كلية الإبتكار التقني

COLLEGE OF TECHNOLOGICAL INNOVATION

Digital Forensics Project

Forensic Insights Hub

Shamma Alhameli

Noura Alremeithi

Zayed University

Dr. Richard Ikuesean

Abstract

This project is a forensic tool designed using the Python language with the GUI library tkinter and direct access to the Windows operating system registry. This tool is aimed at satisfying basic forensic needs – it can reveal the software installed on the victim's computer, detect the details of the hardware components, identify the time of the latest modification of registry keys, discover the user login information, and export the contents of the registry in a fashion that forensic reports require*(Mitnick & Simon, 2011)*. These attributes are designed to help forensic investigators collect needed artifacts during computer examination with or without the need for professional assistance, that is, in self-service mode. It allows the methodology to expand as required and, as such, is one of the most commendable attributes in the realm of forensics analysis and incident response. Through practical usage with the registry, the study elucidates the usefulness and applicability of Python in forensic science applications (Carvey, 2011).

Introduction

The Windows Registry is a tree-structured database repository of system configurations and their settings and options. After all, its proper functioning within a particular system makes it a vital evidence source in digital forensic investigations. Whether it will be discovered that they have committed an offense or that an evil person has enabled their computer and is dirty with malware. Thus, the analysis of the Windows registry becomes the basis for implementing today's forensic efforts (Casey, 2011).

Strangely enough, the process of analytical registry is tedious and subject to many errors and, in many cases, requires specialized knowledge and some software tools. To look for practical solutions, the project introduces a Python-based automated forensic investigation tool, which allows for interacting with the Windows Registry with ease precisely for this purpose by way of Winreg standard use. In particular, the customization provides built-in tools that enable data extraction and registry analysis(Shinder, 2016).

Some of these functions are directed to:

- Software Enumeration: Allows users to check software, installed offices, and similar things to identify potentially vulnerable software or user applications.
- User Account Enumeration: Information such as user profile details can be used to help with action tracing, and there is the possibility of using the conducted research regarding the data in the relevant accounts.
- Hardware Information Retrieval: One that verifies such aspects as the CPU or GPU of the system in question to ascertain the possibility of its threat during the investigations.
- Registry Data Export: The data in the form of prepared registry keys and values reports are created according to the accepted standard for forensic science.
- Modification Timeline Analysis: Recognition of the last modified dates of registry keys for timeline reconstruction.

To help make it more competent, a submenu was created using Tkinter. This tool allows inexperienced users to conduct complicated registry critical analysis without programmatic

skills. Forensic science removes the need for experience code and significantly minimizes the time necessary to perform registry parsing procedures. The game provides a simple stepping stone by placing multiple powerful toolsets in the mouse point-and-click range (Carrier, 2005).

A fine example of Python in action is presented in this project, especially in the forensic capacity. The exceptional benefits of accurate analysis using registry-focused Python scripts do not end with resource labor-saving. As indicated by its kernel usage-driven facilitation of implementation advancement, there would be a need to approach such an obstacle within the development of investigation training. For instance, investigation training should be included in the qualifications package of law-enforcement cadets (Mandia & Prosise, 2003).

Methodology

The purpose we will be using Python in developing a forensic application mainly focusing on an aspect of a graphical user interface designed through `tkinter` and a Registry interaction tool based on `wing.` The tool also focuses on key forensic tasks such as, among others, obtaining the list of installed applications, recognition of hardware components, detection of the trend of changes of specific registry keys, collection of information from the user account, and then the reporting of the registry data. This design is divided into modules to remove most of the complex work and is thus suitable for even first-timers in the field. This approach will also illustrate a structured approach to designing, implementing, and validating the tool for use by the end users in all the phases of the tool (Gutiérrez, 2021).

Content

Our project focuses on the Windows Registry, a tree-structured database repository critical for storing system configuration settings and options on Windows machines. It also serves as a vital evidence source in digital forensic investigations, offering insights and information into user activities, software installations, and system modifications. However, analyzing the

Windows Registry manually is a tedious and error-prone process that requires specialized tools and knowledge. Our tool introduces a Python-based forensic investigation tool that leverages the tkinter library for the graphical user interface (GUI) and the Winreg module for direct interactions with the Windows Registry. Our tool simplifies registry analysis by automating everyday forensic tasks, such as software enumeration, hardware component identification, user login information retrieval, and data extraction. By offering a user-friendly interface, the tool empowers forensic investigators and examiners to perform comprehensive analyses efficiently, even without advanced technical expertise.

Tool Architecture

- GUI Design: Our tool uses the Tkinter library to create the GUI, ensuring easy use with clearly labeled buttons and menus.

- Registry Interaction: The tool utilizes the Winreg library to access and manipulate registry entries on Windows machines.

- Reporting Mechanism: Extracted data can be exported in a format suitable for forensic reporting and sent back to the user as a PDF do

Functional Module

1- Software Enumeration

Objective: Identify software installed on the system, including vulnerable applications.

Instructions:

- Launch the tool.
- Select "Enumerate Installed Software" from the GUI.
- View the list of installed software and their details in the output window.

Results:

```
C:\Users\shamm\PycharmProjects\SEC435\venv\Scripts\python.exe "C:/Users/shamm/PycharmProjects/SEC435
Installed Software:
Cisco Packet Tracer 8.2.0 64Bit
Cisco Packet Tracer 8.2.1 64Bit
Git
HP Documentation
Microsoft 365 Apps for enterprise - ar-sa
```

2- Hardware Component Details

Objective: Extract details about hardware components.

Instructions:

- Open the tool.
- Click on "Hardware Details" to generate a report of hardware components. ● Save the report for further analysis.

Results:

```
C:\Users\shamm\PycharmProjects\SEC435\venv\Scripts\python.exe "C:
CPU Information:
  Processor Name: 11th Gen Intel(R) Core(TM) i7-1165G7 @ 2.80GHz
GPU information registry key not found.
```

3- Modification Time Analysis

Objective: Identify the last modification times of specific registry keys.

Instructions:

- Navigate to the "Registry Key Modifications" tab.
- Input the desired registry path.
- The tool will display the selected keys' last modified date and time.

Results:

```
C:\Users\shamm\PycharmProjects\SEC435\venv\Scripts\python.exe "C:/Users/shamm/PycharmProjects/SEC435/venv/Scripts/
The registry key [Software\Microsoft\Windows\CurrentVersion] was last modified on: 2024-11-25 01:33:43.611168
```

4- User Login Information

Objective: Retrieve user account details and login information.

Instructions:

- Select "User Information" from the main menu.
- View user login details in the output window or export them to a file.

Results:

```
C:\Users\shamm\PycharmProjects\SEC435\venv\Scripts\python.exe "C:/Users/shamm/PycharmProjects/SEC435/venv/
User Accounts:
SID: S-1-5-18
  Profile Path: %systemroot%\system32\config\systemprofile
SID: S-1-5-19
  Profile Path: %systemroot%\ServiceProfiles\LocalService
SID: S-1-5-20
  Profile Path: %systemroot%\ServiceProfiles\NetworkService
SID: S-1-5-21-2651865815-3180320738-158640323-1001
  Profile Path: C:\Users\shamm
```

5- Registry Export Functionality

Objective: Export the registry contents in a forensic-compatible format.

Instructions:

- Choose the "Export Registry" option.
- Select the desired keys.
- Save the exported data in a specified location.

Results:

```
[Software\Microsoft\Windows\CurrentVersion\IrisService\Cache\18118111972904890887]
  IsSuccessStatusCode : 1
  StatusCode : 200
  RequestUri : https://fd.api.iris.microsoft.com/v4/api/selection?&asid=CF77639D86B2456F8E28E850B5F0C
  RawJson : {"batchrsp":{"ver":"1.0","items":[{"item":"{\"f\":\"raf\",\"v\":\"1.0\",\"rdr\":[{\"c\":\
  RefreshTime : 2024-10-24T16:18:25Z
  FirstExpiration : 2025-10-23T16:18:25Z
  LastExpiration : 2025-10-23T16:18:25Z
  UpdatedTime : 2024-10-23T16:18:25Z
```

6- Registry Viewer

Objective: Provide a clear view of the Windows Registry structure and allow users to navigate and inspect specific registry keys and values.

Instructions:

- Launch the tool.
- Select "Registry Viewer" from the GUI.
- Use the navigation panel to explore the registry structure and view key details.

Results:

```
C:\Users\shamm\PycharmProjects\SEC435\venv\Scripts\python.exe "C:/Users/shamm/PycharmProjects/SEC435/venv/Scripts
Values under Software\Microsoft\Windows\CurrentVersion\Run:

HPSEU_Host_Launcher : C:\System.sav\util\HPSEU\HpseuHostLauncher.exe
OneDrive : "C:\Program Files\Microsoft OneDrive\OneDrive.exe" /background
CiscoMeetingDaemon : "C:\Users\shamm\AppData\Local\WebEx\ciscowebexstart.exe" /daemon /from=autorun
MicrosoftEdgeAutoLaunch_04096CEAA4E1DF314C4E35956D69AC48 : "C:\Program Files (x86)\Microsoft\Edge\Application\mse
```

7- Installed Apps Tracker

Objective: Identify and list software installed on the system, including vulnerable applications.

Instructions:

- Launch the tool.
- Select "Installed Apps Tracker" from the GUI.
- View the list of installed software and their details in the output window.

Results:

```
C:\Users\shamm\PycharmProjects\SEC435\venv\Scripts\python.exe "C:/Users/shamm/PycharmProjects
Installed Applications:
App Name: Cisco Packet Tracer 8.2.0 64Bit
  Install Date: 20220823
App Name: Cisco Packet Tracer 8.2.1 64Bit
  Install Date: 20231122
```

Refer to the Appendix below to see the codes and the integrations.

Discussion

Tool Design and Purpose

The focus of creating eNcase software or an open-source version of the same is to simplify the complex data acquisition and examination process while maintaining the integrity of forensic procedures so that experts and beginners can use it effectively.

Aim of Software Development Functions

1. Finding Installed Software: Enhances the discovery of software debris that is likely to show user footprints or even malware.

2. Getting the System Information: it ensures that the specifications/product of the computer are correctly identified for the examination process to be effective.
3. Determining Changes in the Windows Registry: It examines the modification of registry keys whenever changes are performed.
4. All User Accounts are Returned: Searches for reachable or not reachable domains that contain information about users and users' activities and link specific actions to specific users.
5. Save Records from Registry: Create a report containing collected information that will conform to legal status requirements and general practice apt for investigation purposes.

Implementation Approach

1. The function of the proposed software will be outlined in the core libraries and systems it utilizes.

2. Tkinter: This solves the problem of crafting an abstract interface that directs researchers who are not skilled in scripting to utterly complex registry structures.
3. Python win reg module: Gives direct access to registry keys and values, which is necessary for obtaining accurate forensic and evidence–related data.

For the development of functions, each of the functions is engineered to cater to a specific requirement. Additionally, for error handling, there are multiple scenarios where application exceptions would be expected, such as insufficient privileges(Wampler, 2019)., missing keys, corrupted registry entries, etc. In addition, some are not user-related. Additional reports should be generated to plot the log and fix the errors of the functions, including:

Software Listing: This section reads the registry in the following order: HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows\Current

Version\Uninstall lists all software.

Hardware Detection: It retrieves information about the CPU, Video, and other hardware elements from the selected registry keys.

Registry Modification Times: Sheet 1 leverages the registry critical metadata to determine the last modification of the key.

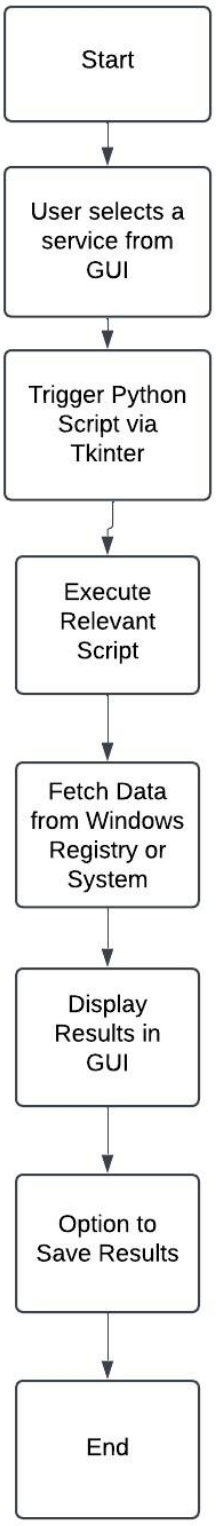
User Account Enumeration: - Reads the ProfileList under HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows NT\CurrentVersion to retrieve user profiles.

Export Report: It went through directories and found all the data, putting it inside a file.

Chain-of-Analysis:

Using SHA-256 hashing and a chain of analysis approach, our program adds additional capabilities that improve its accuracy and dependability. By offering a transparent and verifiable route through the study, the chain of analysis guarantees a methodical inspection procedure. Furthermore, data integrity and authenticity are ensured by incorporating SHA-256, a very safe and distinctive hashing method. As a result, our equipment offers unparalleled precision and reliability, making it efficient and a strong forensic analysis option. Our tool offers cutting-edge features intended to improve accuracy and dependability. It integrates SHA-256 hashing, a safe and distinct algorithm that ensures data integrity and authenticity, and employs a chain of analysis approach, guaranteeing a methodical and traceable inspection procedure. The tool is now reliable and strong for forensic investigations thanks to these upgrades. The tool also provides a thorough reporting option. Users can obtain comprehensive reports in PDF format, which provides an orderly and expert summary of the results. A log function also painstakingly documents each stage of the procedure, offering complete accountability and transparency. Together, these characteristics guarantee that the tool maintains the trust and dependability necessary for forensic analysis and produces accurate results.

Workflow Representation



This workflow Diagram shows how to use Python to communicate with a graphical user interface (GUI) in order to retrieve and display system-related data. The process starts when a user chooses a certain service or activity from the GUI interface. This choice starts a Python script that runs the pertinent script linked to the selected service. The script is probably written with the help of the Tkinter Library. The script retrieves the necessary data, including details from the Windows registry or other system-level information. The GUI gives the user an understandable visualization of the outcomes after retrieving the data. To ensure flexibility and convenience, the user is also allowed to save the results displayed for later use. At last, the procedure concludes, concluding the exchange.

Integration and Testing

Development Integration

- A modular approach was adopted to implement functions to enable easier integration of and independent testing for the functions developed.

- The user interface has been linked to the backend, combining its operations and the crime scene that is being monitored in real-time.

Testing Scenarios

- Environment Validation. The tool was tested on Windows 10 and 11 standard and administrative systems.
- Functionality Verification. To ensure accuracy, the authors manually cross-validated the outputs with the manual registry exploration.
- Performance Test. A performance deployment test assessed how the tool elevated large registry trees.

Expansive Methodology

The rich set of tools is one of the prominent features. This is because it can serve the basic needs of forensic science while still being open to creating other tools, such as malware registry keys or integration with other forensic tools. Its ability to change with time makes people constantly want to use it every time adjustments are made, which is beneficial when conducting a forensic investigation.

Applicability in Forensic Research

This feature demonstrates how Python is an ideal scripting language for forensic science, as illustrated by the automated registry analysis within the tool. The tool is developed to cater to all storage systems found within the forensic sciences, including all subsystems. It is designed to eliminate the need for excessive use of registry entries for working with a particular system, and this provides a means that a criminal investigator can adopt this tool to uncover such evidence.

This methodology is an improvement that bridges the present requirements in forensic analysis and lays down the groundwork for what is to come in incident handling and digital evidence collection. discussion

Discussion

This work achieves a very extensive forensic toolset, making the analysis registry easier and quicker. Each component is unconditionally aimed at a particular forensic field of study to ensure that the investigator is wholly involved in gathering and interpreting electronic evidence. In reviewing the system, the chapter attempts to investigate every facet of the application and explain how each function supports forensic examiners and aids the investigative process Luttgen (Pepe, & Mandia, 2014).

Methods & Materials

Software Enumeration

Objective: The installed programs in the system are to be aforementioned.

Forensic Relevance:

- Discovering wrecked applications significantly aids the understanding of the software system enveloping the system.

- Examiners can locate and root out zenviruses, strangulated applications, or suspicious software, indicating that malware or unauthorized means to break into the system are present

- Effective in locating digital manifestations left behind by a specific application such as logs, caches, or configurations.

Such a feature helps the eminent completion of these forensic examinations because the investigation is now achieved using this program. Without such focus and detailing, vital information within the system environment might be missed due to human error.

Hardware Information Retrieval

Purpose: Tries to extract CPU and GPU details from a registry.

Forensic Relevance:

- Information concerning the hardware setup helps give the baselines against which system operation is evaluated relatively quickly.

- In those exceptional cases, some mismatch or alterations on the hardware may suggest tampering or unreported updates, which could also be used in a forensic audit.
- In contrast, this section also explains how the system records the time it takes to boot hardware.

Last Access Times for Registry Keys

Objective: Allows the user to view the last time a specified registry key was changed.

Importance within the scope of Investigations:

- Modification timestamps, especially on the registry, help keep the activity level in precise order.

  The examiners can relate the times of modification and other logs or artifacts to understand the order of events, for example, the installation of specific programs, changes in the system, or running of malware.

- Thanks to this option, there is no need to overcome the hurdle of fetching and interpreting time stamps, which helps ensure correct event positioning.

User Account Enumeration

Objective: Displays known user accounts and their profile path and directory.

Importance within the scope of Investigations:

- It is important to note individual user accounts and their actions to connect actions or artifacts to individuals.
- The profile's directory path helps the examiners access locations with user-associated information like files on the desktop, downloads, and program data that carry pertinent information.
- This is also useful in looking for any unauthorized or created accounts that may pose an insider threat or a threat of compromise.
- The view that enables so much detail to be shown is useful Davis has automated the enumeration so no accounts are missed out, especially when there are many accounts in the system.

Registry Values Display

Objective: Enumerate and display all values and data under a specified registry key.

Importance within the scope of Investigations:

- The capability of viewing registry values helps determine the various structures as well as identify the various start-up elements and application parameters.
- The Examiners can find strategies for persisting malicious code, which might include such items as hidden services or startup programs that are usually kept in the registry.

  This function also facilitates using simple cut-and-paste techniques in capturing key and value pairs thus making it easier to search for and find incriminating evidence.

Installed Applications with Install Dates

Aim: List the applications with installation dates on the respective devices.

Application of Cartridge:

- This includes the dates of installation and other changes.
- It will help the forensic experts in the investigation.
- It will help the examiner determine whether the suspicious application was installed right after the security breach or corresponds with the suspicious activity in other ways.
- Considering the life cycle of attack or unauthorized utilization of the resources will also be necessary.

Export Registry Data for Reporting Purposes

Aim: Is the exporting of registry data to a formatted file.

Extent of IT Forensic Analysis

Registry data is extracted for legal issues and other reasons and for computer forensics. Such a process of automatic export ensures the integrity of the evidence by giving the complete representation of the data in question.

The database approach is helpful to the investigator in that facts are shown in a regulated format, posing sinking barriers in the path of an investigation.
It also enables obliterating the preparation that people have to make when documenting registry content on paper, thus voiding the making of errors.

Findings Summary

The forensic investigator is equipped with the most required dynamics through this tool to gather evidence called computer and mobile wellness. By eliminating most of the redundant activities and information display, the main objective of the tool is to:

1. Help investigators work efficiently on the data, considering that more time is spent on analysis rather than reinforcement.

2. Eliminate the risk of validity and help ensure the accuracy of the work done.

3. Help relate one sequence after another in computing, detect harmful activities, and determine when a specific user has taken control of the computer and performed certain Windows actions.

Each functionality is designed to perform activities relevant to the discipline and is thus helpful in providing all the necessary tools and techniques to conduct detailed investigations. These extensive functionalities, loaded with ease of use, help the tool be productive within professional and self-imposed boundaries regarding forensic opportunities. For instance, the tools show how Python can be used for digital forensics, particularly for collecting and analyzing information saved in the Windows Registry. Capturing lists and exporting the registry to a file has been previously automated to save time and eliminate errors. The Winreg library allows you to work with registry keys(Microsoft Corporation, 2024).

Thus, any targeting also becomes highly efficient and accurate. On top of that, the functionality is complemented by specific other capabilities, like dating information extraction and investigation of user accounts that are core to forensic science purposes such as timelining and tracking user activities. For example, issues connected with working with protected records were solved using error management techniques, guaranteeing the effective and smooth running of the processes. In summary, it is clear that the toolkit is not only user-friendly for forensic investigators, but it is also a vital addition to open-source forensic tools (Python Software Foundation, 2024).

Conclusion

This forensic toolkit based on Python is an effective tool for analyzing the Registry information found in the Windows Bits registry. Its functions help satisfy many demands associated with forensic protocols, such as software count, hardware information recovery, and user account logs. With the integration of Winreg and Tkinter used for user interface automation, the toolkit fills the gap between technological advances and ease of use. What could be better is to offer functionality extension – automatic timeline creation and connection with higher forensic science specialized instruments and applications, to help cope with changing forensic needs.

References:

Carvey, H. (2011). *Windows Registry Forensics: Advanced Digital Forensic Analysis of the Windows Registry*. Elsevier.

Casey, E. (2011). *Digital Evidence and Computer Crime: Forensic Science, Computers, and the Internet*. Academic Press.

Carrier, B. (2005). *File System Forensic Analysis*. Addison-Wesley.

Mandia, K., & Prosise, C. (2003). *Incident Response: Investigating Computer Crime*. McGraw-Hill.

Luttgens, J., Pepe, M., & Mandia, K. (2014). *Incident Response & Computer Forensics*. McGraw-Hill.

Python Software Foundation. (2024). *Python Documentation: winreg Module*. Retrieved from https://docs.python.org/3/library/winreg.html

Microsoft Corporation. (2024). *Windows Registry Documentation*. Retrieved from https://docs.microsoft.com

Wampler, L. (2019). *Mastering Python for Forensics*. Packt Publishing.

Gutiérrez, J. (2021). *Modern Forensics: Leveraging Open-Source Tools*. Forensic Science Review.

Shinder, D. (2016). *Scene of the Cybercrime: Computer Forensics Handbook*. Elsevier.

Carrier, B. (2005). *File System Forensic Analysis.* Addison-Wesley.

Casey, E. (2011). *Digital Evidence and Computer Crime: Forensic Science, Computers, and the Internet.* Academic Press.

Mitnick, K. D., & Simon, W. L. (2011). *The Art of Deception: Controlling the Human Element of Security.* Wiley.