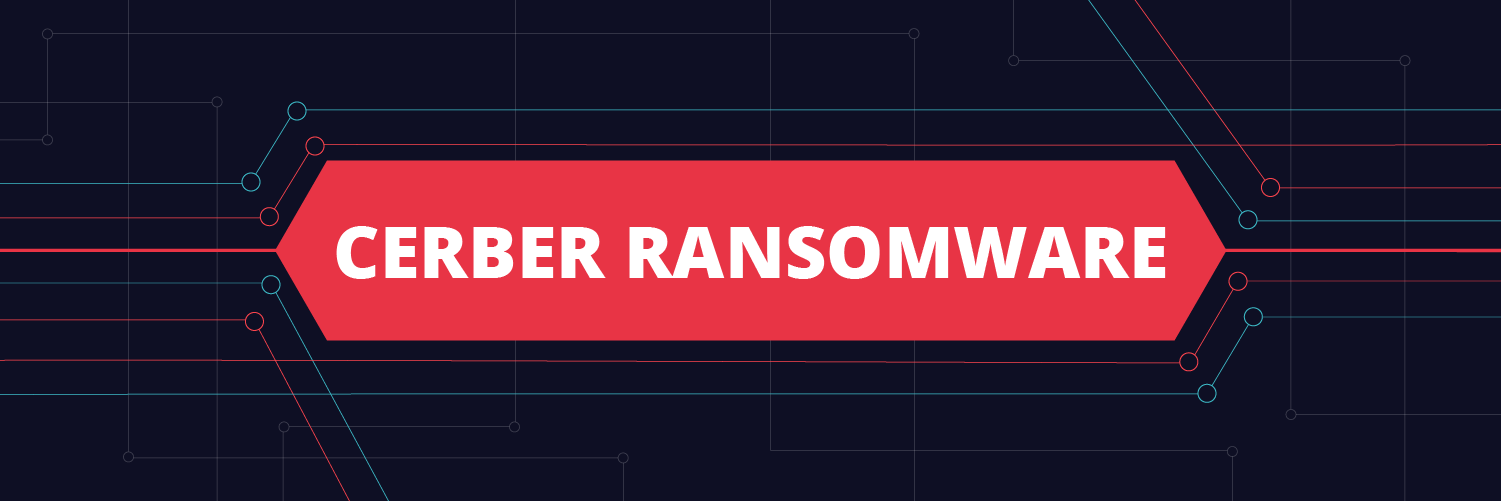
**Mannage M.S.D.S. IT17069564**

**Cerber Ransomware Analysis**

**Introduction**



Ransomware is one of the dangerous malware types where it involves financial gain that usually encrypts the user applications, files, or systems that will be unable to access until a ransom is paid.

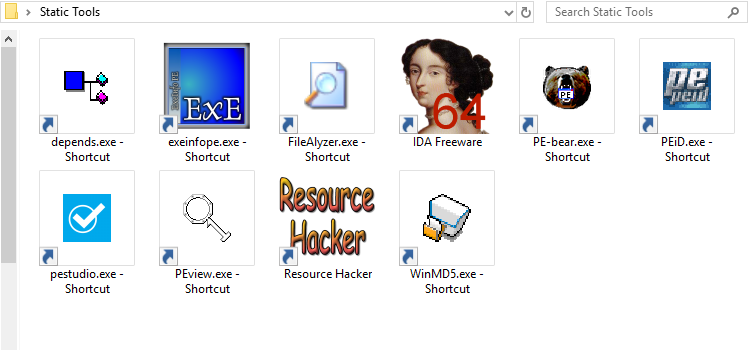
Cerber Ransomware has first seen in 2015 and 2016 Cerber ransomware campaign detected where ransomware was distributed with attached malicious Microsoft Word documents. The ransomware is named after “Cerberus”, a giant multi-headed dog who guards the gates to the underworld. Cerber ransomware became popular as “the talking ransomware” due to its ability to read ransom note loud to the victim.

Cerber ransomware is accessible on the side as Ransomware-as-a-Service (RaaS), which allows cybercriminals who don’t have much technical knowledge to obtain the ransomware from the original creators for monetary gain.

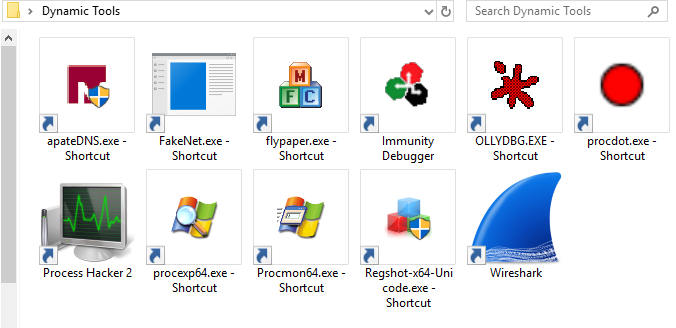
The Cerber ransomware currently only targets specific countries, mainly the U.S. which reports up to 50% of the Cerber infections. Other victims of the malware are Taiwan, Japan, Australia, Spain, Brazil, Germany, Portugal, and Malaysia. Investigations have led to the belief that Cerber has originated in Russia, as the malware will not infect Russian users or its neighboring countries. The malware also detects the keyboard language and does not execute if the country of origin is the same as the country of the malware author.

**Lab Setup**

Malware analysis lab is setup using Windows 10 Virtual machine running using VirtualBox. The VM can be used in any other hypervisor (VM ware etc.). After the Windows 10 machine is installed, the malware analysis tools are downloaded and installed as shown below.



The above figure shows the tools that will be used for **Static Analysis**. They are Dependency Walker, EXEinfo PE, FileAlyzer, IDA, PE-bear, PEid, PE Studio, PE view, Resource Hacker, and WinMD5.



The figure above shows the **Dynamic Analysis** tools: ApateDNS, FakeNet, FlyPaper, Immunity Debugger, OllyDBG, ProcDot, Process Hacker, Process Explorer, Process Monitor, Regshot, and Wireshark.

**Obtaining the Malware**

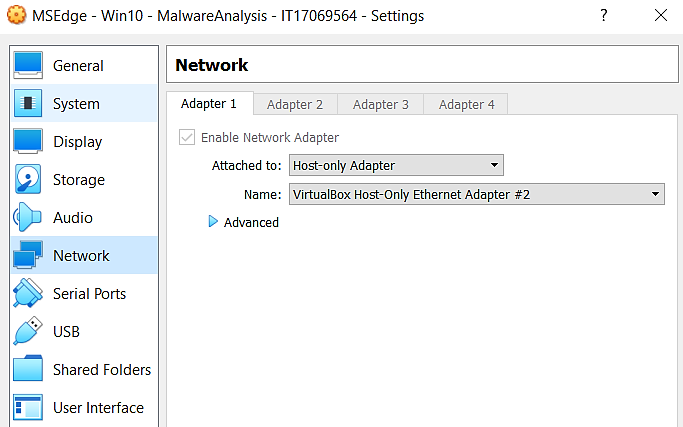
Next, we must obtain the malware file. We can easily download archived versions of popular malware through the GitHub repository called “TheZoo” ( [www.github.com/ytisf/theZoo](http://www.github.com/ytisf/theZoo) ), which is a collection of malware all in one location for ease of use for malware analysts. All the malware available here are archived and protected so that until they are extracted, they cannot infect any host.

For this analysis, the Cerber Ransomware can be downloaded through:

<https://github.com/ytisf/theZoo/blob/master/malwares/Binaries/Ransomware.Cerber/Ransomware.Cerber.zip>

The malware can then be extracted using the password “infected”. But before this, it is necessary to take a few precautions to make sure that the malware will not propagate to the host computer and through the network:

* All share folders between the host and VM must be removed.
* A snapshot of the current state of the VM must be taken before running the malware.
* As the figure shows the Network Adapter must be set to Host-Only Adapter. This removed the connection of the VM with the real network.



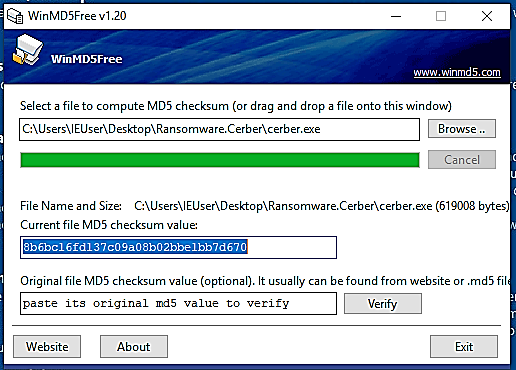
**Static Analysis**

Analyzing malware without executing or running is known as static analysis. The objective of this is to extract information such as metadata, PE headers, type of malware, and what malware can do.

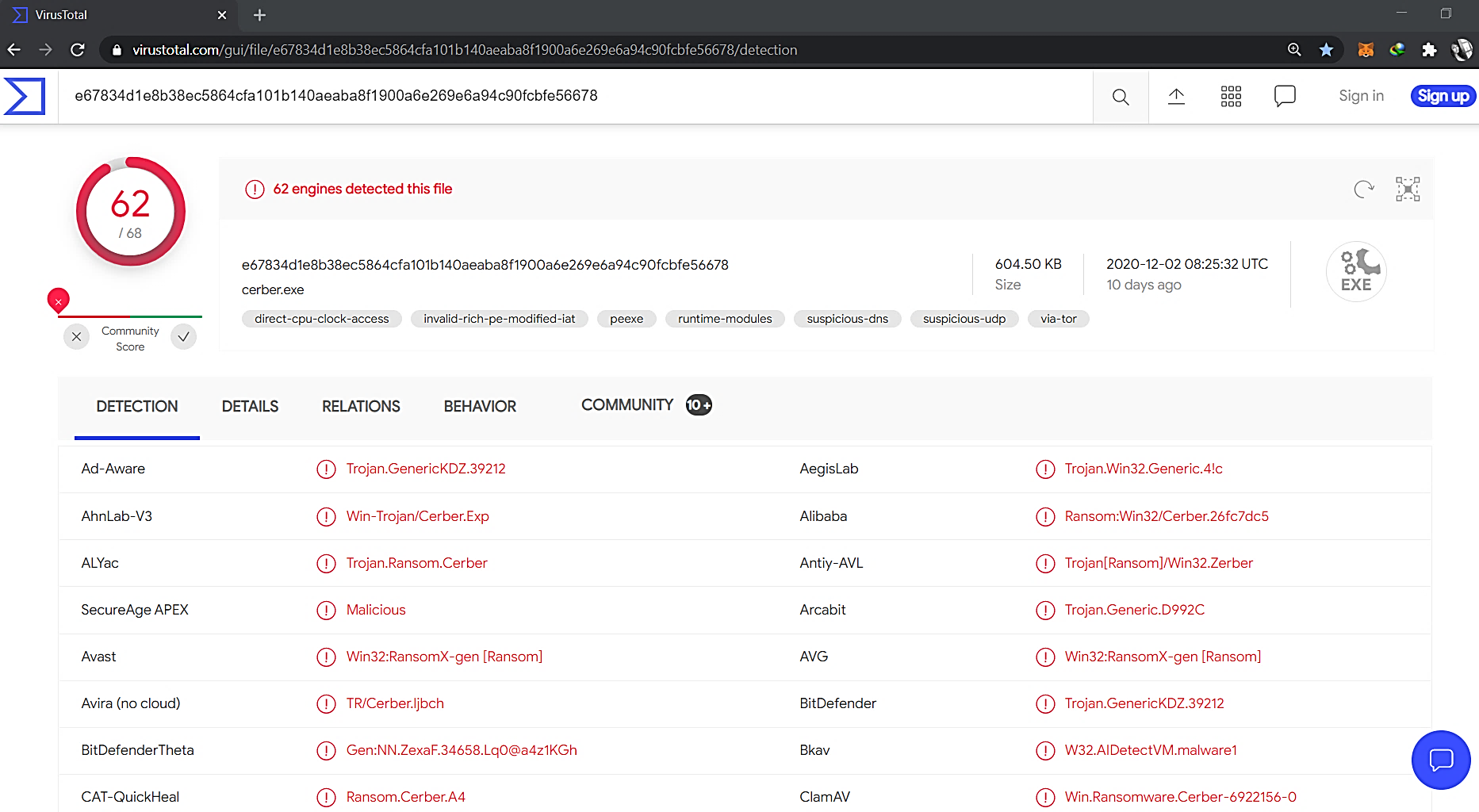
1. **Identifying malware**

The first step of static analysis is to generate a malware hash and analysis it on publicly available tools to obtain more information on it. The main target is to identify whether the sample is already analyzed by antivirus software.

For this purpose, I have used the WinMD5 tool to obtain the MD5 hash value of the malware sample.

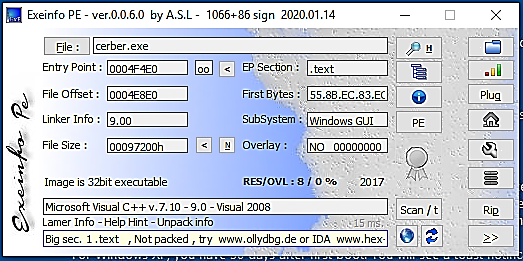


Then scanned the MD5 hash in [www.virustotal.com](http://www.virustotal.com). As shown in the below diagram, Virus Total results indicate the hash file was identified by 62 out of 68 antivirus tools and the hash is Cerber.exe. Also, many malware engines detect this as a trojan. So, we can assume the ransomware sample will show some type of trojan behavior for disguises itself as a legitimate program. Virus Total scan result also relevels much information such as different file names, hashes, and versions, file properties, behaviors, community comments, and signatures.



1. **Packing and Obfuscation**

Packing is used by malware to avoid detection. This is achieved by compressing the executable file. Exeinfo Pe tool can be used to identify whether the malware sample is packed. Also, this tool gives information on how to unpack the file if it is packed. When we open the sample in the tool it shows as malware sample is not packed. This tool also provides other information on malware sample such as what type of compiler sample is detected and header information.

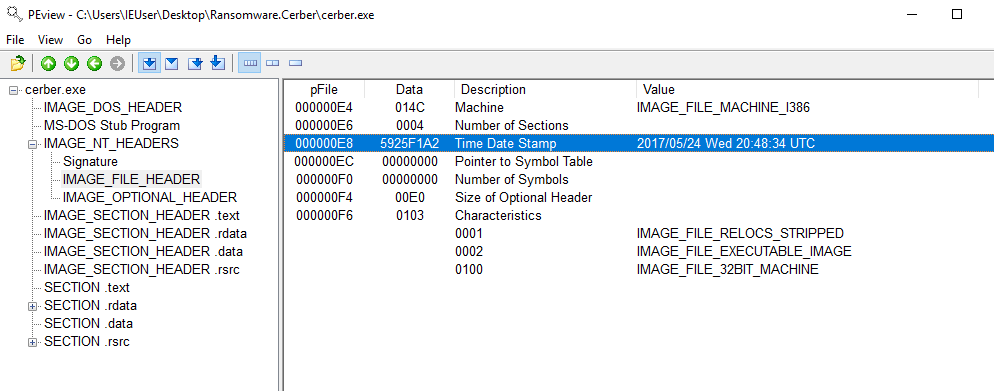


As author of the ExeInfo PE states on their website Exeinfo PE fails to identify older packers. So, I used PEiD, another tool that we can use to identify common packers, compilers, and cryptos for executable files. This tool can used to identify older packer information. When I open the cerber.exe sample in the PEiD it shows as “Nothing Found”. So, we can conclude either sample is actually not packed, or this sample uses some custom packer.

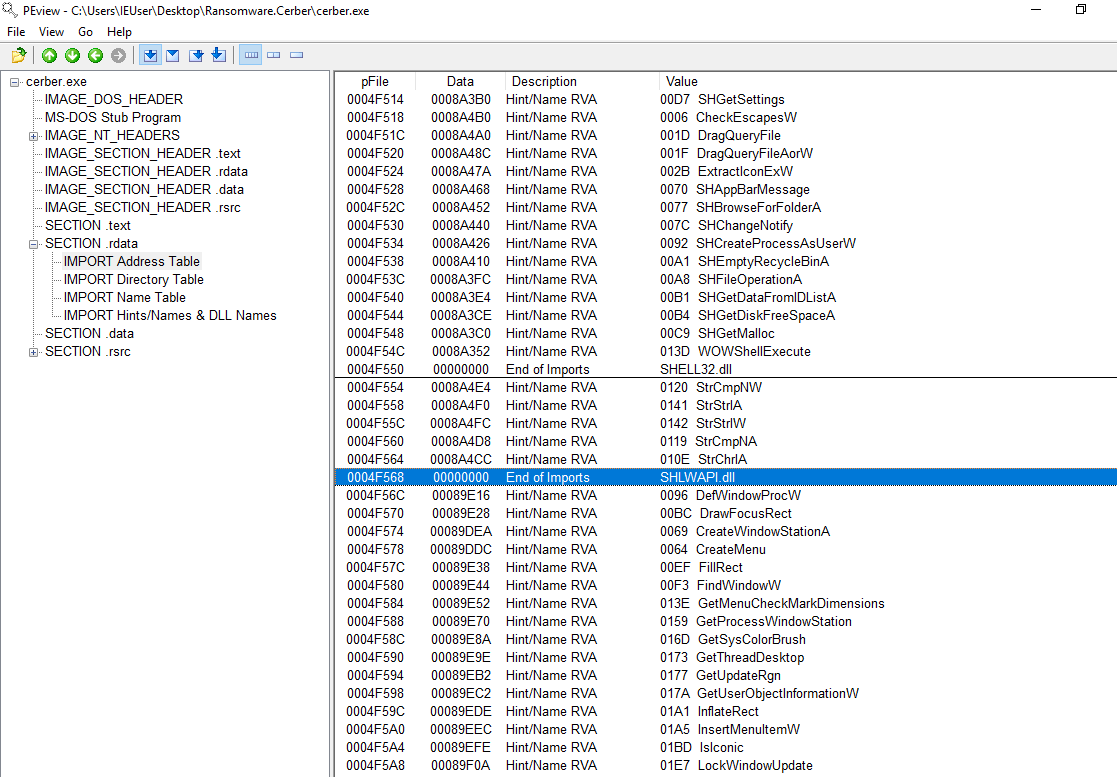


1. **Hidden file executables and Compilation dates**

PEview is another PE file analyzing tool which we can use to obtain more information about the malware sample. This shows information about section headers, signatures, and address tables. I have opened the malware sample on the tool, and we can identify the compilation date of the malware sample on the IMAGE\_FILE\_HEADER section of the results. It shows this was compiled on 2017/05/24 Wed 20:48;34 UTC. So, we can get an idea that this is a fairly old version of Cerber ransomware.

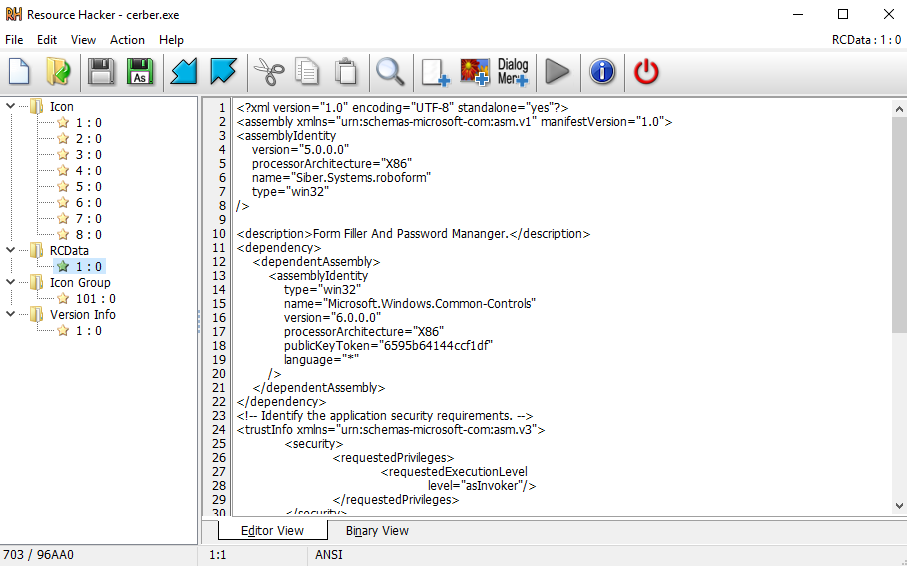


In .rdata section of the results displays read-only data of the file. We can identify the imported functions of the malware sample in this section. I was able to observe some uncommon Dynamic Link Library files being imported. SHELL32.dll which is a Windows API function used when opening files and web pages, SHLWAPI.dll which is a function for registry entries, URL and UNC paths, and color settings and COMCTL32.dll which is a GUI function. From this, we can get a basic idea of this malware sample to involve activities such as process manipulations, registry edits, and console executions.



Also, we can observe PEview results displays .rsrc section information. This is the resource section which includes information on resources and other executable files. It is difficult to identify embedded executables using PEview tool.

For this purpose, I have used Resource Hacker. Resource Hacker shows more information on resources such as icon, version information, and RC Data. RCData section displays raw data resources and embedded executables in the malware sample. As the below snapshot shows sample didn’t include any embedded executables other than the characteristics of the file.



1. **Strings**

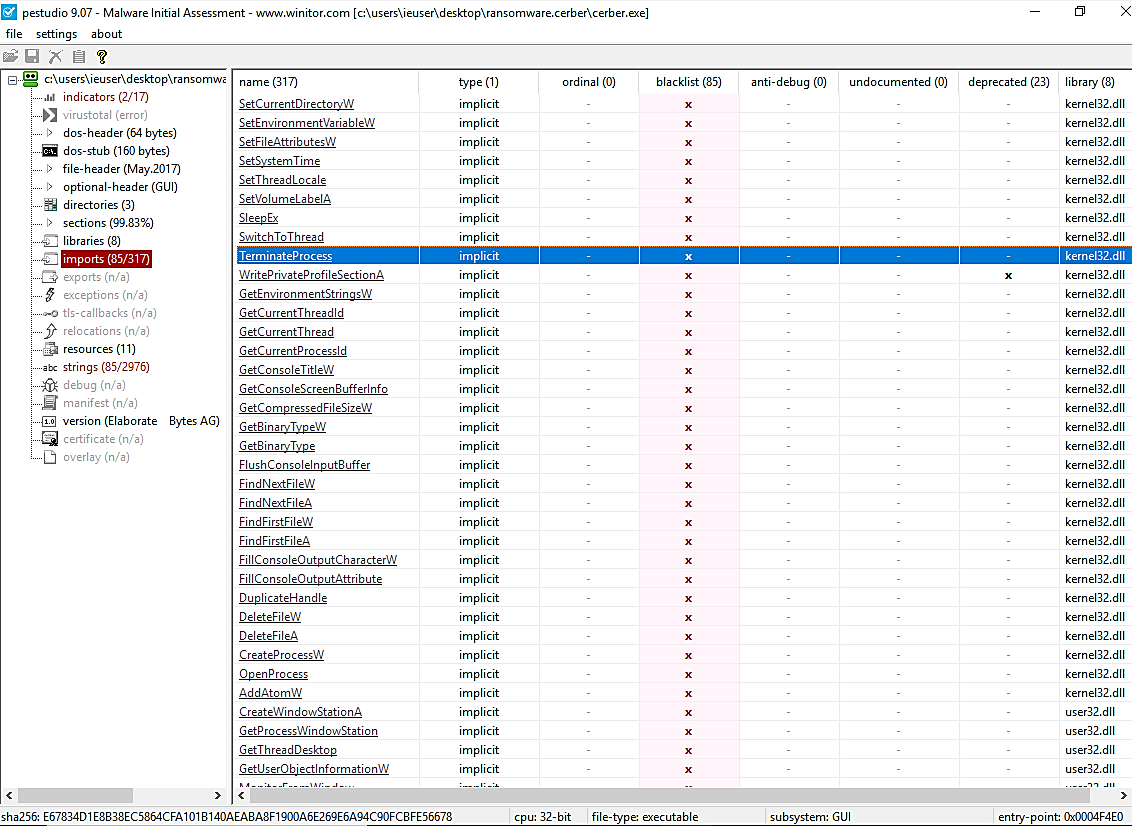
IDA Pro tool can be used to disassemble the malware codes for purpose of reverse engineering and debugging. I have opened the malware sample on the tool and observed the malware sample on the assembly language source code. Also, as the below snapshot shows I have searched for clear text strings in the sample to identify C&C communications and other notes. But there were no identifiable clear strings on the sample.

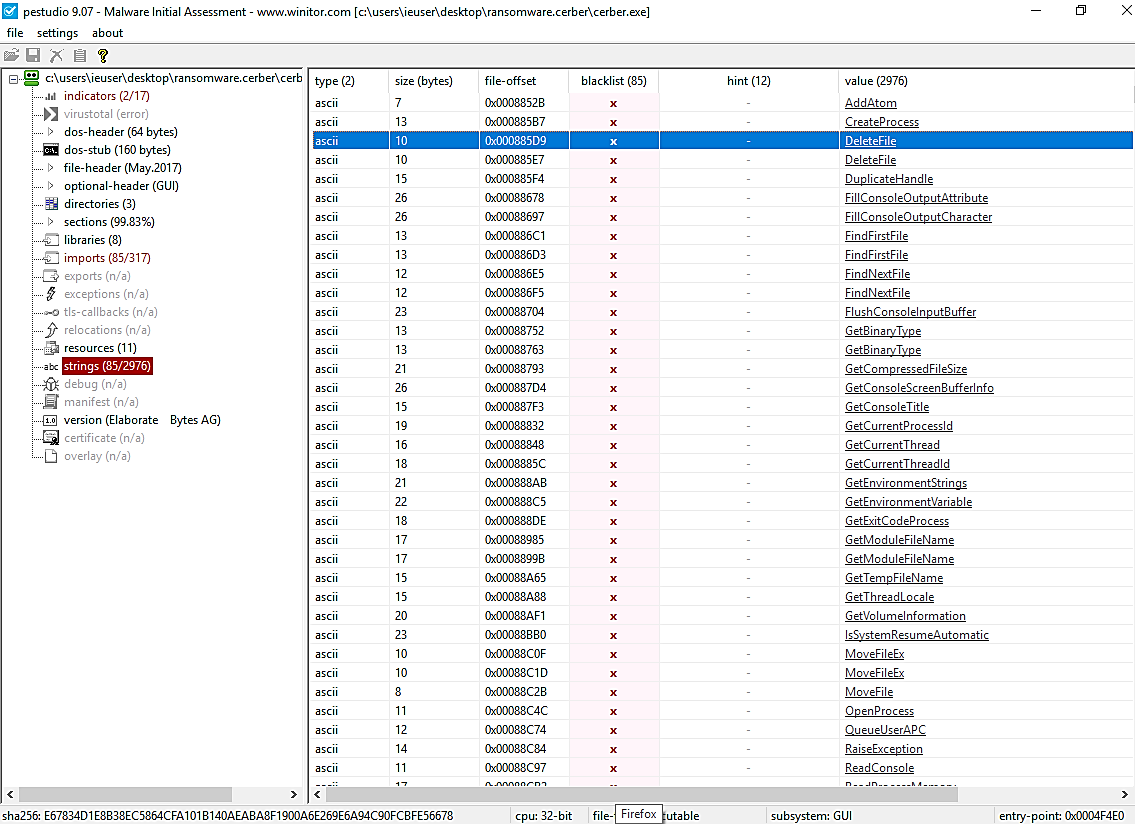


As the last static analysis tool, I have used the PE Studio. We can obtain most of the information we got from other tools from PE Studio as well. This tool shows every aspect of the PE header such as basic information about the file, indicators, dos-stub header, directories, libraries, etc.

The specialty of PE Studio is that it attempts to determine if a file is malicious and shows these using the “blacklist” tab. As below figures shows the imports section and the strings section shows the imported function and strings of the malware sample and we can sort these from the blacklist to easily identify potentially harmful among other things in the list. 85 out of 317 imports are identified as blacklisted. Among them, we can identify Terminate Process, FindFirstFile, FindNextFile, DeleteFile some suspicious files that can commonly identify in ransomware.

Also, even though we know for a fact that this is a ransomware sample, I couldn’t identify any specific crypto calls in these sections. We can assume this is because this malware sample is using obfuscation techniques, or because this uses custom encryption.





**Dynamic Analysis**

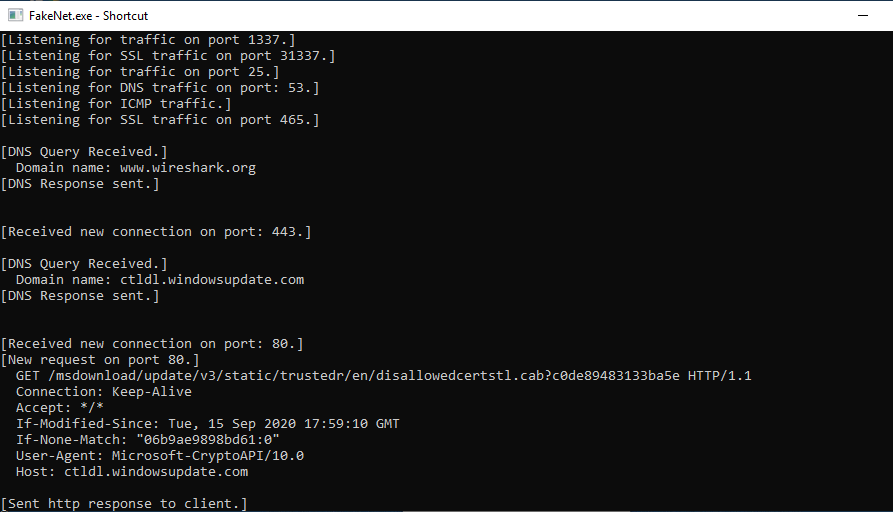
Dynamic analysis can describe as executing malware and analyzing its functionality and behavior. The main objective of this is to understand how and what malware does during execution.

First, before the analysis begins, I have opened all necessary dynamic tools in the analysis environment.

1. **Environment preparation**

**FakeNet**

FakeNet is a tool that simulates a fake network connection so that malware keeps interacting with a remote host and we can observe malware’s network activity without connecting to the internet.



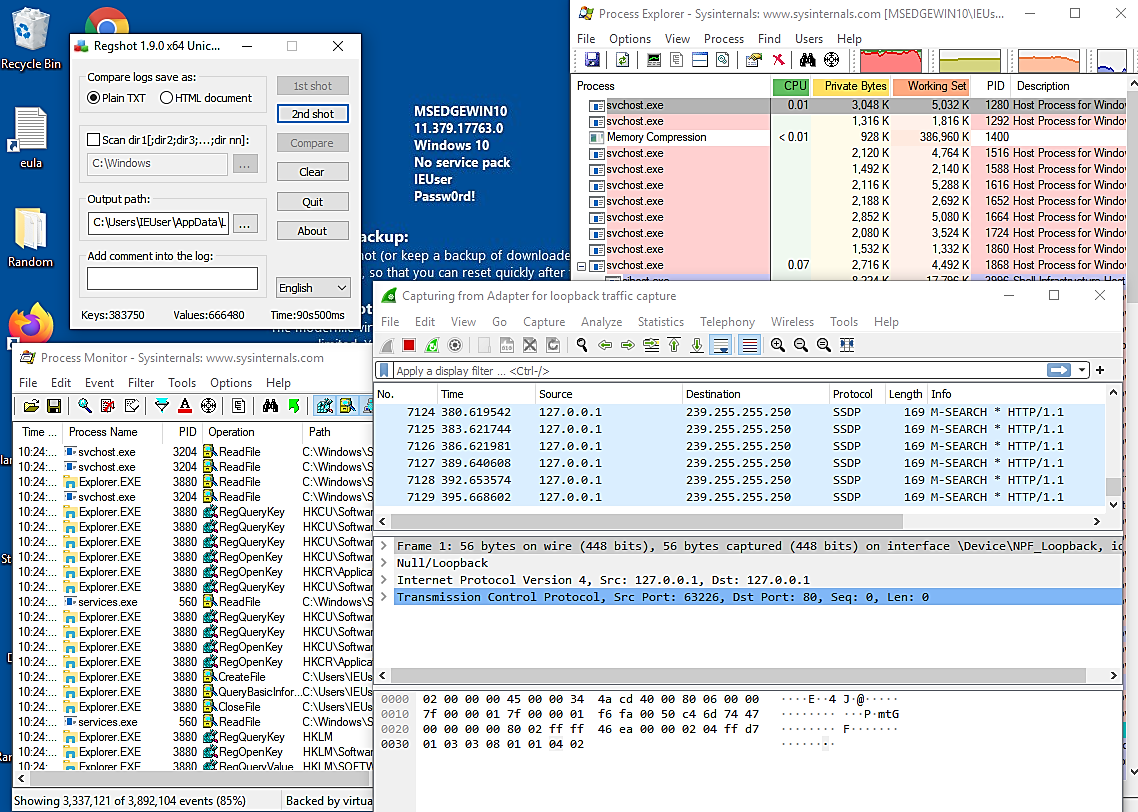
**FlyPaper**

FlyPaper is another tool that gives control over what happens in the virtual environment. Before the analysis begins, we can start the flypaper tool. I have unticked the option block TCP/IP since I want to analyze network traffic. I have prevented processes from exiting and record all behaviors.



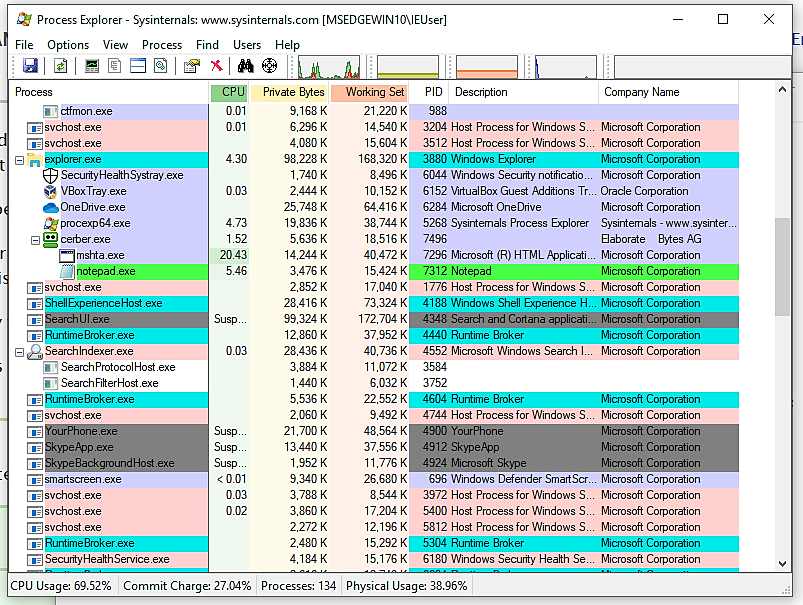
**Regshot**

As the below figure shows I have opened Process Explorer, Process Monitor, Wireshark, and Regshot to analyze the behaviors and changes to processes, network traffic, and registries. Before executing the malware, I have taken the first shot of the windows registry using Regshot.



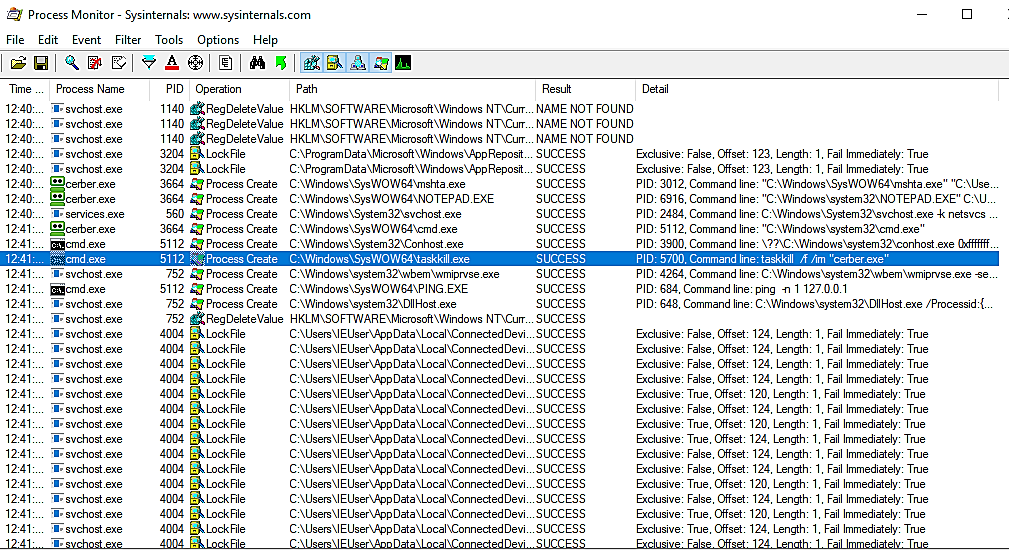
1. **Processes running and Registry changes after the execution of the malware**

When the Cerber.exe malware sample is executed as shown in the below figure we can see that Cerber.exe starts a two-child process, mshta.exe, and notepad.exe. When I further analyze I was able to find out that these two applications were used to display the ransom message.

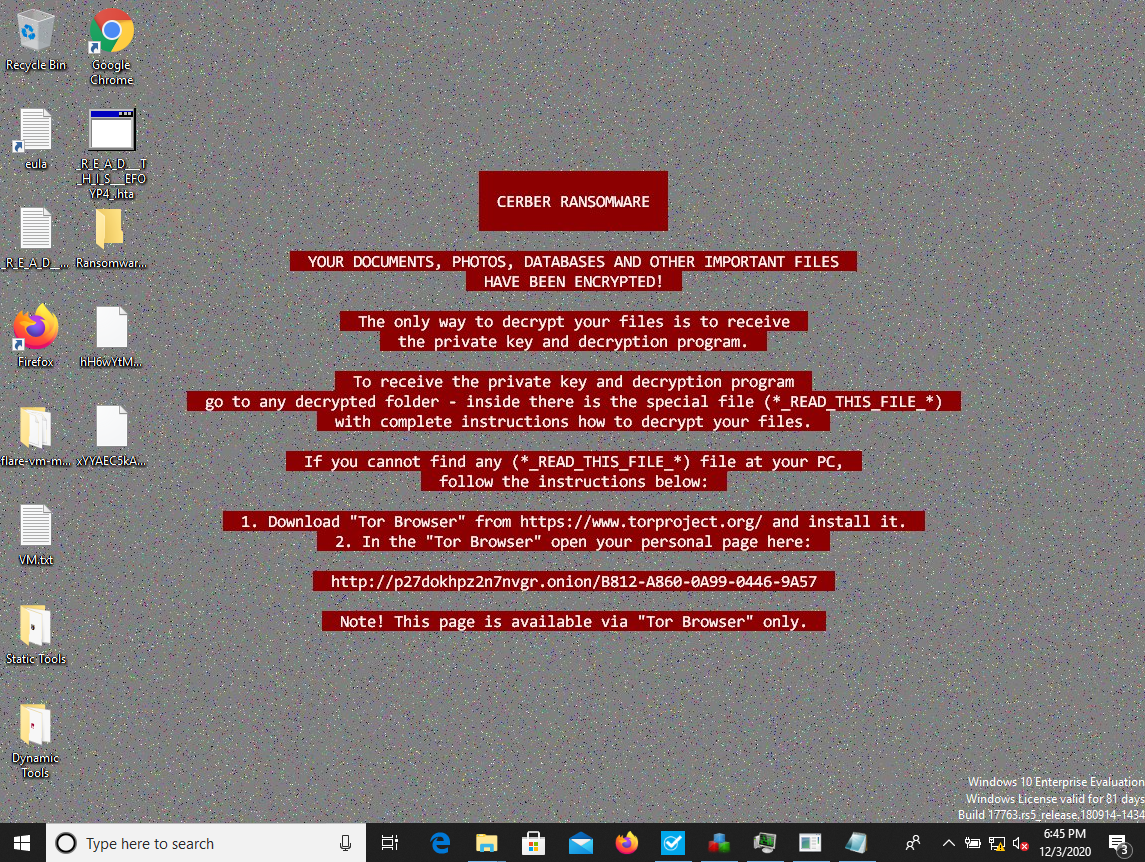


I have configured the Process Monitor to specifically monitor the process creations, lock files, and registry value deletions. As shown below the figure in Process Monitor, we can see that after beginning the two-child process cerber.exe initiating the process cmd.exe with path taskkill.exe. We can see that cerber.exe kills itself within seconds after initiating two child processes.

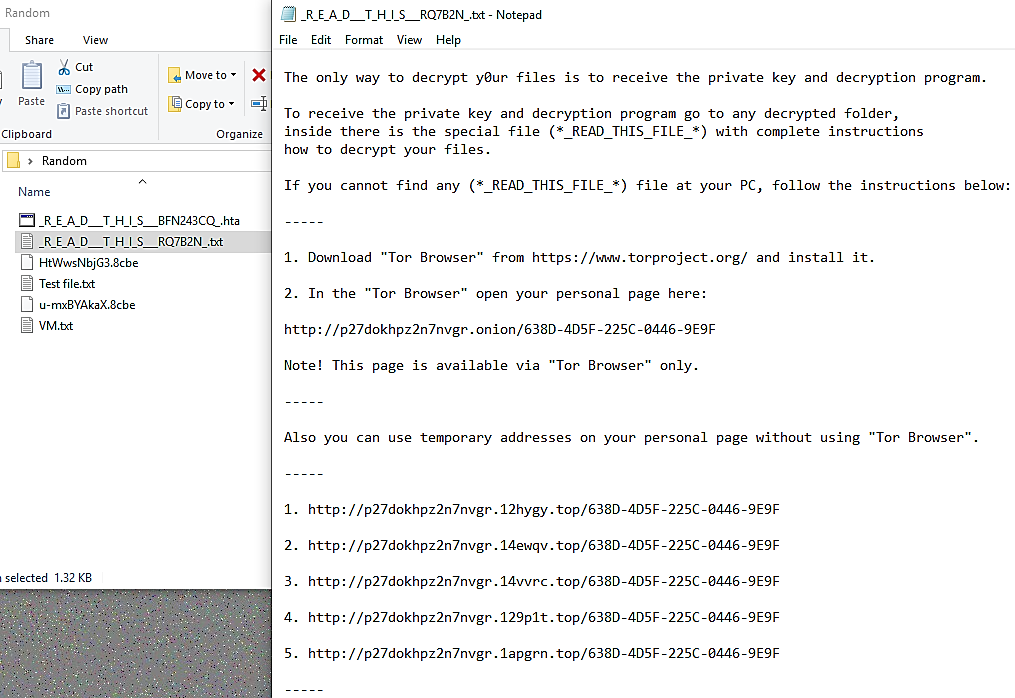
Also, we can identify that the ping command is initiating to the localhost.



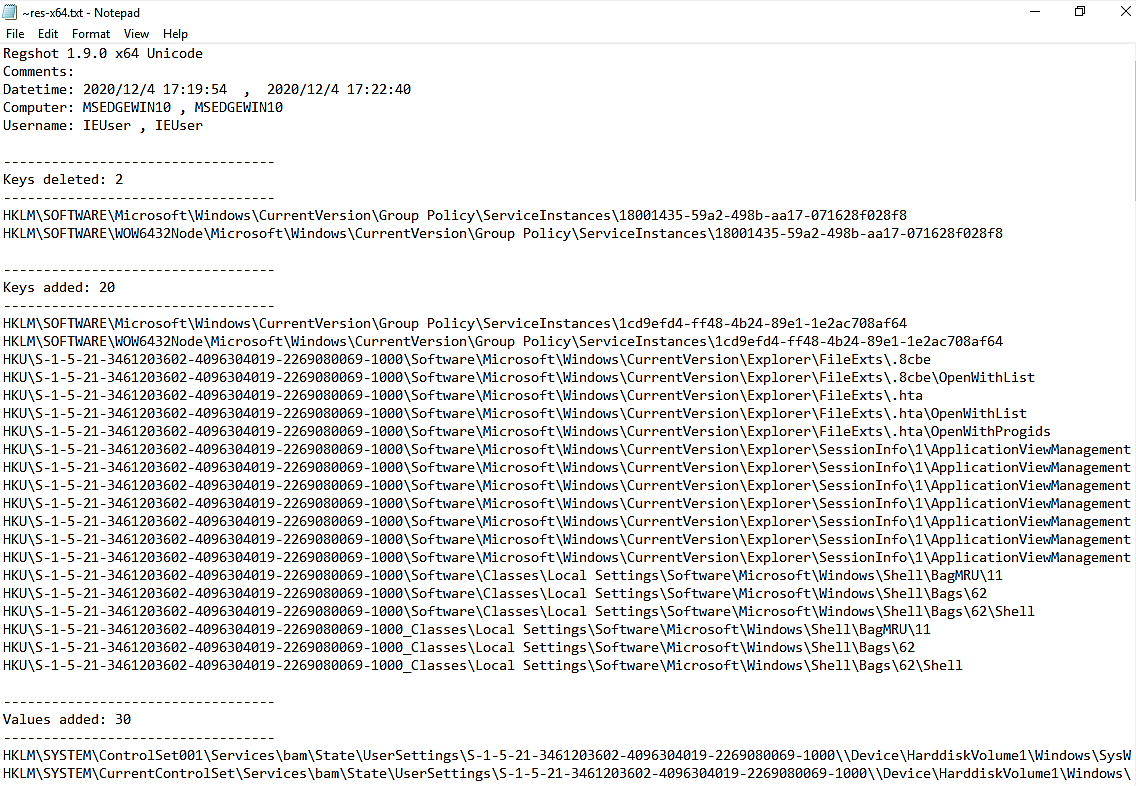
After Cerber.exe kills itself we can see ransom note displays on the screen as below.



Also, we can see that in every folder ransomware encrypt the files shows this file “\_R\_E\_A\_D\_T\_H\_I\_S\_\*\_.txt and .hta files with ransom note available. Ransomware encrypted the files on the machine and replaced all the common extensions with .8cbe extension.

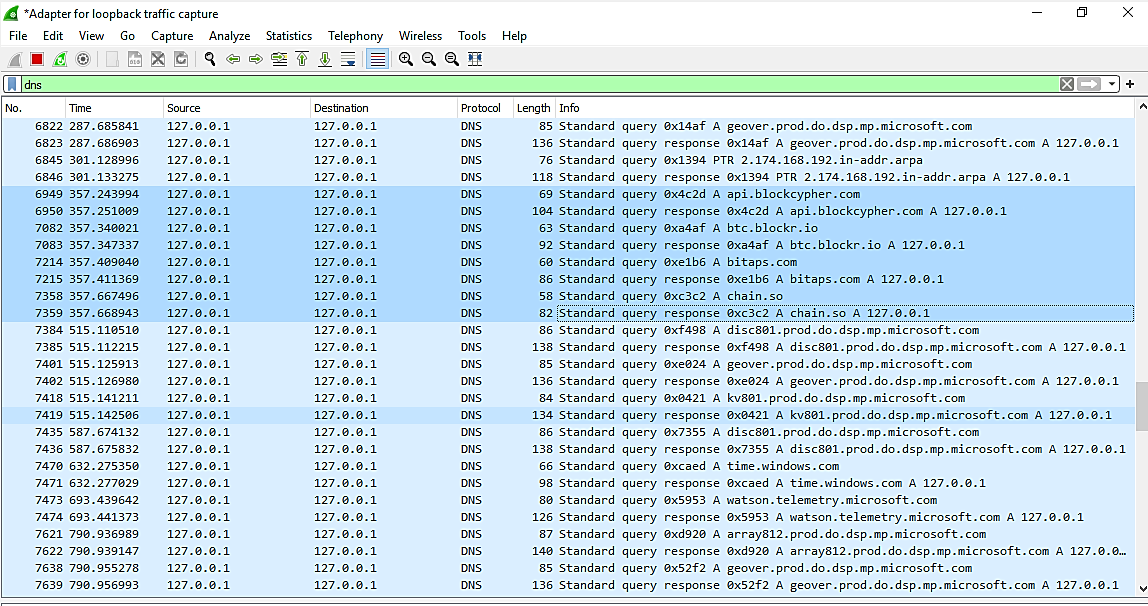


After ransomware completed execution, I have taken the second shot in Regshot and compare those by using the compare button. As shown in the below snapshot we can identify ransomware sample has added some keys, values and delete some keys from the registry.



1. **Network Activities**

I have observed the Wireshark network traffic logs after the execution of the malware and I cloud identify after encrypting the files, malware sample tries to connect to api.blockcypher.com, chain.so, bitaps.com and btc.blocker.io domains.



When I search for these domains, I was identified these are platforms that power cryptocurrency applications. So, I search for HTTP traffics to identify whether there is any GET request to obtain resources.

As the below figures show I was able to identify a sample is tries to get

* <http://btc.blocker.io/api/v1/address/txs/17gd1msp5FnMcEMF1MitTNSsYs7w7AQyCt>?
* <http://api.blockcypher.com/v1/btc/main/addrs/17gd1msp5FnMcEMF1MitTNSsYs7w7AQyCt>?

String 17gd1msp5FnMcEMF1MitTNSsYs7w7AQyCt can be identified as a publicly available blockchain wallet address to obtain the ransom payment.

