Malware Analysis

Abhinandan Kainth | Anuneet Anand | Mayank Rawat | Rakshit Singh | Tejas Dubhir

Malware analysis is the process of understanding the functionality and potential impact of a malicious software. It involves dissecting the malware to comprehend how it works, how it can be identified and how it can be eliminated or defeated.

Computers and smart devices have become an integral part of our lives. People are dependent on computers for various tasks. Any disruption of digital services or loss of data can create havoc in the lives of individuals and organizations. The criminals and sociopaths have a never ending zeal to design custom softwares aimed at disrupting, damaging or gaining access to a computer system. Such malicious softwares are collectively called Malware and include viruses, trojans, worms, ransomwares, etc.

Malware has always posed a threat to computer users. In recent years, malwares have become more invasive and difficult to detect. There is a growing need to study the working of the latest malware and making our systems resilient to their attacks. Analysing the structure and working of malware helps us understand the vulnerabilities in the systems and sharing this knowledge with others motivates people to develop secure systems.

Malware analysis is usually done in two steps :-

- **Static Analysis :** It involves analysing the malware file without actually executing it. The insights are obtained by studying indicators like hashes, strings, IP addresses, headers in the file. The malware might be disassembled to understand the instructions and modifications undertaken by it.

- **Dynamic Analysis :** It involves executing the malware code in a controlled and isolated environment called sandbox. The processes spawned by the malware are studied and network communications are intercepted. Behavioural insights are drawn and analysed.

The purpose of this study is to analyse the working of a keylogger and ransomware. We will analyse the malware with sandboxing, identify their API calls, and try to reverse engineer the mechanism of the malware. We will try to apply static and dynamic analysis and further, try to draw parallels between them.

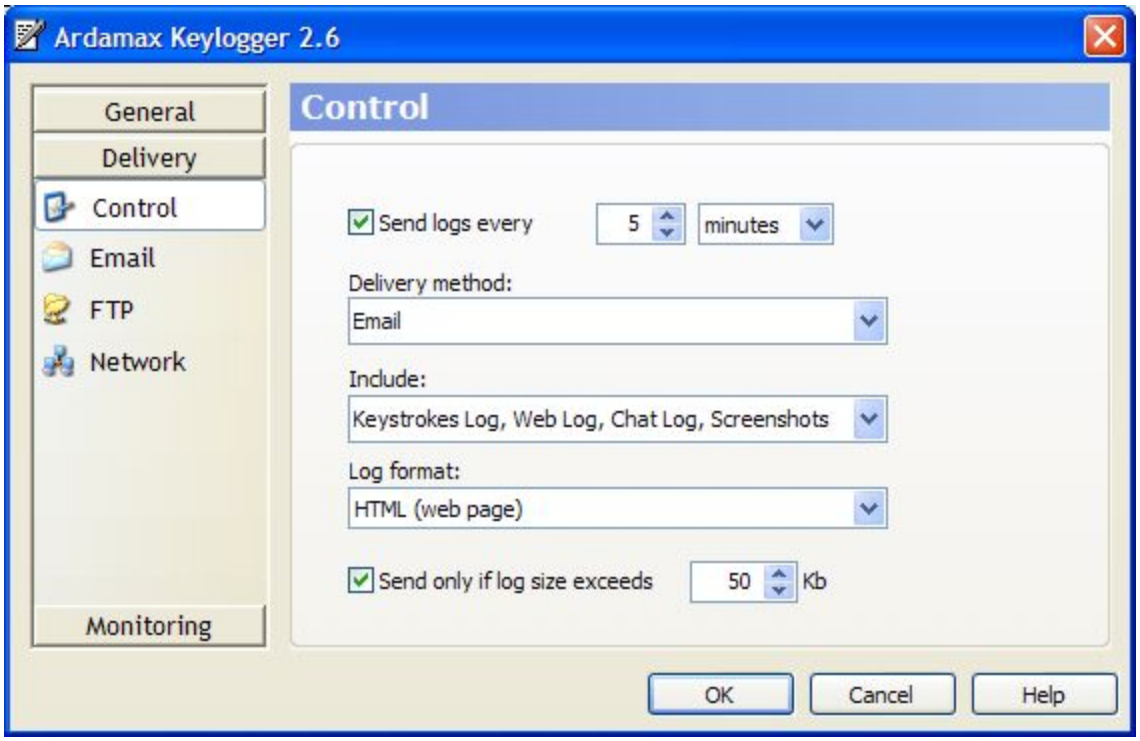
Keylogger - Ardamax

- Ardamax is a keylogger malware which takes periodic screenshots of the screen of the host and records all the logs of the keys pressed.

- It then sends all the recorded data i.e. the key logs and the screenshots through SMTP to a yahoo.mail id which is no longer active, thus the packets are not acknowledged and thus the sending is failed.

- If the host opens its source folder, the process DPBJ.exe which is the main executable process of the Ardamax, is terminated and can be seen in the hidden processes in any process monitor.

- Also, at regular intervals, the logs and screenshots are deleted to avoid suspicion.



Source : https://www.securitystronghold.com/gates/images/ardamax-keylogger.png

Static Analysis

The suspected malware file is run through various antivirus programmes using the website VirusTotal and out of the 68 antivirus engines, 62 flagged it as Ardamax Keylogger, spyware or just as a malware.

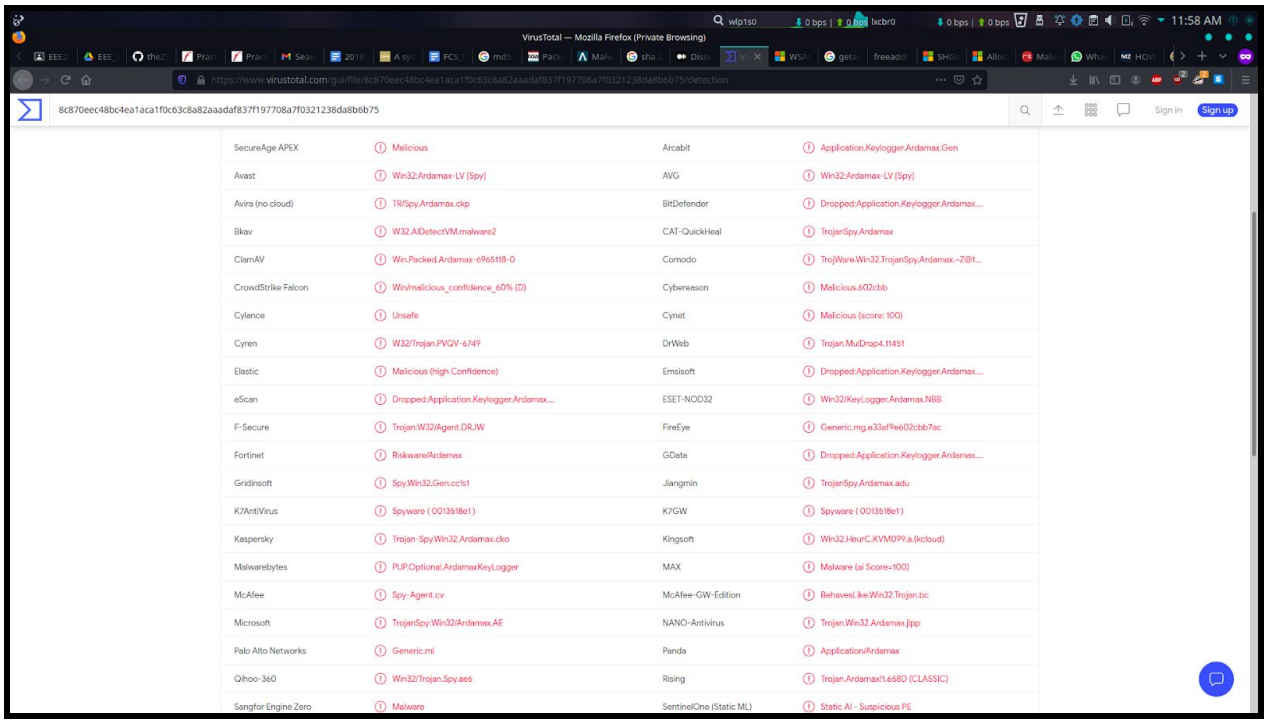
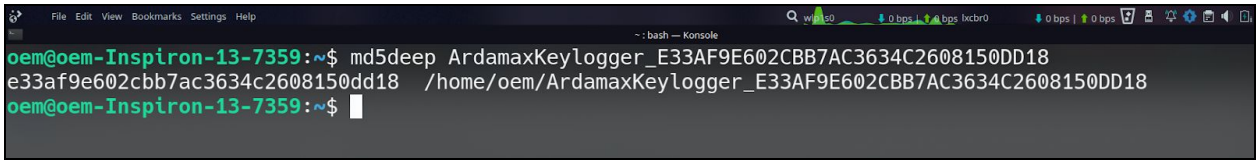


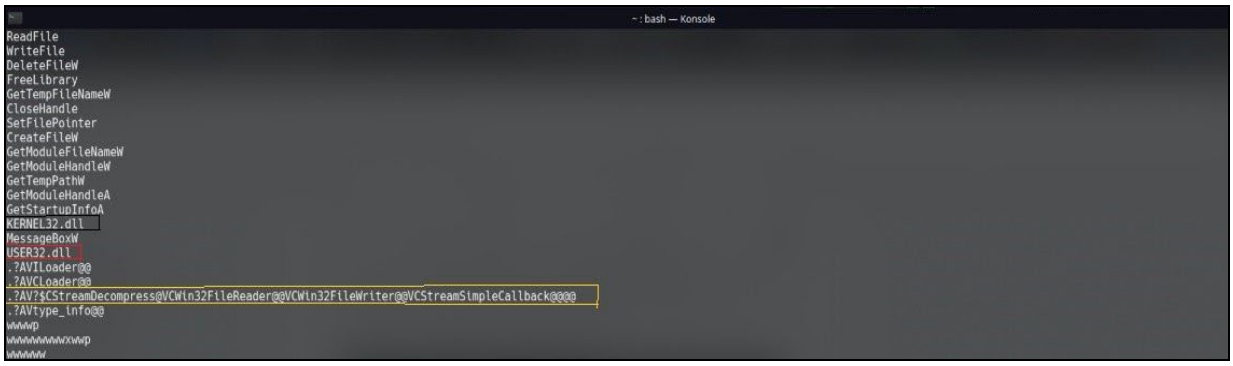
Fig. 1: Results for running the malware file through several antivirus programs

Now, the md5deep hash function is used to calculate the hash of the malware. This hash can be used to uniquely identify the malware and acts as a fingerprint for the malware file.

md5deep hash : e33af9e602cbb7ac3634c2608150dd18



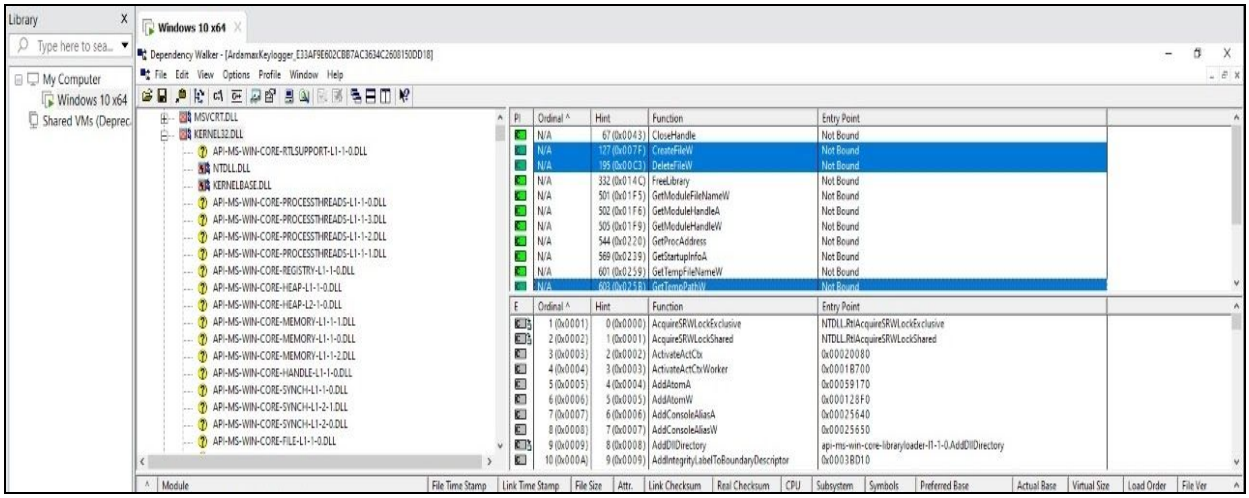
On searching this hash on google, we see that this hash indeed belongs to **Ardamax Keylogge**r.
On analysing the strings present in the keylogger, some of the interesting observations are:

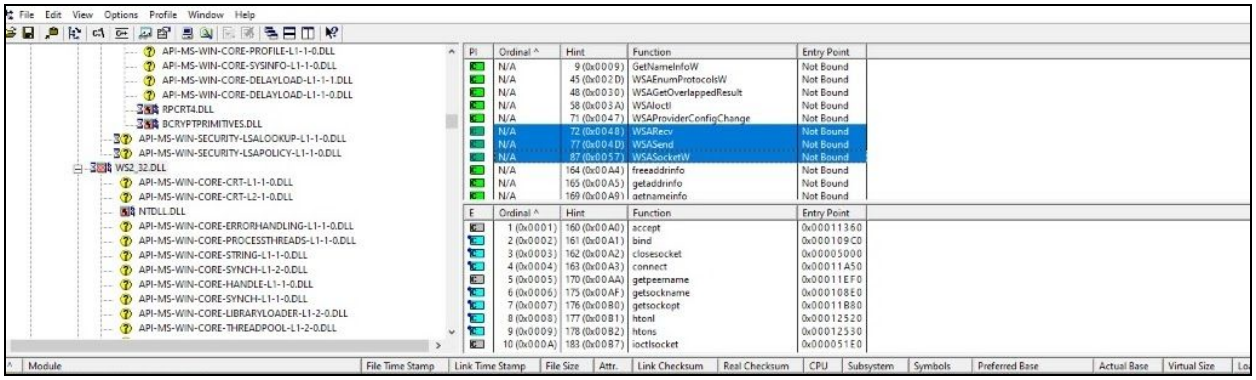


Moving on to further analysis of the malware, the malware is loaded in the *Dependency walker* to see the linked functions present in the executable.

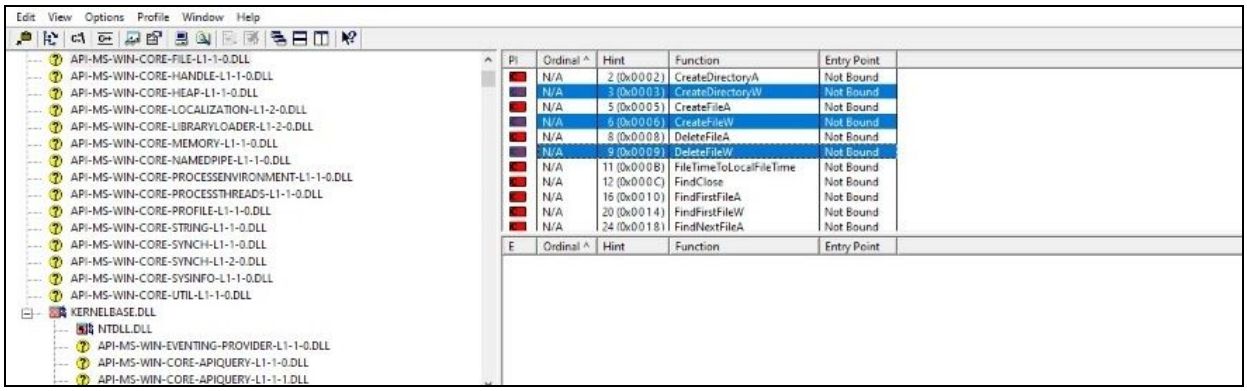
On analysing the imports from *KERNEL32.DLL* we see an interesting function namely, *CreateProcessAsUserA*, the presence of this function in the executable suggests that the **executable may create another process**, so we will watch out for creation of a new process during dynamic analysis.

On further analysis, we see *GetTempPathW, CreateFile, DeleteFile, WriteFile* API calls. We need to check whether the **malware reads or writes to any file in the temp folder**.





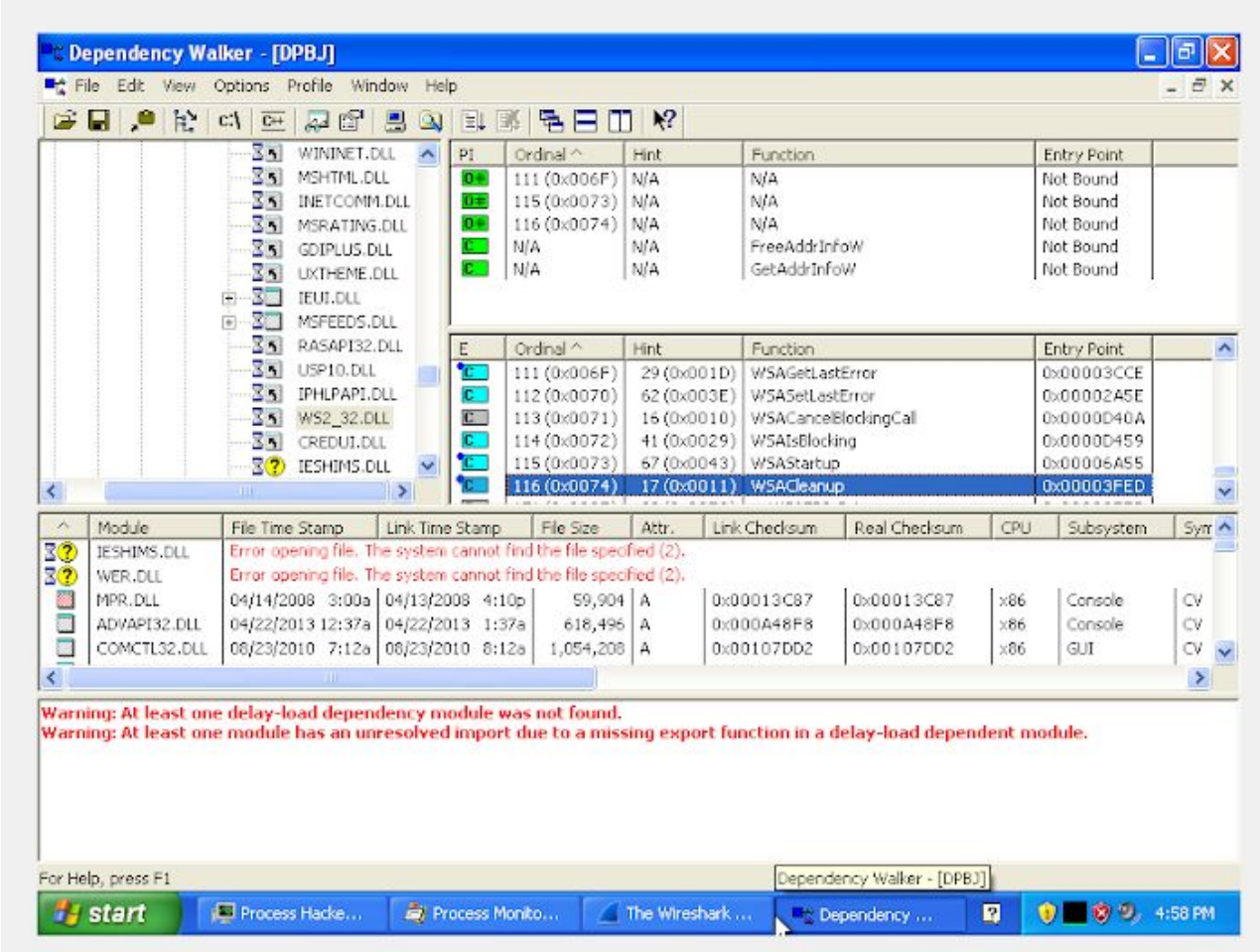
MSVCRT.DLL - imports like *CreteFileW, CreateDirectoryFileW, DeleteFileW, WriteFileW*, suggests that the executable might create a directory or a file and read and write to these files and create a new process. We **need to keep an eye out for any files** created on running the malware.



The functions CreateProcessW and GetStartupInfoW again hint towards the **possible creation of a process by the malware**.

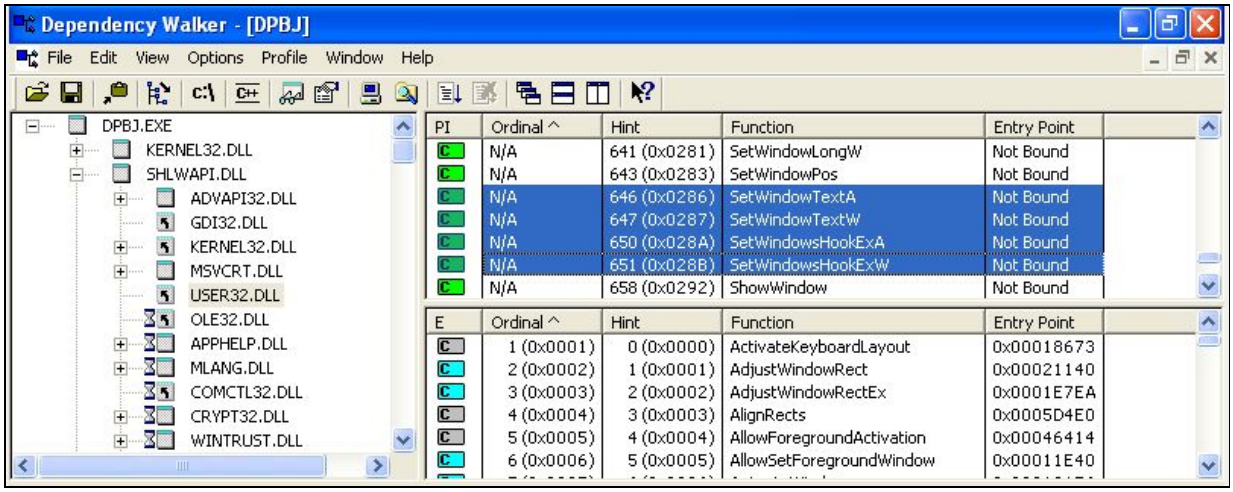
On performing the dynamic analysis, we see that the malware drops an executable file in the systems folder and this executable named DPBJ is the actual executable that does the work of the keylogger.

On loading this executable in the dependency walker, we see the executable imports function from ***WS2_32.DLL(networking DLL)*** which suggests the executable might be talking to the internet.



some useful imports from *USER32.DLL* found are-

- ***SetCapture*** - to capture the mouse input.
- ***SetWindowTextW*** - to set text in the window.
- ***SetWindowsHookExW*** - sets a function to be called whenever an event (ex - say mouse input) occurs. Commonly used with keyloggers. Most common way that keyloggers receive keyboard input.



Since most of the antivirus softwares flagged the program as malicious, the malware is probably a keylogger.

Dynamic analysis

Goal : To observe how the malware executes and what all changes it makes on the host system.

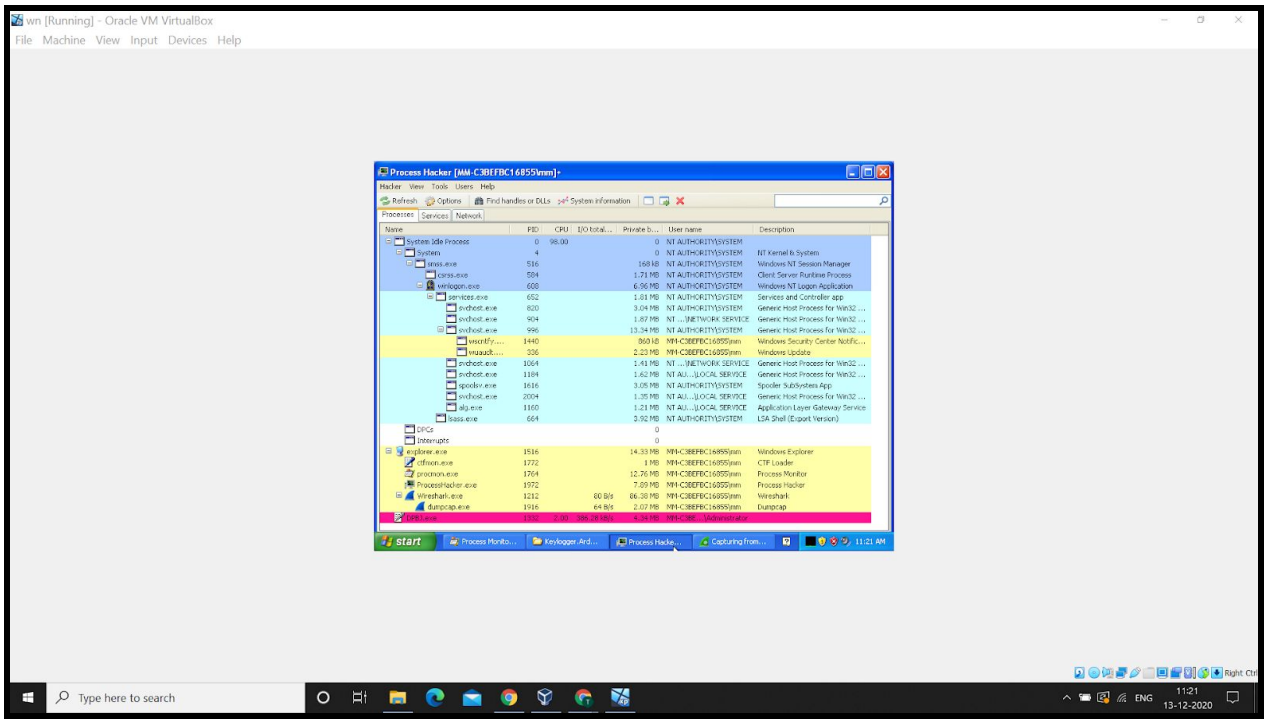
First we activate the process monitors, like *procmon* and *process hacker*, which can track and record all the processes which are being executed and packet tracers like Wireshark to capture the packets sent from the host system by the malware.

We set up a virtual device with *Windows XP* in it so that there are not as many background processes running as the newer versions. And once we close/terminate all the processes and browsers which might send packets outside, we enable the capture process of wireshark and process tracers and execute the malware(here, keylogger).

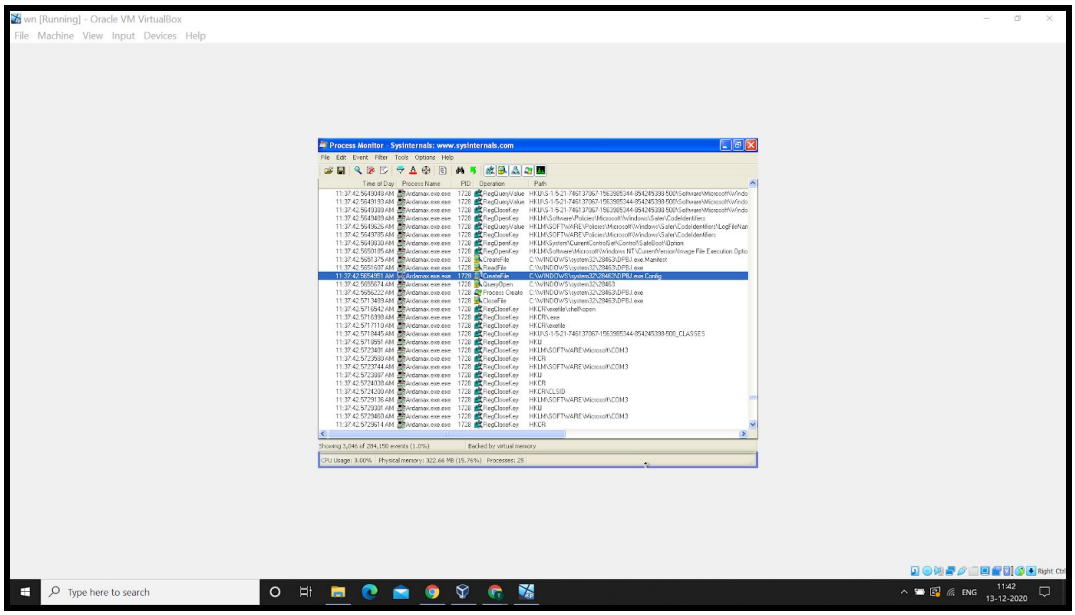
Here, we analyse the keylogger Ardamax, which along with the keylogs also traces and sends the screenshots of the screen at a regular interval through *SMTP* protocol.

Initially, there is no activity from the keylogger for a few seconds, but once some keys are pressed, the malware starts running and we can see its presence in the process Hacker.

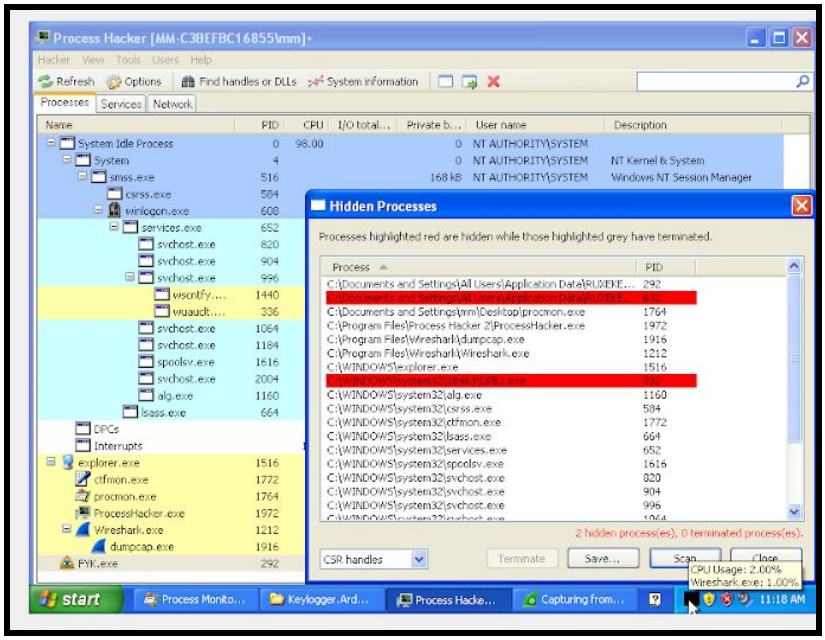
In *Process Hacker* the newer processes are coloured red whereas the already running processes turn blue with time.



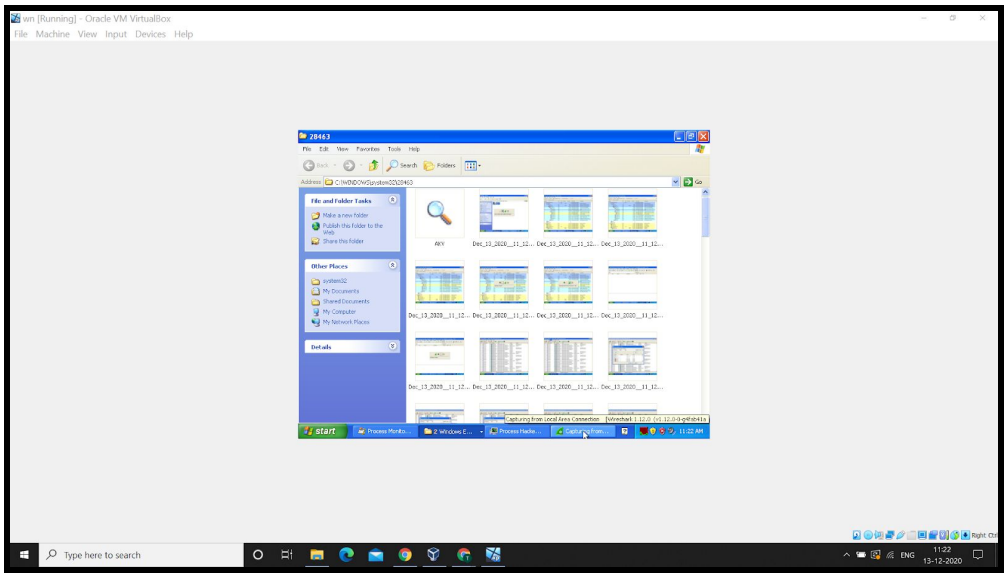
Once the executable file *Ardamax.exe* is run, it forks and the child process is named *PDBJ.exe*. The main executable for the keylogger is *PDBJ.exe* which makes all the moves like recording the key logs and taking continuous screenshots.



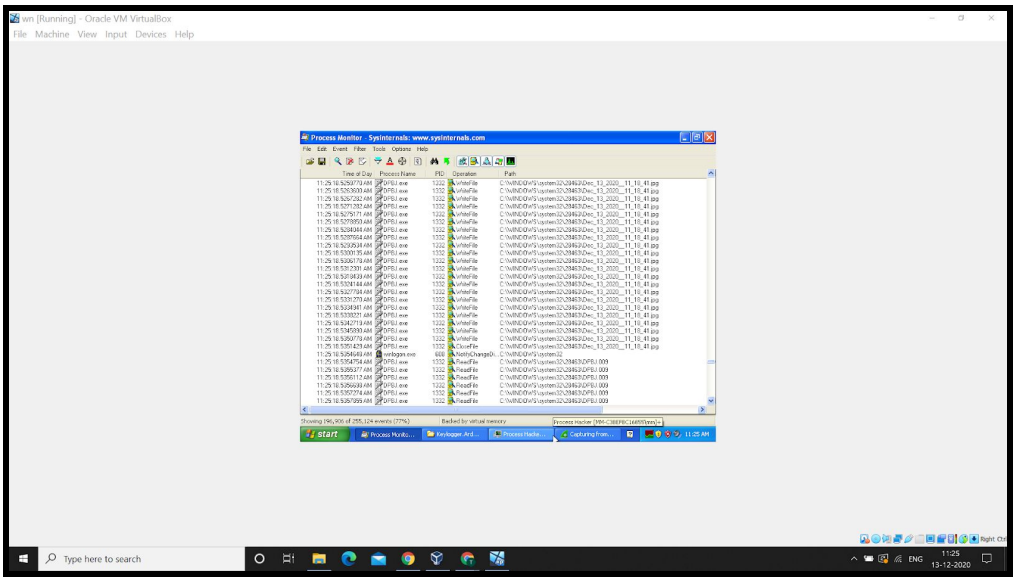
When the keylogger is not capturing the key logs and screenshots, it hides itself and can be traced in the hidden processes. Also, the Ardamax keylogger is run by the name of *PDBJ.exe* and thus can be seen in the active/running processes.



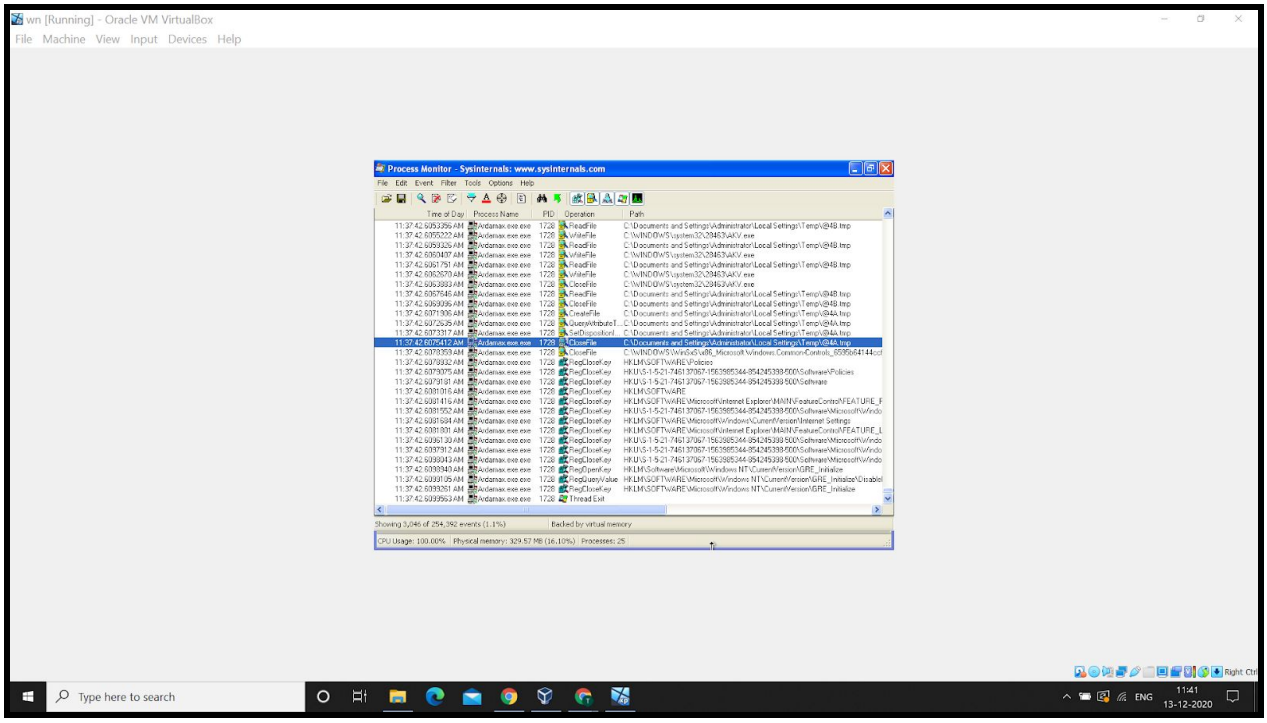
In the *Process Hacker*, we can access the folder in which the process is storing its files. When we open the folder, we can see the media stored in it and after a specific interval, it deletes the screenshots on its own.

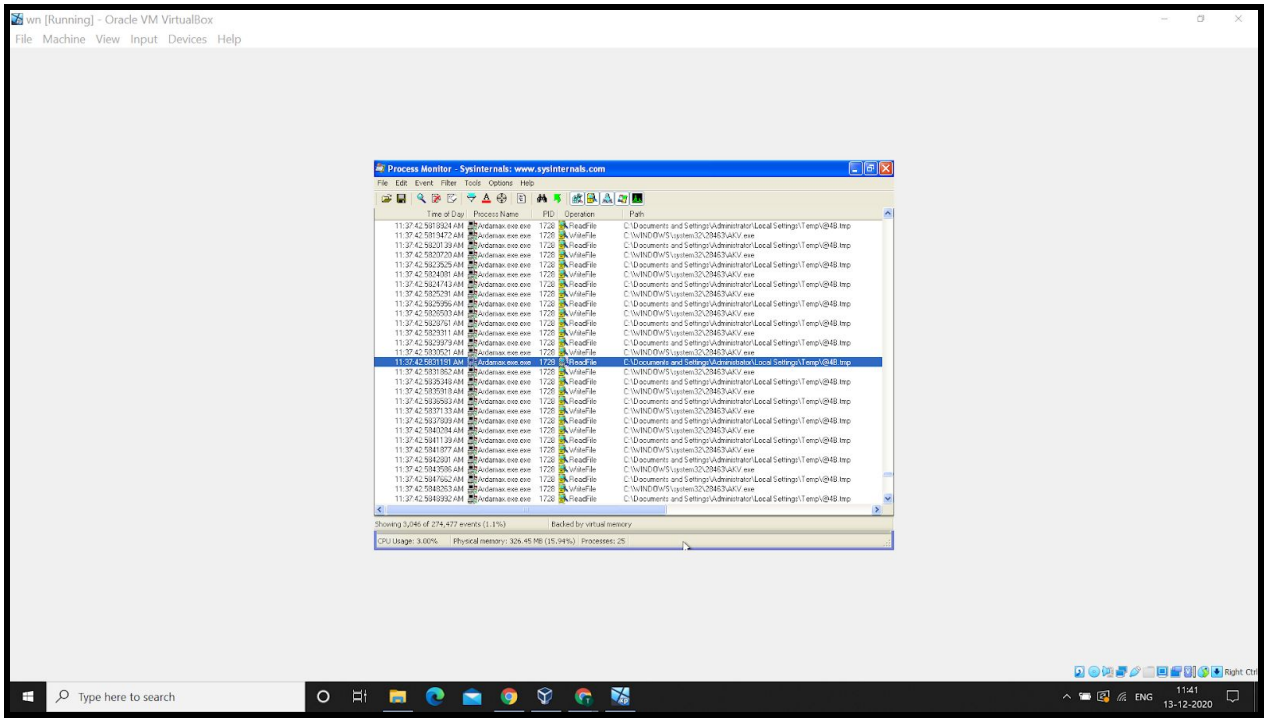


We can confirm that the screenshots are written by the process PDBJ.exe and not already there by finding the “write” operations in the logs of procmon. We can even confirm the destination of the files written which is the same as the root folder of PDBJ.exe.

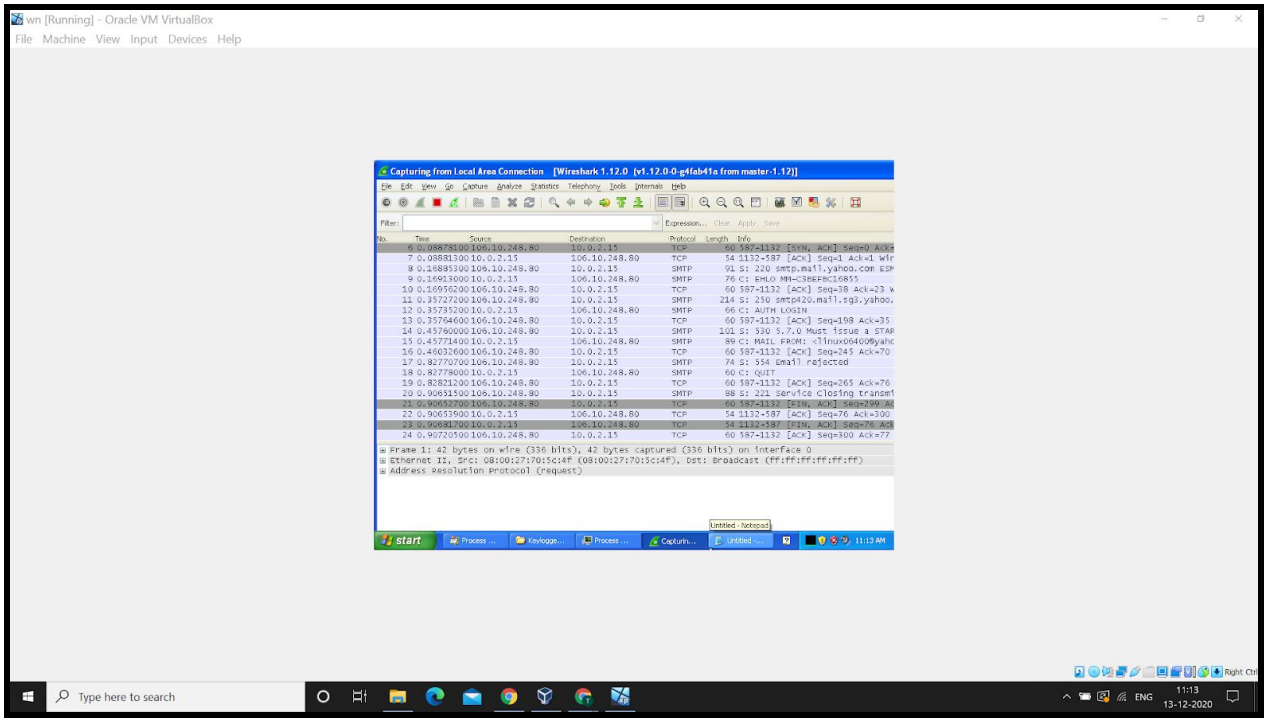


We can confirm that the malware is modifying the contents of the host system stealthily by finding the parent process modifying, erasing and writing temporary files in the temp folder.

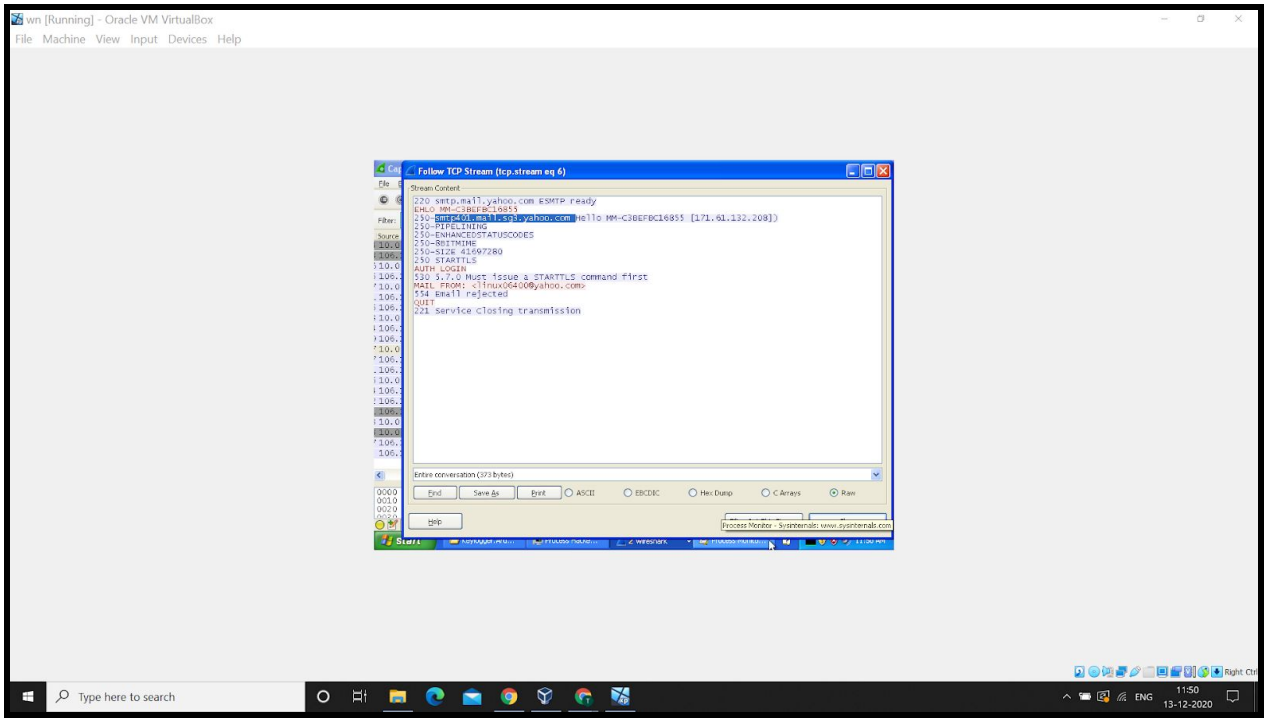




Since, the malware is confirmed to be recording the host's screen and key logs. Now, we try to catch it while sending these files to some other PC through the internet. In wireshark, we can see that there are packets trying to be sent through SMTP protocol, to a yahoo mail account.



We can see the destination IP address in Wireshark along with a lot of more packet details by following the TCP stream.



Ransomware - JigSaw

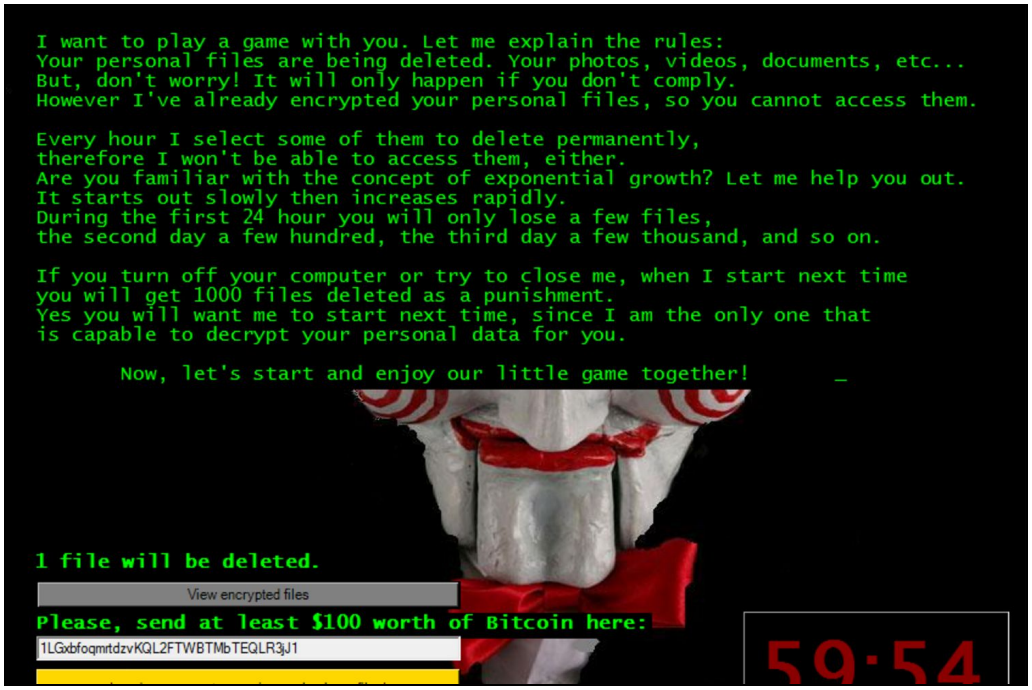
- Jigsaw is an encrypting ransomware malware which was created in 2016. It was initially called "BitcoinBlackmailer" but later became famous as Jigsaw as it featured an image of Billy the Puppet from the Saw film franchise.

- Jigsaw primarily spreads through attachments in spam mails and is activated if the user downloads the attachment.

- Once activated it prompts the user with an intimidating message specifying how the user files have been encrypted and the only way to retrieve them is by paying the required ransom.

- Unlike other ransomwares, JigSaw uses a countdown timer to worsen the situation for the user. Every hour the files get deleted incrementally.

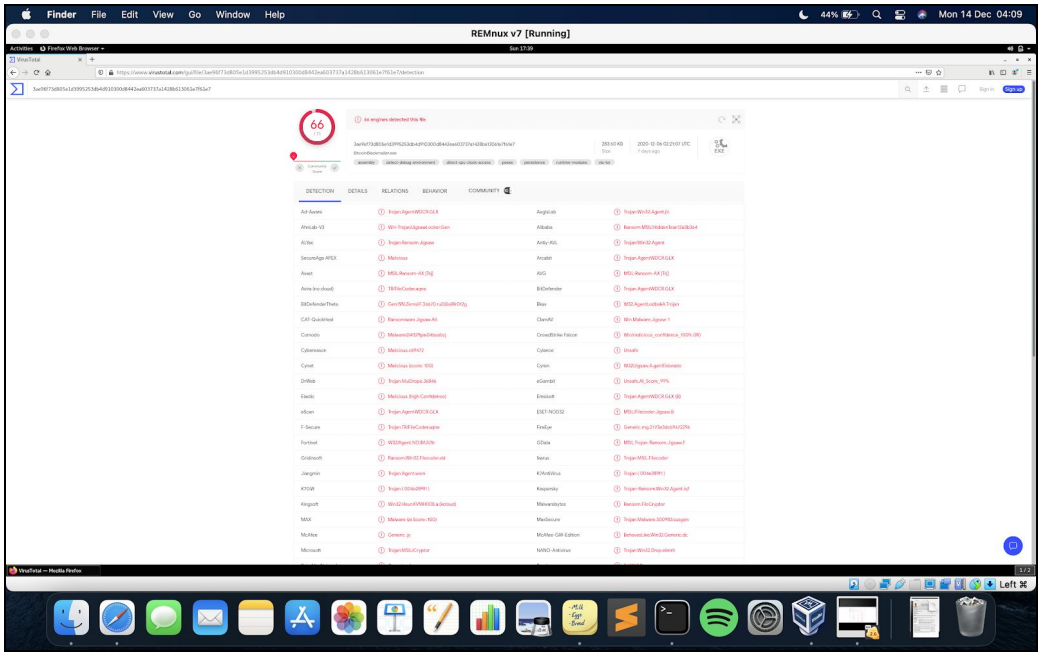
- If the user tries to restart the computer, 1000 files are deleted instantly as a punishment and JigSaw continues to execute.



Source: https://sensorstechforum.com/remove-jigsaw-ransomware-restore-paytounlock-files

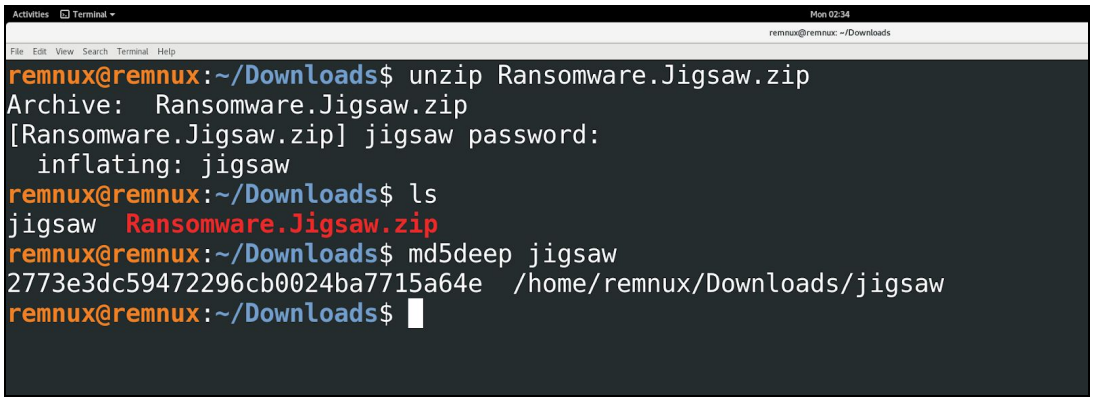
Static Analysis

The suspected malware file is run through various antivirus programmes using the website VirusTotal and out of the 71 antivirus engines, 66 reported it as a malicious software.



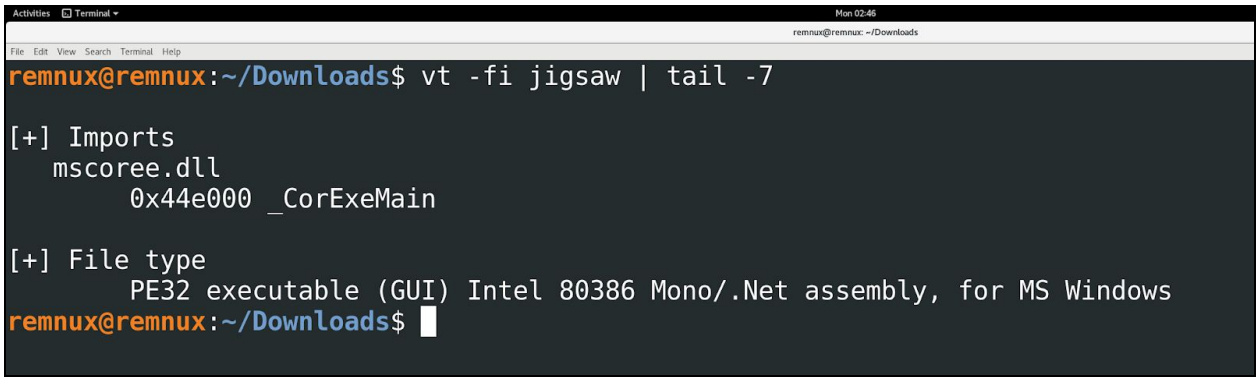
We then calculate the md5deep hash of the jigsaw file. This hash can be used to uniquely identify the malware and acts as a fingerprint for the malware file.

md5deep hash : 2773e3dc59472296cb0024ba7715a64e

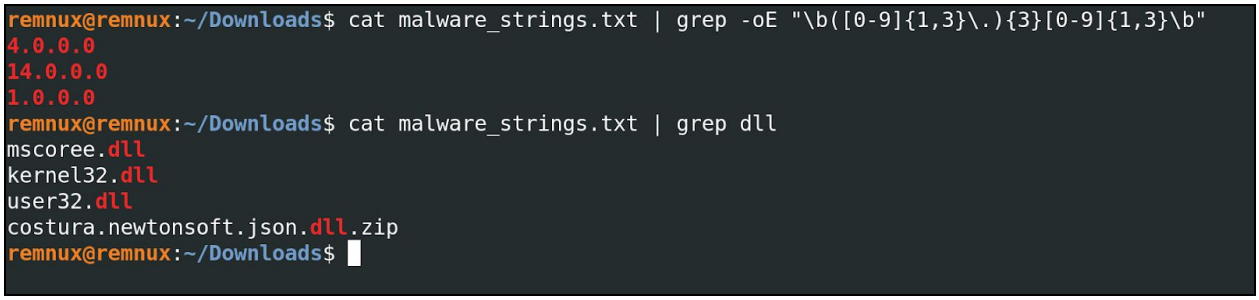


On searching for the hash online, we come across various malware reports for this file. We derive some important insights from the VirusTotal website.

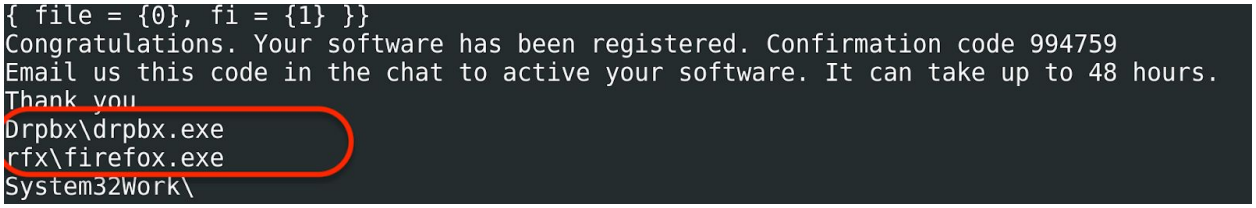
Using the VirusTotal API, we get some more information about the ransomware. Here we note that mscoree library is imported by the jigsaw program.



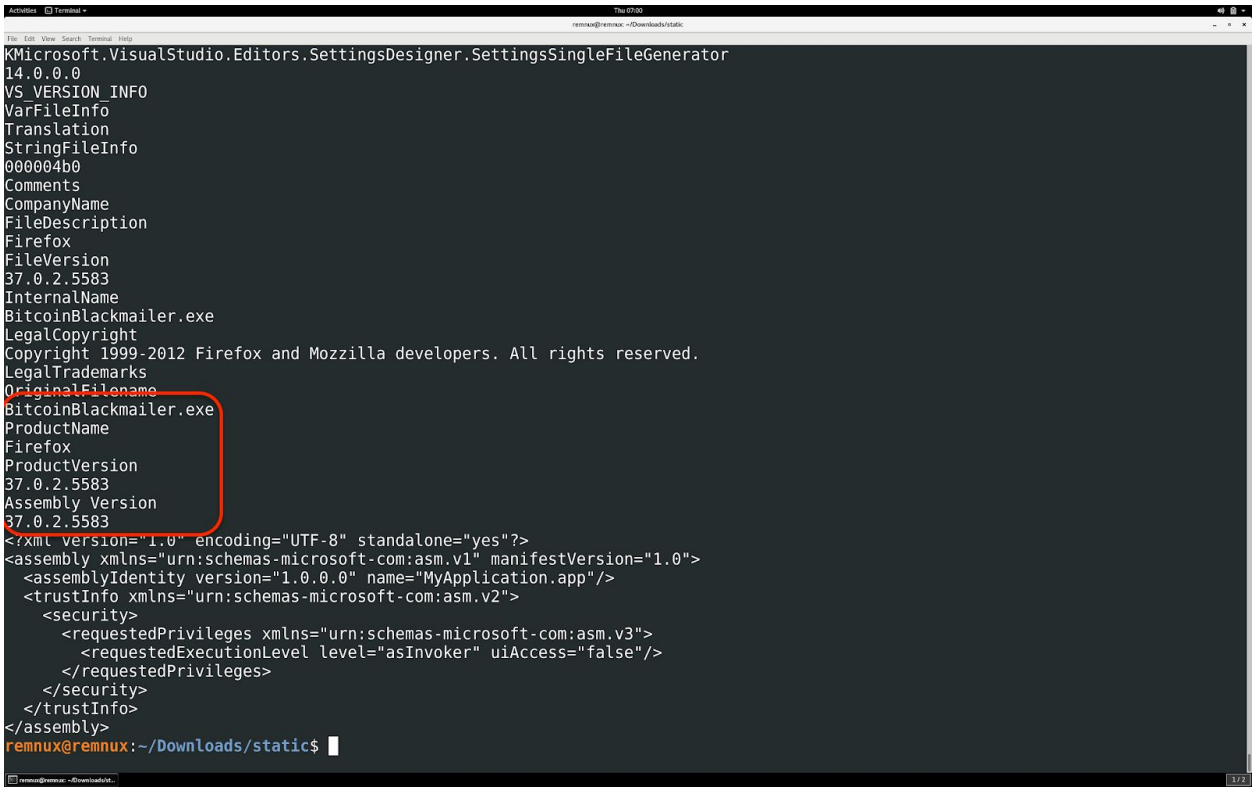
Now we run strings and pestr on the jigsaw file to identify some crucial parts of the file. We lookup possible IP addresses in the file using a regex and also note the different libraries being used by the jigsaw.



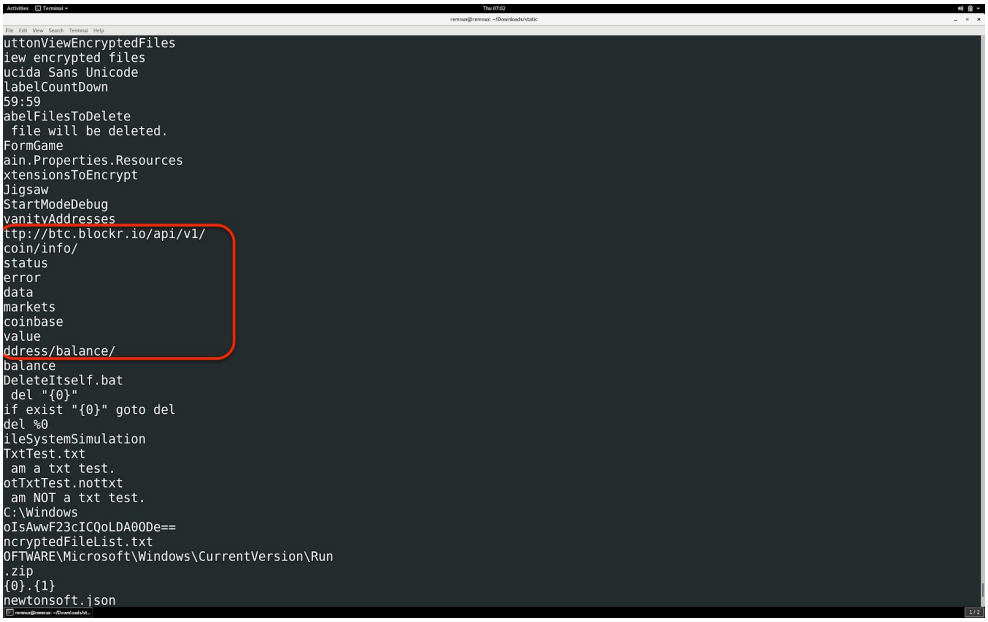
We found an alias name drpbx.exe which might be used by ransomware to hide.



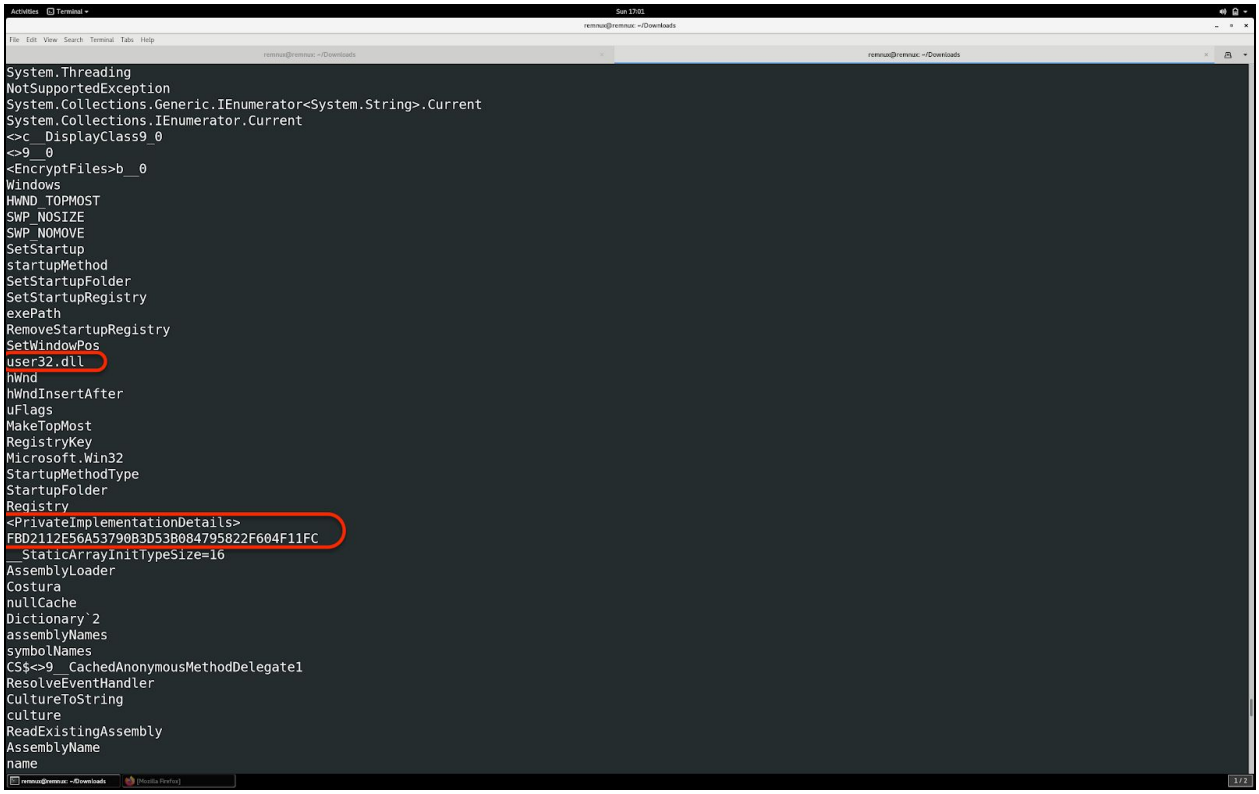
Another interesting thing is how the file has been given the name of Firefox while its original name is actually BitcoinBlackmailer.exe . Thus, we should watch out for any Firefox processes in the dynamic analysis.

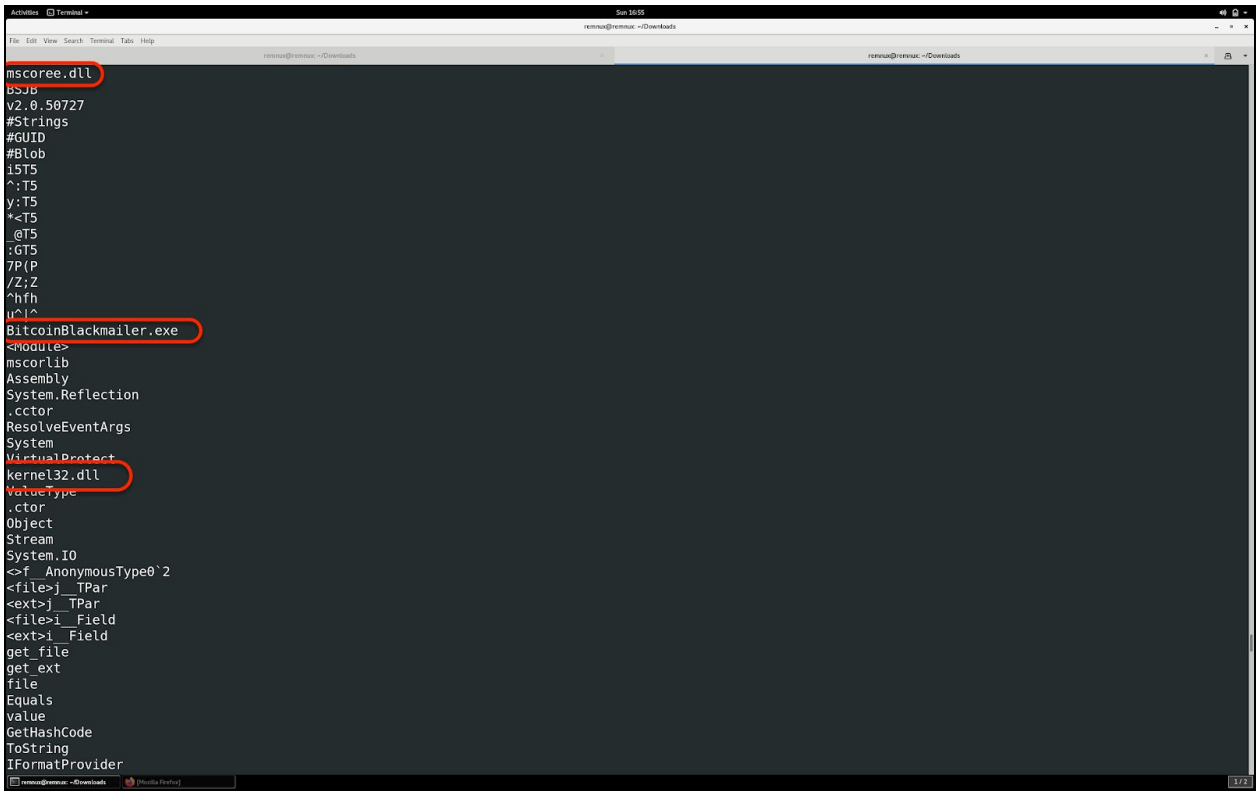


We observe the bitcoin network through which the ransom is to be paid.

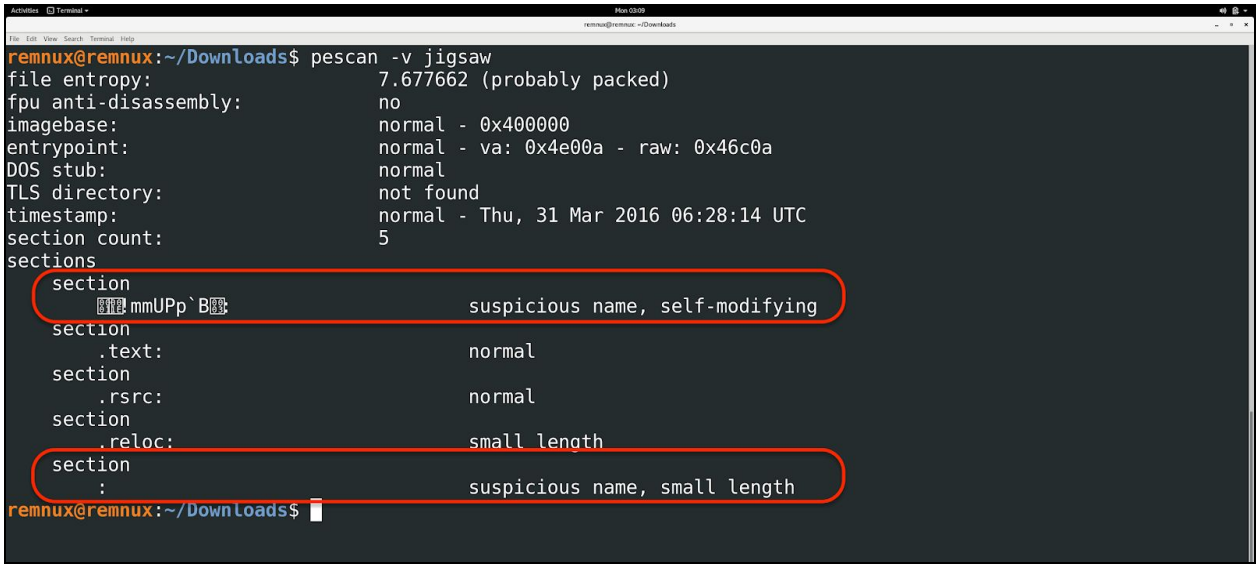


Some other interesting strings in the file are as follows :-





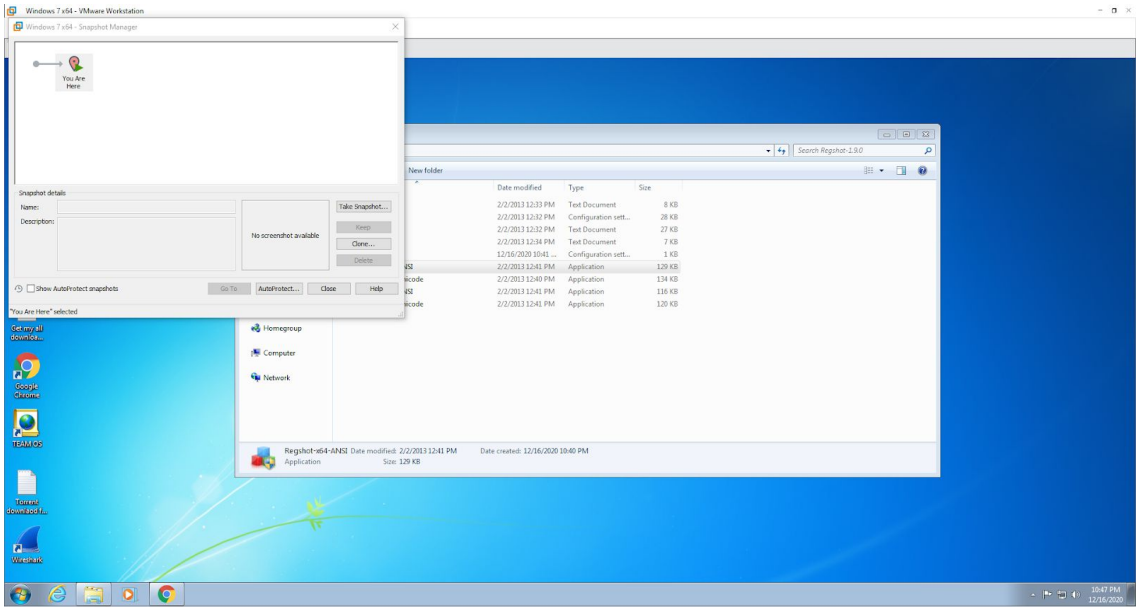
We ran pescan over jigsaw and found two suspicious sections in the file.

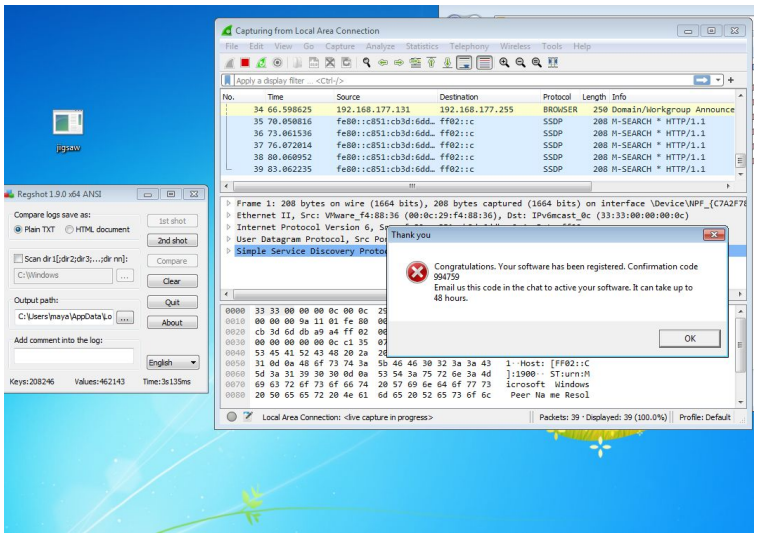


Dynamic Analysis

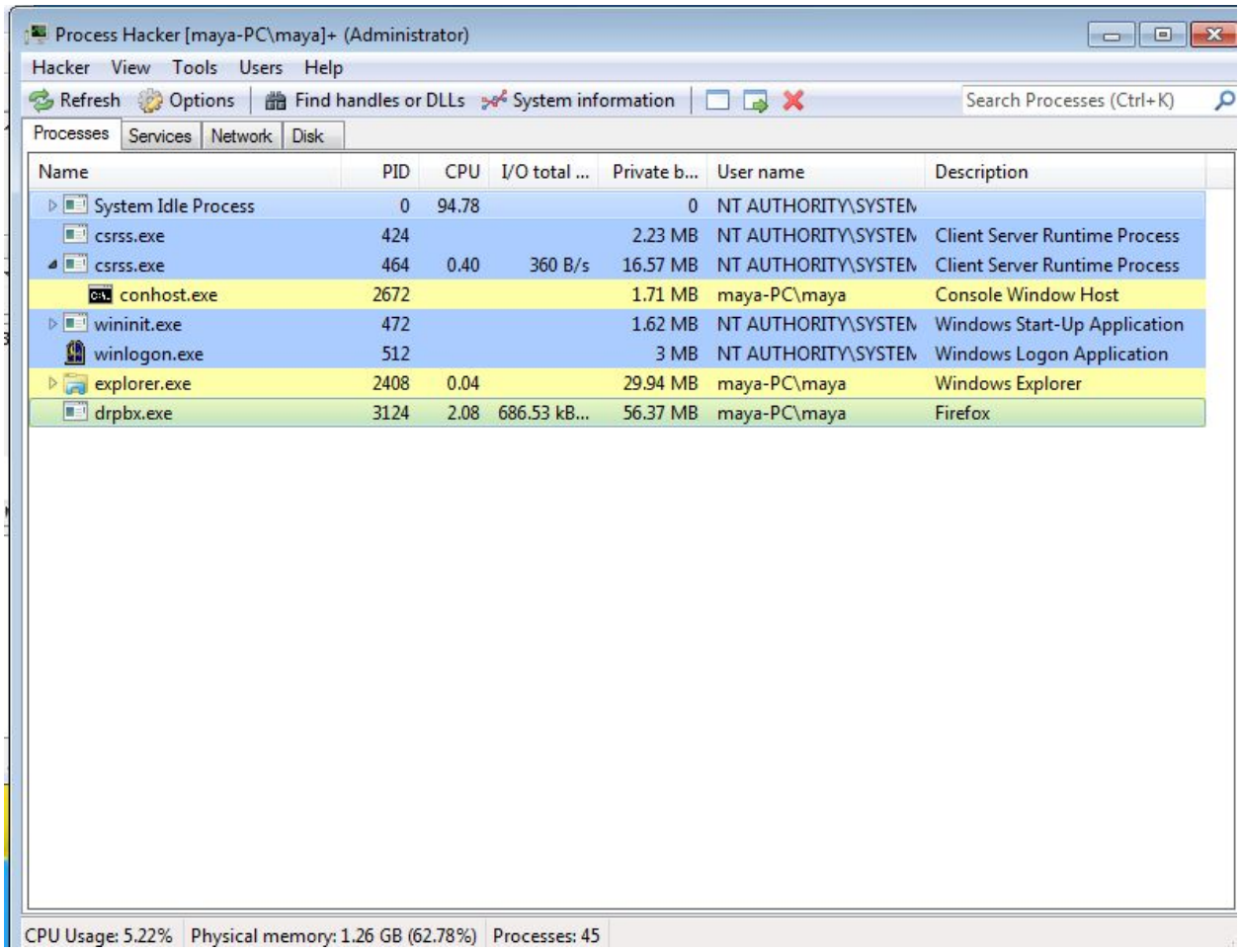
We started dynamic analysis to confirm all the things we found in static analysis and try to find if we would be able to undo encryption of the ransomware

We installed the necessary tools for dynamic analysis and created a snapshot of the virtual machine.

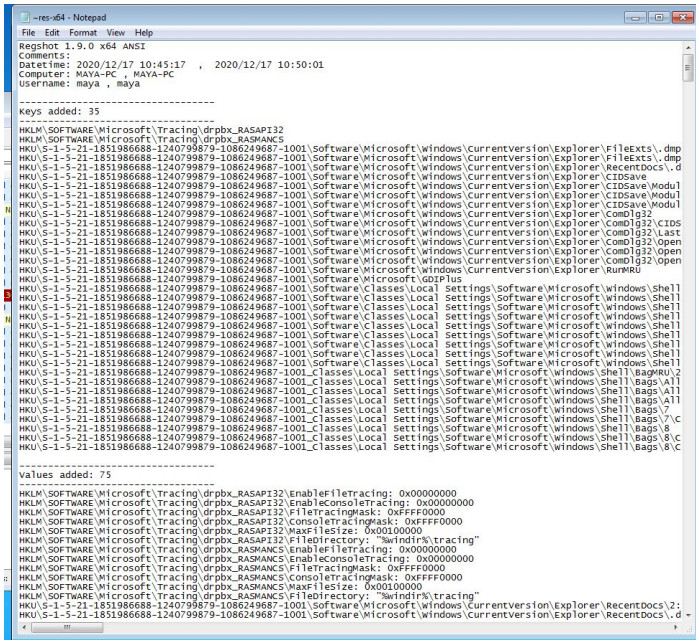




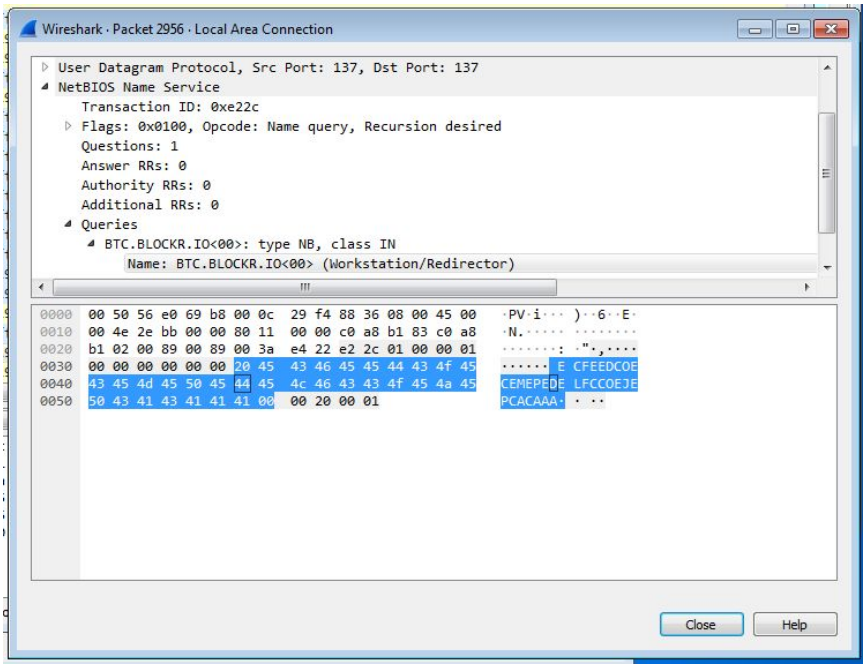
This is the activation prompt for the malware as when it starts executing.



We see a new process created with the name drpbx.exe so as to hide the process. We also see some IO interaction which may be the locking down the system.



We used regshot to see any changes in registry done by the malware. We can see it add some keys to make it automatically run if we restart the computer.



Whenever we press the button that we paid it makes a request to check the bitcoin address btc.blockr.io if it was successful or not.

KEY IDEAS OF RANSOMWARE:-

LOCKS OUR SYSTEM WITH CONVECTION CRYPTOGRAPHIC ALGORITHM IF WE GET THE KEY WE GET THE DATA BACK.

We have 2 ideas to get the key:-

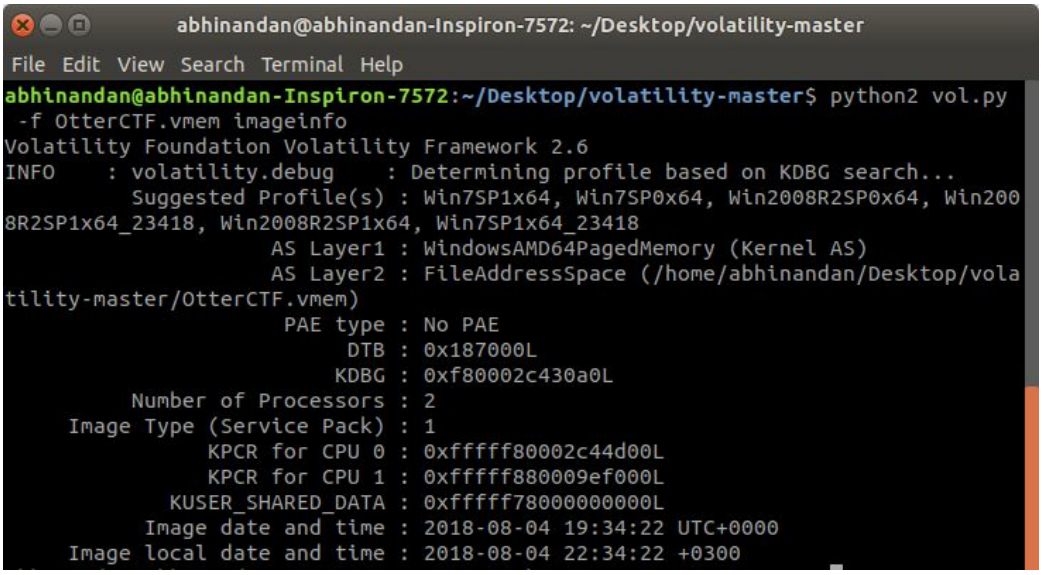
One is sent after payment is done but there is no such network traffic.

This ransomware has a private key with it so we can use a decompiler for finding the key.
Also looking into the code we can see the Ransomware is using AES.

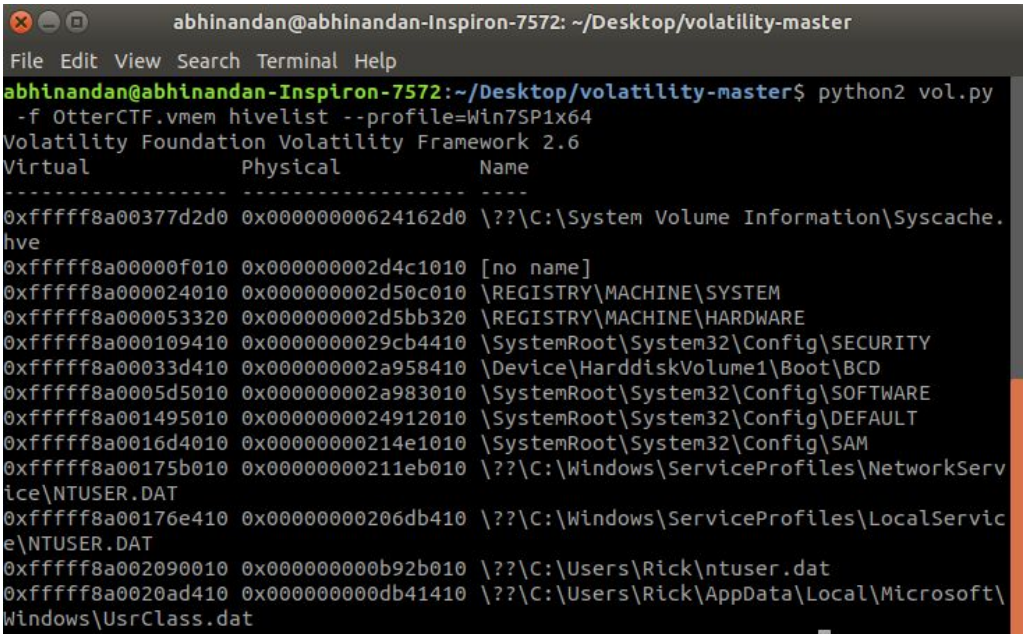
Solutions To OtterCTF

Pre-Analysis

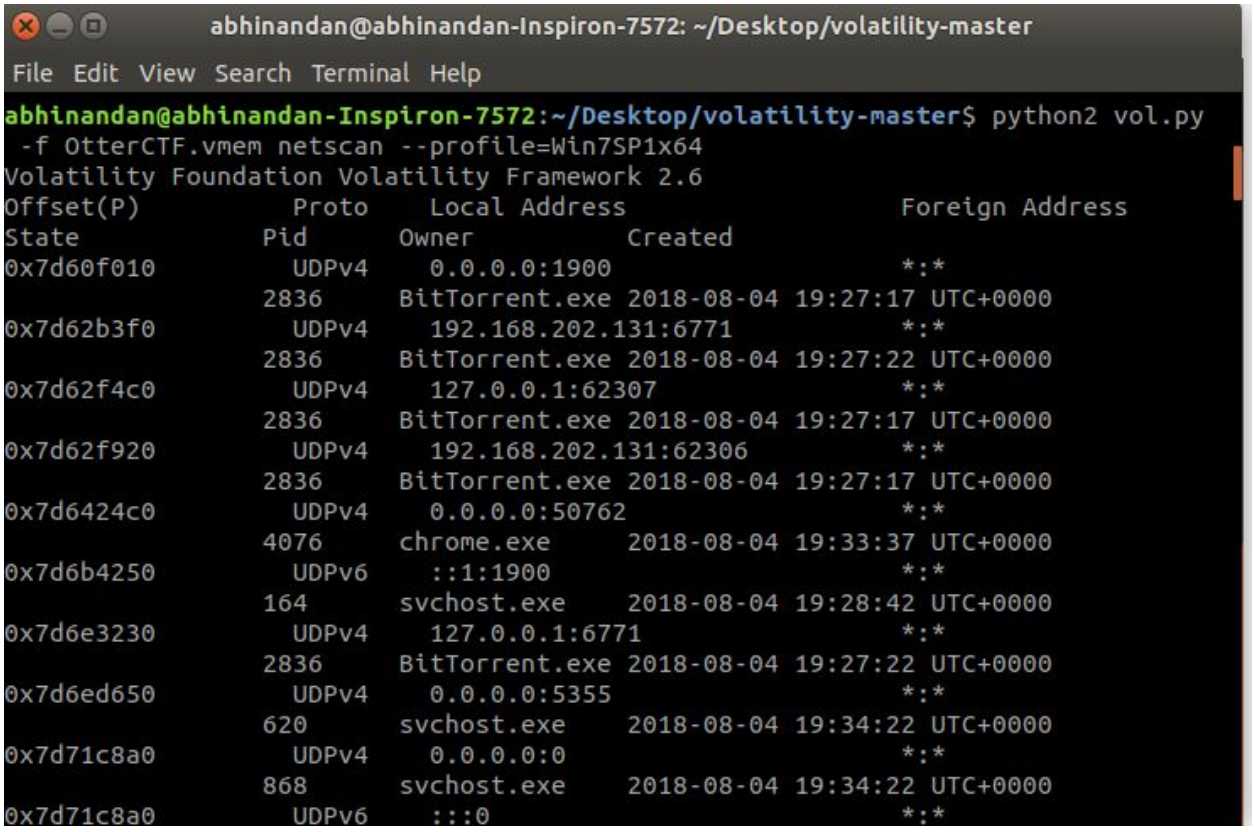
Downloading the .vmem file from OtterCTF and running Volatility Framework.



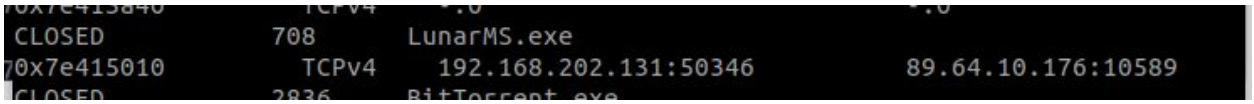
The suggested profile is Win7SP1x64. Listing the registries:



CTF Problem: Get the computer's IP address.



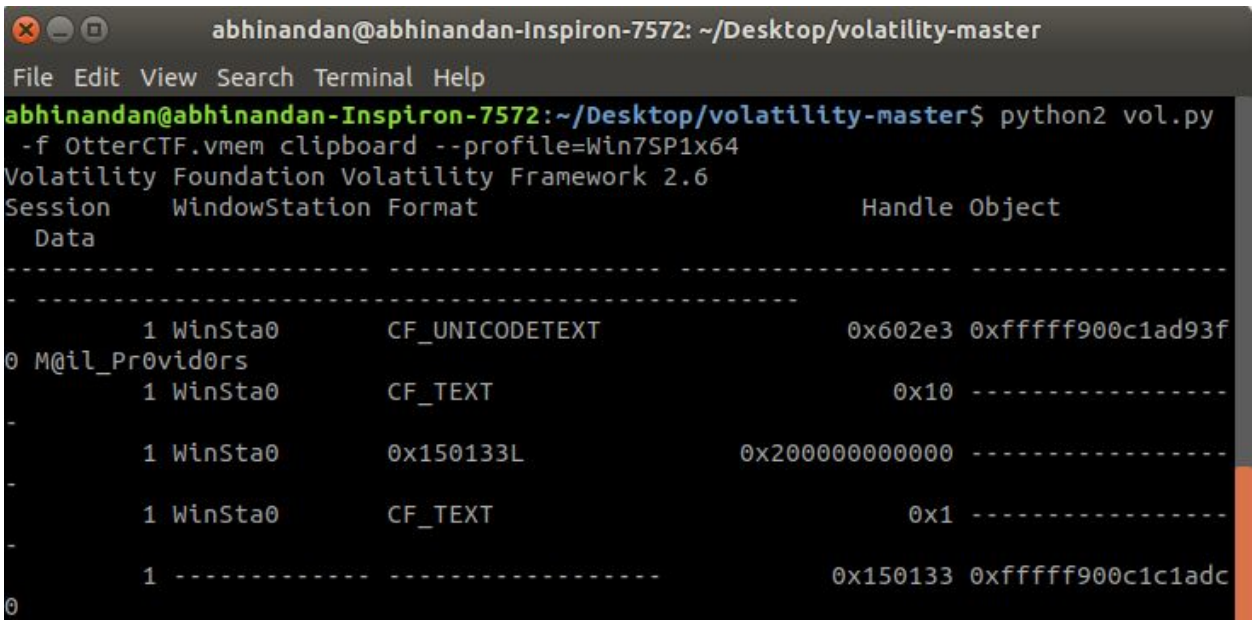
CTF Problem: Guess the name and address of the game Rick is playing?



LunarMS is the name of the game. And its address is 77.102.199.102

CTF Problem: Since Rick pastes the password from the clipboard, we have to get the clipboard items.

Password is M@il_Pr0vid0rs



CTF Problem: finding suspicious processes

Vmware-tray.exe is a child process of Rick and Morty, which is very suspicious.