Malware Project

Readme.txt

Team Members (3 members): Meenal Shah, Nagendra Kaushik Godlaveti, Yash Mahesh Malpathak

Executable: WatchMeTearItApart.exe

SourceCode:

- **Malware downlaoder source code:** Malware_Source_Code.cpp
- **Payload:** Mawalre_Source_Code - Payload.cpp

Introduction

The malware program (Downloader) works by first downloading a secondary payload (a shutdown program) from the internet (GitHub) and saving it in the Startup folder to ensure it executes on system reboot. Once the secondary payload is executed on system startup, it enables the necessary privileges to shut down the system, displays a misleading message box to the user, and then forces the system to shut down.

Requirements to Run

1. A Windows 10 VM
2. An internet connection

Malware functionality - Downloader

1. First Part: The Initial Malware Code

- Debugger Detection: The program checks if it's being run under a debugger. If a debugger is detected, it terminates immediately to avoid analysis.
- Downloading the Payload: The malware attempts to download a file from a specified GitHub URL (shutdown.exe) using the WinINet API. This file is then saved to the Startup folder (%userprofile%\AppData\Roaming\Microsoft\Windows\Start Menu\Programs\Startup) on the victim's system, ensuring it will be executed automatically each time the system restarts.

- Persistence: The malicious executable is saved in the Startup folder (my_application.exe), ensuring it runs when the system starts, making the malware persistent across reboots.
- Message Box Display: After the malicious file is downloaded and saved, the malware shows a message box to the user with the message "You are hacked."

2. Second Part: The Payload Code

The second part is the payload that is downloaded from the Github URL by the initial malware. This payload is designed to shut down the victim's system. Here's an explanation of how the payload works:

2.1 Enable Shutdown Privileges:

- The function EnableShutdownPrivileges() is responsible for enabling the necessary system privileges that allow the program to shut down the computer.
- It uses the Windows API function OpenProcessToken() to open the current process's token, allowing it to adjust privileges.
- The privilege needed to shut down the system is SE_SHUTDOWN_NAME, and the code uses LookupPrivilegeValue() to retrieve the privilege's LUID (Locally Unique Identifier).
- The privilege is then enabled by calling AdjustTokenPrivileges(), allowing the system to be shut down by the program.

2.2 Shutdown the System:

- The ShutdownSystem() function checks whether the shutdown privileges are successfully enabled (by calling EnableShutdownPrivileges()).
- If the privileges are successfully enabled, it then calls ExitWindowsEx() to perform the actual shutdown of the system. The EWX_SHUTDOWN flag ensures that the system is shut down, and EWX_FORCE forces applications to close without prompting the user for unsaved data.
- The shutdown reason codes (SHTDN_REASON_MAJOR_OTHER and SHTDN_REASON_MINOR_OTHER) are used to indicate the reason for the shutdown in system logs.

2.3 Message Box:

- Before proceeding with the shutdown, the program displays a message box that says "Congratulations! You will get free pizza!!".
- After the message box is acknowledged, the system shutdown proceeds.

Malware Techniques Used:

1. Anti-Debugging Mechanism:

The malware includes an anti-debugging technique where it checks if the program is being run under a debugger using the IsDebuggerPresent() function. If a debugger is detected, the program immediately terminates. This prevents the analysis of its behaviour in a controlled environment, making it harder for researchers to reverse engineer the malware.

2. Persistence Mechanism:

The malware ensures persistence by placing a copy of the downloaded file in the Startup folder (%userprofile%\AppData\Roaming\Microsoft\Windows\Start Menu\Programs\Startup). This technique ensures that the malware will run every time the user logs into their system. The Startup folder is a common method used by malware to maintain persistence across system reboots.

3. File Download via HTTP(S):

The malware uses the WinINet API to download an executable from a remote server. This technique is commonly used in malware to retrieve additional payloads after the initial infection. The downloaded file (shutdown.exe) is a payload from a GitHub raw file.

4. Obfuscation with UPX Packing:

The malware executable is packed using UPX (Ultimate Packer for Executables), which is a popular tool to reduce the size of executables and obfuscate their content. Packing makes it harder for static analysis tools to examine the executable's code.