

School of Design and Informatics

BSc Ethical Hacking, 2022/23

**An Evaluation of Modular Incident Response Plans for Efficient Cyber Incident Mitigation in Businesses**

**FileTour Analysis**

**Author:** Martin Georgiev **Supervisor**: Natalie Coull

Head of Division of Cybersecurity Abertay

Contents

[1 Introduction 1](#_Toc134540643)

[**1.1** **Background** 1](#_Toc134540644)

[**1.2** **Aims** 1](#_Toc134540645)

[2 Procedure 2](#_Toc134540646)

[**2.1** **Overview** 2](#_Toc134540647)

[**2.2** **Static Analysis** 2](#_Toc134540648)

[**2.2.1** **File Hashes Discovery** 2](#_Toc134540649)

[**2.2.2** **String Extraction** 3](#_Toc134540650)

[**2.2.3** **In-depth Inspection** 5](#_Toc134540651)

[**2.2.4** **Dropped Files** 8](#_Toc134540652)

[**2.2.4.1** **9840432e051a6fa1192594db02b80a4c1fd73456** 8](#_Toc134540653)

[**2.2.4.2** **lg** 14](#_Toc134540654)

[**2.2.4.3** **LivelyScreenRec** 14](#_Toc134540655)

[**2.2.4.4** **Note866** 15](#_Toc134540656)

[**2.2.4.5** **PBrowFile15** 16](#_Toc134540657)

[**2.2.4.6** **Downloaded payloads** 18](#_Toc134540658)

[**2.3** **Dynamic Analysis** 19](#_Toc134540659)

[**2.3.1** **Detonation Symptoms** 19](#_Toc134540660)

[**2.3.2** **Network Symptoms** 20](#_Toc134540661)

[**2.3.3** **Host Symptoms** 21](#_Toc134540662)

[**2.3.4** **Memory Analysis** 26](#_Toc134540663)

[**2.4** **Yara Rule** 29](#_Toc134540664)

[3 Results 31](#_Toc134540665)

[**3.1** **Static Analysis** 31](#_Toc134540666)

[**3.2** **Dynamic Analysis** 31](#_Toc134540667)

[4 Discussion 32](#_Toc134540668)

[**4.1** **General Discussions** 32](#_Toc134540669)

[**4.2** **Countermeasures** 32](#_Toc134540670)

[**4.2.1** **Pre-infection Countermeasures** 32](#_Toc134540671)

[**4.2.1.1** **Frequent Security Updates** 32](#_Toc134540672)

[**4.2.1.2** **Distinguishing Spam** 32](#_Toc134540673)

[**4.2.1.3** **Blacklisting Unknown Applications and Anti-Virus Software** 32](#_Toc134540674)

[**4.2.1.4** **File Scanners and IDS** 32](#_Toc134540675)

[**4.2.2** **Post-infection Countermeasures** 33](#_Toc134540676)

[**4.2.2.1** **Data Backups** 33](#_Toc134540677)

[**4.2.2.2** **Ransom Payments** 33](#_Toc134540678)

[**4.2.2.3** **Find the Infected Freeware** 33](#_Toc134540679)

[5 Conclusion 34](#_Toc134540680)

[**5.1** **Conclusion** 34](#_Toc134540681)

[**5.2** **Future Work** 34](#_Toc134540682)

[6 References 35](#_Toc134540683)

[7 Appendices 37](#_Toc134540684)

[**Appendix A – Strings** 37](#_Toc134540685)

[**Appendix B – Memory Forensics** 37](#_Toc134540686)

[**Appendix B1 – Command Line** 37](#_Toc134540687)

[**Appendix B2 – Network Connections** 38](#_Toc134540688)

[**Appendix B3 – Malfind** 39](#_Toc134540689)

[**Appendix B4 – Privileges** 40](#_Toc134540690)

[**Appendix B5 – Processes** 41](#_Toc134540691)

[**Appendix C – Yara Rule** 49](#_Toc134540692)

# **Introduction**

## **Background**

**Adware.FileTour** is the generic detection name in Malwarebytes for a family of adware bundlers that originate from Russia and target Windows machines. FileTour is also called Bundleware, as its main purpose is to bundle multiple executables and extract them on the targeted machine. Whether bundleware is malicious or not depends on the payload they deliver to the machine. In FileTour’s case, it depends on the sample – some samples deliver adware, while others act as spyware and gather user data (browser cookies, location, credentials, IP address, browsing history for ad catering, etc.)

Based on the most common capabilities of the delivered executables, FileTour is usually considered adware. However, this may not always be the case. The bundleware heavily obfuscates its contents and it can be difficult to distinguish whether it can steal data, simply deliver intrusive ads or a combination of both. A good example of a payload delivered by FileTour is the **Stantinko** adware botnet. The malware loudly installed several PUP while silently installing the malicious payload to disorient researchers. The malicious software installed altered copies of two adblocker Google Chrome extensions that had the exact opposite effect – they constantly displayed ads based on the user’s browsing data. The botnet also left an open backdoor on the infected machines, allowing the attackers to remotely execute code on the systems.

As the researcher was unable to obtain this specific sample, they analysed a different one that utilised spyware capabilities. This showed that despite the detection name in Malwarebytes, FileTour is not always related to adware but also to more serious threats.

## **Aims**

The report aims to provide the reader with an analysis of a FileTour sample with the simplified Malware Analysis and Digital Forensics Framework and how it can show them information about its capabilities. The information can be used to take precautions until a professional team starts handling the case. It can also be given to the team handling the incident as this could greatly decrease the time to respond to the attack. The report will be split into three major sections to efficiently achieve the goal:

* **Procedure** – Analysis of the sample using the methodology and creation of a Yara rule to detect its signature
* **Results** – Overview and summary of the procedure and its subsections
* **Discussion** – General discussion and appropriate countermeasures

# **Procedure**

## **Overview**

As the developed methodology is aimed at small and medium-sized businesses that may not have a specialised response team due to budget constraints, the procedure will attempt to obtain as much intel as possible with simple techniques and tools. This will ensure that even users without significant technological knowledge can seamlessly follow it without significant difficulties. Considering the beforementioned requirements, an advanced static and dynamic analysis will not be performed on the sample as they require extensive knowledge of how computers operate, as well as high and low-level programming languages (JavaScript, C++, Assembly, etc.)

The hybrid analysis will be conducted on the sample in a safe testing environment. The static analysis will be achieved with a multitude of techniques and tools – obtaining file hashes, inspecting any human-readable strings within the binary (**Strings** (Russinovich, 2021) and **Floss** (Ballenthin, 2016)), analysing the executable’s library imports, functions and file entropy (**PEStudio** (Fox, 2021), and checking whether any known file packers have been used (**Exeinfo PE** (ALS, 2023)).

The Dynamic and Digital Forensic analysis will be combined as both can be done simultaneously. This section will cover the post-execution behaviour of the malicious sample – inspection of system modification (deleted/created/altered files, registries; PowerShell cmdlets execution, detection evasion and persistence mechanisms) with **Procmon**, possible network propagation or attempted communication with a **C&C** (**TCPView** (Russinovich, 2022), **Inetsim** (Hungenberg and Eckert, 2007)and **Wireshark** (Wireshark, 1997 – Present Day)). Additionally, analysing the system’s memory could display hidden processes, stolen data stored in the clipboard, possibly recover encryption keys and many more (**Volatility** (Volatility Foundation, 2020) and **WinPMem** (Cohen, et al., 2019)).

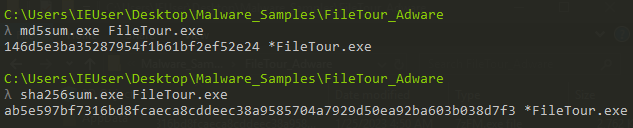
In the end, the obtained intel will be used to create a **Yara** (VirusTotal, 2013) rule for signature scanning that can then be incorporated with a scanner such as Strelka for passive file metadata scanning.

## **Static Analysis**

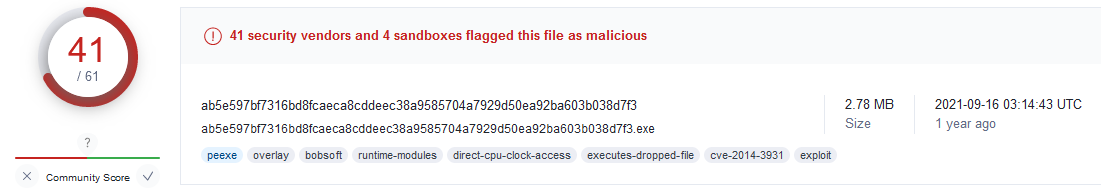
As mentioned in the previous section, the analysis will start with the static analysis of the sample. The basic static analysis provides limited information about the malware’s functionality. This, however, is often enough in introducing analysts with a base overview of the malicious software’s capabilities.

### **File Hashes Discovery**

To obtain the hashes of the malware, tools such as **md5sum.exe** and **sha256sum.exe** can be used from the command prompt. They will display the MD5 and SHA256 hashes of the sample respectively. The user can then copy them and check them in VirusTotal. This will check the databases of multiple antivirus vendors and display any matching results. (**Figure 2.2.1** and **Figure 2.2.2**)



***Figure 2.2.1*** *– Obtaining MD5 and SHA256 hashes.*

**

***Figure 2.2.2*** *– VirusTotal indicating the sample is malicious.*

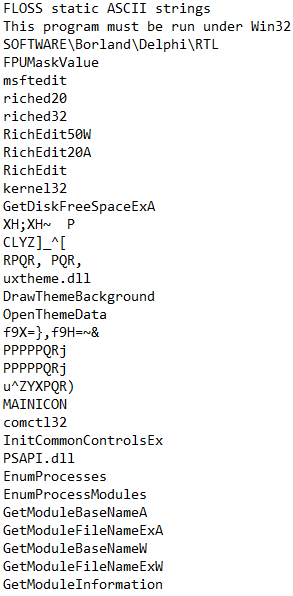
### **String Extraction**

Human-readable strings can be extracted from files with a variety of tools. Some of the most common ones are **strings** and **floss**. In some cases, **floss** may be a better alternative to the former tool as it attempts to de-obfuscate and decode any strings which were intentionally made hard to read. The tool also has a multitude of flags for data filtering. In this case, **-n** was used as this puts a minimum character length of the extracted strings. The analyst set the length to **8** as they were interested in possible sentences, links and/or imported functions/libraries. The output was piped onto a text file (with **>**) to allow easier analysis and further filtering. (**Figure 2.2.3**)



***Figure 2.2.3*** *– Extracting strings with Floss.*

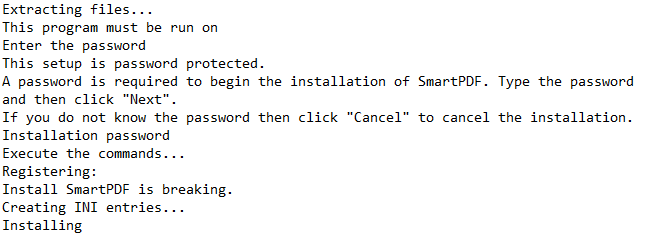
Opening the file ([**Appendix A**](#_Appendix_A_–)) immediately revealed that it is a Win32 executable, and that the software was built using Borland Delphi (a rapid application development tool for Windows). Scrolling down reveals that the application may have the capability to enumerate processes/drivers, obtain names of bases/files, as well as process memory information.



***Figure 2.2.4*** *– Win32 program developed with Borland Delphi.*

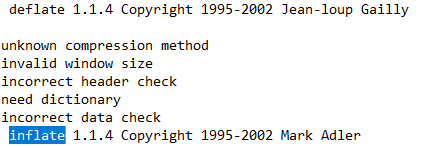
Further analysis showed multiple hardcoded registry locations Software\Windows\CurrentVersion\Uninstall, MapGroups, \Internet Explorer\Quick Launch. One of them indicated that it could launch Internet Explorer, indicating adware capabilities. Other strings indicated shutdown privileges, process/file manipulation, windows (running applications) enumeration, graphics creation, message translation, etc. Additionally, the application could adjust the privilege tokens and some features indicating spyware capabilities (keyboard type, default language, key state, cursor position, etc).

Scrolling down revealed the name of the payload that had to be extracted on the victim’s machine all of them being installed with the **/Verysilent** tag, indicating possible malicious behaviour. It might have been a modified version of the SmartPDF freeware installer as the name of the software is mentioned multiple times throughout the readable strings. The installer also appears to request the user’s password to “successfully install the software”, further indicating spyware functionalities. (**Figure 2.2.5**)



***Figure 2.2.5*** *– Asking the user for a password.*

The strings also indicated the usage of **costura** (software allowing the merge of assemblies as embedded resources) and inflation/deflation software possibly for compression and decompression (**Figure 2.2.6**)

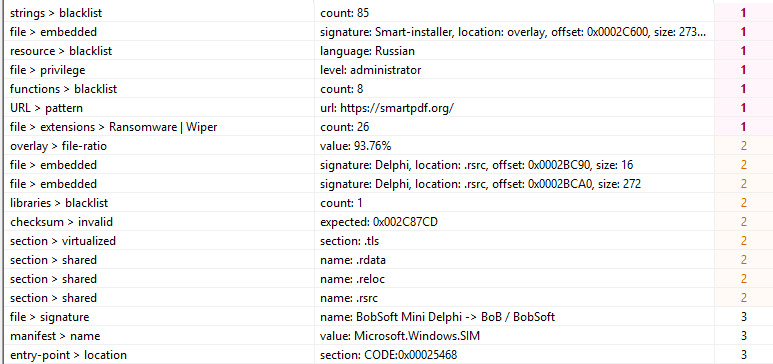


***Figure 2.2.6*** *– Deflate and inflate software.*

### **In-depth Inspection**

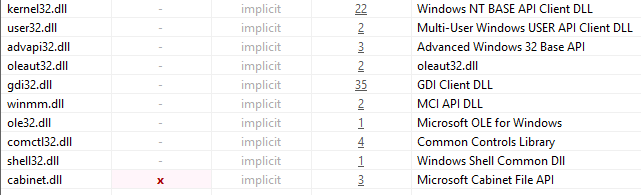
The analysis proceeded with two tools **PEStudio**. **PEStudio** is a tool speeding up the initial malware analysis process and makes it easier. The tool conducts a complete static analysis of the file and provides researchers with indicators, imports, libraries, and file entropy (randomness of data hinting at hidden/suspicious data).

The file header indicated that the initial compilation of the bundleware was in 1992. The tool then showed seven critical and eight serious indicators - blacklisted strings, country of origin, embedded files, privilege escalation blacklisted functions and extensions connected to known ransomware samples. Additionally, the file had an entropy value of 7.981 out of 8, showing that it is heavily obfuscated with a significant amount of hidden data. (**Figure 2.2.7**)



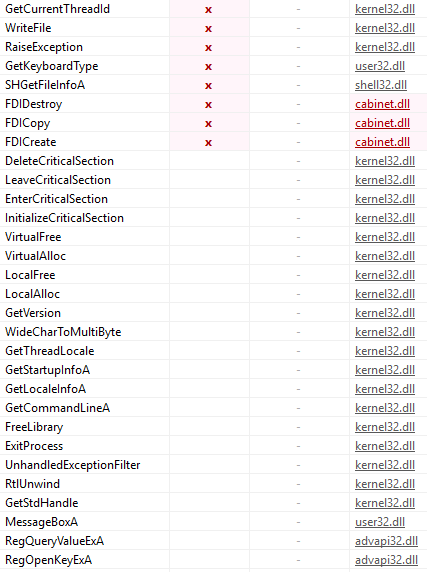
***Figure 2.2.7*** *– Malicious indicators.*

In terms of the libraries, the sample contained only one blacklisted library – cabinet.dll (library handling cabinet files with compressed code). The other libraries, despite not being blacklisted, could still be used for malicious activities. (**Figure 2.2.8**)



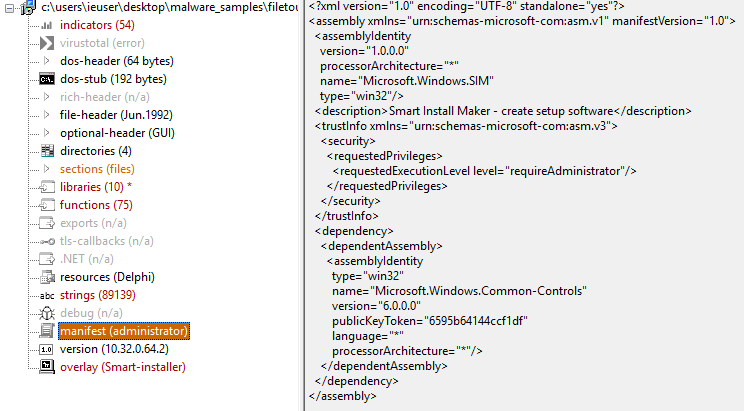
***Figure 2.2.8*** *– List of libraries.*

The tool discovered a total of eight blacklisted libraries – three from **kernel32.dll** (**GetCurrentThreadId**, **WriteFile**, **RaiseException**), one from **user32.dll** (**GetKeyboardType**), one from **shell32.dll** (**SHGetFileInfoA**) and three from **cabinet.dll** (**FDIDestroy**, **FDICopy**, **FDICreate**). As discussed in [**Section** **2.2.2 String Extraction**](#_String_Extraction), the malicious software uses the blacklisted and non-blacklisted functions to extract compressed data, alter the victim’s system and obtain data about their behaviour and credentials, and possibly deceive them or open ads in Internet Explorer through **MessageBoxA** and any of the graphical functions provided by **gdi32.dll**. (**Figure 2.2.9**)



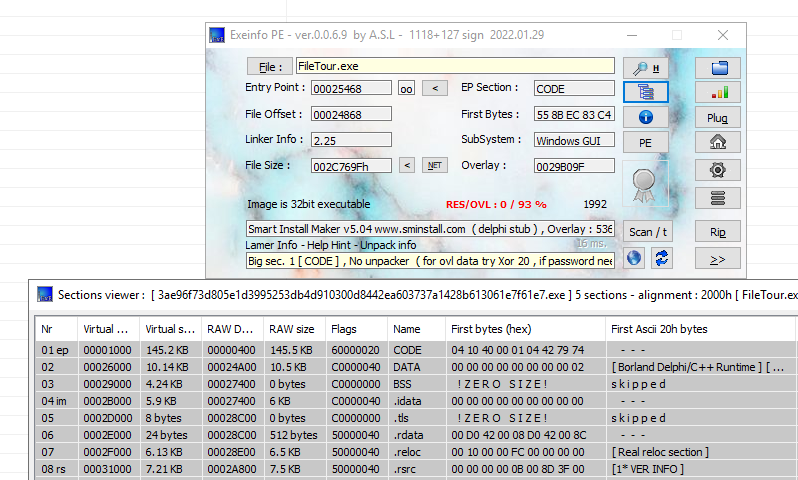
***Figure 2.2.8*** *– Partial list of functions utilised by the malware.*

In the end, the analyst identified a manifest embedded into the software that attempts to escalate the privileges of the process to the Administrator and an overlay with the **Smart Installer** signature, taking 93.76% of the entire executable. (**Figure 2.2.9**)



***Figure 2.2.9*** *– Manifest for privilege escalation.*

The analyst then checked whether the malware was packed using one of the commonly found packers such as UPX1 to evade detection. This was achieved with **Exeinfo PE**. Loading the file into the tool revealed that the EP Section (packer) was named **CODE** – one of the custom sections within the executable. This would make reverse engineering the sample even more difficult as the analyst would first need to understand how the packer works. (**Figure 2.2.10**)

******

***Figure 2.2.10*** *– FileTour using a custom packer within its CODE section.*

### **Dropped Files**

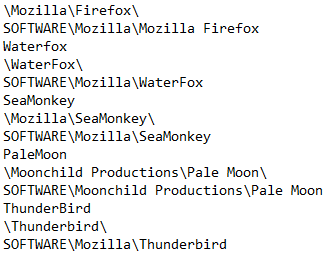
As mentioned in [**Section 2.2.2 String Extraction**](#_String_Extraction), the malware drops multiple other binary files after its execution. It also attempts to download three more files (two after the initial detonation and one more after executing LivelyScreenRec). The C&C servers used to deliver those files were shut down, preventing the analyst from downloading the samples and conducting any research on them. This will be further covered in the Dynamic Analysis section of the report.

#### **9840432e051a6fa1192594db02b80a4c1fd73456**

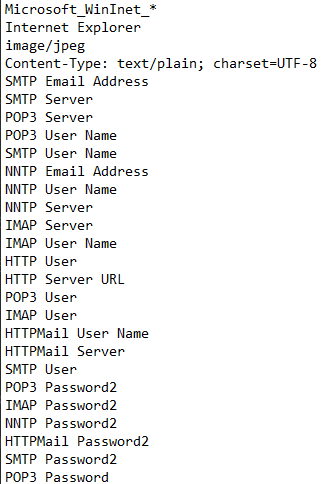
The hashes of the sample were checked in VirusTotal and multiple vendors identified it as a malicious file:

* SHA256: a45d807e4bee0a8fdb47ca4d899b0b44a0834bafc222eaaa589917d7714f05b3
* MD5: 93e0a54822a5609a66f51bd9bf60b071

The file is executed together with SmartPDF.exe. Analysing the strings revealed hard-coded registry locations connected with Mozilla products (**Figure 2.2.11**), as well as networking (**Figure 2.2.12**) and encryption functionalities (**Figure 2.2.13**). The encryption capability is further proven by the list of languages within the strings – something common for ransomware samples (i.e., WannaCry) to display the messages in the local language of the victim.



***Figure 2.2.11*** *– Mozilla related registries within the strings.*

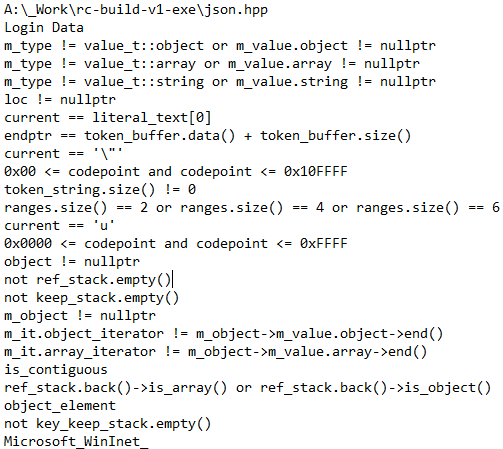
**

***Figure 2.2.12*** *– Network and credentials-related strings.*

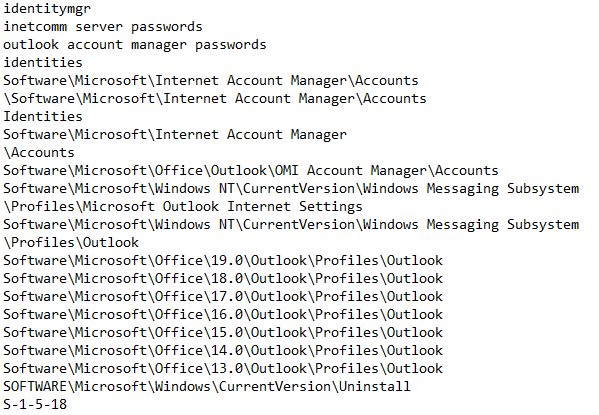
**

***Figure 2.2.13*** *– Partial list of languages within the strings.*

Additionally, partial snippets of code related to token manipulation and registries connected to the Microsoft Messaging Subsystem and Outlook were also identified with the SID for Local System (S-1-5-18 – the ID of System, a hidden member of the Administrators group). (**Figure 2.2.14** and **Figure 2.2.15**)



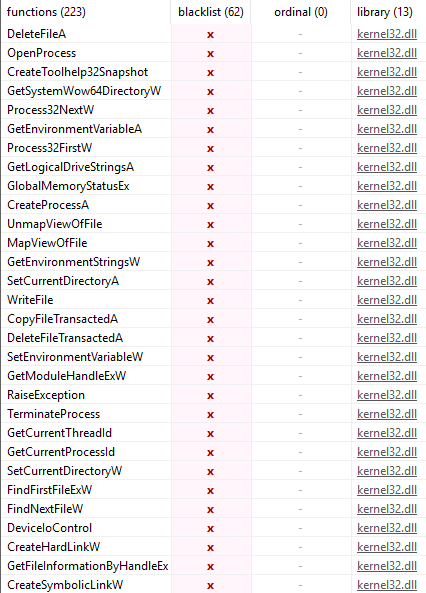
***Figure 2.2.14*** *– Code snippets related to token manipulation.*

**

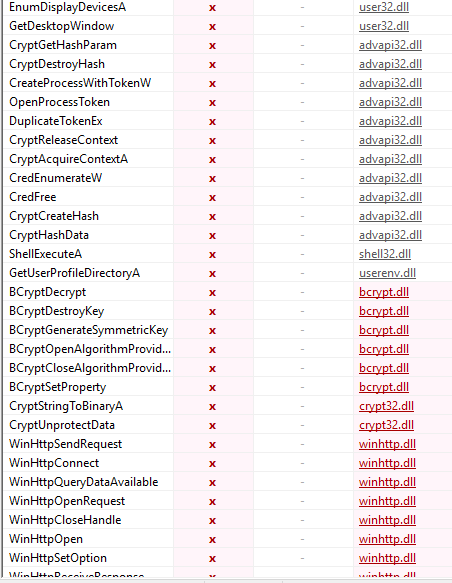
***Figure 2.2.15*** *– Strings related to the registry locations of the Messaging Subsystem and Outlook.*

Launching the sample in **PEStudio** revealed three critical indicators – blacklisted strings and functions, and file extensions connected to known Ransomware and Wiper samples, indicating that this specific file provided the ransomware capabilities of the sample. In terms of blacklisted libraries, a total of three were identified – **bcrypt.dll** and **crypt32.dll** (Cryptographic libraries) and **winhttp.dll** (HTTP connections).

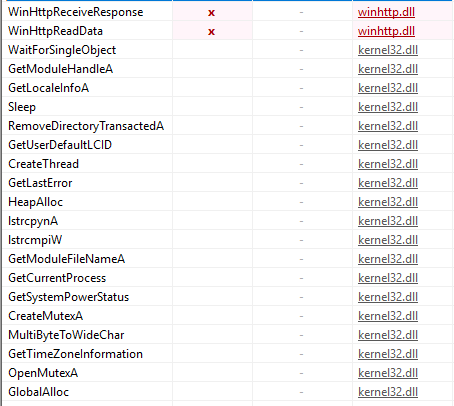
Regarding blacklisted functions, the tool identified a total of sixty-two related to process/file manipulation, exception raising, encryption and network communication (connect, read, query and request data/connections through HTTP). The software could also enumerate the victim’s displays, execute shells, and have access to the machine on a hardware level (**DeviceIOControl** – the same function is used in NotPetya to access the infected machine’s hard drive on a hardware level). The functions can be seen in **Figures 2.2.16, 2.2.17** and **2.2.18**.



***Figure 2.2.16*** *– Blacklisted functions part one.*



***Figure 2.2.17*** *– Blacklisted functions part two.*



***Figure 2.2.18*** *– Blacklisted functions part three.*

#### **lg**

The hashes of the sample were checked in VirusTotal and multiple vendors identified it as a malicious file:

* SHA256: 68d1b6dbfc303f1949267ce03ac2164ee9cda951231e72e6a5e39a44764ebbf2
* MD5: adfe31c40569ca5b0b403f0ba3f7b24c

Analysing lg revealed that the sample had multiple other executables embedded within it. Examining the strings showed a total of four executables with a wide variety of used functions covering Microsoft Root Certificates, anti-virus types, exception raising, file manipulation, environmental strings/variable acquisition, etc. Additionally, **Floss** successfully de-obfuscated strings within the file, revealing that the malware also uses an embedded SQLite database. The database may store information regarding the infected machine, as the sample also contains a multitude of OLE functions, allowing it to display information about other Windows applications. Furthermore, the de-obfuscated strings showed “RealVNC” within them – software allowing remote access to machines. It may be used as a backdoor to the infected machine, possibly to create a sophisticated botnet.

Examining the sample in **PEStudio** revealed a total of four critical indicators covering blacklisted strings, embedded files, blacklisted functions, and a URL (the SQLite copyright webpage). In terms of blacklisted functions were similar to the previous samples – acquisition of environmental variables and strings, process termination, acquisition of current thread ID and exception raising. (**Figure 2.2.19**)

Table

Description automatically generated

***Figure 2.2.19*** *– Blacklisted strings in lg.*

#### **LivelyScreenRec**

The hashes of the sample were checked in VirusTotal and multiple vendors identified it as a malicious file:

* SHA256: d986c5f06a66c89f8bdf03288177b2a5c9e9413c96c3d8b0878a49b431735654
* MD5: 53b01ccd65893036e6e73376605da1e2

Examining the sample revealed that it may be a modified version of the screen recorder application developed by ModuleArt and available on GitHub. Multiple links were found within the extracted strings – three for the GitHub page of the software and an additional one (hxxp://activityhike.com/files/sonia30.exe) from which the sample downloads a binary upon execution. (**Figure 2.2.20**)

A screenshot of a computer

Description automatically generated with medium confidence

***Figure 2.2.20*** *– Launching the executable.*

Loading it in PEStudio showed multiple critical indicators – blacklisted strings/functions and URL patterns. Checking the blacklisted functions confirmed that the software is modified to download and execute binaries over the Internet. The software could also register/unregister hotkeys, and acquire temporary file names and **System.IO.Stream**, allowing it to read/write data within memory. (**Figure 2.2.21**)

Table

Description automatically generated with medium confidence

***Figure 2.2.21*** *– Blacklisted functions in LivelyScreenRec.*

#### **Note866**

The hashes of the sample were checked in VirusTotal, and multiple vendors identified it as a malicious file:

* SHA256: de95d03777407422fac23d6c1f0740e131a0d38c5ef19aca742c7bcf1a994fd7
* MD5: 53b01ccd65893036e6e73376605da1e2

Examining the strings did not reveal much information about the sample’s possible functionalities as the only readable strings were four functions from the **kernel32** dynamic link library. The analyst loaded the file into PEStudio. Based on the results, they identified a total of six critical indicators regarding blacklisted strings, blacklisted resources (**Chinese-simplified**), suspicious binary section behaviour, library count and extensions connected to ransomware/wipers. (**Figure 2.2.22**)

Graphical user interface, application

Description automatically generated

***Figure 2.2.22*** *– Indicators in PEStudio.*

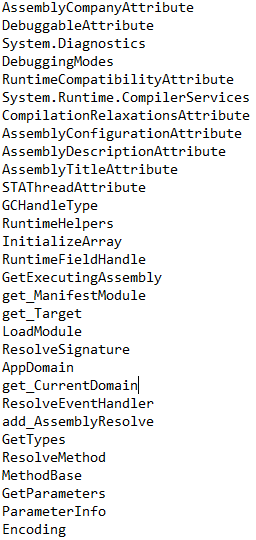
The binary also appeared to contain self-modifying code, possibly to bypass antivirus detection. As the static analysis did not reveal any useful information, the sample had to be executed and its behaviour closely monitored. This will be covered further in the dynamic analysis section where its data theft capabilities will be shown.

#### **PBrowFile15**

The hashes of the sample were checked in VirusTotal, and multiple vendors identified it as a malicious file:

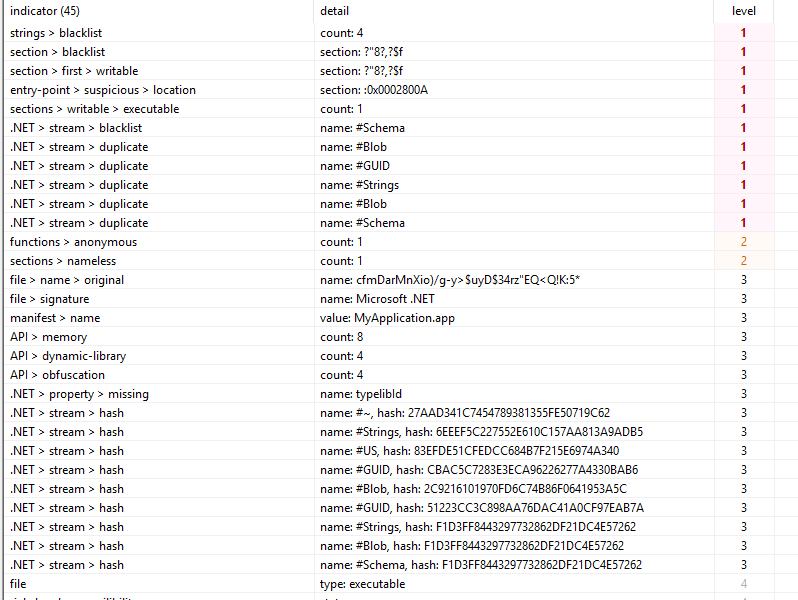
* SHA256: d50c43ec7c8dc7954302defe5f9cf729e9919231f031f34c953b76a0af81ac09
* MD5: 2cf1c9c0bf6f1cdc50c8a18e460c88ec

As the sample was heavily obfuscated, the analyst could not identify its specific purpose in an attack. It is possible that the file was used as a compiler as **System.Runtime.CompilerServices** was seen within the strings (a namespace for compilers used to specify attributes in the metadata which can affect the runtime behaviour). There are also functions regarding the COM threading model for the application, block copying and obtaining user credentials of the user who ran the malware (**get\_CurrentDomain**). Encoding functions could also be seen, specifically **Encoding** and **ToBase64String**. (**Figure 2.2.23**)



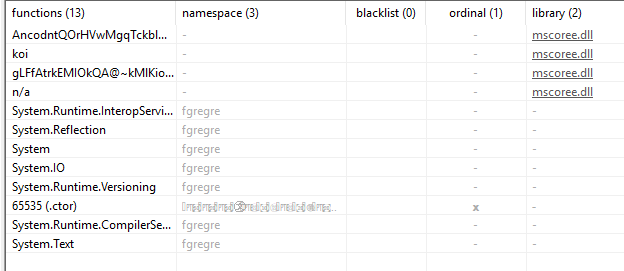
***Figure 2.2.23*** *– Identified functions within the strings.*

Examining the sample in **PEStudio** revealed a total of eleven critical indicators regarding blacklisted strings/sections, writable/nameless sections, .NET blacklisted names and suspicious entry locations. (**Figure 2.2.24**)



***Figure 2.2.24*** *– Indicators are shown in PEStudio.*

The blacklisted section appeared to also be self-modifying, possibly to bypass anti-virus detections. The tool could not identify any blacklisted functions, however, there were multiple allegedly from the mscoree.dll library but their names were altered. There was also a function in a custom namespace with an illegible name. (**Figure 2.2.25**) All of those signs point to malicious behaviour in the executable.



***Figure 2.2.25*** *– Functions list.*

#### **Downloaded payloads**

The downloaded payloads could not be examined by the analyst as the C&C servers were taken down and the malware could not successfully download them. The files were stored in the **Temp** directory within the **AppData** directory. It is possible that they would automatically delete themselves after execution and altering services/registries to ensure their persistence. According to the automated analysis by Any.run, the sample downloads and executes versions of Racoon and Redline (Anyrun, 2021)– two spyware software used for data theft.

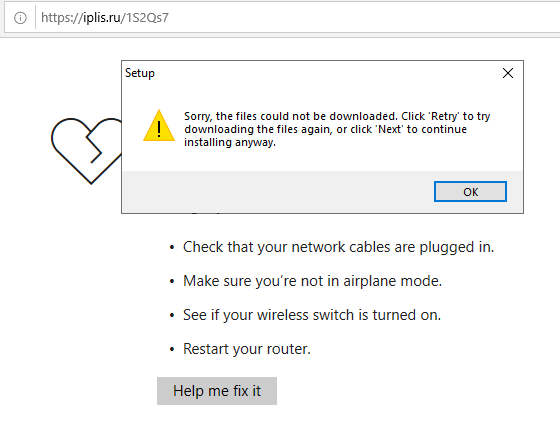
## **Dynamic Analysis**

Dynamic analysis of malware is achieved by detonating the sample in a safe environment (or surveying an already compromised environment) to see how it behaves on a local and network level. This may be dangerous if the safe environment is not properly set up as it may allow the malicious software to propagate to the physical machine, the user’s network, and possibly even other connected networks.

### **Detonation Symptoms**

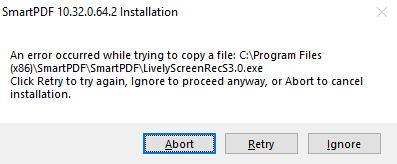
The dynamic analysis started with the initial detonation of the sample and inspection of the symptoms of the infected virtual machine. They attempted to detonate it both with and without administrator rights to see if there would be any differences. It appeared that the sample did not require access to an administrator account to execute itself.

The first visible symptom was Internet Explorer (or the default browser of the user) launching itself and trying to connect to a domain called **hxxps://iplis.ru/1S2QS7**. The analyst checked the domain in reverse look-up services (**https://who.is**) and identified that it was currently inactive. An alert also appeared from an application called **Setup**, claiming that the files could not be downloaded. (**Figure 2.3.1**)



***Figure 2.3.1*** *– Connecting to the domain and displaying the alert.*

Clicking on **OK** and executing the binary again showed a different message revealing that **9840432e051a6fa1192594db02b80a4c1fd73456** and **LivelyScreenRecS3.0.exe** could be successfully copied. It is possible that their capabilities (encryption and remote access) were not carried out due to missing parts of the files or decryptors. Some samples contact domains or download files with decryption keys which are then used to decrypt the contents of other files. This is done to bypass initial anti-virus detections. (**Figure 2.3.2**) Ignoring both actions attempted to connect to the beforementioned link again and redownload the files. This was done in a host-only environment without connection to the internet simulation (**Inetsim**).



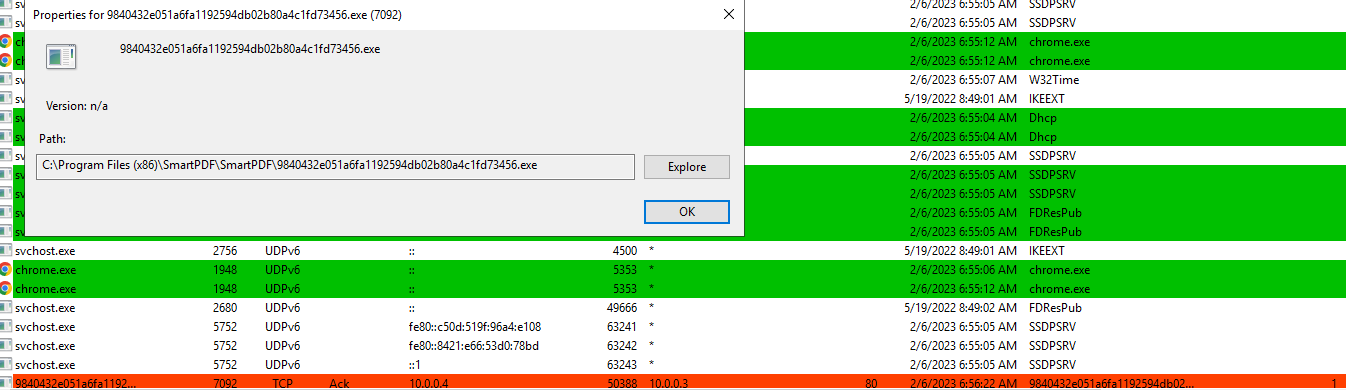
***Figure 2.3.2*** *– SmartPDF Installation indicating that the files could not be copied.*

The sample did not show any other visible symptoms, which may be due to it having multiple stages. The second stage payload could not be delivered which halted the detonation to a specific state that did not apparent harm to the infected virtual machine.

### **Network Symptoms**

To examine the network symptoms, the analyst used tools such as **TCPView** and **Wireshark** while the machine was connected to a second one (**Remnux**) that had **Inetsim** running in the background. **Inetsim** is a tool which simulates an internet connection and a DHCP server, allowing it to capture all traffic in a host-only environment. **TCPView** allows the analyst to examine all connections going from and to the machine, while **Wireshark** is used to capture and examine the traffic towards **Inetsim**.

Opening **TCPView** revealed an attempted connection towards the machine hosting **Inetsim** on port **80** (**HTTP**) by **9840432e051a6fa1192594db02b80a4c1fd73456**. This further enhances the statement that the executable is attempting to download a second-stage payload or decryption keys. The connection kept showing and soon turning red, indicating that it was automatically attempted multiple times and dropped when it was unsuccessful. (**Figure 2.3.3**)



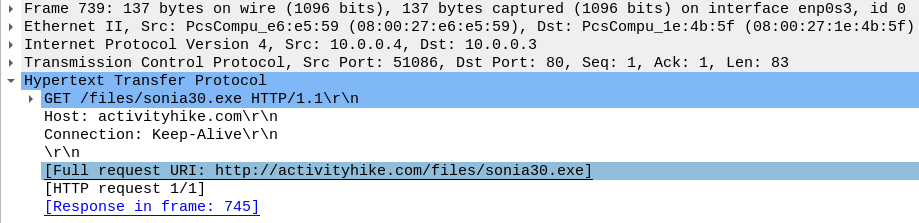
***Figure 2.3.3*** *– Attempted HTTP connection is shown in TCPView.*

Afterwards, the analyst examined the network traffic after it was captured with Wireshark. Based on the connection shown in **Figure 2.3.3**, the analyst filtered only the HTTP (Web) traffic. They immediately noticed **GET** requests about the country and IP of the machine. (**Figure 2.3.4**)



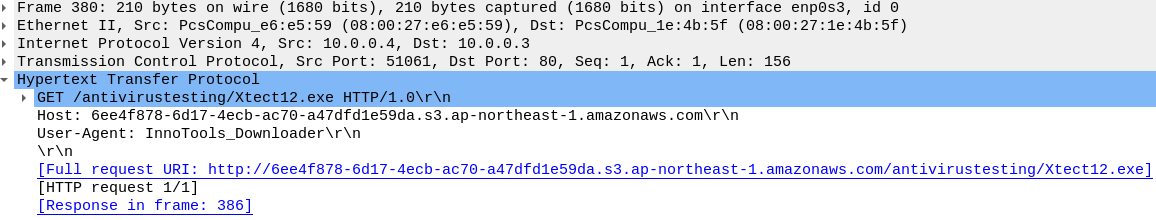
***Figure 2.3.4*** *– Attempting to obtain the country and IP of the infected machine.*

Further, inspection showed the malware attempting to download to executables – **sonia30.exe** and **Xtect12.exe**. The former file was downloaded by **LivelyScreenRecS3.0.exe** while the latter was downloaded by **9840432e051a6fa1192594db02b80a4c1fd73456**. The analyst could not identify what **sonia30.exe** will be used for, as the domain also did not give any useful hints (**Figure 2.3.5**)

****

***Figure 2.3.5*** *– Payload being downloaded from* ***hxxp://activityhike.com/files***

The latter file appeared to be an anti-virus tester downloaded from an **AWS** cloud server. (**Figure 2.3.6**) Both domains were shut down at the time of the testing and the analyst was unable to obtain the payloads for further analysis.



***Figure 2.3.6*** *– AV testing payload being downloaded from the AWS cloud server.*

Based on the obtained information, the analyst could assume that **9840432e051a6fa1192594db02b80a4c1fd73456** first runs the AV test and then executes itself if the testing was successful. It was possible that it did not execute itself as the payload could not be delivered and the testing did not take place.

Alongside **iplis.ru**, the sample appeared to also connect to the following domains:

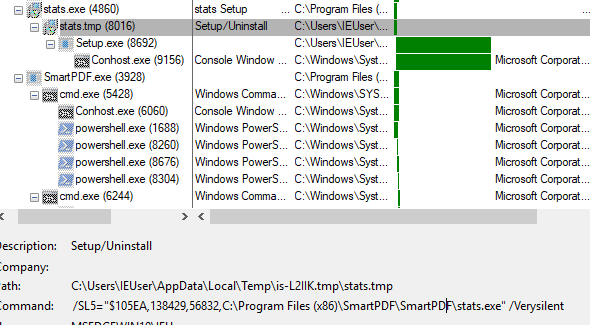
* delete.in
* theonlinesportsgroup.net
* ipinfo.io
* ipqualityscore.com
* iplogger.org
* sanctam.net
* github.com

Most of them appeared to be websites for IP address testing and logging. **Sanctam.net** was inactive so no information could be obtained. GitHub is a website for storing and managing software projects. The sample might have scanned it to check whether a connection can be made and would have then downloaded another payload from it.

### **Host Symptoms**

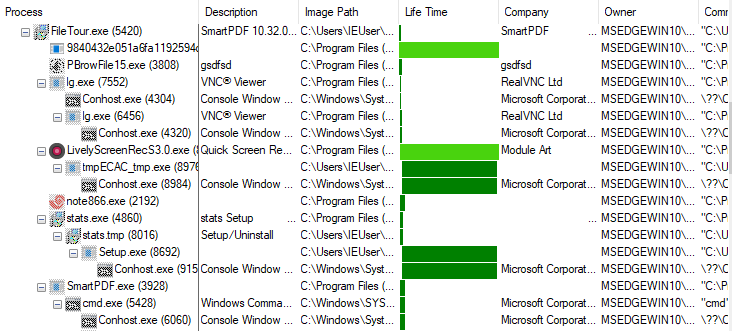
To effectively analyse the host symptoms, the researcher used a tool called **Procmon**. The tool allows users to monitor the processes on their machine with the option to extensively filter them to their needs. It also allows them to display them in a tree model, showing which applications have their subprocesses.

Opening the tree view showed multiple applications related to the malware. One of them was **stats.exe**. It is executed from within a temporary directory with a **/Verysilent** flag (Triage, 2021). This may install all subsequently dropped binaries. (**Figure 2.3.7**) In the command, it also has flags specific for **Inno Setup** – free installation software. The reason why it was in a temporary directory is how Inno Setup functions. The installer has two processes, the primary one is hidden, extracting a child installer to a temporary folder, and elevating to administrator privileges if required. As the child process needs to communicate with the primary one, it requires said flags/arguments for inter-process communication.

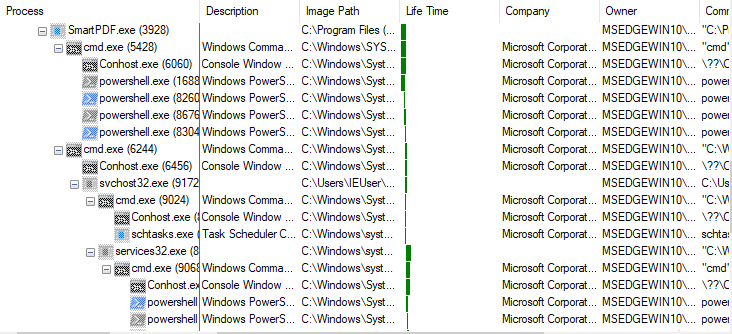


***Figure 2.3.7*** *– Silently installing payloads.*

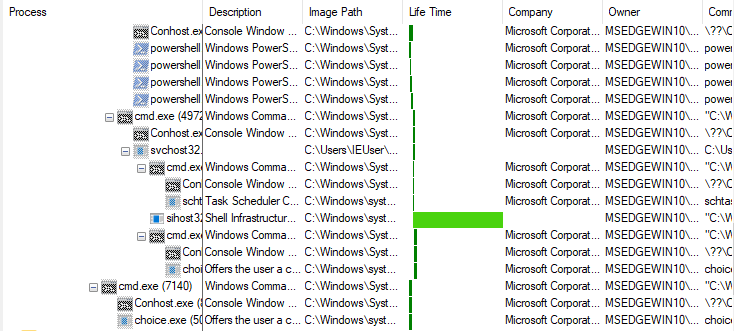
The process tree showed that all the subsequently delivered payloads were executed, all of them with a different lifetime. All of them can be seen in **Figures 2.3.8**, **2.3.9**, and **2.3.10**.



***Figure 2.3.8*** *– Part one of the process tree.*

**

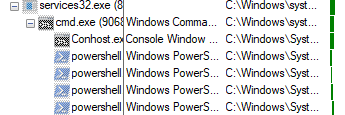
***Figure 2.3.9*** *– Part two of the process tree.*

**

***Figure 2.3.10*** *– Part three of the process list.*

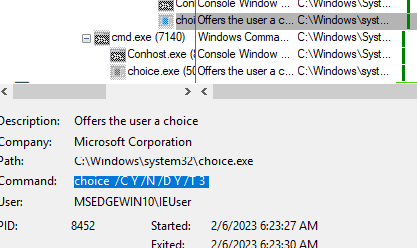
The multiple PowerShell processes seen under **SmartPDF.exe** and **services32.exe** indicate possible malicious activity. (**Figure 2.3.11**) Examining them showed scripts for detection evasion – excluding specific directories from being scanned by Windows Defender. This was achieved with the following command:

* "cmd" /c powershell -Command Add-MpPreference -ExclusionPath '%UserProfile%' & PowerShell -Command Add-MpPreference -ExclusionPath '%AppData%' & PowerShell -Command Add-MpPreference -ExclusionPath '%Temp%' & PowerShell -Command Add-MpPreference -ExclusionPath ‘%SystemRoot%’ & exit



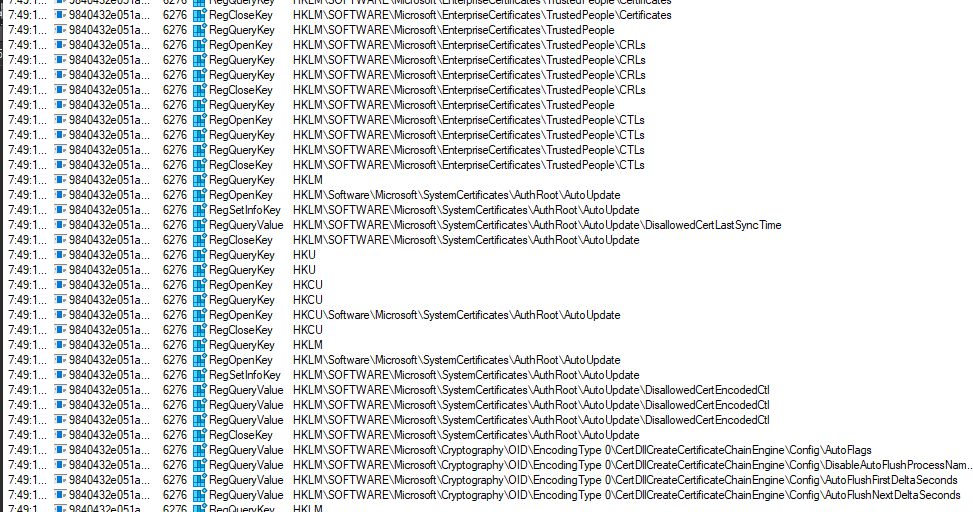
***Figure 2.3.11*** *– Excluding specific directories from being scanned by Windows Defender.*

**Choice.exe** was also launched with some of the processes with the following arguments: **/C Y /N /D Y /T 3**. This is used to delete the process after it finishes its execution – possibly another method of evading detection and/or distracting analysts. (**Figure 2.3.12**)

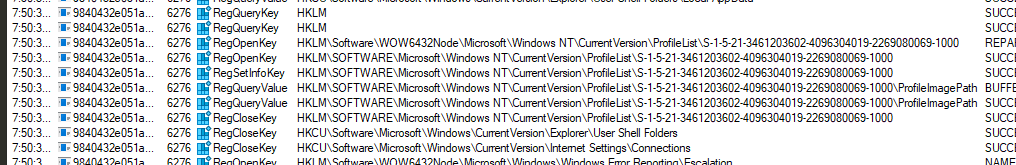


***Figure 2.3.12*** *– Deleting application on exit.*

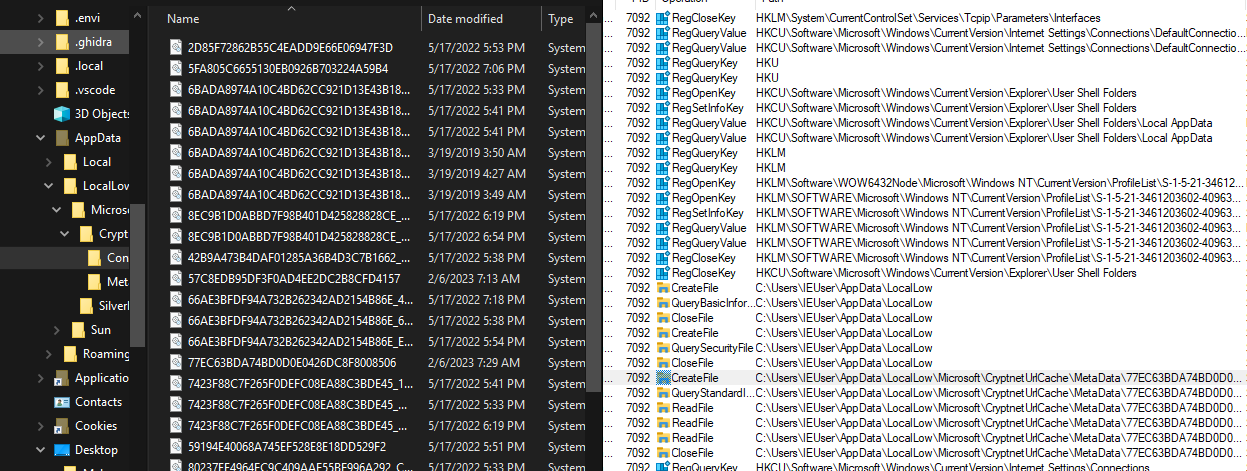
As the analyst could not get more information about the initial sample, they moved on to the delivered payloads. **9840432e051a6fa1192594db02b80a4c1fd73456.exe** appeared to query various registry keys regarding system and enterprise windows certificates. (**Figure 2.3.13**) Said certificates are used to authenticate the identity of the owner of the machine. It also enumerated the available profiles through the **Profile List** registry keys. (**Figure 2.3.14**) It might have used this as a method of propagation across organisations. It also accessed and read files within **CryptnetUrlCache** – a directory storing various files acquired from the internet even without the knowledge of the user. (**Figure 2.3.15**) This was possibly done to find the user’s browsing behaviour and data (SSL certificates, cookies, etc.) It was also possible that the cryptography functions found in the static analysis were used to decrypt those files and not to encrypt the user files and request a ransom payment.



***Figure 2.3.13*** *– Enumerating System and Enterprise certificates.*

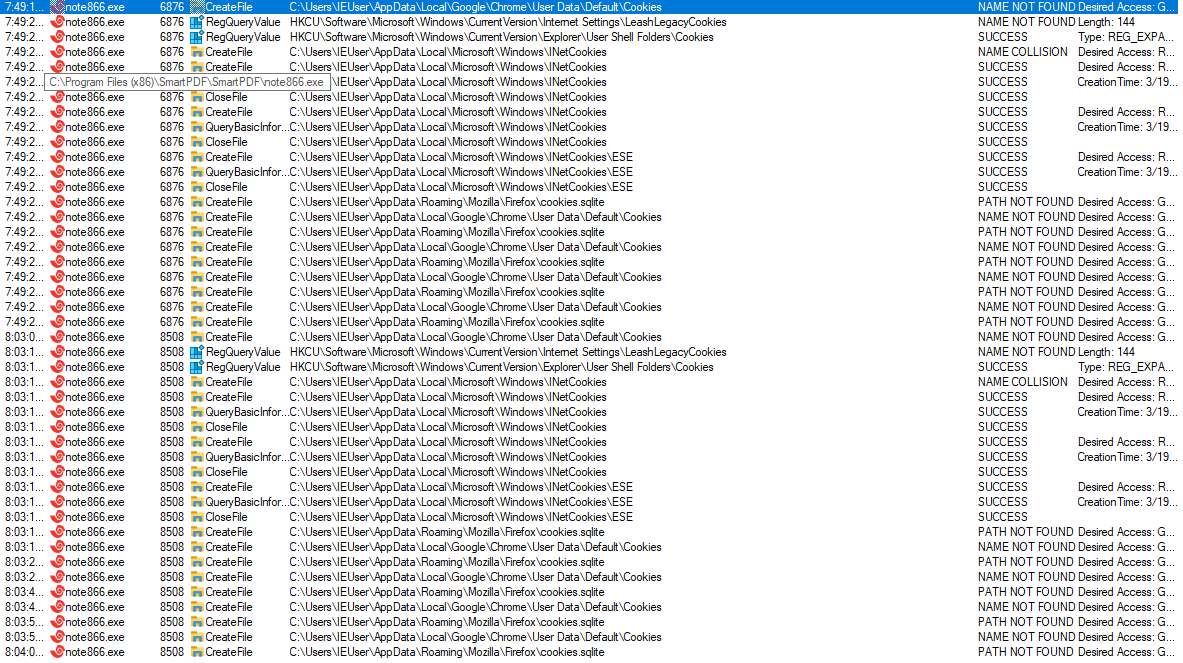
**

***Figure 2.3.14*** *– Enumerating profile lists.*

**

***Figure 2.3.15*** *– Attempting to read cached data in CryptnetUrlCache.*

The other dropped payload that showed obvious signs of data tampering was **Note866**.exe. By examining its processes, the analyst identified that it read the web browser cookies of all available browsers on the system – Firefox, Chrome, and Internet Explorer. Stolen cookies could be used to compromise user accounts or simply monitor their online behaviour for ad catering. (**Figure 2.3.16**) The other downloaded payloads mentioned in [**Section** **2.2.4.6 Downloaded Payloads**](#_Downloaded_payloads)did not show any obvious malicious activity possibly due to missing additional payloads or improper detonation conditions.

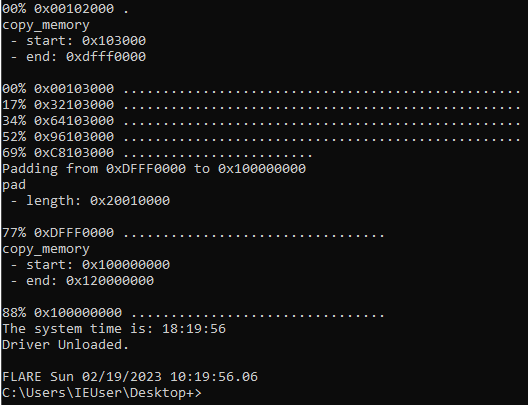


***Figure 2.3.16*** *– Examining browser cookies.*

### **Memory Analysis**

Memory analysis can be a tedious task, but simple checks may reveal malicious behaviour in different processes. To analyse the memory of the infected virtual machine, the analyst used **Volatility 3.0** and **WinPMem** to dump the memory.

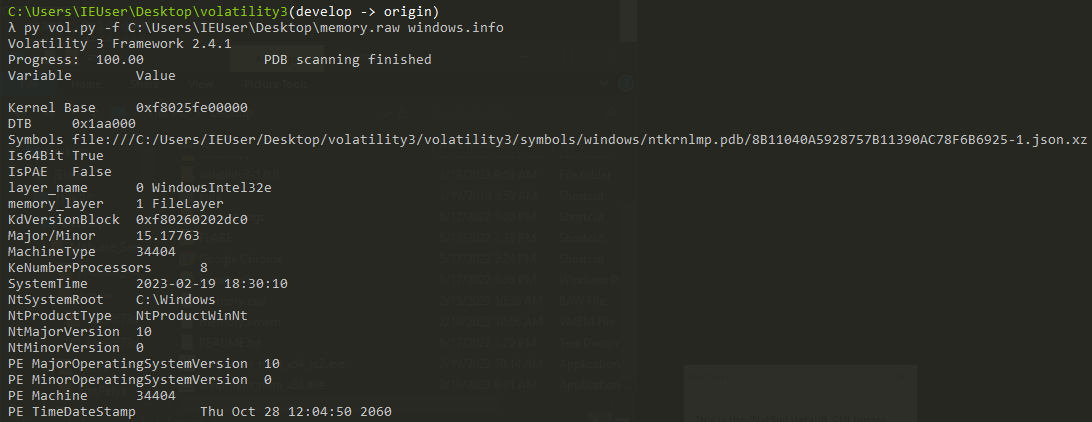
Dumping the memory could be easily achieved with the beforementioned tool. The analyst ran the following command in a **CMD** window to acquire the memory data in raw format: **winpmem\_mini\_x86.exe memory.raw**. This command wrote the contents of the RAM in a file called **memory.raw**. (**Figure 2.3.17**)

****

***Figure 2.3.17*** *– Dumping the memory using WinPMem.*

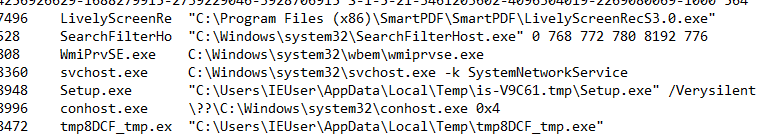
Afterwards, the researcher analysed different data from the raw memory dump. (**Figure 2.3.18**) This was achieved with various plugins, in this case specifically ([**Appendix B**](#_Appendix_B_–)):

* windows.info – basic information about the windows machine
* windows.cmdline – executed cmd commands
* windows.envars – list of environmental variables
* windows.malfind – looking for applications/services with injected malicious code
* windows.privileges – display the privileges for all dumped processes
* windows.pslist – lists the visible processes
* windows.certificates – lists the user certificates on the machine



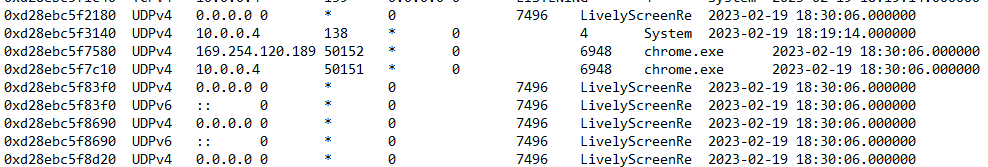
***Figure 2.3.18*** *– Running the windows.info plugin on the memory dump.*

Examining the command list showed that some of the commands were missing and were marked as possible exited processes. However, some of the PowerShell commands connected to the malicious software remained. (**Figure 2.3.19**)



***Figure 2.3.19*** *– Commands list.*

The connections plugin revealed that **LivelyScreenRecorder** attempted/expected connections from port 7496. The foreign address was set to 0, indicating that there is currently no specific remote address, but it may accept connections from any address which tries to connect to the machine. (**Figure 2.3.20**)

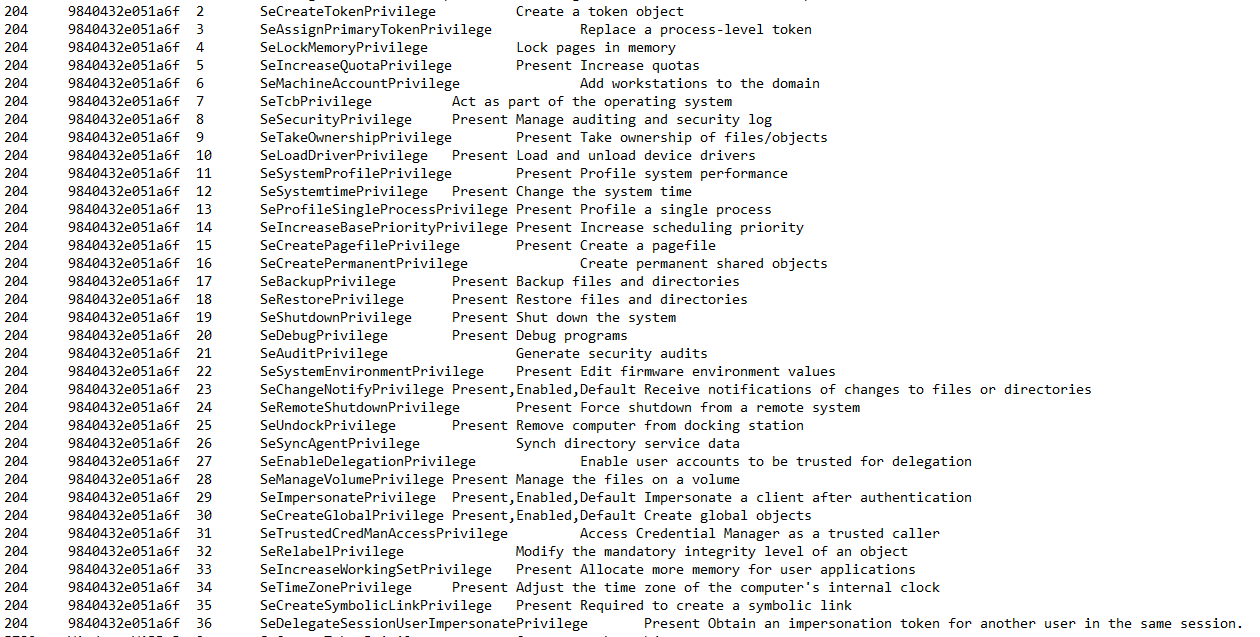
**

***Figure 2.3.20*** *– LivelyScreenRecorder connections.*

The environmental variables did not show anything of importance, so the analyst moved to the available kernel modules. Kernel modules are a piece of software with direct access to the system’s hardware. The windows.modules plugin also did not return any suspicious data which could be connected to the sample.

Using the windows. privileges plugin revealed malicious behaviour from the **9840432e051a6fa1192594db02b80a4c1fd73456** executable. It had multiple privileges which could be used for malicious purposes (**Figure 2.3.21**):

* System shutdown
* Program debugging
* Load/unload device drivers
* Change system time
* Manage security logs
* Act as a part of the OS
* Process-level tokens
* Lock pages in memory
* Manage files on volumes
* Obtain an impersonation token for another user in the same session
* Client impersonation after authentication
* Memory allocation for user—space applications

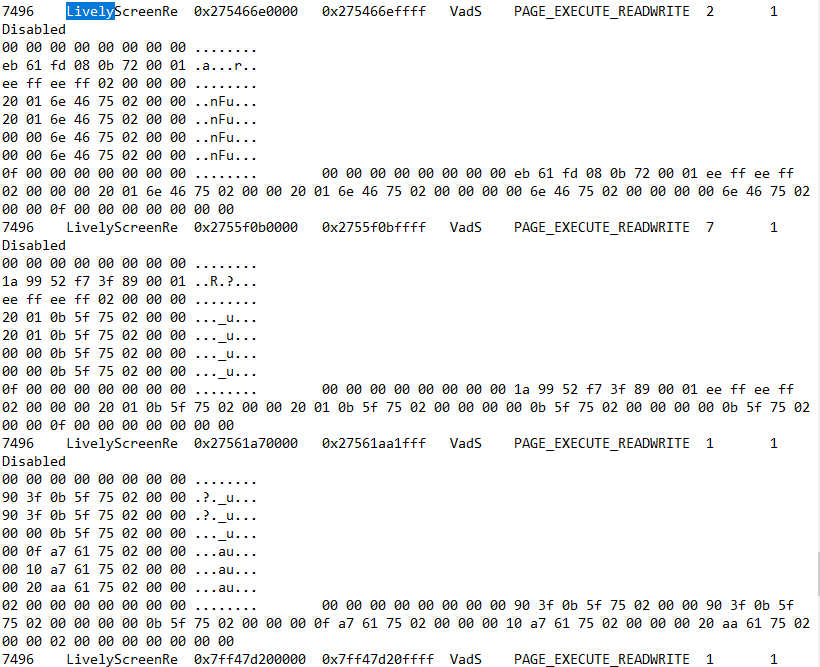


***Figure 2.3.21*** *– Process privileges for* **9840432e051a6fa1192594db02b80a4c1fd73456**.

This effectively shows that the payload has complete access to the resources of the infected machine, as well as impersonates all users in the session and any authenticated clients.

Finally, the analyst used the windows.malfind plugin to look for possible injected malicious code within applications. It must be noted that the plugin may show false-positive results but it can still be a great indication of possible malicious processes. The analyst noticed multiple locations in **svchost.exe** starting with **MZ** (**4D5A**) – this is the starting point of a Windows DOS executable, indicating that the process may have injected payloads within it. Further indications are the **PAGE\_EXECUTE\_READWRITE** permissions that would let an attacker read and write code to those memory addresses. However, it is normal for svchost.exe to run multiple other services. An easy way to identify whether there are malicious applications is by examining the PIDs in the command line dump. If the svchost.exe process is called from **%systemroot%/system32/svchost.exe**, then the processes are most likely benign. Malware authors attempt to blend their processes by giving them similar names – i.e., **scvhost** or **svch0st** would indicate malicious behaviour.

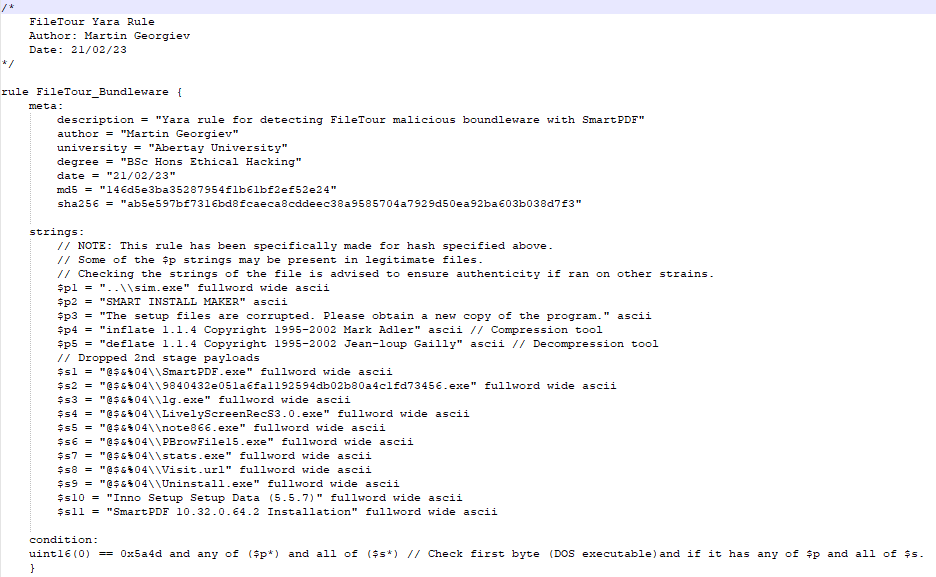
One of the processes in malfind stood apart – **LivelyScreenRec**. There were multiple instances (five) of the process, with the locations having **PAGE\_EXECUTE\_READWRITE** permissions, further solidifying that a legitimate binary may have been exploited to execute a malicious payload on start-up. (**Figure 2.3.22**)



**Figure 2.3.22** – Instances of LivelyScreenRecorder in malfind.

## **Yara Rule**

The Yara rule was created with notable data from the analysed strings. This is the easiest method to create simple Yara rules which can detect the malware – file names, specific commands and other strings related to this specific malware. (**Figure 2.4.1**) The Yara rule can be found in [**Appendix C**](#_Appendix_C_–).



***Figure 2.4.1*** *– FileTour Yara Rule.*

# **Results**

## **Static Analysis**

The Static Analysis allowed the tester to acquire detailed intel regarding the functionality of the malware. The multitude of tools made procedure faster, yet efficient. The analyst successfully obtained the hashes for the main executable and all available second-stage payloads. They also examined the strings of each sample. Combining this with PEStudio, they identified a lot of information about the capabilities of each payload and how they may affect a potentially infected system.

## **Dynamic Analysis**

Regarding the dynamic analysis, the investigator obtained further information about the infection. The network-based tests showed that it contacts multiple domains for partial system enumeration (IP and country) as well as some domains for downloading third-stage payloads. The C&C servers that were supposed to deliver the payloads were taken down, which halted the macro execution of the sample – it only executed on a micro-scale, enumerating the local system, and obtaining valuable data that could not be sent anywhere due to the lack of a destination. It had no embedded propagation capabilities, hinting that social engineering would be required to infect a machine (may it be through pirated software, fake freeware installers or email attachments).

On a micro-level, the malware attempted to steal data regarding the user’s browsing behaviour (cookies, certificates, etc.) as well as steal their user identity and the identity of any other user in the same session by impersonating their tokens. It could also manage security logs and audits, possibly to detect evasion and confuse analysts.

# **Discussion**

## **General Discussions**

Analysing the sample revealed that it could be capable to do more than show interruptive ads. The results showed that it also attempts to steal a lot of data about the infected user and impersonate them and every other user sharing the same session. A lack of propagation capabilities was expected based on the functionalities of other samples using the FileTour tag – mainly propagating as false pirated software, freeware and in phishing emails. It also uses legitimate software that was exploited to launch malicious code upon its execution.

As advanced static and dynamic analyses were not conducted and some of the payloads were missing because of inactive C&C servers, the analyst could not identify the entirety of its capabilities and how it could affect the machine. Many of its functions may be hidden within the two missing executables. Nevertheless, the obtained information is proof that the malware could be more harmful to the victim than displaying a series of ads.

## **Countermeasures**

### **Pre-infection Countermeasures**

The most effective way to protect a system from infection would be before it becomes infected. As some of the modules contain cryptography functions, a fully working sample could encrypt the victim’s files. This, in most cases, would not allow them to retrieve their files.

#### **Frequent Security Updates**

One of the reasons why malware is successful is the lack of security patches or users refusing to apply the newest updates to their operating systems and/or anti-virus applications. Additionally, using pirated versions may lack the newer patches and/or contain other vulnerabilities – or even malware as can be seen from the analysis of FileTour. Keeping your system and anti-virus software up to date would ensure that publicly known vulnerabilities could not be exploited, and the AV may have updated signature databases to detect the sample.

#### **Distinguishing Spam**

As the malware is primarily distributed through social engineering, users must be able to distinguish spam emails from real ones. This also applies to legitimate and fake websites and/or files. Users should not open any links or execute files unless they know the sender and the nature of the link or file. Additionally, users should look for bad grammar, fearmongering, rushed actions or similar addresses to legitimate ones.

#### **Blacklisting Unknown Applications and Anti-Virus Software**

System administrators could put restrictions on users by blacklisting unknown software. This way they would not be able to execute suspicious applications and provide the system/network with damage control to prevent any harm. It could be achieved with Anti-Virus software and integrated browser protection.

An updated Anti-Virus software could be used to perform system scans or simply scan newly generated files. Some may even prevent the malicious software from executing itself if they recognise specific code patterns or behaviours.

#### **File Scanners and IDS**

Intrusion Detection Systems will alert security analysts if they detect any suspicious behaviour – phishing emails, specific signatures, etc. Some of them can also be combined with file scanners such as Strelka for greater detection accuracy. This way the internal SOC team could notice the threat before it causes any harm to the system.

### **Post-infection Countermeasures**

#### **Data Backups**

As some of the payloads contained cryptography-related libraries and functions, it would be beneficial to keep data backups. With such, the company could wipe the infected drive and simply replace it with the information they have stored elsewhere. It is also advised to keep such data in physical storage if possible or in locations which are not directly connected to the network of the infected machine as some malware could propagate to it and destroy the backups.

#### **Ransom Payments**

Ransom payments should **NOT** be considered even in dire situations. In the case of encrypted files, the adversary may attempt to fearmonger the victim by threatening them to publicly post their data or delete it. Paying the ransom does not guarantee that the data can be recovered as the attacker may send a fake decryption key or they may not send one at all.

#### **Find the Infected Freeware**

Such malware usually depends on the freeware remaining on the system. One of the best ways to deal with such infections would be to identify the malicious software and remove it from the system. The more in-depth analysis could also reveal altered registry keys and/or services used for persistence. Restoring them to their initial state would be required to fully remove the infection.

# **Conclusion**

## **Conclusion**

FileTour may be marked as adware, but this may not always be the case. The bundleware is quite complex, implementing multiple techniques to confuse both the infected user and security analysts during their analysis of the infection. Being bundleware also allows it to deliver a wide variety of payloads, some of which are even embedded in vulnerable freeware.

This specific sample contained numerous functions connected to encryption, data theft and user impersonation. Depending on the capabilities of the two undiscovered binaries, the sample may be sophisticated spyware for stealing the users’ data to cater for their ads. It may as well attempt to create a large botnet similar to Stantinko based on the “RealVNC” string in one of the samples. The best way a user could protect themselves from a potential infection would be by not downloading and launching any suspicious applications and illegally pirated software.

## **Future Work**

As the strain has a wide variety of samples with different capabilities, the researcher would analyse multiple executables to compare their functionalities and behaviour. With more time, they would also attempt to obtain the two additional binary files and then reverse engineer them to obtain more information about their complete lifecycle. This would also show what steps it takes until it fully executes itself, allowing the tester to debug it and attempt to bypass any possible kill switches.

# **References**

ytsif (2014). TheZoo. [online] Available at: https://github.com/ytisf/theZoo [Accessed 15 Oct. 2022].

VirusTotal (2004 – Present Day). VirusTotal. [online] Available at: https://www.virustotal.com/gui/home/upload [Accessed 15 Nov. 2022].

Ballenthin, W. (2016). Floss. [online] Available at: https://github.com/mandiant/flare-floss. [Accessed 20 Nov. 2022].

Russinovich, M. (2021). Strings v2.54. [online] Available at: https://learn.microsoft.com/en-us/sysinternals/downloads/strings [Accessed 20 Nov. 2022].

ASL (2023). ExeInfoPE. [online] Available at: http://www.exeinfo.byethost18.com/?i=1 [Accessed 24 Nov. 2022].

Hungenberg, T and Eckert, M. (2007). INetSim. [online] Available at: https://www.inetsim.org/. [Accessed 27 Nov. 2022].

Russinovich, M. (2022). TCPView v4.17. [online] Available at: https://docs.microsoft.com/en-us/sysinternals/downloads/tcpview. [Accessed 27 Nov. 2022].

Wireshark. (1997 - Present Day). Wireshark Documentation. [online] Available at: https://www.wireshark.org/docs/. [Accessed 27 Nov. 2022].

Russinovich, M. (2022). Process Monitor v3.89. [online] Available at: https://docs.microsoft.com/en-us/sysinternals/downloads/procmon. [Accessed 28 Nov. 2022].

Fox, N. (2021). PeStudio Overview: Setup, Tutorial and Tips. [online] Available at: https://www.varonis.com/blog/pestudio. [Accessed 28 Nov. 2022].

Volatility Foundation (2020). Volatility 3.0. [online] Available at: https://www.volatilityfoundation.org/3 [Accessed 15 Dec. 2022].

Cohen, M., et al. (2019). WinPmem. [online] Available at: https://github.com/Velocidex/WinPmem [Accessed 15 Dec. 2022].

ESET (2017). Stantinko Teddy Bear Surfing Out of Sight. [online] Available at: https://www.welivesecurity.com/wp-content/uploads/2017/07/Stantinko.pdf [Accessed 14 Jan. 2023].

Triage (2021). FileTour Behavioural Analysis. [online] Available at: https://tria.ge/210905-jspx5acfaq/behavioral2 [Accessed 19 Jan. 2023].

Anyrun (2021). FileTour Analysis. [online] Available at: https://any.run/report/ff2fba623a5fef5ad2ab852079c88fbe33d12e48cfb0a06c90390d4a19270d2c/c157b4ab-5081-49c2-b41a-561066bbc555 [Accessed 20 Jan. 2023].

Stamus Labs (2022). Threat Detection Update. [online] Available at: https://www.stamus-networks.com/stamus-labs/detection-update-2022-05-31 [Accessed 8 Feb. 2023].

Netenrich (2021). FileTour. [online] Available at: https://know.netenrich.com/threatintel/malware/FileTour [Accessed 8 Feb. 2023].

AbuseCH (2020). MalwareBazaar. [online] Available at: https://bazaar.abuse.ch/ [Accessed 24 Mar. 2023].

VirusTotal (2013). Yara - The pattern matching Swiss knife for malware researchers (and everyone else). [online] Available at: https://virustotal.github.io/yara/ [Accessed 29 Mar. 2023].

# **Appendices**

## **Appendix A – Strings**

*Note: The output was truncated to remove a lot of illegible data.*

As the size of the strings remained substantial, they were not included in the report. They can be found in the folders of each binary in .txt files.

## **Appendix B – Memory Forensics**

### **Appendix B1 – Command Line**

3300 cmd.exe Required memory at 0x2d5c261020 is not valid (process exited?)

6312 cmd.exe Required memory at 0x7ecdf85020 is not valid (process exited?)

5376 cmd.exe Required memory at 0x3245807020 is not valid (process exited?)

3772 git.exe Required memory at 0x259515f020 is not valid (process exited?)

7432 cmd.exe Required memory at 0x5b275ad020 is not valid (process exited?)

1404 git.exe Required memory at 0x6e5267020 is not valid (process exited?)

7276 cmd.exe Required memory at 0xbe8d5c7020 is not valid (process exited?)

7956 git.exe Required memory at 0x89fb84c020 is not valid (process exited?)

5692 cmd.exe Required memory at 0xf29b2ff020 is not valid (process exited?)

1864 git.exe Required memory at 0xaad0ba1020 is not valid (process exited?)

7100 cmd.exe Required memory at 0x684ba62020 is not valid (process exited?)

7748 git.exe Required memory at 0xa0366d020 is not valid (process exited?)

828 notepad++.exe Required memory at 0x954a2e4020 is not valid (process exited?)

7304 SearchProtocol "C:\Windows\system32\SearchProtocolHost.exe" Global\UsGthrFltPipeMssGthrPipe82\_ Global\UsGthrCtrlFltPipeMssGthrPipe82 1 -2147483646 "Software\Microsoft\Windows Search" "Mozilla/4.0 (compatible; MSIE 6.0; Windows NT; MS Search 4.0 Robot)" "C:\ProgramData\Microsoft\Search\Data\Temp\usgthrsvc" "DownLevelDaemon"

7996 chrome.exe "C:\Program Files\Google\Chrome\Application\chrome.exe" --type=renderer --disable-gpu-compositing --lang=en-US --device-scale-factor=1 --num-raster-threads=4 --enable-main-frame-before-activation --renderer-client-id=52 --time-ticks-at-unix-epoch=-1676791729701543 --launch-time-ticks=38755753857 --mojo-platform-channel-handle=4788 --field-trial-handle=1976,i,13851431908316038218,5585174308964132067,131072 /prefetch:1

3244 cmd.exe Required memory at 0xca3b5d3020 is not valid (process exited?)

3964 git.exe Required memory at 0x9d7bef5020 is not valid (process exited?)

4168 cmd.exe Required memory at 0xbe88bc020 is not valid (process exited?)

6160 git.exe Required memory at 0xd1ebbb020 is not valid (process exited?)

6380 cmd.exe Required memory at 0x3a0ea68020 is not valid (process exited?)

7588 git.exe Required memory at 0x88661f9020 is not valid (process exited?)

3668 TrustedInstall C:\Windows\servicing\TrustedInstaller.exe

6592 svchost.exe C:\Windows\system32\svchost.exe -k netsvcs -p -s gpsvc

7776 TiWorker.exe C:\Windows\winsxs\amd64\_microsoft-windows-servicingstack\_31bf3856ad364e35\_10.0.17763.1934\_none\_56ae2b0b9948b43a\TiWorker.exe -Embedding

6980 svchost.exe C:\Windows\System32\svchost.exe -k netsvcs -p -s NetSetupSvc

6568 backgroundTask Required memory at 0x2b9985e020 is not valid (process exited?)

3780 MicrosoftEdge. "C:\Windows\SystemApps\Microsoft.MicrosoftEdge\_8wekyb3d8bbwe\MicrosoftEdge.exe" -ServerName:MicrosoftEdge.AppXdnhjhccw3zf0j06tkg3jtqr00qdm0khc.mca

1392 browser\_broker C:\Windows\system32\browser\_broker.exe -Embedding

4972 svchost.exe C:\Windows\System32\svchost.exe -k LocalServiceNetworkRestricted

2844 Windows.WARP.J C:\Windows\system32\Windows.WARP.JITService.exe 42268828-5abf-47b2-b08c-7c1fc14a8a3d S-1-15-2-3624051433-2125758914-1423191267-1740899205-1073925389-3782572162-737981194 S-1-5-21-3461203602-4096304019-2269080069-1000 564

6780 RuntimeBroker. C:\Windows\System32\RuntimeBroker.exe -Embedding

1040 MicrosoftEdgeC "C:\Windows\System32\MicrosoftEdgeCP.exe" -ServerName:Windows.Internal.WebRuntime.ContentProcessServer

1564 svchost.exe C:\Windows\system32\svchost.exe -k appmodel -p -s camsvc

3472 MicrosoftEdgeS C:\Windows\system32\MicrosoftEdgeSH.exe SCODEF:3780 CREDAT:9730 APH:5F00000000007 JITHOST /prefetch:2

6844 svchost.exe C:\Windows\system32\svchost.exe -k wsappx -p -s AppXSvc

204 9840432e051a6f "C:\Program Files (x86)\SmartPDF\SmartPDF\9840432e051a6fa1192594db02b80a4c1fd73456.exe"

5780 Windows.WARP.J C:\Windows\system32\Windows.WARP.JITService.exe 42268829-5abf-47b2-b08c-7c1fc14a8a3d S-1-15-2-3624051433-2125758914-1423191267-1740899205-1073925389-3782572162-737981194-4256926629-1688279915-2739229046-3928706915 S-1-5-21-3461203602-4096304019-2269080069-1000 564

7496 LivelyScreenRe "C:\Program Files (x86)\SmartPDF\SmartPDF\LivelyScreenRecS3.0.exe"

528 SearchFilterHo "C:\Windows\system32\SearchFilterHost.exe" 0 768 772 780 8192 776

808 WmiPrvSE.exe C:\Windows\system32\wbem\wmiprvse.exe

8360 svchost.exe C:\Windows\system32\svchost.exe -k SystemNetworkService

8948 Setup.exe "C:\Users\IEUser\AppData\Local\Temp\is-V9C61.tmp\Setup.exe" /Verysilent

8996 conhost.exe \??\C:\Windows\system32\conhost.exe 0x4

8472 tmp8DCF\_tmp.ex "C:\Users\IEUser\AppData\Local\Temp\tmp8DCF\_tmp.exe"

8504 conhost.exe \??\C:\Windows\system32\conhost.exe 0x4

9148 WmiPrvSE.exe C:\Windows\system32\wbem\wmiprvse.exe

9096 winpmem\_mini\_x winpmem\_mini\_x64\_rc2.exe memory.raw

8804 sihost32.exe Required memory at 0x78 is not valid (process exited?)

### **Appendix B2 – Network Connections**

0xd28ebc5f2180 UDPv4 0.0.0.0 0 \* 0 7496 LivelyScreenRe 2023-02-19 18:30:06.000000

0xd28ebc5f3140 UDPv4 10.0.0.4 138 \* 0 4 System 2023-02-19 18:19:14.000000

0xd28ebc5f7580 UDPv4 169.254.120.189 50152 \* 0 6948 chrome.exe 2023-02-19 18:30:06.000000

0xd28ebc5f7c10 UDPv4 10.0.0.4 50151 \* 0 6948 chrome.exe 2023-02-19 18:30:06.000000

0xd28ebc5f83f0 UDPv4 0.0.0.0 0 \* 0 7496 LivelyScreenRe 2023-02-19 18:30:06.000000

0xd28ebc5f83f0 UDPv6 :: 0 \* 0 7496 LivelyScreenRe 2023-02-19 18:30:06.000000

0xd28ebc5f8690 UDPv4 0.0.0.0 0 \* 0 7496 LivelyScreenRe 2023-02-19 18:30:06.000000

0xd28ebc5f8690 UDPv6 :: 0 \* 0 7496 LivelyScreenRe 2023-02-19 18:30:06.000000

0xd28ebc5f8d20 UDPv4 0.0.0.0 0 \* 0 7496 LivelyScreenRe 2023-02-19 18:30:06.000000

0xd28ebe402bf0 TCPv4 10.0.0.4 51468 10.0.0.3 443 CLOSED - - N/A

### **Appendix B3 – Malfind**

7496 LivelyScreenRe 0x275466e0000 0x275466effff VadS PAGE\_EXECUTE\_READWRITE 2 1 Disabled

00 00 00 00 00 00 00 00 ........

eb 61 fd 08 0b 72 00 01 .a...r..

ee ff ee ff 02 00 00 00 ........

20 01 6e 46 75 02 00 00 ..nFu...

20 01 6e 46 75 02 00 00 ..nFu...

00 00 6e 46 75 02 00 00 ..nFu...

00 00 6e 46 75 02 00 00 ..nFu...

0f 00 00 00 00 00 00 00 ........ 00 00 00 00 00 00 00 00 eb 61 fd 08 0b 72 00 01 ee ff ee ff 02 00 00 00 20 01 6e 46 75 02 00 00 20 01 6e 46 75 02 00 00 00 00 6e 46 75 02 00 00 00 00 6e 46 75 02 00 00 0f 00 00 00 00 00 00 00

7496 LivelyScreenRe 0x2755f0b0000 0x2755f0bffff VadS PAGE\_EXECUTE\_READWRITE 7 1 Disabled

00 00 00 00 00 00 00 00 ........

1a 99 52 f7 3f 89 00 01 ..R.?...

ee ff ee ff 02 00 00 00 ........

20 01 0b 5f 75 02 00 00 ...\_u...

20 01 0b 5f 75 02 00 00 ...\_u...

00 00 0b 5f 75 02 00 00 ...\_u...

00 00 0b 5f 75 02 00 00 ...\_u...

0f 00 00 00 00 00 00 00 ........ 00 00 00 00 00 00 00 00 1a 99 52 f7 3f 89 00 01 ee ff ee ff 02 00 00 00 20 01 0b 5f 75 02 00 00 20 01 0b 5f 75 02 00 00 00 00 0b 5f 75 02 00 00 00 00 0b 5f 75 02 00 00 0f 00 00 00 00 00 00 00

7496 LivelyScreenRe 0x27561a70000 0x27561aa1fff VadS PAGE\_EXECUTE\_READWRITE 1 1 Disabled

00 00 00 00 00 00 00 00 ........

90 3f 0b 5f 75 02 00 00 .?.\_u...

90 3f 0b 5f 75 02 00 00 .?.\_u...

00 00 0b 5f 75 02 00 00 ...\_u...

00 0f a7 61 75 02 00 00 ...au...

00 10 a7 61 75 02 00 00 ...au...

00 20 aa 61 75 02 00 00 ...au...

02 00 00 00 00 00 00 00 ........ 00 00 00 00 00 00 00 00 90 3f 0b 5f 75 02 00 00 90 3f 0b 5f 75 02 00 00 00 00 0b 5f 75 02 00 00 00 0f a7 61 75 02 00 00 00 10 a7 61 75 02 00 00 00 20 aa 61 75 02 00 00 02 00 00 00 00 00 00 00

7496 LivelyScreenRe 0x7ff47d200000 0x7ff47d20ffff VadS PAGE\_EXECUTE\_READWRITE 1 1 Disabled

00 00 00 00 00 00 00 00 ........

78 0d 00 00 00 00 00 00 x.......

0c 00 00 00 49 c7 c2 00 ....I...

00 00 00 48 b8 40 4a 8f ...H.@J.

ef fc 7f 00 00 ff e0 49 .......I

c7 c2 01 00 00 00 48 b8 ......H.

40 4a 8f ef fc 7f 00 00 @J......

ff e0 49 c7 c2 02 00 00 ..I..... 00 00 00 00 00 00 00 00 78 0d 00 00 00 00 00 00 0c 00 00 00 49 c7 c2 00 00 00 00 48 b8 40 4a 8f ef fc 7f 00 00 ff e0 49 c7 c2 01 00 00 00 48 b8 40 4a 8f ef fc 7f 00 00 ff e0 49 c7 c2 02 00 00

7496 LivelyScreenRe 0x7ff47d210000 0x7ff47d2affff VadS PAGE\_EXECUTE\_READWRITE 2 1 Disabled

d8 ff ff ff ff ff ff ff ........

08 00 00 00 00 00 00 00 ........

01 00 00 00 00 00 00 00 ........

00 02 0e 03 38 00 00 00 ....8...

68 01 d7 07 0c 00 00 00 h.......

d8 5d 31 ee fc 7f 00 00 .]1.....

00 10 2f ee fc 7f 00 00 ../.....

08 4a 48 ee fc 7f 00 00 .JH..... d8 ff ff ff ff ff ff ff 08 00 00 00 00 00 00 00 01 00 00 00 00 00 00 00 00 02 0e 03 38 00 00 00 68 01 d7 07 0c 00 00 00 d8 5d 31 ee fc 7f 00 00 00 10 2f ee fc 7f 00 00 08 4a 48 ee fc 7f 00 00

### **Appendix B4 – Privileges**

204 9840432e051a6f 2 SeCreateTokenPrivilege Create a token object

204 9840432e051a6f 3 SeAssignPrimaryTokenPrivilege Replace a process-level token

204 9840432e051a6f 4 SeLockMemoryPrivilege Lock pages in memory

204 9840432e051a6f 5 SeIncreaseQuotaPrivilege Present Increase quotas

204 9840432e051a6f 6 SeMachineAccountPrivilege Add workstations to the domain

204 9840432e051a6f 7 SeTcbPrivilege Act as part of the operating system

204 9840432e051a6f 8 SeSecurityPrivilege Present Manage auditing and security log

204 9840432e051a6f 9 SeTakeOwnershipPrivilege Present Take ownership of files/objects

204 9840432e051a6f 10 SeLoadDriverPrivilege Present Load and unload device drivers

204 9840432e051a6f 11 SeSystemProfilePrivilege Present Profile system performance

204 9840432e051a6f 12 SeSystemtimePrivilege Present Change the system time

204 9840432e051a6f 13 SeProfileSingleProcessPrivilege Present Profile a single process

204 9840432e051a6f 14 SeIncreaseBasePriorityPrivilege Present Increase scheduling priority

204 9840432e051a6f 15 SeCreatePagefilePrivilege Present Create a pagefile

204 9840432e051a6f 16 SeCreatePermanentPrivilege Create permanent shared objects

204 9840432e051a6f 17 SeBackupPrivilege Present Backup files and directories

204 9840432e051a6f 18 SeRestorePrivilege Present Restore files and directories

204 9840432e051a6f 19 SeShutdownPrivilege Present Shut down the system

204 9840432e051a6f 20 SeDebugPrivilege Present Debug programs

204 9840432e051a6f 21 SeAuditPrivilege Generate security audits

204 9840432e051a6f 22 SeSystemEnvironmentPrivilege Present Edit firmware environment values

204 9840432e051a6f 23 SeChangeNotifyPrivilege Present,Enabled,Default Receive notifications of changes to files or directories

204 9840432e051a6f 24 SeRemoteShutdownPrivilege Present Force shutdown from a remote system

204 9840432e051a6f 25 SeUndockPrivilege Present Remove computer from docking station

204 9840432e051a6f 26 SeSyncAgentPrivilege Synch directory service data

204 9840432e051a6f 27 SeEnableDelegationPrivilege Enable user accounts to be trusted for delegation

204 9840432e051a6f 28 SeManageVolumePrivilege Present Manage the files on a volume

204 9840432e051a6f 29 SeImpersonatePrivilege Present,Enabled,Default Impersonate a client after authentication

204 9840432e051a6f 30 SeCreateGlobalPrivilege Present,Enabled,Default Create global objects

204 9840432e051a6f 31 SeTrustedCredManAccessPrivilege Access Credential Manager as a trusted caller

204 9840432e051a6f 32 SeRelabelPrivilege Modify the mandatory integrity level of an object

204 9840432e051a6f 33 SeIncreaseWorkingSetPrivilege Present Allocate more memory for user applications

204 9840432e051a6f 34 SeTimeZonePrivilege Present Adjust the time zone of the computer's internal clock

204 9840432e051a6f 35 SeCreateSymbolicLinkPrivilege Present Required to create a symbolic link

204 9840432e051a6f 36 SeDelegateSessionUserImpersonatePrivilege Present Obtain an impersonation token for another user in the same session.

### **Appendix B5 – Processes**

Volatility 3 Framework 2.4.1

PID PPID ImageFileName Offset(V) Threads Handles SessionId Wow64 CreateTime ExitTime File output

4 0 System 0xd28eb3869080 175 - N/A False 2022-05-19 16:48:58.000000 N/A Disabled

2168 680 svchost.exe 0xd28eb38b4080 7 - 0 False 2022-05-19 16:49:01.000000 N/A Disabled

1892 680 svchost.exe 0xd28eb38c1080 2 - 0 False 2022-05-19 16:49:01.000000 N/A Disabled

2056 680 svchost.exe 0xd28eb38cc080 10 - 0 False 2022-05-19 16:49:01.000000 N/A Disabled

2296 680 svchost.exe 0xd28eb38d7080 3 - 0 False 2022-05-19 16:49:01.000000 N/A Disabled

2288 680 svchost.exe 0xd28eb38db080 14 - 0 False 2022-05-19 16:49:01.000000 N/A Disabled

2508 680 svchost.exe 0xd28eb3928080 19 - 0 False 2022-05-19 16:49:01.000000 N/A Disabled

136 4 Registry 0xd28eb39cc040 4 - N/A False 2022-05-19 16:48:56.000000 N/A Disabled

7064 680 SecurityHealth 0xd28eb54eb080 7 - 0 False 2022-05-22 11:21:19.000000 N/A Disabled

364 4 smss.exe 0xd28eb592a0c0 2 - N/A False 2022-05-19 16:48:58.000000 N/A Disabled

6856 680 svchost.exe 0xd28eb92c9440 5 - 1 False 2022-05-22 11:19:19.000000 N/A Disabled

6580 680 SgrmBroker.exe 0xd28eb92e2540 3 - 0 False 2022-05-22 11:19:18.000000 N/A Disabled

5944 680 svchost.exe 0xd28eb92e9080 4 - 0 False 2023-01-25 12:48:44.000000 N/A Disabled

7156 680 svchost.exe 0xd28eb92ec080 9 - 0 False 2023-02-19 17:54:53.000000 N/A Disabled

3112 680 svchost.exe 0xd28eb9414540 10 - 0 False 2022-05-22 11:19:18.000000 N/A Disabled

5780 4972 Windows.WARP.J 0xd28eb9416080 4 - 0 False 2023-02-19 18:19:14.000000 N/A Disabled

7496 5776 LivelyScreenRe 0xd28eb9417080 15 - 1 False 2023-02-19 18:19:14.000000 N/A Disabled

456 444 csrss.exe 0xd28eb94d4140 12 - 0 False 2022-05-19 16:49:01.000000 N/A Disabled

208 680 svchost.exe 0xd28eb9588080 5 - 0 False 2022-05-22 14:10:47.000000 N/A Disabled

4472 680 svchost.exe 0xd28eb958e080 8 - 0 False 2022-05-22 21:04:23.000000 N/A Disabled

3392 4104 ConEmu64.exe 0xd28eb961a080 11 - 1 False 2023-01-25 13:14:40.000000 N/A Disabled

6368 680 svchost.exe 0xd28eb96fc080 14 - 0 False 2022-05-22 11:18:16.000000 N/A Disabled

7320 3392 ConEmuC64.exe 0xd28eb980e080 8 - 1 False 2023-01-25 13:14:41.000000 N/A Disabled

532 444 wininit.exe 0xd28eba2bb080 1 - 0 False 2022-05-19 16:49:01.000000 N/A Disabled

548 524 csrss.exe 0xd28eba2c9440 14 - 1 False 2022-05-19 16:49:01.000000 N/A Disabled

6076 680 svchost.exe 0xd28eba2e2080 13 - 0 False 2022-05-22 11:17:55.000000 N/A Disabled

636 524 winlogon.exe 0xd28eba2ed240 5 - 1 False 2022-05-19 16:49:01.000000 N/A Disabled

680 532 services.exe 0xd28eba311080 9 - 0 False 2022-05-19 16:49:01.000000 N/A Disabled

688 532 lsass.exe 0xd28eba3230c0 9 - 0 False 2022-05-19 16:49:01.000000 N/A Disabled

820 680 svchost.exe 0xd28eba3a30c0 1 - 0 False 2022-05-19 16:49:01.000000 N/A Disabled

844 532 fontdrvhost.ex 0xd28eba3a9080 5 - 0 False 2022-05-19 16:49:01.000000 N/A Disabled

860 680 svchost.exe 0xd28eba3aa080 22 - 0 False 2022-05-19 16:49:01.000000 N/A Disabled

852 636 fontdrvhost.ex 0xd28eba3ac080 5 - 1 False 2022-05-19 16:49:01.000000 N/A Disabled

6160 6620 git.exe 0xd28eba3af080 0 - 1 False 2023-02-19 18:17:51.000000 2023-02-19 18:17:51.000000 Disabled

972 680 svchost.exe 0xd28eba3f1240 17 - 0 False 2022-05-19 16:49:01.000000 N/A Disabled

1016 680 svchost.exe 0xd28ebaa52080 5 - 0 False 2022-05-19 16:49:01.000000 N/A Disabled

3300 1396 cmd.exe 0xd28ebaaa92c0 0 - 1 False 2023-02-19 18:04:36.000000 2023-02-19 18:04:36.000000 Disabled

404 636 dwm.exe 0xd28ebaaab080 23 - 1 False 2022-05-19 16:49:01.000000 N/A Disabled

396 680 svchost.exe 0xd28ebab10080 3 - 0 False 2022-05-19 16:49:01.000000 N/A Disabled

7304 4520 SearchProtocol 0xd28ebab14080 8 - 0 False 2023-02-19 18:12:10.000000 N/A Disabled

1112 680 svchost.exe 0xd28ebab42080 5 - 0 False 2022-05-19 16:49:01.000000 N/A Disabled

1120 680 svchost.exe 0xd28ebab6a0c0 6 - 0 False 2022-05-19 16:49:01.000000 N/A Disabled

1168 680 svchost.exe 0xd28ebab76080 16 - 0 False 2022-05-19 16:49:01.000000 N/A Disabled

1184 680 svchost.exe 0xd28ebab950c0 9 - 0 False 2022-05-19 16:49:01.000000 N/A Disabled

1352 680 svchost.exe 0xd28ebabde080 7 - 0 False 2022-05-19 16:49:01.000000 N/A Disabled

1272 680 svchost.exe 0xd28ebabe6080 2 - 0 False 2022-05-19 16:49:01.000000 N/A Disabled

1380 680 svchost.exe 0xd28ebac63080 10 - 0 False 2022-05-19 16:49:01.000000 N/A Disabled

1448 680 svchost.exe 0xd28ebac7f080 16 - 0 False 2022-05-19 16:49:01.000000 N/A Disabled

1536 680 svchost.exe 0xd28ebacef080 10 - 0 False 2022-05-19 16:49:01.000000 N/A Disabled

1604 680 VBoxService.ex 0xd28ebad19080 11 - 0 False 2022-05-19 16:49:01.000000 N/A Disabled

1680 680 svchost.exe 0xd28ebad31080 17 - 0 False 2022-05-19 16:49:01.000000 N/A Disabled

1780 680 svchost.exe 0xd28ebade3080 14 - 0 False 2022-05-19 16:49:01.000000 N/A Disabled

1812 680 svchost.exe 0xd28ebadeb080 9 - 0 False 2022-05-19 16:49:01.000000 N/A Disabled

1804 680 svchost.exe 0xd28ebadee080 8 - 0 False 2022-05-19 16:49:01.000000 N/A Disabled

1820 680 svchost.exe 0xd28ebadf1080 7 - 0 False 2022-05-19 16:49:01.000000 N/A Disabled

3244 1396 cmd.exe 0xd28ebae17540 0 - 1 False 2023-02-19 18:17:40.000000 2023-02-19 18:17:40.000000 Disabled

1404 5452 git.exe 0xd28ebae2e300 0 - 1 False 2023-02-19 18:04:55.000000 2023-02-19 18:04:55.000000 Disabled

1956 680 svchost.exe 0xd28ebae32080 16 - 0 False 2022-05-19 16:49:01.000000 N/A Disabled

1964 4 MemCompression 0xd28ebae47080 42 - N/A False 2022-05-19 16:49:01.000000 N/A Disabled

2032 680 svchost.exe 0xd28ebae74080 9 - 0 False 2022-05-19 16:49:01.000000 N/A Disabled

2336 680 svchost.exe 0xd28ebaf3b080 8 - 0 False 2022-05-19 16:49:01.000000 N/A Disabled

2484 680 spoolsv.exe 0xd28ebaf79080 9 - 0 False 2022-05-19 16:49:01.000000 N/A Disabled

2556 680 svchost.exe 0xd28ebb032080 3 - 0 False 2022-05-19 16:49:01.000000 N/A Disabled

2680 680 svchost.exe 0xd28ebb07d080 13 - 0 False 2022-05-19 16:49:01.000000 N/A Disabled

2756 680 svchost.exe 0xd28ebb083080 13 - 0 False 2022-05-19 16:49:01.000000 N/A Disabled

2764 680 svchost.exe 0xd28ebb087080 6 - 0 False 2022-05-19 16:49:01.000000 N/A Disabled

2600 680 svchost.exe 0xd28ebb0b5080 7 - 0 False 2022-05-19 16:49:01.000000 N/A Disabled

2848 680 svchost.exe 0xd28ebb0d9080 7 - 0 False 2022-05-19 16:49:01.000000 N/A Disabled

2908 680 svchost.exe 0xd28ebb111080 20 - 0 False 2022-05-19 16:49:01.000000 N/A Disabled

2944 680 svchost.exe 0xd28ebb114080 12 - 0 False 2022-05-19 16:49:01.000000 N/A Disabled

2936 680 svchost.exe 0xd28ebb11a080 3 - 0 False 2022-05-19 16:49:01.000000 N/A Disabled

2952 680 svchost.exe 0xd28ebb11d080 17 - 0 False 2022-05-19 16:49:01.000000 N/A Disabled

2920 680 wlms.exe 0xd28ebb125080 2 - 0 False 2022-05-19 16:49:01.000000 N/A Disabled

2964 680 ruby.exe 0xd28ebb129080 9 - 0 False 2022-05-19 16:49:01.000000 N/A Disabled

6524 680 svchost.exe 0xd28ebb159080 7 - 0 False 2022-05-22 11:19:19.000000 N/A Disabled

2176 680 svchost.exe 0xd28ebb1d00c0 20 - 0 False 2022-05-19 16:49:01.000000 N/A Disabled

2624 680 svchost.exe 0xd28ebb1d6080 10 - 0 False 2022-05-19 16:49:01.000000 N/A Disabled

2648 680 svchost.exe 0xd28ebb1db080 4 - 0 False 2022-05-19 16:49:01.000000 N/A Disabled

7588 6024 git.exe 0xd28ebb2222c0 0 - 1 False 2023-02-19 18:17:53.000000 2023-02-19 18:17:53.000000 Disabled

3800 680 svchost.exe 0xd28ebb2b0080 4 - 1 False 2022-05-19 16:49:07.000000 N/A Disabled

3764 680 svchost.exe 0xd28ebb2b2080 4 - 1 False 2022-05-19 16:49:07.000000 N/A Disabled

3732 1536 sihost.exe 0xd28ebb2b4080 15 - 1 False 2022-05-19 16:49:07.000000 N/A Disabled

1636 680 svchost.exe 0xd28ebb2b8080 0 - 0 False 2022-05-22 11:17:56.000000 2022-05-22 11:18:01.000000 Disabled

4024 680 svchost.exe 0xd28ebb2ba080 8 - 0 False 2022-05-19 16:49:07.000000 N/A Disabled

1864 6900 git.exe 0xd28ebb3d2540 0 - 1 False 2023-02-19 18:05:08.000000 2023-02-19 18:05:08.000000 Disabled

4032 3948 ctfmon.exe 0xd28ebb4ec080 11 - 1 False 2022-05-19 16:49:07.000000 N/A Disabled

6036 4736 conhost.exe 0xd28ebb536080 4 - 0 False 2022-05-22 11:17:55.000000 N/A Disabled

3912 680 svchost.exe 0xd28ebb593080 9 - 0 False 2022-05-19 16:49:07.000000 N/A Disabled

3948 680 svchost.exe 0xd28ebb59d080 5 - 0 False 2022-05-19 16:49:07.000000 N/A Disabled

3116 680 svchost.exe 0xd28ebb62b080 9 - 0 False 2022-05-19 16:49:07.000000 N/A Disabled

3824 636 userinit.exe 0xd28ebb6e0080 0 - 1 False 2022-05-19 16:49:07.000000 2022-05-19 16:49:30.000000 Disabled

2672 3824 explorer.exe 0xd28ebb6ec080 118 - 1 False 2022-05-19 16:49:07.000000 N/A Disabled

4000 1168 taskhostw.exe 0xd28ebb716080 12 - 1 False 2022-05-19 16:49:07.000000 N/A Disabled

4204 680 svchost.exe 0xd28ebb76b080 9 - 1 False 2022-05-19 16:49:07.000000 N/A Disabled

8684 9008 powershell.exe 0xd28ebb773080 0 - 1 False 2023-02-19 18:30:13.000000 2023-02-19 18:30:13.000000 Disabled

5568 7944 software\_repor 0xd28ebb81d080 0 - 1 False 2023-02-07 15:41:02.000000 2023-02-13 14:48:23.000000 Disabled

4520 680 SearchIndexer. 0xd28ebb83a3c0 53 - 0 False 2022-05-19 16:49:08.000000 N/A Disabled

4828 860 SearchUI.exe 0xd28ebb844080 45 - 1 False 2022-05-19 16:49:08.000000 N/A Disabled

4736 1168 cmd.exe 0xd28ebb8724c0 1 - 0 False 2022-05-22 11:17:55.000000 N/A Disabled

5880 5872 conhost.exe 0xd28ebb8c10c0 4 - 0 False 2022-05-22 11:17:55.000000 N/A Disabled

4676 860 ShellExperienc 0xd28ebb8de080 20 - 1 False 2022-05-19 16:49:08.000000 N/A Disabled

3164 680 svchost.exe 0xd28ebb8f6080 6 - 0 False 2022-05-22 14:10:47.000000 N/A Disabled

4976 860 RuntimeBroker. 0xd28ebb9bd540 10 - 1 False 2022-05-19 16:49:08.000000 N/A Disabled

6844 680 svchost.exe 0xd28ebba04240 7 - 0 False 2023-02-19 18:19:14.000000 N/A Disabled

6948 2672 chrome.exe 0xd28ebba0a080 31 - 1 False 2023-02-19 17:54:35.000000 N/A Disabled

3964 1944 git.exe 0xd28ebba0b540 0 - 1 False 2023-02-19 18:17:40.000000 2023-02-19 18:17:40.000000 Disabled

6312 1396 cmd.exe 0xd28ebba0c080 0 - 1 False 2023-02-19 18:04:37.000000 2023-02-19 18:04:37.000000 Disabled

3352 860 WindowsInterna 0xd28ebba2c540 32 - 1 False 2022-05-22 11:38:45.000000 N/A Disabled

5960 2672 VBoxTray.exe 0xd28ebba2d540 13 - 1 False 2022-05-19 16:49:19.000000 N/A Disabled

5044 860 RuntimeBroker. 0xd28ebbb19080 13 - 1 False 2022-05-19 16:49:08.000000 N/A Disabled

3668 680 TrustedInstall 0xd28ebbb1d500 6 - 0 False 2023-02-19 18:18:41.000000 N/A Disabled

6568 860 backgroundTask 0xd28ebbb68080 8 - 1 False 2023-02-19 18:18:49.000000 N/A Disabled

6380 1396 cmd.exe 0xd28ebbbdf540 0 - 1 False 2023-02-19 18:17:53.000000 2023-02-19 18:17:53.000000 Disabled

2932 680 svchost.exe 0xd28ebbc55080 3 - 0 False 2022-05-22 11:19:18.000000 N/A Disabled

5560 860 RuntimeBroker. 0xd28ebbc5b080 10 - 1 False 2022-05-19 16:49:12.000000 N/A Disabled

6016 6948 chrome.exe 0xd28ebbe3d540 8 - 1 False 2023-02-19 17:54:36.000000 N/A Disabled

5872 1168 cmd.exe 0xd28ebbeb3080 1 - 0 False 2022-05-22 11:17:55.000000 N/A Disabled

4300 6948 chrome.exe 0xd28ebbed5540 19 - 1 False 2023-02-19 17:54:36.000000 N/A Disabled

5376 1396 cmd.exe 0xd28ebbee1580 0 - 1 False 2023-02-19 18:04:50.000000 2023-02-19 18:04:50.000000 Disabled

4168 1396 cmd.exe 0xd28ebbfbe540 0 - 1 False 2023-02-19 18:17:51.000000 2023-02-19 18:17:51.000000 Disabled

6800 680 svchost.exe 0xd28ebc0d5080 7 - 0 False 2022-05-22 11:19:18.000000 N/A Disabled

5136 2672 Code.exe 0xd28ebc0ea500 0 - 1 False 2022-05-22 14:37:49.000000 2022-05-22 14:37:58.000000 Disabled

5752 680 svchost.exe 0xd28ebc2e4080 12 - 0 False 2022-05-22 11:17:57.000000 N/A Disabled

6128 680 svchost.exe 0xd28ebc2ea0c0 7 - 0 False 2022-05-22 11:18:06.000000 N/A Disabled

2216 860 dllhost.exe 0xd28ebc39a080 7 - 1 False 2022-05-22 11:18:14.000000 N/A Disabled

7776 860 TiWorker.exe 0xd28ebc3a1500 4 - 0 False 2023-02-19 18:18:41.000000 N/A Disabled

6672 1168 taskhostw.exe 0xd28ebc3a9080 6 - 1 False 2022-05-22 11:32:18.000000 N/A Disabled

828 2672 notepad++.exe 0xd28ebc3bf540 0 - 1 False 2023-02-19 18:11:58.000000 2023-02-19 18:12:01.000000 Disabled

6592 680 svchost.exe 0xd28ebc8ee500 8 - 0 False 2023-02-19 18:18:41.000000 N/A Disabled

1908 680 svchost.exe 0xd28ebcae4080 2 - 0 False 2023-01-25 12:48:44.000000 N/A Disabled

3096 860 dllhost.exe 0xd28ebcc67080 5 - 1 False 2022-05-22 16:32:30.000000 N/A Disabled

5128 680 sppsvc.exe 0xd28ebcc76080 5 - 0 False 2023-01-25 12:43:23.000000 N/A Disabled

3472 6780 MicrosoftEdgeS 0xd28ebcf36080 12 - 1 False 2023-02-19 18:19:14.000000 N/A Disabled

1136 680 svchost.exe 0xd28ebcf41240 18 - 0 False 2023-01-25 12:48:45.000000 N/A Disabled

3780 860 MicrosoftEdge. 0xd28ebcf52080 43 - 1 False 2023-02-19 18:19:13.000000 N/A Disabled

4992 860 SppExtComObj.E 0xd28ebcf7b080 3 - 0 False 2023-01-25 12:43:24.000000 N/A Disabled

2844 4972 Windows.WARP.J 0xd28ebcf87080 4 - 0 False 2023-02-19 18:19:14.000000 N/A Disabled

1564 680 svchost.exe 0xd28ebcf96080 7 - 0 False 2023-02-19 18:19:14.000000 N/A Disabled

6780 860 RuntimeBroker. 0xd28ebd0b9080 14 - 1 False 2023-02-19 18:19:14.000000 N/A Disabled

6912 860 ApplicationFra 0xd28ebd0bd080 19 - 1 False 2023-01-25 12:49:08.000000 N/A Disabled

8928 6368 CompatTelRunne 0xd28ebd0be080 4 - 0 False 2023-02-19 18:30:44.000000 N/A Disabled

8504 8472 conhost.exe 0xd28ebd146080 6 - 1 False 2023-02-19 18:30:06.000000 N/A Disabled

7900 7320 conhost.exe 0xd28ebd159080 3 - 1 False 2023-01-25 13:14:41.000000 N/A Disabled

7956 7632 git.exe 0xd28ebd1632c0 0 - 1 False 2023-02-19 18:04:56.000000 2023-02-19 18:04:56.000000 Disabled

5356 7944 software\_repor 0xd28ebd16b080 0 - 1 False 2023-02-07 15:41:02.000000 2023-02-13 14:48:23.000000 Disabled

1396 7320 cmd.exe 0xd28ebd38c080 2 - 1 False 2023-01-25 13:14:41.000000 N/A Disabled

3220 680 svchost.exe 0xd28ebd3b0080 4 - 0 False 2023-01-25 12:49:09.000000 N/A Disabled

7664 680 svchost.exe 0xd28ebd3b9080 6 - 0 False 2023-01-25 12:49:24.000000 N/A Disabled

4972 680 svchost.exe 0xd28ebd585080 5 - 0 False 2023-02-19 18:19:14.000000 N/A Disabled

7120 6948 chrome.exe 0xd28ebd6b9540 17 - 1 False 2023-02-19 17:57:40.000000 N/A Disabled

3648 680 svchost.exe 0xd28ebd6c6080 6 - 0 False 2023-02-13 14:47:08.000000 N/A Disabled

9096 7952 winpmem\_mini\_x 0xd28ebd7ed080 4 - 1 False 2023-02-19 18:30:10.000000 N/A Disabled

7952 2672 cmd.exe 0xd28ebd8d4540 1 - 1 False 2023-02-19 17:59:39.000000 N/A Disabled

7276 1396 cmd.exe 0xd28ebdb0e2c0 0 - 1 False 2023-02-19 18:04:56.000000 2023-02-19 18:04:56.000000 Disabled

5692 1396 cmd.exe 0xd28ebdb8e080 0 - 1 False 2023-02-19 18:05:08.000000 2023-02-19 18:05:08.000000 Disabled

8472 7496 tmp8DCF\_tmp.ex 0xd28ebdb9f080 4 - 1 True 2023-02-19 18:30:06.000000 N/A Disabled

1512 6948 chrome.exe 0xd28ebdda1080 8 - 1 False 2023-02-19 17:54:36.000000 N/A Disabled

1392 860 browser\_broker 0xd28ebded3080 13 - 1 False 2023-02-19 18:19:13.000000 N/A Disabled

7432 1396 cmd.exe 0xd28ebdf1d080 0 - 1 False 2023-02-19 18:04:55.000000 2023-02-19 18:04:55.000000 Disabled

5816 7952 conhost.exe 0xd28ebe1b8540 4 - 1 False 2023-02-19 17:59:39.000000 N/A Disabled

7100 1396 cmd.exe 0xd28ebe1c6080 0 - 1 False 2023-02-19 18:11:26.000000 2023-02-19 18:11:26.000000 Disabled

7148 680 svchost.exe 0xd28ebe1c7080 6 - 0 False 2023-02-07 15:37:09.000000 N/A Disabled

8360 6128 svchost.exe 0xd28ebe1ea080 7 - 1 False 2023-02-19 18:19:14.000000 N/A Disabled

3772 6720 git.exe 0xd28ebe2ec080 0 - 1 False 2023-02-19 18:04:50.000000 2023-02-19 18:04:50.000000 Disabled

8996 8948 conhost.exe 0xd28ebe2f2080 7 - 1 False 2023-02-19 18:19:15.000000 N/A Disabled

528 4520 SearchFilterHo 0xd28ebe416080 7 - 0 False 2023-02-19 18:19:14.000000 N/A Disabled

6980 680 svchost.exe 0xd28ebe418080 8 - 0 False 2023-02-19 18:18:48.000000 N/A Disabled

1040 860 MicrosoftEdgeC 0xd28ebe447080 43 - 1 False 2023-02-19 18:19:14.000000 N/A Disabled

7024 6948 chrome.exe 0xd28ebe460540 18 - 1 False 2023-02-19 17:54:36.000000 N/A Disabled

8948 5412 Setup.exe 0xd28ebe5e1080 4 - 1 True 2023-02-19 18:19:15.000000 N/A Disabled

204 5776 9840432e051a6f 0xd28ebe5eb080 11 - 1 True 2023-02-19 18:19:14.000000 N/A Disabled

9148 860 WmiPrvSE.exe 0xd28ebe5f4080 10 - 0 False 2023-02-19 18:30:07.000000 N/A Disabled

8600 860 MicrosoftEdgeC 0xd28ebe5fa080 20 - 1 False 2023-02-19 18:30:20.000000 N/A Disabled

7748 8156 git.exe 0xd28ebe8460c0 0 - 1 False 2023-02-19 18:11:26.000000 2023-02-19 18:11:26.000000 Disabled

808 860 WmiPrvSE.exe 0xd28ebe84b080 11 - 0 False 2023-02-19 18:19:14.000000 N/A Disabled

6992 7240 Code.exe 0xd28ebe9a3540 0 - 1 False 2023-02-19 17:56:55.000000 2023-02-19 17:57:18.000000 Disabled

9128 860 MicrosoftEdgeC 0xd28ebe9a8080 20 - 1 False 2023-02-19 18:30:19.000000 N/A Disabled

8804 9184 sihost32.exe 0xd28ebe9ab080 9 - 1 False 2023-02-19 18:30:18.000000 N/A Disabled

7996 6948 chrome.exe 0xd28ebe9ad540 11 - 1 False 2023-02-19 18:14:22.000000 N/A Disabled

## **Appendix C – Yara Rule**

