Malware Analysis Report

1. Executive Summary

This report presents the findings and analysis from a comprehensive malware analysis . The objective of this analysis was to assess the threat posed by a specific malware variant and to provide insights into its behavior, characteristics, and potential impact on organizational systems and data.

2. Introduction

The proliferation of malware poses significant risks to organizations, threatening the integrity, availability, and confidentiality of data and systems. To effectively mitigate these risks, it is essential to conduct thorough malware analysis, employing both static and dynamic analysis techniques. This report outlines the methodology employed, tools utilized, and the key findings from the analysis.

3. Methodology

The analysis methodology consisted of two primary approaches:

1. Static Analysis: This involved examining the code and structure of the malware without executing it. Techniques such as code inspection, signature matching, and pattern recognition were employed to identify indicators of compromise and potential vulnerabilities.

2. Dynamic Analysis: Dynamic analysis involved executing the malware in a controlled environment to observe its behavior in real-time. This allowed for the monitoring of runtime behavior, including file modifications, network communication, process creation, and system calls.

4. Tools Used

A suite of specialized tools was utilized for both static and dynamic analysis:

Static Analysis Tools:

- PEiD:https://www.aldeid.com/wiki/PEiD

- PEView:https://www.aldeid.com/wiki/PEView

- Dependency Walker:https://www.dependencywalker.com/

- Sysinternal Suite:https://learn.microsoft.com/en-us/sysinternals/downloads/sysinternals-suite

Dynamic Analysis Tools:

- Regshot:https://github.com/Seabreg/Regshot

- Process Monitor:https://learn.microsoft.com/en-us/sysinternals/downloads/procmon

- Process Explorer:https://learn.microsoft.com/en-us/sysinternals/downloads/process-explorer

- Process Hacker: https://processhacker.sourceforge.io/

- Apache DNS: https://httpd.apache.org/docs/2.4/dns-caveats.html

- Inetsim: https://www.inetsim.org/

Reverse Engineering:-

OllyDbg:- https://www.ollydbg.de/

Network Monitoring:-

Wireshark:- https://www.wireshark.org/

5. Analysis Findings

The analysis revealed the following key findings:

1. File Type: The malware sample was identified as an executable (.EXE) file, commonly found in Windows systems.

2. Hash Values:

- **MD5:** CFB11BD8FB20CC92AB5AAB606B2CFAF1

- **SHA1:** 4560996ecf46c1ab377cb6c22ab88ac8b5d44608

- **SHA256:** 39a8f452064a1daeeee2af8e3f411877851c07c13cb1a41a0e08d9b04c2525fd

3. Language: The analyzed file was compiled in C# language, denoted by the .EXE extension.

4. Dependencies: The malware relies on the following malicious DLLs:

1. KERNEL32.DLL

2. USER32.DLL

3. ADVAPI32.DLL

4. SHLWAPI.DLL

5. VERSION.DLL

6. OLEAUT32.DLL

7. URLMON.DLL

8. MSCOREE.DLL

5.Indicators of Compromise (IoCs):

- **Host-based IoCs:**Host-based Indicators of Compromise (IoCs) include file anomalies like suspicious changes, unknown processes exhibiting abnormal behavior, unauthorized network connections, privilege escalations, and modifications to system configurations or logs. Additionally, anomalies in user behavior and anti-forensic techniques may signal compromise. Detecting these IoCs helps identify security incidents, prevent further damage, and initiate appropriate response measures, safeguarding the integrity and security of the host system.

- Network-based IoCs:Network-based IoCs are identifiers used to detect cybersecurity threats within a network infrastructure. These IoCs include suspicious IP addresses, domain names, URLs, and network traffic patterns. By continuously monitoring network activity and comparing it against known IoCs, security teams can swiftly identify and respond to potential threats such as malware infections, data breaches, or unauthorized access attempts. Network-based IoCs play a crucial role in bolstering the security posture of organizations and mitigating cyber risks.

6. Conclusion

In conclusion, this analysis underscores the importance of proactive cybersecurity measures and robust defense strategies against evolving malware threats. By employing a structured approach to malware analysis and leveraging advanced tools and techniques, organizations can effectively detect, mitigate, and respond to cyber threats. The findings from this analysis provide valuable insights that can inform cybersecurity strategies and enhance organizational resilience against malware attacks.