**Reverse Engineering Malware**

**of**

**Agent Tesla Malware**

**By: Sakshat Bhattarai**

**Contents**

[**1. Summary 3**](#_Toc184739810)

[**2. Introduction 3**](#_Toc184739811)

[**3. Literature Review 5**](#_Toc184739812)

[**4. Methodology 5**](#_Toc184739813)

[**I. Static Analysis 8**](#_Toc184739814)

[**II. Dynamic Analysis 13**](#_Toc184739815)

[**5. Results and Discussion 20**](#_Toc184739816)

[**6. Conclusion 21**](#_Toc184739817)

[**7. Recommendations 22**](#_Toc184739818)

[**8. References 22**](#_Toc184739819)

# Summary

This report provided a comprehensive analysis of Agent Telsa, an enhanced RAT and malicious spyware first detected in 2014. First, discovered as key logging malware, Agent Tesla has expanded its functional capabilities and can be used for credential theft, keylogging, data stealing, and gaining remote access to the victim’s device. It is dangerous to ordinary people and business by using complex techniques to avoid detection. To persist in the system and to communicated in encrypted ways.

Static and dynamic analysis approaches in the report have reveled that Agent tesla is build of packing with Autolt, registry key manipulation and persistence, and direct memory injection for invisibility. Static tools as Virus Total, Detect it easy, PEStudio and dnSpy were critical for decoding the layout of the malware and dynamic ones as RegShot, Process Monitor and PESieve for its actual activity.

**Key findings:**

* Static and dynamic analysis revealed critical insights into Agent Tesla’s behavior
* Static tools revealed code obscurities and disguise, import of viruses, and encoded information.
* The dynamic analysis also emerged registry modification, key logging and district communication with the command-and-control servers through the different protocols including the Telegram and HTTP.
* Capturing credentials from the email clients, browsers, and other applications.
* Capturing keylogging, sensitive personal data, and screenshot capture.
* Communication with command-and-control servers through encrypted networks.
* Escaping the detection by utilizing the virtual box or sandbox detection techniques.
* The presented malware called Agent Tesla, is versatile and active in phishing campaigns, proving how dangerous it is to cybersecurity.

# Introduction

Agent Tesla is a complex RAT (Remote Access Trojan) and spyware, which appeared in May 2014 and was actively exploited in attacks targeting Windows OS. It was designed as a sort of key logger to capture keyboard inputs but has now become a full-fledged malware toolkit designed to steal various types of information from affected computers and give access to the hackers, including sign-in credentials and passwords, financial data, and even logs of online communication. Despite the simplicity with which it can be manipulated and its shared adaptability, this malware has been used in many hacking incidents like phishing campaigns, data exfiltration and target attacks.

**Key features and capabilities:**

* **Credential and Data theft:** The cyber attackers can steal personal data, passwords, account details, contact information, financial information, and browsing history using Agent Tesla on more than fifty applications including mail clients and web browsers.
* **Keystroke logging:** It captures the keystrokes allowing a hacker to get all the content that is keyed in, for instance, password and messages.
* **Screenshot capture:** The malware can seize screenshots of the computer display of the victim, thus proceeding to capture its information and browsing pattern.
* **Communication Interception:** Looking at its spying features, Agent Tesla is capable of spying on victim’s emails and chats, as well as other messages exchanged in social networks and other means of communication.
* **File upload and Downloads:** The use of files enables uploading and downloading from the devices it makes it easy for the attackers to load even unwanted additional modules or steal information.
* **Network Propagation:** It means that the malware can spread to other computers on the network through the shared folder, or through some network-related issues.

**Distribution Methods:**

Agent Tesla is downloaded mainly via phishing attacks, in which the victim becomes enticed into downloading a dangerous payload in the form of an e-mail attachment link. They fraudulently look like genuine documents including invoices, shipment details, and so on. Once activated the malware is ran which starts the infection process.

**Technical Details:**

* **Obfuscation Techniques:** Agent Tesla is heavily encoded by using code packing and data encoding techniques (for instance, Base 64 or XOR encryption). It also comprises anti-analysis functions to prevent security analysis/
* **Multi-stage execution:** Agent Tesla’s execution typically involves infection process. Once a malicious attachment is placed on peer-to-peer link, followed by scripts/exploits to launch a download/decrypt the final payload.
* **Persistence Mechanisms:** To ensure it executes on continuous basis on infected machine, it may deposit copies in the startup folder or registry run keys.

**Communication Channels:**

Using HTTP, SMTP, FTP and even a Telegram chat, Agent Tesla can communicate with its command and control (c2) servers. The protocol to be followed has to do with the setup of the attackers.

**Notable Incidents:**

This cyber-attack tool has revealed connections to many years’ worth of attacks across numerous sectors. In the period of the COVID-19 pandemic, it was typical for the campaigns that exploited the COVID-19 topics to deceive their targets. For example, in 2020, a campaign used to lure of the COVID-19 fake updates for personal protective equipment containing Agent Tesla.

# Literature Review

Agent Tesla has been a widely researched and written about Remote Access Trojan (RAT) and spyware that originated in 2014 and continues to remain active. Security analysts have established its development from a simple keyloggers malware that gathers information about the pressed keys to a complex malware framework that gathers credentials, transfers data out to the network, and controls remote systems. Being versatile and easy usable in any form, it was a preferred option of cybercriminals for phishing and targeted attacks.

Numerous cases and investigations describe how it instrumentalized global occurrences for this purpose. As observed in the COVID-19 crisis, the malicious program was disseminated thorough phishing emails representing the appearance of general sanitary recommendations or fake order for individual protective equipment (IPE). These campaigns effectively project at the individuals and organizations as they sought to maximize their fears during the crisis.

Another incident includes agent Tesla to tap into the email of the oil and gas industry, whereby fake document relating to delivery schedule were used in the phishing emails infected to corporate systems. The case described in the paper illustrates the use of industry-specific behavior in communication to access and retrieve data.

Studying also reveals that, Agent Tesla is heavily reliant on different levels of obfuscation and that it can easily bypass conventional detection mechanisms. Such campaigns concerning Microsoft office updates revealed it is a credential stealer and can extract data from the email clients, web browsers, and FTP applications.

The above cases and studies show the popularity of Agent Tesla and the major problems it creates for security. Following these recorded cases, this report endeavor to expand the current understanding of the malware and intends to elaborate on the malwares’ technological aspects, dissemination methodologies as well as potential countermeasures.

# Methodology

We have used both Static and Dynamic analysis for our analysis. The Methods used are

**Static Analysis:**

This analysis involves examination of a file without executing it to understand its functionality, behavior and its structure. The investigation on Agent Tesla malware requires the use of static analysis as the malware usually employs obfuscation and packing.

* **VirusTotal:** It is an online platform tool that scans for malware and then submits this to several antivirus engines for assessment. Submitting the samples of Agent Tesla through the analysis to VirusTotal helps understand whether it is a know variant. The results are useful to understand the reputation of the malware, related domains, and potential variants to support the first-stage identification.
* **Detect it easy:** A tool for identification of the compiler and the packer used in preparation of the malware. Agent Tesla utilizes packers to conceal its code in most of their instances. Detect it easy thereby informs the kind of packer used, the level of sophistication and assists in the unpacking of samples for analysis.
* **PEStudio:** It is tool that analyses PE files for deviations and rather risky characteristics. Analyzing the PE file of Agent tesla in PEStudio, we can find out the suspicious behaviors like API calls GeyAsyncKeyState of InternetOpenUrlA, typically utilized to key log or transfer data out of the network, strings coded or hard coded c2 server addresses, anti-analysis techniques include packed section or modified header, it’s essential to include techniques that aim to prevent reverse engineering process.
* **PESieve:** It is a program to debug code injection within a programs memory of in the executable files already modified by a particular code. Trojan was discovered to inject a variety of commands into legitimate processes, with the name of the agent being Agent Tesla. PESieve can detect these injections and dump the memory, these could be used to identify the infection points and provide analysts with the malicious payload to study further.
* **PeBear:** It is a PE file editor and viewer. To examine the structure of resources section where Agent Tesla could be hiding encoding code, we utilized PEBear to narrow down signs of potential obfuscation like oversized .text sections, or encrypted data disguised in resource areas.
* **Autolt Extractor:** It is used to disassemble scripts compiled/ The traditional mode of operation of Agent Tesla variants has been to encode the applications with Autolt scripts. It can decompile these scripts to decrypt the payload delivery/logic, unpack and or getting the settings configuration.

**Dynamic Analysis:**

Dynamic analysis in runtime entails great importance when it comes to using Agent Tesla in operation since it conducts such activities as keylogging, account stealing, and communication with the C2 server.

* **Process Monitor (ProcMon) :** ProcMon tracks Agent Tesla activity and some of the processes it can perform in real-time like modification of the registry keys for persistence. For example, adding the code to the startup keys, file system activities like creating files, modified or closed files could be related to creating some temporary files or logs for storing credentials and passwords stolen, we observed network traffic associated with communication with its command-and-control C2 server.
* **Process Explorer:** It is an application that can be considered as the real replacement for windows Task Manager including the advanced information about processes. With its help, one can detect processes created by Agent Tesla, for example infected system processes (explore.exe). That’s why it shows related modules and DLLs that the malware loads.
* **RegShot:** RegShot takes two captures of the system registry before and after executing agent tesla. It helps monitoring of new registry keys where changes made to detect registry keys added, modified, or deleted by agent tesla for persistence mechanisms, for example in HKCU\Software\Microsoft\Windows\CurrentVersion\Run, presents comprehensive registry change report featuring differences, which complements Promon’s event tracking.
* **dnSpy:** Because Agent Tesla is sometimes coded in .NET and because of dnSpy, on can easily decompile its code and see the hardcoded credentials or C2 server addresses, the motivations for its data leakage capabilities like keylogging or FTP/SMTP file transfer etc. This means that some of the commonly used anti-analysis techniques such as sandboxing or debugging.

**How these tools together Aid analysis of Agent Tesla:**

* **Initial Detection:** VirusTotal explains what the nature of Agent Tesla is observed version of the virus.
* **Unpacking and obfuscation detection:** Detect It Easy, PeBear, and Autolt Extractor assist in detecting and undoing the malware’s main steps that employ various forms of obfuscation to mask their payload; it unpacks their contents.
* **Structural and Functional insights:** PEStudio and dnSpy give more unique insights into its internal workings, most importantly, the key functions such as what they call key logging routines, C2 communication, and data exfiltration techniques.
* **Registry change Tracking:** It is here that RegShot focuses on the registry changes spelled out by Agent tesla, including the entries that enable persistence at start up or modify system settings which are so important for its counteraction.
* **Runtime Behavior Monitoring:** Promon and Process explorer reflect the activity of Agent Tesla in real-time with regard to files and registry keys form the operating system and networks through which it communicates, is essential for better comprehension of its effects.
* **Memory and injection analysis:** PESieve detects code injection attempts and modifications to memory, inactivity that is characteristics of the functioning of Agent Tesla.

## Static Analysis

First of all, while doing static analysis we downloaded the malware sample and analyzed it using Detect It Easy and check it whether the malware was packed or unpacked.

A screenshot of a computer

Description automatically generated

As per our susception it was packed with the AutoIt binary as mentioned shown in the screenshot.

A screenshot of a computer

Description automatically generated

Once we found out it was packed with an AutoIt binary our main task was to unpack it, so we converted the file into an unpacked form, we used an command upx -d DHL008976.exe (Malware Sample name) to unpack it.

A screenshot of a computer

Description automatically generated

Then we were successfully able to view unpacked version of Malware sample in Detect It Easy.

A screenshot of a computer program

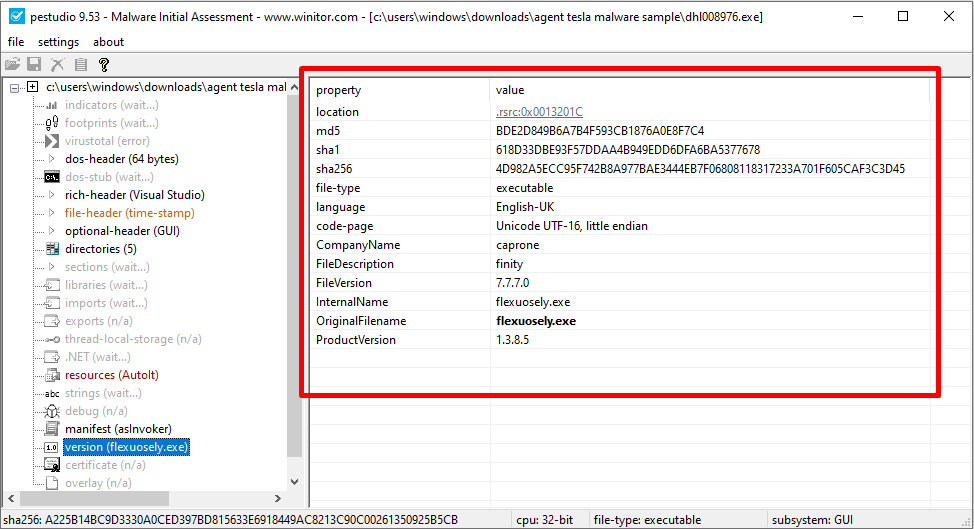
Description automatically generated

As the file was still packed with an AutoIt component, we utilized an AutoIt extractor to unpack and extract its components, enabling us to access the underlying scripts and binaries. This step was essential for conducting a deeper analysis to understand the malware's functionality and potential payload.

A screenshot of a computer

Description automatically generated

When we ran a program in pestudio we could analyzed it further as it gave us an insight about it such as a file type as a executable file, file version as 7.7.7.0, original file name as flexuosely and much more other information.



I also found out it’s AutoIt component which have a high entropy of 7.999 which gave an indication of a file being either an encoded or an encrypted file.

A screenshot of a computer

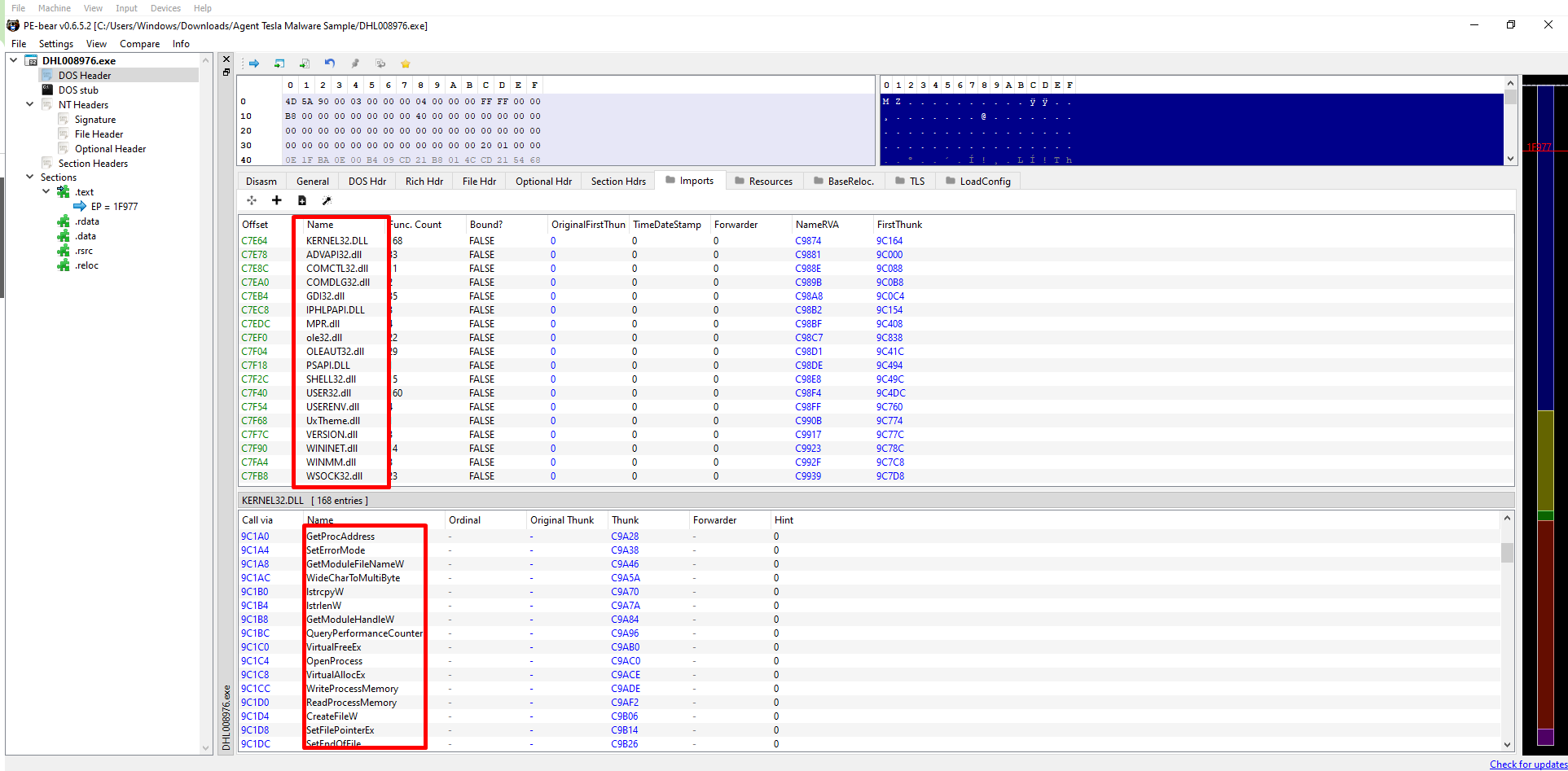
Description automatically generated

Along with it there were multiples of imports which had a process injection and process discovery capability which gave us a strong indication of the process injecting a certain import in the process.

A screenshot of a computer

Description automatically generated

When we analyzed the malware sample in the PEBear we could see a high number of the DLL as an import and there was a process such as a ReadProcessMemory, WriteProcessMemory, VirtualAllocEx, OpenProcess, etc. From the imports it had a high possibility of the malware doing an injection.



## Dynamic Analysis

Upon executing the malware sample and analyzing it using RegShot, we observed that the malware made 33 modifications to the system registry. These changes are indicative of the malware's attempt to establish persistence, alter system configurations, or prepare the environment for its malicious activities.

A screenshot of a computer

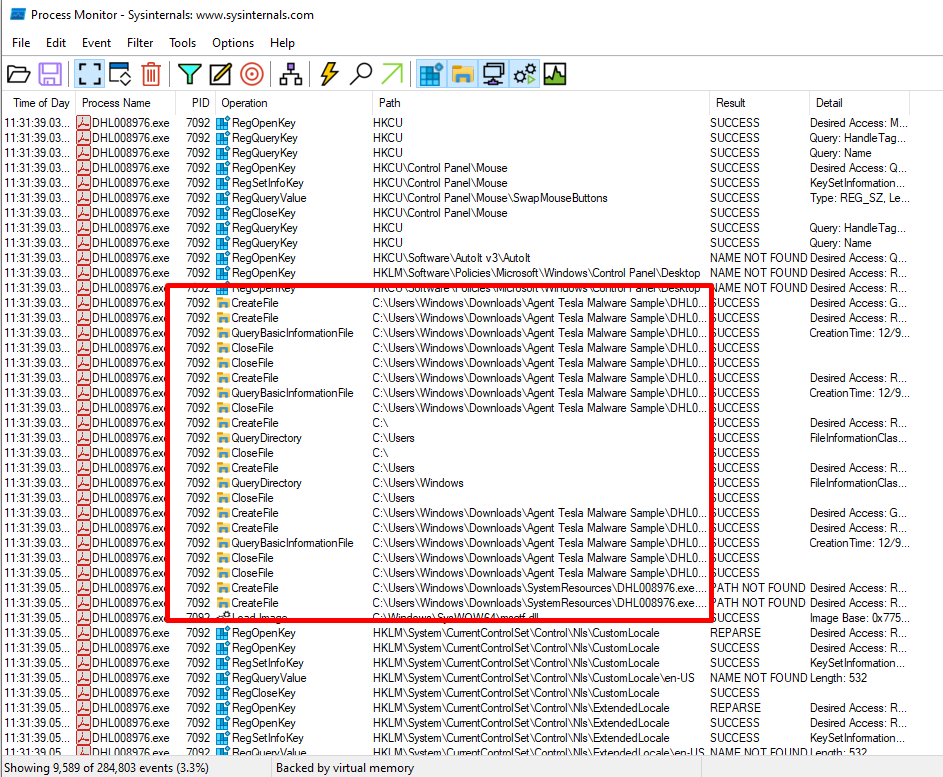
Description automatically generatedA screenshot of a computer

Description automatically generated

A screenshot of a computer program

Description automatically generated

During the process, we used a popular dynamic analysis tool called ProcMon to observe the behavior of the malware after execution. We discovered that it was creating files, closing files, and accessing the system directory, leading us to strongly suspect that the malware sample was an information stealer and credential harvester.

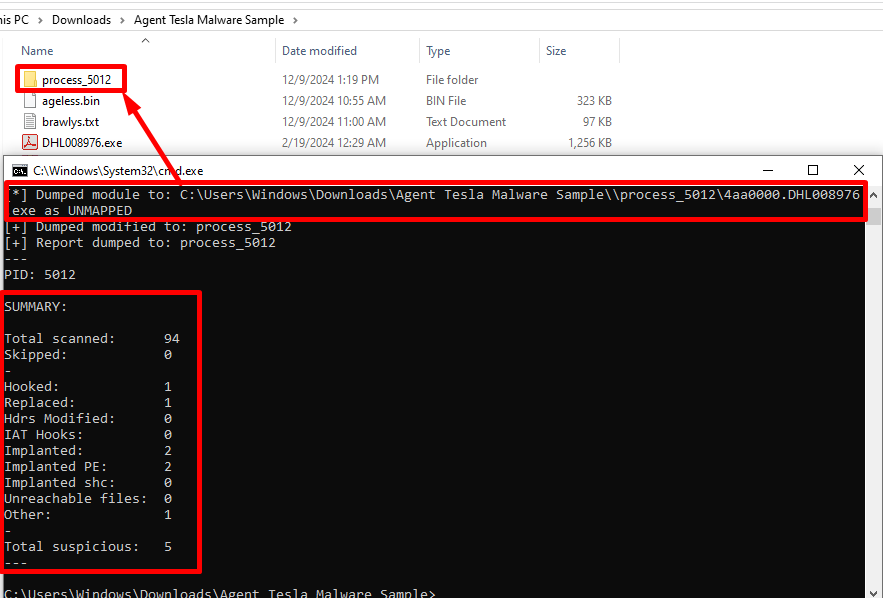


While it was still running in the memory we ran a pe-sieve and executed a process which was running a malware sample for further analysis.

A screenshot of a computer

Description automatically generated

When we ran the pe-sieve it determined 5 suspicious components of the particular binary file, along the process we dumped the particular process, from the pe-sieve we found out that it has a 2 implanted pe files and additional 2 implanted process.



During our analysis of the file in dnSpy, we discovered that it was engaging in network communication, as TLS-related information was defined within a specific function. This indicates that the malware may be using encrypted channels to securely transmit data and has a high potentiality to evade detection and protect the information which is being exfiltrated.

A screenshot of a computer program

Description automatically generated

We also discovered the presence of several keyboard hooks, with their definitions indicating keylogging behavior, such as detecting keystrokes and capturing input from the keyboard. This analysis strongly suggests that the malware possesses keylogging capabilities, enabling it to record sensitive user inputs, and further confirms its functionality as an information-stealing threat.

A screenshot of a computer program

Description automatically generated

A screenshot of a computer program

Description automatically generated

A screenshot of a computer program

Description automatically generated

Along the process of performing an analysis on of a particular class on a dnSpy we found there was a mention of a chat id along and there is also a telegram api present, looks like it using it to send all the captured information to the particular telegaram channel.

A screenshot of a computer program

Description automatically generated

Upon further analysis, we also identified the use of api.ipify.org, a legitimate public service commonly utilized for retrieving the user's public-facing IP address. This information can be crucial for attackers to determine the geographic location, network details of the victim which can be used to further attacks or establish connection.

A screenshot of a computer program

Description automatically generated

Since we identified that this malware sample is capturing sensitive information, we also found indications that it is extracting data such as the origin URL, username, and password. This behavior suggests that the malware may be designed to target credentials stored in browsers or applications, potentially sending the harvested information to a remote server for malicious use.

A computer screen shot of a program code

Description automatically generated

The malware sample also performs checks to detect if it is running in a virtualized environment, such as VirtualBox or a VM. It is designed to evade analysis by altering or terminating its behavior if such an environment is detected, making it more difficult for researchers to study its properties.

A screenshot of a computer

Description automatically generated

# Results and Discussion

The analysis of Agent Tesla turned out to consist of structural and behavioral components whise presence validated the malware as a complex framework for stealing data and initiating unauthorized remote access. Below are the significant findings from both static and dynamic analyses.

Static Analysis:

* Packing and obfuscation: When the initial examination of the malware was conducted it was established that it was packed using Autolt. Decoding was needed to obtain its real contents. Using Detect its easy, the obfuscation was uncloaked and using the Autolt extractor the components were unpacked so that they could be examined in detail.
* Key Features identified: When using PEStudio, there were some API imports flagged including, GetAsyncKeyState crucial for keylogging as well as InternetOpenUrlA important for data transfer through an internet connection. An entropy level of over 6 in some parts of the sample suggested encrypted or encoded contents, facts supported by Agent Tesla’s coding.
* Indicators of Process injection: The observation of process injection such as read and write process memory and processes memory was made from the analyzing with PEBear. This proved that Agent Tesla could drop code into discussed processes in order to bypass future detection.
* Communication Mechanisms: Based on a decomplication using dnSpy, the program used api.ipify.org for identification of the users IP address and the telegram API for the transmission of stolen data. This underlines the fact that, for data theft, most of the malware tend to use encrypted channels of communication.

**Dynamic Analysis:**

* Persistence Mechanisms: On analyzing using RegShot, 33 changes to the registry were noted. These were new entries in HKCU\Software\Microsoft\Windows\CurrentVersion\Run which pointed towards persistence by Agent Tesla.
* System Behavior: ProcMon clearly pointed this to me by showing the malware creating and closing files in the system directories; accessing the temporary files to store credentials, PESieve memory analysis showed several implanted PE files and several components that are typical of Agent Tesla and showed its ability to inject code.
* Keylogging and credential Harvesting: The hooks which were identified by dnSpy justified the capability of the malware to log keystrokes. The extraction contained source URLs, usernames and password to show that it mainly focuses on stored browser and application data.
* The malware and attackers are smart and nowadays even show an anti-analysis characteristics which means they stop their activity when they detect they are in virtual box.

**Discussion:**

The results found herein prove are in line with the functionalities of Agent Tesla as a RAT and spyware. An example is its ability to be packed, to use cryptographic means of transmitting information, and to act stably, which testify to its ability to penetrate conventional security systems. The fact that it uses telegram and public APIs means that it fits in modern attacking trends.

# Conclusion

The examination of Agent tesla malware showed the program’s complexity and versatility as spyware that covers the possibility of stealing credentials, keylogging, and transferring data to a command center. That’s why its work is based on the already mentioned technologically complex tools for obfuscation, encrypted communication channels, and anti-analysis mechanisms. The results stress the need to perform both the static and dynamic analysis for an effective comprehension of the essence and effects of the malware.

Using tools such as VirusTotal, PEStudio, Regshot and dnSpy, we managed to reveal functions of operation on Agent Tesla, including persistence elements, interaction with C&C servers as well as credential stealing. Such understanding is useful to support the creation of proper countermeasures as well as enhance the detection capabilities.

Possible future work is look for methods of real-time detection and combating the different types of evasion used by such malware, thus taking a more active approach to dealing with threats posed by new and growing malware families that include Agent Tesla.

# Recommendations

To mitigate the risks posed by Agent Tesla and similar malware, the following recommendations are proposed:

* Implemented Advanced Threat Detection: Employ behavioral models to suspicious activities comparing with the normal states such as registry changes, call to APIs and any attempts show injection processes, use machine learning algorithms to identify any kind of hidden or packed binaries.
* Enhanced Email Security: Employ high sophisticated filters to detect and filter out phising emails with an attempt of having malicious attachments. Employees and users should be informed on how to avoid such scams particularly those inconvenient emails that contain attachments or links.
* Regular updates and patching: Make sure the operating systems, applications are updated since malware operating them may take advantage of security back doors.
* Strengthen endpoint security: Using endpoint protection that can scan real-time files and process activities in the computer. Apply application control which concerns the operation of unauthorized programs only that has been authorized to run.
* Network security measures: Supervise excess centric networks traffic for abnormity, for example, attempt to connect to bizarre Ips or C2 servers and enabling firewalls.
* Forensic Readiness: It is necessary to keep logs of system and network in particular to facilitate malware analysis and incident response activities. Use RegShot and Process monitor on a regular basis for the registry and process activity analysis.
* Research and collaboration: Adapt collections of new agent tesla variants and methods by syncing with other cyber security related subgroups. Exchanging IOCs with threat intelligence platforms must be done to strengthen the global network of protection.

By doing so, the organization can greatly minimize the chances of getting caught up with such malware as Agent tesla and thereby improving the steadiness of organizations against such emerging threats.

# References

Coan, J. (2022, June 14). Cybersecurity Skills: Dynamic and Static Malware Analysis. CyberMaxx. <https://www.cybermaxx.com/resources/dynamic-and-static-malware-analysis/>

Dedenok, R. (2022, September 26). Trojan-stealer discovered in spam mailouts to businesses. Kasphersky. <https://www.kaspersky.com/blog/agent-tesla-spam-mailout/45621/>

Ramanl. (2024, April 2). Agent Tesla Targeting United States & Australia: Revealing the Attackers’ Identities. Check Point Research. <https://research.checkpoint.com/2024/agent-tesla-targeting-united-states-and-australia/>

*MalwareBazaar | Browse Checking your browser*. (2024, February). Retrieved October 25, 2024, from <https://bazaar.abuse.ch/sample/e17062f3e40417b32b67892e68cd134a6b5ea179e75182749ced9249fe049fa4>

Devanny, J., Martin, C., & Stevens, T. (2021, September 2). *On the strategic consequences of digital espionage*. Journal of Cyber Policy. <https://doi.org/10.1080/23738871.2021.2000628>

Noorbakhsh-Sabet, N., Zand, R., Zhang, Y., & Abedi, V. (2019, January 31). *Artificial Intelligence Transforms the Future of Healthcare*. PubMed Central (PMC). <https://doi.org/10.1016/j.amjmed.2019.01.017>

Seals, T. (2021, July 8). Oil & Gas Targeted in Year-Long Cyber-Espionage Campaign. *Threat Post*. <https://threatpost.com/oil-gas-cyber-espionage-campaign/167639/>