Project

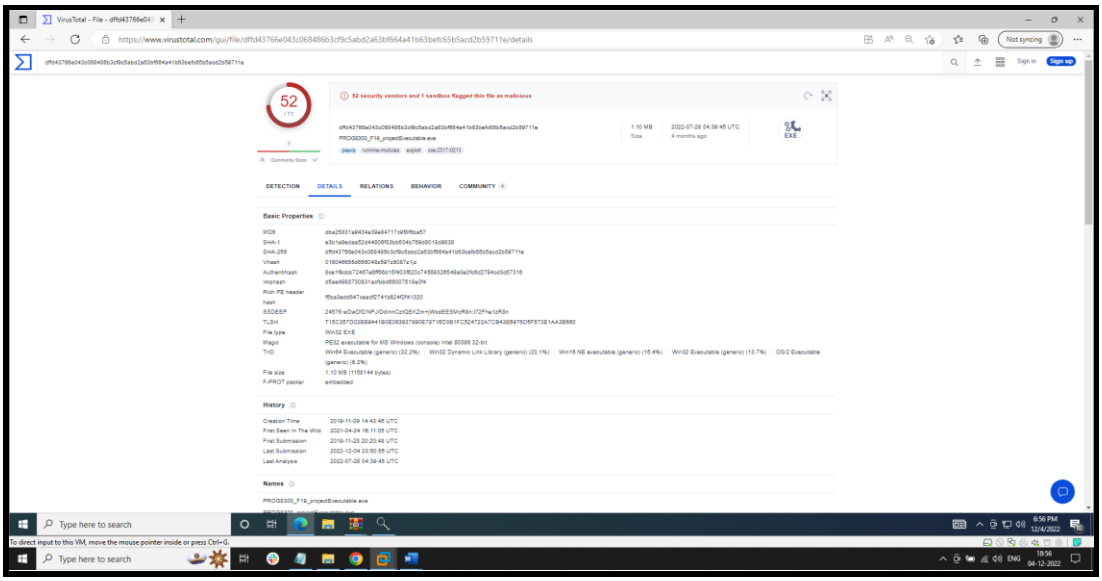
Mayank Rahalkar

Table of Contents

Static Analysis

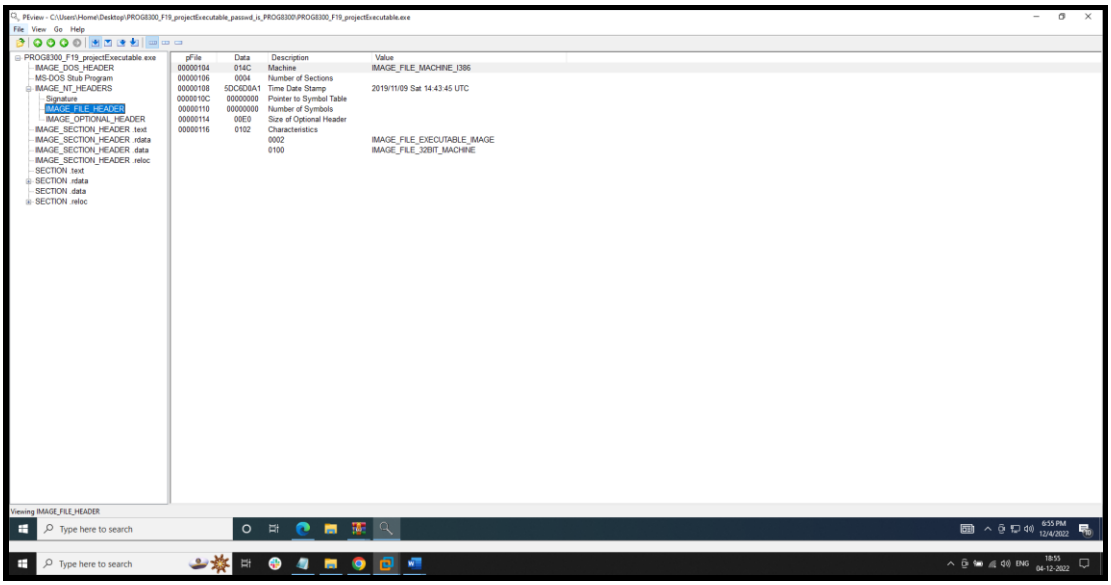
Virus Total

Firstly, I used the virus total website to scan the given executable. Virus total has a list of security vendors which scan the target. Upon scanning the target in Virus Total, it showed that 52/71 security vendors were able to detect it.



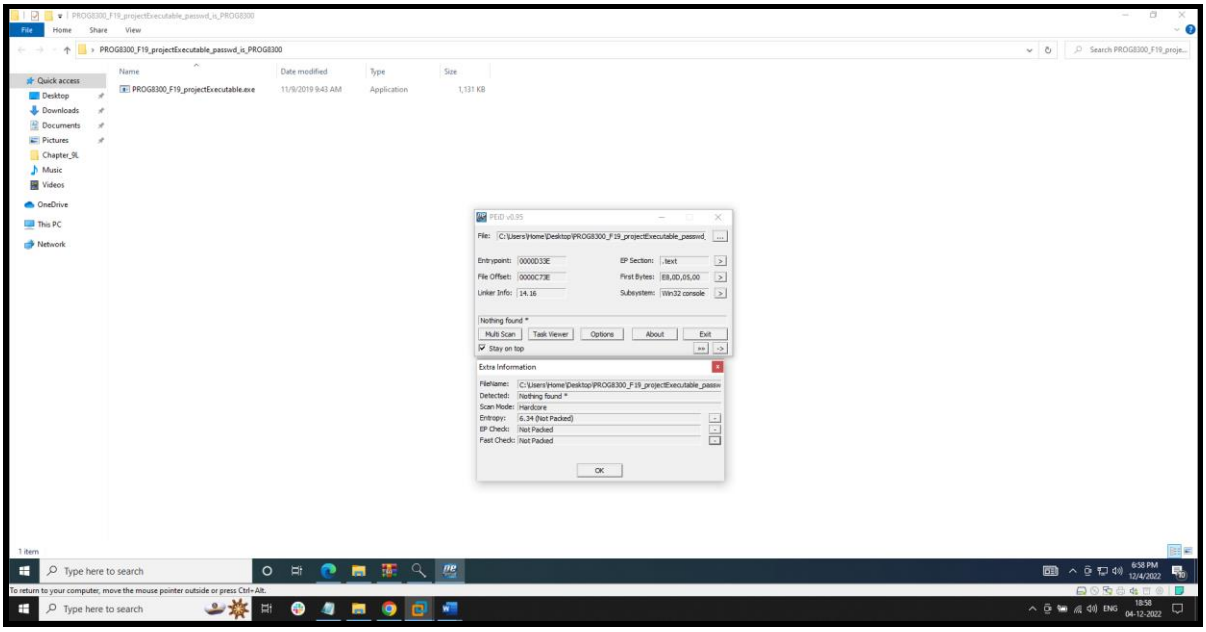
PEView

Then I used the tool called as PEView to check for the compilation date of the binary. The tool showed that the malicious binary was compiled on 2019/11/09.

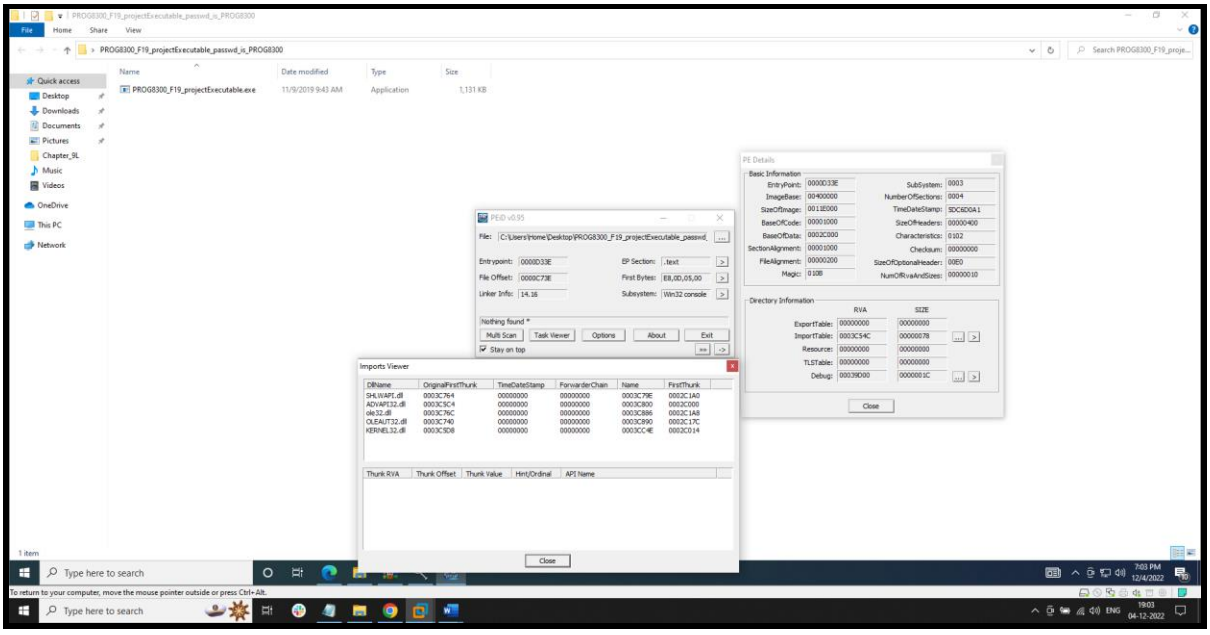


PEiD

Once I knew the compilation time of the binary, I checked if there are any packing programs used to pack the binary. However, the binary was not packed.

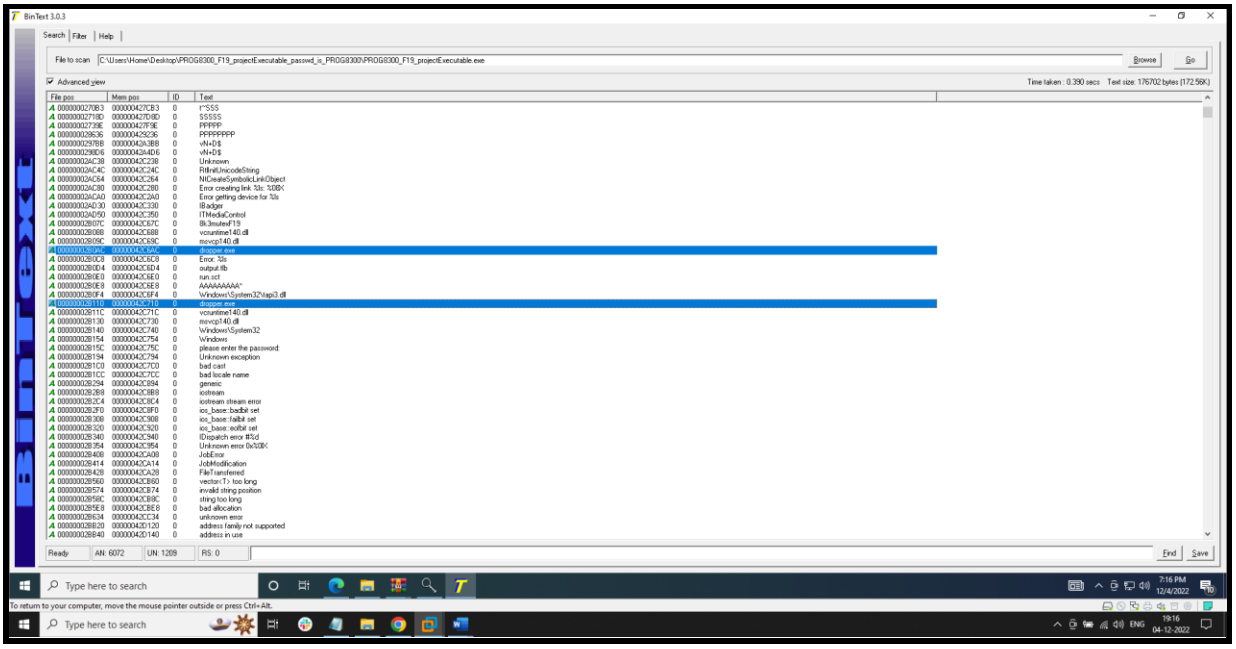


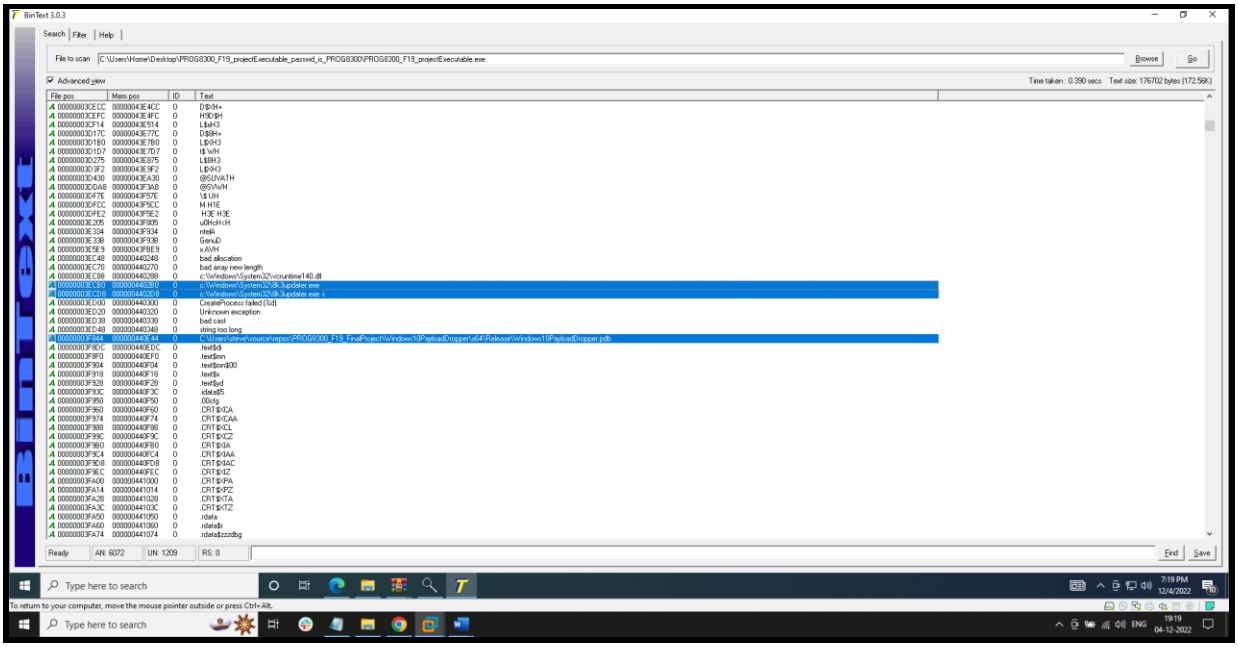
The tool even showed the imports that were used in the binary.



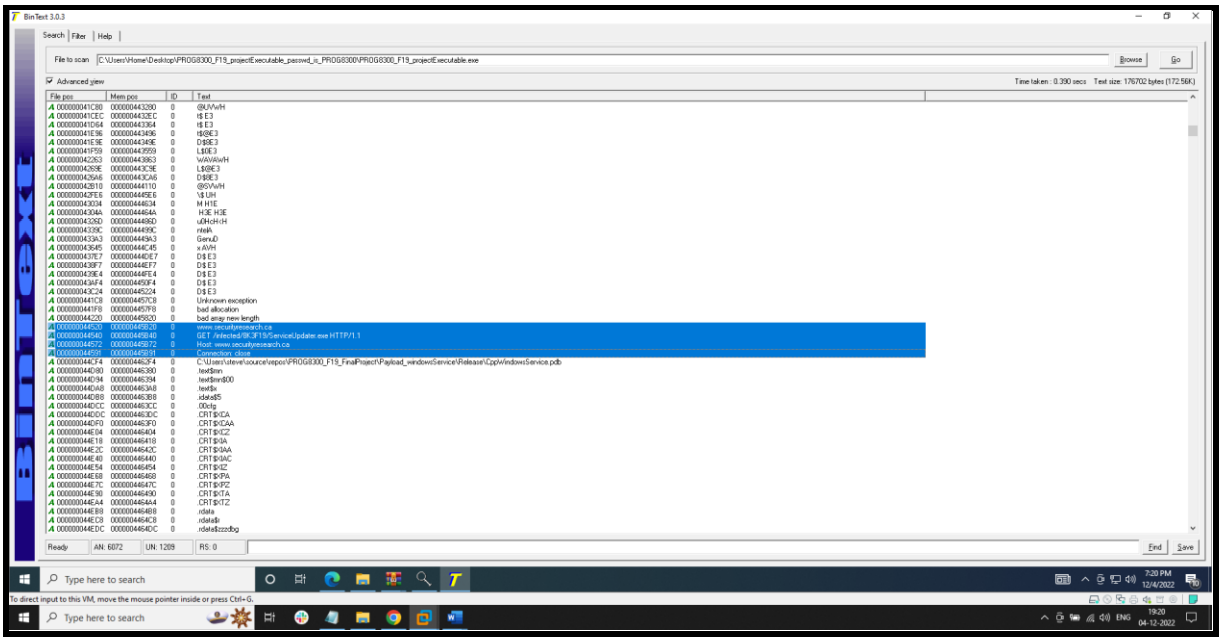
BinText

I then used the tool called as BinText to view the human readable strings in the binary. This tool is like “strings.exe” or “floss.exe”. It showed me 3 host-based indicators. One of them was “dropper.exe” file. The 2nd one was the 8k3updater.exe and the 3rd one was “Windows10PayloadDropper.pdb”.



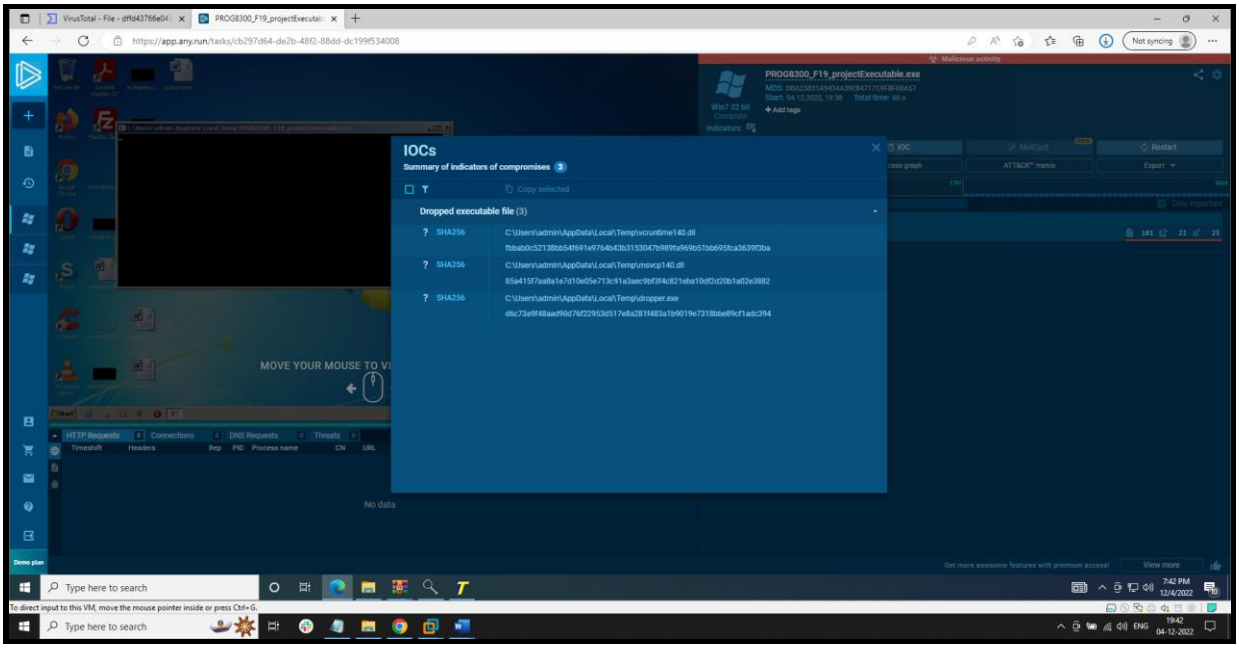


I then checked the file for network-based indicators. It showed that it making a GET request to /infected/8K3F19/ServiceUpdater.exe which was present on the host “www.securityresearch.ca”.

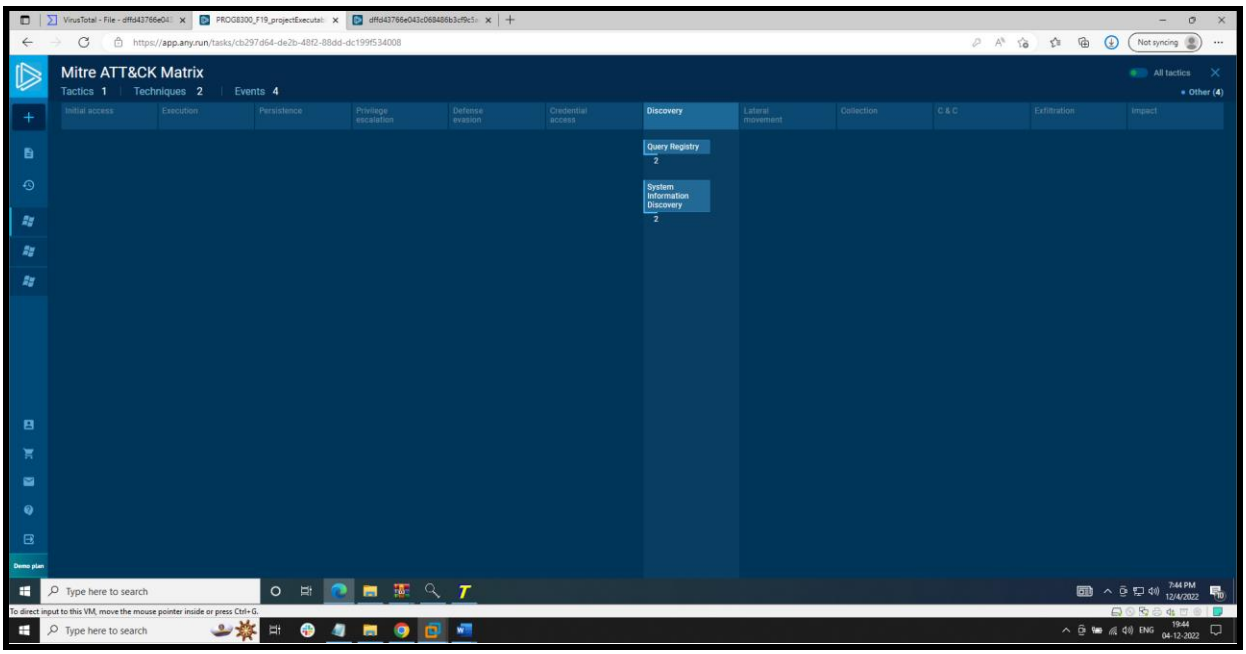


AnyRun

I then used the website called as Any.Run to check the executable. Upon uploading the executable, it showed me some Indicators of Compromise (IOCs). It showed 3 more executable files that are been dropped.



Looking at the MITRE section on the website, it showed the following in the Discovery Tab.



According to MITRE, the Query Registry means the following:

Query Registry

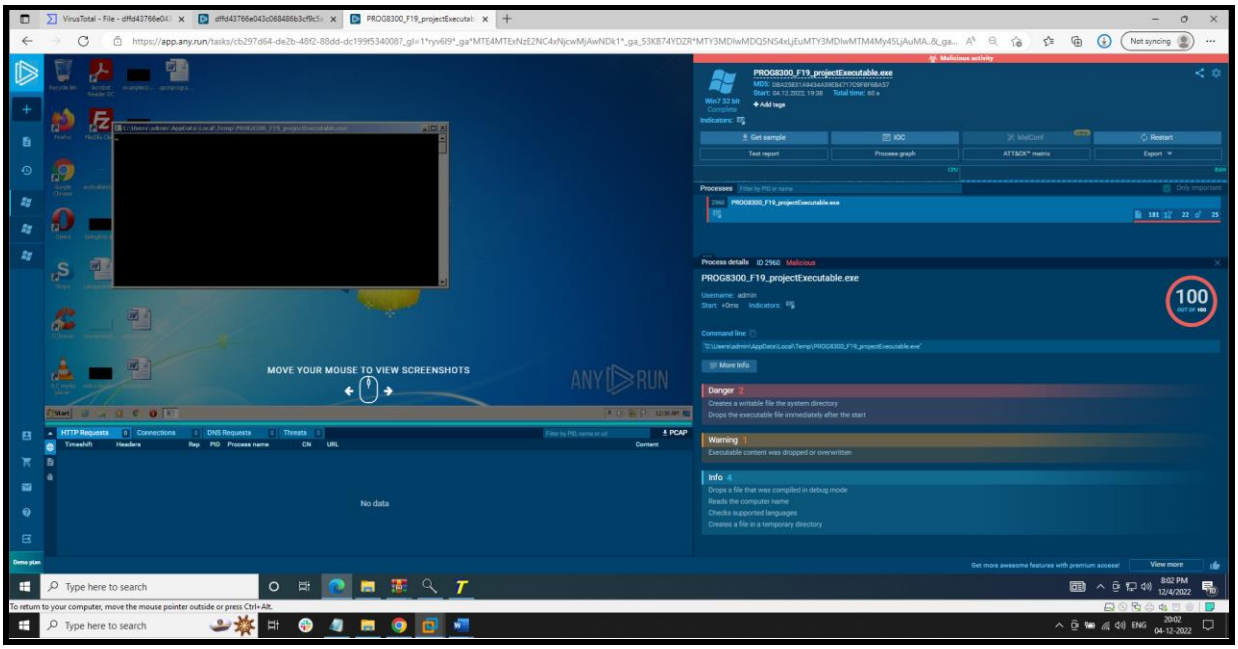
Adversaries may interact with the Windows Registry to gather information about the system, configuration, and installed software.

According to MITRE, the System Information Discovery

System Information Discovery

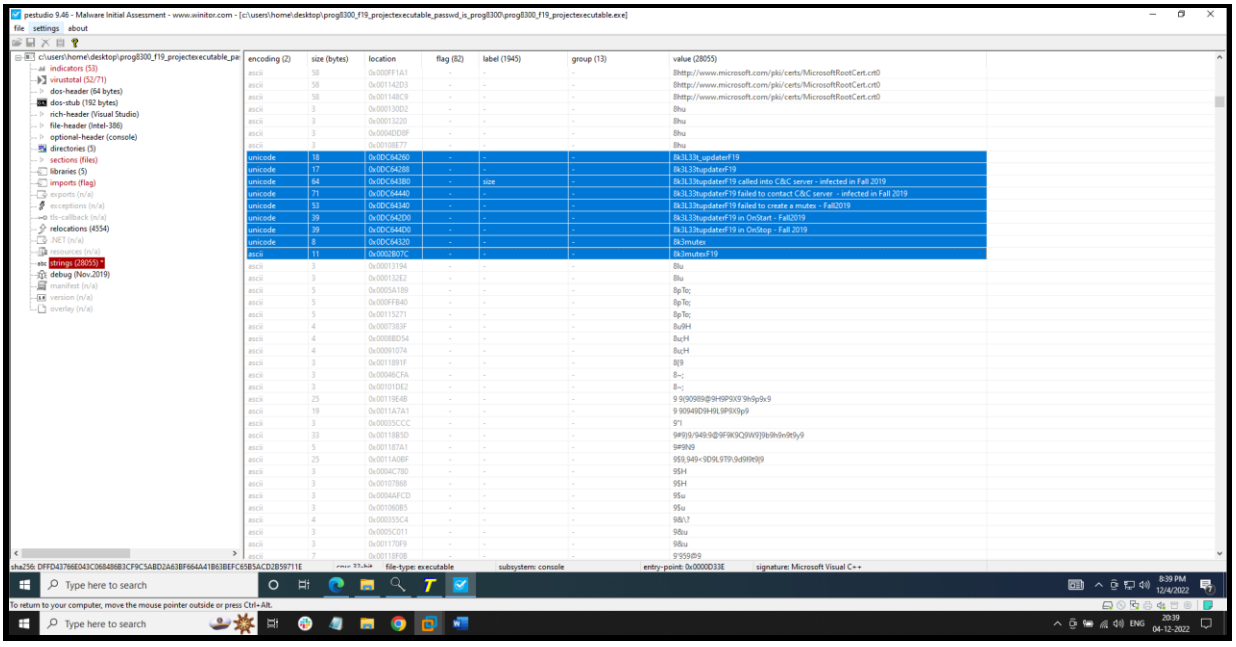
An adversary may attempt to get detailed information about the operating system and hardware, including version, patches, hotfixes, service packs, and architecture. Adversaries may use the information from System Information Discovery during automated discovery to shape follow-on behaviors, including whether or not the adversary fully infects the target and/or attempts specific actions.

Further exploring the website, I was able to find a section which showed the activity of the binary. It states that it creates a writeable file in the system directory and then drops a executable in the directory after the start.

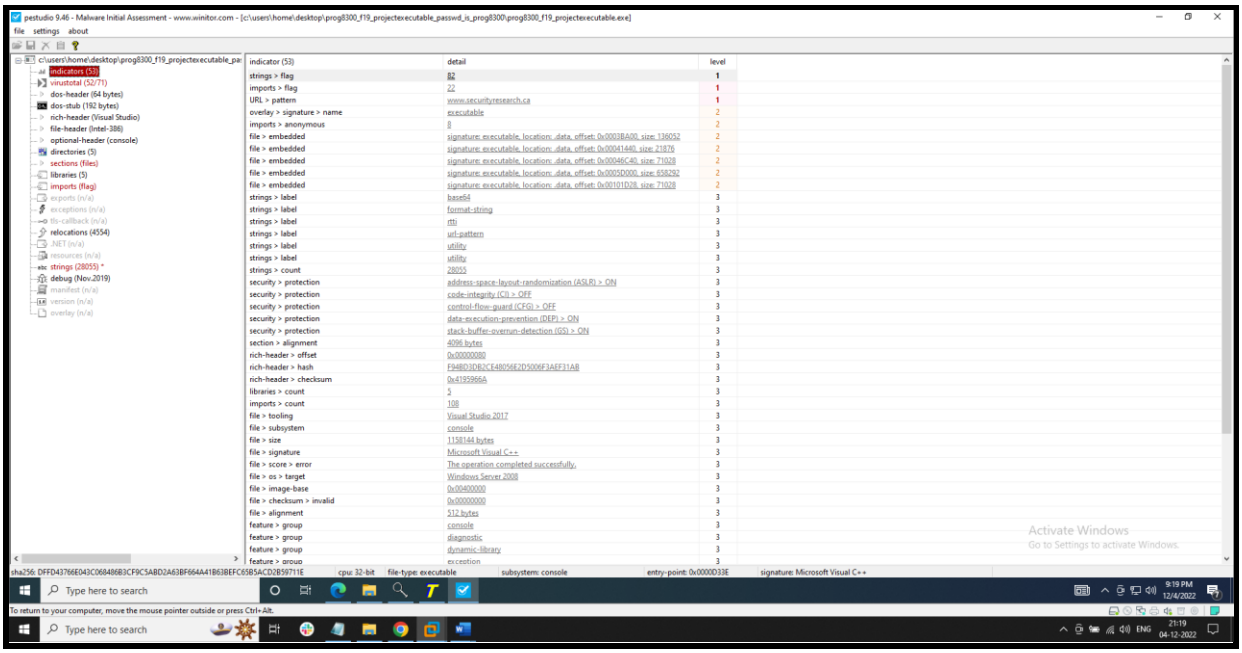


PEStudio

PEStudio was then used to analyze the binary. It can also be used to perform reverse engineering. Upon inspecting the strings, it showed that the binary is trying to contact the Command and Control server.

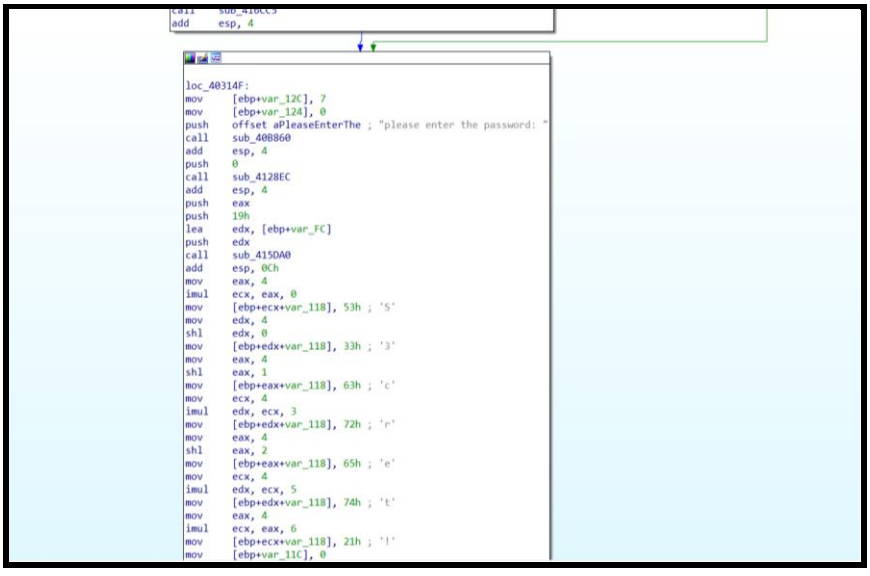


Further inspecting the “indicators” section, it shows a reference made to the website we found earlier. It also shows that the code was compiled on Visual Studio 2017 and is a C++ code. It also shows that the executable was made to run on Windows Server 2008.



IDA

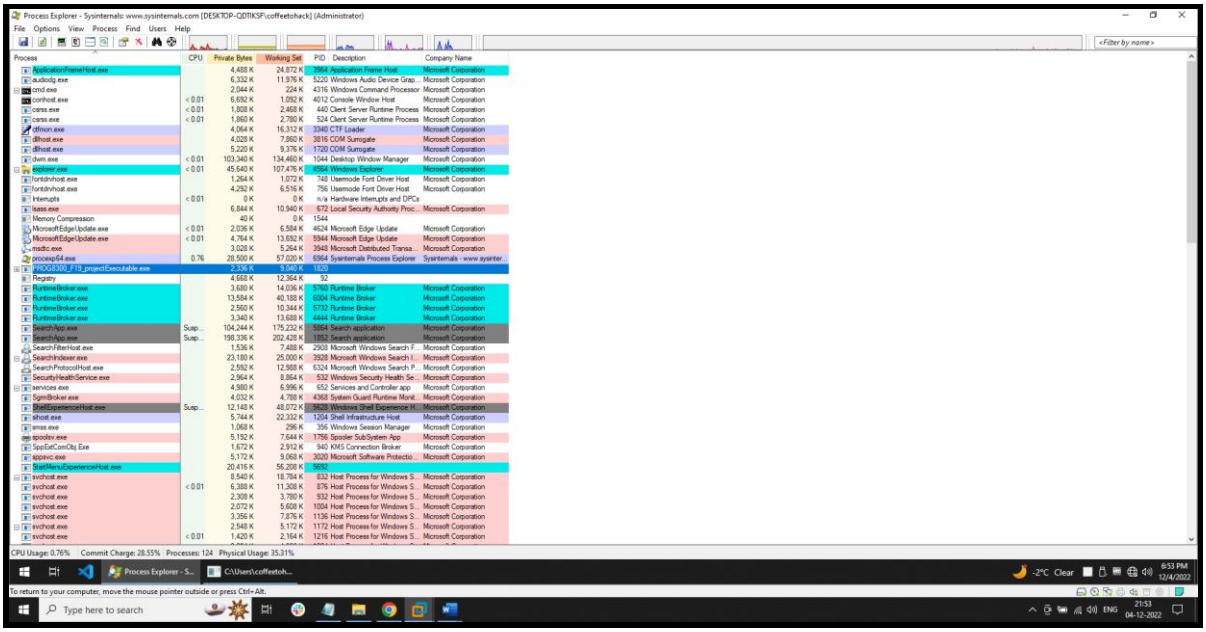
Disassembling the application in IDA, it showed that the password for the file was: S3cret!



Dynamic Analysis

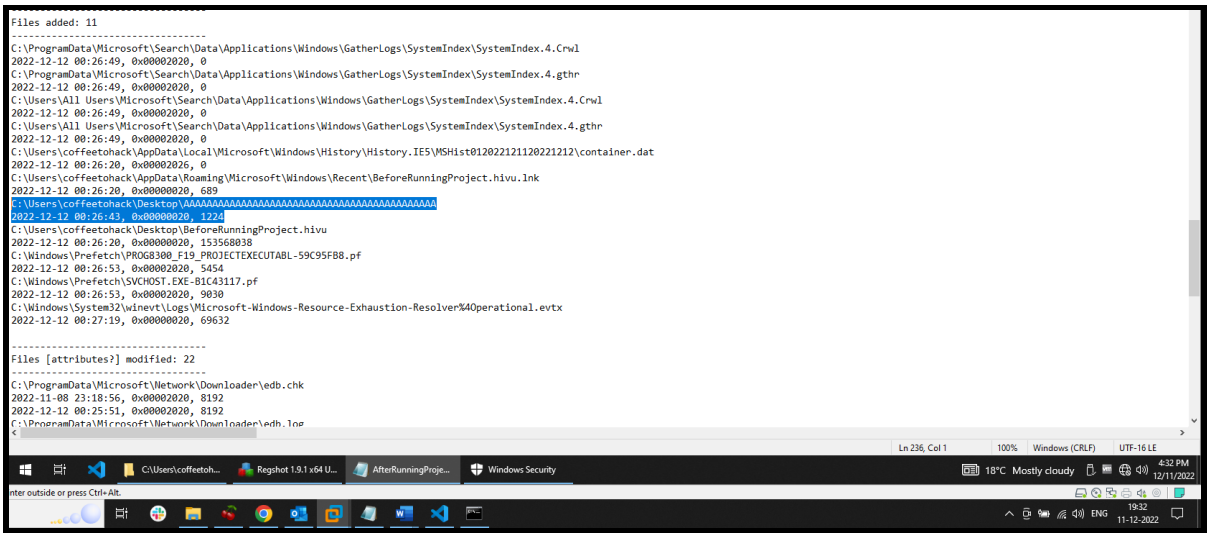
Process Explorer

For dynamic analysis I used process explorer to check for processes it is creating. I did not find any interesting processes with this. It only showed the executable.

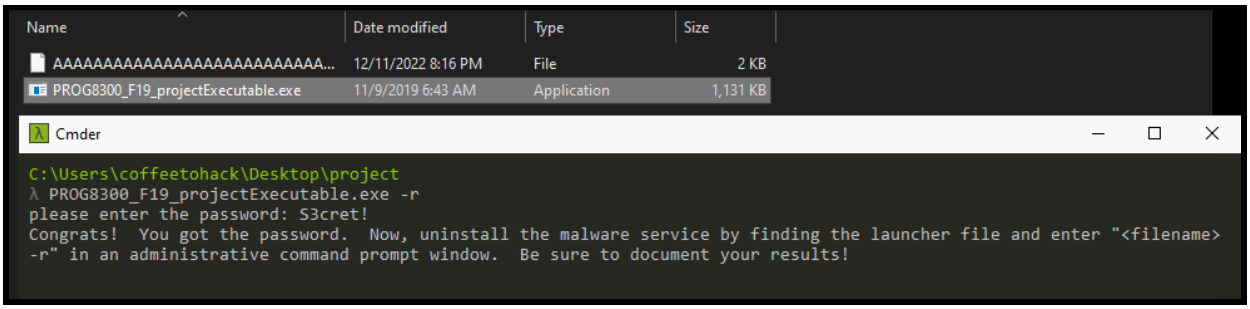


RegShot

I then used a tool called as regshot. I took a shot before running the binary and after running the binary. It showed me that a file named “AAAAAAAA…” is created in the same directory in which the malware resides.



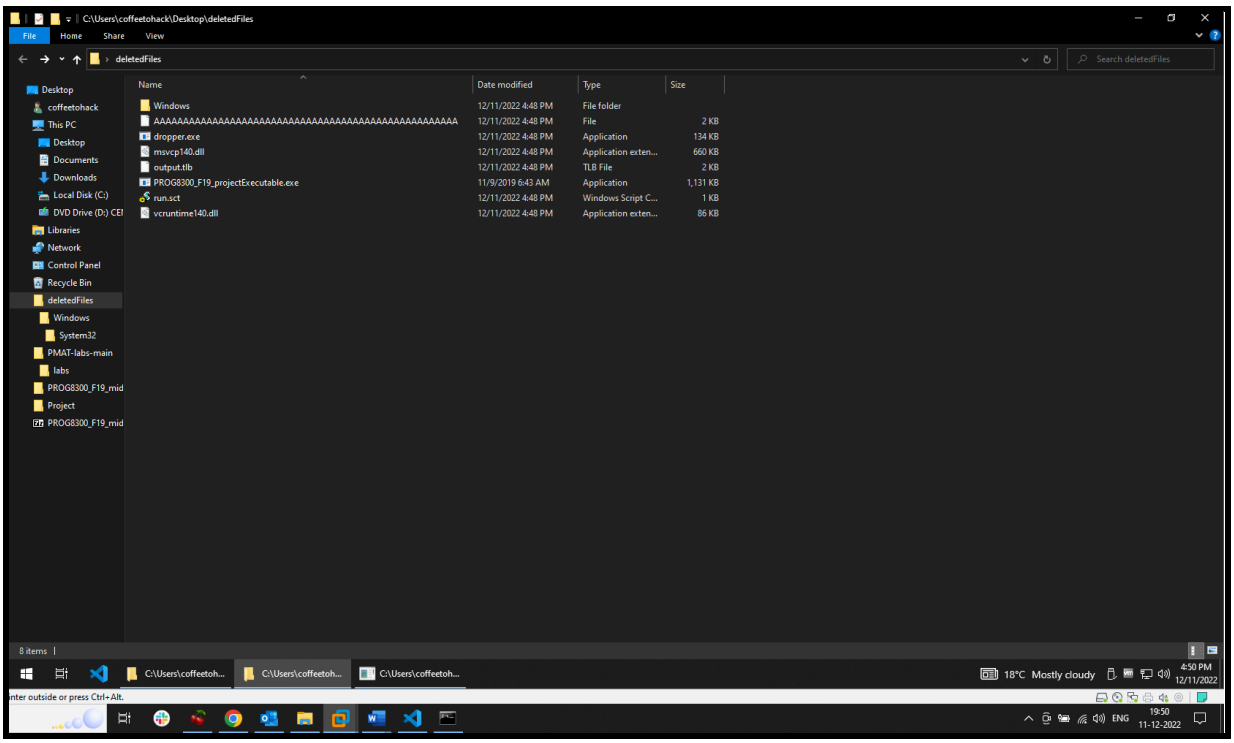
After running the binary with an administrative command prompt and with the “-r” flag, it asked me to enter the password. I entered the password that I found using IDA earlier. And I got the following output. It says that the password was cracked and I can uninstall the malware.



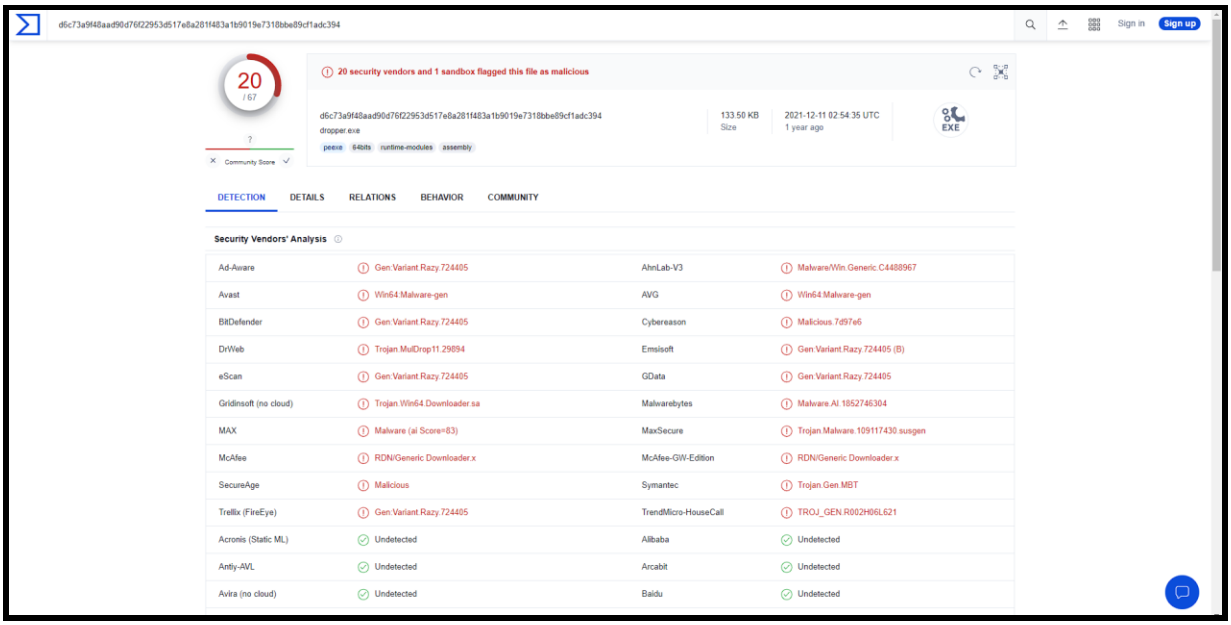
Online Analysis

During the static analysis phase, I had noted a “dropper.exe” file which was been created. However, the regshot output didn’t show any such file. So, I decided to run the binary again. Upon inspecting closely, I saw few more files been created but they got deleted immediately. I’m not sure if they got deleted or got moved to some other place. So, I ran the executable again and decided to copy the files that were created to a new location. However, this time there were no files that were created. This meant that the binary only generates these files when it is run for the first time. So, I reverted my VM to the earlier snapshot and decided to perform the above-mentioned task again. This time I was successful in copying the files to a new directory.

It showed the following list of files that were created.

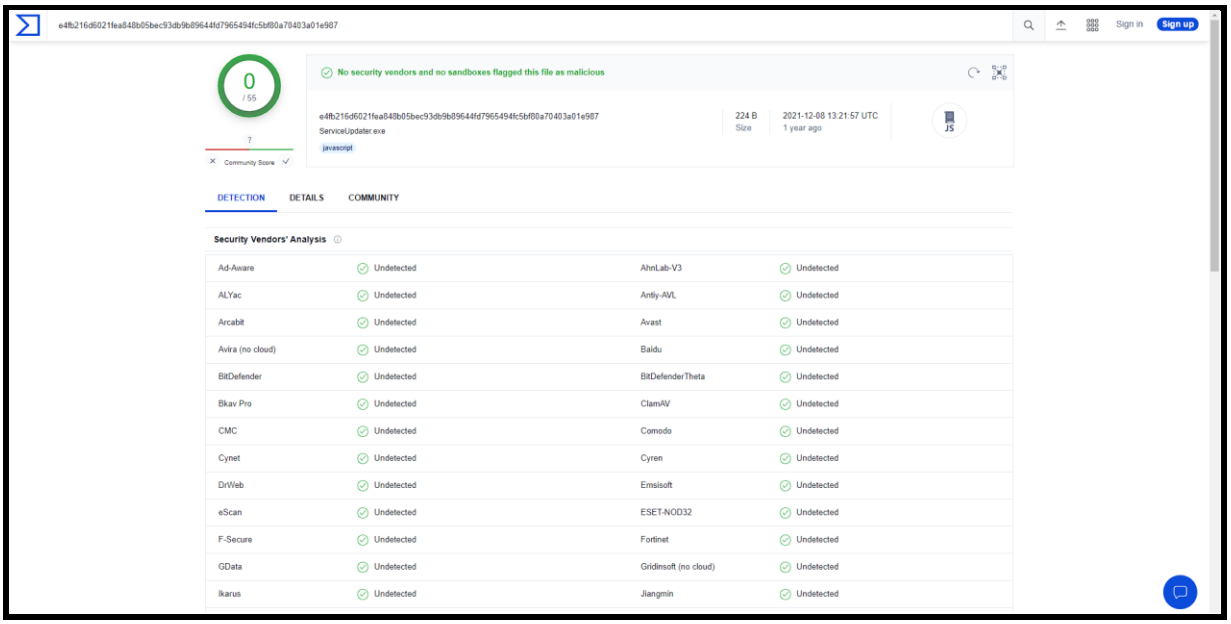


I found that the “dropper.exe” file was created in the directory. I uploaded that file on Virus Total to scan it. It was detected by 20/67 security engines.

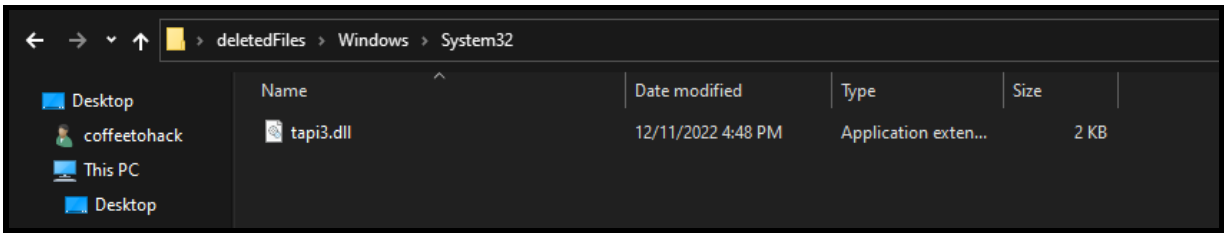


Then I accessed the URL that I had found earlier. I found that “ServiceUpdater.exe” file that was referenced before in the static analysis part.

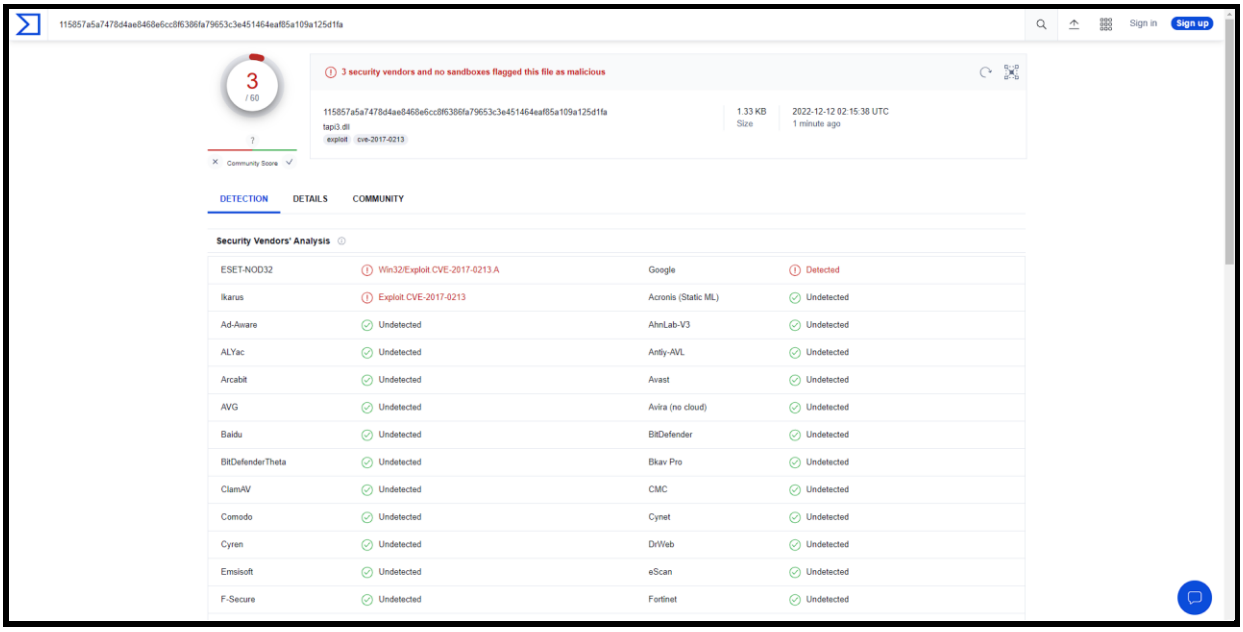
I downloaded the file and uploaded on Virus Total to check if it is malicious. However, the file was clean.



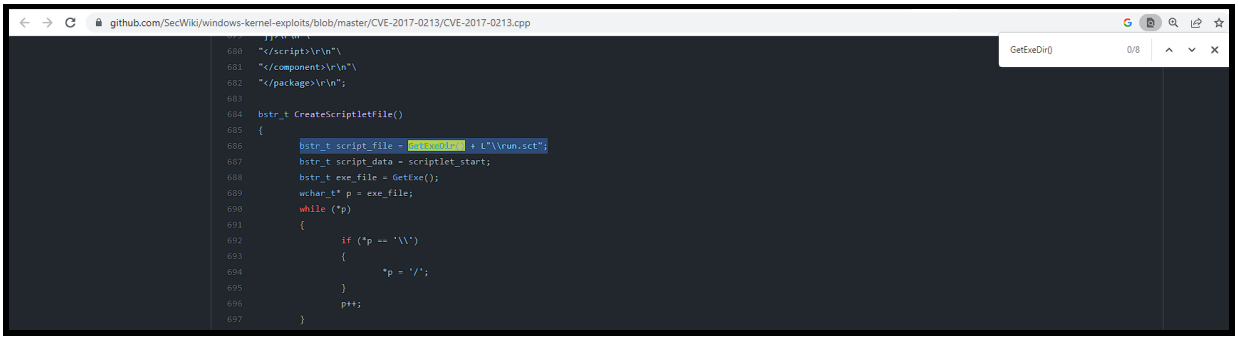
Looking back to the files that were created, I found a “tapi3.dll” file in the “Windows\System32” directory that was also created when the binary was run.



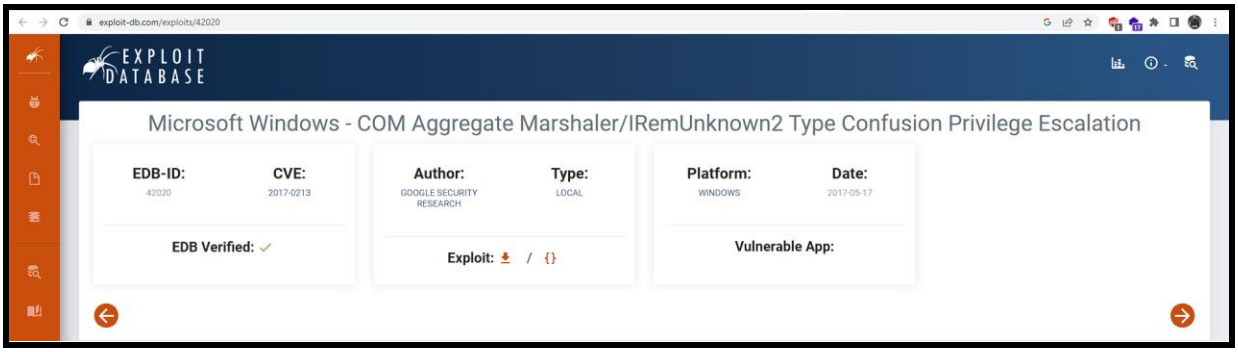
I uploaded this DLL file on Virus Total. It showed that 3/60 security engines were able to detect the file. Looking closely at the detections, I saw that it says CVE-2017-0213.



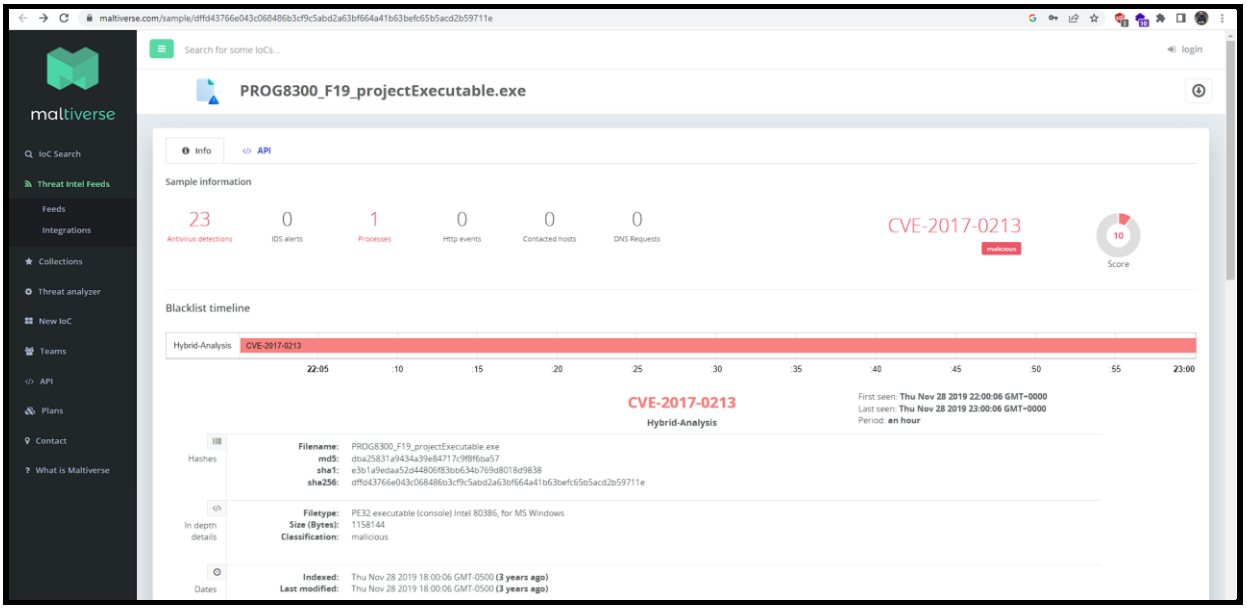
I searched for that CVE and found a C++ exploit code. This code showed me the “output.tlb”, “tapi3.dll”, “run.sct” files. This made my assumption stronger that the file might actually be the CVE that I discovered above.



The following was the same code hosted at ExploitDB.



I then randomly checked the signature of the main project file that was provided to us in Maltiverse website. Even this website showed the same CVE that I discovered earlier.



Description of Malware

Based on the research I have made above; I have concluded that the binary is “CVE-2017-0213”. During the research I discovered that the binary creates multiple files and deletes some of them. However, I was still able to save those files in another directory. The above-mentioned CVE is titled “Microsoft Windows - COM Aggregate Marshaler/IRemUnknown2 Type Confusion Privilege Escalation”. Privilege Escalation is the process in which an attacker tries to gain elevated access in the system. This access can be an admin access or a system access in a Windows environment. So according to me the files that disappear are replaced in the system directory. I was unable to get any GET request made to the “securityresearch.ca” website that I had discovered earlier even in Wireshark. If the vulnerability gets exploited, an attacker can run commands with elevated privileges. This vulnerability needs to be exploited with another vulnerability.