ADVANCED MALWARE ANALYSIS OF A ZEUS MALWARE

By

1.1 INTRODUCTION

Malware, also called malicious software is any software whose primary intention is to infiltrate and compromise the computer system without the knowledge of the system's owner (Alsmadi et al., 2021). It intentionally and maliciously executes malicious payloads to the targeted system including smartphones, computers and computer networks. Though there are different types and forms of malware including Trojan, Virus, Worm, Rootkit and ransomware with different attack vectors, they can be destructive and primarily aimed at causing harm to the targeted system or network (Talukder et al., 2020).

1.2 DOCUMENTATION OF ANALYSIS PROCESS

The documentation of the analysis process is divided into sections including:

1. **File Transfer and Identification:** In this section, the file is downloaded from the GitHub repository, called Zeus.rar and downloaded into a virtual environment, Oracle Virtual Machine. Furthermore, the file will be decompressed to find out the malicious executables in the file. In this case, the executables are Zsb.exe in the builder directory of the output folder and Zsbcs.exe in the section directory of the output folder and extract them out into a new folder, for analysis.
2. **Static Analysis:** After the malware has been identified, Static analysis techniques were applied to the malicious folder using Virus Total to confirm that the files are malicious and identify the type of malware it is, the file hashing, file size, timestamps and file packer. Both executables are UPX packed according to Virus Total and therefore it need to be unpacked to get important details.
3. **Executable Unpacked:** The executable was unpacked using the upx.exe and command prompt. after unpacking files were analysed on PEStudio to provide information about their structure, functionality and behavioural tendencies based on the static properties.
4. **Dynamic Analysis:** in this section, the malicious executable is run on the virtual machine to observe its behaviour on the system in real-time. Dynamic analysis tools including Process Monitor, RegShot and Wireshark are used to capture its runtime activities such as file creation and file modification. Registry modifications and network communications.

1.3 MALWARE IDENTIFICATION OF THE PACKED UPX FILES

These file identifications were conducted in the Oracle virtual machine according to the analysis made on the RAR file, called Zeus.rar in the repository, upon decompressing the file, there were two executable files found, the zsbcs.exe and the zsb.exe. PE executable and malicious file (.exe). Further analysis was made using both static analysis and dynamic analysis using tools such as Virus Total, PEStudio and IDA Pro (Wang et al., 2023.

he files, Zeus.rar contains two executables called zsbc.exe and zsb.exe when these executables were analysed in Virus Total, a threat intelligence database, the details of the two executables were found.

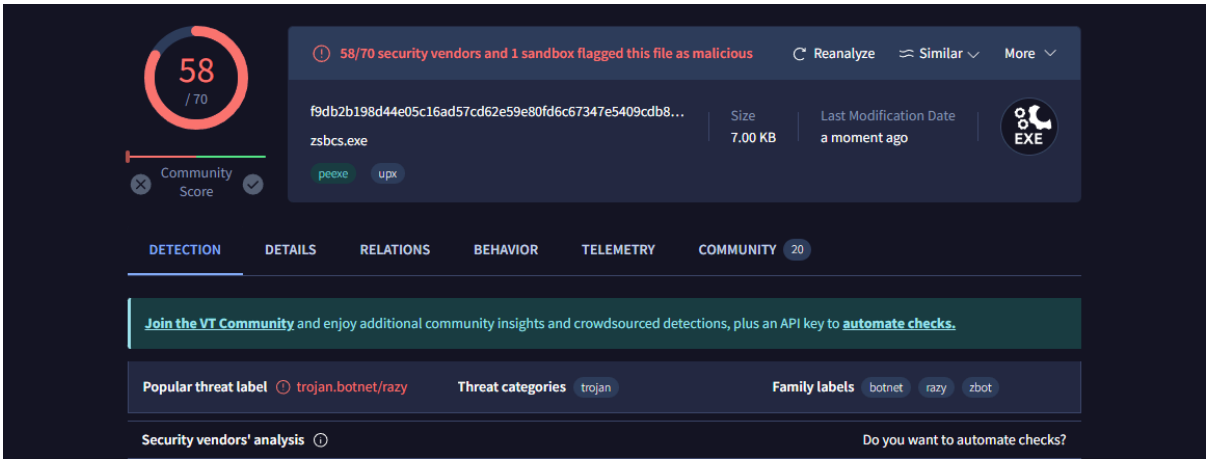


Figure 1: **Identification Of Zsbcs.exe File**

As shown in Figure 1, the reputation of the file was assessed and the virus total revealed that 58 out of 70 entities identified it as a malicious file showing the identification of zsbcs.exe file. The file was categorised as a trojan, associated with the botnet family and includes variants like Razy and Zbot. Upon checking the details section of the virus total,

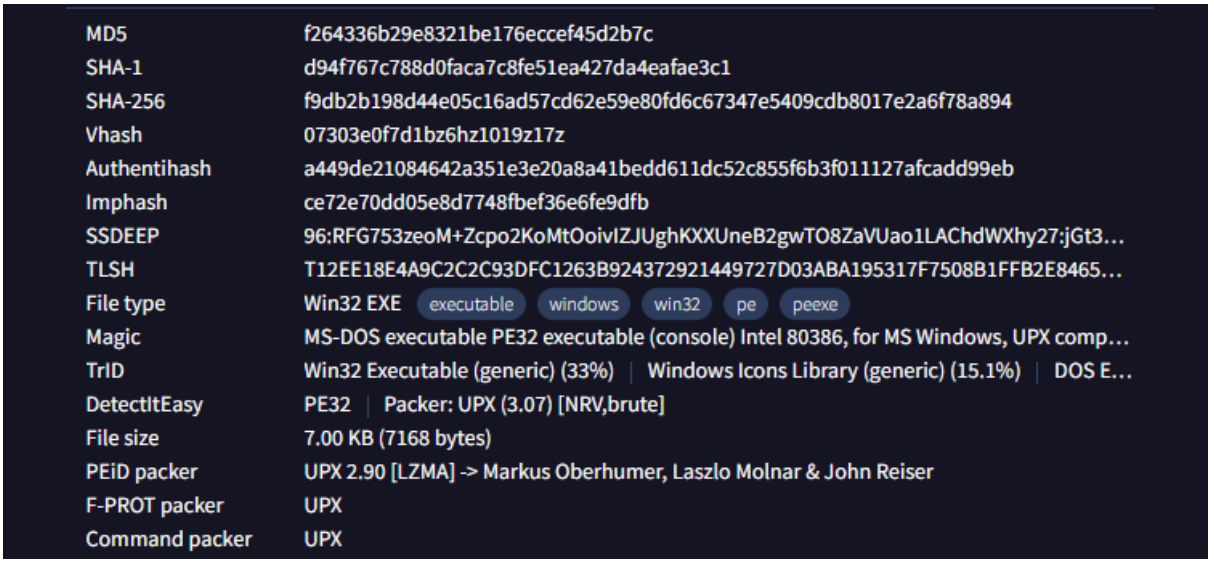


Figure 2: Details of zsbcs.exe packed file

As shown in Figure 2, the hash values including MD5, SHA-1 and SHA-256 indicate the unique identifiers of the file, signifying the digital fingerprints for identification and verification of files without checking the contents (Hoffman, 2010). The file type is a Win32 EXE, indicating that it is a portable executable file that is compatible with the 32-bit Windows operating system having a file size of 7168 bytes. Also, the file is packed with the UPX packer making it difficult to analyse due to the high compression and obfuscation capabilities, evading detection of malicious codes (Muralidharan et al., 2022).

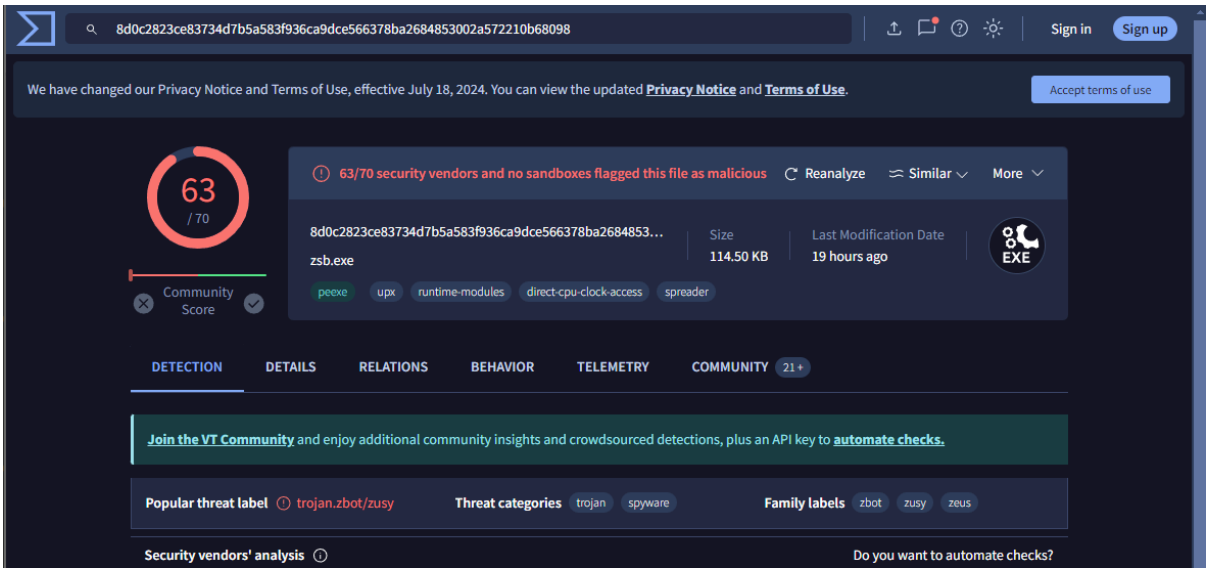


Figure 3: **Identification of packed zsb.exe**

As shown in Figure 3, the reputation of the file was assessed, and the virus total showed that 63 out of 70 entities identified it as a malicious file. the zsb.exe file was created on 14th, April 2011 UTC and became popular on 3rd August 2020 UTC. It is detected as a Trojan, associated with the Zbot family. Additionally, it has the capabilities of spyware, which means it can spy on user activities and steal sensitive information such as banking and login details.

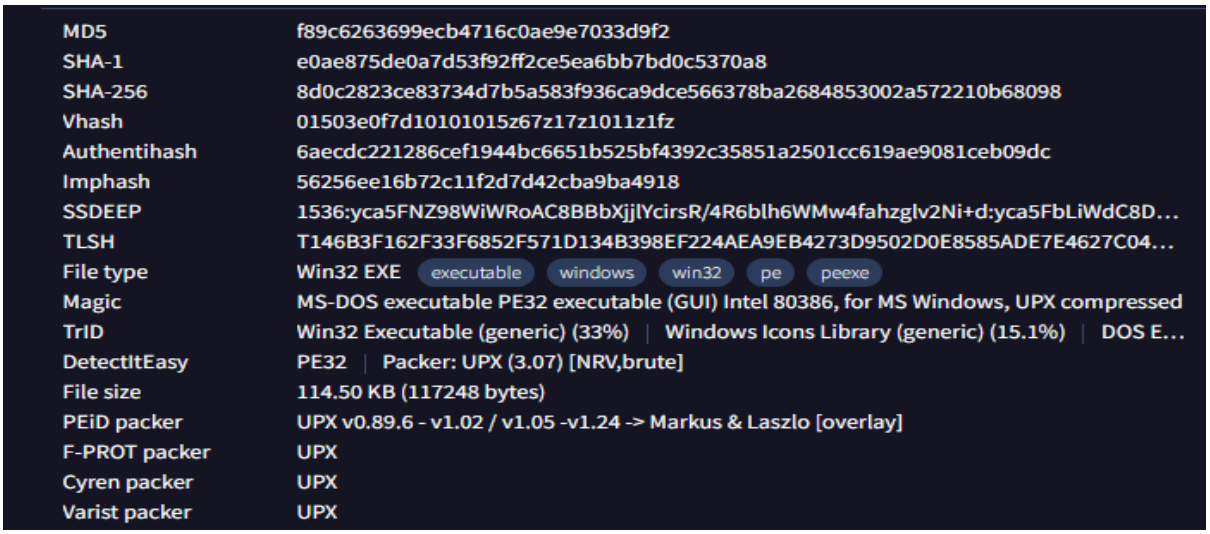


Figure 4: Details of Packed Zsb.exe file

As shown in Figure 2, the hash values including MD5, SHA-1 and SHA-256 indicate the unique identifiers of the fil, signifying the digital fingerprints for identification and verification of files without checking the contents (Hoffman, 2010). The file type is a Win32 EXE, indicating that it is a portable executable file that is compatible with the 32-bit Windows operating system having a file size of 117248 bytes. Also, the file is packed with the UPX packer making it difficult to analyse due to the high compression and obfuscation capabilities, evading detection of malicious codes (Muralidharan et al., 2022).

1.4 THE UNPACKED UPX FILE

The Zsbcs_unpacked.exe file

The file was unpacked by downloading the upx executable and running the unpacking command on the command prompt, thereby unpacking the file (Kamble, 2022). Upon unpacking the file, it was rerun on virus total, with the following detection

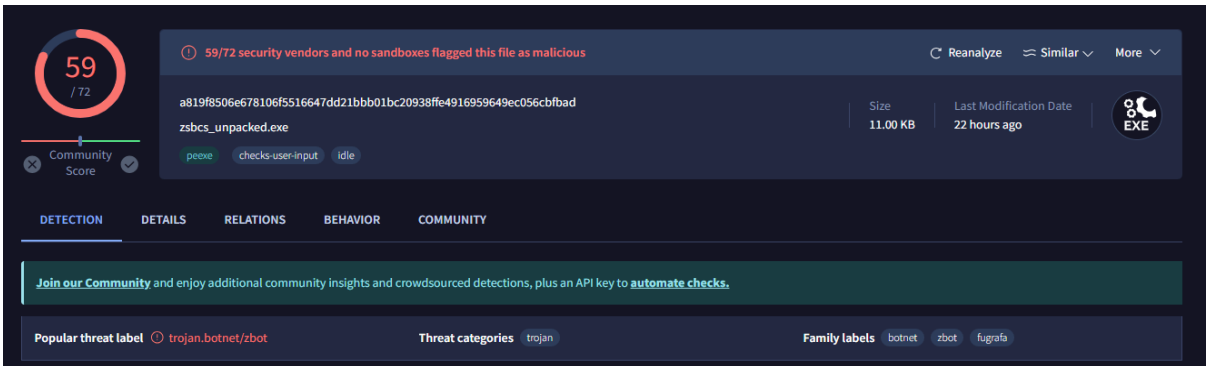


Figure 5: The Zsbcs_unpacked.exe file

As seen in Figure 5, unpacking the file gave us the original executable of the file in i9ts uncompressed state. of the file, the zsbcs_unpacked.exe unpacked file is a Trojan malware, associated with the botnet family, created on the 14th of April, 2011.

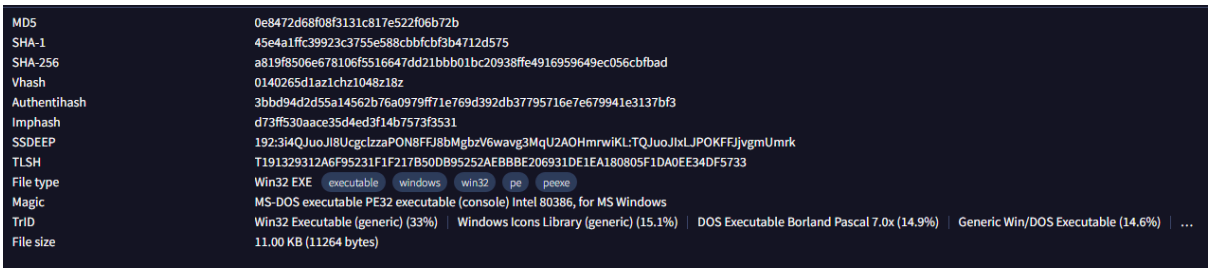


Figure 6: Details of zsbcs.exe file on Virus Total

Figure 6 shows the details of the unpacked file where some of them are different from the packed file including the file size and hash codes. The file size is 11264 bytes which is more than the packed file, a portable executable compatible with the 32-bit Windows operating system.

The Zsb_unpacked.exe file

The file was unpacked by downloading the upx executable and running the unpacking command on the command prompt, thereby unpacking the file. Upon unpacking the file, it was rerun on virus total, with the following detection;

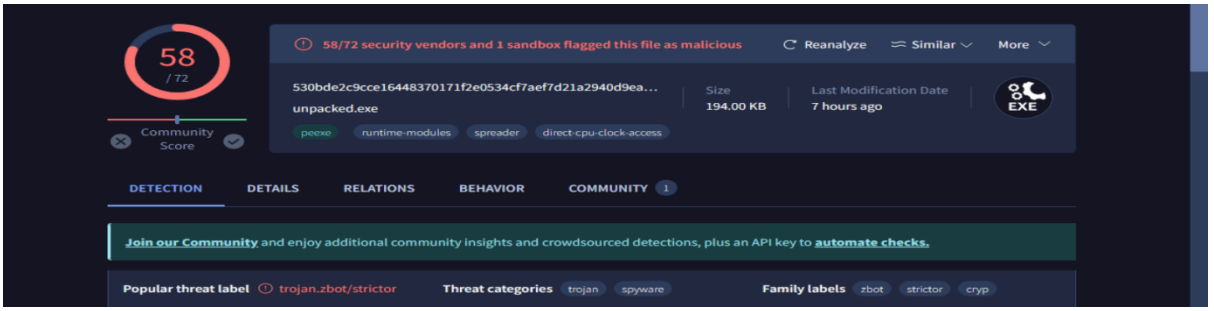


Figure 7: The zsb_unpacked.exe file identified on virus total

As seen in Figure 5, unpacking the file gave us the original executable of the file in its uncompressed state. of the file, the zsb_unpacked.exe unpacked file is a trojan malware, associated with the Zeus family and capabilities of scriptor and cryp, created on the 14th of April, 2011.

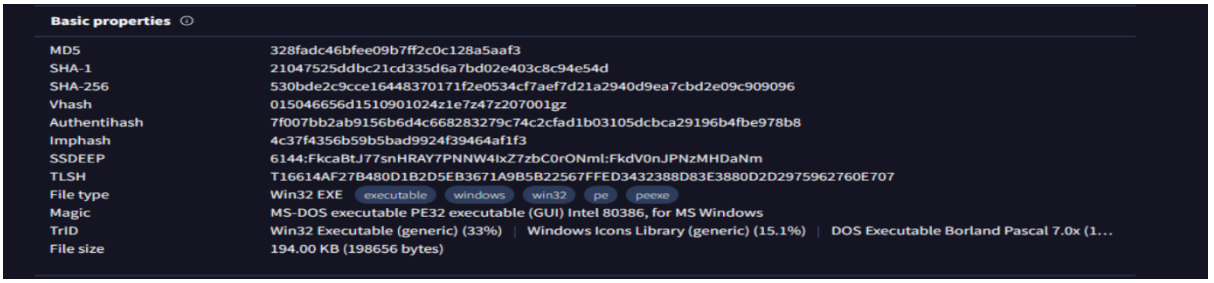


Figure 8: The zsb_unpacked.exe file details identified on the virus total

1.5 BEHAVIOURAL OVERVIEW

Introduction to Trojans:

Trojan was named after the Greek Myth of the Trojan Horse. They are malware that disguise themselves as legitimate software and trick users into executing them, it is one of the most destructive malware in the cyber world that is uneasily to detect. Trojans, unlike other malware like virus and worms rely on social engineering techniques to infiltrate systems and carry out their malicious operations. Trojans mostly leverage the Online Social network users and lure them into visiting malicious websites that are disguised as legitimate and harmful software and websites such as Adobe FLASH Players and Social Networks., once users click on it, it is activated, and it starts his malicious operation (Faghani and Nughen, 2017).

Types and Variants of Trojans:

1. **Backdoor**: A backdoor is a type of malware that grants unauthorised access to a computer system or network, specifically, to create a secret entrance into the infected system, thereby enabling remote access, hence remote control of the system (Li et al., 2022)
2. **Banking Trojans**: these are variants of trojans that are targeted towards financial institutions, they aim to steal banking credentials including login details, sensitive financial information and card details. Examples of banking trojans include Zeus, Emotet and TrickBot (Grammatikakis et al., 2021).
3. **Hardware Trojans**: these are malicious modifications to a hardware component during manufacturing to control, disable, monitor and modify its logic (Gubbi et al., 2023)
4. **Spyware Trojans:** this is a variant of trojan that is installed on the user's computer to monitor the activities of the user stealthy without the user's knowledge, they are mostly used for information gathering about a person including screenshots, web browsing activities, keystrokes. Examples of spyware include Adware and Keylogger (Stafford and Urbaczewski, 2004)
5. **Remote Administration Trojans (RATs):** are the types of trojans that provide control of the compromised computer from a remote location. Once on the compromised system, they perform a range of activities such as interacting with the victim's system in real-time and monitoring and manipulating the system (Kondalwar and Shele, 2014).

Introduction to Trojan.Zbot/Zusy

Zbot, also called Zeus is a sophisticated malware associated with the Trojan. It is one of the first banking Trojan families that existed since 2007. It used two configuration files, config.txt, containing basic configuration files and webinject.txt, containing basic data for web injection, it receives the configuration information by downloading an encrypted version of the configuration file from a command and control server (Nelson and Noteboom, 2023). Zeus became popular in 2009 when it targeted the United States Department of Transportation. Its source codes were disclosed in mid-2011, producing more sophisticated variants such as Citadel and GameoverZeus.

Zeus primarily targets the Microsoft Windows Operating System and its primary goals are to steal financial information by employing keystroke logging and web injection to capture sensitive information it builds botnets by adding infected machines to a botnet, allowing for remote control (Grammatikakis et al., 2021).

Behavioural Overview of the Zbot botnet

This section presents how Zeus malware operates and acts when it infects a system. Zeus infects systems through malicious attachments, phishing emails and exploit kits. The behaviour of Zeus includes injecting malicious data into the victim's website using the **webinject**, which lures the victim into interacting with malicious webpages instead of legitimate websites and browsers. This allows the malware to extract sensitive information about the victim including credentials and cookies. The

Webinject.txt contains instructions that need to be injected into the victim's website (Nelson and Noteboom, 2023).

1.6 STATIC ANALYSIS OF ZSB.EXE

Imports:

The import table gives an overview of the lists of Dynamic Link Libraries (DLL) contained in the two executables

LIBRARIES	SUMMARY	RISKS
KERNEL32.dll	the kernel32.dll is located in both executables and provides core functions and services that perform process management, memory management and dynamic linking.	while the functions are used by legitimate programmers and for legitimate purposes, Zeus is capable of manipulating the memory, processes, and files and evading detection.
COMCTL32.dll	COMCTL32 manages common controls for Windows graphic user interface applications	The malware, if executed, can manipulate the GUI elements which can lead to arbitrary code execution
SHLWAPI.dll	Shlwapi is in both executables and provides ways to manipulate text data in a way that provides accurate and efficient string processing in Windows application	malicious code can manipulate the strings, path and URLs
USER32.dll	this library manages the user interface elements such as the windows and button	malware can manipulate the User interface elements to create and manipulate windows user inputs to deceive users or evade detection
COMDLG32.dll	provides common dialogue box functionality such as file open/ save, selecting folders	malware can provide unauthorised access to files, thereby facilitating the spread of malicious payloads through deception.
ADVAPI32.dll	Advapi provides and manages the registry, security and Windows service operation	malware could execute privilege escalation in the

		registry and authentication bypass in the security.
SHELL32.dll	Shel provides any functions related to the Windows shell operations including shortcut creation, executing shell commands and file manipulation.	Zeus botnet can manipulate files including creating, copying, deleting or moving files. Furthermore, they can execute files that launch other payloads.
ole32.dll	Ole supports Object Linking and Embedding (OLE) Technology in Windows, it also facilitates the communication between software components.	Ole is at risk of privilege escalation attacks
WS2_32.dll	it provides a function used to assign a local address and port number to a socket before it can be used for network communication	Zeus can receive commands from the CnC, exfiltrate stolen data and propagate to other systems.

Table 1: Import table for the Zsbcs_unpacked and Zsb_upnacked.exe

The libraries contained in each executable are presented in Table 1, while each of these dlls are legitimate dlls used by Windows libraries, malware can infiltrate the executables and can pose a risk to the system if not properly monitored and prevented.

On ZSBCS_Unpacked.EXE

Indicators:

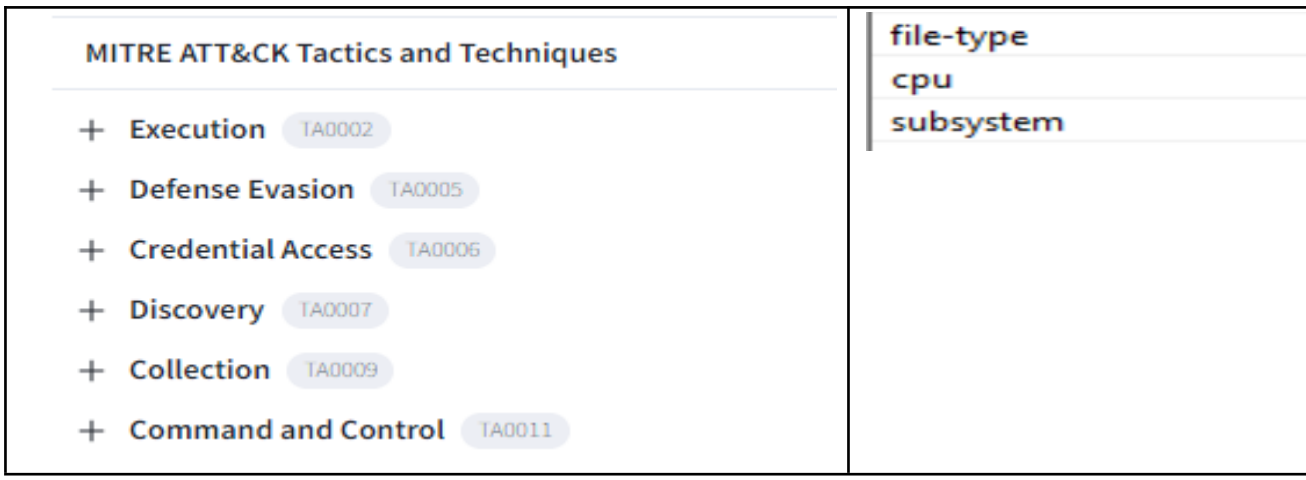


Figure 9: Zsbcs_unpacked.exe Behavioural structure on Virus total and PEStudio.

The analysis of the file on virus total and PEstudio indicated that the Zsbcs executable file is designed to interact with the Windows 32-bit system's command line interface and employs XOR encoding to obfuscate its data, making it difficult to detect. Additionally, It queries network configuration and system information settings to gather information about the system and the network communication and captures user inputs including user login details using Keystrokes from the target system. Lastly, the file is capable of using legitimate application layer protocols to communicate with the CnC servers.

Libraries:

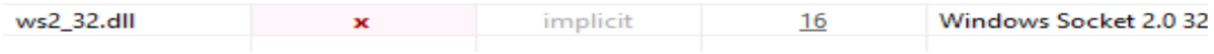


Figure 10: Zsbcs_unpacked.exe Library on PEstudio.

According to Pestudio, ws3_32.dll was identified as suspicious for establishing connections and communication between the sender malware and the target over a TCP/IP network which could suggest that the file is manipulating the network communication process of the system and trying to establish a command and control in it.

Function

Upon further observation in the function section, it shows that the bind function in the library is abused by the malware to establish communication between the Command and Control (C&C) server and the system (Geng et al., 2024). The VirtualProtect was also blacklisted which could indicate memory manipulation by modifying the memory permission or bypassing the security mechanism.

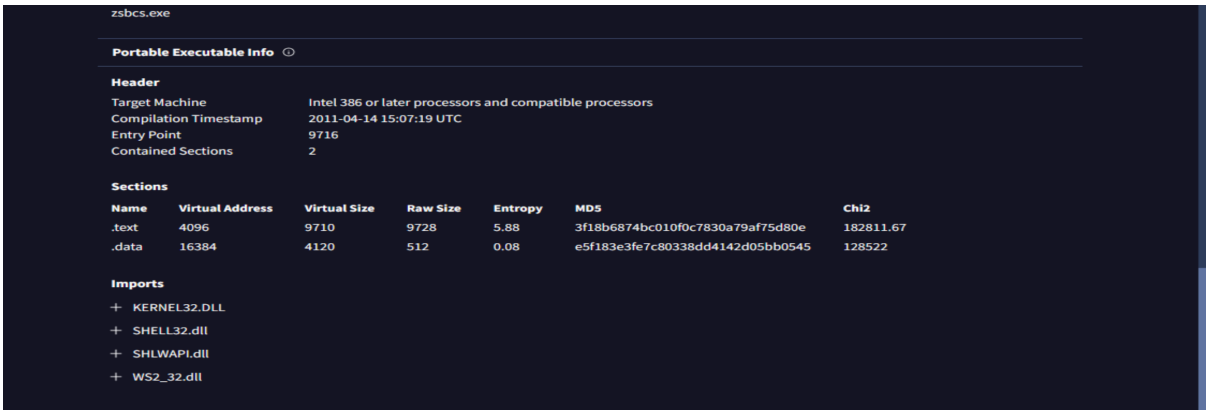
Strings:

36	section:.text	-	format-string	-	-	ZeuS Backconnect Server %u.%u.%u.%u.	
43	section:.text	-	format-string	-	-	Usage: %s <command> -<switch 1> -<switch N>	
28	section:.text	-	format-string	-	-	ERROR: Unknown command "%s".	
25	section:.text	-	format-string	-	-	ERROR: Syntax error "%s".	
27	section:.text	-	format-string	-	-	ERROR: Unknown switch "%s".	
52	section:.text	-	format-string	-	-	Accepted new conection from bot (BotID: %s	IP: %s).
89	section:.text	-	format-string	-	-	Accepted new conection from client (IP: %s)	but bot not connected! Disconnecting client!
52	section:.text	-	format-string	-	-	Accepted new conection from client (IP: %s	ID: %u).
73	section:.text	-	format-string	-	-	Waiting for incoming connections (port of bot: %u	port of client: %u)...

Figure 11: Zsbcs_unpacked.exe in pestudio.

Figure 11 suggests that the malware functions as a back-connect server for the Zeus botnet to remotely communicate and control compromised systems, called bots (Sood et al., 2013). While the %u could be the version of the server. Using a format, the user can interact effectively with the system, with output messages based on the type of connection formed.

Section

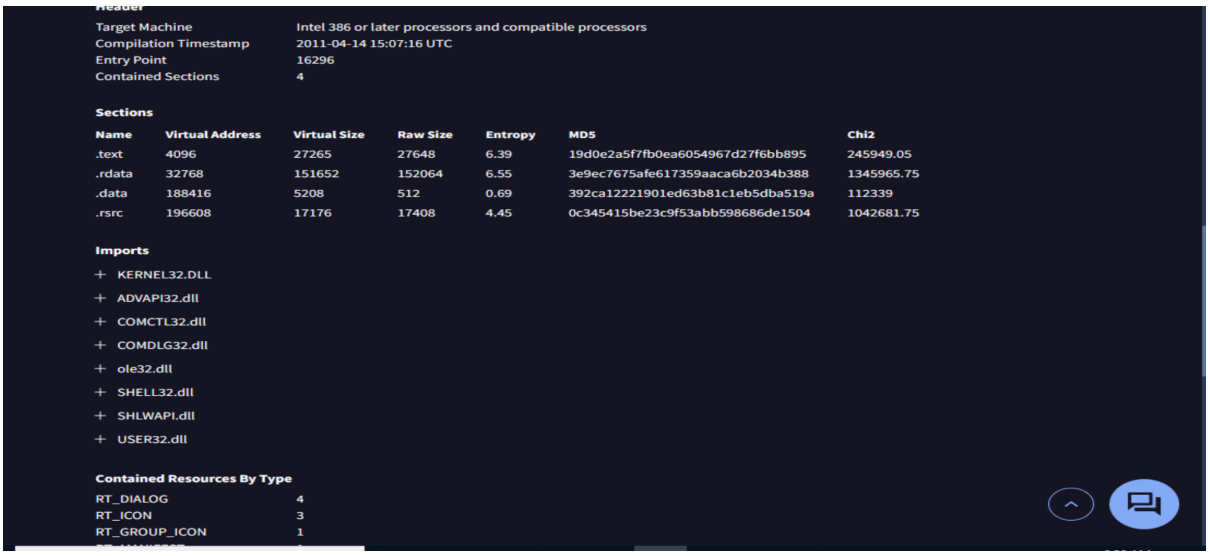


The Section illustrates two segments of sections; **".text"; ** contains the instructions/code that are obtained from the Central Processing Unit**, and ".data"** contains readable data that are used by the malware during the runtime, it includes strings, configurations and encrypted data (Poudyal et al., 2019).

Static Analysis for zsb_unpacked.exe

Static analysis is the analysis of malware without executing the malware. From the analysis provided

Sections:



The Section illustrates two segments of sections; **".text"; ** contains the instructions/code that are obtained from the Central Processing Unit**, and ".data"** contains readable data that are used by the malware during the runtime, it includes strings, configurations and encrypted data (Poudyal et al., 2019).

Indicators

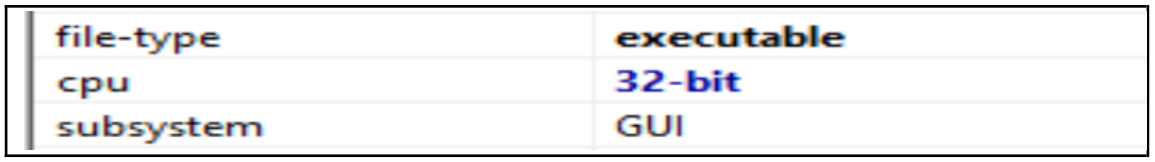


Figure 12: MITRE ATT&CK Techniques from Virus Total and Indicators on PeStudio

The zsb_unpacked file interacts with the Windows Graphical User Interface and it is compatible with the 32-bit CPU architecture that ensures a smooth run on the processor.

Strings

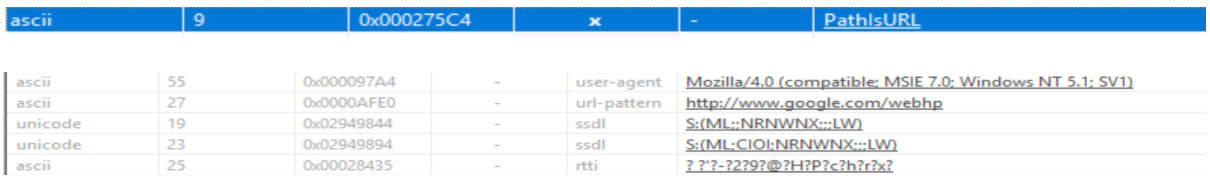


Figure 13: URL String Analysis on PEStudio

Figure 13 presents the string analysis on PEstudio which suggests that the file has injected a malware code to a function that distinguishes between a web URL and a

Local file, this means that the file can interact with online resources. Additionally, the user-agent being Mozilla/4.0 which is used for web browsers suggests that the malware could masquerade as a legitimate web browser when it is communicating over a network and the URL pattern could indicate that the URL is communicating with the URL for command and control or to download payloads.

1.7 DYNAMIC ANALYSIS OF ZSBCS.EXE

Dynamic Analysis is the observation of the behaviour of the executable during execution to understand the behaviour, function, interaction and extent of impact on the system. Process Monitoring and command prompt are tools used for the dynamic analysis of this executable (Alazab et al., 2020).

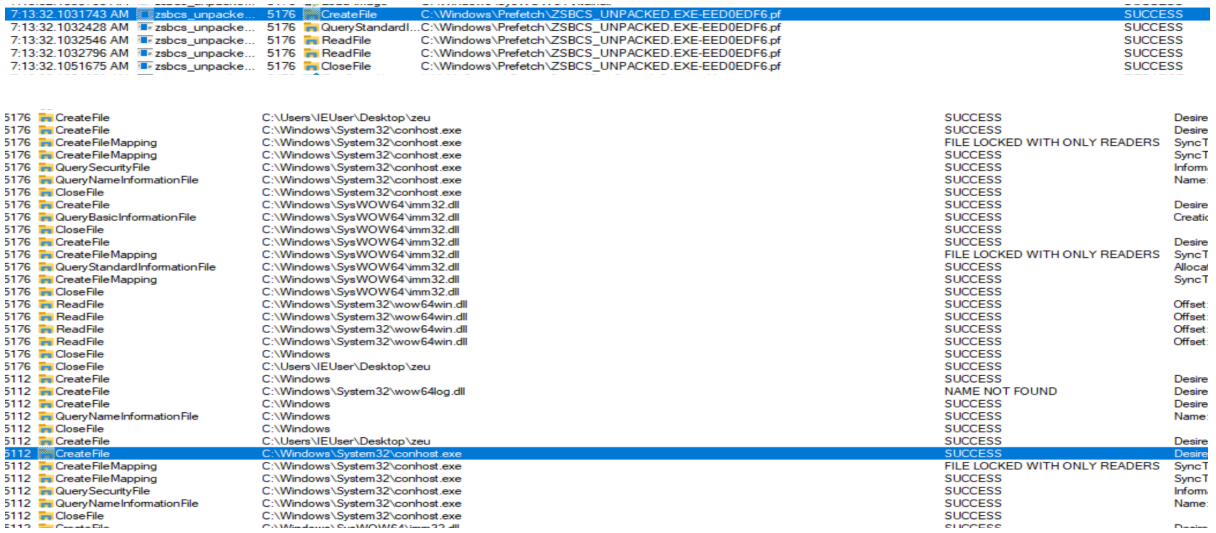


Figure 14: zsbcs_unpacked.exe analysis in process monitor

This report gave a rundown of the zcbcs after it was run. When the malware was executed, it prompted the command prompt console, copied itself hid to another file in the prefetch directory and disguised itself as a prefetch file, this to evade detection and integrate itself with other legitimate prefetch files (Heriyanto, 2014). The prefetch files are a part of the operating system used to store information about frequently run programs on the system. After creating the file in the prefetch directory, the Zeus botnet was successfully able to request standard information about the file, which could include the file size, timestamps and attributes. Additionally, the malware was observed to manipulate the imm32.dll and conhost.exe by compromising the legitimate file including manipulating the memory and gathering information about their file path, file attributes, name, access control settings and permission.

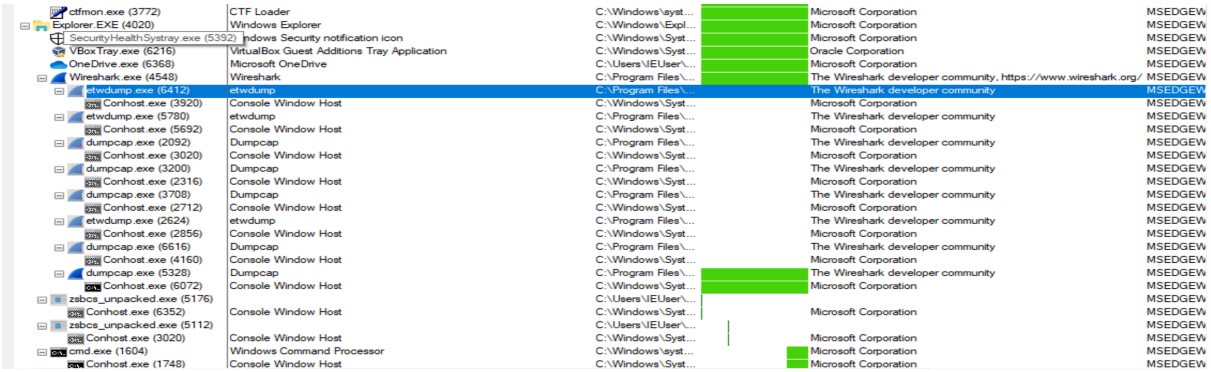


Figure 15: Zcsbc.exe across the network using Process Monitor

The multiple copies of the conhost.exe in the Wireshark sub-process indicated that the malware attempted to spread itself to other systems across the network

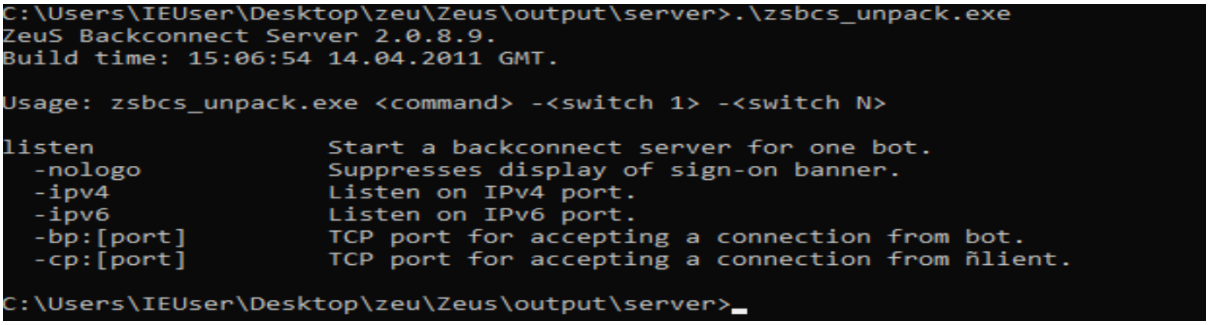


Figure 16: Zcbcs.exe command prompt

Figure 16 shows the command prompt of the zsbcs_unpacked file ran on command prompt, it shows

Dynamic Analysis of Zsb.exe File

Dynamic analysis is the observation of the behaviour of the analysis of the zsb.exe file in runtime.

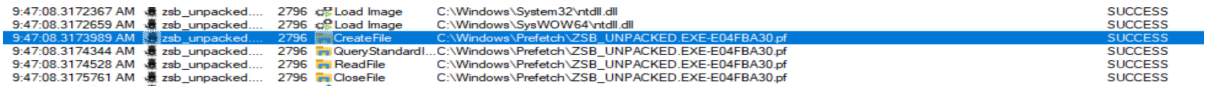


Figure 17: Process Monitoring of Zsb_unpacked.exe

The file created itself in a pf file in the prefetch folder as other legitimate pf files to evade detection initiating persistence (Heriyanto, 2014).

Figure 18: RegSetValue for Zsb_unpacked.exe

In addition to that, the Zeus malware modified the registry value in the user settings of the local machine to escalate privileges and for persistence for the containing binary data queried a bunch of values and keys in the registry in the local machine which suggests that the malware was establishing itself for persistence.

```
.idata:00408000 ; BOOL __stdcall OpenProcessToken(HANDLE ProcessHandle, DWORD DesiredAccess, PHANDLE TokenHandle)
.idata:00408000                 extrn OpenProcessToken:dword
.idata:00408000                                         ; CODE XREF: sub_402F3B+10↑p
.idata:00408000                                         ; DATA XREF: sub_402F3B+10↑r ...
.idata:00408004 ; BOOL __stdcall CryptGetHashParam(HCRYPTHASH hHash, DWORD dwParam, BYTE *pbData, DWORD *pdwDataLen,
.idata:00408004                 extrn CryptGetHashParam:dword
.idata:00408004                                         ; CODE XREF: sub_40279B+65↑p
.idata:00408004                                         ; DATA XREF: sub_40279B+65↑r
.idata:00408008 ; BOOL __stdcall CryptReleaseContext(HCRYPTPROV hProv, DWORD dwFlags)
.idata:00408008                 extrn CryptReleaseContext:dword
.idata:00408008                                         ; CODE XREF: sub_40279B+86↑p
.idata:00408008                                         ; DATA XREF: sub_40279B+86↑r
.idata:0040800C ; BOOL __stdcall CryptCreateHash(HCRYPTPROV hProv, ALG_ID Algid, HCRYPTKEY hKey, DWORD dwFlags, HCRYP
.idata:0040800C                 extrn CryptCreateHash:dword
.idata:0040800C                                         ; CODE XREF: sub_40279B+31↑p
.idata:0040800C                                         ; DATA XREF: sub_40279B+31↑r
.idata:00408010 ; BOOL __stdcall CryptDestroyHash(HCRYPTHASH hHash)
.idata:00408010                 extrn CryptDestroyHash:dword
.idata:00408010                                         ; CODE XREF: sub_40279B+7C↑p
.idata:00408010                                         ; DATA XREF: sub_40279B+7C↑r
.idata:00408014 ; BOOL __stdcall CryptHashData(HCRYPTHASH hHash, const BYTE *pbData, DWORD dwDataLen, DWORD dwFlags)
.idata:00408014                 extrn CryptHashData:dword
.idata:00408014                                         ; CODE XREF: sub_40279B+4D↑p
.idata:00408014                                         ; DATA XREF: sub_40279B+4D↑r
.idata:00408018 ; DWORD __stdcall GetLengthSid(PSID pSid)
.idata:00408018                 extrn GetLengthSid:dword
.idata:00408018                                         ; CODE XREF: sub_404084+9B↑p
.idata:00408018                                         ; DATA XREF: sub_404084+9B↑r
.idata:0040801C ; BOOL __stdcall CryptAcquireContextW(HCRYPTPROV *phProv, LPCWSTR szContainer, LPCWSTR szProvider, DW
.idata:0040801C                 extrn CryptAcquireContextW:dword
.idata:0040801C                                         ; CODE XREF: sub_40279B+19↑p
.idata:0040801C                                         ; DATA XREF: sub_40279B+19↑r
.idata:00408020 ; BOOL __stdcall GetTokenInformation(HANDLE TokenHandle, TOKEN_INFORMATION_CLASS TokenInformationClas
```

Figure 19 ZSB_UNPACKED crypt values in Ida pro.

Figure 19 shows a list of functions that was extracted from the zsb_unpacked file which indicates cryptographic operations, encryption and system manipulation. The operations are used to escalate privileges in the system to access the security system and infiltrate the authentication process.

1.8 INFECTION TECHNIQUES DISCUSSION

Figure 19: Suspicious Http file found in zsb.exe file on Wireshark.

Figure 19 shows one of the suspicious HTTP files obtained from the analysis of Wireshark. While the HTTP file itself looked legitimate as it showed it is from virus total, according to the virus total, the host, URL and the destination IP address were analysed as malicious. The primary goal of Zeus malware is to steal sensitive information such as user credentials and web cookies and send it to the command and control server. It used encryption techniques to communicate with the command and control server. Zeus utilizes a range of tactics to infect targeted systems and networks including phishing emails, Drive-by Downloads and social malware when the user visits the website. According to the file analysed, there was a set range of attack vectors used in infecting the system

The infection technique of Zeus malware begins when the user unintentionally clicks on a compromised website/email, and downloads injected links or infected files. The Zeus malware gains access into the system after being executed and gains entrance into the user's system. It installs itself into the system and duplicates itself into specific paths of the systems including the registry and file system, modifies system settings and establishes a connection with its C&C server. It utilizes encryption techniques to obfuscate itself and communicate with the C&C server (Riccardi et al., 2011).

1.9 PROPAGATION TECHNIQUES

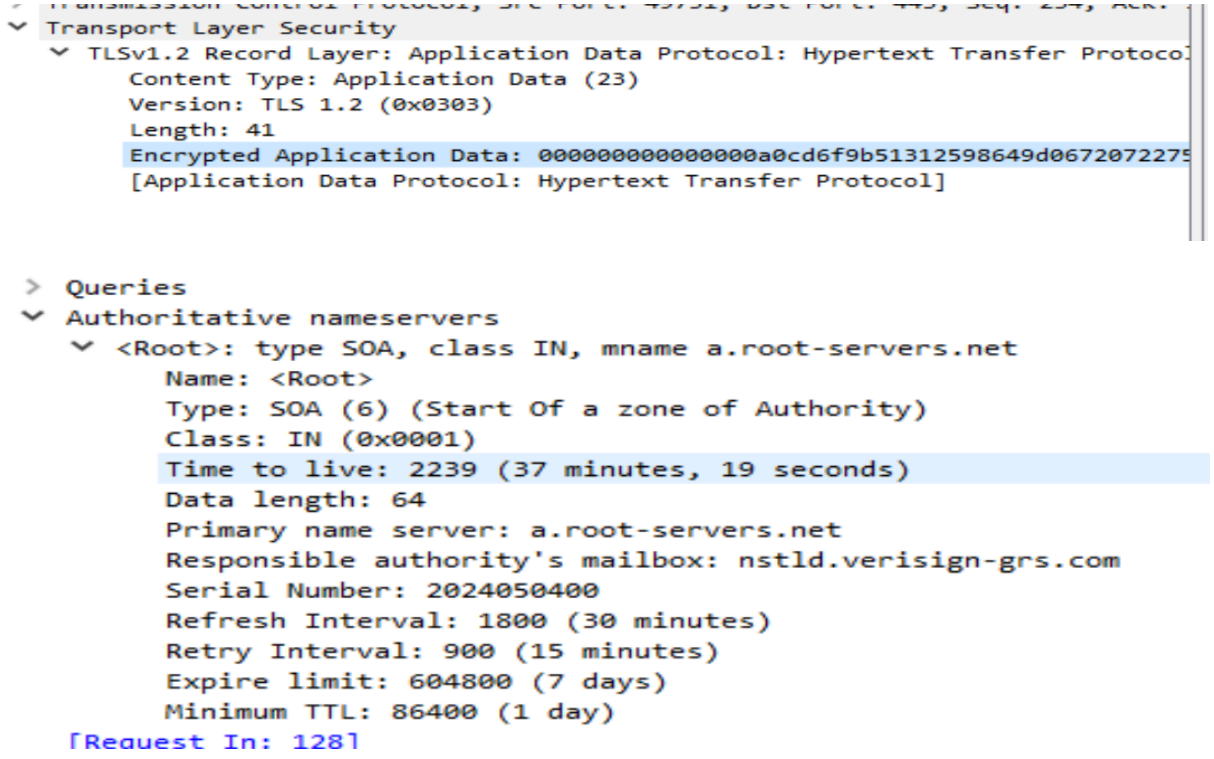


Figure 20: Encrypted Data for zsb.exe on Wireshark

Once the affected systems are formed and established connection between the system and the C&C server, the Zeus malware communicates with the C&C in an encrypted format by sending basic bot information and reports from the infected system through

HTTP POST and GET requests, this information would include stolen credentials, configuration files and web cookies. The Zeus botnet is widely used to carry out malicious activities including Distributed Denial of Service (DDoS) attacks (Barse and Tidke, 2020).

In the two executable files, it propagated the system through the network communications between the system and the virtual machine over HTTP GET and POST protocol communication and there were different url seen in the Wireshark analysis, which suggests the extent of expansion that the file went. In addition, Zeus modifying the configuration of the local machine registry of the system indicated that it facilitated propagation by running automatically at the start of the system and spreading over the system and users within the system or connected to the same network (Riccardi et al., 2011).

1.10 Covert Techniques Discussion

Zeus employs obfuscation and XOR encryption to evade detection and its communication. And ensures that the command and control are not detected by network monitoring (Barse and Tidke, 2020). In the two files analysed, the file employed the encryption key to communicate with the command and control and hide its URL under legitimate files with injected web codes. Additionally, the file attaches itself to legitimate files like “svchost” to evade detection while performing its malicious act.

1.11 DAMAGE ASSESSMENT

The extent of the damage that Zeus can cause is massive. In its first detection, there was an estimated loss of 100 million US dollars. Zeus malware causes damages that stem from the stealing of data such as login credentials and credit card details (Kok and Kurz, 2011)

During the analysis of the files executed, some things were observed such as the memory of the system became full at a point in the analysis, causing the programs in the system to fluctuate, Additionally, the changes in the screen of the system were also observed during the analysis of the files.

Lastly, the malware created lots of bots by spoofing legitimate but vulnerable HTTP and injecting them with malicious code to download payloads into the system.

1.12 EMERGING TECHNOLOGIES IN MALWARE DETECTION

The sophistication of the threat of Zeus and spyware has brought about the study of emerging technology to detect and strengthen the organisation’s cyberspace. Emerging technologies like Machine learning and deep learning are used to combat these attacks and fortify the organization from war. Aljabri et al. (2021) built a model that could generate and classify malware signatures automatically using a Deep Belief

Network (DBN), giving an accuracy of 98.6% and the effectiveness of deep learning in malware analysis.

Mohaisen and Alrawi (2013) developed Auto-mal, an automated malware analysis system using different machine learnings that runs malware samples and captures the behavioural characteristics of each sample in real-time, it employed various tools including file system, network traffic and registry and stores it in MySQL database, it identified 95% of Zeus samples.

Presented an Intersection Feature Approach that employed feature selection and classification algorithm to enhance the classification of Android botnet based on Permission and API Call features, it achieved an accuracy of 98.6%, which showed the effectiveness of the algorithm in enhancing the challenge of botnet detection.

1.13 COUNTERMEASURES

Countermeasures are important for the safeguarding of malicious activity from individuals and organizations to avoid financial loss and data breaches. Countermeasures such as

1. **Regular Security Updates and Patch Management:** Regular security updates and patch management are important to reduce the risk of zero-day attacks or exploitation and improve the overall security posture of the system or network (Di et al., 2022).
2. **Robust Cybersecurity Solutions:** Technologies and processes like antivirus, encryption and firewalls should regularly be checked and ensured that it is in good condition to help detect and prevent these cyber threats and mitigate risks in the organization, thereby ensuring business continuity. In addition to that, there are automated robust cybersecurity defence system systems (Mohammed and Zaheer, 2023)
3. **User Awareness and Education:** user awareness and education such as awareness campaigns, phishing exercises and simulations increase the awareness of employees and stakeholders to the importance of cybersecurity best practices and promote a safe security culture (Zaki, 2024).
4. **Internet Security:** network and internet security measures including a strong firewall, secure web browser and three-factor authentication should be put in place to protect all users from data breaches identity theft privacy escalation and phishing attacks.

CONCLUSION

In conclusion, the analysis of Zeus malware depicts the importance of malware analysis and implementing a robust security measure to protect digital assets against cyber-attacks. Though there has been research on the operation and propagation of the malware and its impacts on compromised systems, together with the emerging technologies that are used and implemented to compact the malware, it is of great importance for continuous regular

monitoring of the system infrastructure to safeguard the digital ecosystem and mitigate the impact of sophisticated attacks like Zeus.

REFERENCE

1. Alazab, M., Alazab, M., Shalaginov, A., Mesleh, A. and Awajan, A., 2020. Intelligent mobile malware detection using permission requests and API calls. *Future Generation Computer Systems*, *107*, pp.509-521.
2. Aljabri, M., Aljameel, S.S., Mohammad, R.M.A., Almotiri, S.H., Mirza, S., Anis, F.M., Aboulnour, M., Alomari, D.M., Alhamed, D.H. and Altamimi, H.S., 2021. Intelligent techniques for detecting network attacks: review and research directions. *Sensors*, *21*(21), p.7070.
3. Barse, Y. and Tidke, S., 2020. A study on BOTNET attacks and detection techniques. *IOSR J Electri Electron Eng (IOSR-JEEE)*, *15*(3), pp.1-5.
4. Di Tizio, G., Armellini, M. and Massacci, F., 2022. Software updates strategies: A quantitative evaluation against advanced persistent threats. *IEEE Transactions on Software Engineering*, *49*(3), pp.1359-1373.
5. Faghani, M.R. and Nugyen, U.T., 2017. Modelling the propagation of trojan malware in online social networks. *arXiv preprint arXiv:1708.00969*.
6. Grammatikakis, K.P., Koufos, I., Kolokotronis, N., Vassilakis, C. and Shiaeles, S., 2021, July. Understanding and mitigating banking trojans: From Zeus to emotet. In *2021 IEEE International Conference on Cyber Security and Resilience (CSR)* (pp. 121-128). IEEE.
7. Geng, J., Wang, J., Fang, Z., Zhou, Y., Wu, D. and Ge, W., 2024. A survey of strategy-driven evasion methods for PE malware: Transformation, concealment, and attack. *Computers & Security*, *137*, p.103595.
8. Gubbi, K.I., Saber Latibari, B., Srikanth, A., Sheaves, T., Beheshti-Shirazi, S.A., PD, S.M., Rafatirad, S., Sasan, A., Homayoun, H. and Salehi, S., 2023. Hardware trojan detection using machine learning: A tutorial. *ACM Transactions on Embedded Computing Systems*, *22*(3), pp.1-26.
9. Hoffman, S.P., 2010. An illustration of hashing and its effect on illegal file content in the digital age. *Intellectual Property and Technology Law Journal*, *22*(2).
10. Ismail, N.S., Yusof, R., Saad, H. and Abdollah, M.F., Intersection Features for Android Botnet Classification.
11. Kamble, M.T., 2022, September. Feature Extraction and Analysis of Portable Executable Malicious File. In *2022 Second International Conference on Computer Science, Engineering and Applications (ICCSEA)* (pp. 1-6). IEEE.
12. Kok, J. and Kurz, B., 2011, May. Analysis of the botnet ecosystem. In *10th Conference of Telecommunication, Media and Internet Techno-Economics (CTTE)* (pp. 1-10). VDE.
13. Kondalwar, M.N. and Shelke, C.J., 2014. Remote administrative trojan/tool (RAT). *Int. J. Comput. Sci. Mob. Comput*, *3333*(3), pp.482-487.
14. Li, Y., Jiang, Y., Li, Z. and Xia, S.T., 2022. Backdoor learning: A survey. *IEEE Transactions on Neural Networks and Learning Systems*.
15. Mohaisen, A. and Alrawi, O., 2013, May. Unveiling zeus: automated classification of malware samples. In *Proceedings of the 22nd International Conference on World Wide Web* (pp. 829-832). Mohaisen, A. and Alrawi, O., 2013, May. Unveiling zeus:

automated classification of malware samples. In *Proceedings of the 22nd International Conference on World Wide Web* (pp. 829-832).
16. Mohammed, G.G. and Zaheer, Z., 2023. NeuroCyberGuard: Developing a Robust Cybersecurity Defense System through Deep Neural Learning-Based Mathematical Modeling. *Journal of Smart Internet of Things*, *2022*(1), pp.133-145.
17. Muralidharan, T., Cohen, A., Gerson, N. and Nissim, N., 2022. File packing from the malware perspective: techniques, analysis approaches, and directions for enhancements. *ACM Computing Surveys*, *55*(5), pp.1-45.
18. Nelson, T., Nance, C. and Noteboom, C., 2023, March. Web Injection and Banking Trojan Malware-A Systematic Literature Review. In *2023 6th International Conference on Information and Computer Technologies (ICICT)* (pp. 45-53). IEEE.
19. Poudyal, S., Gupta, K.D. and Sen, S., 2019. PEFile analysis: a static approach to ransomware analysis. *Int J Forens Comput Sci*, *1*(34-39), p.88.
20. Riccardi, M., Di Pietro, R. and Vila, J.A., 2011, November. Taming Zeus by leveraging its own crypto internals. In *2011 eCrime Researchers Summit* (pp. 1-9). IEEE.
21. Sood, A.K., Enbody, R.J. and Bansal, R., 2013. Dissecting SpyEye–Understanding the design of third generation botnets. *Computer Networks*, *57*(2), pp.436-450.
22. Stafford, T.F. and Urbaczewski, A., 2004. Spyware: The ghost in the machine. *The Communications of the Association for Information Systems*, *14*(1), p.49.
23. Zaki, H., 2024. *Mobile Malware: Patterns, Consequences, and Approaches for Prevention* (No. 12017). EasyChair.