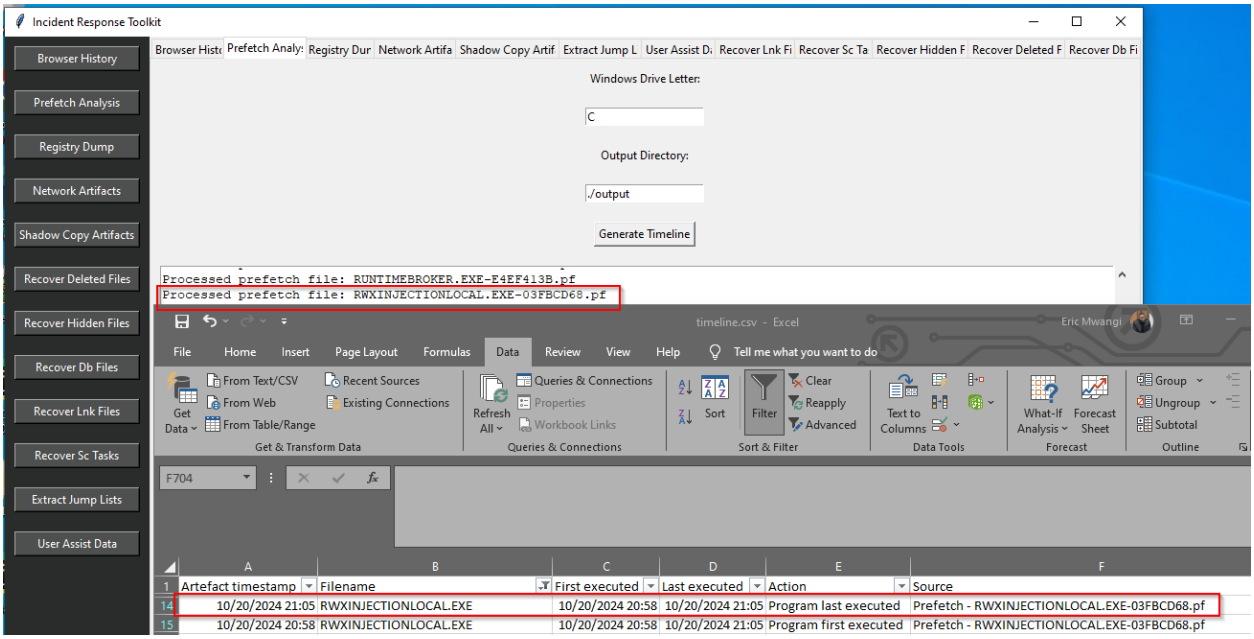


Figure 5.4: Prefetch logs

5.3.3.3 Registry Dump

The registry dumper class was defined to extract and dump data from the registry hives. The different hives used were HKEY_CLASSES_ROOT, HKEY_CURRENT_USER, HKEY_LOCAL_MACHINE, HKEY_USERS and HKEY_CURRENT_CONFIG. The function

used opens each registry hive, iterates through keys and values, and writes them to a .reg file using Windows Registry Editor format. Figure 5.5 and 5.6 dump the registry hives which can be views using Regripper or the registry editor.

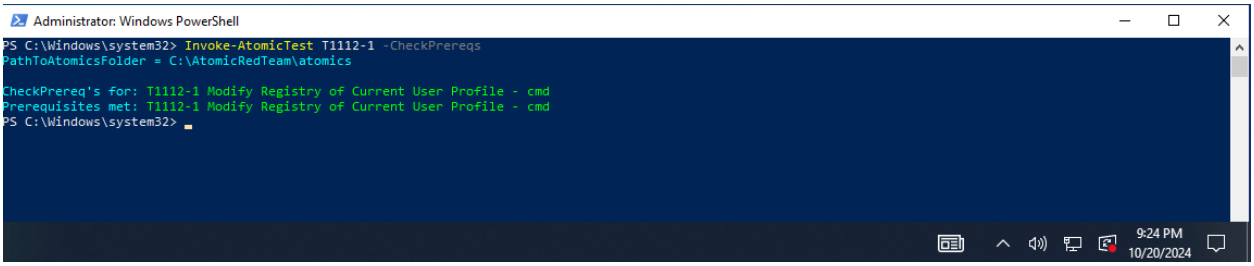


Figure 5.5: Modify Registry Test

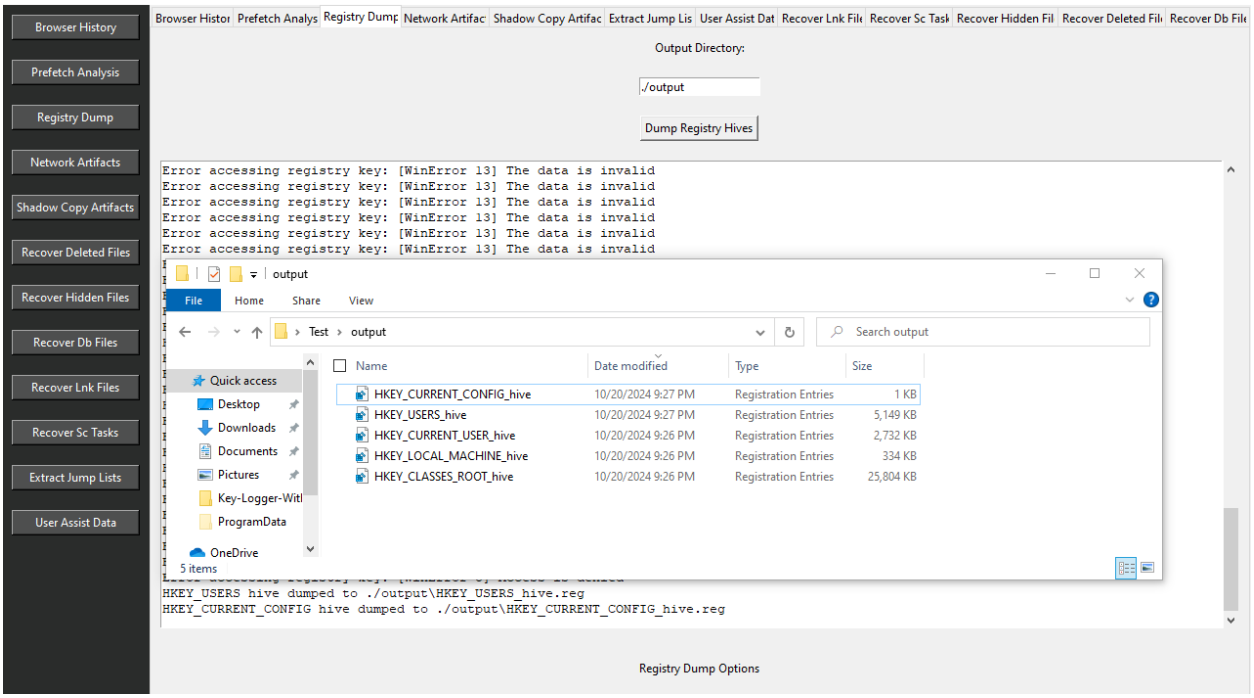


Figure 5.6: Registry dump

5.3.3.4 Network Information Dumper

The network artifatcs captured by the tool include IP configuration, ARP Table, network connections and running processes. The tool was able to fetch the artifacts by using the specific command: ipconfig, arp, netstat and tasklist which are output into a single text file for investigation and analysis. Figure 5.7 and 5.8 showcase execution of T1071-1 Telnet C2 which establishes an open connection via localhost. The connection serves as a command and control which threat

actors can use to execute arbitrary commands on the host. The network connection is captured on the tool as captioned.

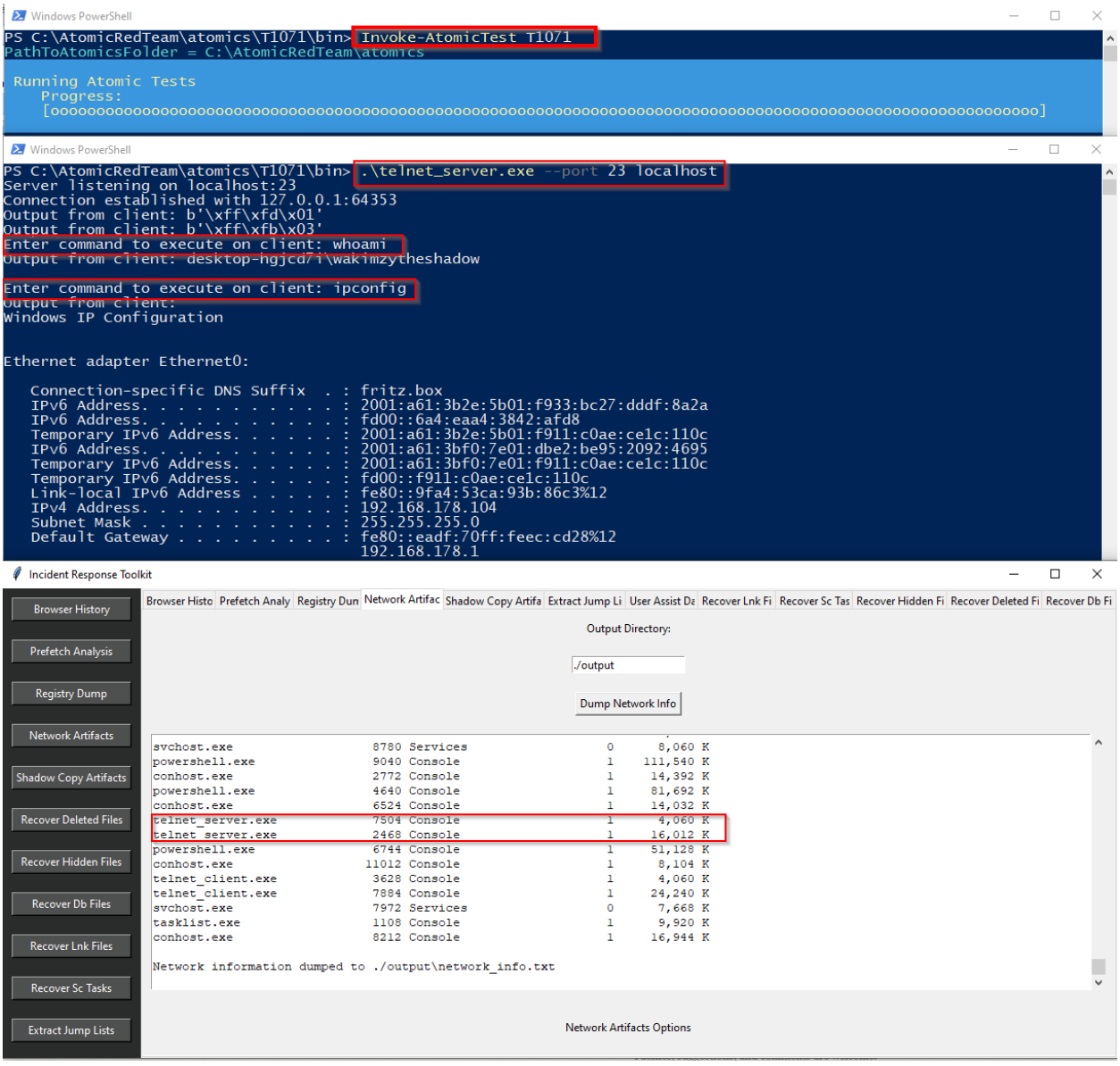


Figure 5.7: Telnet C2

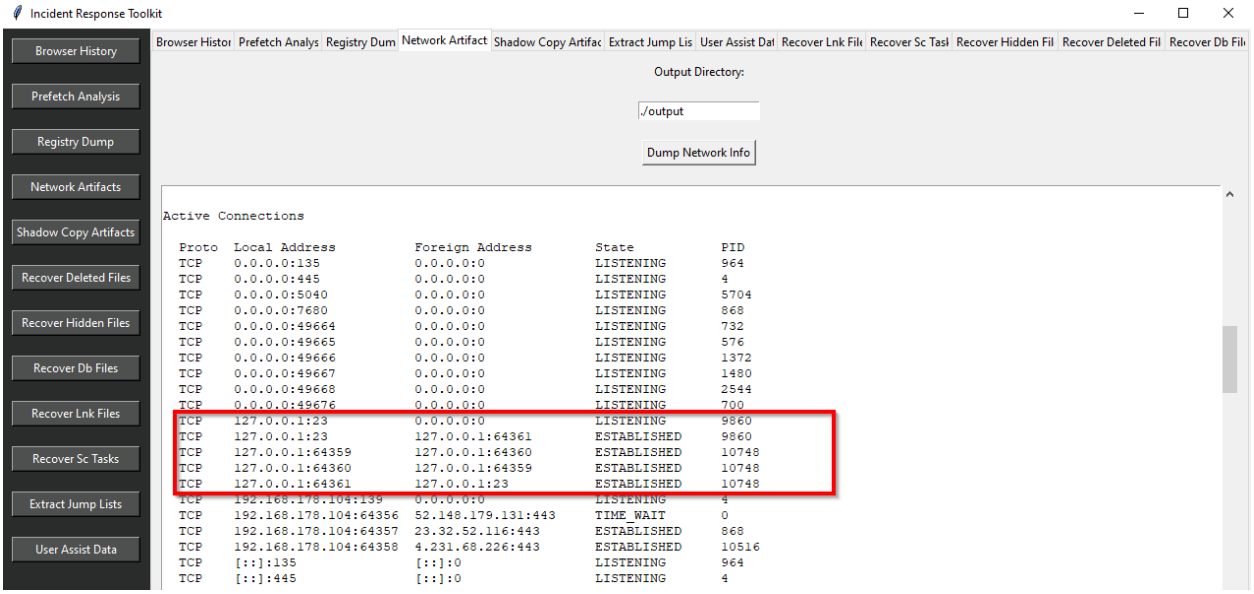


Figure 5.8: Telnet service

5.3.3.5 Shadow Copy Artifacts

WMI interface was used to create a shadow copy using a Powershell script. The script created a shadow copy on disk C:\ and the ClientAccessible parameter was used to specify the shadow copy should be accessible to the user. Figure 5.9 shows the creation and capture of the shadow copy created via the incident response tool which is saved in a CSV report.

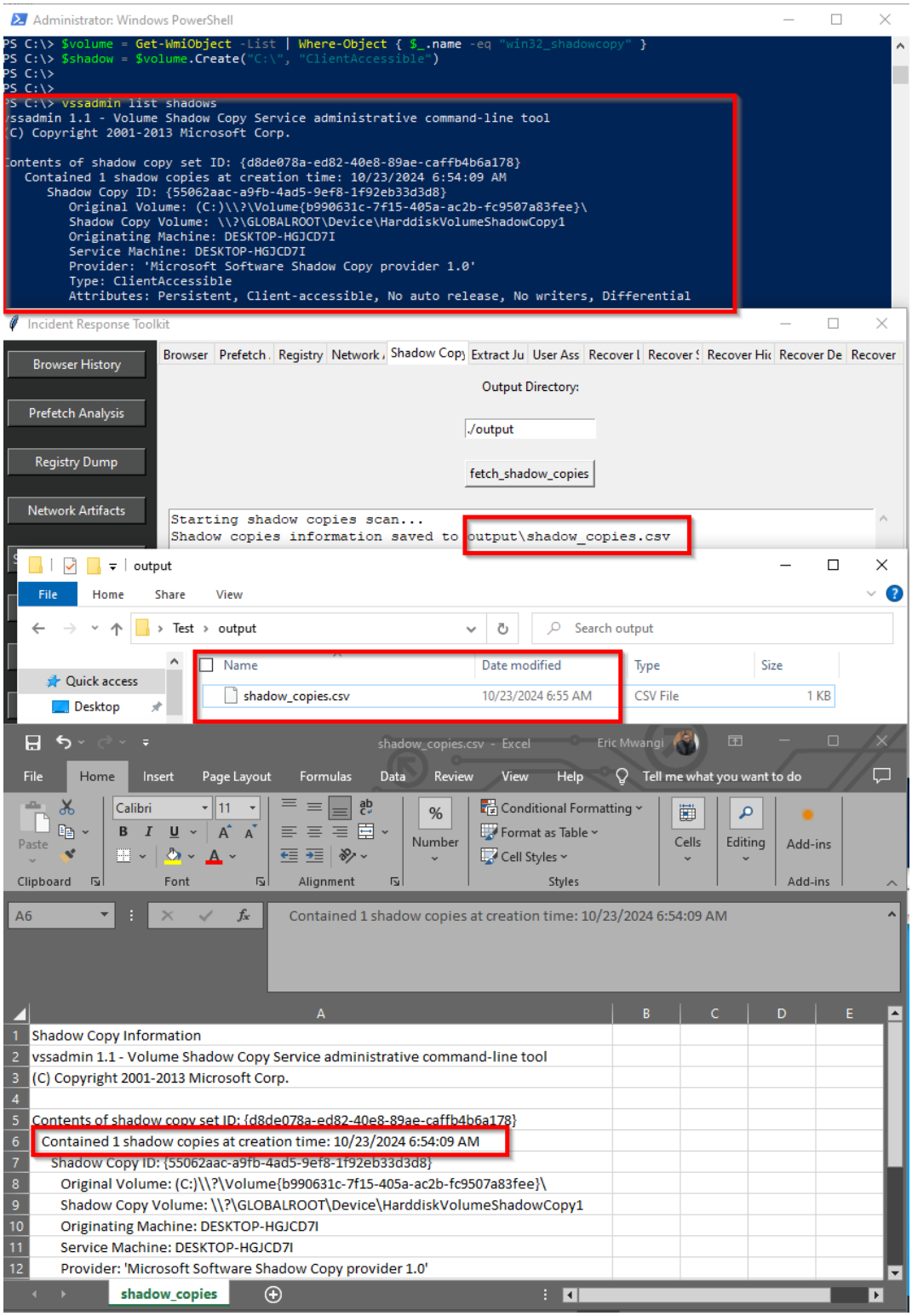


Figure 5.9: Shadow Copy Artifact

5.3.3.6 Recover Hidden Files

The atomic red team test does not have a technique for creating hidden files, this pre-empted the use of a Powershell script to create the hidden file. The attrib command was used with switch h to make the file hidden in windows explorer. Echo created the file with a specified name and path for the file. Figure5.10 shows the recovery of hidden files artifacts capturing the text file created from the Powershell script. The research tool checks for files with value attrs & 2 (attribute and bit flag 2) to check for the hidden attribute.

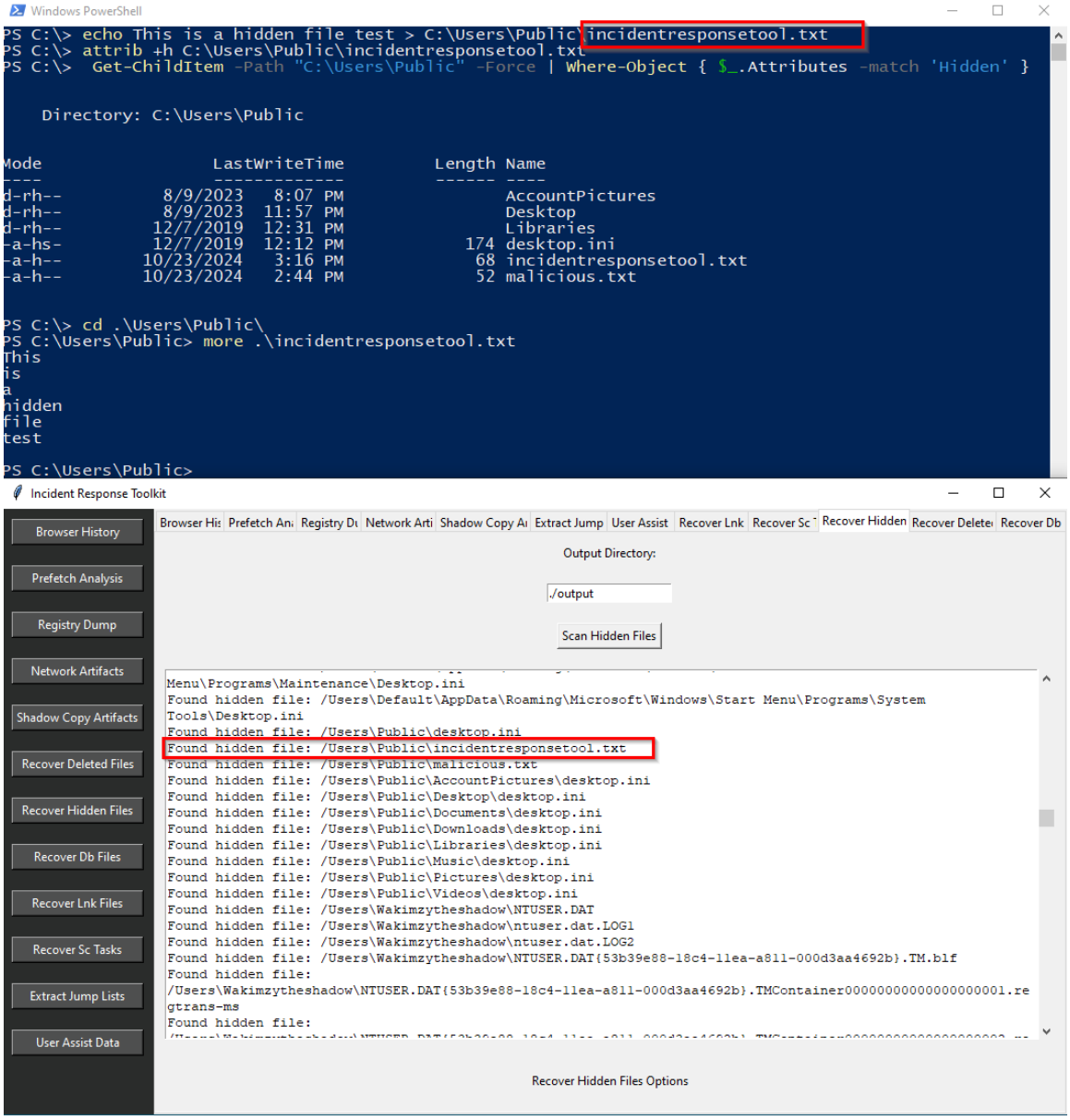


Figure 5.10: Hidden file artifact

5.3.3.7 Recover Database Files

The objective of the functionality is to scan the user's home directory for database (.db) and backup (.bak) files, determine if they are valid SQLite databases, and record their details. The tool sets up the user's home directory as the starting point for scanning. The tool looks for files with .db and .bak extensions, checks their size, and determines if they are databases using SQLite connection attempts. The tool then writes the collected file information to a text file as captioned in Figure 5.11.

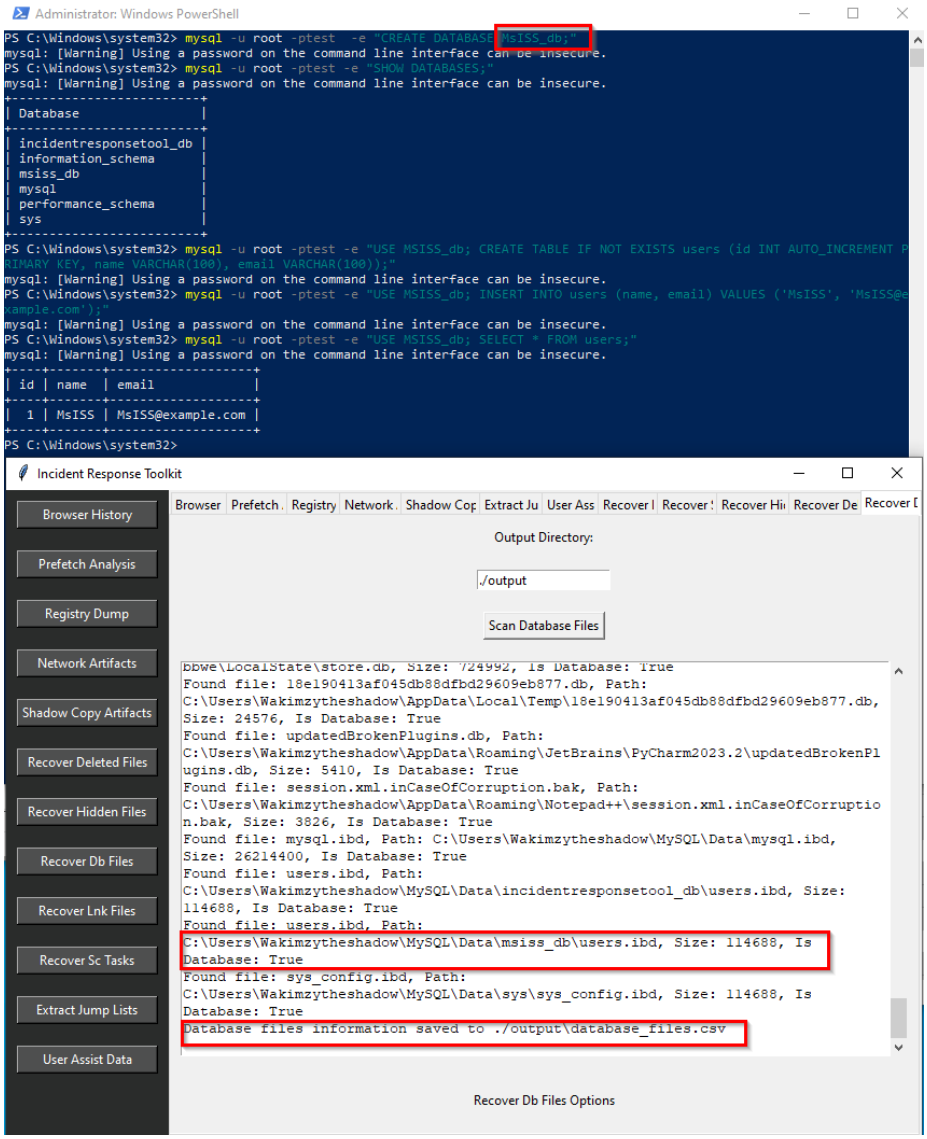


Figure 5.11: Database Artifact

5.3.3.8 Shortcut Files Extraction

The tool reads the binary data of a .lnk file, parses the header and various flags, and interprets additional data blocks to reconstruct the shortcut information. Methods like header, lnk flags, and file flags decode specific parts of the .lnk file structure which outputs the shortcut in human-readable or JSON formats as shown in Figure 5.12.

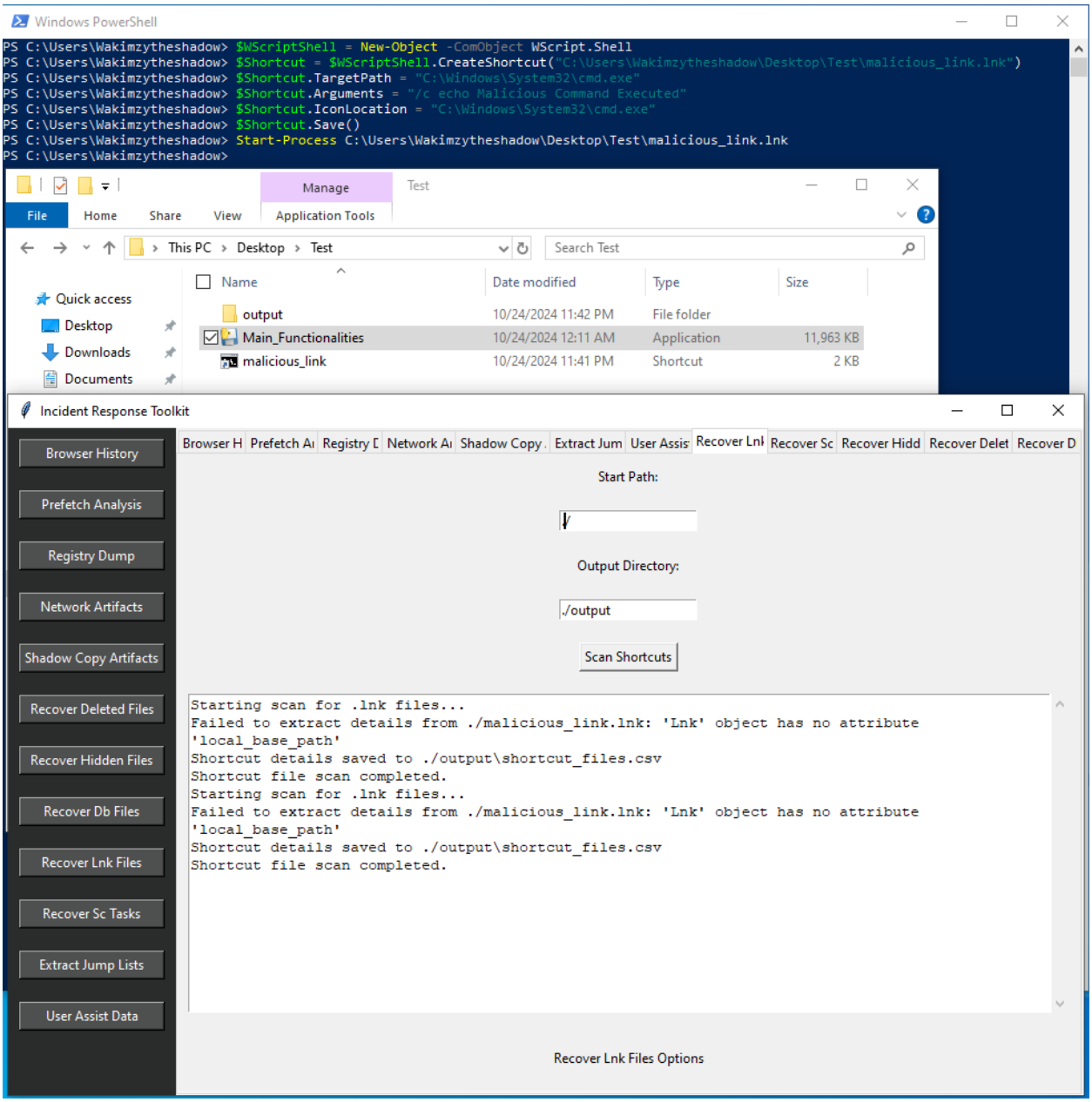


Figure 5.12: Recovered lnk file

5.3.3.9 Scheduled Task Artifact

Atomic test T1053.005-1 Scheduled Task Startup Script was used to simulate the creation of a scheduled task that executes on device startup to maintain persistence on a system. Administrative privileges were required for the task to be executed to inject a trigger on the system.

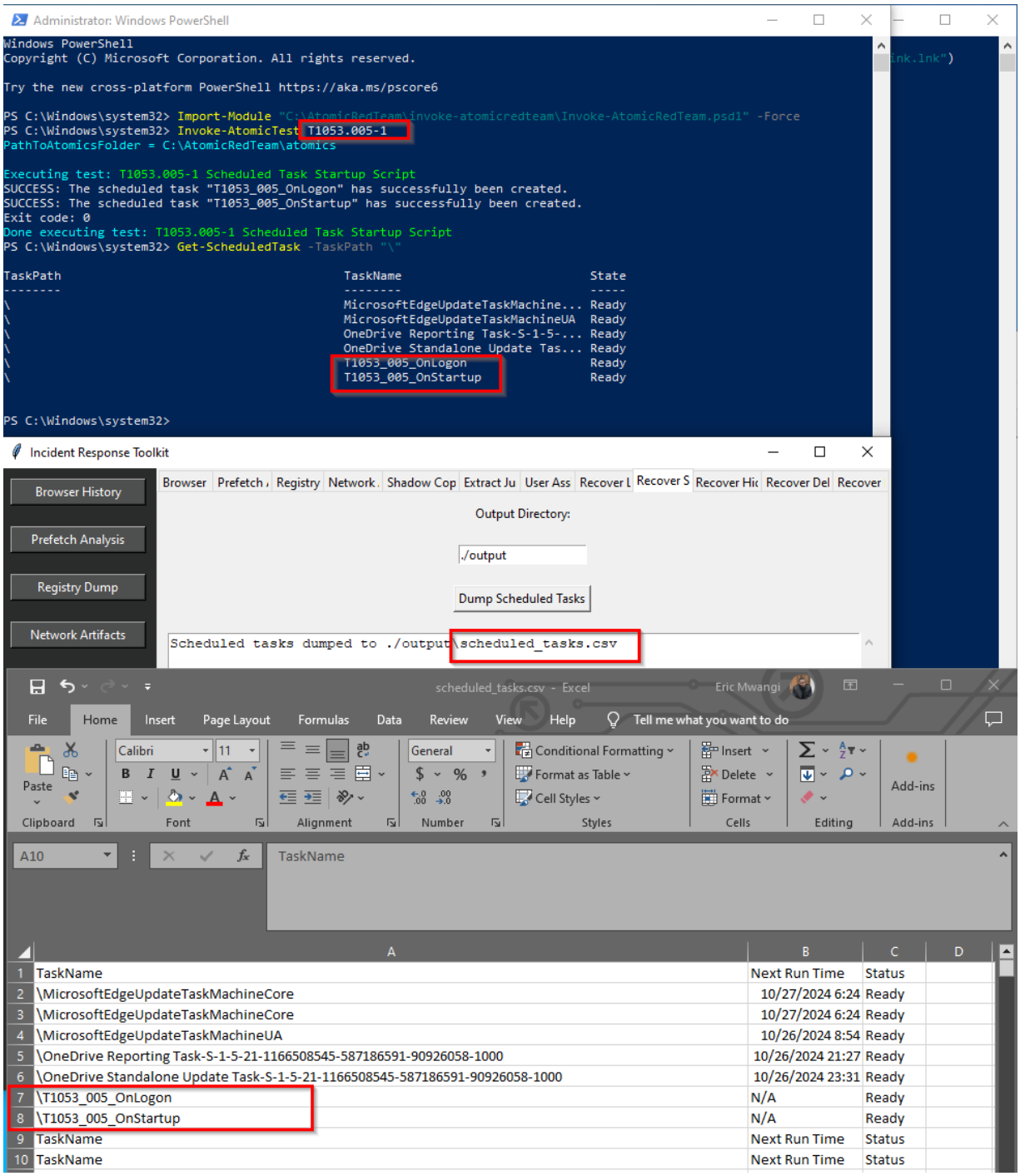


Figure 5.13: Recovered scheduled task

5.3.3.10 User Assist Data

Powershell was used in manipulating user assist data by injecting a new user assist key as captioned in Figure 5.14. This was achieved through identification and listing all the user assist GUIDS, and adding a custom key which signified a threat actors interaction with a malicious file or tool which was extracted by the incident response toolkit.

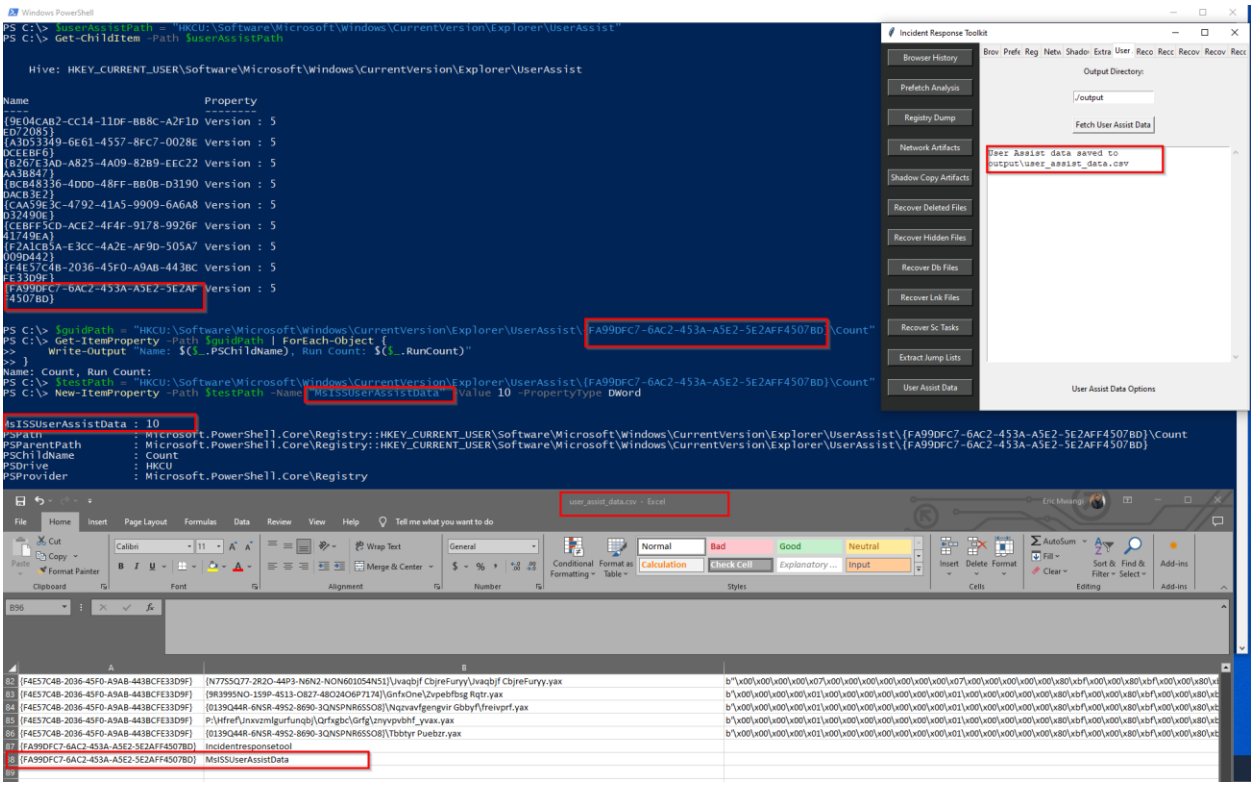


Figure 5.14: User assist data artifact

5.3.4 Recovery of Deleted Files

The objective of the functionality was to recover files deleted by threat actors in order to clear their tracks. The functionality is limited to .exe and .jpeg files which are used by attackers for concealing malware and steganography. File recovery is also limited to files smaller than 2MB to minimize the time and resources required to recover the files.

5.3.4.1 General Code Flow

i. Open Drive: The functions begin by opening a handle to the specified drive to access its information.

ii. Retrieve Disk Geometry: Uses device input output control to get the disk geometry, which is crucial for understanding how to read from the disk accurately.

iii. Read and Interpret VBR: Reads the first sector of the drive to interpret the Volume Boot Record and identify the file system or calculate the cluster size.

iv. Error Handling: Includes checks for errors at each step, printing error messages and ensuring handles and memory are released properly.

v. File Signature Structure: This structure defines the properties of a file signature, which includes a file extension, a pattern of bytes that identifies the file type, and the size of this pattern.

vi. Scan For Deleted Files: Opens the drive for reading, allocates a buffer for reading data from the drive-in chunks, iterates through the drive reading each cluster and checking for known file signatures and if a file signature is detected, it calls the recover file function to attempt recovery and manages file pointers and offsets to ensure the entire drive is scanned.

vii. Recover File: The tool creates a recovery directory if it does not exist, generates a unique name for the recovered file based on a static counter, opens a new file for writing the recovered data, reads data from the drive starting at the detected signature, writing it to the new file until the maximum file size is reached or no more data is available and handles potential errors in reading and writing, ensuring resources are freed properly.

5.3.4.2 Application Recovery

Figure 5.15 contains Penetration testing tools used by threat actors for Vulnerability assessment, password cracking, remote administration, lateral movement and network discovery toolsets located in drive S. The research aimed at recovering the files after permanent deletion from disk. The files were deleted from drive S and File recovery initiated from the tool as captioned on Figure 5.16 shows the disk which an incident responder can select to recover data. Figure 5.17 and 5.18 display the deleted files and a validation of the files recovered.

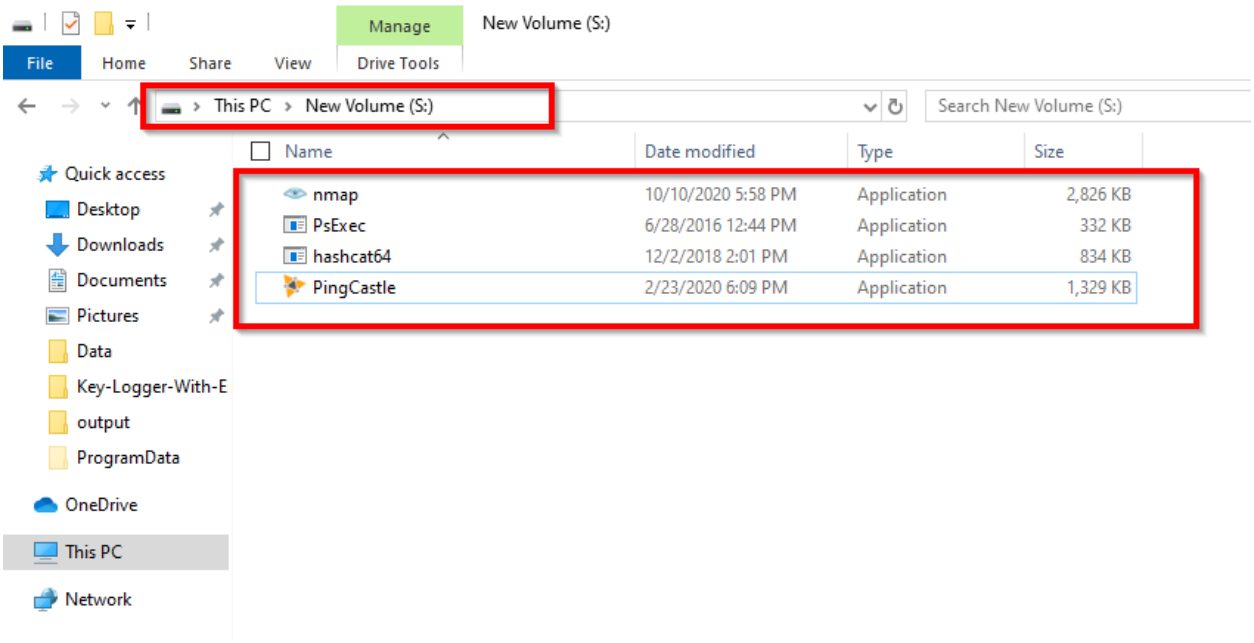


Figure 5.15: Assessment and penetration testing tools:

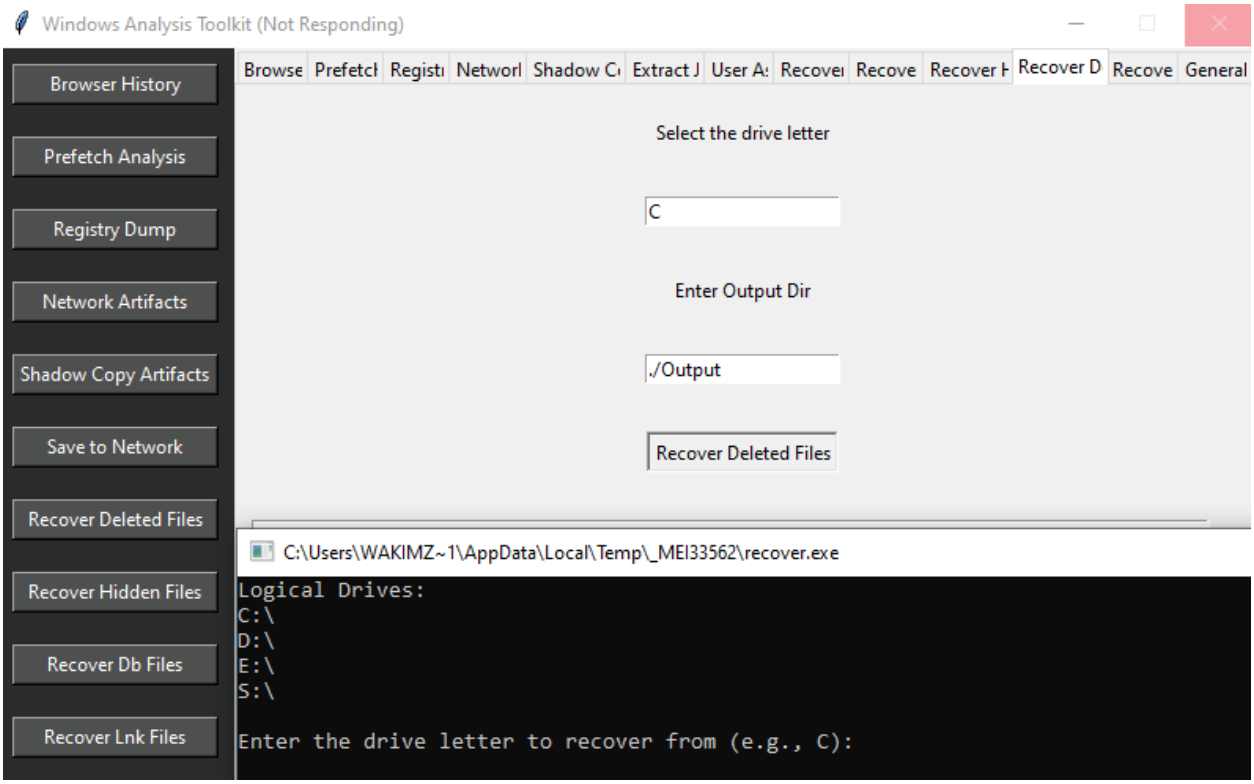


Figure 5.16: Selection of Recovery disk

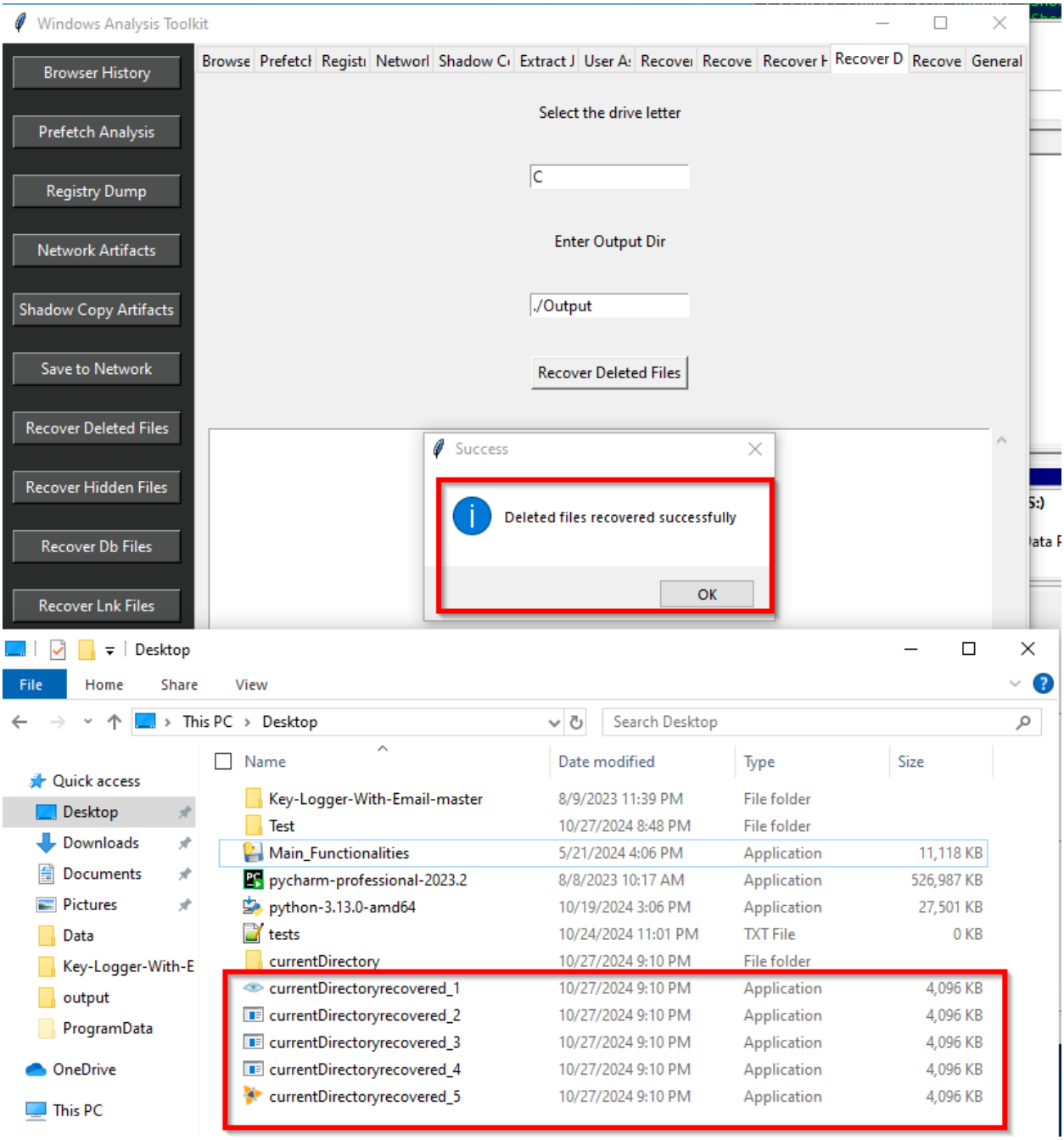


Figure 5.17: Recovered Files

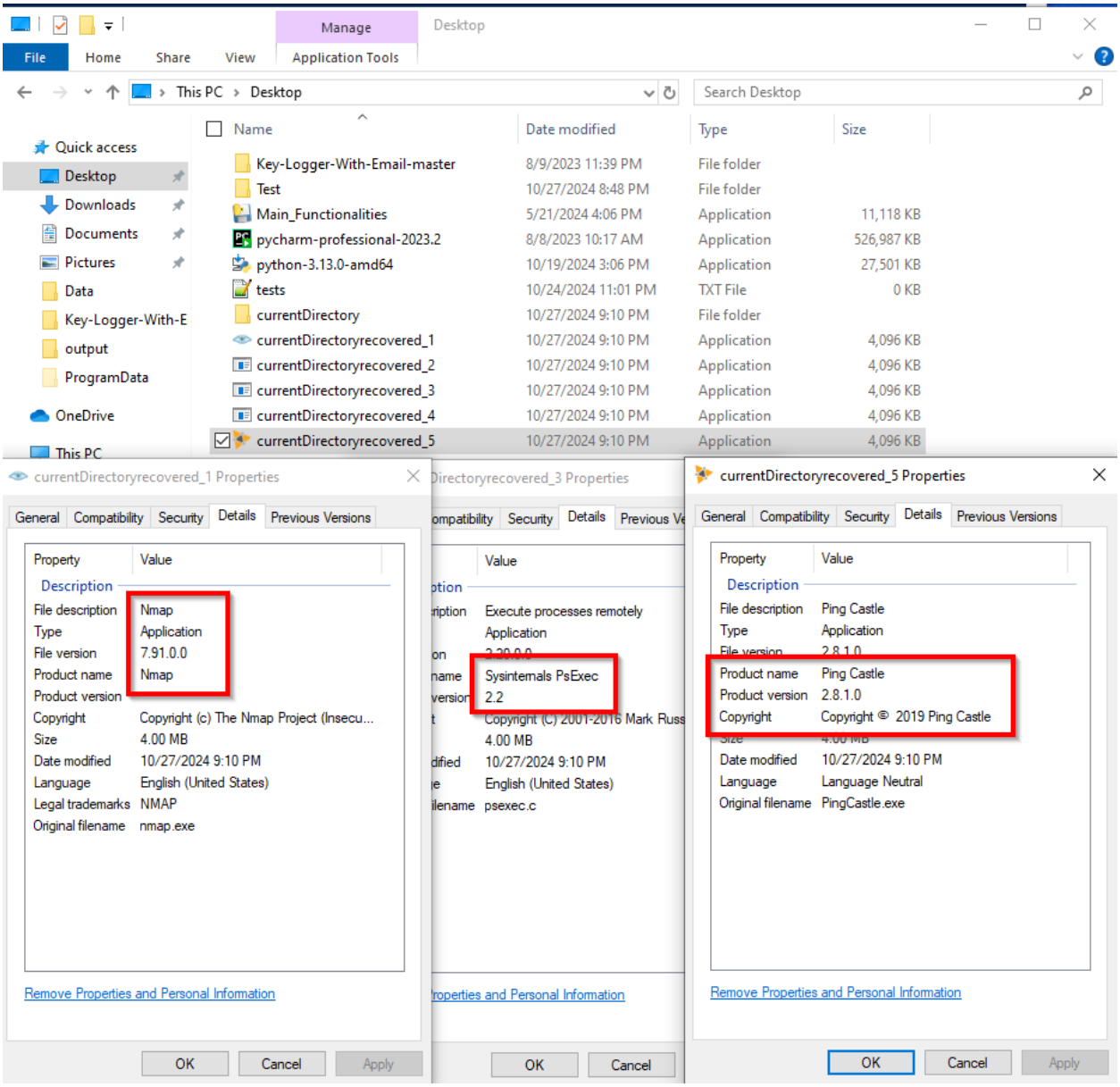


Figure 5.18: Recovered files validation

5.4 System Testing

This section describes the tests that were performed on the incident response tool, the researcher focused on user inputs on the GUI and acceptance testing. The tests were based on several metrics: functionality, user friendliness, performance and integration testing.

5.4.1 Functional Testing

Functional tests were performed to determine if the system designed and implementation were successful and the system worked as shown in Table 5.3.

Table 5.3: Results from Functional Testing

Interface Tested	Test Performed	Expectation	Observation	Result: Pass / Fail
Browser History	Incident responder should be able fetch to add their own valid browser path if the default path was not used.	The tool should fetch the browser history and provide an error message if the database browser is not found on the custom path.	The tool is able to fetch history from the default path. Error message is provided if history database is not found on the default or custom path.	Pass
Prefetch Analysis	Fetch prefetch files from the drive provided and extract to the output directory	The tool should fetch .pf files from the Drive provided.	The tool is able to fetch .pf files if the prefetch folder exists on the drive and provides an error message if it does not.	Pass
Registry Dump	Extract registry hives from the drive and save in the output directory.	The tool should extract all valid HKEY registry hives	The tool extracts all HKEY registry hives	Pass
Network Artifacts	Dump all network information.	The tool should dump all network configurations, active connections and running processes.	The tool is able to output the configurations, running process and active connections.	Pass
Shadow Copy Artifacts	Fetch shadow copy artifacts from drive	The tool should fetch shadow copies created on the drive.	The tool is able to output the shadow copy artifacts metadata.	Pass
Recover deleted files	Recover permanently deleted EXE or JPEG files	The tool should recover deleted files in scope less than 2MB in size.	The tool is able to recover and rebuild the deleted files.	Pass
Recover hidden files	Recover hidden folders and files.	The tool should recover all hidden files from the drive.	The tool is able to recover and output all hidden files from	

Recover databases	Recover all SQL and MySQL database.	The tool should recover databases and tables created in the drive.	The tool is able to recover created DBs and corresponding tables created on the DB.	Pass
Recover shortcut files	Recover LNK file from drive	The tool should check LNK files from the start path provided by the responder and output to the report.	The tool is able to scan for shortcut files.	Pass
Recover scheduled tasks	Recover scheduled tasks created.	The tool should be able to list all scheduled tasks to the output directory.	The tool is able to fetch and output the scheduled tasks on the drive.	Pass
Extract jump lists	Extract jump list from the main drive.	The tool should be able to extract all jump list from the main drive and output to the provided directory.	The tool is able to extract the jump lists and an output report.	Pass
Fetch User assist data	Fetch user assist data from the main drive	The tool should be able to extract user assist data from the main drive and output to the directory provided.	The tool is able to fetch the User assist data and an output report.	Pass

5.4.2 Integration Testing

Integration testing was performed to determine whether the developed tool will be able to function on the different personal computer and server platform. The tool is compatible with Windows 10 Education (Version 22H2) and Windows Server 2016 (Version 1607), ensuring seamless operation on both platforms as shown in T able 5.4.

Table 5.4: Results from Integration Testing

Windows Edition	Version	Compatible
Windows 10 Education	22H2	Yes
Windows Server 2016	Version 1607	Yes

5.4.3 User Friendliness

The tool was tested by incident responders and 90% of the incident responders indicated that the tool was easy to use, navigate and learn as shown in Figure 5.19.

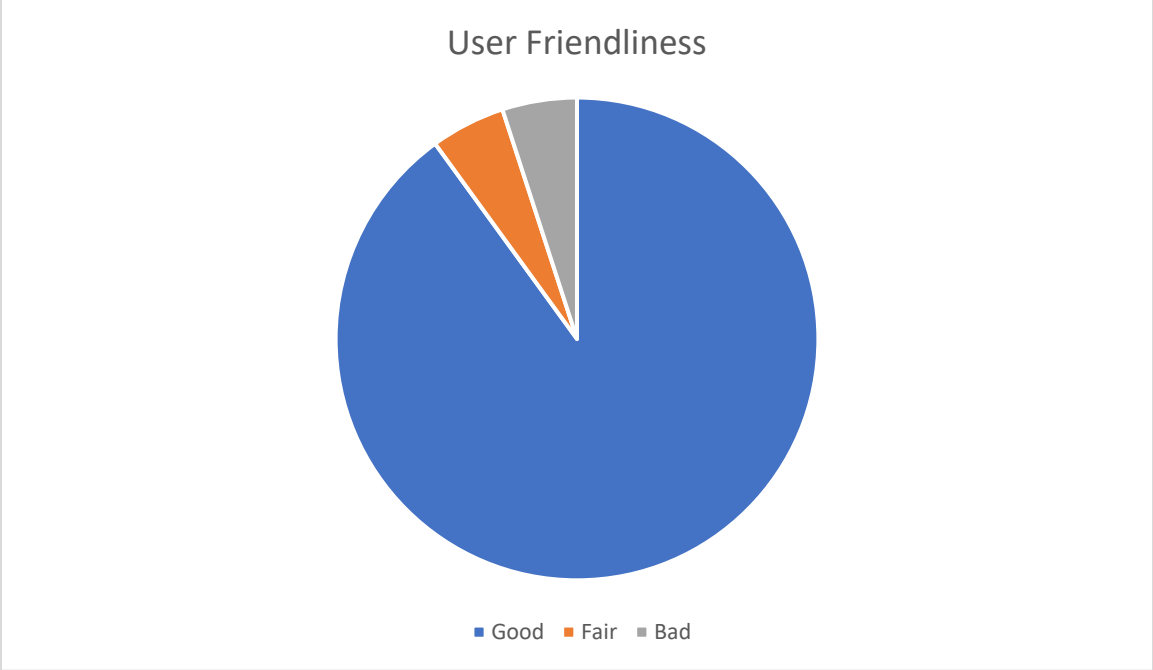


Figure 5.19:User friendliness Testing

5.4.4 Performance

The tool was tested by incident responders to see whether they could run different functionalities concurrently considering the device specifications. 95% 0f the incident responders confirmed they could run the functionalities simultaneously as shown in Figure 5.20. These suggests that the tool performs well in handling multiple functionalities at once, with only a small percentage potentially experiencing issues. This indicates high reliability and compatibility with the tested devices.

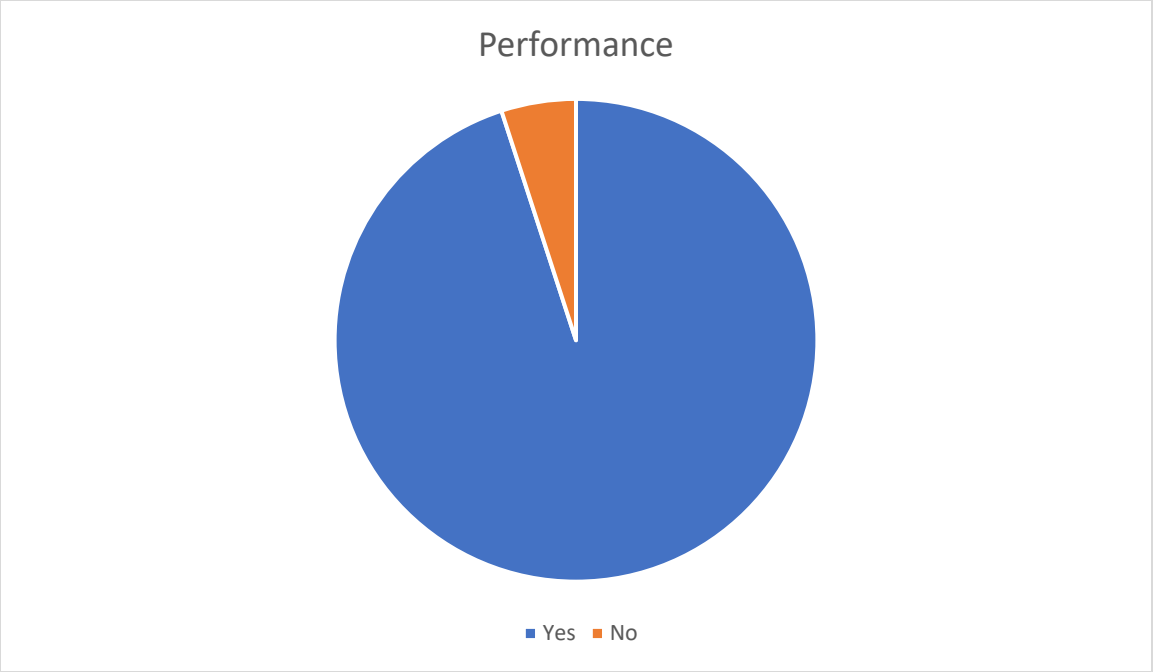


Figure 5.20: Performance testing

5.5 System Evaluation and Validation

This evaluation was carried out in order to ascertain that the research solution helped the incident responder by recovering artifacts and files from the windows device to identify crucial threat indicators traces on the device. Figure 5.21 shows the response from the incident responder regards to the implemented solution and if the tool solves the problem statement. The sample size was 5 incident responders. 80% (4 out of 5) of the incident responders were satisfied with the tool's ability to recover artifacts and files for identifying threat indicators, while 20% (1 out of 5) was not satisfied

The four satisfied incident responders found the tool effective in recovering crucial artifacts and files needed for identifying threat indicators on a Windows device. They appreciated its accuracy in retrieving relevant data, ease of use, and intuitive interface, which made the investigation process smoother. The tool’s speed and efficiency in processing data, along with its compatibility with device specifications, contributed to their positive experience. These responders also found the tool reliable, ensuring minimal disruptions during forensic analysis. The one unsatisfied responder encountered issues that impacted their experience. One concern was incomplete or missing artifact recovery, where the tool failed to extract all necessary data for a thorough investigation.

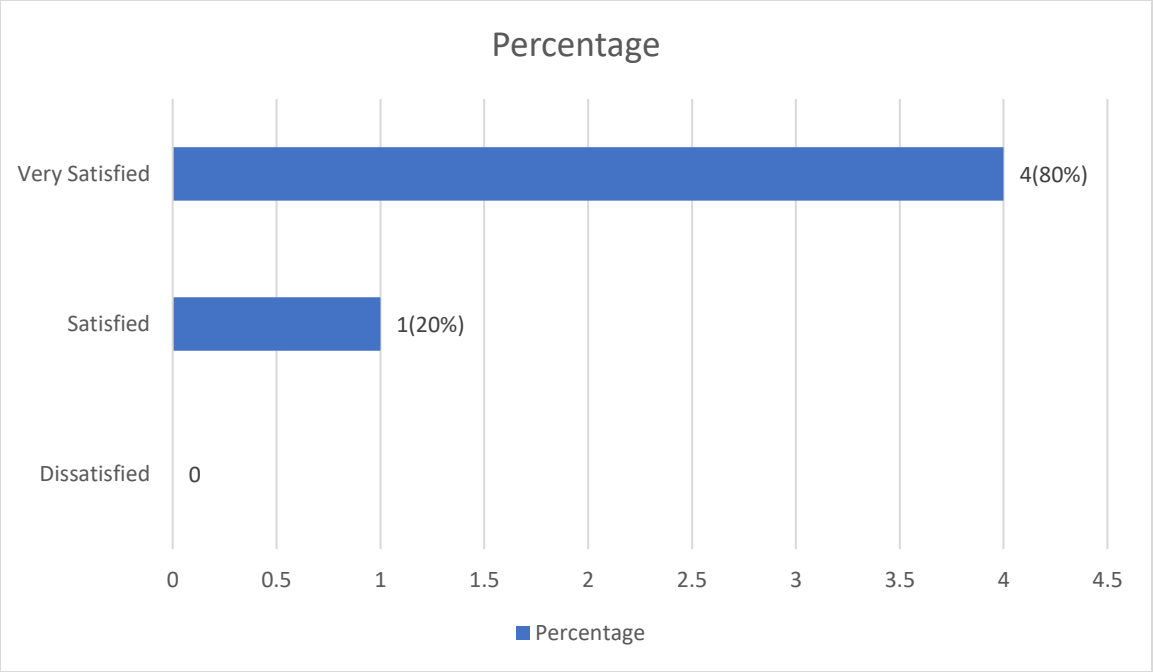


Figure 5.21: System Evaluation and Validation testing

Chapter 6 : Discussion

6.1 Introduction

This chapter presents the study's findings by analyzing the recorded information gathered during the research process. The discussion is structured around the study objectives, providing a clear assessment of how each objective was achieved. Additionally, it outlines how the research and development of the reporting system contributed to fulfilling these objectives, demonstrating the effectiveness of the study in meeting its goals.

6.2 Review of Current Technologies Used in Extracting File Artifacts

Through the review of literature conducted in Chapter 2, the study analysed forensic tools like FTK Imager, EnCase, Autopsy, and X1 Search which are widely used in digital investigations, but they present several challenges compared to the implemented tool. One common limitation is their resource-intensive nature, which can lead to slow processing times, making them less efficient for real-time incident response. Many of these tools focus heavily on disk imaging, indexing, or full-system analysis, which, while comprehensive, can result in unnecessary data extraction, requiring analysts to manually filter relevant forensic artifacts. Additionally, some tools lack specialized support for Windows-specific artifact recovery, such as detailed Prefetch analysis, scheduled tasks, or jump list data, limiting their effectiveness in targeted forensic investigations.

The designed tool overcomes these challenges by providing a streamlined and efficient approach to Windows artifact recovery, ensuring that only the most relevant forensic data is extracted. Unlike traditional forensic tools, which often require extensive manual effort to sift through large datasets, the designed tool focuses on recovering key system artifacts such as browser history, registry dumps, shadow copies, and event logs. This targeted approach enhances forensic efficiency, reducing the time required for investigation and improving incident response capabilities. By addressing the shortcomings of existing forensic tools, the designed tool offers a more practical and focused solution for security analysts handling Windows-based cyber incidents.

6.3 Review of Research Objectives

The dissertation identified the challenges faced by cybers security personnel in identifying and extracting Windows artifacts for cyber threat investigations. A tool was designed and developed after reviewing existing technologies in the literature review and system results from the system analysis. This study was conducted in alignment with the five research objectives presented in Chapter 1.

The first objective of the research was to analyze Windows file system artifacts for identifying indicators of compromise associated with malicious activities. The research achieved its objective by using PowerShell scripting and Atomic Red Team tests to simulate and analyze malicious activities. These tactics and techniques were used to generate controlled security incidents, allowing the researcher to observe system behaviour and identify IoCs. By executing predefined attack techniques, the study captured traces of unauthorized access, file execution patterns, and system modifications. Additionally, deleted executable files were recovered and analyzed to detect traces of malware or unauthorized applications. By leveraging both qualitative and quantitative analysis, the research successfully demonstrated how Windows file system artifacts enhance incident response capabilities, helping analysts detect, investigate, and mitigate cyber threats more effectively.

The second objective of the research was to review tools and techniques available in the Windows environment used in fetching Windows file system artifacts and reconstruction of deleted files. The literature revealed the need for a specialized tool that efficiently extracts Windows file system artifacts and recovers deleted files for cyber investigations. Existing tools often suffer from slow processing, excessive data extraction, and limited support for targeted Windows artifact recovery, making forensic analysis time-consuming and inefficient. A dedicated tool that focuses on streamlined artifact extraction and deleted file recovery would enhance cyber investigations by minimizing unnecessary data collection and improving recovery accuracy. This would significantly strengthen incident response and cyber threat analysis, providing a more effective solution for identifying IoCs.

The third and fourth objectives of the research were to design and develop a forensic analysis tool tailored for the Windows Operating system aimed at extracting file system artifacts and recovering

deleted indicators of compromise from Memory by threat actors and test the tools accuracy in recovery, reconstruction of deleted IoCs and generating an output report to assist in the Incident Response Process. The objective was achieved by designing, implementing, and testing the integrated system. The designed tool addresses the limitations of existing tools by focusing on efficient recovery of Windows file system artifacts and deleted files crucial for cyber investigations. It streamlines the extraction of key artifacts such as browser history, registry dumps, event logs, and Prefetch files, ensuring targeted recovery. The tool also overcomes the challenge of recovering deleted files within practical limits, providing timely and accurate data crucial for identifying IoCs. This focused approach enhances forensic efficiency and improves incident response, making it a valuable asset for cyber investigations.

6.4 Validation of the Forensic Tool for Windows Artifact Extraction in Incident Response

Based on the user feedback collected through structured questionnaires and practical testing, the study validated the functionality and reliability of the proposed forensic analysis tool. The validation process focused on key aspects relevant to security incident response, including accuracy of artifact extraction, reliability of deleted file recovery, and usability of the tool interface.

To begin with, the tool's effectiveness in extracting key Windows file system artifacts was confirmed through multiple tests run across various logical drives. The artifacts retrieved included registry hives, event logs, prefetch files, and lnk files commonly used in forensic investigations. Users reported that the tool consistently identified and parsed these artifacts with minimal false positives. Additionally, the recovery module was able to restore deleted .exe files from selected drives, particularly those suspected to have been removed during malicious activity. The tool's file recovery feature operated by scanning unallocated space and matching known executable file signatures, thereby validating its utility in post-compromise analysis.

The usability and interface design were also evaluated. The tool employed a tab-based layout, allowing users to switch seamlessly between different artifact categories and perform concurrent extractions. This was validated through user testing where participants successfully retrieved multiple types of evidence without needing advanced technical knowledge. Feedback indicated that the interface was intuitive and aligned well with the workflow of digital investigators.

Validation further confirmed that the tool preserved the integrity of extracted artifacts by performing read-only operations on the file system and maintaining metadata where possible. All outputs were logged and timestamped, supporting accountability and traceability—core requirements in forensic investigations. In conclusion, the tool demonstrated strong alignment with incident response needs and proved to be a dependable asset for evidence collection and preliminary analysis in Windows environments.

Chapter 7 : Conclusions and Recommendations

7.1 Conclusions

The study proposed and a forensic tool to enhance the recovery and analysis of Windows file system artifacts during incident response. The findings revealed that while Windows artifacts contain valuable forensic data, their full potential in identifying indicators of compromise is often underutilized. By leveraging the tool, security professionals can extract critical information, including deleted executable files, to aid in cybersecurity investigations. Ensuring the integrity and availability of these artifacts is essential for effective threat detection and response within a security function.

7.2 Recommendations

To further enhance the forensic tool's capabilities in incident response, several improvements can be recommended. One key enhancement would be expanding the tool's file recovery capabilities beyond the current 2MB threshold, allowing for the retrieval of larger deleted files that may contain crucial forensic evidence. Additionally, incorporating advanced metadata analysis would help provide deeper insights into recovered artifacts, such as file creation, modification, and access details, aiding investigators in reconstructing attack timelines. The integration of YARA rule scanning for recovered files and artifacts would also improve the detection of known malware signatures and suspicious activity, enhancing the tool's threat identification capabilities.

Another valuable addition would be automated correlation and reporting features, enabling investigators to efficiently analyze recovered artifacts in relation to known IoCs. Implementing registry anomaly detection could highlight suspicious changes in registry entries, providing further evidence of potential attacks. Moreover, enhancing shadow copy analysis to detect tampered or hidden system states could improve the tool's ability to uncover sophisticated threats. Lastly, adding customizable artifact extraction filters would allow forensic analysts to focus on specific data relevant to their investigations, improving efficiency and usability in diverse forensic scenarios.

7.3 Future Works

Future enhancements to the forensic tool can leverage AI and ML to improve automation and threat detection in incident response. One key advancement would be AI-driven anomaly detection, where machine learning models analyze recovered Windows artifacts, such as registry entries, event logs, and scheduled tasks, to identify suspicious activity linked to cyber threats. By training AI models on historical attack data, the tool could automatically highlight unusual system modifications, minimizing manual analysis. Additionally, Natural Language Processing (NLP) could be applied to interpret event logs and registry descriptions, aiding in the classification and prioritization of potential security incidents.

Another future direction involves AI-assisted file recovery, where deep learning models could reconstruct fragmented or partially deleted files, surpassing the current 2MB recovery limit. Incorporating automated correlation engines powered by AI would improve the linkage between recovered artifacts, allowing for better attack timeline reconstruction. Furthermore, AI-based behavioural analysis could enhance the tool's ability to detect emerging threats by continuously learning from new attack patterns. These AI-driven enhancements would significantly improve accuracy, efficiency, and the overall effectiveness of the forensic tool in responding to sophisticated cyber threats.

References

Alsmadi, I., & Alazab, M. (2017). A Model Based Approach for the Extraction of Network Forensic Artifacts. *2017 Cybersecurity and Cyberforensics Conference (CCC)*, 16–18. https://doi.org/10.1109/CCC.2017.13

Angamutu, K. A., Rahman, N. A. A., & Suki, N. N. A. N. (2020). A Customized Data Recovery Tool. *Journal of Physics: Conference Series*, *1712*(1), 012019. https://doi.org/10.1088/1742-6596/1712/1/012019

Angamutu, K. A.-P., & Selvarajah, V. A.-P. (2023). An Insight into the Data Recovery of Deleted or Heavily Damaged Storage Media Through the Lens of R-Studio. *2023 International Conference on Integrated Intelligence and Communication Systems (ICIICS)*, 1–5. https://doi.org/10.1109/ICIICS59993.2023.10421397

Barakat, A., & Hadi, A. (2016). Windows Forensic Investigations Using PowerForensics Tool. *2016 Cybersecurity and Cyberforensics Conference (CCC)*, 41–47. https://doi.org/10.1109/CCC.2016.18

Budhrani, A., Singh, U., & Singh, B. (2022a). Analysis of Windows 11 Link File Artifact for Evidence Gathering. *2022 International Conference on Futuristic Technologies (INCOFT)*, 1–6. https://doi.org/10.1109/INCOFT55651.2022.10094555

Budhrani, A., Singh, U., & Singh, B. (2022b). Forensic Analysis of Windows 11 Prefetch Artifact. *2022 IEEE Bombay Section Signature Conference (IBSSC)*, 1–6. https://doi.org/10.1109/IBSSC56953.2022.10037260

Carvajal, L., Varol, C., & Lei Chen. (2013). Tools for collecting volatile data: A survey study. *2013 The International Conference on Technological Advances in Electrical, Electronics and Computer Engineering (TAEECE)*, 318–322. https://doi.org/10.1109/TAEECE.2013.6557293

Cichonski, P., Millar, T., Grance, T., & Scarfone, K. (2012). *Computer Security Incident Handling Guide : Recommendations of the National Institute of Standards and Technology*. https://doi.org/10.6028/NIST.SP.800-61r2

Duranec, A., Topolcic, D., Hausknecht, K., & Delija, D. (2019). Investigating file use and knowledge with Windows 10 artifacts. *2019 42nd International Convention on Information and Communication Technology, Electronics and Microelectronics (MIPRO)*, 1213–1218. https://doi.org/10.23919/MIPRO.2019.8756877

Etikan, I. (2016). Comparison of Convenience Sampling and Purposive Sampling. *American Journal of Theoretical and Applied Statistics*, *5*(1), 1–4. https://doi.org/10.11648/j.ajtas.20160501.11

Farnan, M., Pratt, J., David, A., & Shakiba, M. (2024). Digital Forensic Tools: Comparison of Autopsy TSK and Forensic Explorer. *2024 IEEE International Conference on Information Technology, Electronics and Intelligent Communication Systems (ICITEICS)*, 1–5. https://doi.org/10.1109/ICITEICS61368.2024.10625076

Fauzi, M. A., Tribiakto, H., Moniva, A., Amir, F., Ilyas, I. K., & Utami, E. (2023). Systematic Literature Reviews on Rapid Application Development Information System. *Bulletin of Computer Science and Electrical Engineering*, *4*(1), 57–64. https://doi.org/10.25008/bcsee.v4i1.1181

Fayyazi, R., Taghdimi, R., & Yang, S. J. (2024). Advancing TTP Analysis: Harnessing the Power of Large Language Models with Retrieval Augmented Generation. *2024 Annual Computer Security Applications Conference Workshops (ACSAC Workshops)*, 255–261. https://doi.org/10.1109/ACSACW65225.2024.00036

Gorecki, A. (2020). Crafting an Incident Response Plan. In *Cyber Breach Response That Actually Works* (pp. 143–194). Wiley. https://doi.org/10.1002/9781119679349.ch4

Haber, M. J., & Rolls, D. (2024). Indicators of Compromise. In *Identity Attack Vectors* (pp. 87–107). Apress. https://doi.org/10.1007/979-8-8688-0233-1_10

Haider, R. Z., Aslam, B., Abbas, H., & Iqbal, Z. (2024). *C2-Eye:Framework for Detecting Command and Control (C2) Connection of Supply Chain Attacks*. https://doi.org/10.21203/rs.3.rs-3867295/v1

Himanshu, Bhatt, S., & Garg, G. (2021). Comparative analysis of acquisition methods in digital forensics. *2021 Fourth International Conference on Computational Intelligence and Communication Technologies (CCICT)*, 129–134. https://doi.org/10.1109/CCICT53244.2021.00035

Ishak, I., & Alias, R. (2005). DESIGNING A STRATEGIC INFORMATION SYSTEMS PLANNING METHODOLOGY FOR MALAYSIAN INSTITUTES OF HIGHER LEARNING (ISP-IPTA). *Issues In Information Systems*. https://doi.org/10.48009/1_iis_2005_325-331

Javed, M. S., Sajjad, S. M., Mehmood, D., Mansoor, K., Iqbal, Z., Kazim, M., & Muhammad, Z. (2024). Analyzing Tor Browser Artifacts for Enhanced Web Forensics, Anonymity, Cybersecurity, and Privacy in Windows-Based Systems. *Information*, *15*(8), 495. https://doi.org/10.3390/info15080495

Jerrim, J., & Vries, R. (2023). Are peer reviews of grant proposals reliable? An analysis of Economic and Social Research Council (ESRC) funding applications. *The Social Science Journal*, *60*(1), 91–109. https://doi.org/10.1080/03623319.2020.1728506

Kävrestad, J., Birath, M., & Clarke, N. (2024). *Finding Artifacts* (pp. 135–161). https://doi.org/10.1007/978-3-031-53649-6_14

Kim, J., Son, B., Yu, J., & Yun, J. (2024). AI-Driven Prioritization and Filtering of Windows Artifacts for Enhanced Digital Forensics. *Computers, Materials & Continua*, *81*(2), 3371–3393. https://doi.org/10.32604/cmc.2024.057234

Kondapally, B. P. (2015). *Forensically Important Artifacts in Windows Operating systems*.

Lee, S., Lee, S., Park, J., Kim, K., & Lee, K. (2023). Hiding in the Crowd: Ransomware Protection by Adopting Camouflage and Hiding Strategy With the Link File. *IEEE Access*, *11*, 92693–92704. https://doi.org/10.1109/ACCESS.2023.3309879

Liew, S. P., & Ikeda, S. (2019). Detecting Adversary using Windows Digital Artifacts. *2019 IEEE International Conference on Big Data (Big Data)*, 3210–3215. https://doi.org/10.1109/BigData47090.2019.9006552

maheswari, K. U., & Shobana, G. (2021). The State of the art tools and techniques for remote digital forensic investigations. *2021 3rd International Conference on Signal Processing and Communication (ICPSC)*, 464–468. https://doi.org/10.1109/ICSPC51351.2021.9451718

McCluskey, Q. R., Chowdhury, M. M., Latif, S., & Kambhampaty, K. (2022). Computer Forensics: Complementing Cyber Security. *2022 IEEE International Conference on Electro Information Technology (EIT)*, 507–512. https://doi.org/10.1109/eIT53891.2022.9813886

Meisheng, Y., & Huang, W. (2008). The Quickly Solving Method of File Recovery in Windows Environment. *2008 International Conference on Computer Science and Software Engineering*, 859–862. https://doi.org/10.1109/CSSE.2008.1009

Microsoft. (2024). *Windows File Recovery*.

Oh, J., Lee, S., & Hwang, H. (2022). Forensic Recovery of File System Metadata for Digital Forensic Investigation. *IEEE Access*, *10*, 111591–111606. https://doi.org/10.1109/ACCESS.2022.3213030

Red Canary. (2024). *Atomic Red Team*. GitHub, Inc.

S, D., J, A., V, I., & M, S. (2020). Cyber Forensics: Discovering Traces of Malware on Windows Systems. *2020 IEEE Recent Advances in Intelligent Computational Systems (RAICS)*, 141–146. https://doi.org/10.1109/RAICS51191.2020.9332496

Salman, M., bani-slman, B., Aljaidi, M., Saleem, R. bani, Alsarhan, A., Qasem, M. H., Injadat, M. N., & Igried, B. (2023). A Study of Forensic Tools Data Recovery Performance. *2023 2nd International Engineering Conference on Electrical, Energy, and Artificial Intelligence (EICEEAI)*, 1–6. https://doi.org/10.1109/EICEEAI60672.2023.10590383

Smith, C., Dietrich, G., & Choo, K.-K. R. (2018). *Identification of Forensic Artifacts in VMWare Virtualized Computing* (pp. 85–103). https://doi.org/10.1007/978-3-319-78816-6_7

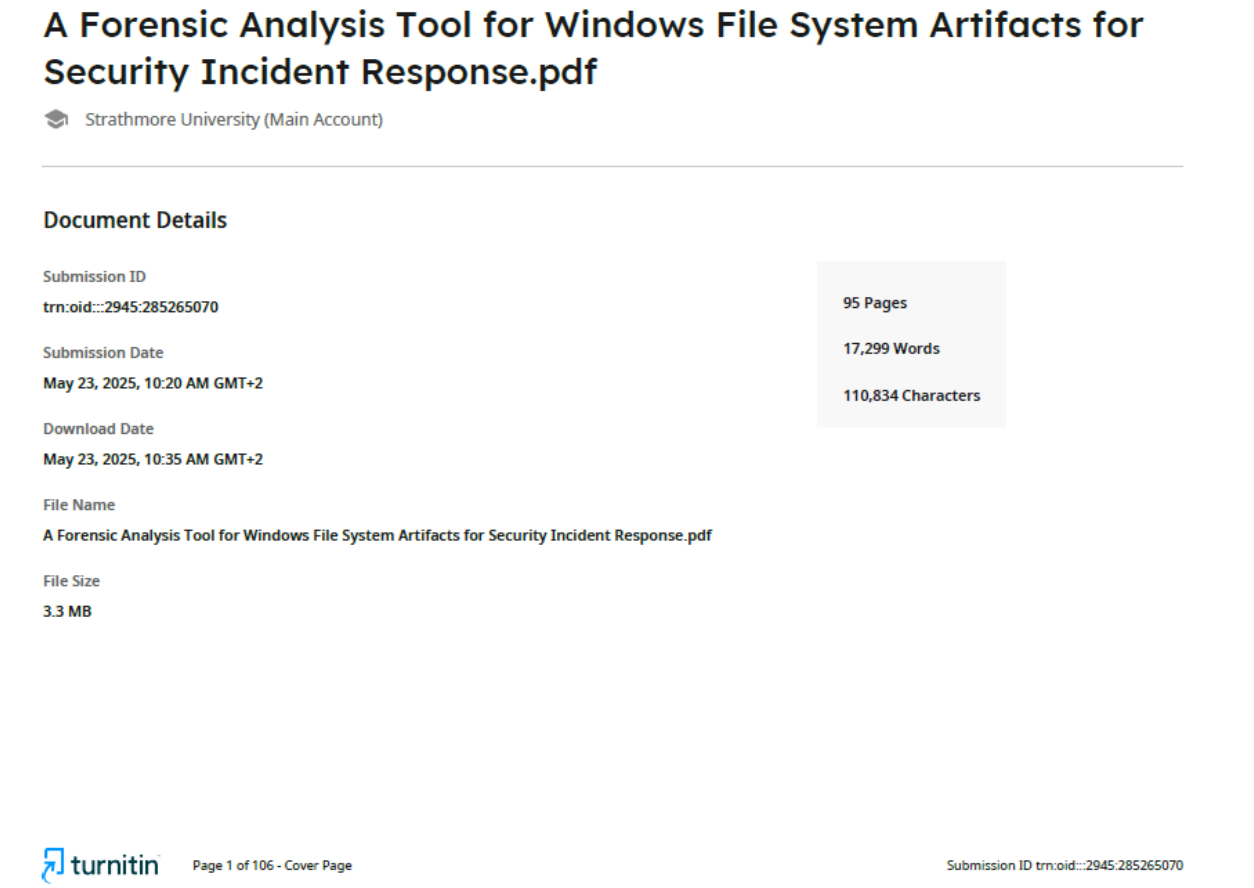
Sreeja, S. C., & Balan, C. (2016). Forensic analysis of volume shadow copy in Windows 7. *2016 International Conference on Emerging Technological Trends (ICETT)*, 1–6. https://doi.org/10.1109/ICETT.2016.7873670

Statcounter. (2024). *Desktop Operating System Market Share Worldwide*.

Zhang, N., Jiang, Y., & Wang, J. (2020). The Research of Data Recovery on Windows File Systems. *2020 International Conference on Intelligent Transportation, Big Data & Smart City (ICITBS)*, 644–647. https://doi.org/10.1109/ICITBS49701.2020.00141

Appendices

Appendix A: Similarity Report



A Forensic Analysis Tool for Windows File System Artifacts for Security Incident Response.pdf

Strathmore University (Main Account)

Document Details

Submission ID
trn:oid:::2945:285265070

Submission Date
May 23, 2025, 10:20 AM GMT+2

Download Date
May 23, 2025, 10:35 AM GMT+2

File Name
A Forensic Analysis Tool for Windows File System Artifacts for Security Incident Response.pdf

File Size
3.3 MB

95 Pages

17,299 Words

110,834 Characters

turnitin Page 1 of 106 - Cover Page Submission ID trn:oid:::2945:285265070

turnitin Page 2 of 106 - Integrity Overview Submission ID trn:oid:::2945:285265070

16% Overall Similarity

The combined total of all matches, including overlapping sources, for each database.

Appendix B: Ethical Clearance Confirmation

Strathmore UNIVERSITY

10th April 2025

Mr Mwangi Eric,
ericmwangi.kimani@strathmore.edu

Dear Mr Mwangi,

RE: A Forensic Analysis Tool for Windows File System Artifacts: Case of Security Incidents Response

This is to inform you that SU-ISERC has reviewed and **approved** your above **SU-masters** proposal. Your application reference number is **SU-ISERC2665/25.** The approval period is from **10th April 2025 to 9th April 2026.**

This approval is subject to compliance with the following requirements:

i. Only approved documents including (informed consents, study instruments, MTA) will be used.
ii. All changes including (amendments, deviations, and violations) are submitted for review and approval by SU-ISERC.
iii. Death and life-threatening problems and serious adverse events or unexpected adverse events whether related or unrelated to the study must be reported to SU-ISERC within 72 hours of notification.
iv. Any changes anticipated or otherwise that may increase the risks or affected safety or welfare of study participants and others or affect the integrity of the research must be reported to SU-ISERC within 72 hours.
v. Clearance for the export of biological specimens must be obtained from relevant institutions.
vi. Submission of a request for renewal of approval at least 60 days prior to the expiry of the approval period. Attach a comprehensive progress report to support the renewal.
vii. Submission of an executive summary report within 90 days of completion of the study to SU-ISERC.

Before commencing your study, you will be expected to obtain a research license from National Commission for Science, Technology, and Innovation (NACOSTI) https://research-portal.nacosti.go.ke/ and obtain other clearances needed.

Yours sincerely,

Mr Ambrose Rachier,
Chairperson; SU-ISERC

Ole Sangale Rd, Madaraka Estate. PO Box 59857-00200, Nairobi, Kenya. Tel +254 (0)703 034000
Email admissions@strathmore.edu www.strathmore.edu

Appendix C: Survey Questionnaire

Title of Study:
Forensic Analysis of Windows File System Artifacts for Incident Response of Indicators of Compromise for Security Operation Centre

Researcher Details:
Researcher: Eric Mwangi
Institution: Strathmore University
Supervisor: Dr. Victor Rop
Contact Information: ericmwangi.kimani@strathmore.edu

Section 1: Participant Background

1. What is your current role or profession?
   - ☐ Security Analyst
   - ☐ Digital Forensics Investigator
   - ☐ SOC Analyst
   - ☐ IT Administrator
   - ☐ Other (please specify): _____
2. How many years of experience do you have in cybersecurity or digital forensics?
   - ☐ 0-2 years
   - ☐ 3-5 years
   - ☐ 6-10 years
   - ☐ More than 10 years
3. Have you previously conducted forensic analysis on Windows file system artifacts?
   - ☐ Yes
   - ☐ No
4. If yes, what tools have you used for forensic analysis? (Select all that apply)
   - ☐ Autopsy

- ☐ EnCase
- ☐ FTK (Forensic Toolkit)
- ☐ Volatility
- ☐ Other (please specify): _____

Section 2: Windows File System Artifacts and Incident Response

5. How familiar are you with Windows file system artifacts (e.g., MFT, Prefetch, Registry, Event Logs)?
   - ☐ Not familiar
   - ☐ Somewhat familiar
   - ☐ Familiar
   - ☐ Very familiar
6. Which Windows file system artifacts do you find most useful in forensic investigations? (Select all that apply)
   - ☐ Master File Table (MFT)
   - ☐ Prefetch Files
   - ☐ Windows Registry
   - ☐ Event Logs
   - ☐ Other (please specify): _____
7. How frequently do you analyze Windows file system artifacts in your work?
   - ☐ Rarely
   - ☐ Occasionally
   - ☐ Frequently
   - ☐ Always

8. In your experience, what are the most common indicators of compromise (IoCs) found in Windows file system artifacts? (Select all that apply)

   ☐ Unauthorized file modifications

   ☐ Suspicious Registry changes

   ☐ Presence of malware-related files

   ☐ Abnormal Event Log entries

   ☐ Other (please specify): _____

9. What challenges do you face in analyzing Windows file system artifacts for incident response?

Section 3: Security Operations Centre (SOC) and Incident Response

10. Does your organization have a dedicated SOC team for handling security incidents?

☐ Yes

☐ No

11. How effective do you think Windows file system artifacts are in detecting security incidents?

☐ Not effective

☐ Somewhat effective

☐ Effective

☐ Very effective

12. What additional forensic techniques or tools do you use to support incident response?

13. What improvements would you suggest for using Windows file system artifacts in forensic investigations?

Section 4: Additional Comments

14. Do you have any additional insights or suggestions related to forensic analysis of Windows file system artifacts?

Participant's Name: _____

Date: _____

Signature: _____

Thank you for your participation!

Appendix D: Participant Information Sheet and Consent Form

Title of Study:

Forensic Analysis of Windows File System Artifacts for Incident Response of Indicators of Compromise for Security Operation Centre

Researcher Details:

Researcher: Eric Mwangi

Institution: Strathmore University

Supervisor: Dr. Victor Rop

Contact Information: ericmwangi.kimani@strathmore.edu

Participant Information Sheet

1. Introduction

You are invited to take part in a research study conducted as part of my dissertation. This document provides information about the study, its objectives, and what your participation will involve.

2. Purpose of the Study

This research aims to analyze Windows file system artifacts to identify indicators of compromise (IoCs) that can support security teams in incident response. The findings will help improve forensic investigation methodologies and enhance security measures.

3. What Participation Involves

If you agree to participate, you will be asked to:

i. Review system logs, file metadata, or other relevant forensic data.

ii. Engage in interviews or surveys regarding forensic investigation techniques.

iii. Participate in controlled experiments related to file system artifacts.

Your participation is entirely voluntary, and you may withdraw at any time without any consequences.

4. Expected Duration of Participation

The expected duration of your participation in this study will be approximately 1 hour session online over 8 weeks depending on the level of involvement you choose.

5. Potential Risks and Benefits

Risks: Minimal risk is anticipated; however, handling forensic data may include exposure to sensitive system information. Appropriate measures will be taken to anonymize data and maintain confidentiality.

Benefits:

i. Participants will gain insights into forensic analysis methodologies.

ii. The research findings will contribute to improved security strategies.

iii. It will enhance knowledge in incident response and cybersecurity measures.

6. Confidentiality and Data Protection

All collected data will be anonymized and stored securely. No personally identifiable information will be published. Data will be used solely for academic research purposes in compliance with Strathmore University and Kenya Data Protection regulation.

7. Voluntary Participation and Right to Withdraw

Participation is voluntary. You may withdraw at any point without giving a reason. If you withdraw, your data will be deleted unless you consent to its continued use in anonymized form.

8. Contact Information

If you have any questions or concerns, please contact me at ericmwangi.kimani@strathmore.edu or my supervisor Dr. Victor Rop at vrop@strathmore.edu.

9. Ethics Committee Contact Details

If you have any concerns about the ethical aspects of this research, you may contact the Review Ethics Committee at:

Review Ethics Committee Contact Information:

Secretary–Strathmore University Institutional Ethics Review Board

ethicsreview@strathmore.edu

+254(0)730 734 418

P. O. BOX 59857, 00200, Nairobi

Consent Form

Title of Study:
Forensic Analysis of Windows File System Artifacts for Incident Response of Indicators of Compromise for Security Operation Centre

Researcher Name: Eric Mwangi

Please read the statements below and indicate your agreement by signing at the bottom.

i. I confirm that I have read and understood the Participant Information Sheet for this study and have had the opportunity to ask questions.

ii. I understand that my participation is voluntary and that I can withdraw at any time without any consequences.

iii. I consent to the collection and use of anonymized data for this research.

iv. I understand that my personal data will remain confidential and will not be disclosed to unauthorized parties.

v. I agree to participate in this study.

Participant's Name: _____

Participant's Signature: _____

Date: _____

Researcher's Name: _____

Researcher's Signature: _____

Date: _____