Supply Chain Management of

Footwear Industries in Bangladesh

Supply Chain Management of

Footwear Industries in Bangladesh

**Reverse Engineering TTC6510-3002**

Joonatan Ovaska

A K M MAHMUDUL HAQUE

AB0208

**Student number:** 2110841

**WinLab01**

Date: 24.10.2023

**First Step**

* Conducting both the static and the dynamic analysis.
* Using **FakeNet** malware traffic is spotted.
* It uses DNS port 53 and HTTP port 80.

**A screen shot of a computer

Description automatically generated**

* It creates a file **wqaeoiur.exe** after the malware is run.
* **wqaeoiur.exe** is set to autorun when Windows starts by making changes in registry.
* **procmon** program show these traffics

**A screenshot of a computer

Description automatically generated**

HKCU\SOFTWARE\Microsoft\Windows\CurrentVersion\Run\(Default) “C:\Users\user\AppData\Local\wqaeoiur.exe”

* Malware also modifies registries. One obvious example is that it runs the **wqaeoiur.exe** to run at every time computer starts.

HKU\S-1-5-21-2882983514-2000211610-2302286010- 1001\SOFTWARE\Microsoft\Windows\CurrentVersion\Run\: "C:\Users\user\AppData\Local\wqaeoiur.exe"

**Second Step**

* Doing the static analysis using **exeinfo** or **CFF Explorer,** dependencies the malware is using is found.
  + **winhttp.dll**
    - HTTP server interaction related functions
  + **kernel32.dll**
    - Core functionalities such as access and manipulation of memory, files, hardware
  + **advapi32.dll**
    - Access to Service Manager and Registry
  + **shell32.dll**
    - Functions related to file operations, search, desktop management, taskbar and start menu, UI elements.
* MD5 Hash Comparison: The MD5 hashes of two files, winlab01.exe and wqaeoiur.exe, were compared.
* Matching Hash: The MD5 hashes matched, indicating that the two files are identical and likely represent the same malware.
* MD5 Hash Value: The MD5 hash value for both files was identified as e3d948329c3c96013706a8270cf52853.
* Internet Search: Using this MD5 hash, a search was conducted on the internet, revealing that someone else had also analyzed this malware.

**Link:** [*https://www.virustotal.com/gui/home/upload*](https://www.virustotal.com/gui/home/upload)

**Third Step**

* **HTTP GET Request:** The malware initiates an HTTP GET request to a specific IP address.
* **WinHTTP Function Disassembly:** The WinHTTP function responsible for handling the HTTP communication was disassembled for analysis.
* **Preparation:** The function starts with preparations for opening an HTTP connection, including setting up necessary parameters and configurations.
* **Connect Call:** After the preparations, the malware calls the connect function, likely establishing the HTTP connection to the specified IP address.

A screenshot of a computer program

Description automatically generated

WinHTTP Open, Connect

A screenshot of a computer

Description automatically generated

Request, OpenRequest, SendRequest, ReceiveResponse

A screenshot of a computer program

Description automatically generated

WinHTTP QueryDataAvailable, ReadData

A screenshot of a computer program

Description automatically generated

CloseHandle, GetLastError

**Fourth Step**

* malware creates **wqaeoiur.exe** and modifies registries to set it to autoexecute.

**A screen shot of a computer code

Description automatically generated**

* lpSubKey is set to ”SOFTWARE\Microsoft\Windows\CurrentVersion\Run”.

****

RegSetValueExA, RegCloseKey

**Fifth Step**

* **Anti-Debugging Technique:** This technique aims to prevent or complicate live debugging of the malware.
* **Custom Error Handling:** The malware includes a function that allows it to handle errors on its own.
* **Normal Scenario:** In regular situations, if an error occurs, the operating system steps in, displaying a message or terminating the program.
* **Debugging Scenario:** If the application is being debugged, the custom error handler is bypassed, allowing standard debugging processes to take over.

**A screen shot of a computer code

Description automatically generated**

* Check for debugger presence, set Unhandled exceptions in figure.

**A computer screen shot of a computer program

Description automatically generated**

* Malware Check: The malware attempts to determine if the processor has the **\_fastfail()** feature (represented as 0x17 or 23 in decimal).
* Purpose of **\_fastfail()**: This feature can be used to trigger Windows exception handling or activate custom exception handling.
* Int 3 as Breakpoint: int 3 is a common instruction used as a breakpoint in debugging.
* Conditional Jump: If the check for **\_fastfail()** feature passes (test **eax, eax**), the program continues to the next instructions.
* Subroutine Call: The malware then calls **sub\_405E74**, which contains instructions related to another Unhandled Exception Filter.
* in the **loc\_405A08** there is call **sub\_402FF8**. This call eventually leads to (given  
  the conditions met) **GetCurrentProcess**, **TerminateProcess** and **ExitProcess**

**A screenshot of a computer

Description automatically generated**

* After checking for debugger presence andUnhadled Exception calls, malware might be preventing debugging