***Malware Analysis***

1. **Introduction**

“Any sufficiently advanced technology is equivalent to magic.” - Arthur C. Clarke.

This is a famous quote that shows how much technologies had evolved and how it became like what we used to only can image as magic a few hundreds of years ago. This is a sign that lets us see how much technologies have and will growth in the future and it will be more rapidly than ever as it plays an irreplaceable part in everyone’s life nowadays. But as technologies growth, there are many who want to use it for malicious purposes, and one of the most common ways to do so is through malware. This method is also listed as one of the most dangerous and hardest ways to counter as virus developers always update their programs to attack in many different areas.

So, what exactly is malware and why does it have such a dangerous impact on our today life. Malware is a short term for malicious software which is developed by cybercriminals or hackers to steal information and destroy or damage computers and computer systems. Some of the most common malwares are viruses, worm, Trojan, spyware, adware, and ransomware (Cisco. (n.d.). *What Is Malware? - Definition and Examples*.). To counter such measurements, it is essential to rely on malware analysis specialists as they help us understand more about how different malware works and how to stop them. In today’s report, we will learn more about how these common types of malwares work and look in deep at how malware analysis conduces their work.

1. **Malware Type**
2. Ransomware:

Ransomware is encryption software that blocks the target from accessing their data until a ransom is paid. This results in disoperation partially or totally until the payment of the ransom, but there is no guarantee that after the payment, the hacker will give the key for decrypting the data or the given key is working or not (Baker, K. (2021). *11 Types of Malware + Examples That You Should Know*).

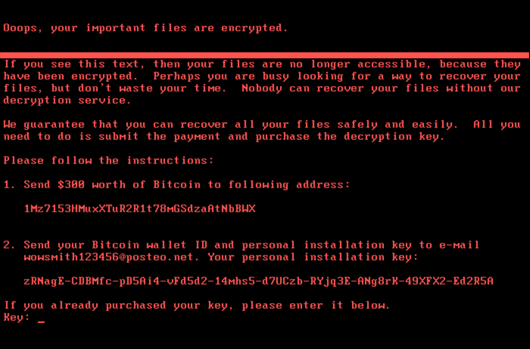


Figure 1: Example of a ransom letter

(Source: <https://www.crowdstrike.com/cybersecurity-101/malware/types-of-malware/>)

1. Spyware:

Spyware gathers the information of the target activities without their knowing or consent. This information can be password, pins, payment information. Spyware can operate in many platforms and applications, making it very versatile and hard to spot (Baker, K. (2021). *11 Types of Malware + Examples That You Should Know*).

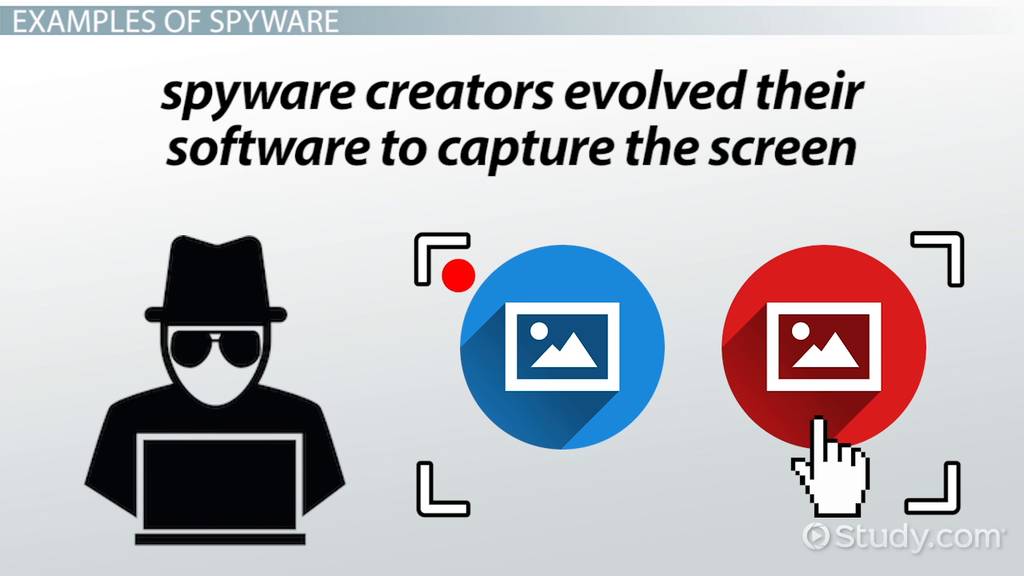


Figure 2: Example of hacker can use camera to capture sensitive data.

(Source: https://study.com/learn/lesson/spyware-types-examples.html)

1. Adware:

Similar to spyware, adware also collects information on the target activities, but adware doesn’t install any software in the victim system and doesn’t capture victim keystrokes.

The danger of adware doesn’t come from stolen data but from user privacy, the malware will collect user internet activities and create a personal profile of that person and that information will be sold to advertisers without the victim’s consent (Baker, K. (2021). *11 Types of Malware + Examples That You Should Know*).

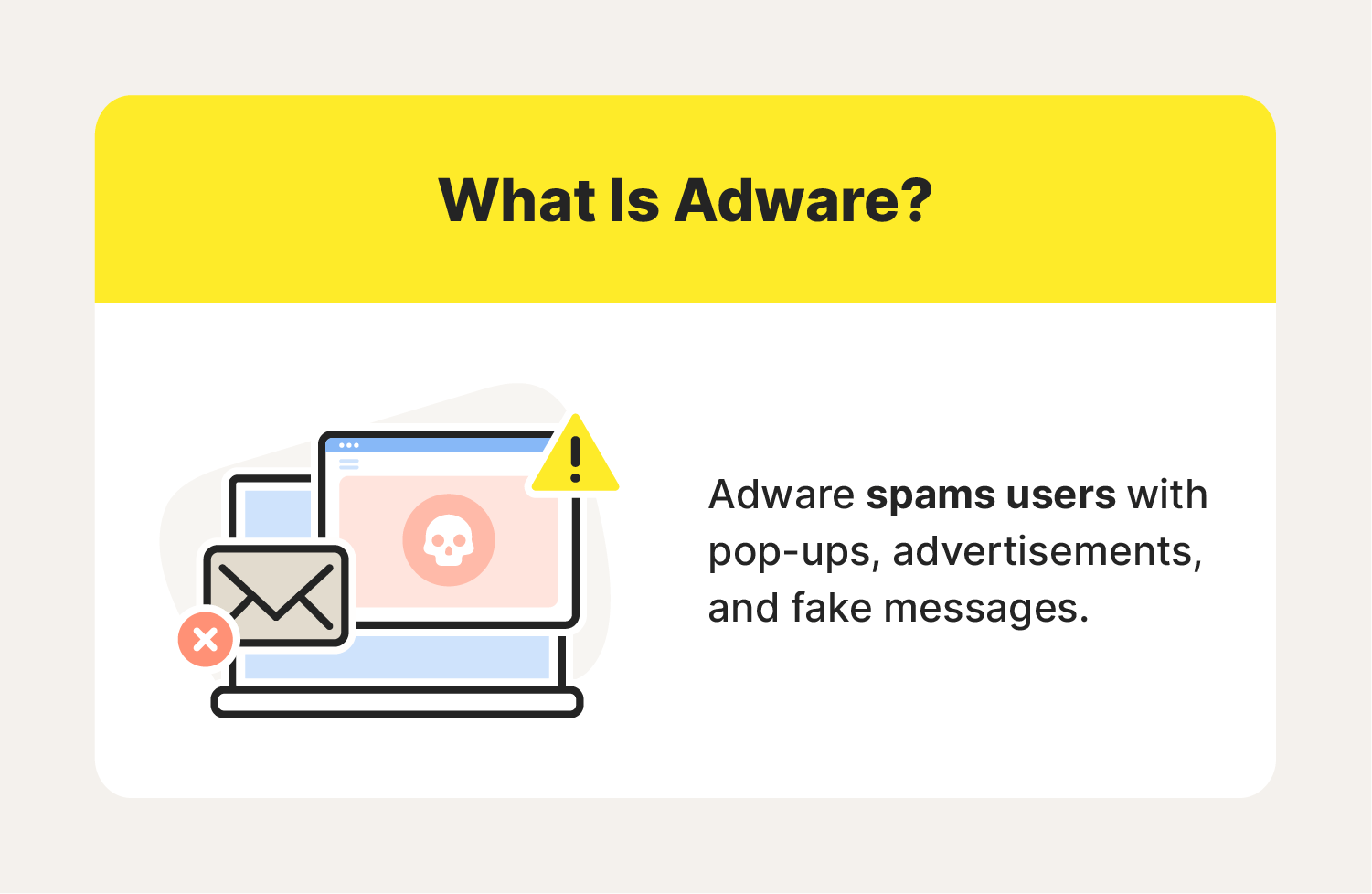


Figure 3: Adware can spam user pop-ups, advertisement

(Source: https://us.norton.com/blog/malware/adware)

1. Worm:

Worms install themselves into the networks through the vulnerability in the operating system, this can either by going through the backdoor built into the software, the unintentional software vulnerability, or through flash drive. Them create risk of losing sensitive data, target of ransomware or DDoS attack (Baker, K. (2021). *11 Types of Malware + Examples That You Should Know*).

1. Trojan:

A Trojan is software that disguises itself as target desirable code or software, one install into the system can cause the victim system control fall into the hacker hand for malicious purposes. Trojan can disguise itself in many types of software such as games, apps, software patches or they can be embedded in attachment included in phishing email (Baker, K. (2021). *11 Types of Malware + Examples That You Should Know*).

1. Virus:

Like Trojan, viruses have the same pattern and harm, but unlike trojan, viruses are not software but a malicious code that insert themselves in the software and can only be executed when the infected app is running. This is the main difference between virus and trojan, trojan only needs the software to be downloaded to be execute and virus need the app it in to run to be execute (Baker, K. (2021). *11 Types of Malware + Examples That You Should Know*).

1. **Malware Analysis**
2. What is malware analysis/ what does malware analysis do?

A malware analysis specialize job is to study the unique features, objective, sources, and potential effect of the malware. By doing this, malware analysis can understand how these malware various from different kinds and help incident respond and security analysts to identify the incident by level of security, uncover hidden indicators of compromise that should be blocked, improve the IOC alert and notifications more effectively, and enrich context when threat hunting (Fortinet. (n.d.). *What is Malware Analysis? Types and Stages of Malware Analysis*.).

1. How to detect a malware in the system?

A malware is an unwanted program in your system, due to this, it can cause odd change in the system, these change can be:

* Sudden shutdown, slow speed or many error messages display
* Can’t shutdown or restart
* Software can’t be remove
* Many pop-up tab and advertisement
* Unexpected toolbar or icons appear in your browser or desktop
* Open a website that you not intended to open

(Consumer Information. (2021). *How To Recognize, Remove, and Avoid Malware*.)

1. Type of malware analysis

To able to understand how the malware operate, malware analysis will often analysis them by 3 ways to know their behavior more in deep.

* 1. Static analyze

Static malware analysis is not required to run the code to analysis, but instead the analysis will look into the sources code to identify malicious intent. By reading the code manually, malicious infrastructure, libraries or packed file can be identified.

To understanding a file is malicious or not, technical indicator is identified such as file names, hashes, strings, such as IP address, domains, and file header data. To able to read the sources code, it is encouraging to use tool like disassemblers and network analyzers without running the code (Fortinet. (n.d.). *What is Malware Analysis? Types and Stages of Malware Analysis*.).

As an example of a static analysis, I will analyst a file call “statistic.exe”

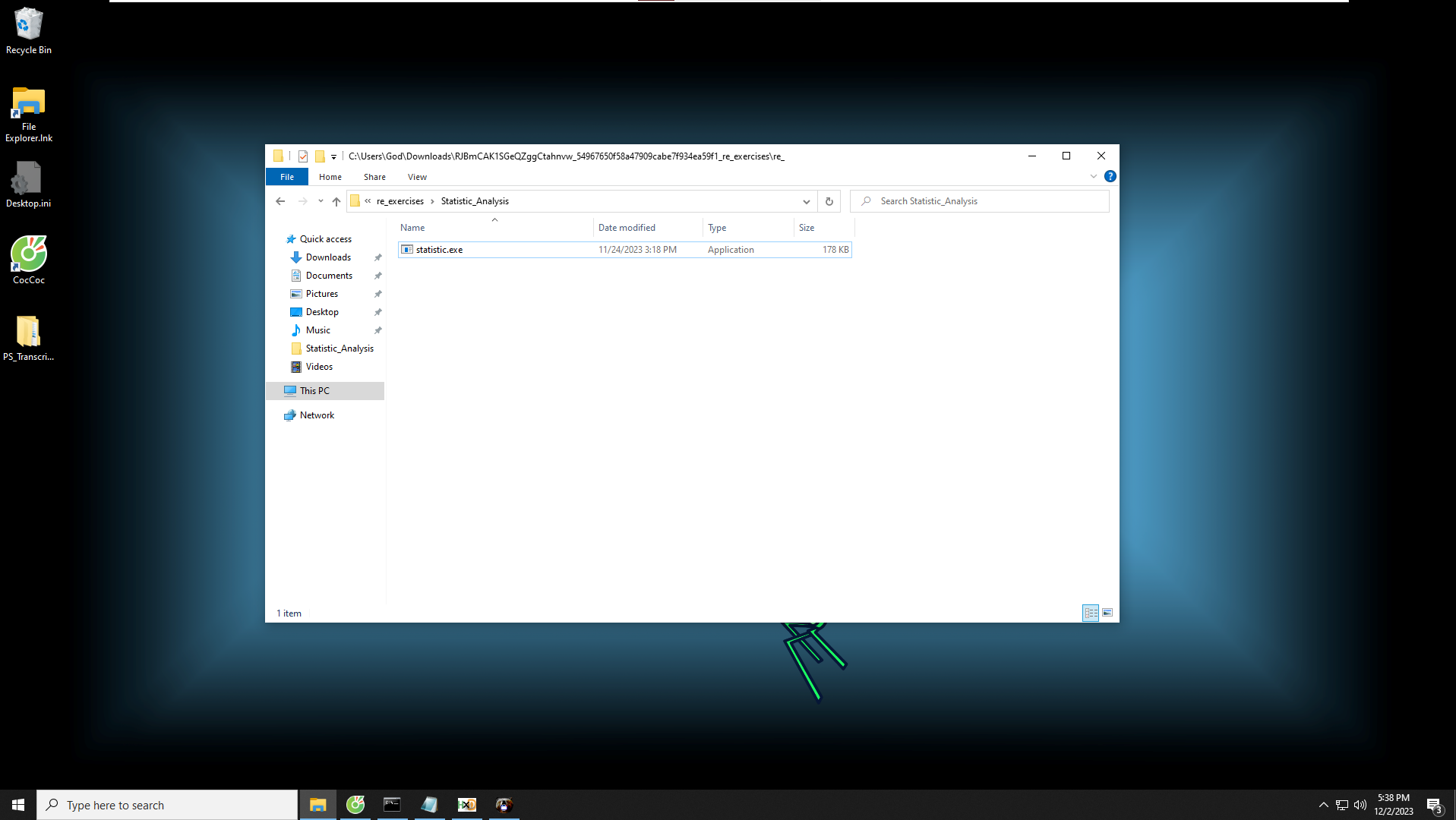


Figure 4: Statistics.exe file

First I run it with DIE (Detect it easy) to look at it file type and compiler

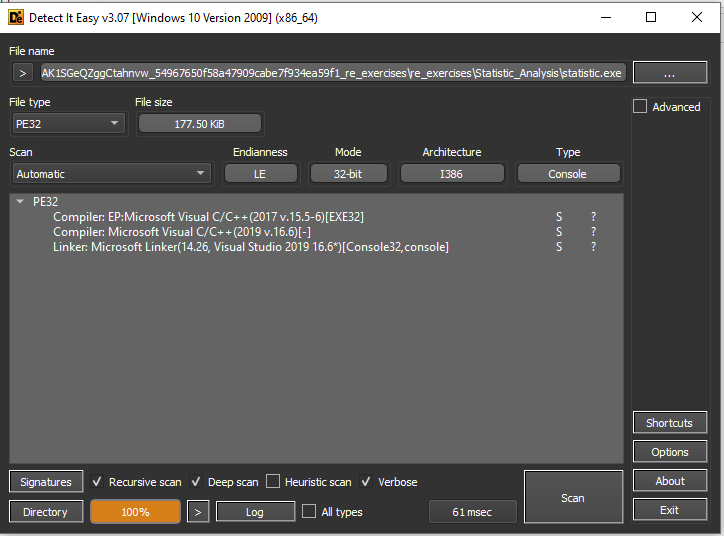


Figure 5: Running DIE to identify “statiscs.exe”

Then I can run HXD for the content of the code and use PE-Bear for clear view

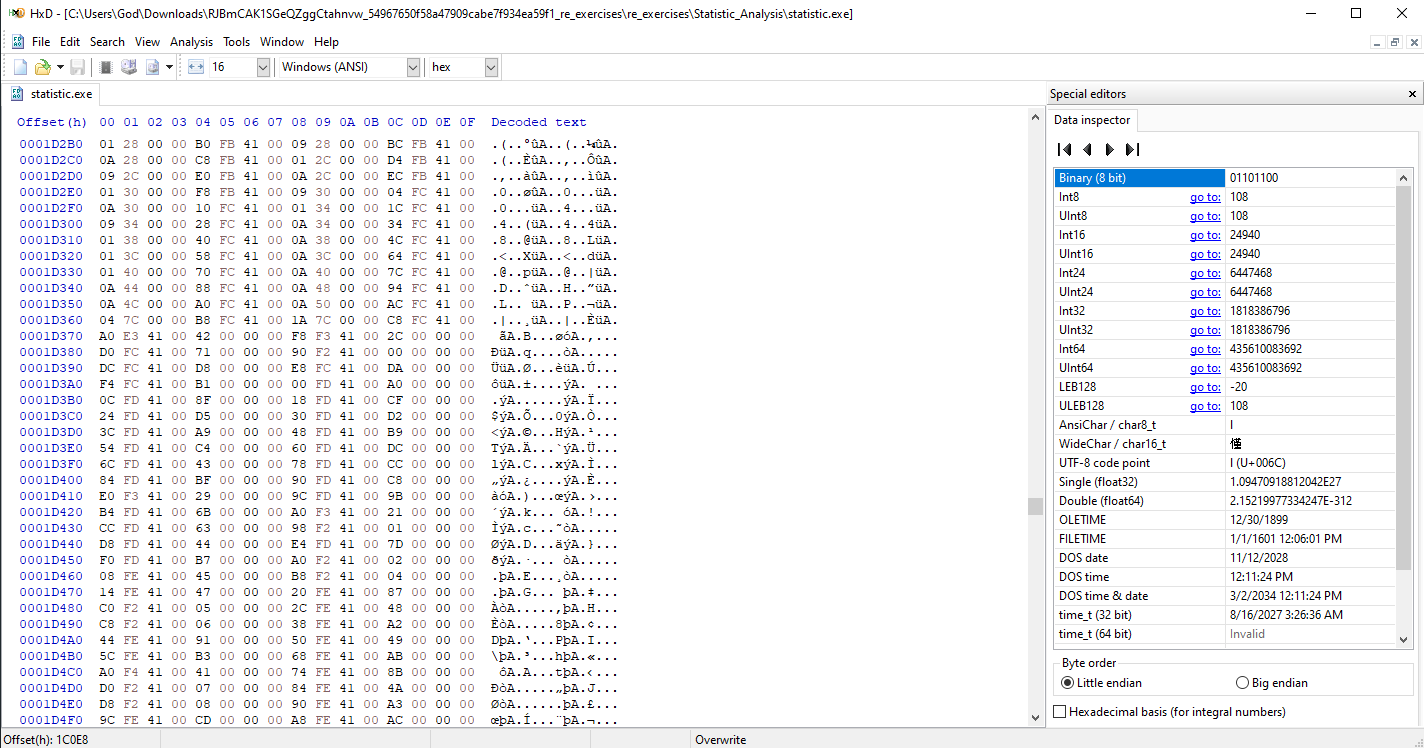


Figure 6: HXD showing the source code

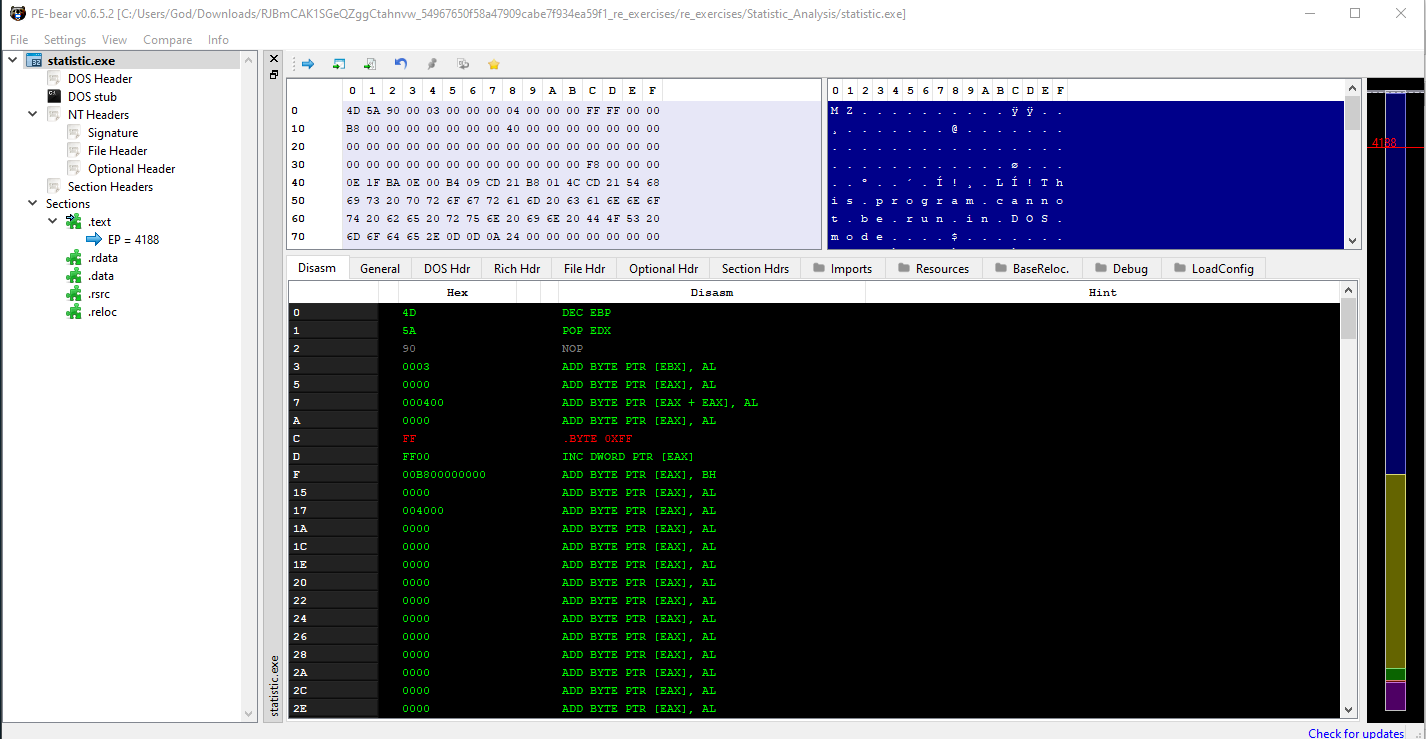


Figure 7: PE-bear giving a better look at the source code

By using PE-bear, we can see more than just the source code, we can the hash, checksum, file size and more

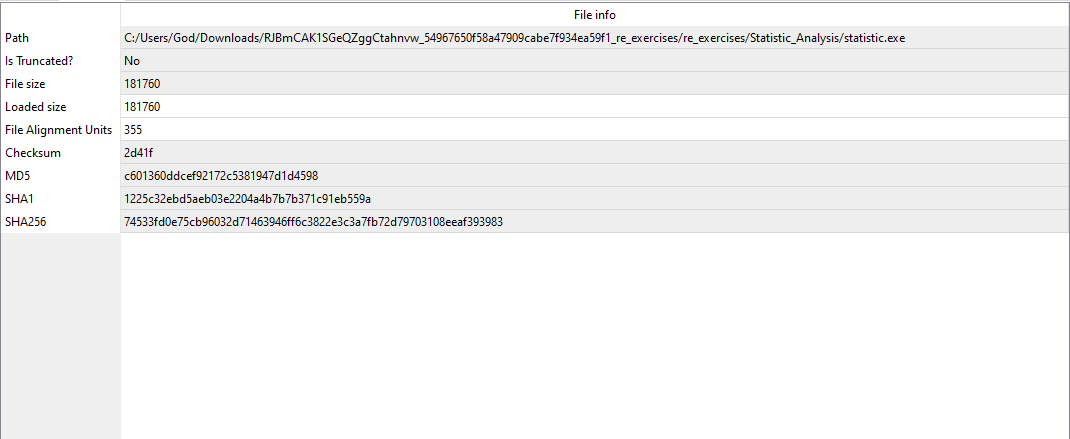


Figure 8: other content that PE-bear show

Finally, it is best to view it in strings format for easier understand, to do so let convert the exe file to txt file using strings command and see what useful in the source code

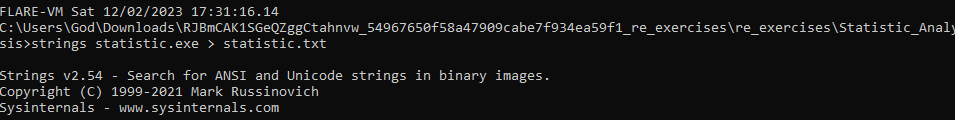


Figure 9: Strings command use on statiscs.exe

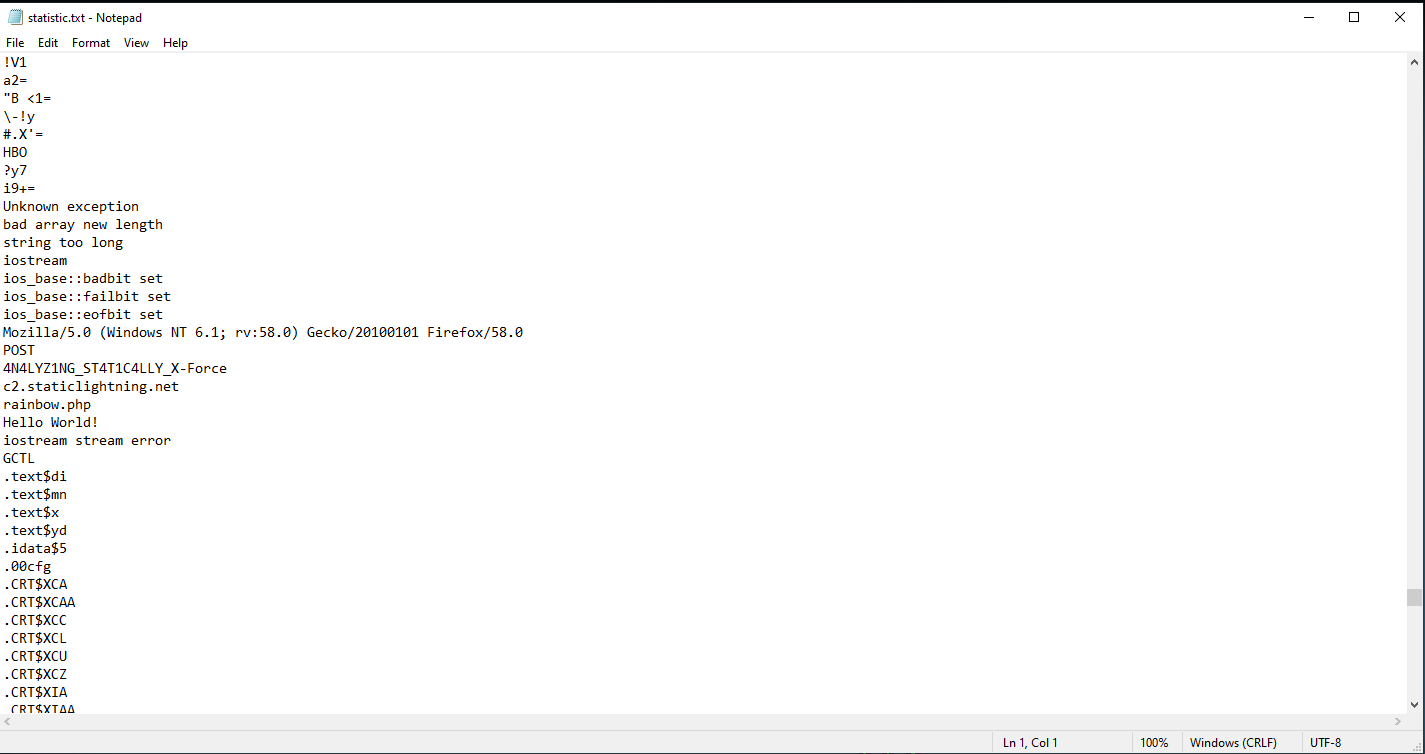


Figure 10: String txt file of statiscs.exe

In the txt file, I can see the flag that I need to found and a url link and a php file call “rainbow.php”

* 1. Dynamic analyze

In contrast with static analysis, dynamic analysis doesn’t required analyzers to read the code, but instead observing the malicious program behaver in a safe environment call a “sand box”. By observing the code in a closed system let the security professionals understand the changed it made without affecting the main system or infecting the main network.

Despite the safe environment, hacker know that “sand box” are out there and they have become very good at detecting them. Thank to this, attacker often hide their code and only let the code run when it meat a certain condition (Fortinet. (n.d.). *What is Malware Analysis? Types and Stages of Malware Analysis*.).

Like static analysis, I will also use dynamic analysis on “dynamo.exe”

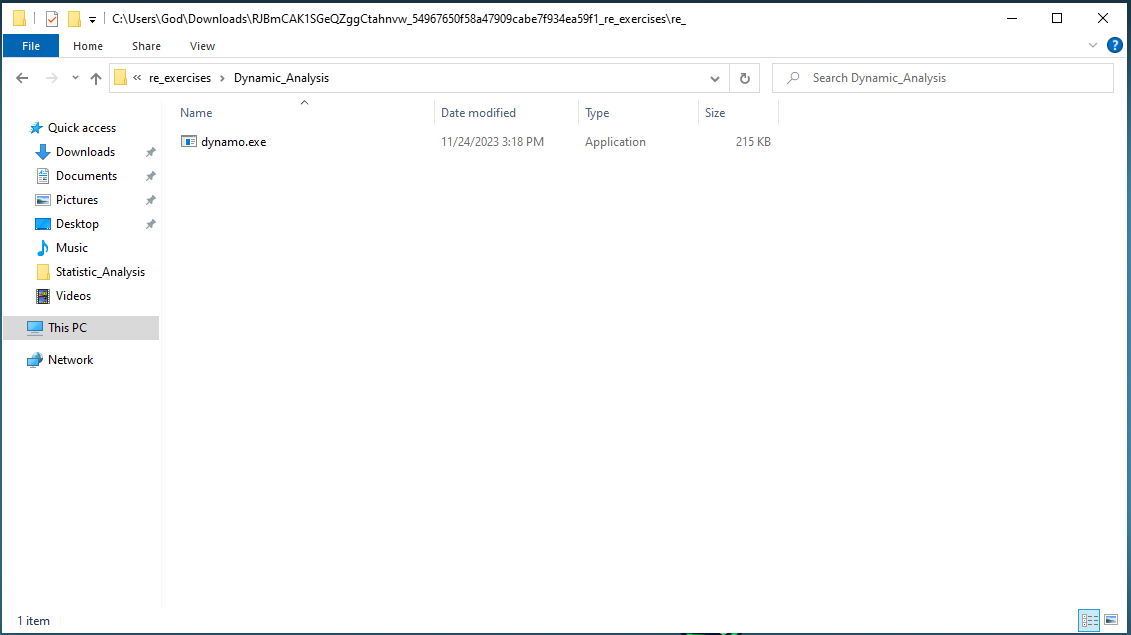


Figure 11: dynamo.exe file

The first step of identify what the malware does is to run it and record the change to the system, for this I use a tool call Regshot. Regshot allow you to take 2 shot, one from before running the program and one after executing it and show the change of the system.

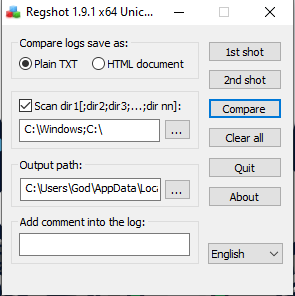


Figure 12: Regshot record 2 shot for comparison

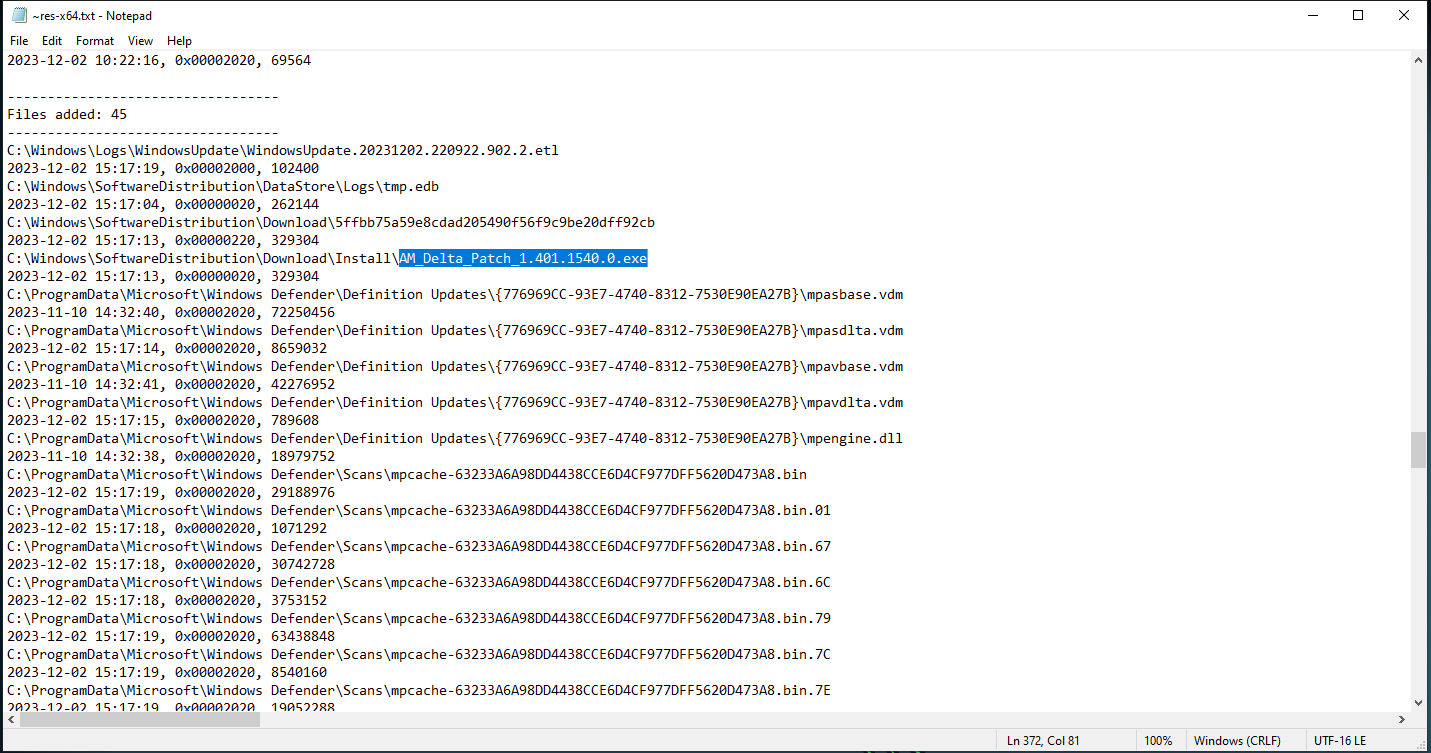


Figure 13: The compare txt file shows 45 files were added after running the program

By using wireshark, we can see that rainbow.php had been added to our system, this can also be see in Fakenet.

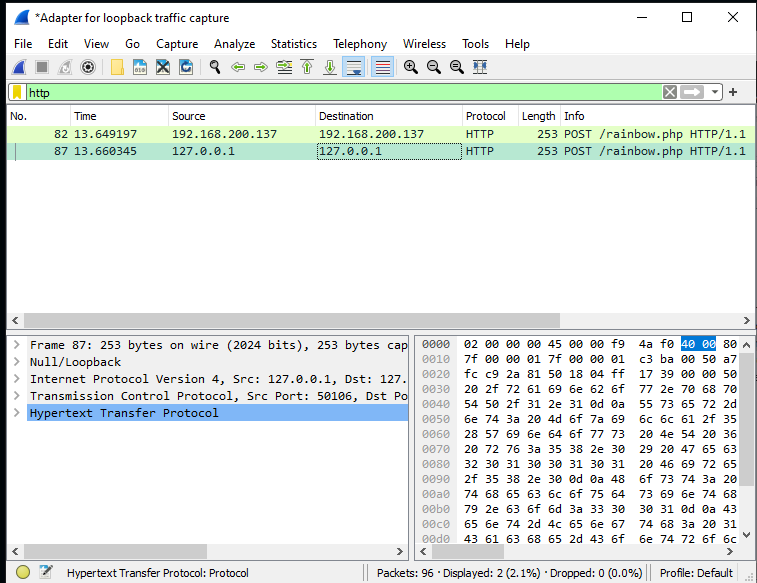


Figure 14: wireshark showing bad php file

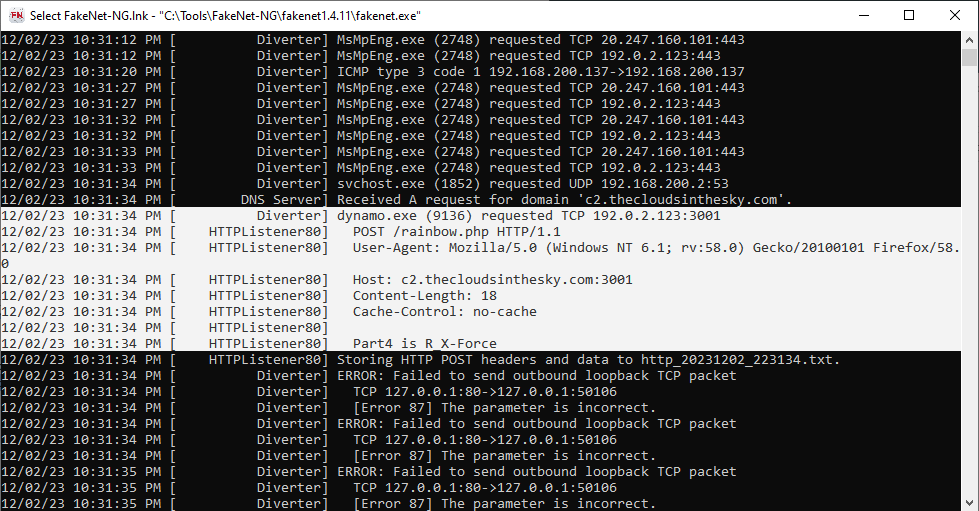


Figure 15: Fakenet show the full request content

* 1. Hybrid analyze

Hybrid analysis, as the name suggests, is the combination of both static and dynamic analysis. By mixing the two types of analysis, it covers the weakness of both types. Doing basic static analysis is not too reliable when trying to detect sophisticated malicious code, and doing basic dynamic analysis can sometime be difficult when some malicious can hide itself from the present of sandbox technology. That is why it is recommended to use hybrid as it is considered the best way to analyze malware as we can look at the source code to see the running condition of the malware and can extract more indicators of compromise(IOC). After finding the condition for executing the code, we can observe how it behave to understand it more, even from the most sophisticated malware.

For example, when the system detects a malicious code is running and generates data after a dynamic analysis, the analyst will take that piece of data to perform a static analysis. As a result, more IOCs would be generated, and zero-day exploit would be exposed.

1. The 4 stages of malware analysis

To be able to conduct a malware analysis as detail as possible, analysts often need to follow 4 stages with various of task, some are simpler than other. These stages layered on top of each other and form a pyramid that grows upwards in complexity. The higher you get in the pyramid, the more skill and effort you need to make to be able to accomplish the tasks.

* 1. Fully Automated Analysis

The first and easiest step is to use a fully-automation tool to scan suspicious files that are running in the system. These tools often identify what the specimen might do if run in the system. Other than showing what those files do, it also gives detailed information such as the registry key, its mutex values, file activity, network traffic, etc.

Due to these tools are fully automatic, they don’t provide much insight as a human analyst would when examining the specimen manually. However, thanks to their automatic nature they can handle what analysts can’t, which is to handle large amount of malware. This helps analysts to save a huge amount of time when they only need to focus on the case that truly requires an indeed sight (zeltser.com. (n.d.). *Mastering 4 Stages of Malware Analysis*.).

* 1. Static Properties Analysis

In this stage, analysts will take a closer look into the suspicious file by examining its static properties. This detail can easily be obtained thanks to not needing to run the program. When taking a closer look at the static properties, we can see the strings embedded into the file, header details, hashes, embedded resources, packet signatures, metadata such as the creation date, etc.

Sometime, by looking at the static properties can be sufficient for defining basic indicators of compromise. This stage is mainly used to determine whether the analyst should take a closer look into the malware with more comprehensive technique and where to focus the subsequent steps (zeltser.com. (n.d.). *Mastering 4 Stages of Malware Analysis*.).

* 1. Interactive Behavior Analysis

After automation and examining static property, if the analyst decides to take the malware into account for deeper investigation. This often includes running it in a safe environment to observe its behaver, this will show the analyst its registry, file system, process, and network activities. Knowing how the program uses memory could be a huge help. Other than running the malware in an isolate sand box, the analyst can also attempt to connect it to a particular host, this is used to mimic the experience to see what the malware would act after it is able to connect. By doing this can help the analyst have a deeper understanding about the characteristics that applies to files, registry keys and other dependencies that the malware might have. Interactive behavior analysis is different from the 2 first steps due to how much time consuming and requires more skill to perform (zeltser.com. (n.d.). *Mastering 4 Stages of Malware Analysis*.).

* 1. Manual Code Reversing

The final stage of the 4 stages is Manual Code Reversing, where the analyst takes the source code to investigate further for more valuable data. Only executing the malware won’t give the full characteristic of the specimen, by manual reversing the code, we can learn:

* + Decoding encrypted data stored or transferred by the sample.
  + Determining the logic of the malicious program's domain generation algorithm.
  + Understanding other capabilities of the sample that didn't exhibit themselves during behavior analysis.

To be able to reverse the code, a disassemble and a debugger is needed, which could be aided by various of tool and plugins. Memory forensics can assist at this stage of the pyramid as well. This is the most difficult stage as it requires a high level of skill and a large amount of time, but it is recommending to be at least have some basic level of reversing code to understand how it work (zeltser.com. (n.d.). *Mastering 4 Stages of Malware Analysis*.).

1. Analyzing malware

To give example how malware analyst does their job, I will give some example of different type of malware that are commonly use how to analyze them with various of tool.

* 1. Code reversing

As explained about, code reversing is the process of view the source code and make it readable for the analyst. This process often required a disassemble and a debugger, in this example I will use various of tool to revers a malware call “furiousfast.EXE” include: x64dbg, dnSpy, IDA, Ghidra.

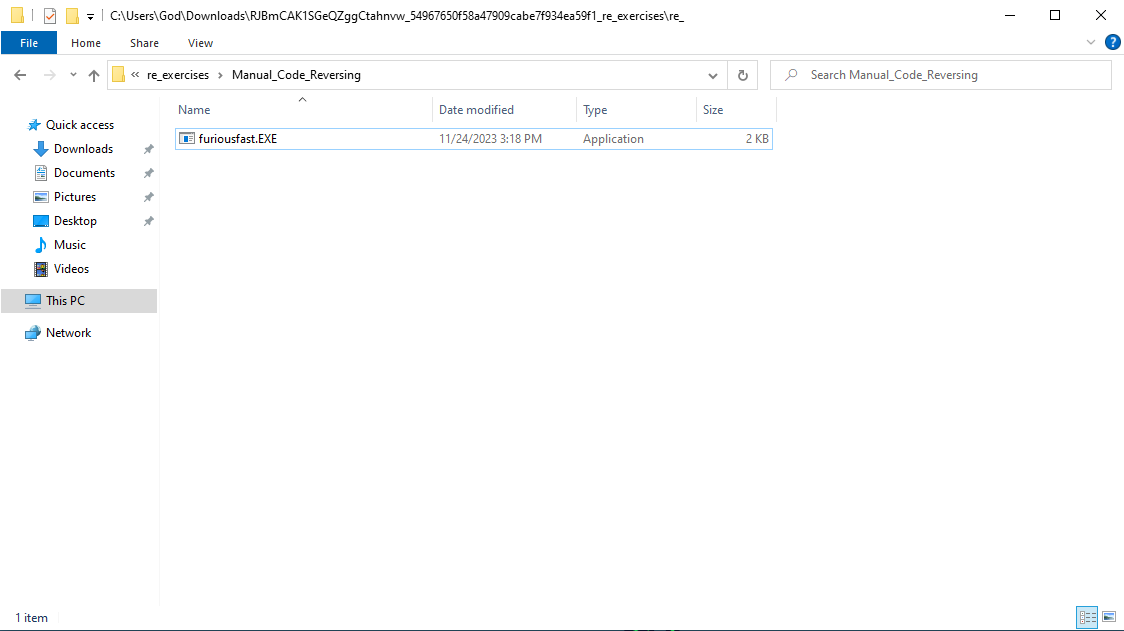


Figure 16: furiousfast.EXE file

First, I run DIE to see it file type and compiler

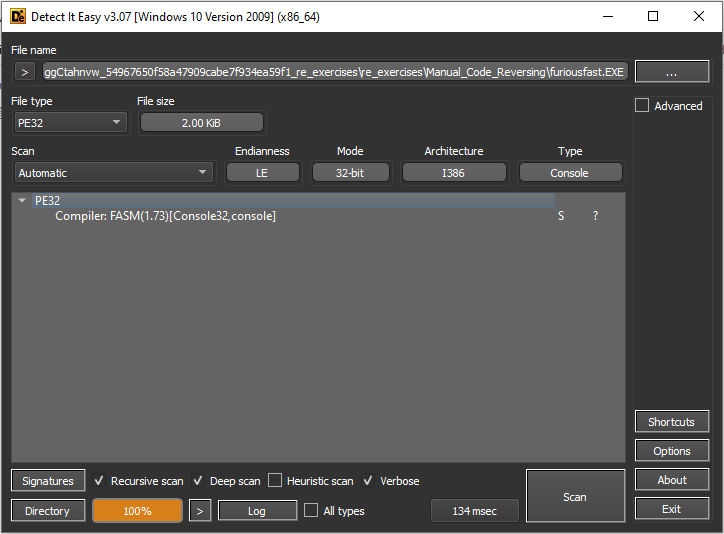


Figure 17: using DIE to identify furiousfast.EXE

After learning the file type and complier, I run snSpy for further detail

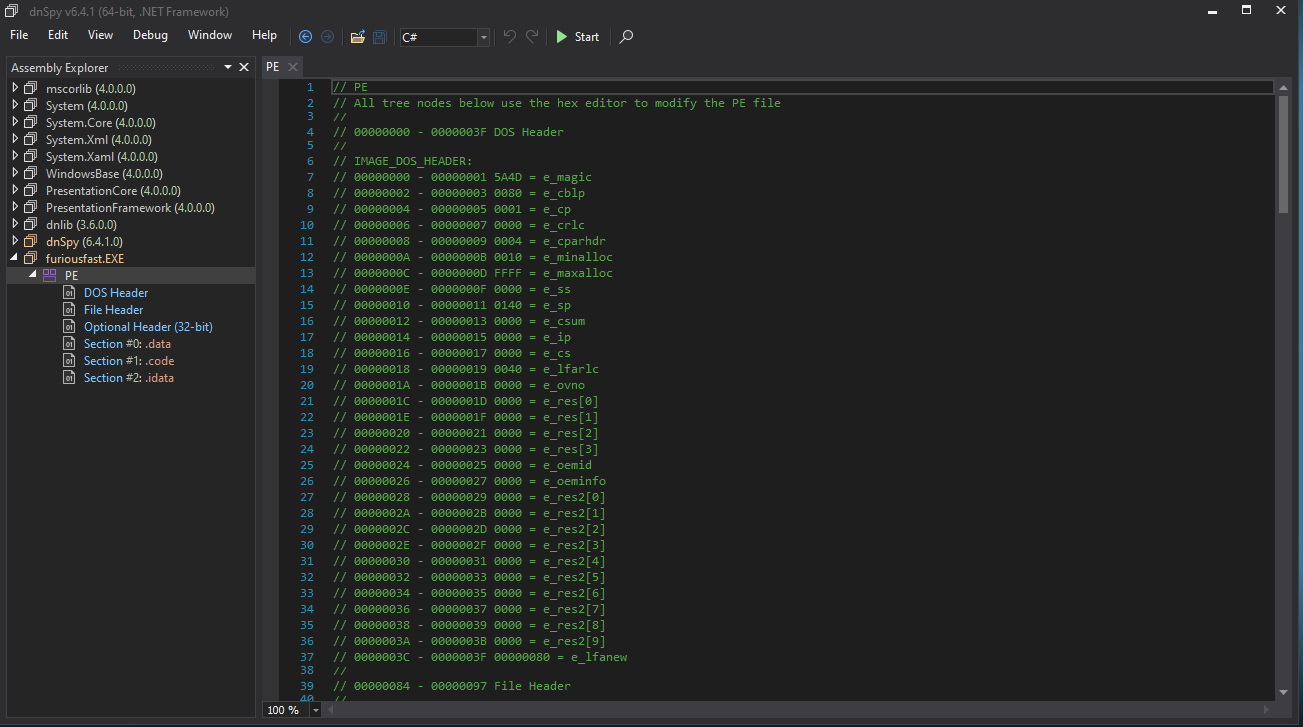


Figure 18: dnSpy showing header file

The dnSpy is a great tool to learn about the malware header and information about the file, then it is recommended to use IDA to learn about the program function and how they connect, it also show the start function which is useful when you want to run it in x64dgb

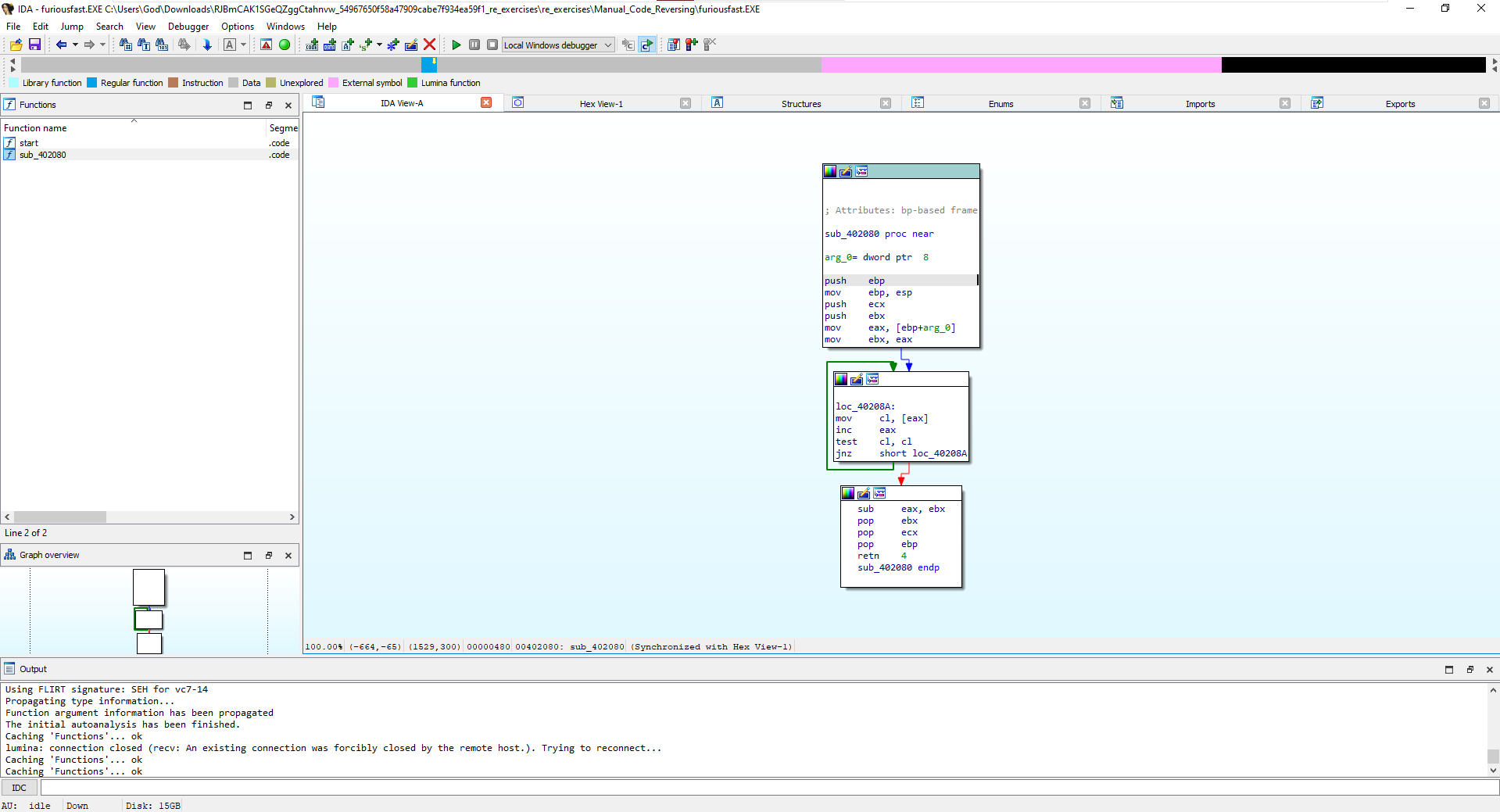


Figure 19: IDA showing the connection between function

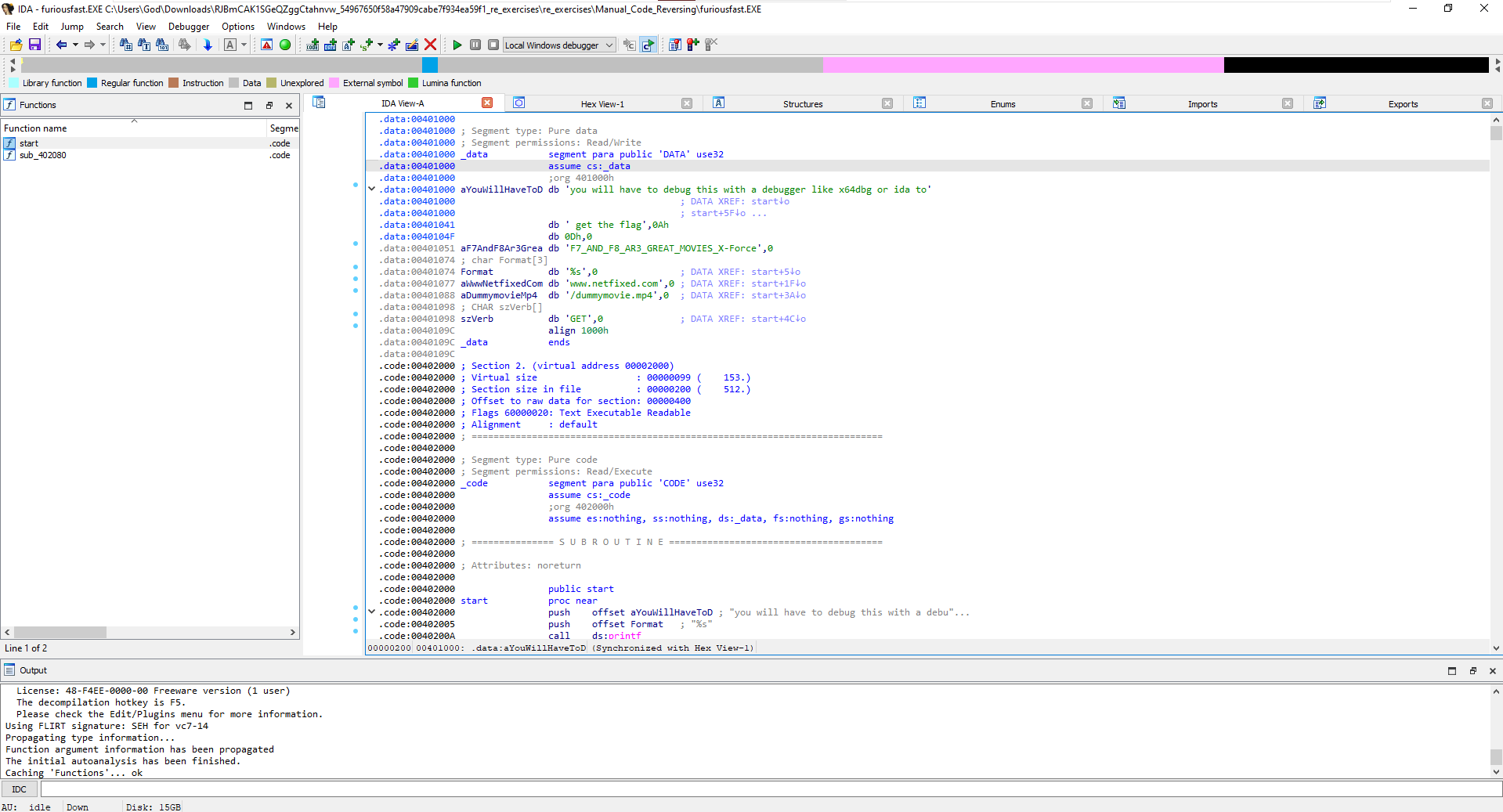


Figure 20: IDA showing the source code of the file

Reading the disassemble file, we know the flag, the link that the code connects to name “www.netfixed.com”, a mp4 file call “/dummymovie.mp4”, and the most important is the start function address “00402000” so we can debug in x64dgb.

Similar to IDA, ghidra is also a very powerful tool to reverts the code, but ghidra also can translate the code from assemble to C for further and easier understanding.

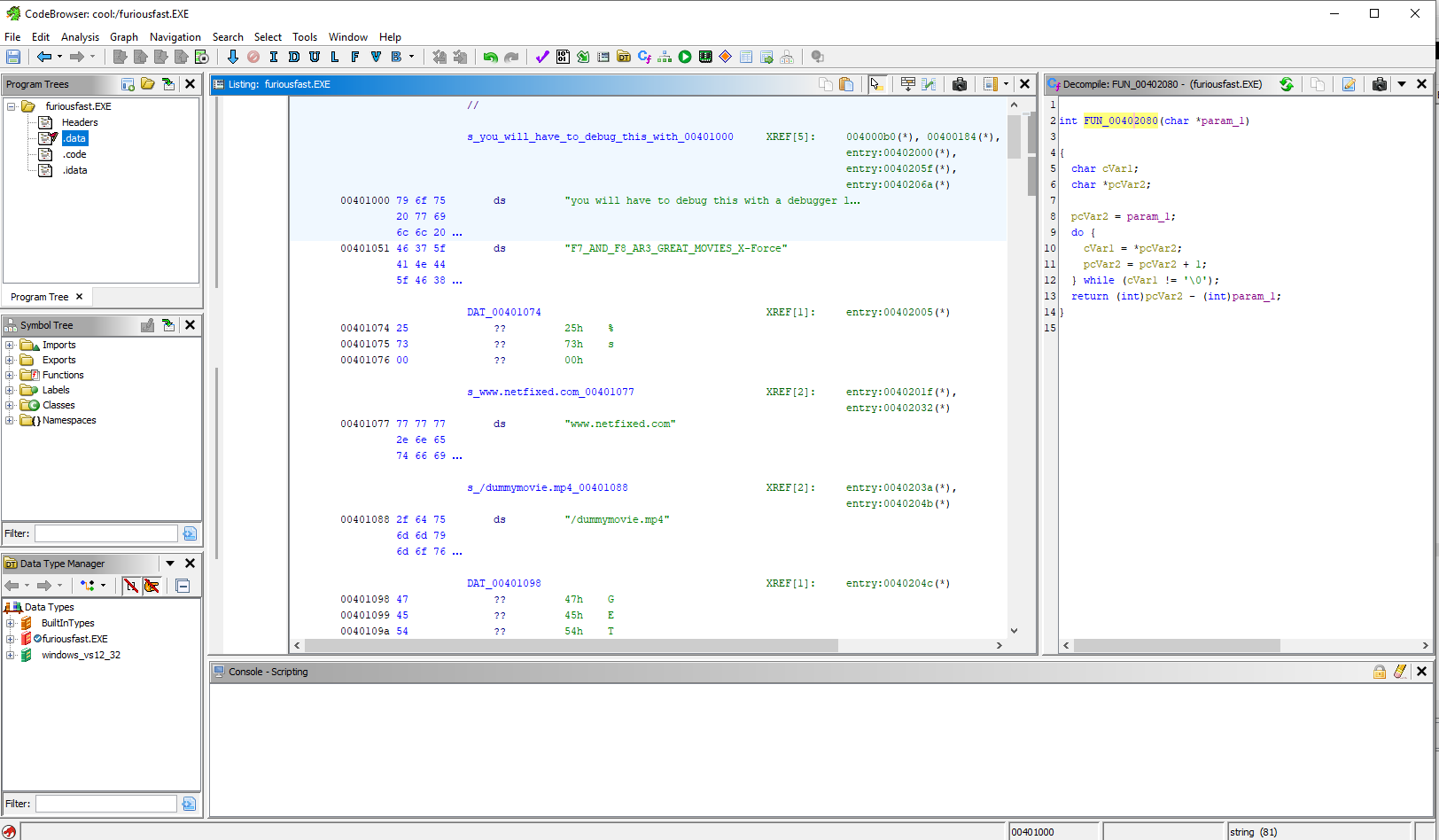


Figure 21: Ghidra give the C translation of the code

Finally, if you want to see how the malware run in it environment then analyst will begin to debug it with x64dgb, by placing a breakpoint in the code we can see how each line of code being executed starting from the breakpoint, that why the start function address is so important, it let the analyst know where to place the breakpoint. After setting the breakpoint, just step over to view each line and what it does.



Figure 22: x64dgb find function to look for the start address

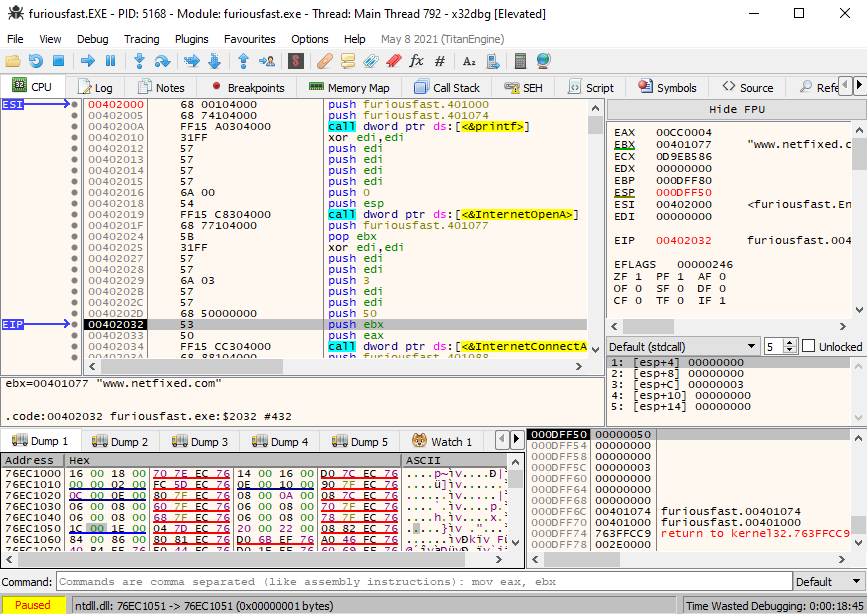


Figure 23: Running the code step by step in x64gdb

* 1. Analyzing Fileless Malware/PowerShell scripts

Fileless malware or zero-footprint attacks most common uses PowerShell to execute attacks on your system without leaving any traces. This attack is dangerous due to not having a file to act as the infection source, which make it hard to detect by anti-virus (www.varonis.com. (n.d.). *What is Fileless Malware? PowerShell Exploited*.). To analyze a PowerShell script name “MegaProject.ps1 “, I will use PowerShell ISE in this example.

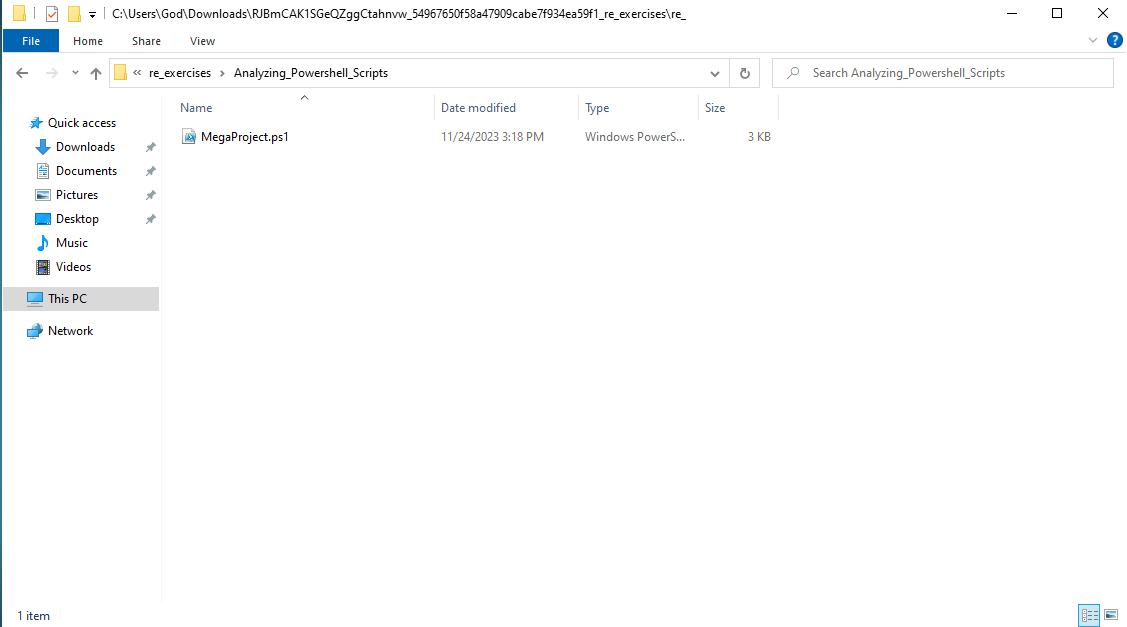


Figure 24: MegaProject.ps1 file

The first step is to check it with DIE, even with the extension of ps1, DIE show this as plain text.

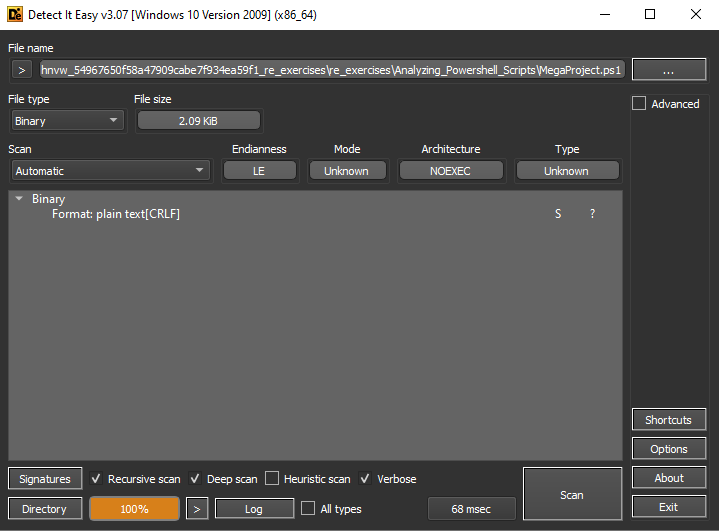


Figure 25: using DIE to identify MegaProject.ps1 file

Running the file in PowerShell ISE show a very difficult format to read, so it is best to make it more readable by make a new after each”;”.

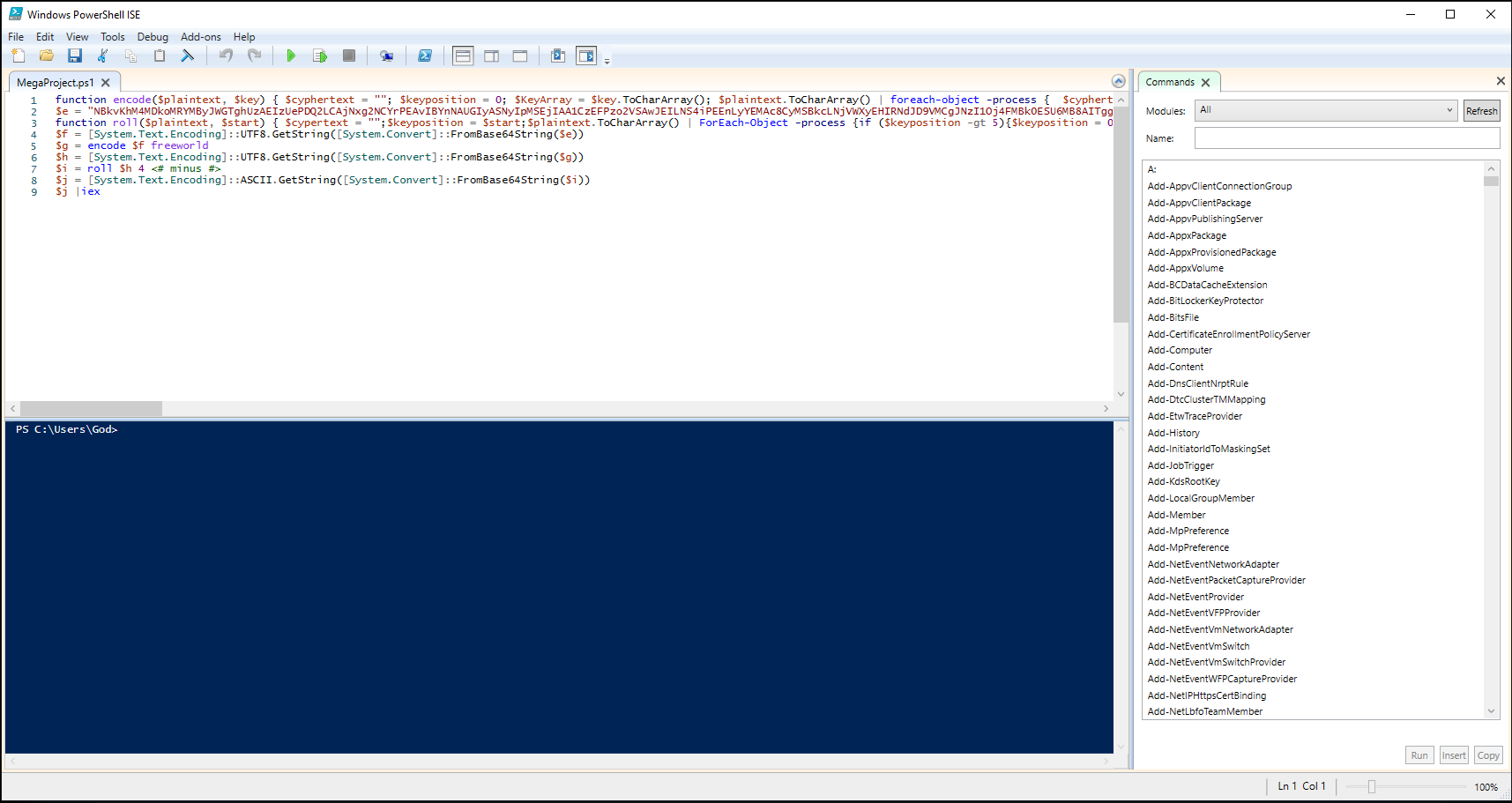


Figure 26: Viewing the source code in PowerShell IDA

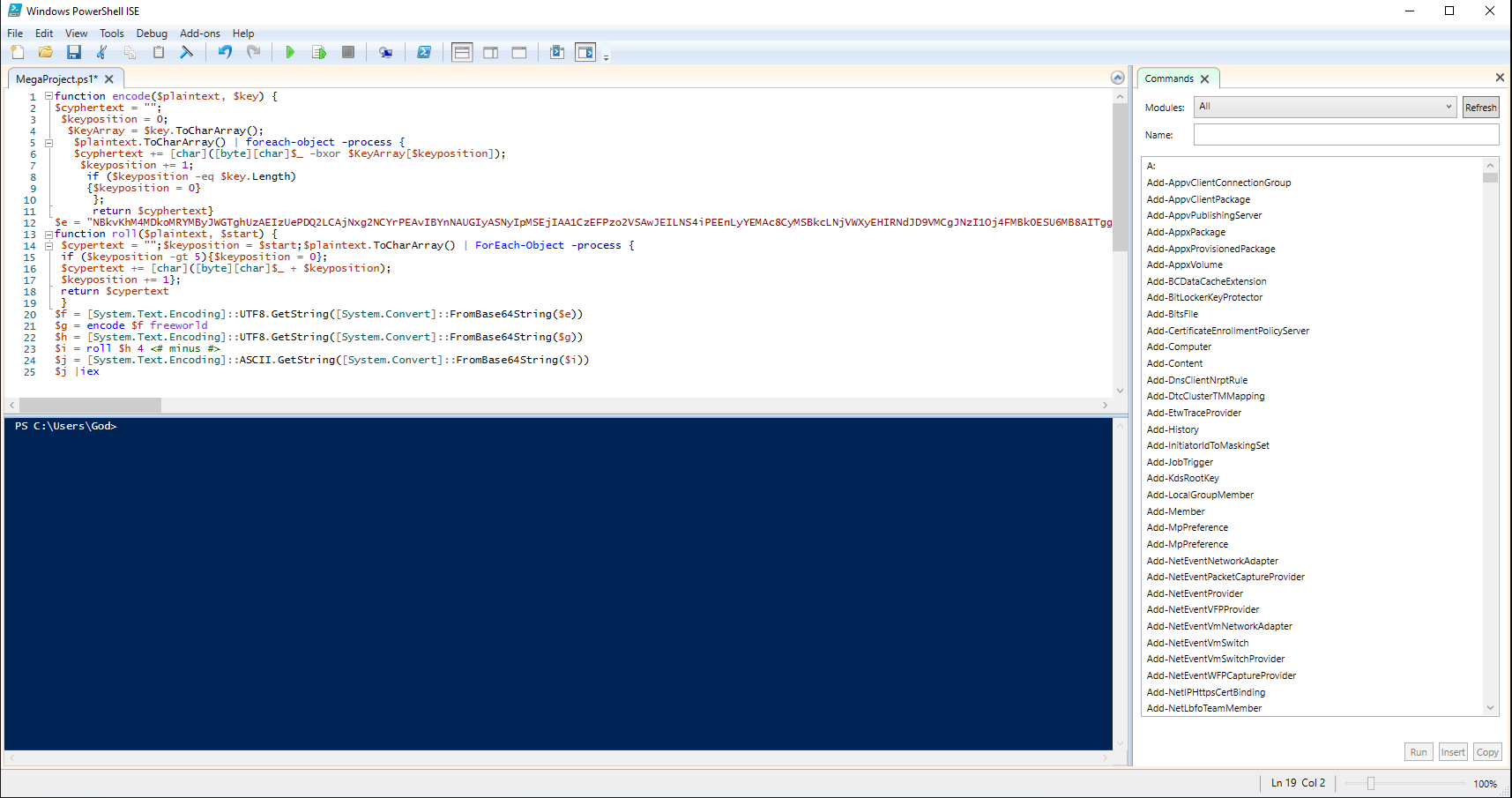


Figure 27: Organizing the code

After reorganize the code, the next step is to set break point so we can see when the code reaches the break point, what have change in the program, I will set breakpoint in each word variable $e, $f, $g, $h, $i and $j then run the code, when setting break point, it is important to set it before iex command due to when the code reach this line, the code will run and stop immediately.

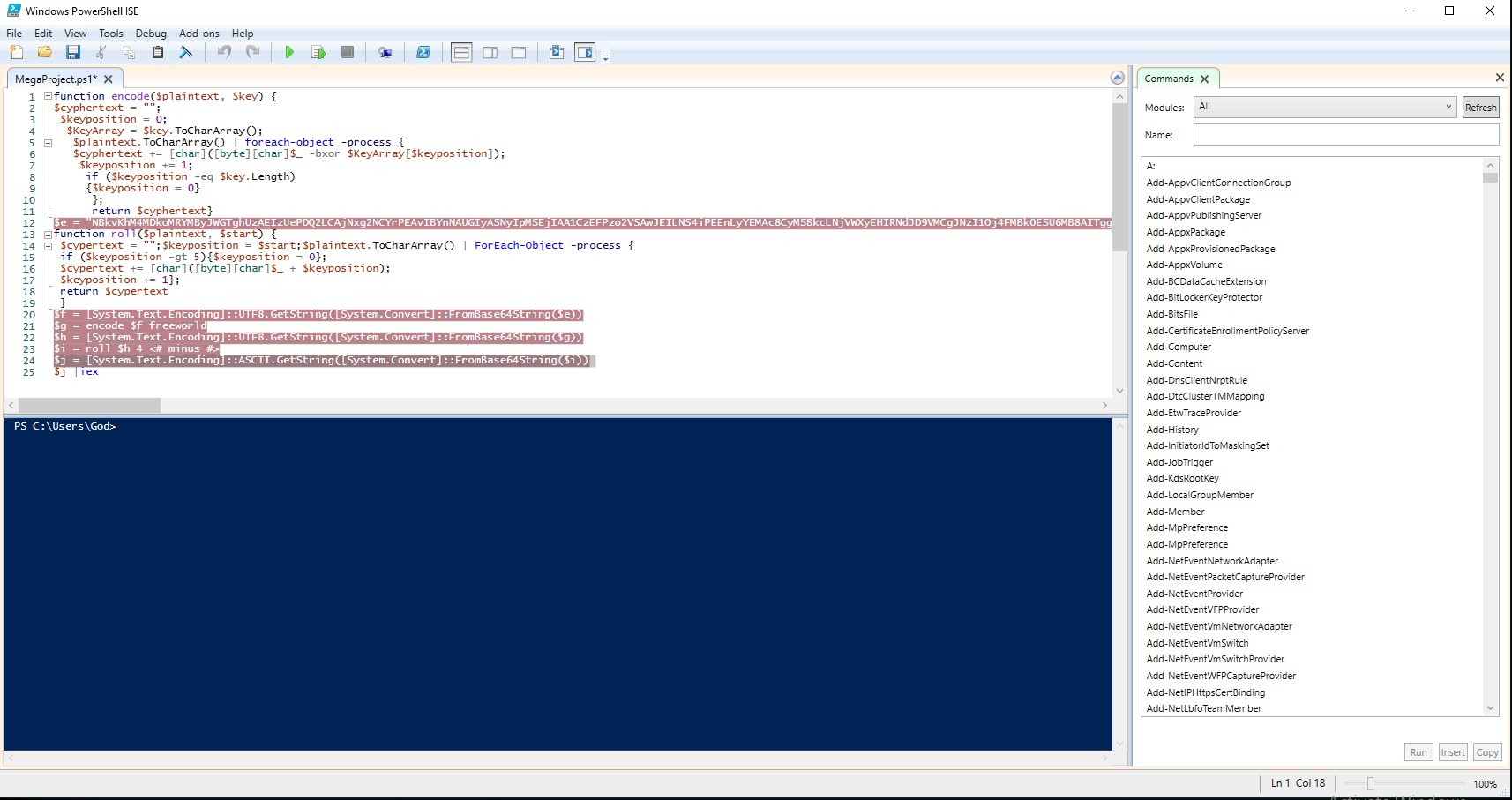


Figure 28: Setting breakpoint for each variable

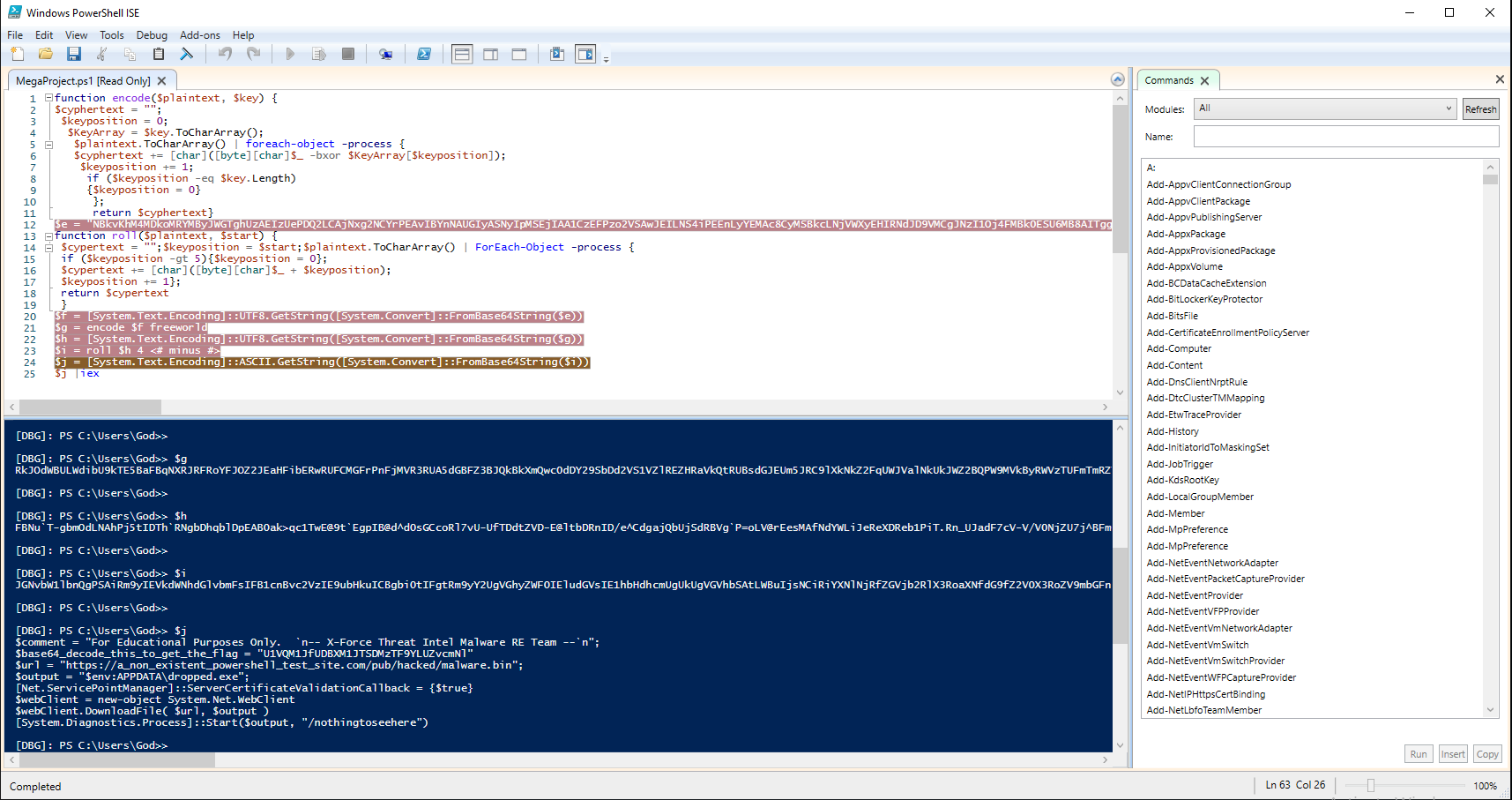


Figure 29: The content of each variable when run step by step

When stepping thought each variable, I check each of them to see the value and at the end, or the last variable when the all the function finish, I can see the code in plan text.

* 1. Analyzing JavaScript malware

Creating malware or malicious script with JavaScript is one of the most dangerous threat in the internet. By creating link or past script in to the html web can cause serious harm like spyware, ransomware, stealing personal data, etc (success.trendmicro.com. (n.d.). *DCX*). I will use Visual Studio Code to debug a malicious JavaScript name” js.html”.

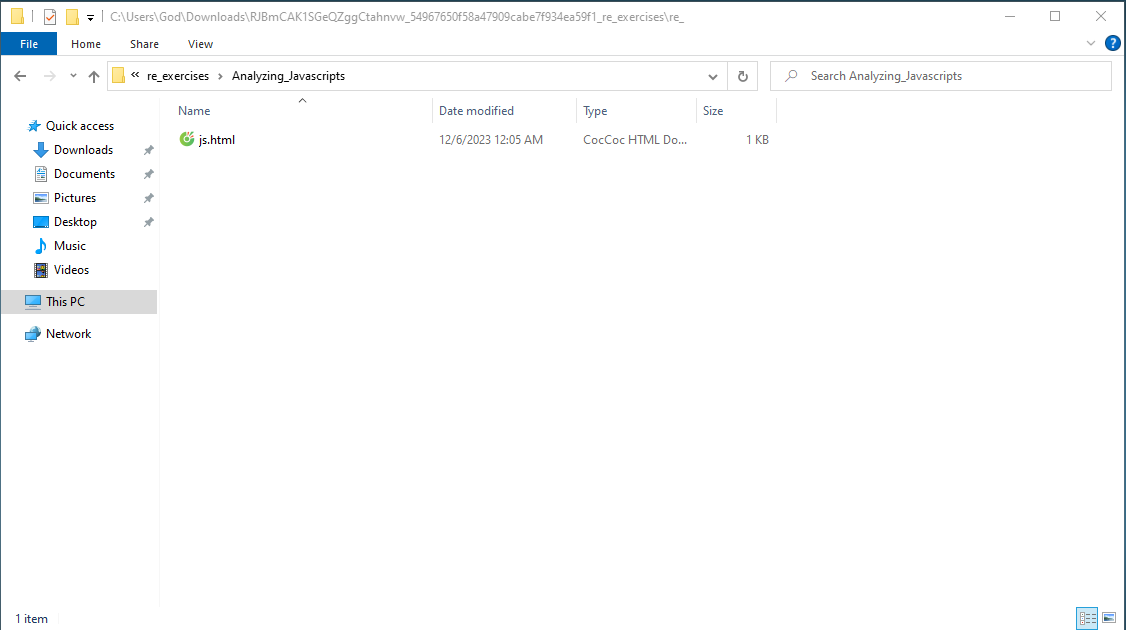


Figure 30: js.html file

Run the file in Visual Studio Code then set the breakpoint in the eval function and document if it has any by clicking next to the line number.

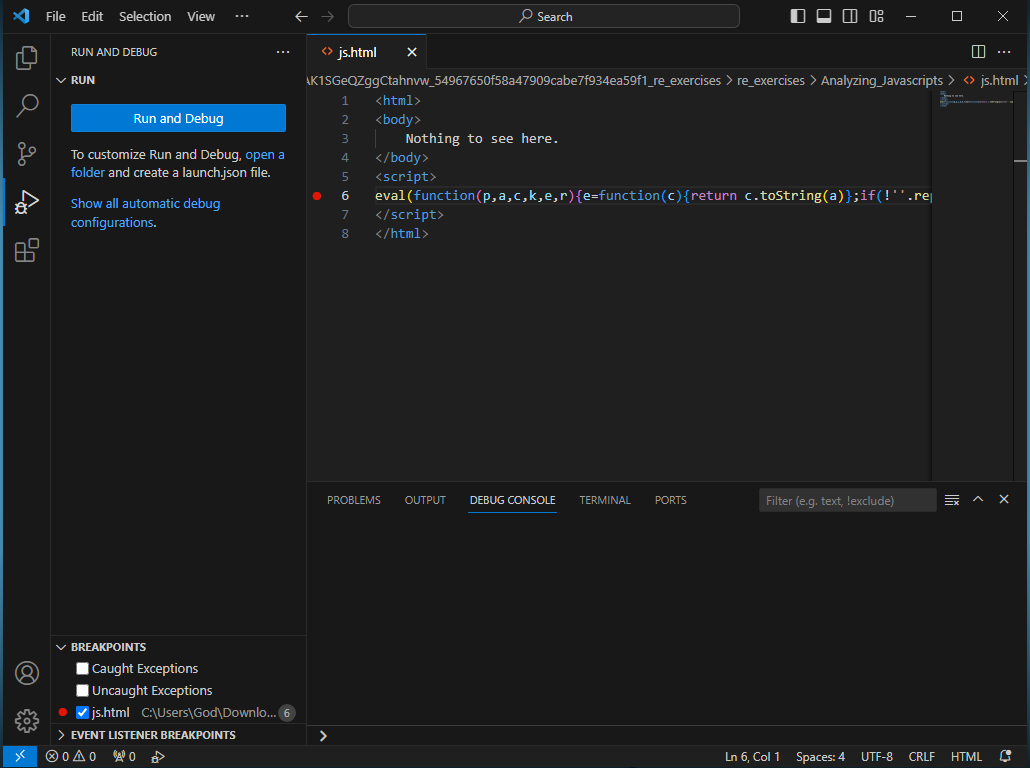


Figure 31: Setting breakpoint in Visual Studio Code

After setting the breakpoint, run and debug the code, then step into the function and step out of it to get to the file.

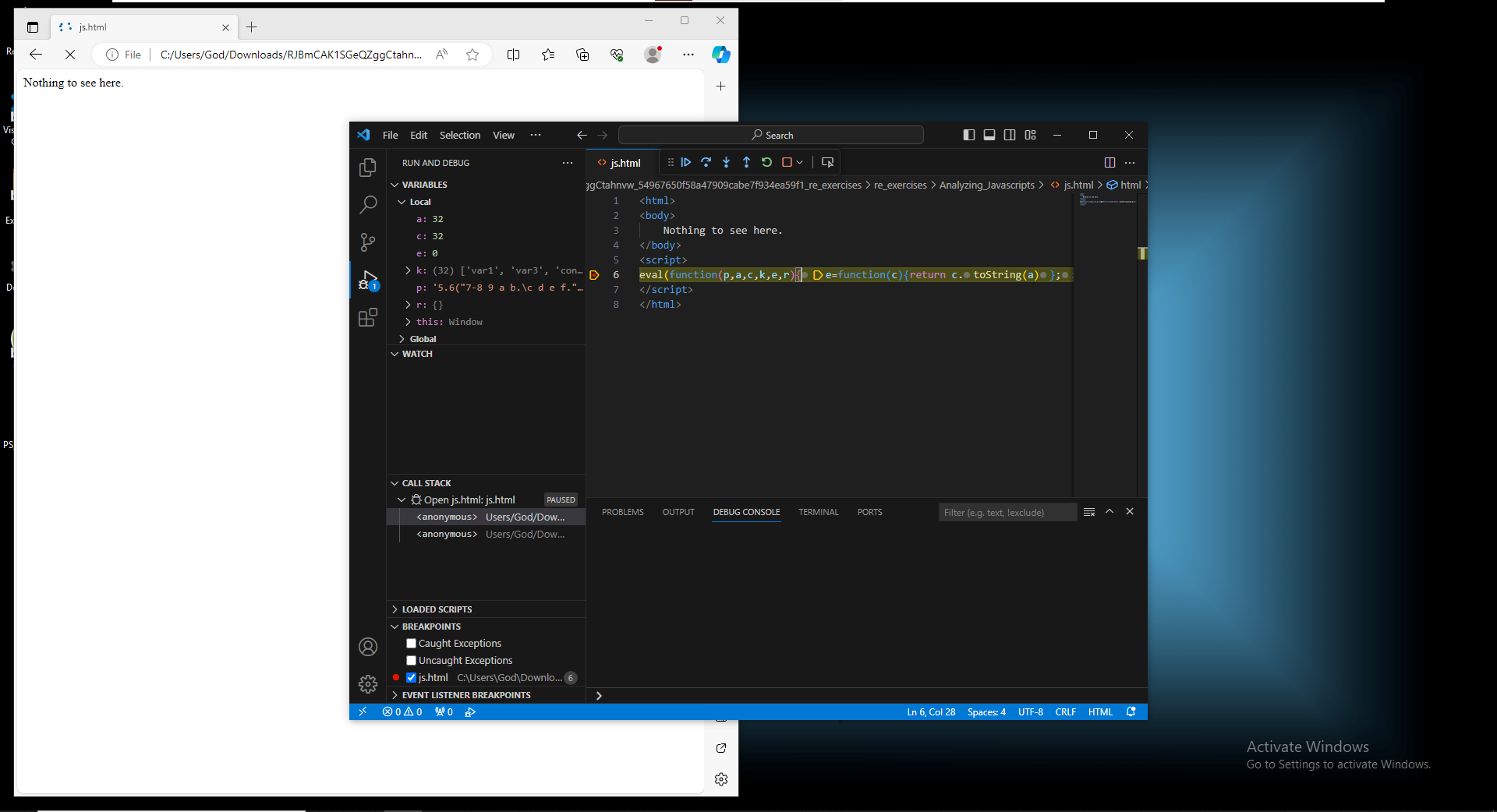


Figure 32: Stepping into the code

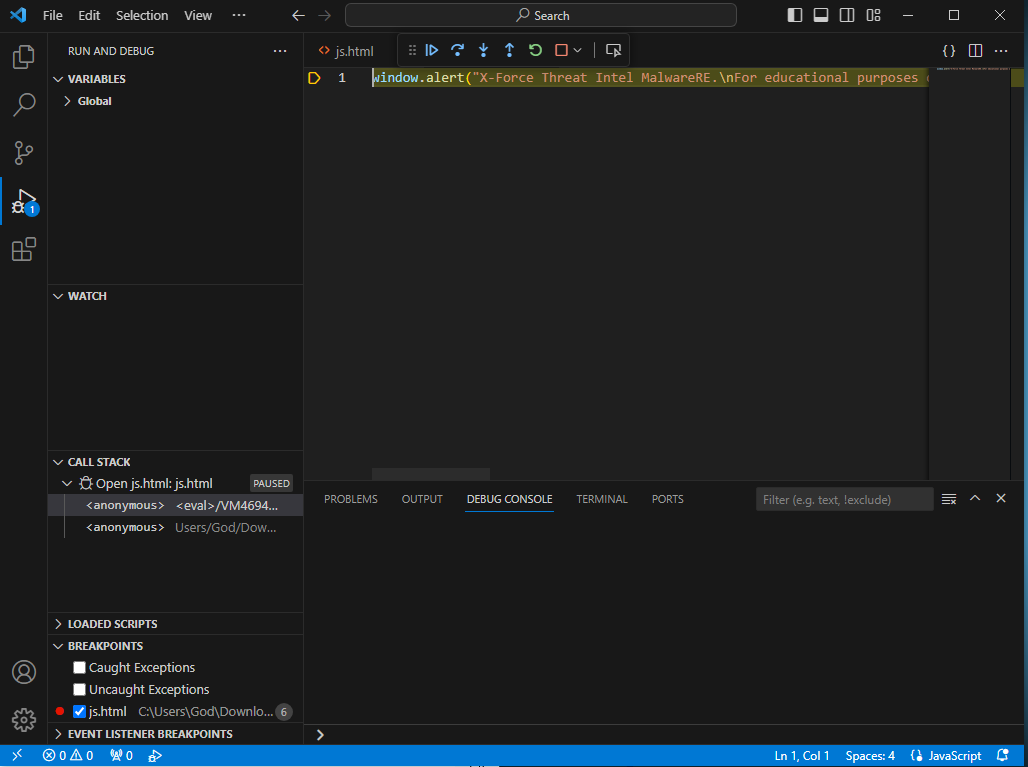


Figure 33: The function after setting into the code

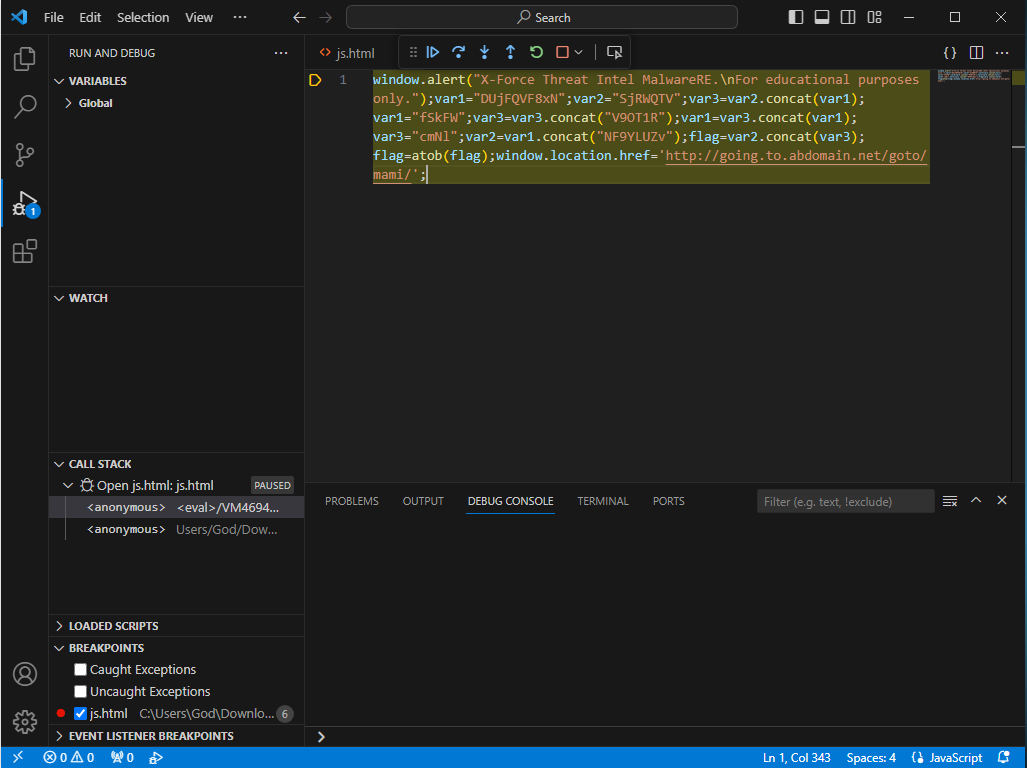


Figure 34: Simplified the code with word warp

We can use word warp for better view, by looking at the code, we can see that it leads user to another web call

” http://going.to.abdomain.net/goto/mami/”

* 1. Analyzing Macro-Enabled Documents

Macro is a set of commands that automates a software to perform a certain action. By abusing the Visual Basic of Application(VBA) programming in Microsoft Office, hacker crate Marco Malwares to spread other form of malware. This attack trigger when a victim clicks on the document link and “Enable Content”, which will affect the recipient. A typical macro malware will execute a base64 PowerShell code, then a file will be download from %UserProfile% or in %Temp%. To understand how to break the malware down to analyze, I will demonstrate with a file name “re\_test\_dlx.doc” with olevba and VBA developer window.

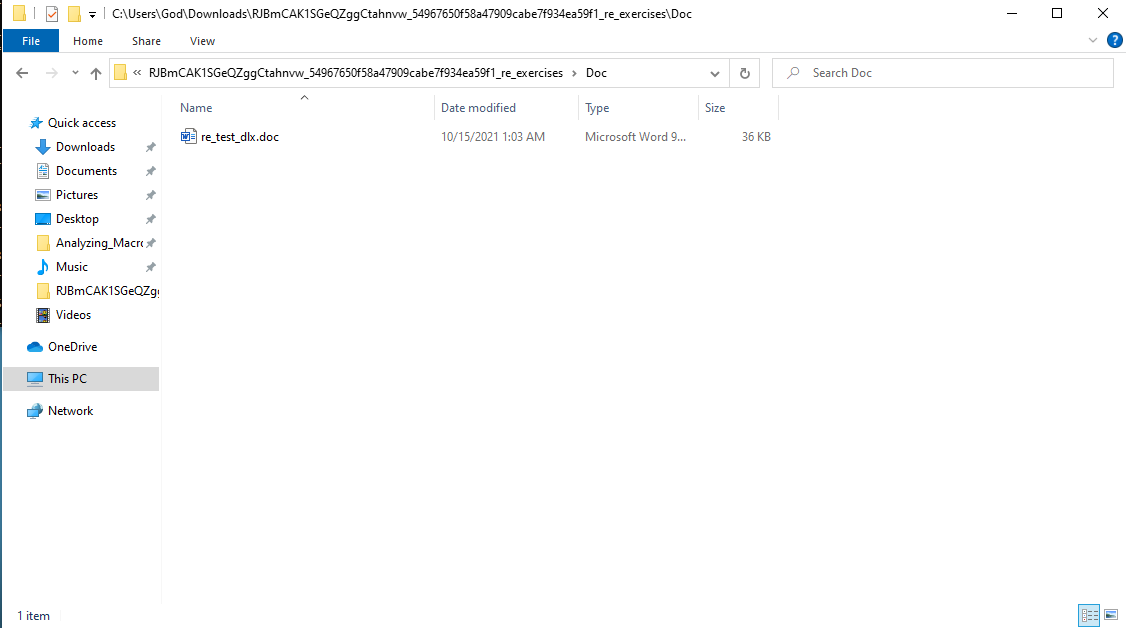


Figure 35: re\_test\_dlx.doc file

First let run olevba and make a copy of the result in “result.txt” and check it out.

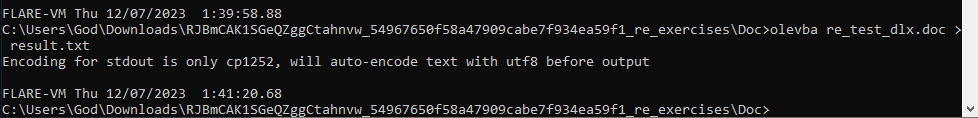


Figure 36: copying the result of olevba the file into result.txt file command



Figure 37: the table of function of the file

It seems like there are 2 AutoExecute functions name “autoopen” and “Document\_Open”. It seems like “autoopen” lead to “Document\_Open”, which execute some function.

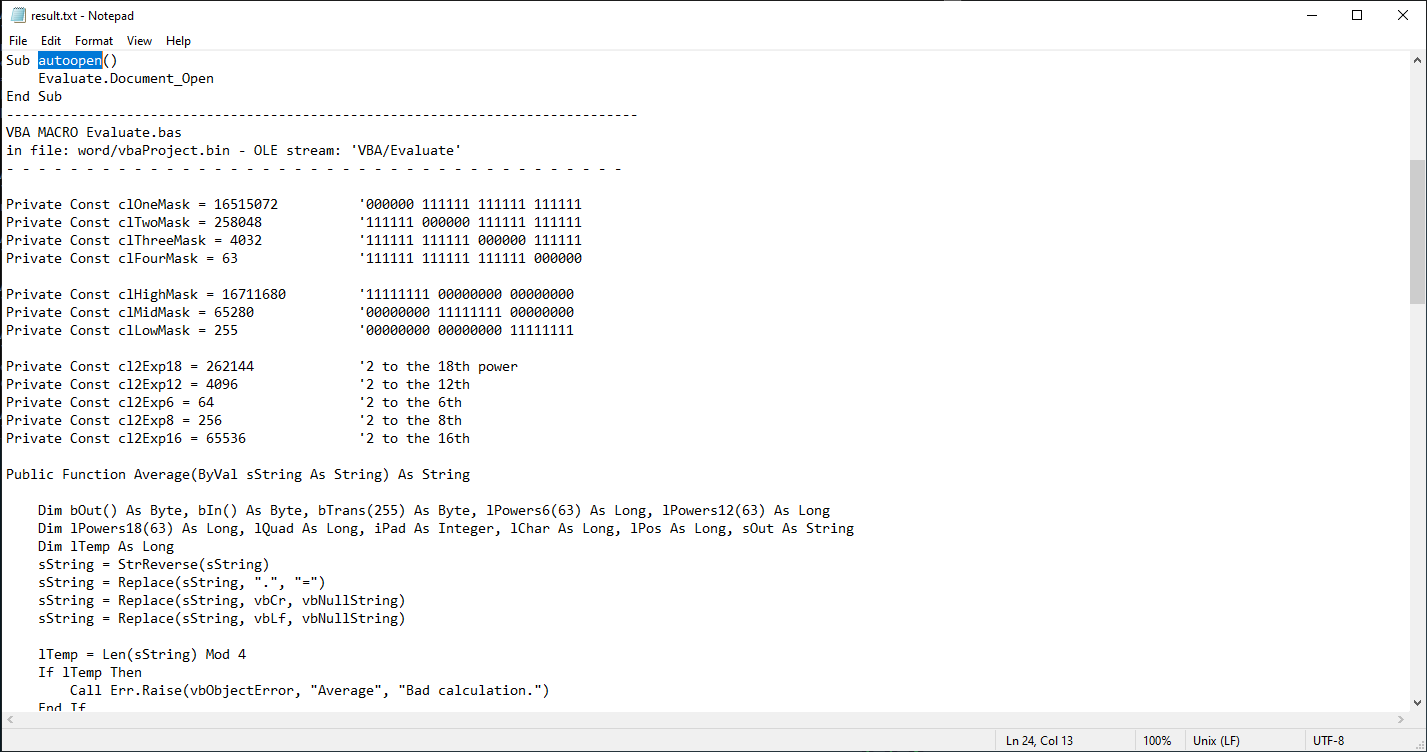


Figure 38: The autoopen function



Figure 39: The Document\_Open Function

Now let open the file in word, then enter the VBA developer window that is built into Word, set a break point at the start of the function with the command “end”



Figure 40: The file in Microsoft Word

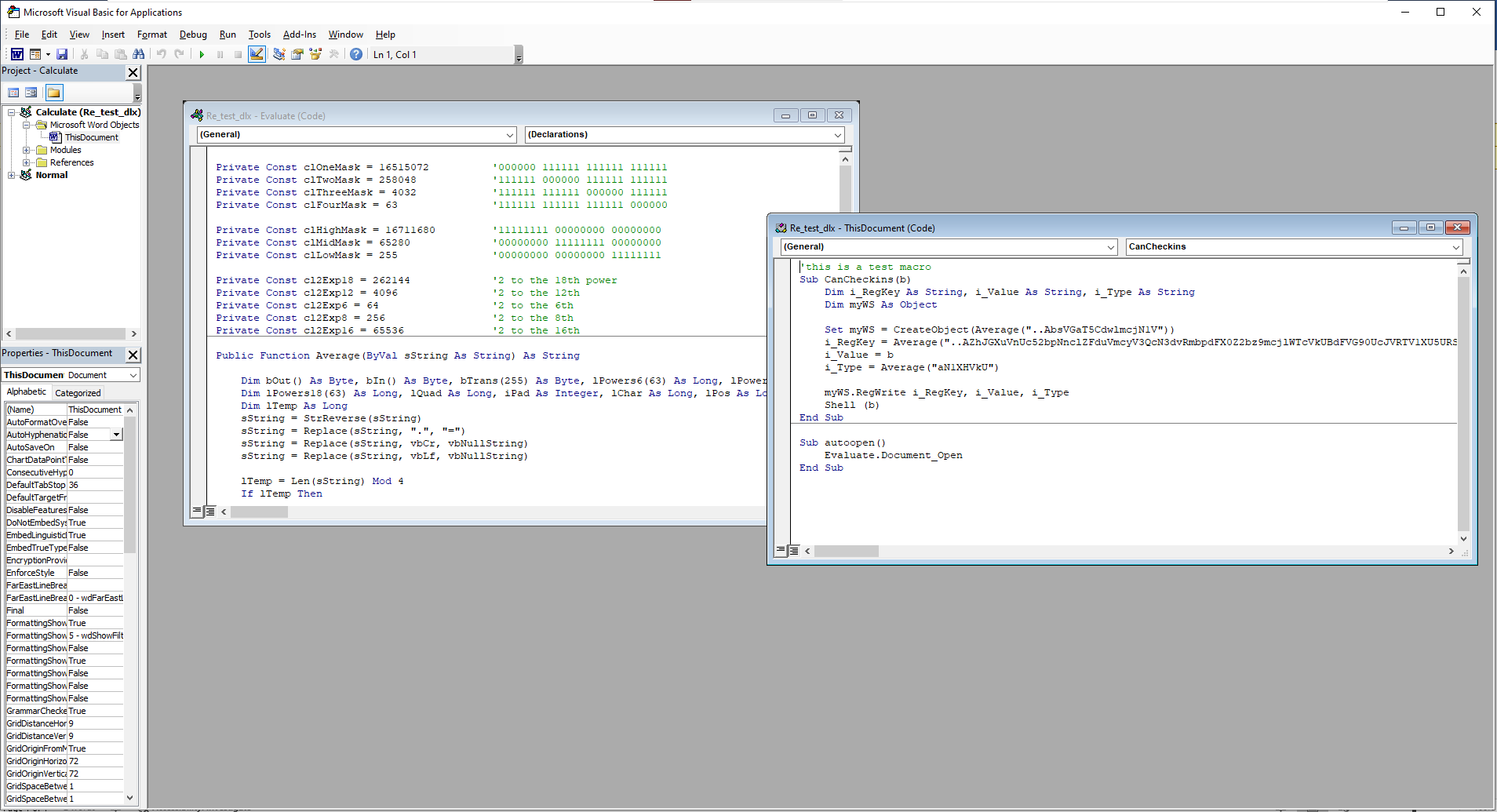


Figure 41: The source code of the file

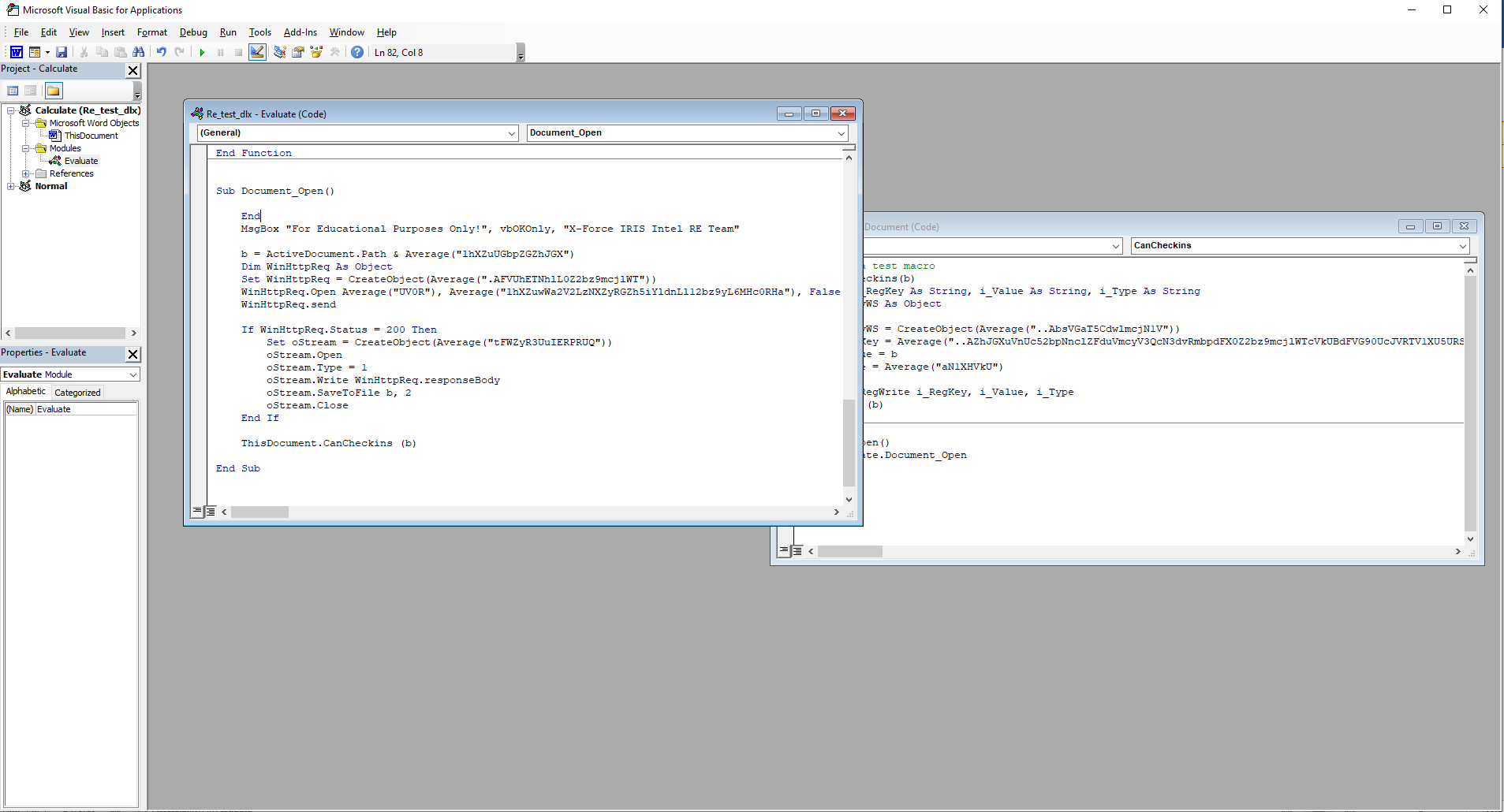


Figure 42: adding “End” code in the source code

After setting the break point the go back and run the code, then delete the “end” command so we can step into the function

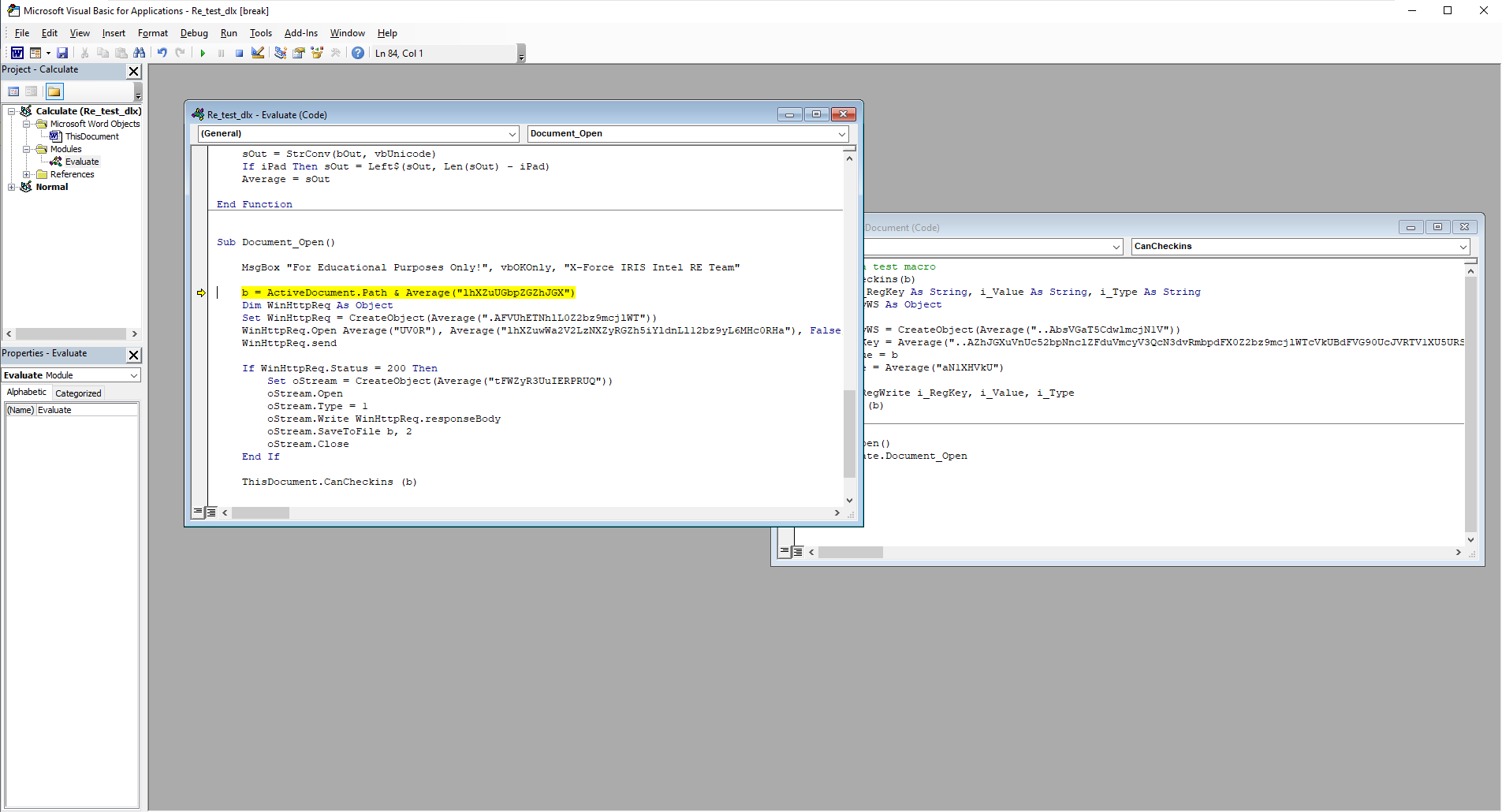


Figure 43: Run the code step by step to view changed

Finally, add all the variable and encrypted data into the watch list and we can se some interesting information.

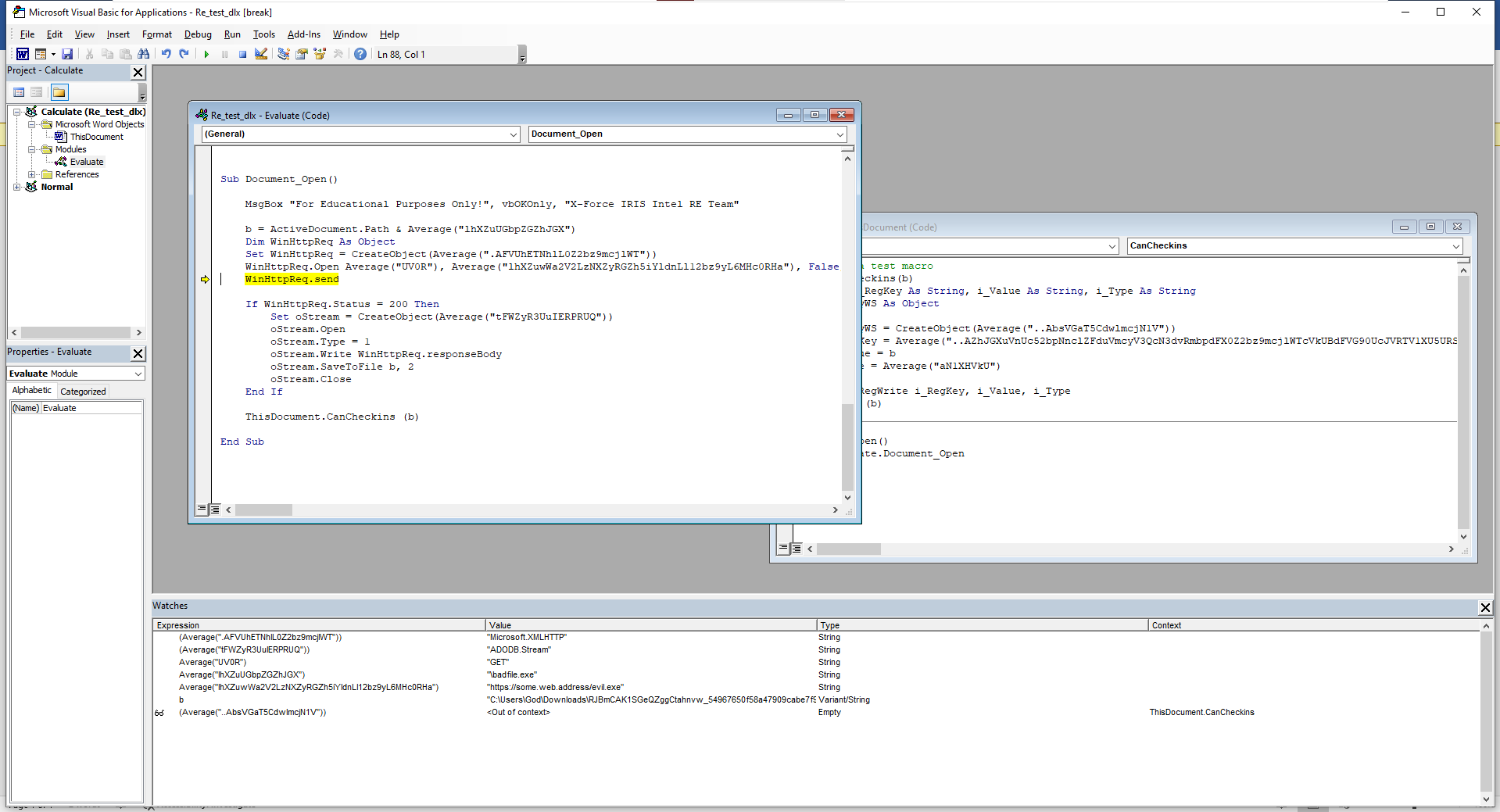


Figure 44: The list of variable and it content after the function

1. **Recommended Framework**

According to NIST Special Publication 800-83 Revision 1 Guide to Malware Incident Prevention and Handling for Desktops and Laptops, this is a company policy on how to prevent malware attack:

A close-up of a document

Description automatically generated

Figure 45: The NIST policy

(Source: <https://nvlpubs.nist.gov/nistpubs/specialpublications/nist.sp.800-83r1.pdf>)

Other than NIST, using Mitigating malware and ransomware attacks framework can help preventing malware from running in your device:

A screenshot of a computer

Description automatically generated

Figure 46: Mitigating malware and ransomware attacks framework

(Source: <https://www.ncsc.gov.uk/guidance/mitigating-malware-and-ransomware-attacks>)

1. **Conclusion**

In conclusion, malware attacks are one of the most dangerous and hard to predict type of attack due to the variety type with the continuous being updated and created by hacker in the future. To counter this risk, malware analyst is composure to help create a solution for preventing the malware by updating the database and anti-virus program. By applying the frameworks to this, companies can help reduce the risk of being targeted in a malware attack.

Word count: 3048

References:

Cisco. (n.d.). *What Is Malware? - Definition and Examples*. [online] Available at: <https://www.cisco.com/site/us/en/learn/topics/security/what-is-malware.html>.

Baker, K. (2021). *11 Types of Malware + Examples That You Should Know*. [online] crowdstrike.com. Available at: <https://www.crowdstrike.com/cybersecurity-101/malware/types-of-malware/>.

Fortinet. (n.d.). *What is Malware Analysis? Types and Stages of Malware Analysis*. [online] Available at: <https://www.fortinet.com/resources/cyberglossary/malware-analysis#:~:text=Malware%20analysis%20is%20the%20study>.

Consumer Information. (2021). *How To Recognize, Remove, and Avoid Malware*. [online] Available at: <https://consumer.ftc.gov/articles/how-recognize-remove-avoid-malware>.

zeltser.com. (n.d.). *Mastering 4 Stages of Malware Analysis*. [online] Available at: <https://zeltser.com/mastering-4-stages-of-malware-analysis/>.

www.varonis.com. (n.d.). *What is Fileless Malware? PowerShell Exploited*. [online] Available at: <https://www.varonis.com/blog/fileless-malware>.

Souppaya, M. and Scarfone, K. (2013). Guide to Malware Incident Prevention and Handling for Desktops and Laptops. *Guide to Malware Incident Prevention and Handling for Desktops and Laptops*. [online] doi:https://doi.org/10.6028/nist.sp.800-83r1.

National Cyber Security Centre (2020). *Mitigating malware and ransomware attacks*. [online] www.ncsc.gov.uk. Available at: <https://www.ncsc.gov.uk/guidance/mitigating-malware-and-ransomware-attacks>.

success.trendmicro.com. (n.d.). *DCX*. [online] Available at: <https://success.trendmicro.com/dcx/s/solution/000279049?language=en_US&sfdcIFrameOrigin=null>.

‌

‌

‌

‌

‌

‌

‌

‌