**Malware Analysis Lab Tools**

Nwachukwu Edumanichukwu

Bellevue University

CYBR 445-308H Advanced Intrusion Detection and Response

Professor Chelsie Thompson

July 10, 2023

**Malware Analysis Lab Tools**

It is crucial to have a wide range of tools that can effectively dissect and analyzer malicious codes when setting up a complete malware analysis lab. This document outlines and provides overviews of the specific tools of choice for different category and its functions, costs, and use.

* **Decompiler, Disassembler, Software Reverse Engineering Framework: Ghidra**

Ghidra is a powerful and feature-rich software reverse engineering framework developed by the National Security Agency (NSA). It offers a suite of tools for analyzing and understanding binary code. Ghidra's decompiler and disassembler modules assist in transforming machine code into a human-readable format, aiding in the understanding of the malware's behavior and inner workings.

Cost: Free and open source

Use: Ghidra will be used extensively to disassemble and decompile malware samples, enabling analysts to study the code and identify any malicious functionalities or vulnerabilities.

* **Hex Editors: Cerebro Suite**

Cerebro Suite is a comprehensive set of hex editors designed specifically for reverse engineering tasks. It provides a range of features to analyze and modify binary files at a low-level, making it a valuable tool for inspecting malware samples and uncovering hidden information.

Cost: Commercial tool with a pricing model based on licensing options.

Use: The hex editors within Cerebro Suite will allow analysts to examine the raw binary content of malware files, perform manual modifications, and gain insights into the structure and behavior of the malicious code.

* **PE Analysis and Miscellaneous Analysis: PE Studio**

PE Studio is a specialized tool for static analysis of Portable Executable (PE) files, which are commonly used by Windows executable files. It provides a detailed overview of the file's characteristics, dependencies, resources, and imports, allowing analysts to understand the malware's capabilities and potential impact.

Cost: Commercial tool with a free version available for basic functionality.

Use: PE Studio will be employed to analyze PE files, identify suspicious or malicious attributes, and extract valuable information about the malware's behavior and potential payloads.

* **.NET Disassemblers: .NET Reflector**

.NET Reflector is a widely used decompiler and static analysis tool for Microsoft .NET applications. It enables analysts to decompile .NET binaries into high-level C# or VB.NET code, making it easier to comprehend and analyze the inner workings of .NET-based malware.

Cost: Commercial tool with a free version available with limited features.

Use: .NET Reflector will be used to decompile and analyze .NET malware, allowing analysts to study the source code, understand its functionality, and identify any potentially malicious behaviors.

* **YARA Editor: Atom/Sublime**

Atom and Sublime are popular text editors with extensive plugin ecosystems, making them suitable choices for editing YARA rules. YARA rules are used for pattern matching and identification of specific malware characteristics or indicators.

Cost: Atom and Sublime have free versions available, with paid versions offering additional features.

Use: Atom or Sublime will serve as the YARA rule editors in the lab, providing a user-friendly interface for creating, editing, and managing YARA rules, which are vital for the detection and classification of malware samples.

* **YARA Scanner: Loki**

Loki is a YARA-based scanner tools that utilize YARA rules to scan files and directories for the presence of malware indicators. It offers efficient and scalable scanning capabilities to detect known malware patterns.

Cost: Loki is open source and freely available.

Use: Loki will be used to scan files and directories for malware using the YARA rules defined in the lab. It will help identify potentially malicious files and provide an initial assessment.

* **Sigma Editor: Atom/Sublime**

Atom and Sublime, being versatile text editors, can also be used for editing Sigma rules. Sigma rules provide a standardized format for describing and detecting security events in log data.

Cost: Atom and Sublime have free versions available, with paid versions offering additional features.

**Use:** Atom or Sublime will serve as the Sigma rule editors in the lab, enabling analysts to create and modify Sigma rules to enhance log analysis and identify potential security events related to malware activities.

* **Sigma Converter/Translator: Sigma2SplunkAlert/Sigmac**

Sigma2SplunkAlert and Sigmac are tools that aid in the conversion and translation of Sigma rules to specific SIEM (Security Information and Event Management) formats, such as Splunk. They streamline the integration of Sigma rules into existing security infrastructure.

Cost: Sigma2SplunkAlert and Sigmac are open source and freely available.

Use: These tools will facilitate the conversion of Sigma rules to SIEM-specific formats, enabling seamless integration with SIEM systems like Splunk, thus enhancing the lab's ability to detect and respond to malware-related security events.

**References**

Ghidra- <https://ghidra-sre.org>

Cerbero Suite -<https://cerbero.io>

PE Studio- <https://www.winitor.com>

.Net Reflector- <https://www.red-gate.com/products/reflector/>

Atom- <https://github.com/atom/atom>

Sublime- <https://www.sublimetext.com>

Loki- <https://github.com/Neo23x0/Loki>

Sigma- <https://github.com/SigmaHQ/sigma>

Sigma2SlunkAlert- <https://github.com/P4T12ICK/Sigma2SplunkAlert>