**Assignment 2 (The Unknown Malware)**

Done by:

Mo’ath Azaizeh 20190287

Abdelrahman Meri 20180145

Abdallah Atieh 20180384

* **Introduction**

In order to gain a deeper understanding of the malware and its potential consequences, we conducted a thorough analysis of the malware. This analysis involved employing both static and dynamic techniques, using some tools like X32 debugger, strings, Wireshark, and CFF.

During static analysis, we closely examined the malware's code or binary structure, as well as relevant metadata such as file properties and network traffic data. This helped us gain insights into the inner workings and characteristics of the malware.

For dynamic analysis, we executed the malware within a Windows 10 virtual machine environment. By observing its behavior in this controlled setting, we were able to study its actions and interactions in real time. This allowed us to uncover any malicious activities or potentially harmful effects.

* **Analysis**

1. Static analysis

We start our analysis by opening the properties of the file and we can see that the extenuation of the file is .EXE but it’s shown as a JPG file and which indicates that it is an executable file as shown in figure 1. This information suggests that the file is intended to be run and executed on a computer system. Understanding this characteristic helps us establish the nature of the file and proceed with our analysis accordingly.

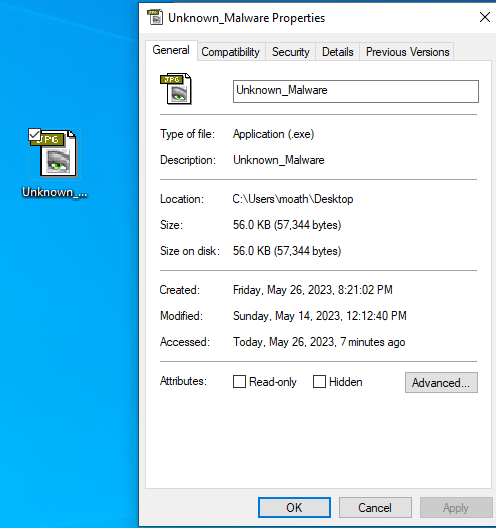


Figure 1

After examining the file properties, we proceeded to utilize the CFF Explorer Suite tool to extract additional details about the malware, after using the CFF Explorer Suite tool, we were able to retrieve various information about the malware, including its hash value, and the programing language of the malware which serves as a unique identifier for the file. This hash can be essential for tracking and referencing the specific instance of the malware during analysis and investigations as shown in Figure 2.

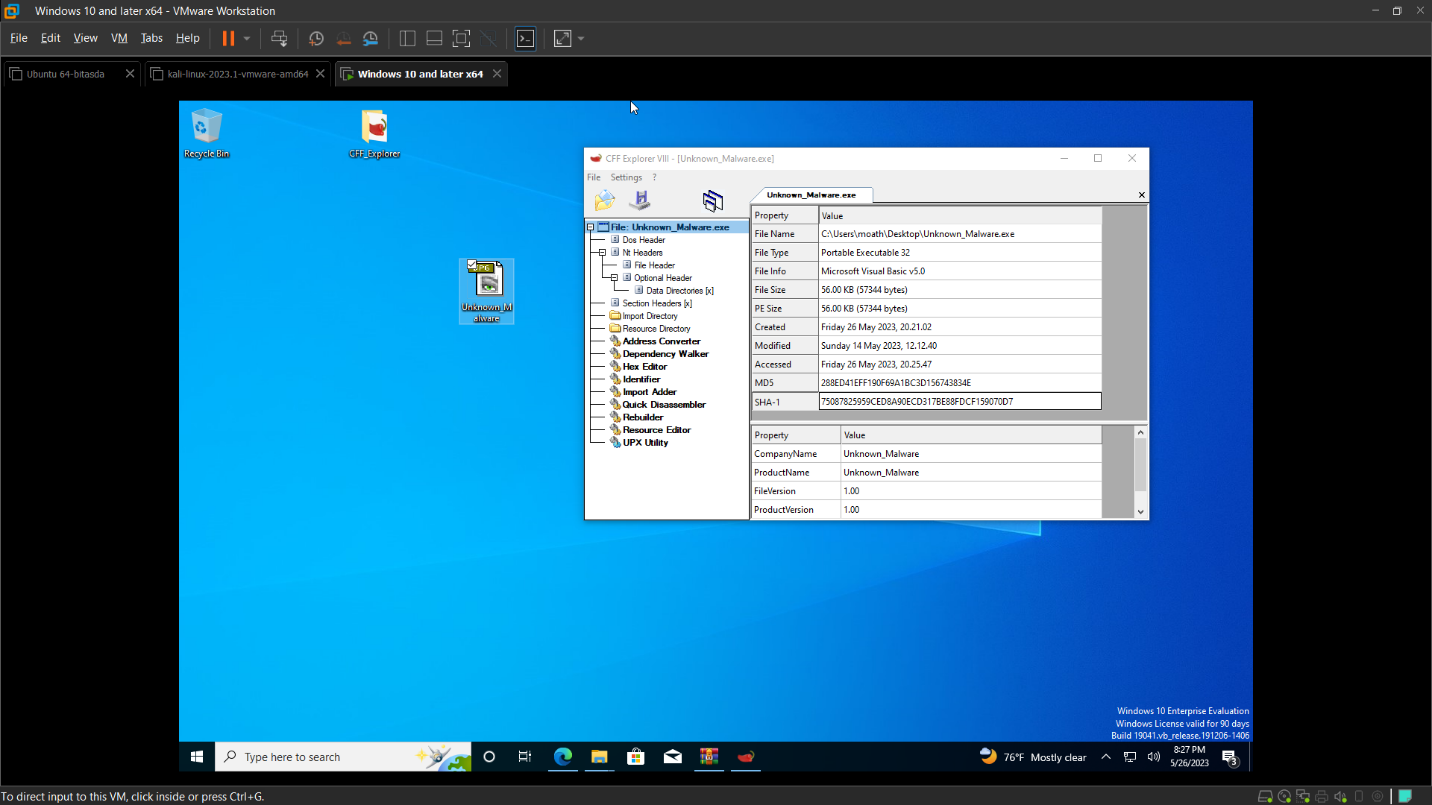
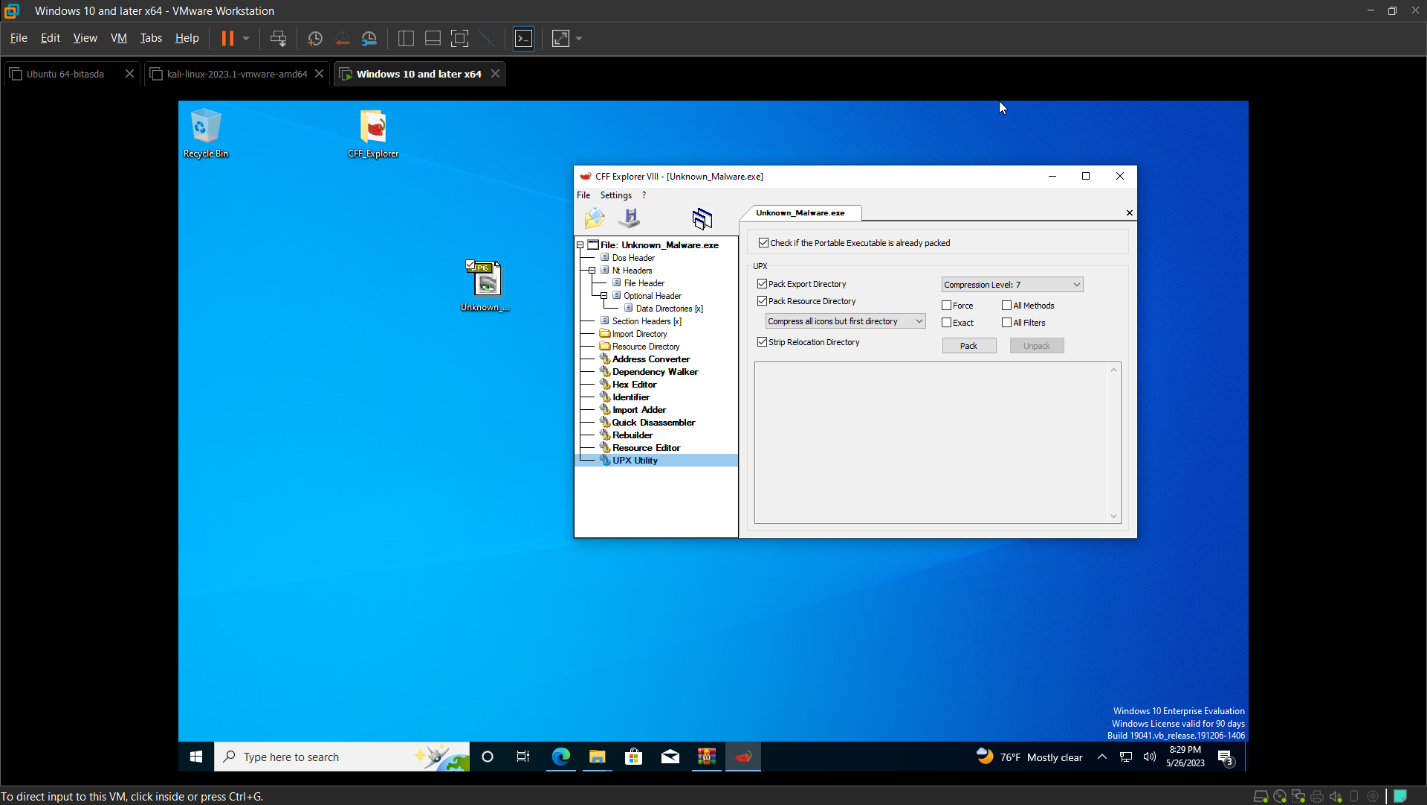


Figure 2

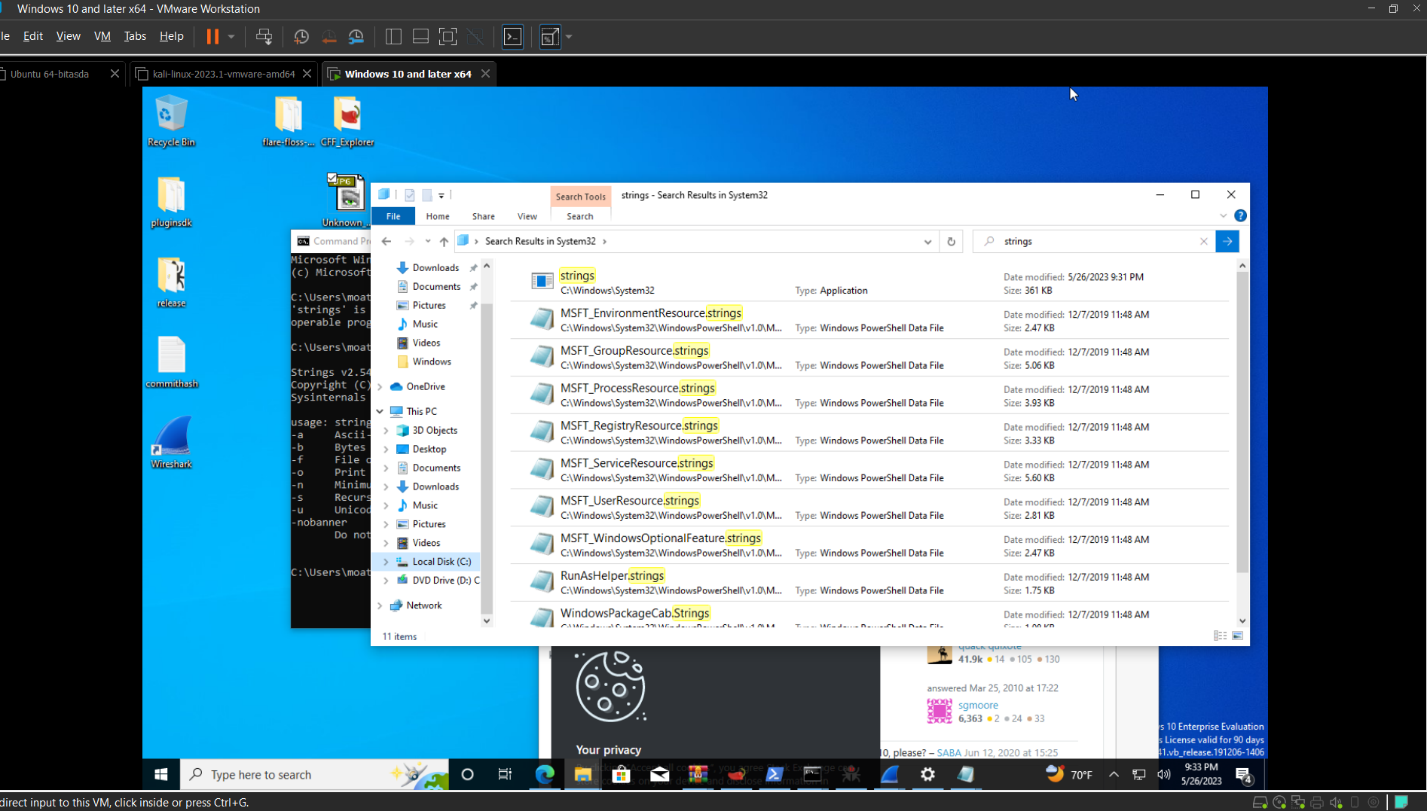
In figure3, we checked the UPX.

Figure 3



Subsequently, we executed the "strings" tool as part of our analysis process. To ensure a proper environment for the tool, we first moved it to the "system32" directory as shown in figure 4. By doing so, we could utilize the tool effectively and extract ASCII strings from the malware file.

Figure 4



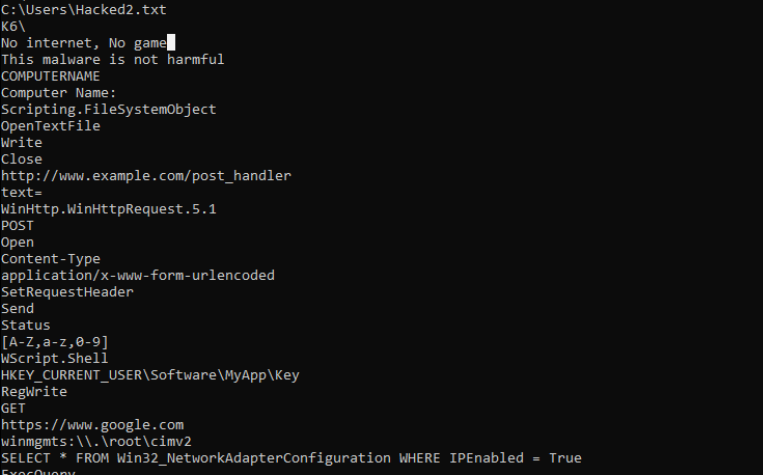
The "strings" tool is specifically designed to identify and extract readable text strings from binary files. By executing this tool, we aimed to uncover any meaningful strings embedded within the malware that could provide insights into its functionality, intentions, or potential indicators of compromise as shown in Figures 5 to 9.

Figure 5



In Figure 5 we found the path of the encrypted data that had been created by the malware.

Figure 6



In Figure 6 we found the malware uses the [www.google.com](http://www.google.com) URL to test connectivity with the internet, and if there isn’t any internet available it would give us “No internet, No game” as shown in Figure 7.

Figure 7

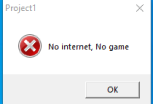
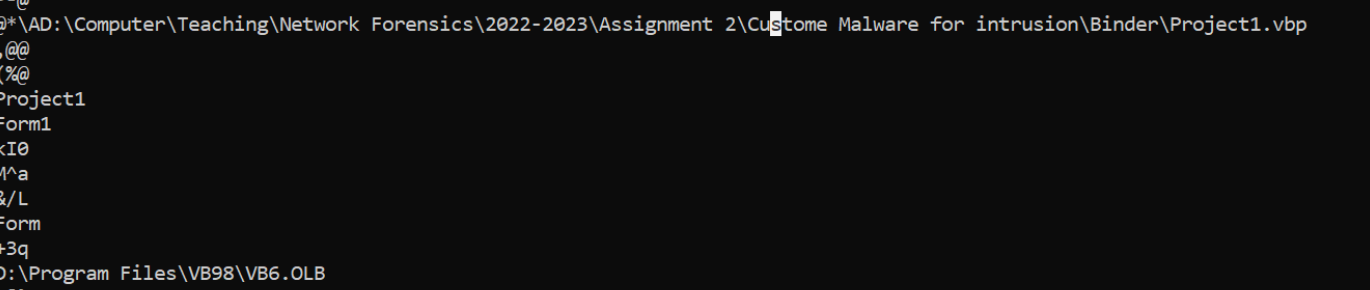
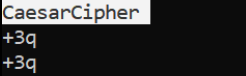


Figure 8



In Figure 8 we found the path that the malware was on your pc ;)

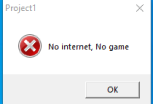
Figure 9



In Figure 9 the malware uses the Caesar cipher encryption algorithm.

1. Dynamic Analysis

Firstly, we used the Wireshark tool to analyze the behavior of the malware while the machine is connected to the internet and opening (running) the malware as shown in Figures 10 and 11.

We start by run the unknown malware without internet and as we know it gave us this pop up   


But we noticed that the program creates updator.exe even though there is no internet connectivity



So we knew that the malware needs internet for other purposes

Figure 10

A screenshot of a computer

Description automatically generated

In Figure 10 we see that the malware using the URL google.com to test connectivity then we found the post method on <http://www.example.com/post_handler> as shown in figure 11.

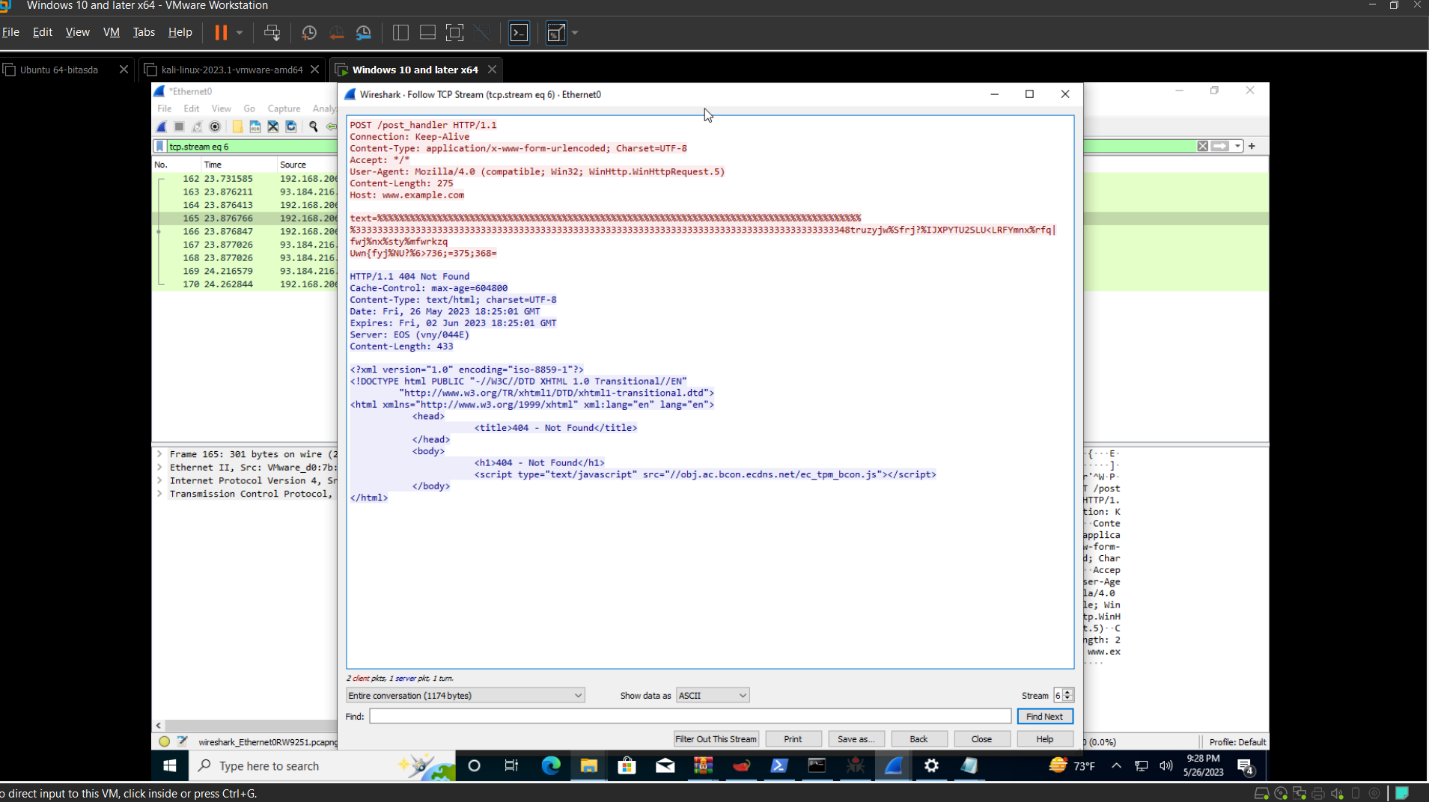
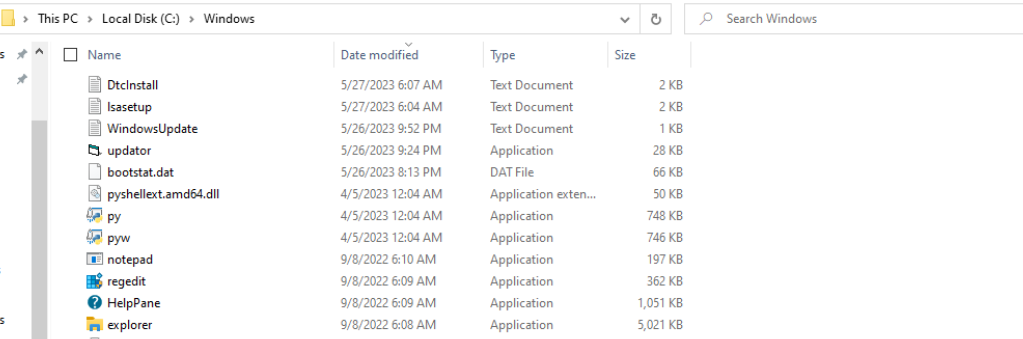


Figure 11

Here the hacked2.txt was posted to post handler.

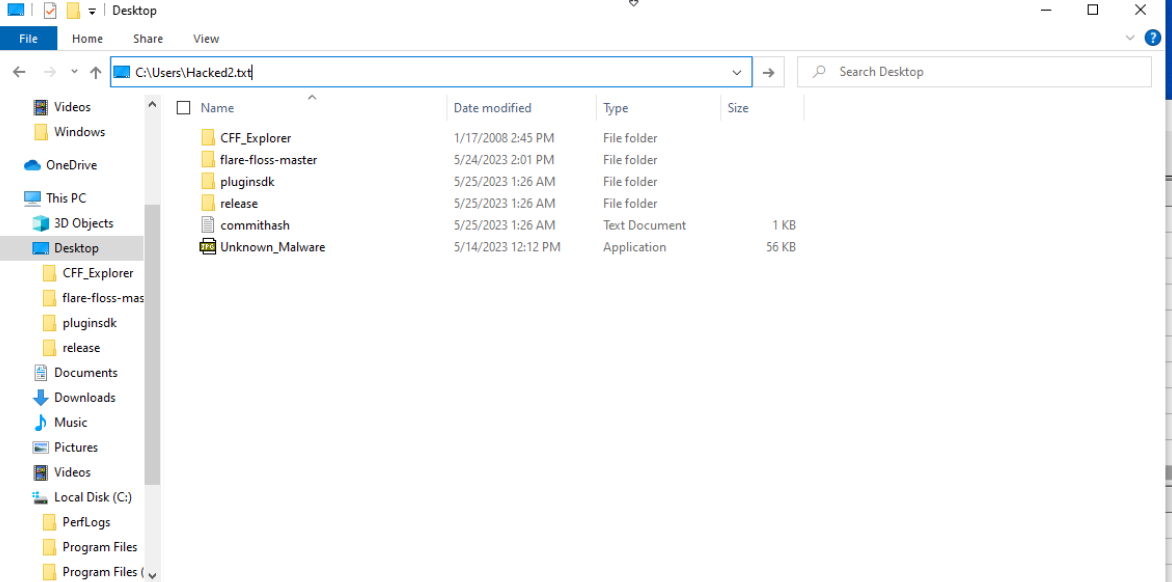
Secondly, in the flowing figures from 12 to 15, we see that the updator.exe is being downloaded and the hacked2 file was sent and encrypted.

Figure 12



In figure 12 we see that the updator.exe the file was successfully created on our machine.

Figure 13



In Figure 13 we searched for the path that we found in the strings “c:\Users\Hacked2.txt”.

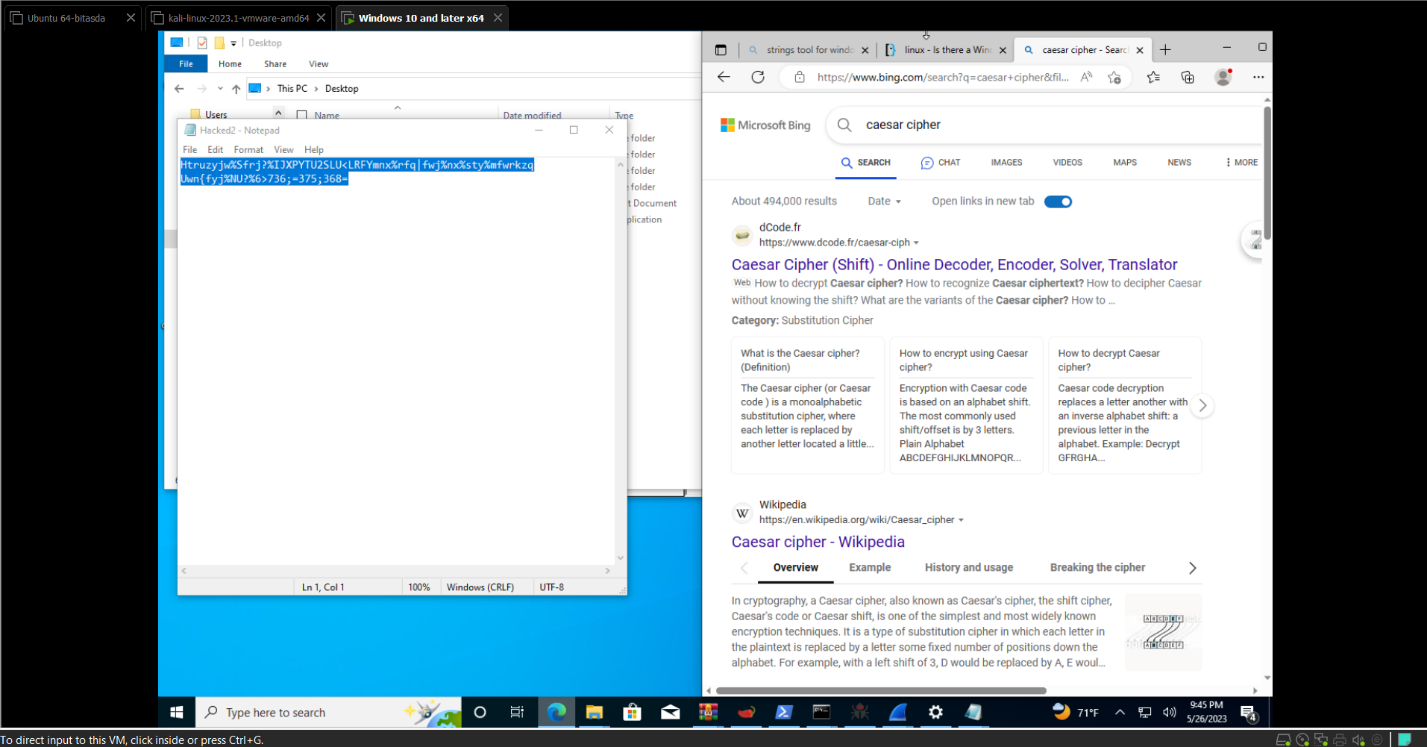
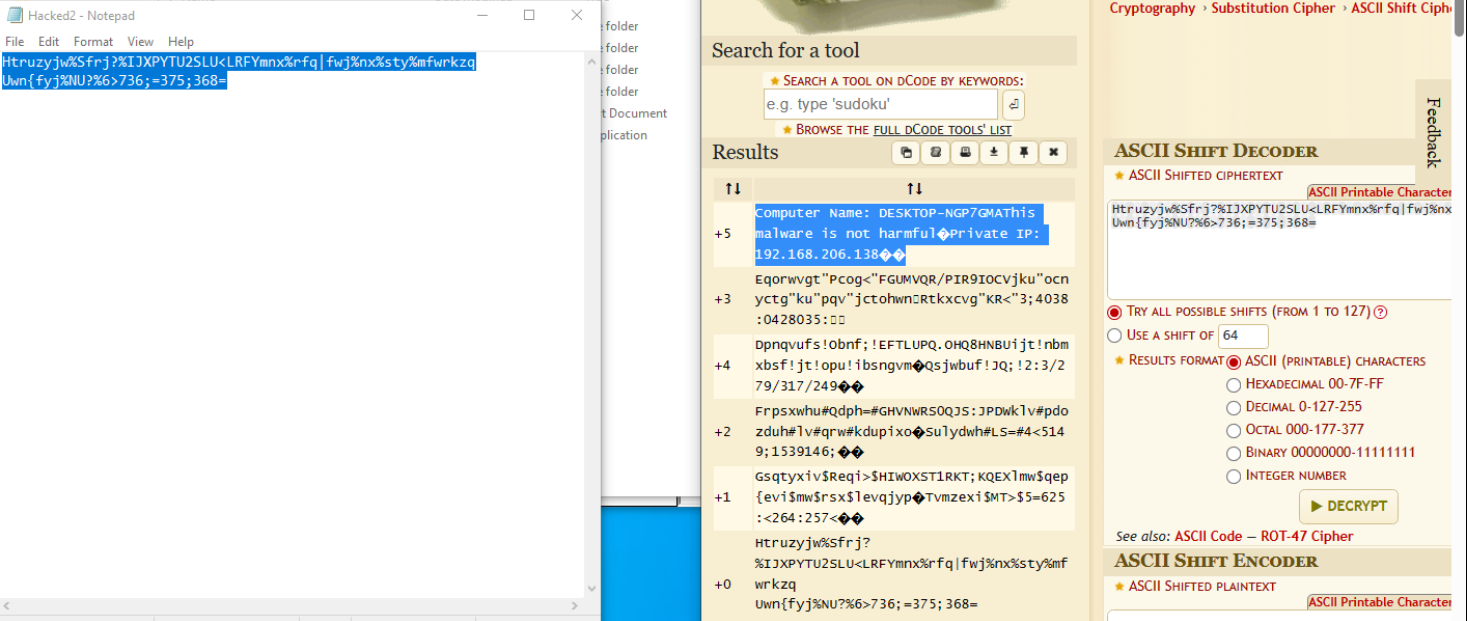


Figure 14

In figure 14 we found the encrypted data in the file after opening it.

Figure 15



After decrypting the data using Caesar cipher decryption we found our private IP address and massage saying that the malware is not harmful as shown in figure 15.

* **Yara Rules**

rule unknown\_malware

{

meta:

description = "Detects a specific unknown malware that steals device information and exfiltrates it"

author = "Abdallah, Abdulrahman, Moath"

malware\_author = "Dr. Haitham Al Ani"

strings:

$mz = { 4D 5A }

$networkEndpoint = "http://www.example.com/post\_handler" ascii wide

$httpMethod = "POST" ascii wide

$dataParameter = "text=" ascii wide

$encryptionAlgorithm = "CaesarCipher"

$executableName = "updator.exe" ascii wide

$textFileName = "Hacked.txt" ascii wide

condition:

($encryptionAlgorithm and $executableName and $textFileName) or ($dataParameter and $httpMethod and $networkEndpoint) and $mz

}

* **Conclusion**

The unknown\_malware is an executable file that creates a file named "updator.exe". This file is responsible for collecting information from the victim's device and encrypting it using the Caesar cipher algorithm. The encrypted data is then stored in a file called "hacked2.txt". The malware further attempts to exfiltrate this data by sending it to the domain "example.com". It is worth noting that the malware employs a deceptive tactic by displaying a pop-up message named "PSUT.dll is messing". However, this message is merely a distraction as the primary concern of the malware lies in establishing network connectivity.