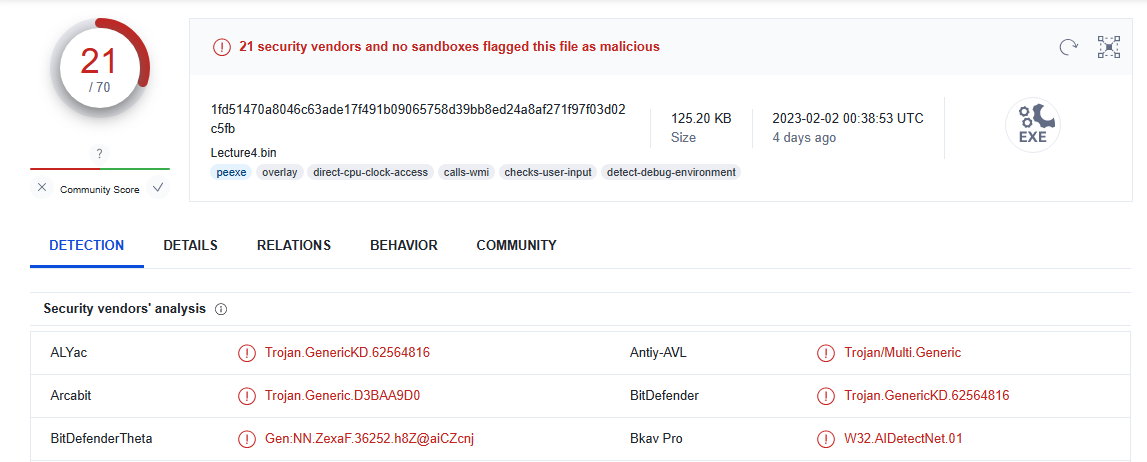
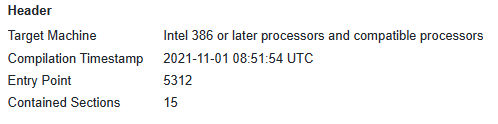
**VIRUSTOTAL**

Signatures: Matches 21 of 70 existing antivirus signatures.

MD5 Hash: f028e27be2a50a8ea83a31c6dd773dab

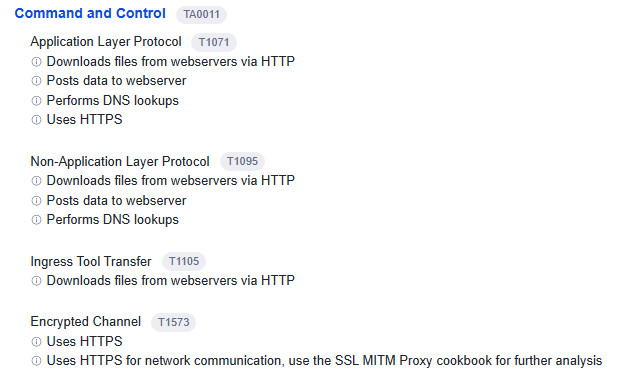


**Compilation Timestamp**: 01 Nov 2021 at 08:51:54 UTC



**What is the malware known for?**

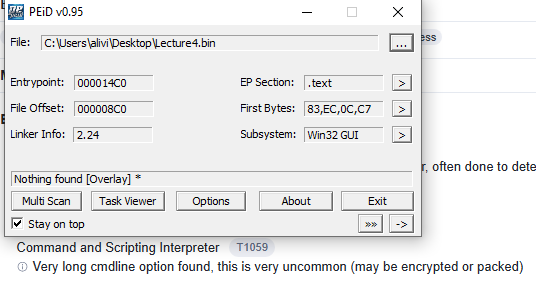
VirusTotal has identified this malware as a generic trojan, making this simple upload ineffective in identifying an infamous malware that has a notorious history. VirusTotal describes many behaviors this malware may perform, such as privilege escalation, defense evasion, input capture, and potentially network connections over HTTP, HTTPS, and DNS (ports 80, 443, and 53 respectively). The command and control portion suggests that the malware downloads files and potentially posts data to a web server.



**STATIC ANALYSIS**

**Packing/Obfuscation**

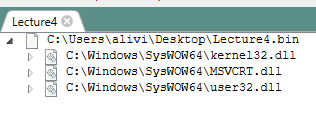
PEiD did not detect any indicators of packing. However, VirusTotal said that the detection of a long cmdline option suggests encryption or packing.



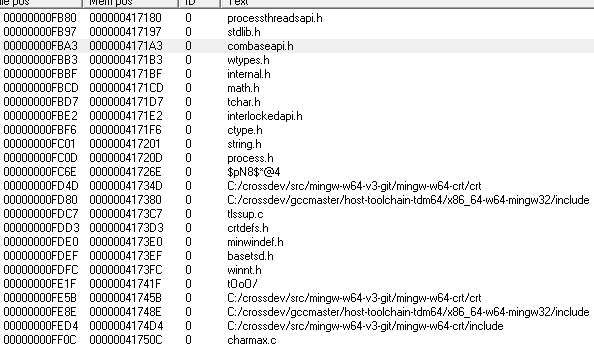
A further investigation in PEview as well as BinText showed that there were plenty of strings available for analysis. Although the excessive amount of strings found in the file result in a lengthy analysis, they also allow a detailed investigation into exactly what the file could do. Therefore, the malware is neither packed nor obfuscated.

**Imports**

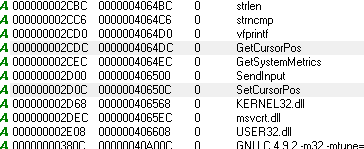
The dll imports of this malware are kernel32, MSVCRT, and user32. These imports suggest the malware accesses the kernel in some way using kernel32 and potentially alters the user interface in some way using user32. MSVCRT.dll is the “C standard library for Visual C++ compiler” (Wikipedia). This suggests that the malware will compile and then execute its code which was written in C++.



Additionally, there are lots of calls to files from or to the C:/crossdev folder. This folder is used for C++ code compiling, confirming that the malware was written in C++. Additionally, there are calls to the multiple header files, likely library imports that the C++ malware uses to execute its code.

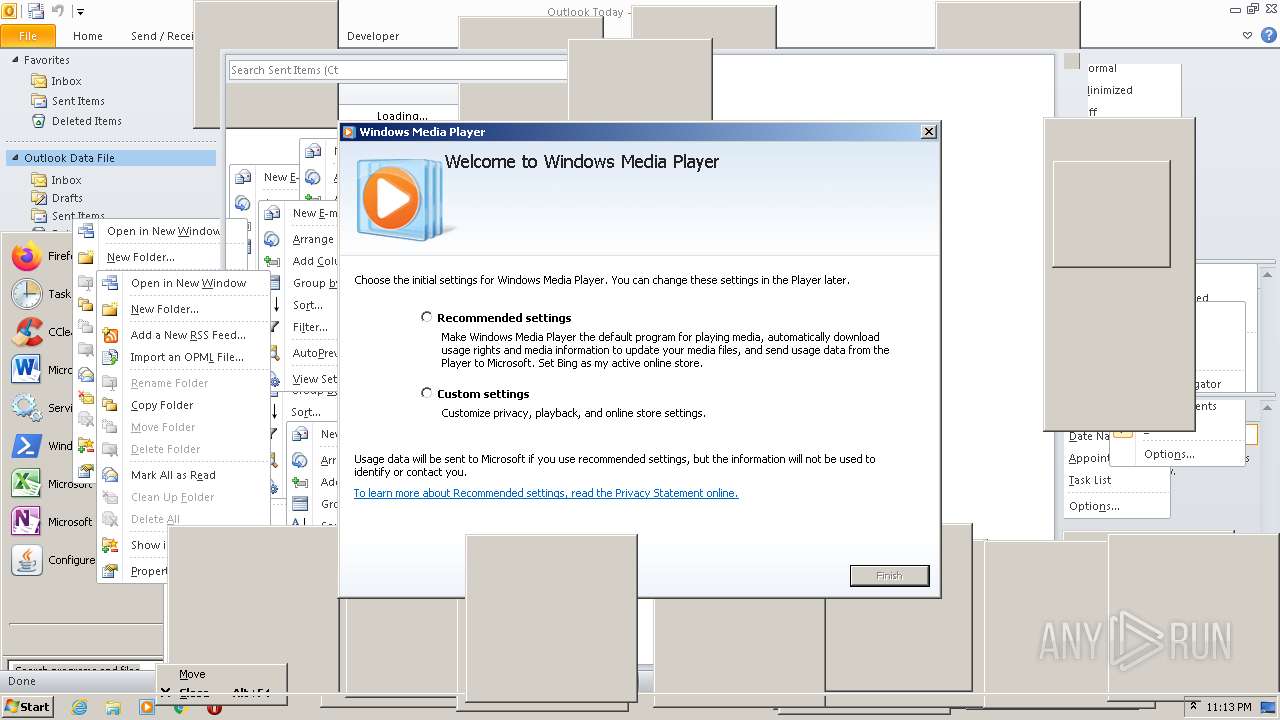


Additionally, there are some imports that indicate cursor position data gathering and position setting. It is possible that malware locates the cursor position and possibly locks the cursor on a specific position on the screen. This could have the user inadvertently click on a malicious link that the malware supplies. This would make sense because VirusTotal stated that there potentially are HTTP, HTTPS, and DNS calls by the malware.

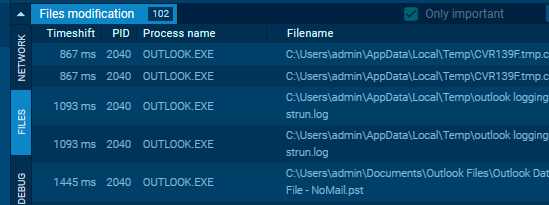


**Other files and host-based indicators**

After uploading the malware to app.any.run, it would be immediately clear what indicators there would be on an infected system. There were multiple spontaneous openings of outlook, windows media player, and calculator.



There were also 102 file modifications within Outlook.exe, internet explorer, opera web browser, and windows media player. Sorting through this file list and finding any new files that do not belong would be good file-based indicators of an infected system. If a host system does not have Windows Media player installed deliberately and it gets installed spontaneously would be a good file-based indicator of infection.



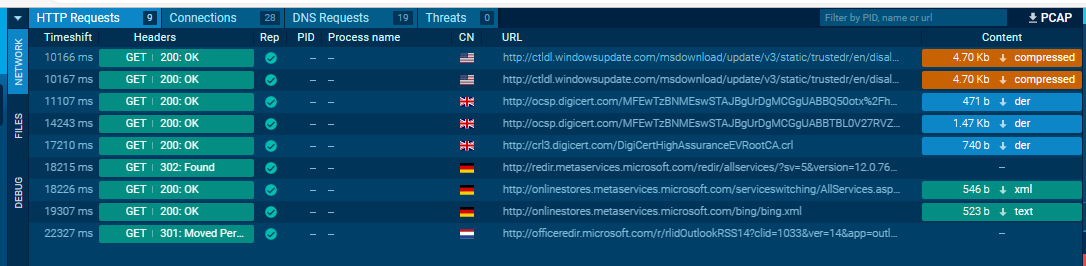
**Would I consider this file malware?**

Absolutely. The preliminary running of the malware within app.any.run demonstrates a clear indication that this is file is malware. Since the strings analysis do not specifically make calls to internet explorer, opera, or Windows Media Player, it appears that it primarily executes using the cursor position sets and saves. These manipulate the cursor, clicks on applications and other files in standard locations, and then manipulates the data or downloads additional malicious payloads.

**RUNNING THE MALWARE**

**Network-based observations**

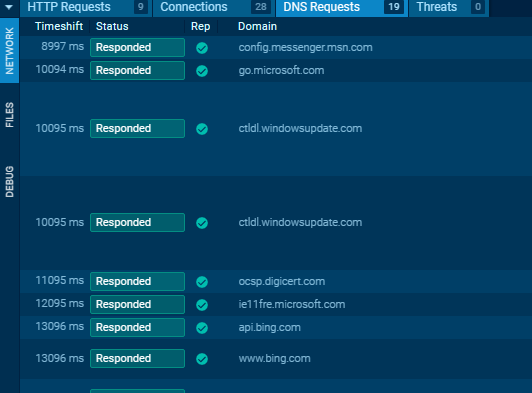
The initial observation of this malware within app.any.run showed 9 HTTP requests, 28 connections, and 19 DNS requests. The HTTP requests were made to the windowsupdate website, digital certificate websites, and Microsoft service switching stores. They were located in 4 different countries: USA, Germany, UK, and the Netherlands. These HTTP requests and connections over HTTP and HTTPS, particularly to Germany, UK, and Netherlands (if the host is located in the US) would be particularly potent identifiers of infection.



**Hypothesis of Purpose**

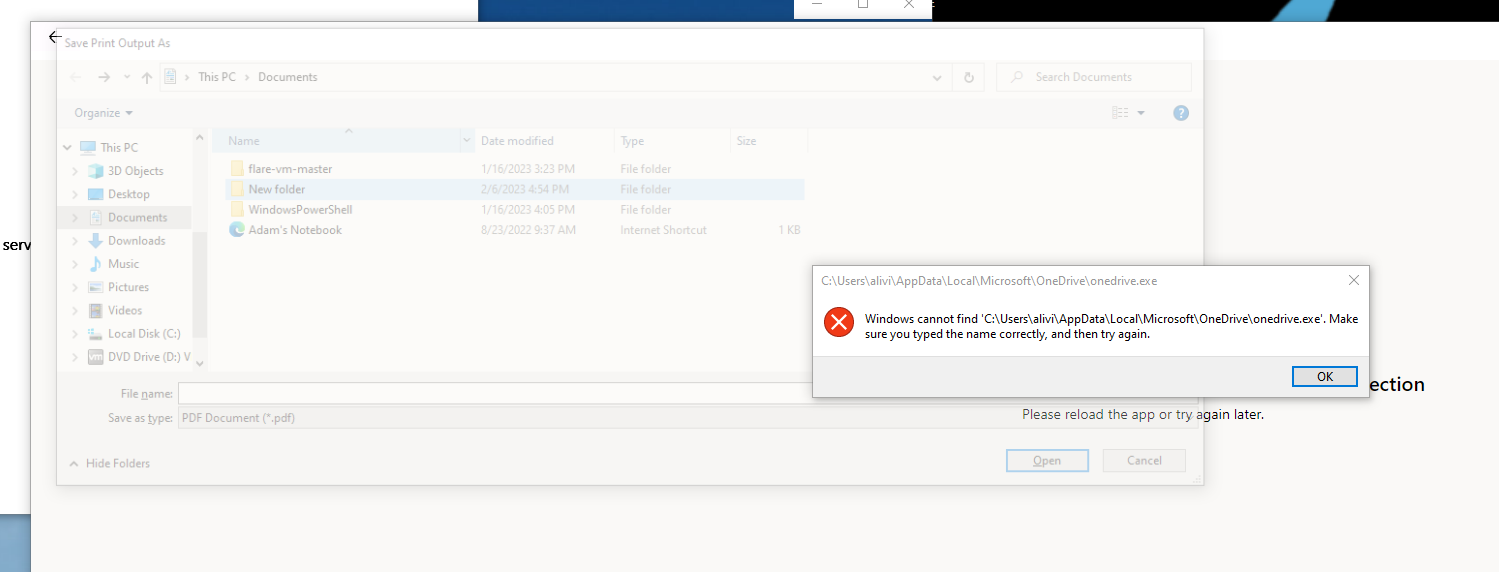
Based on the URLs that this malware calls out to, as well as the programs that it spontaneously opens, I believe that the purpose of this file is to steal the software license keys for Microsoft office programs. This would render the programs useless on the license holder’s PC. It is also possible that since the malware opens Outlook it will exfiltrate sensitive emails (if infected on an enterprise machine). However, because there aren’t any URLs or IP addresses outside of well-known websites, this is unlikely.

Additionally, there are callouts to digital certificate websites and OCSP domains which would possibly have the malware place the host machine’s digital certificate on the Certificate Revocation List, thereby making it unable to be trusted in the Public Key Infrastructure hierarchy.

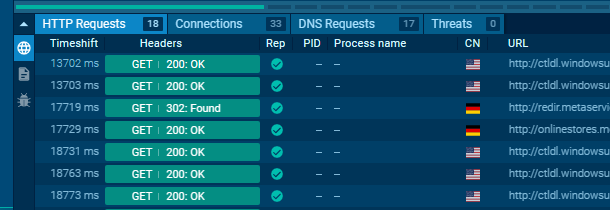


**Running In My Windows 10 VM**

Immediately, as was observed in app.any.run, there was total loss of control over the functions of the machine. There were multiple apps that were opened, many of which were internet-based. The mail app and attempts to access OneDrive were especially concerning. File explorer was also opened and attempted to save a .pdf document in the “Documents” folder, but it is unknown as to what the nature of this document is. Multiple openings of mouse settings were opened.



When the malware was run a second time, different programs were opened. I noticed that instead of trying to open specific files, the mouse was randomly clicking on whatever it could find to cause as much chaos as possible. All applications and folders that were opened were readily-accessible by simply clicking. The randomness of the program produces different results each time. Based on this observation, I suspect the malware is attempting to cause as much chaos as possible and does not have a specific goal. Because of this chaos and randomness, the goal is most likely to overwhelm the system and deprive it of its resources. This was confirmed when the malware was run again in app.any.run and produced completely different outputs for network connections and applications that it opened.



**Running in the VM – How often?**

If I did not have snapshots, I would still be able to run this machine. Simply using Ctrl+Alt+Del and clicking the restart option in the lower-right-hand corner is a simple fix. After the restart and not reverting back to a snapshot, the program was not running. This indicates that the service manager was not altered in order to run the program on startup. After restart, the task manager was compared from an uninfected state to the infected state for an instance of the malware and no differences were detected.

It is unclear if the registry keys were altered deliberately by the malware as using RegShot prior to running it and then restarting the machine also restarts RegShot. Therefore, the first shot is not saved and cannot be compared to the second shot.



**Recommendation to Management**

Since the malware was ran on machines without an internet connection for safety reasons, it is difficult to determine the impact to the enterprise network that this malware may have caused. A network connection combined with the random clicking and accessing network-based applications could have caused a lot of inadvertent damage. If an untrained user had clicked on it, it is uncertain as to how long the malware would have run and therefore it is impossible to know the extent of the damage. However, even moderately-competent people know about “Ctrl+Alt+Del” and would have likely figured to restart their computer from that screen, even in a panic. If not, then it is reasonable to assume that they would conduct a hard-shutdown by holding the power button on the machine.

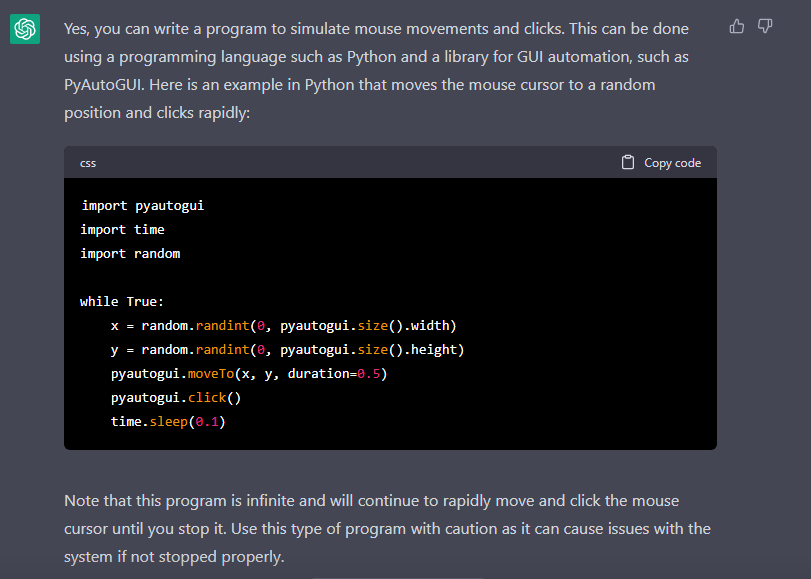
Based on the uncertainty of damage to network and host systems due to the chaotic nature of the malware, an analysis would have to be conducted on outbound network traffic from the infected machine’s IP or MAC address from the enterprise’s network monitoring system. An analysis would also have to be conducted of enterprise servers and database systems during the time the malware was running to see what, if any, information was altered by the infected machine.

I would recommend that management activate a small-scale incident response team to detect what, if any, damage was done to high-value enterprise systems by the host machine during the time period that the malware was running. I would also recommend the immediate quarantine of the infected machine for further analysis to determine if any registry edits were made that could possibly spread the infection. Furthermore, a total scan for the hash value of the malware executable on the enterprise network should be conducted to ensure that all instances of it are deleted if it deliberately or inadvertently copied itself to other machines or file repositories.

**ChatGPT**

This malware was extremely simple to create in ChatGPT. Since it was just random mouse-clicking at random positions, I asked the question, “can you create a program that randomly moves the mouse cursor and clicks rapidly?”. It spat out a simple .py program that performed just that. Based on what I know about python, it would be simple to import the screen resolution of the host system as where the mouse is allowed to be. I could then alter the clicking and cursor position interval for maximum chaos. It also runs on an endless loop until the program is closed. Easy-peasy.

I did not run the program on my machine because I don’t have an IDE for python installed on my VM. Additionally, I did not want to run it on my actual machine due to potential “unintended consequences”.



* Make an account on <https://chat.openai.com/chat>, which is ChatGPT, the AI bot. Ask the bot to recreate the program to the best of your abilities.
  + What prompts did you use?
  + What was the output?
  + Does it work and include the text output?
* Go to two other student's posts and observe their findings. Post if you agree or disagree with the results.
  + Did they find something using a new technique, and if so, would you use this next time?
  + From a business perspective, if you were both being paid at the same rate. Would your analysis be more cost-productive and achieve the same results?
    - Suppose you were working on this malware to see if this could be allowed in your organization. Did your analysis provide enough detail to make this determination?