Assignment Guidance and Front Sheet

This front sheet for assignments is designed to contain the brief, the submission instructions, and the actual student submission for any WMG assignment. As a result the sheet is completed by several people over time, and is therefore split up into sections explaining who completes what information and when. Yellow highlighted text indicates examples or further explanation of what is requested, and the highlight and instructions should be removed as you populate 'your' section.

This sheet is only to be used for components of assessment worth more than 3 CATS (e.g. for a 15 credit module, weighted more than 20%; or for a 10 credit module, weighted more than 30%).

To be completed by the student(s) prior to final submission:

Your actual submission should be written at the end of this cover sheet file, or attached with the cover sheet at the front if drafted in a separate file, program or application.

Student ID or IDs for group work	**U2136249**

To be completed (highlighted parts only) by the programme administration after approval and prior to issuing of the assessment; to be consulted by the student(s) so that you know how and when to submit:

Date set	26th October 2023
Submission date (excluding extensions)	08th December 2023 by 12:00pm.
Submission guidance	Tabula link
Marks return date (excluding extensions)	11/03/2024
Late submission policy	If work is submitted late, penalties will be applied at the rate of **5 marks per University working day** after the due date, up to a **maximum of 10 working days** late. After this period the mark for the work will be reduced to 0 (which is the maximum penalty). "Late" means **after the submission deadline time as well as the date** – work submitted after the given time even on the same day is counted as 1 day late.
Resubmission policy	If you fail this assignment or module, please be aware that the University allows students to remedy such failure (within certain limits). Decisions to authorise such resubmissions are made by Exam Boards. Normally these will be issued at specific times of the year, depending on your programme of study. More information can be found from your programme office if you are concerned.

To be completed by the module owner/tutor prior to approval and issuing of the assessment; to be consulted by the student(s) so that you understand the assignment brief, its context within the module, and any specific criteria and advice from the tutor:

Module title & code	WM3B3 Low Level Tools and Techniques for Cyber Security
Module owner	Christo Panchev
Module tutor	Christo Panchev
Assessment type	Coursework 1: Malware Analysis

Weighting of mark	60

Word count	2500 words (excluding source code, programme input/output) You will not be penalised for producing under length work, provided quality is not sacrificed to brevity. Learning to write to a limit is one of the skills the degree is designed to encourage you to cultivate.		
Does the word count allow +10%? **Select ONE**	Does the word count include tables? **Select ONE**	Does the word count include references? **Select ONE**	Does the word count include appendices? **Select ONE**
Yes	Yes	No	Yes
			If appendices are included, will they be marked? Yes

Submission format	PDF

Module learning outcomes (numbered)	1. Identify common idioms and patterns used during code transformation and explain the origin and organisation of arbitrary code and/or data fragments within an executable program. 2. Apply tools and techniques as appropriate to infer the executable's overall high-level function, potentially obfuscated, potentially malicious code. 3. To perform malicious code analysis, vulnerability identification and evaluation independently from the findings generated by automated analysis tools.
Learning outcomes assessed in this assessment (numbered)	As above
Marking guidelines	In depth analysis related to the given scenario: 20 marks Critical discussion with appropriate justifications: 30 marks Utilised appropriate tools and Techniques: 20 marks IOC findings automation 20 marks Presentation of findings 10 marks
Academic guidance resources	You will have an opportunity to ask questions and get support on the assessment after it has been handed to you. You will be supported in this assessment through: 1. A special Moodle forum. 2. Through emails directed to the module tutor. **Notes to students:** If support is provided on a Teams Channel or a Moodle forum, please ensure you check previous questions posted on the channel. The Teams/Moodle channel will typically be closed one week before the submission date and no new questions will be addressed, please organise your time accordingly. Please be patient with module tutors. Please turn on your Teams Channel/Moodle notifications. If a tutor has not responded to a query within 5 working days, please email the module tutor.
Special instructions	Do not include the PMA specification in the submission. **Spelling/grammar.** Ensure that you spell check the submission, use a grammar checker and ensure that you proofread your work prior to submission. Spell/grammar checkers must be set to UK English, do not use ‘Americanised’ spellings.

	References. References are to be included at the end of the report (or do you want footnotes?) using the Harvard referencing system. You should not include a bibliography. Each reference must be connected to a citation within the main body of the report. Do not attempt to hide text within JPEGs, this will be construed as an attempt to mislead the assessor. **Coherence**. A poorly worded report will hide excellent content. The narrative should be easy to read, and arguments should be presented coherently and convincingly. **Presentation.** At this stage in your studies, there is no excuse for poor presentation. You will not receive marks for presentation; however, your submission will be penalised for poor presentation. **Formatting**. All figures and tables must be properly labelled and captioned. All pages must be numbered. Formatting must be consistently applied throughout the submission. Submissions that stray from this guidance may be penalised.

Malware Threat Analysis Report

Contents

Executive Summary

In response to DodoSOC's abnormal behaviour on an employee's computer, a technical analysis of the incident was carried out. Employing a systematic approach to malware analysis (SAMA), the investigation is divided into four process stages: initial actions, classification, code and behavioural analysis to provide an understanding of the operation, identification and potential prevention techniques against the malware. Using tools such as x64dbg Debugger and VirtusTotal, a malicious program was uncovered and classified as a Trojan Ransomware, known as "TeslaCrypt". Code analysis, involving static and dynamic techniques demonstrate the malware's behaviour during runtime, including file system alterations to encrypt victim's files, network connections, registry changes for persistence mechanisms, process injection and obfuscation techniques to avoid detection.

Methodology

To conduct the malware investigation, a systematic approach to malware analysis (SAMA) was used to get a full understanding of the malicious software, its operation, its identification and the ways of removing the threat (Bermejo Higuera et al., 2020). The process associated with this methodology is as follows on Figure 1:

Process	Objective
1. **Initial Actions**: begin with a series of actions to start the analysis of the malware in a clean state without any possible infection, including setting up a virtual machine without network access or folder sharing.	• Preserve Integrity • Obtain a snapshot / reference backup. • key observations • Collect supporting elements; logs, network captures, memory dumps, screenshots.
2. **Classification**: examine the malicious artifacts, without accessing code, to identify and obtain information about the type and functionality of the files	• Analyse file format, type, size. • Hash files. • Identity family of malware. • Find Opensource information about malware. • Antivirus identification • Analyse techniques used in malware obfuscation
3. **Code Analysis**: It consists of a static and dynamic analysis of the malware's code to get a better understanding of the malware's functionality. This is done through reverse engineering techniques to find hidden features.	• Overall Operation • Static Analysis Code • Dynamic Analysis Code • Memory Analysis
4. **Behaviour Analysis**: it consists in analysing the implementation of the malware in a safe environment, to observe the malware behaviour, network traffic generations and thus, the actions that are taken on the target system such as registry modifications or deletion of files.	• Determine pre-execution tasks • Identity services in execution • Detect of file system changes • Detect changes in registry. • Collect DNS queries. • Analyse network traffic

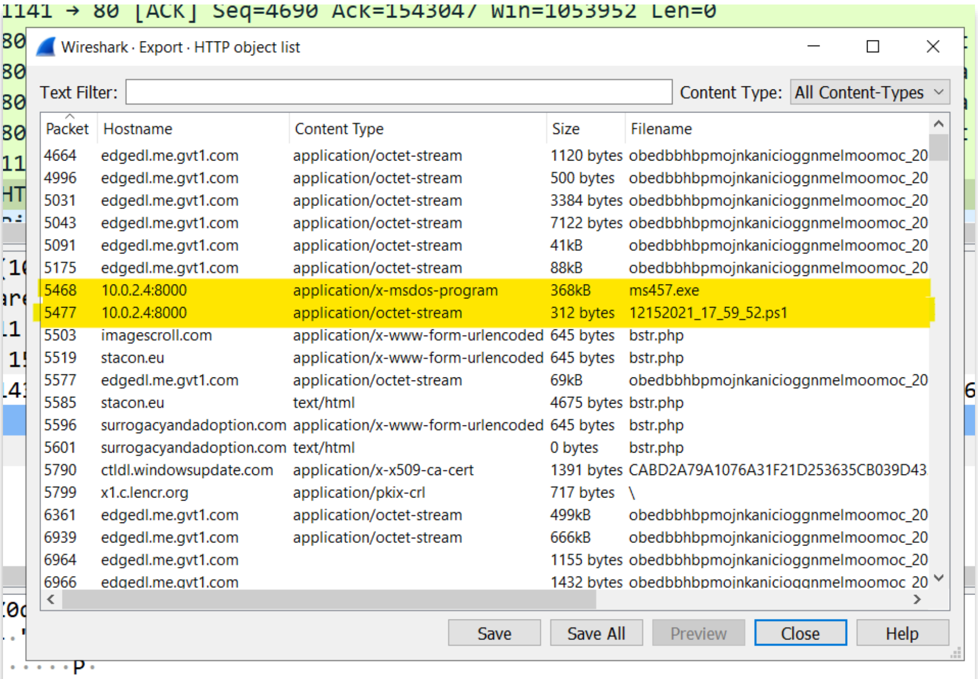
Figure 1 - Malware Analysis Methodology Used

Malware Analysis

To uncover the threat landscape of the malware found in the employee computer, an analysis of the given artifacts was performed including the malware binary file, memory dump and network traffic capture from the relevant employee's workstation to examine and reveal indicators of compromise.

Indicators of Compromised (ICOs).

Using Wireshark to analyse the employee's network traffic capture for unusual network behaviour, Figure 2, illustrates the first indicator of compromise; two successful HTTP GET Requests made by the employee to the IP address "10.0.2.4" on port "8080". These requests downloads two files; a windows executable "ms457.exe" and a PowerShell script "12152021_17_59_52.ps1". Using Wireshark's object export feature, the files were exported to a virtualised environment for further examination.



Packet	Hostname	Content Type	Size	Filename
4664	edgedl.me.gvt1.com	application/octet-stream	1120 bytes	obedbbhbpmojnkanicioggnmelmoomoc_20
4996	edgedl.me.gvt1.com	application/octet-stream	500 bytes	obedbbhbpmojnkanicioggnmelmoomoc_20
5031	edgedl.me.gvt1.com	application/octet-stream	3384 bytes	obedbbhbpmojnkanicioggnmelmoomoc_20
5043	edgedl.me.gvt1.com	application/octet-stream	7122 bytes	obedbbhbpmojnkanicioggnmelmoomoc_20
5091	edgedl.me.gvt1.com	application/octet-stream	41kB	obedbbhbpmojnkanicioggnmelmoomoc_20
5175	edgedl.me.gvt1.com	application/octet-stream	88kB	obedbbhbpmojnkanicioggnmelmoomoc_20
5468	10.0.2.4:8000	application/x-msdos-program	368kB	ms457.exe
5477	10.0.2.4:8000	application/octet-stream	312 bytes	12152021_17_59_52.ps1
5503	imagescroll.com	application/x-www-form-urlencoded	645 bytes	bstr.php
5519	stacon.eu	application/x-www-form-urlencoded	645 bytes	bstr.php
5577	edgedl.me.gvt1.com	application/octet-stream	69kB	obedbbhbpmojnkanicioggnmelmoomoc_20
5585	stacon.eu	text/html	4675 bytes	bstr.php
5596	surrogacyandadoption.com	application/x-www-form-urlencoded	645 bytes	bstr.php
5601	surrogacyandadoption.com	text/html	0 bytes	bstr.php
5790	ctldl.windowsupdate.com	application/x-x509-ca-cert	1391 bytes	CABD2A79A1076A31F21D253635CB039D43
5799	x1.c.lencr.org	application/pkix-crl	717 bytes	\
6361	edgedl.me.gvt1.com	application/octet-stream	499kB	obedbbhbpmojnkanicioggnmelmoomoc_20
6939	edgedl.me.gvt1.com	application/octet-stream	666kB	obedbbhbpmojnkanicioggnmelmoomoc_20
6964	edgedl.me.gvt1.com		1155 bytes	obedbbhbpmojnkanicioggnmelmoomoc_20
6966	edgedl.me.gvt1.com		1432 bytes	obedbbhbpmojnkanicioggnmelmoomoc_20

Figure 2 – Suspicious files found in network capture.

To create a footprint of these IOCs file hashes of both files were generated, as shown in Figure 3.

Artifact	MD5 Hash
1. 12152021_17_59_52.ps1	740ded988d0005f6892ba8176e60352d
2. ms457.exe	7991c88d40bbbfddcc8c85b427350af4

Figure 3 – List of Indicators of Compromise (IOCs)

Figure 4, showcases the contents of the PowerShell script. The script tries to manipulate the executable file "ms457.exe" by changing the hexadecimal value at offset 0x3C to be 4D (line 3-5) and then saving the modified binary as "MSUpdate.exe" in the employee's AppData directory (line 6). After that, it removes the original file (line 7) and executes the newly created "MSUdate.exe" file (line 8).

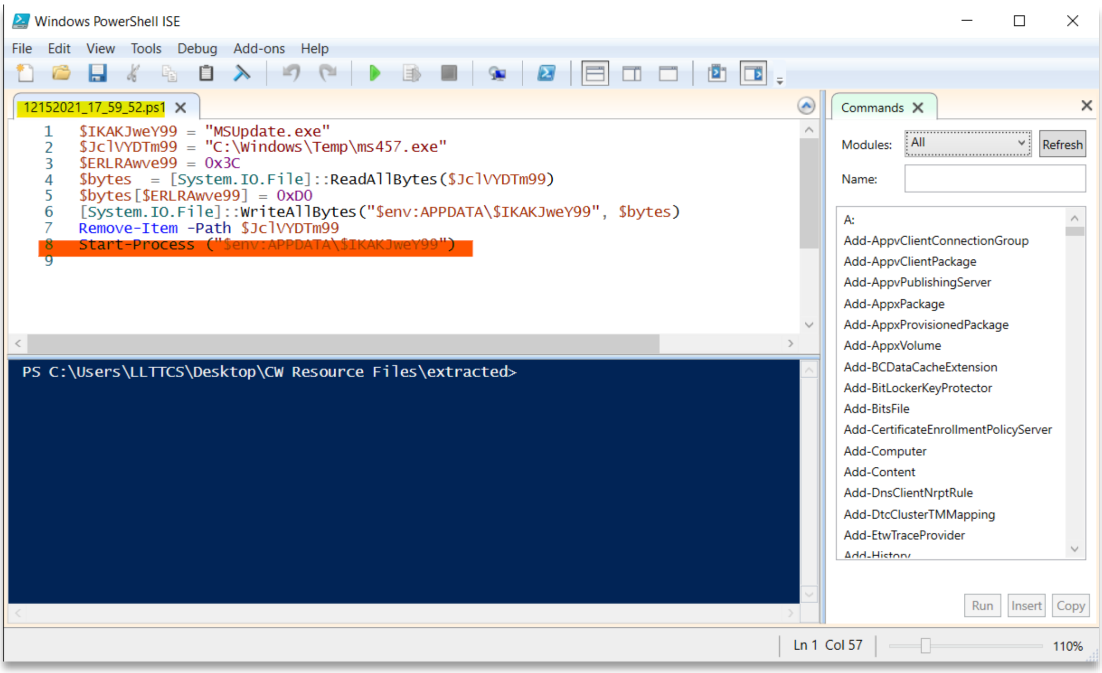


Figure 4 - Malicious PowerShell Script Found Network Capture

This unusual behaviour modifies the downloaded executable and hides it under a different name and directory, indicating another IOC. To avoid the Powershell script from running the malicious executable before inspecting its functionality and behaviour, line 8 was removed and the extracted file ("ms547.exe") from the employee's network capture was moved to the directory specified in the PowerShell script manually. Once this was completed, the script was run to generate the new file named "MSUpdate.exe" saved at "%APPDATA%" location (appendix A). From the modified executable a MD5 file was generated to add another IOC footprint, as shown in Figure 5.

Artifact	MD5 Hash
3. MSUpdate.exe	9ce01dfbf25dfea778e57d8274675d6f

Figure 5 – List of Indicators of Compromise (IOCs) (Cont.)

Malware Extraction

x64dgb Debugger	DiE	PEviewer

Figure 6 - Tools Used for Malware Extraction

To extract the malware, the modified file generated by the script was examined with the goal of extracting the actual malicious code which is hidden inside the executable. To achieve this, identifying whether the malware is obfuscated or packed is essential. One way to determine this, is by examining the entropy values of the executable. Figure 7, shows high entropy values for some code sections of the executable using DiE, indicating that the malware may be packed. However, considering this alone is a weak indicator as it could also indicate legitimate compression or encryption. Malware writers can also use polymorphic techniques to introduce high entropy and alter the appearance of the executable to cause confusion (Sikorski and Honig, 2012).

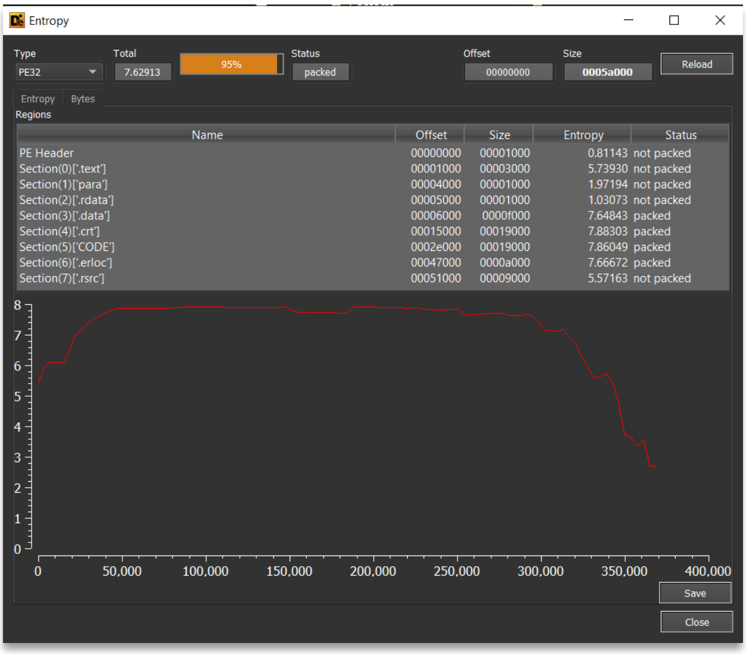


Figure 7 - Entropy Values for Modified Executable using DiE

A stronger indicator is to examine the PE Header of the modified executable and examine the size differences between the allocated "Virtual Size", and the actual size of "Raw Data". Figure 8, illustrates the .text section "IMAGE_SECTION_HEADER" using PEviewer, to showcase the differences in size. Looking at the differences, the allocated Virtual Size" is much larger than the "Raw Size" on disk, indicating that the executable contains packed code inside the .text section.

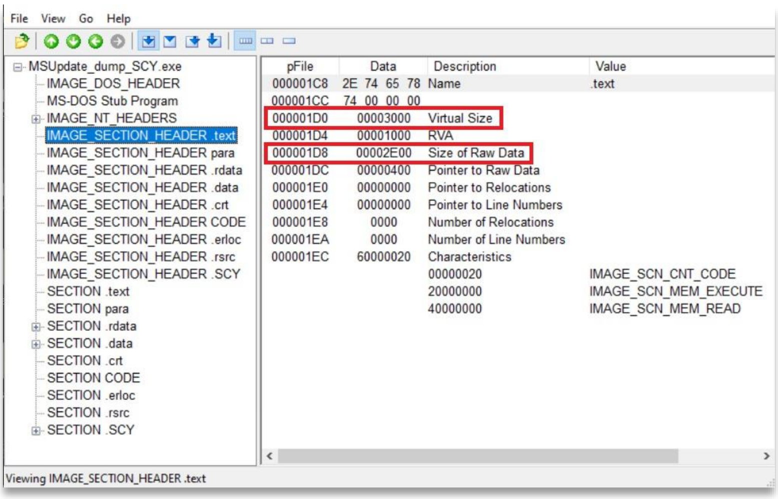


Figure 8 - "Virtual Size" and "Raw Size" Difference

To unpack the malware code from the executable, the methodology detailed above will be followed, focusing on Windows APIs as they have the capacity to create new processes and allocate memory to them. In particular the "VirtualAlloc" function will be examined using breakpoints when debugging the executable, since its main function is to allocate memory spaces to new processes.

Using the x64dgb debugger, the shortcut "CTRL+G" was used to search for the "VirtualAlloc" Windows API (appendix B). Figure 9, shows the current location module "Kernel32.dll" and displays the address where "VirtualAlloc" function is located. The objective is to determine the Original Entry Point (OEP), which is the address of the first instruction in the malicious code before packing. A breakpoint is place at the end of the function represented by the "jmp" command at the memory address "7514A6C" (appendix B), to investigate the code every time a breakpoint is hit when a call to "VirtualAlloc" is made. The aim is to let the malware stub to unpack itself in memory and then pause the execution at the OEP to reveal the actual malicious code (appendix B).

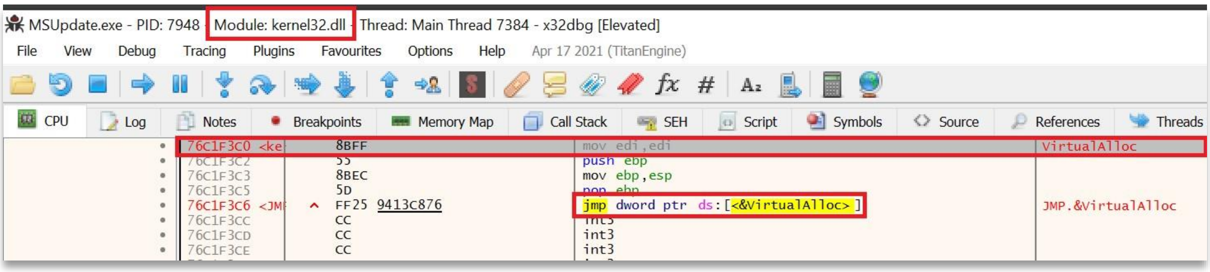


Figure 9 - "VirtualAlloc" in Kenel32.dll module

Figure 10, reveals the address location of the OEP address after letting the executable unpack in memory. Using the utility Scylla automatically detects the OEP address, where the actual malicious code beings before it was packed (appendix B). Figure 11, shows the "IAT Autosearch" feature of Scylla, it scans the memory address space of the current process and locates the import address table (IAT). To get all of the DLL imports the "Get Imports" feature was used. After this was done, the "Dump" feature was used to save the unpacked process in memory to disk. Finally, the "Fix Dump" feature was used to fix the IAT of the dumped file, leaving only the unpacked version of the malicious program to further analysis.

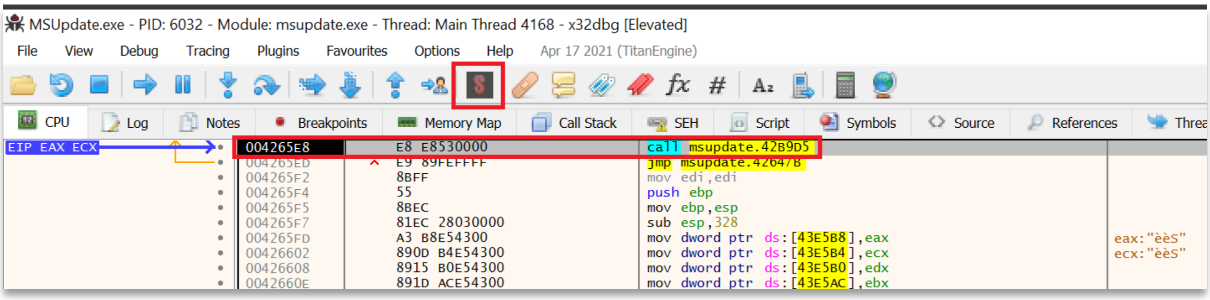


Figure 10 - Original Entry Point (OEP) address for the unpacked malware

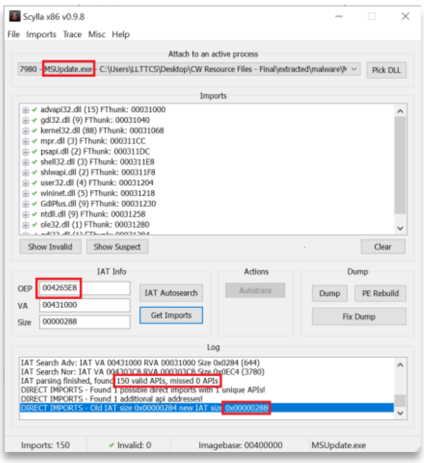


Figure 11 - Extracting all Import Address Table (IAT) using Scylla.

Malware Behaviour & Chain of Events

Static Analysis

PEviwer	PEstudio

Figure 12 - Tools Used for Static Analysis

Once the unpacked malware ("MSUpdate_dump_SYC.exe") was extracted, a MD5 file hash was generated to create another IOC footprint, as shown in Figure 13.

Artifact	MD5 Hash
4. MSUpdate_dump_SYC.exe	10ee3fe9f4b2e4468062b26baffeaa4e

Figure 13 - List of Indicators of Compromise (IOCs) (Cont.)

Examining the modified executable by the script extracted from the employee's computer using the antivirus Virtustotal, confirms that the sample is flagged as Trojan Ransomware under the name "TeslaCrypt" as shown in Figure 14.

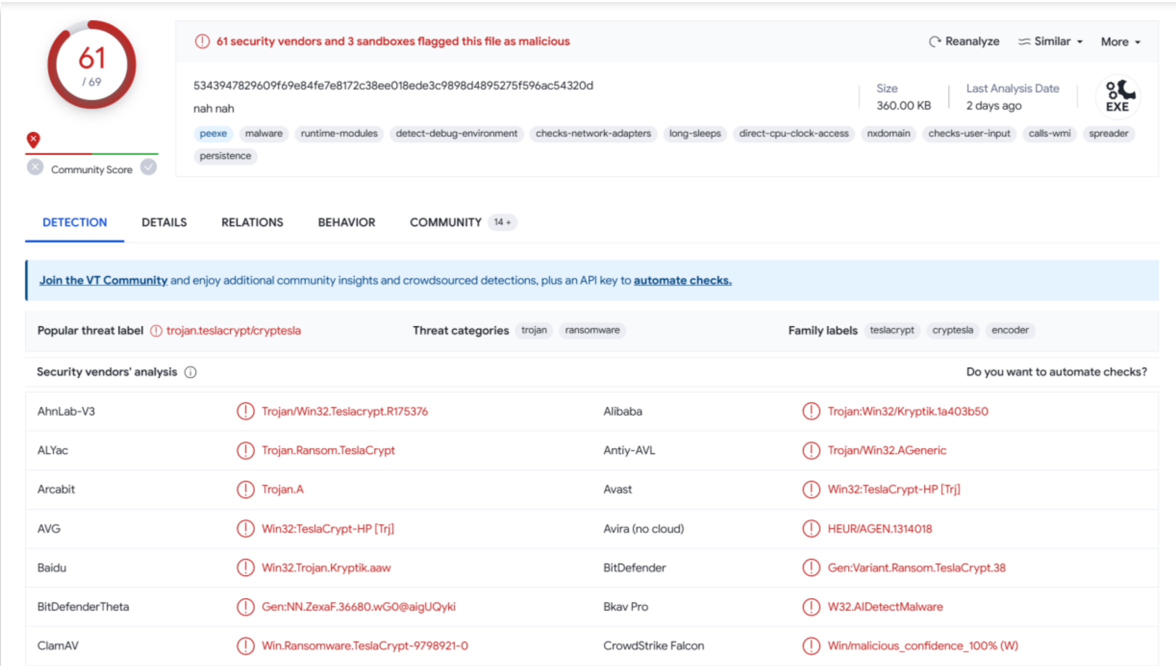


Figure 14 - VirusTotal Scan Result for Malware Sample

PE Header

To understand the malware's functionality, the PE Header of the unpacked malware was analysed instead of the modified sample as it will reveal much more important information such as all required libraries, directories and imports necessary for the operating system (OS) to run the malicious program. Figure 15 showcases all the Windows API calls the malicious program made, it requires the use of tree important APIs; "mspr.dll" (Multiple Provider Router), "psapi.dll" (Process Status API), and "winet.dll" (Windows Internet). The combination of these APIs provides potential risks to the employee's environment and to DodoSOC organisation. For instance, using the "mspr.dll" allows the malware to manage network connections and resources, which could lead to the exfiltration of sensitive company data, download additional payloads or establish malicious network connections.

library (12)	flag (3)	first-thunk-original (INT)	first-thunk (IAT)	imports (150)	group	description
mpr.dll	x	0x0009E1CC	0x000311CC	3	network	Multiple Provider Router Library
psapi.dll	x	0x0009E1DC	0x000311DC	2	execution	Process Status Library
wininet.dll	x	0x0009E218	0x00031218	5	network	Internet Extensions for Win32 Library
advapi32.dll	-	0x0009E000	0x00031000	15	-	Advanced Windows 32 Base API
gdi32.dll	-	0x0009E040	0x00031040	9	-	GDI Client Library
kernel32.dll	-	0x0009E068	0x00031068	88	-	Windows NT BASE API Client
shell32.dll	-	0x0009E1E8	0x000311E8	3	-	Windows Shell Library
shlwapi.dll	-	0x0009E1F8	0x000311F8	2	-	Shell Light-weight Utility Library
user32.dll	-	0x0009E204	0x00031204	4	-	Multi-User Windows USER API Client Library
GdiPlus.dll	-	0x0009E230	0x00031230	9	-	Microsoft GDI+ Library
ntdll.dll	-	0x0009E258	0x00031258	9	-	NT Layer
ole32.dll	-	0x0009E280	0x00031280	1	-	Microsoft OLE for Windows

Figure 15 - Extracted Windows API used by Malware.

Figure 16 breaks down all the flagged IAT imports extracted from the PE Header of the malicious program (appendix C). Based on the imports potential impact, the IAT were divided into five manipulation categories; token, registry, file system, network, and shell execution to illustrate how a threat actor intentionally used them for malicious purposes.

Category	IAT Function Name	Impact	MITRE ATT&CK Technique ID
Token Manipulation	"AdjustTokenPrivileges" "CheckTokenMembership"	Can be used to manipulate security tokens of a running process and bypass security mechanisms or elevate privileges to perform malicious actions with elevated permissions.	T1134
Registry Manipulation	"RegSetValueExW" "RegFlushKey" "RegCreateKeyExA/ RegCreateKeyExW"	Can be used for malware persistence, configuration changes or hiding its presence by creating, modifying, or deleting registry keys.	T1112, T1070
File System Manipulation	DeleteFileW SetFileAttributesW MoveFileExW	Can be use for hiding, deleting or moving files to different locations to affect the victim's system stability or hide malware traces.	T1485, T1106, T1543
Network Operations	"InternetOpenA" "InternetCrackUrlA" "HttpSendRequestA" "InternetSetOptionA" "InternetCloseHandle"	Allows the malware to communicate with a remote server, download / upload / exfiltrate data.	T1659, T1105, T1485, T1041
Shell Execution	"ShellExecuteExW"	Allows malware to execute other programs or scripts, potentially downloading and running additional payloads.	T1106

Figure 16 - Malicious IAT Functions

The identified functions in the IAT suggest that the malicious program can perform, privilege escalation, registry and file system manipulation, network communication, and process control. Using a combination of MITRE ATT&CKs techniques such as T1112, T1485, T1070 a threat actor can use legitimate built in Windows API functions such as “AdjustTokenPrivileges”, “RegSetValueExW” or “DeleteFileW” to interact with the Windows Registry, elevate privileges by manipulating access tokens of running processes and bypass security access controls, modify artifacts to hide or remove evidence of their presence or even execute malicious payloads to encrypt or delete files in the employee’s system.

Dynamic Analysis

Process Monitor / Explorer	ApateDNS	INetSim	Wireshark	Regshot	ApiLogger

Figure 17 - Tools utilised for Dynamic Malware Analysis

Dynamic analysis techniques were used to understand the malware’s behaviour. To initiate the analysis, a virtual network was set up to allow the malware run in a sandboxed environment. The virtual network contains two hosts; a malware analysis Windows Virtual Machine (VM) and a Linux VM running INetSim as illustrated in Figure 18. The Linux machine emulates many ports, including HTTP, HTTPS and FTP to look like a real server. This was done to record all inbound requests and connections to determine the malware’s network behaviour. The Windows VM listens on port “53” for DNS requests using ApateDNS, by configuring the DNS server to localhost (“127.0.0.1”), ApateDNS could redirect the traffic to the Linux VM located at address “192.168.100.5”, tricking the malware to communicate with the fake server and enabling the extraction of network IOC footprints, such as DNS names, IP addresses and packet signatures.

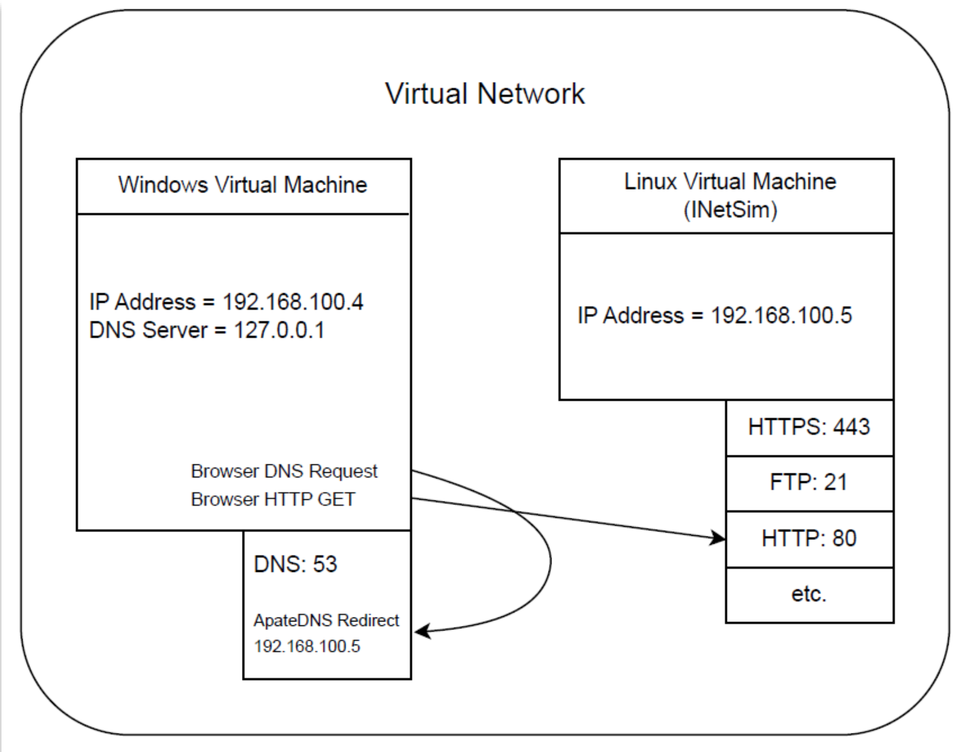


Figure 18 - Virtual Network for Dynamic Analysis

A registry snapshot was taken using RegShot before and after running the unpacked malware sample to compare and determine registry changes. At this point, the malware sample was run, and after some time the event captures were stopped to being the analysis as follows:

1. Network Connections:

Using ApateDNS reveals all the redirected DNS requests made by the malware (appendix C). It indicates that, once the malware program runs, it makes DNS requests to malicious domains as shown in Figure 19, which have been detected by VirusTotal as suspicious URLs (appendix C).

Artifact	MD5 Hash
1. biocarbon.com.ec	4e4f0a7655bf67d3e6b3551ad8569414
2. imagescroll.com	ed7e8d9f82e4b1a7b07b9b3e8fb02b58
3. music.mbsaeger.com	6ef7e08ddd1333e9b9a0928f3b41e349
4. surrogacyandadoption.com	edb98e67c4c93a6678ac89303d8fc363
5. stacon.eu	0fe918b14ce67d913cdf810efd58a504
6. worldisonefamily.info	c657a6b8b758d7ddd6c97043ffee7fc9

Figure 19 – Network Indicators of Compromise

Following the TCP stream of the malware using Wireshark to understand the network activity of the malicious program, Figure 20, reveals that the malware makes a DNS request to "stacon.eu" and then gets a HTTP POST response containing unknown data. These behaviours were captured as potential IOC of malicious activity, including: the domain name "statcon.eu", the POST response metadata, and the data received as a network fingerprint as shown in Figure 21.

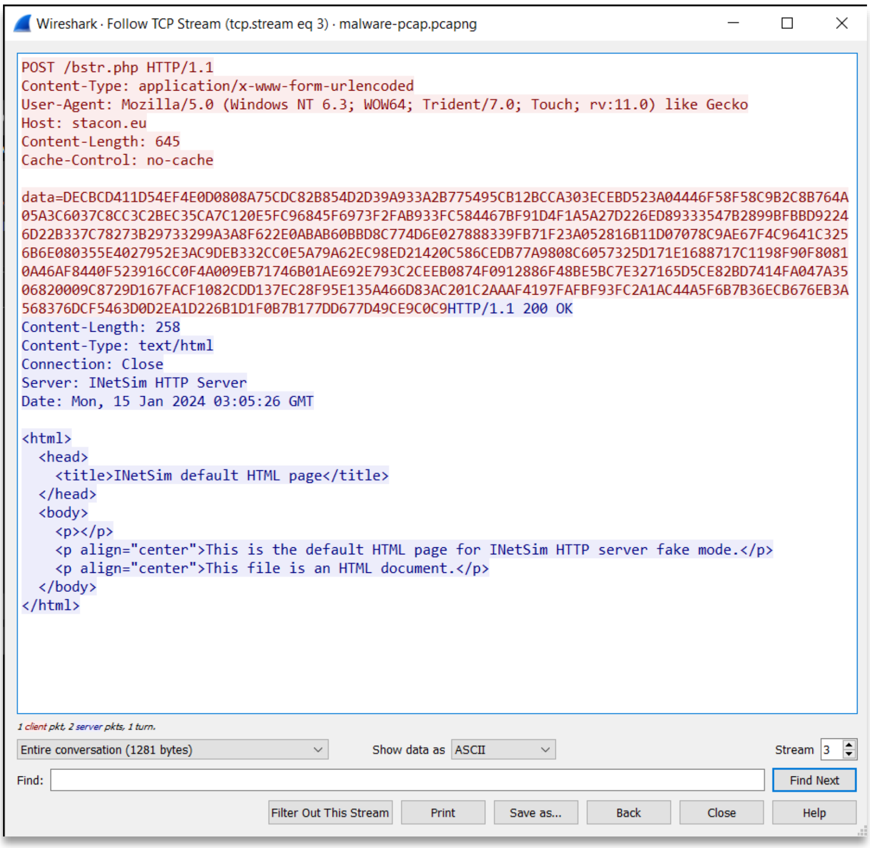


Figure 20 - TCP Stream of Malware using Wireshark

Artifact	MD5 Hash
7. HTTP POST response data	284e228654d09c22721be3c3e65af473

Figure 21 - Network Indicators of Compromise (Cont.)

2. Process Injection:

Figure 22, shows the process tree of all the processes created after the malware sample was ran to analyse its behaviour during runtime. First, the parent process “MSUpdate_dump_SYC.exe” (malware) beings its execution, and spawns a child process “rtisstssrcqj.exe” (malware copy) to replicate itself under the “C:\Windows” directory. After, the malware copy runs, it executes the “WMIC” Windows utility to delete its shadow copy using the “noniteractive” option to not prompt the user for confirmation before deleting its shadow copy. Finally, the malware invokes the command prompt to perform a “DEL” operation which deletes the malware from whichever location is in the system.

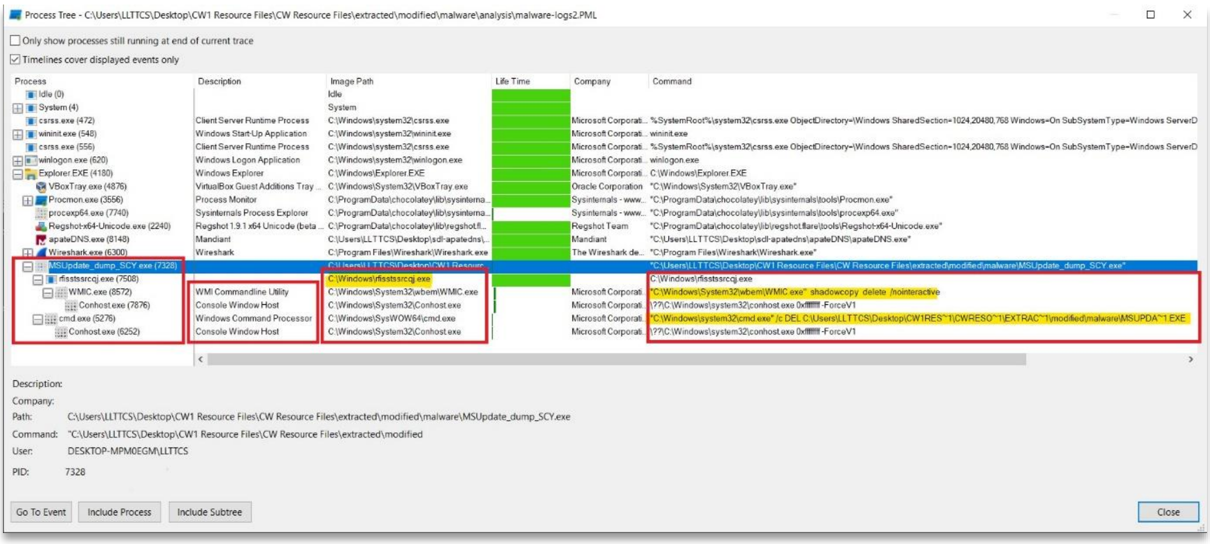


Figure 22 - Procom Process Tree

Using the operation filter ‘WriteFile” of Procom (Process Monitor) to identify file system modifications, confirms that the malware writes a file named “rfisstssrcqj.exe” at the “C:\Windows” directory. Upon comparing the malware sample’s MD5 file hash against the “rfisstssrcqj.exe” executable, the results show that they are identical, determining that the malware copies itself to that location for persistence mechanisms (see appendix C).

3. Registry Changes:

Comparing the two registry snapshots taken with Regshot to identify changes, reveals that 8 registry keys were deleted, and 89 new registry keys were added (see appendix C). By examining the log file, the malware program installs the autorun registry value “rfisstssrcqj” at “HKLM\SOFTWARE\Microsoft\Windows\CurrentVersion\Run”). The data written to that value is where the malicious program copied itself (“C:\Windows\rfisstssrcqj.exe”) as persistence mechanism, as that newly copied binary will execute upon system reboot.

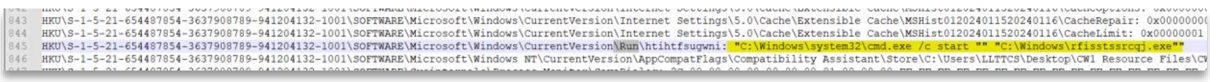


Figure 23 - RegShot log file

4. Windows API Calls:

Figure 24, showcases the Windows API logs made by the malware during its execution using ApiLogger to analyse the way that malware interacts with the OS. Running the malware for a second time, reveals another persistence technique used by the malware. In this case, the malware changes its child process name every time it runs, indicating that the file name written to “C:\Windows” will change to avoid being detected by antivirus or other preventive mechanisms that may be implemented.

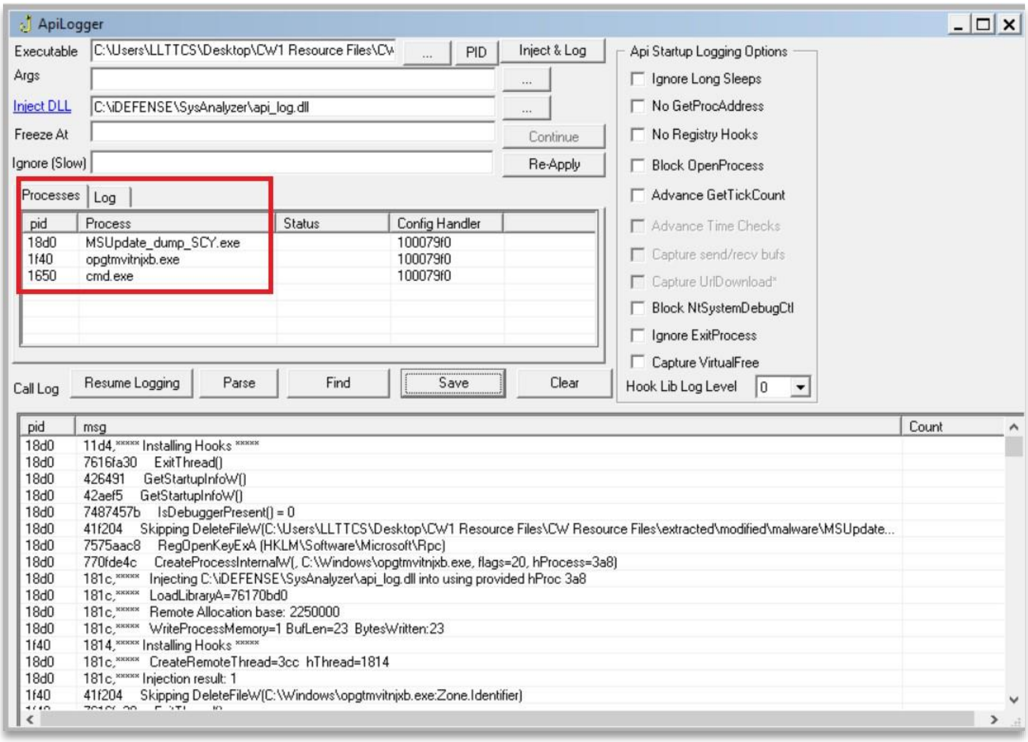


Figure 24 - ApiLogger Log File

Chain of Events

To complete the malware analysis, the MITRE ATT&CK Framework is implemented to get an overview of the chain of events and the threat landscape of the malware is illustrated in Figure 25.

ATT&CK Tactic	Description	ATT&CK Technique ID
Persistence	Creates an AutoStart registry key pointing to a malicious binary in “C:\Windows”	T1547.001
Privilege Escalation	Spawns process in suspended mode to inject arbitrary code	T156.001
Defence Evasion	Obfuscate child processes and drops bath files with force delete cmd (self deletion)	T1036, T1027
Discovery	Queries system information (incl. serial number, name, etc), enumerates the file system and monitors registry keys	T012, T1082, T1083
Impact	Encodes data using XOR, writes notice file (html, txt) to demand a ransom. (appendix D)	T1486

Figure 25 - MITRE ATT&CK for Chain of Events

Prevention & Automation

YARA rules were used to transform the listed IOC from the malware analysis stage to identify and prevent similar infections in DodoSOC corporate network in the future as shown in Figure 26.

```
rule TrojanTeslaCryptA
{
    meta:
        Description = "Trojan.TeslaCrypt.A"
        type = "Header"

    strings:
        $signature = {4D 50}
        $hash = "9ce01dfbf25dfea778e57d8274675d6f"

    condition:
        signature and hash


}
rule TrojanTeslaCryptB
{
    meta:
        Description = "Trojan.TeslaCrypt.B"
        type = "File System"

    strings:
        $ = "RECOVER.TXT" ascii wide nocase
        $ = "RECOVERY.PNG" ascii wide nocase
        $ = "RECOVERY.HTML" ascii wide nocase
    condition:
        any of them


}
rule TrojanTeslaCryptC
{
    meta:
        Description = "Ransom.TeslaCrypt.C"
        Type = "Network Traffic"

    strings:
        $a1 = "bicarbon.com.ec"
        $a2 =  /\/(.*?)\/wp-content/uploads/bstr.php/

        $b1 = "imagescroll.com"
        $b2 = /\/(.*?)\/cgi-bin/Templates/bstr.php/

        $c1 = "music.mbsaeger.com"
        $c2 = /\/(.*?)\/music/Glee/bstr.php/

        $d1 = "stacon.eu"
        $d2 = /\/(.*?)\/bstr.php/

        $e1 = "surrogacyandadoption.com"
        $e2 = /\/(.*?)\/bstr.php/

        $f1 = "worldisonefamily.info"
        $f2 = /\/(.*?)\/zz/libraries/bstr.php/

    condition:
        ($a1 and $a2) or ($b1 and $b2) or ($c1 and $c2) or ($d1 and $d2) or ($e1 and $e2) or ($f1 and $f2)

}
```

Figure 26 - YARA Rules

References

alvinashcraft (2023). *Security and Identity - Win32 Apps*. [online] learn.microsoft.com. Available at: https://learn.microsoft.com/en-us/windows/win32/api/_security/ [Accessed 14 Jan. 2024].

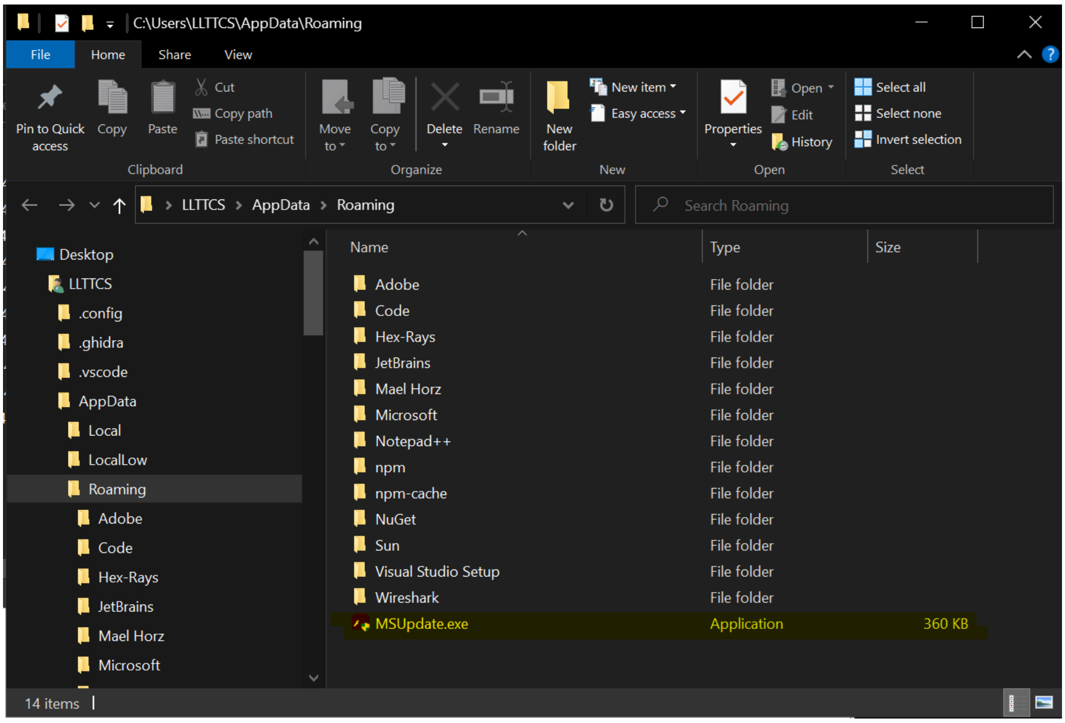
Bermejo Higuera, J., Abad Aramburu, C., Bermejo Higuera, J.-R., Sicilia Urban, M.A. and Sicilia Montalvo, J.A. (2020). Systematic Approach to Malware Analysis (SAMA). *Applied Sciences*, 10(4), p.1360. doi:https://doi.org/10.3390/app10041360.

Fox , N. (2022). *How to Unpack Malware with x64dbg*. [online] www.varonis.com. Available at: https://www.varonis.com/blog/x64dbg-unpack-malware.

QuinnRadich (2021). *RegFlushKey Function (winreg.h) - Win32 Apps*. [online] learn.microsoft.com. Available at: https://learn.microsoft.com/en-us/windows/win32/api/winreg/nf-winreg-regflushkey.

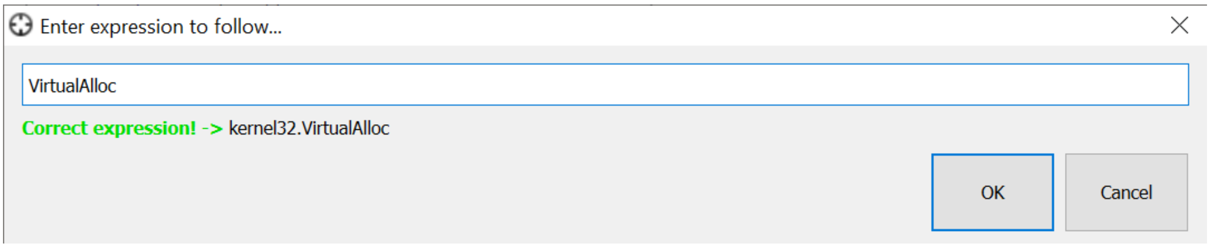
Sikorski, M. and Honig, A. (2012). *Practical Malware Analysis : the hands-on Guide to Dissecting Malicious Software*. San Francisco No Starch Press.

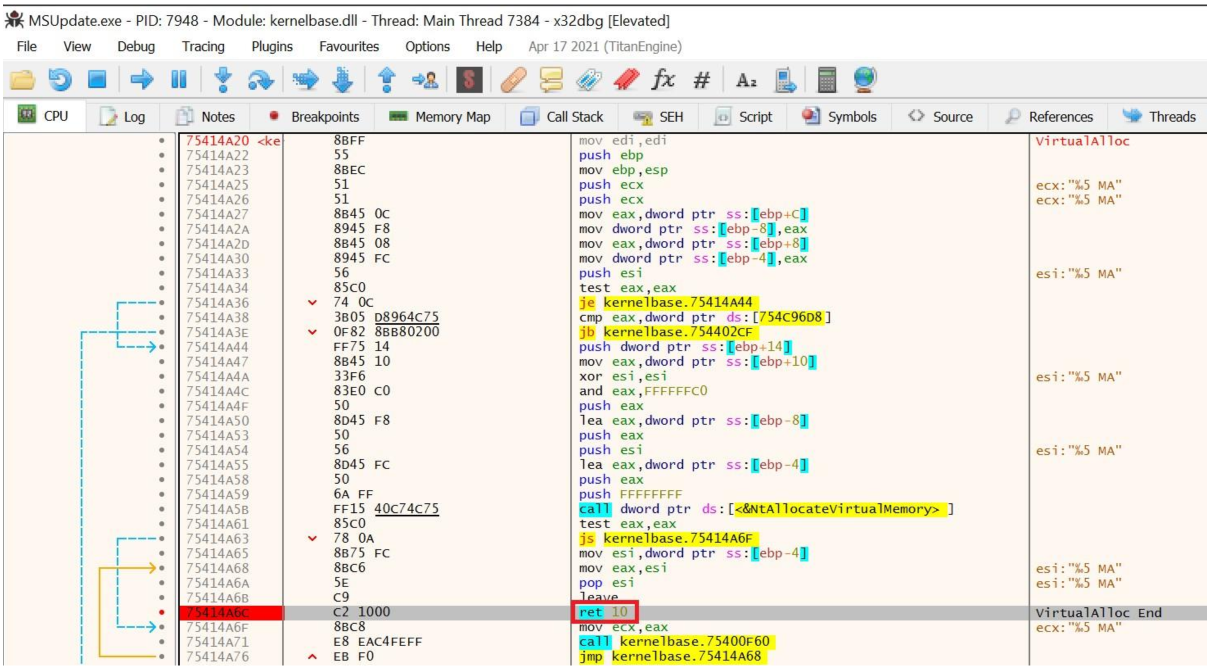
Appendix A

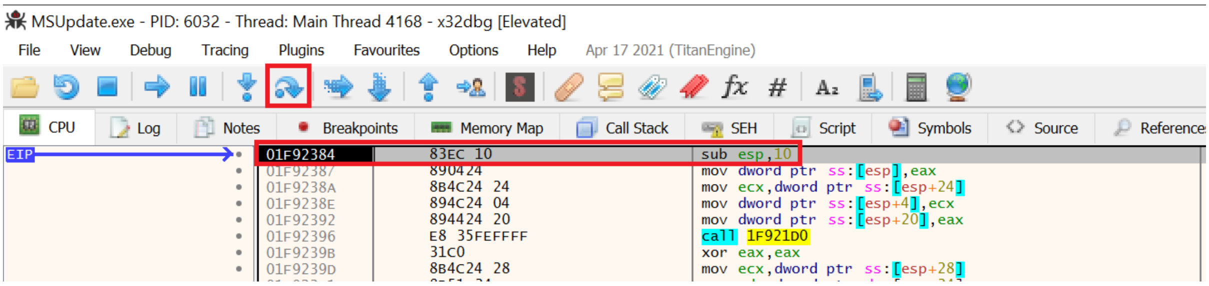


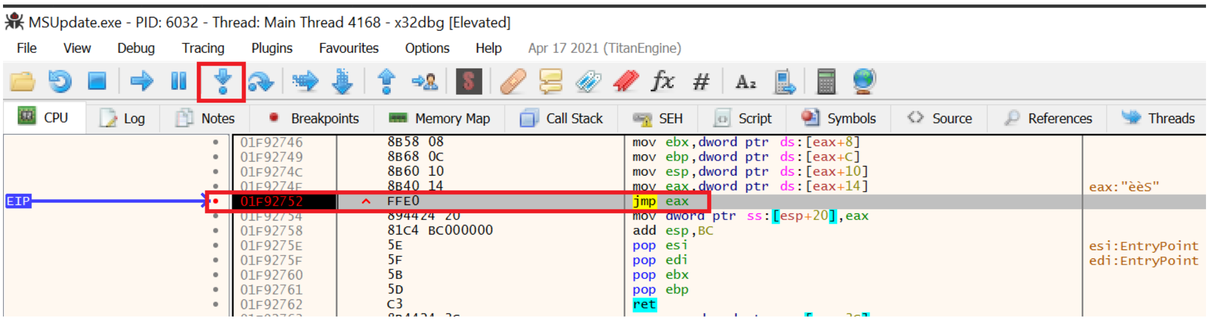
C:\Users\LLTTCS\AppData\Roaming
File
Home
Share
View
Pin to Quick access
Copy
Paste
Cut
Copy path
Paste shortcut
Clipboard
Move to
Copy to
Delete
Rename
Organize
New folder
New item
Easy access
New
Properties
Open
Edit
History
Open
Select all
Select none
Invert selection
Select
LLTTCS › AppData › Roaming
Search Roaming
Desktop
LLTTCS
.config
.ghidra
.vscode
AppData
Local
LocalLow
Roaming
Adobe
Code
Hex-Rays
JetBrains
Mael Horz
Microsoft
Name
Type
Size
Adobe File folder
Code File folder
Hex-Rays File folder
JetBrains File folder
Mael Horz File folder
Microsoft File folder
Notepad++ File folder
npm File folder
npm-cache File folder
NuGet File folder
Sun File folder
Visual Studio Setup File folder
Wireshark File folder
MSUpdate.exe Application 360 KB
14 items

Appendix B


Enter expression to follow...
VirtualAlloc
Correct expression! -> kernel32.VirtualAlloc
OK
Cancel

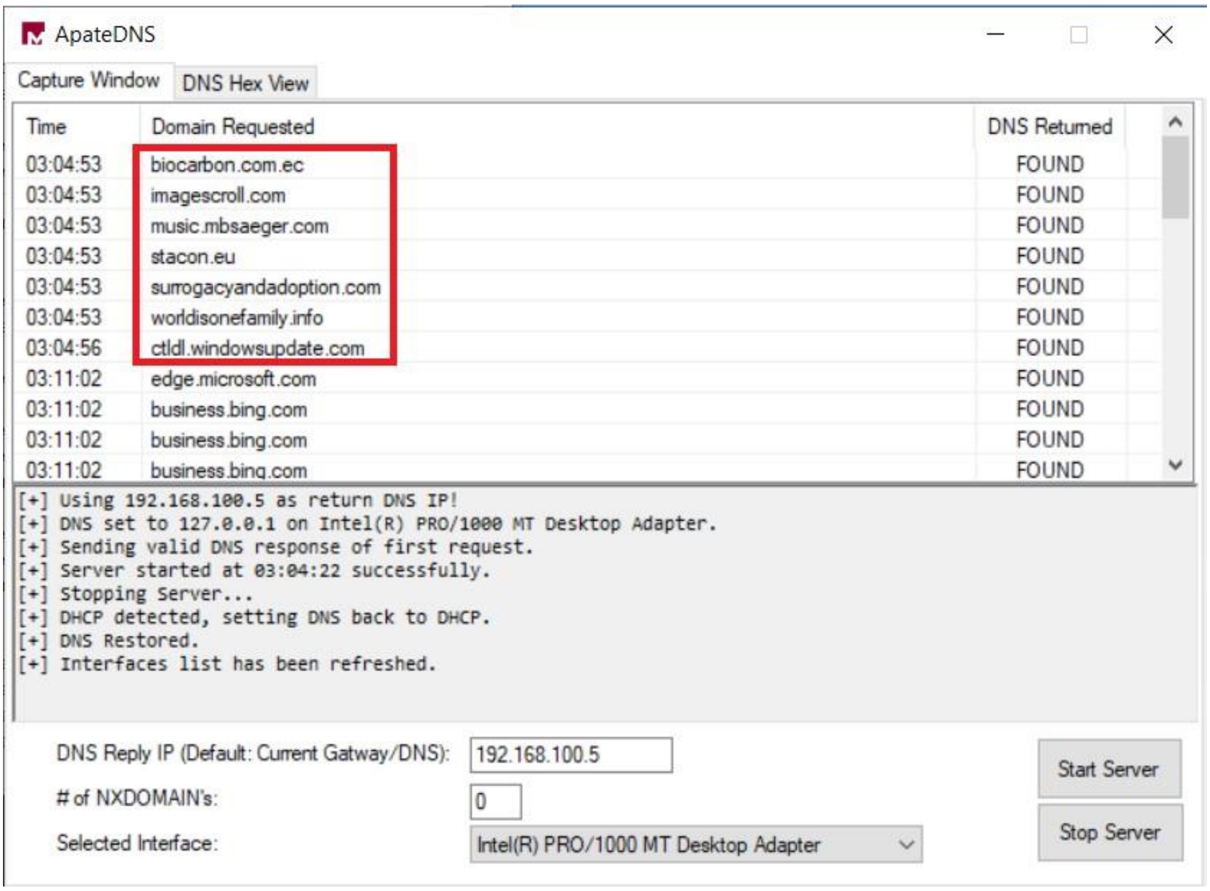

MSUpdate.exe - PID: 7948 - Module: kernelbase.dll - Thread: Main Thread 7384 - x32dbg [Elevated]
File View Debug Tracing Plugins Favourites Options Help Apr 17 2021 (TitanEngine)
CPU Log Notes Breakpoints Memory Map Call Stack SEH Script Symbols Source References Threads
VirtualAlloc
call dword ptr ds:[<&NtAllocateVirtualMemory>]
ret 10
VirtualAlloc End

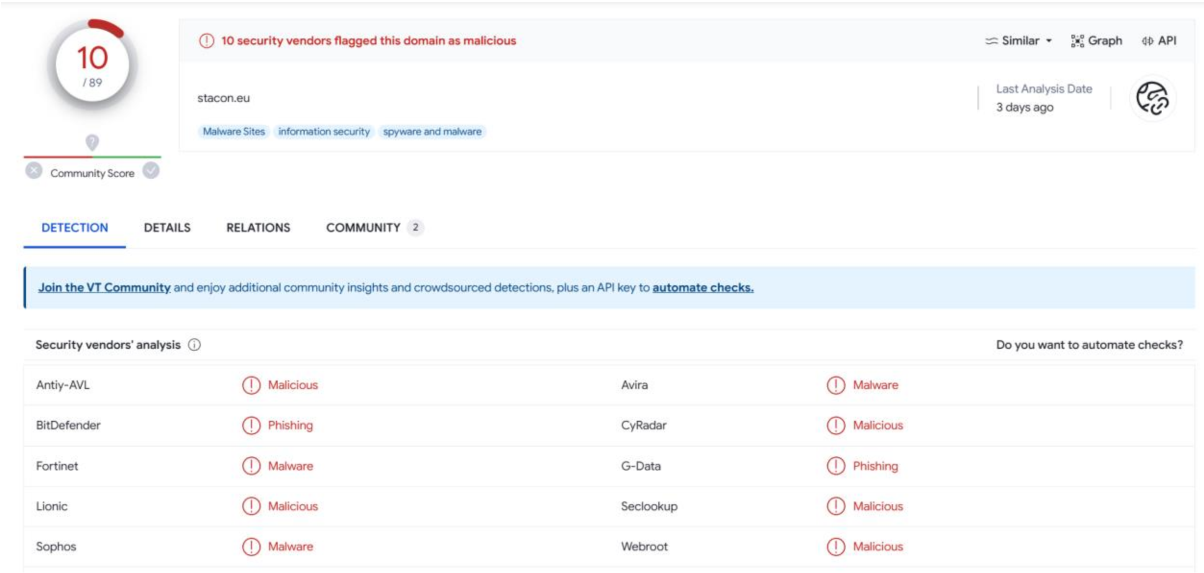

MSUpdate.exe - PID: 6032 - Thread: Main Thread 4168 - x32dbg [Elevated]
File View Debug Tracing Plugins Favourites Options Help Apr 17 2021 (TitanEngine)
CPU Log Notes Breakpoints Memory Map Call Stack SEH Script Symbols Source Reference
EIP
01F92384 83EC 10 sub esp,10

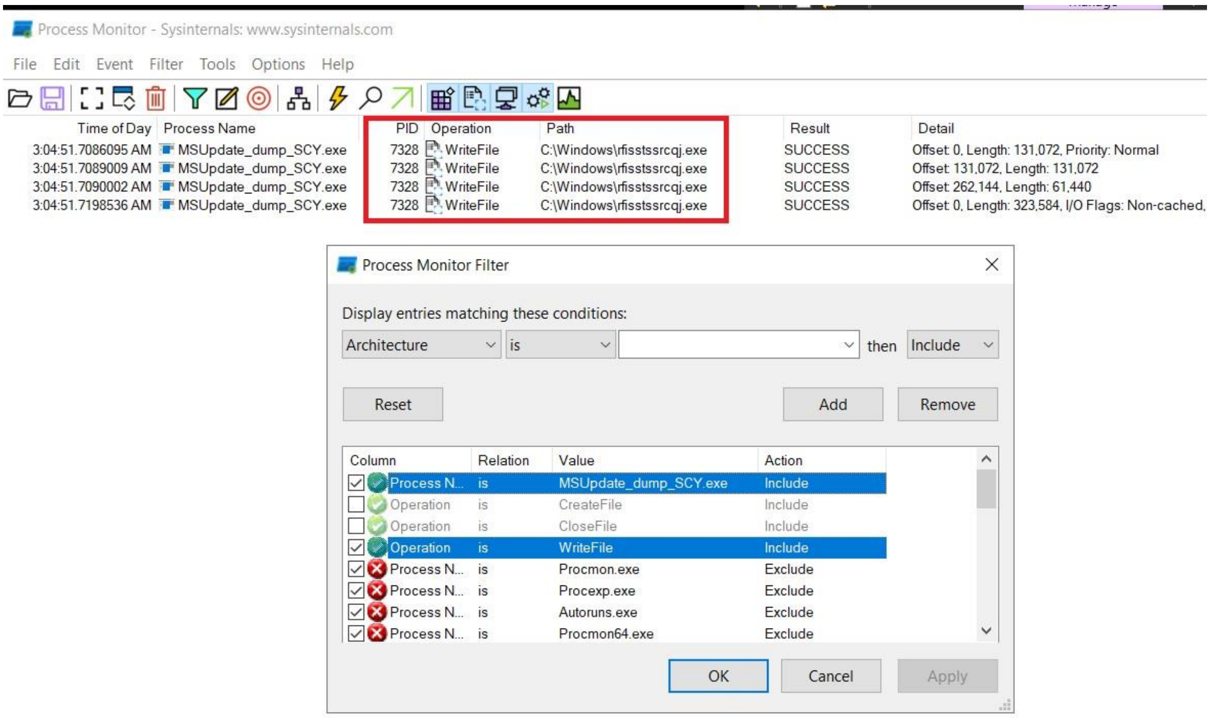

MSUpdate.exe - PID: 6032 - Thread: Main Thread 4168 - x32dbg [Elevated]
File View Debug Tracing Plugins Favourites Options Help Apr 17 2021 (TitanEngine)
CPU Log Notes Breakpoints Memory Map Call Stack SEH Script Symbols Source References Threads
EIP
01F92752 FFE0 jmp eax

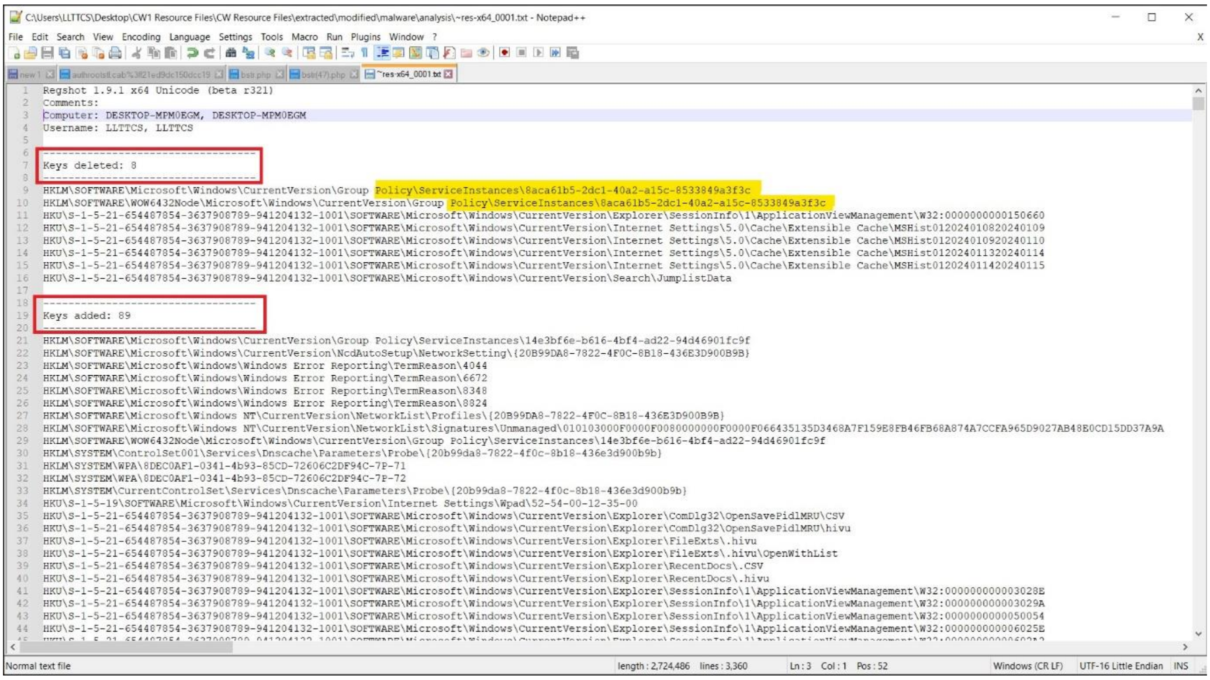
Appendix C

imports (150)	flag (36)	callback (0)	first-thunk-original (INT)	first-thunk (IAT)	hint	group (12)	technique (11)	type (1)	ordinal (0)	library (12)
AdjustTokenPrivileges	x	-	0x0009E3AC	0x1E376C08	31 (0x001F)	security	T1134 | Access Token Manipulation	implicit	-	advapi32.dll
CheckTokenMembership	x	-	0x0009E3C4	0xDF8EEB99	95 (0x005F)	security	T1134 | Access Token Manipulation	implicit	-	advapi32.dll
FreeSid	x	-	0x0009E3DB	0x2748774C	308 (0x0134)	security	-	implicit	-	advapi32.dll
AllocateAndInitializeSid	x	-	0x0009E3E5	0xE19B48A8	32 (0x0020)	security	-	implicit	-	advapi32.dll
LookupPrivilegeValueA	x	-	0x0009E400	0x34B0BCB5	430 (0x01AE)	security	T1134 | Access Token Manipulation	implicit	-	advapi32.dll
GetSidSubAuthority	x	-	0x0009E42E	0x391C0CB3	364 (0x016C)	security	-	implicit	-	advapi32.dll
OpenProcessToken	x	-	0x0009E443	0xE3418ACB	533 (0x0215)	security	T1134 | Access Token Manipulation	implicit	-	advapi32.dll
RegSetValueExW	x	-	0x0009E456	0x4ED8AA4A	681 (0x02A9)	registry	T1112 | Modify Registry	implicit	-	advapi32.dll
RegFlushKey	x	-	0x0009E475	0x5B9CCA4F	638 (0x027E)	registry	T1112 | Modify Registry	implicit	-	advapi32.dll
RegCreateKeyExA	x	-	0x0009E483	0xD6B2B8A3	611 (0x0263)	registry	T1112 | Modify Registry	implicit	-	advapi32.dll
RegCreateKeyExW	x	-	0x0009E4A8	0x5DEFB2FC	612 (0x0264)	registry	T1112 | Modify Registry	implicit	-	advapi32.dll
GetLogicalDriveStringsW	x	-	0x0009E6AF	0x28DB77F5	618 (0x026A)	reconnaissance	-	implicit	-	kernel32.dll
GetCurrentProcessId	x	-	0x0009E6C9	0x40C72493	539 (0x021B)	reconnaissance	T1057 | Process Discovery	implicit	-	kernel32.dll
GetEnvironmentVariableW	x	-	0x0009E6DF	0x32CAAB7B	572 (0x023C)	reconnaissance	-	implicit	-	kernel32.dll
WNetEnumResourceW	x	-	0x0009EB76	0x34B0BCB5	37 (0x0025)	network	-	implicit	-	mpr.dll
WNetOpenEnumW	x	-	0x0009EB8A	0x391C0CB3	70 (0x0046)	network	-	implicit	-	mpr.dll
WNetCloseEnum	x	-	0x0009EB9A	0x4ED8AA4A	25 (0x0019)	network	-	implicit	-	mpr.dll
InternetOpenA	x	-	0x0009ECA4	0x00000040	198 (0x00C6)	network	-	implicit	-	wininet.dll
InternetCrackUrlA	x	-	0x0009ECB4	0x00000041	158 (0x009E)	network	-	implicit	-	wininet.dll
HttpSendRequestA	x	-	0x0009ECC8	0x00000014	127 (0x007F)	network	-	implicit	-	wininet.dll
InternetSetOptionA	x	-	0x0009ECDB	0x0000000C	220 (0x00DC)	network	-	implicit	-	wininet.dll
InternetCloseHandle	x	-	0x0009ECF0	0x0041B2B0	149 (0x0095)	network	-	implicit	-	wininet.dll
GlobalMemoryStatus	x	-	0x0009E654	0xBEF90DAE	828 (0x033C)	memory	-	implicit	-	kernel32.dll
WriteFile	x	-	0x0009E57E	0xE372532B	1559 (0x0617)	file	-	implicit	-	kernel32.dll
DeleteFileW	x	-	0x0009E619	0x06F067AA	280 (0x0118)	file	T1485 | Data Destruction	implicit	-	kernel32.dll
SetFileAttributesW	x	-	0x0009E627	0xA2C898A6	1311 (0x051F)	file	-	implicit	-	kernel32.dll
MoveFileExW	x	-	0x0009E82C	0xB5C0FBCF	1004 (0x03EC)	file	T1105 | Remote File Copy	implicit	-	kernel32.dll
PathFindExtensionW	x	-	0x0009EC36	0xBEF9A3F7	75 (0x004B)	file	-	implicit	-	shlwapi.dll
PathFindFileNameW	x	-	0x0009EC4B	0xC67178F2	77 (0x004D)	file	-	implicit	-	shlwapi.dll
CreateProcessW	x	-	0x0009E6F9	0x15C9BEBC	232 (0x00E8)	execution	T1106 | Execution through API	implicit	-	kernel32.dll
GetEnvironmentStringsW	x	-	0x0009E8D1	0x72BE5D74	570 (0x023A)	execution	-	implicit	-	kernel32.dll
GetCurrentThreadId	x	-	0x0009E9BD	0xA831C66D	543 (0x021F)	execution	T1057 | Process Discovery	implicit	-	kernel32.dll
TerminateProcess	x	-	0x0009EB0E	0xF40E3585	1424 (0x0590)	execution	-	implicit	-	kernel32.dll
GetProcessImageFileNameW	x	-	0x0009EBB4	0x682E6FF3	20 (0x0014)	execution	-	implicit	-	psapi.dll
EnumProcesses	x	-	0x0009EBCF	0x748F82EE	6 (0x0006)	execution	T1057 | Process Discovery	implicit	-	psapi.dll
ShellExecuteExW	x	-	0x0009EBEB	0x84C87814	430 (0x01AE)	execution	T1106 | Execution through API	implicit	-	shell32.dll
WaitForSingleObject	-	-	0x0009E71E	0x9C100D4C	1499 (0x05DB)	synchronization	-	implicit	-	kernel32.dll




10
/ 89
10 security vendors flagged this domain as malicious
Similar
Graph
API
stacon.eu
Malware Sites
information security
spyware and malware
Last Analysis Date
3 days ago
Community Score
DETECTION
DETAILS
RELATIONS
COMMUNITY 2
Join the VT Community and enjoy additional community insights and crowdsourced detections, plus an API key to automate checks.
Security vendors' analysis
Do you want to automate checks?
Antiy-AVL Malicious
Avira Malware
BitDefender Phishing
CyRadar Malicious
Fortinet Malware
G-Data Phishing
Lionic Malicious
Seclookup Malicious
Sophos Malware
Webroot Malicious


Process Monitor - Sysinternals: www.sysinternals.com
File Edit Event Filter Tools Options Help
Time of Day Process Name PID Operation Path Result Detail
3:04:51.7086095 AM MSUpdate_dump_SCY.exe 7328 WriteFile C:\Windows\rfisstssrcqj.exe SUCCESS Offset: 0, Length: 131,072, Priority: Normal
3:04:51.7089009 AM MSUpdate_dump_SCY.exe 7328 WriteFile C:\Windows\rfisstssrcqj.exe SUCCESS Offset: 131,072, Length: 131,072
3:04:51.7090002 AM MSUpdate_dump_SCY.exe 7328 WriteFile C:\Windows\rfisstssrcqj.exe SUCCESS Offset: 262,144, Length: 61,440
3:04:51.7198536 AM MSUpdate_dump_SCY.exe 7328 WriteFile C:\Windows\rfisstssrcqj.exe SUCCESS Offset: 0, Length: 323,584, I/O Flags: Non-cached,
Process Monitor Filter
Display entries matching these conditions:
Architecture is then Include
Reset
Add
Remove
Column Relation Value Action
Process N... is MSUpdate_dump_SCY.exe Include
Operation is CreateFile Include
Operation is CloseFile Include
Operation is WriteFile Include
Process N... is Procmon.exe Exclude
Process N... is Procexp.exe Exclude
Process N... is Autoruns.exe Exclude
Process N... is Procmon64.exe Exclude
OK
Cancel
Apply

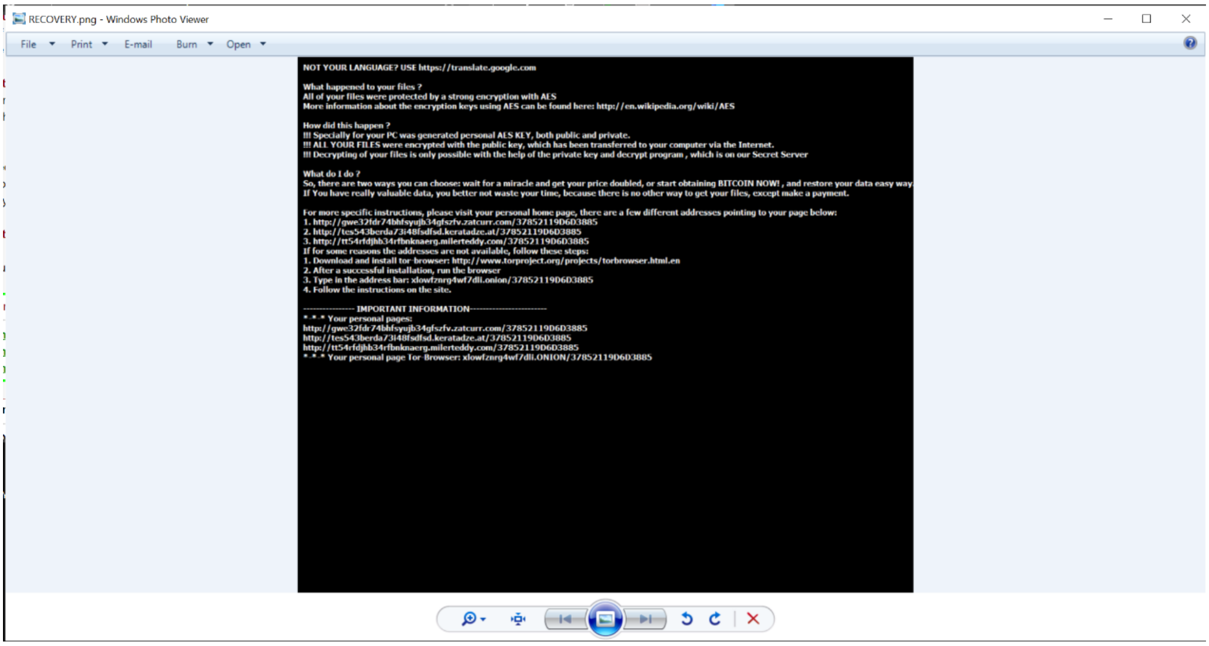


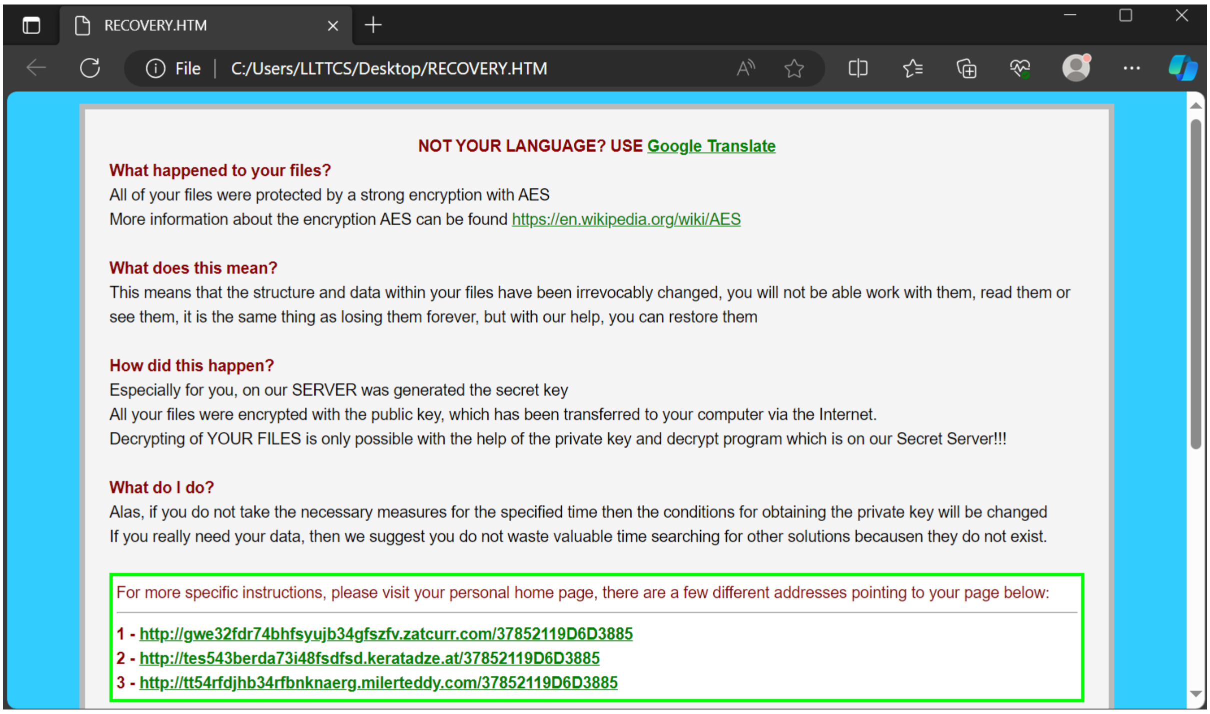

```
Regshot 1.9.1 x64 Unicode (beta r321)
Comments:
Computer: DESKTOP-MPM0EGM, DESKTOP-MPM0EGM
Username: LLTTCS, LLTTCS

----------------------------------
Keys deleted: 8
----------------------------------
HKLM\SOFTWARE\Microsoft\Windows\CurrentVersion\Group Policy\ServiceInstances\8aca61b5-2dc1-40a2-a15c-8533849a3f3c
HKLM\SOFTWARE\WOW6432Node\Microsoft\Windows\CurrentVersion\Group Policy\ServiceInstances\8aca61b5-2dc1-40a2-a15c-8533849a3f3c
HKU\S-1-5-21-654487854-3637908789-941204132-1001\SOFTWARE\Microsoft\Windows\CurrentVersion\Explorer\SessionInfo\1\ApplicationViewManagement\W32:0000000000150660
HKU\S-1-5-21-654487854-3637908789-941204132-1001\SOFTWARE\Microsoft\Windows\CurrentVersion\Internet Settings\5.0\Cache\Extensible Cache\MSHist012024010820240109
HKU\S-1-5-21-654487854-3637908789-941204132-1001\SOFTWARE\Microsoft\Windows\CurrentVersion\Internet Settings\5.0\Cache\Extensible Cache\MSHist012024010920240110
HKU\S-1-5-21-654487854-3637908789-941204132-1001\SOFTWARE\Microsoft\Windows\CurrentVersion\Internet Settings\5.0\Cache\Extensible Cache\MSHist012024011320240114
HKU\S-1-5-21-654487854-3637908789-941204132-1001\SOFTWARE\Microsoft\Windows\CurrentVersion\Internet Settings\5.0\Cache\Extensible Cache\MSHist012024011420240115
HKU\S-1-5-21-654487854-3637908789-941204132-1001\SOFTWARE\Microsoft\Windows\CurrentVersion\Search\JumplistData

----------------------------------
Keys added: 89
----------------------------------
HKLM\SOFTWARE\Microsoft\Windows\CurrentVersion\Group Policy\ServiceInstances\14e3bf6e-b616-4bf4-ad22-94d46901fc9f
HKLM\SOFTWARE\Microsoft\Windows\CurrentVersion\NcdAutoSetup\NetworkSetting\{20B99DA8-7822-4F0C-8B18-436E3D900B9B}
HKLM\SOFTWARE\Microsoft\Windows\Windows Error Reporting\TermReason\4044
HKLM\SOFTWARE\Microsoft\Windows\Windows Error Reporting\TermReason\6672
HKLM\SOFTWARE\Microsoft\Windows\Windows Error Reporting\TermReason\8348
HKLM\SOFTWARE\Microsoft\Windows\Windows Error Reporting\TermReason\8824
HKLM\SOFTWARE\Microsoft\Windows NT\CurrentVersion\NetworkList\Profiles\{20B99DA8-7822-4F0C-8B18-436E3D900B9B}
HKLM\SOFTWARE\Microsoft\Windows NT\CurrentVersion\NetworkList\Signatures\Unmanaged\010103000F0000F0080000000F0000F066435135D3468A7F159E8FB46FB68A874A7CCFA965D9027AB48E0CD15DD37A9A
HKLM\SOFTWARE\WOW6432Node\Microsoft\Windows\CurrentVersion\Group Policy\ServiceInstances\14e3bf6e-b616-4bf4-ad22-94d46901fc9f
HKLM\SYSTEM\ControlSet001\Services\Dnscache\Parameters\Probe\{20b99da8-7822-4f0c-8b18-436e3d900b9b}
HKLM\SYSTEM\WPA\8DEC0AF1-0341-4b93-85CD-72606C2DF94C-7P-71
HKLM\SYSTEM\WPA\8DEC0AF1-0341-4b93-85CD-72606C2DF94C-7P-72
HKLM\SYSTEM\CurrentControlSet\Services\Dnscache\Parameters\Probe\{20b99da8-7822-4f0c-8b18-436e3d900b9b}
HKU\S-1-5-19\SOFTWARE\Microsoft\Windows\CurrentVersion\Internet Settings\Wpad\52-54-00-12-35-00
HKU\S-1-5-21-654487854-3637908789-941204132-1001\SOFTWARE\Microsoft\Windows\CurrentVersion\Explorer\ComDlg32\OpenSavePidlMRU\CSV
HKU\S-1-5-21-654487854-3637908789-941204132-1001\SOFTWARE\Microsoft\Windows\CurrentVersion\Explorer\ComDlg32\OpenSavePidlMRU\hivu
HKU\S-1-5-21-654487854-3637908789-941204132-1001\SOFTWARE\Microsoft\Windows\CurrentVersion\Explorer\FileExts\.hivu
HKU\S-1-5-21-654487854-3637908789-941204132-1001\SOFTWARE\Microsoft\Windows\CurrentVersion\Explorer\FileExts\.hivu\OpenWithList
HKU\S-1-5-21-654487854-3637908789-941204132-1001\SOFTWARE\Microsoft\Windows\CurrentVersion\Explorer\RecentDocs\.CSV
HKU\S-1-5-21-654487854-3637908789-941204132-1001\SOFTWARE\Microsoft\Windows\CurrentVersion\Explorer\RecentDocs\.hivu
HKU\S-1-5-21-654487854-3637908789-941204132-1001\SOFTWARE\Microsoft\Windows\CurrentVersion\Explorer\SessionInfo\1\ApplicationViewManagement\W32:000000000003028E
HKU\S-1-5-21-654487854-3637908789-941204132-1001\SOFTWARE\Microsoft\Windows\CurrentVersion\Explorer\SessionInfo\1\ApplicationViewManagement\W32:000000000003029A
HKU\S-1-5-21-654487854-3637908789-941204132-1001\SOFTWARE\Microsoft\Windows\CurrentVersion\Explorer\SessionInfo\1\ApplicationViewManagement\W32:0000000000050054
HKU\S-1-5-21-654487854-3637908789-941204132-1001\SOFTWARE\Microsoft\Windows\CurrentVersion\Explorer\SessionInfo\1\ApplicationViewManagement\W32:000000000006025E
```

Appendix D


RECOVERY.png - Windows Photo Viewer
File
Print
E-mail
Burn
Open
NOT YOUR LANGUAGE? USE https://translate.google.com
What happened to your files ?
All of your files were protected by a strong encryption with AES
More information about the encryption keys using AES can be found here: http://en.wikipedia.org/wiki/AES
How did this happen ?
!!! Specially for your PC was generated personal AES KEY, both public and private.
!!! ALL YOUR FILES were encrypted with the public key, which has been transferred to your computer via the Internet.
!!! Decrypting of your files is only possible with the help of the private key and decrypt program , which is on our Secret Server
What do I do ?
So, there are two ways you can choose: wait for a miracle and get your price doubled, or start obtaining BITCOIN NOW! , and restore your data easy way
If You have really valuable data, you better not waste your time, because there is no other way to get your files, except make a payment.
For more specific instructions, please visit your personal home page, there are a few different addresses pointing to your page below:
1. http://gwe32fdr74bhfsyujb34gfszfv.zatcurr.com/37852119D6D3885
2. http://tes543berda73i48fsdfsd.keratadze.at/37852119D6D3885
3. http://tt54rfdjhb34rfbnknaerg.milerteddy.com/37852119D6D3885
If for some reasons the addresses are not available, follow these steps:
1. Download and install tor-browser: http://www.torproject.org/projects/torbrowser.html.en
2. After a successful installation, run the browser
3. Type in the address bar: xlowfznrg4wf7dli.onion/37852119D6D3885
4. Follow the instructions on the site.
---------------- IMPORTANT INFORMATION------------------------
--* Your personal pages:
http://gwe32fdr74bhfsyujb34gfszfv.zatcurr.com/37852119D6D3885
http://tes543berda73i48fsdfsd.keratadze.at/37852119D6D3885
http://tt54rfdjhb34rfbnknaerg.milerteddy.com/37852119D6D3885
--* Your personal page Tor-Browser: xlowfznrg4wf7dli.ONION/37852119D6D3885


RECOVERY.HTM
File | C:/Users/LLTTCS/Desktop/RECOVERY.HTM
NOT YOUR LANGUAGE? USE Google Translate
What happened to your files?
All of your files were protected by a strong encryption with AES
More information about the encryption AES can be found https://en.wikipedia.org/wiki/AES
What does this mean?
This means that the structure and data within your files have been irrevocably changed, you will not be able work with them, read them or see them, it is the same thing as losing them forever, but with our help, you can restore them
How did this happen?
Especially for you, on our SERVER was generated the secret key
All your files were encrypted with the public key, which has been transferred to your computer via the Internet.
Decrypting of YOUR FILES is only possible with the help of the private key and decrypt program which is on our Secret Server!!!
What do I do?
Alas, if you do not take the necessary measures for the specified time then the conditions for obtaining the private key will be changed
If you really need your data, then we suggest you do not waste valuable time searching for other solutions becausen they do not exist.
For more specific instructions, please visit your personal home page, there are a few different addresses pointing to your page below:
1 - http://gwe32fdr74bhfsyujb34gfszfv.zatcurr.com/37852119D6D3885
2 - http://tes543berda73i48fsdfsd.keratadze.at/37852119D6D3885
3 - http://tt54rfdjhb34rfbnknaerg.milerteddy.com/37852119D6D3885

RECOVERY.TXT - Notepad

File Edit Format View Help

```
NOT YOUR LANGUAGE? USE https://translate.google.com

What happened to your files ?
All of your files were protected by a strong encryption with AES
More information about the encryption keys using AES can be found here: http://en.wikipedia

How did this happen ?
!!! Specially for your PC was generated personal AES KEY, both public and private.
!!! ALL YOUR FILES were encrypted with the public key, which has been transferred to your c
!!! Decrypting of your files is only possible with the help of the private key and decrypt

What do I do ?
So, there are two ways you can choose: wait for a miracle and get your price doubled, or st
If You have really valuable data, you better not waste your time, because there is no other

For more specific instructions, please visit your personal home page, there are a few diffe
1. http://gwe32fdr74bhfsyujb34gfszfv.zatcurr.com/37852119D6D3885
2. http://tes543berda73i48fsdfsd.keratadze.at/37852119D6D3885
3. http://tt54rfdjhb34rfbnknaerg.milerteddy.com/37852119D6D3885
If for some reasons the addresses are not available, follow these steps:
1. Download and install tor-browser: http://www.torproject.org/projects/torbrowser.html.en
2. After a successful installation, run the browser
3. Type in the address bar: xlowfznrg4wf7dli.onion/37852119D6D3885
4. Follow the instructions on the site.
```

Ln 1, Col 1 | 100% | Windows (CRLF) | UTF-8