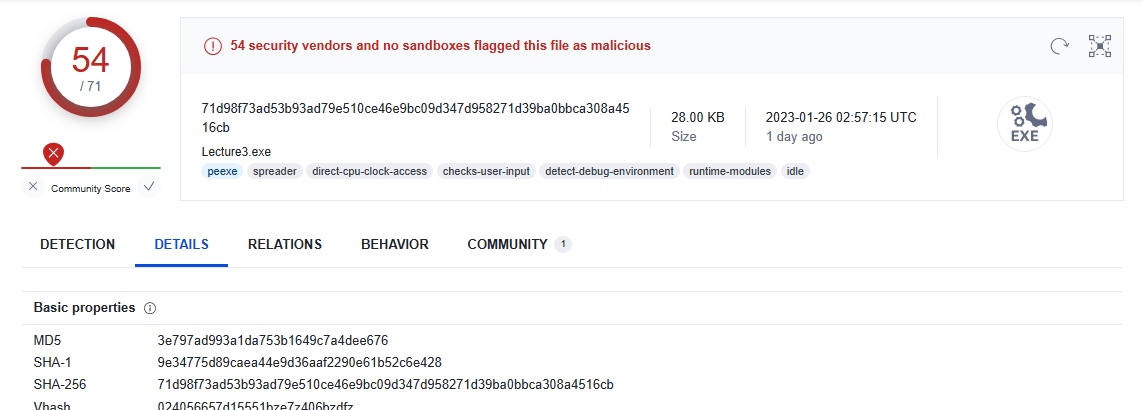
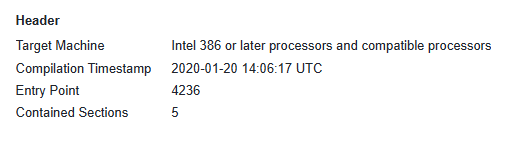
**VIRUSTOTAL**

Signatures: Matches 54 of 71 existing antivirus signatures.

MD5 Hash: 3e797ad993a1da753b1649c7a4dee676

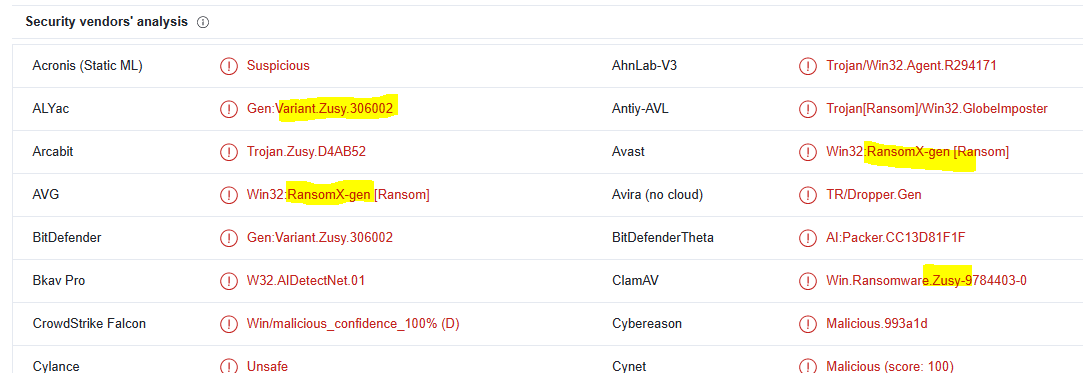


**Compilation Timestamp**: 20 Jan 2020 at 14:06:17 UTC

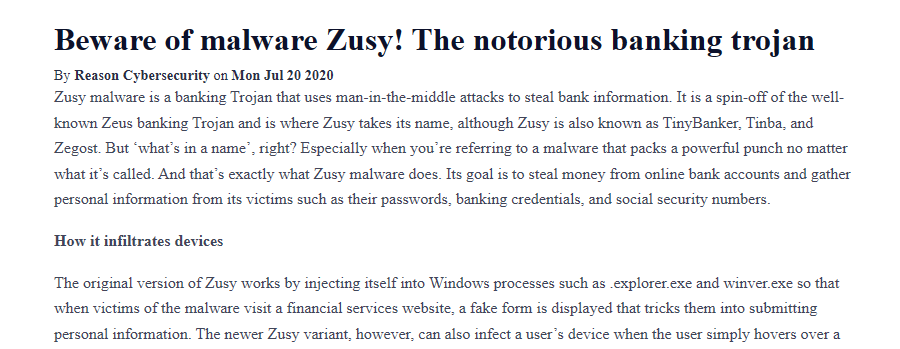


**What is the malware known for?**

Many of the antivirus signatures on VirusTotal suggest that this piece of malware is ransomware. This is based off of naming conventions such as RansomX and “Zusy”.



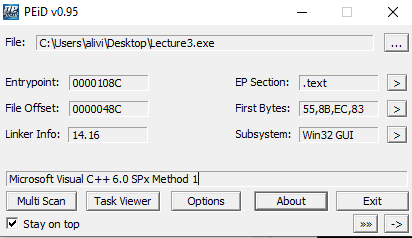
According to Reasonlabs Blog, Zusy is a piece of malware that is known for MITM attacks to steal banking information. It tricks users to visiting a banking website to fill out a fake form that harvests the user’s credentials. A newer variant of Zusy can infect a machine by simply hovering over an infected PowerPoint document. https://blog.reasonlabs.com/2020/07/20/beware-of-malware-zusy/



**STATIC ANALYSIS**

**Packing/Obfuscation**

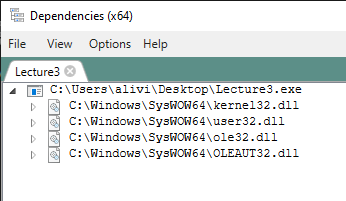
PEiD nor VirusTotal detected any indications of packing. The output from PEiD suggests that this malware was written in C++.



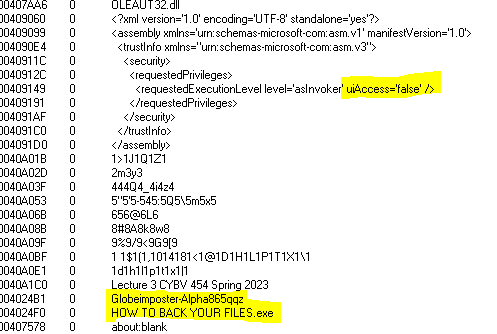
**Imports**

Any imports indicate what the malware does?

Using Dependencies and cross-referencing with VirusTotal, .dll imports that the malware calls for are kernel32.dll, user32.dll, ole32.dll, and OLEAUT32.dll. These imports suggest the malware accesses the kernel in some way using kernel32, alters the user interface in some way using user32, and alters the functionality of Windows programs using ole32 and oleaut32.

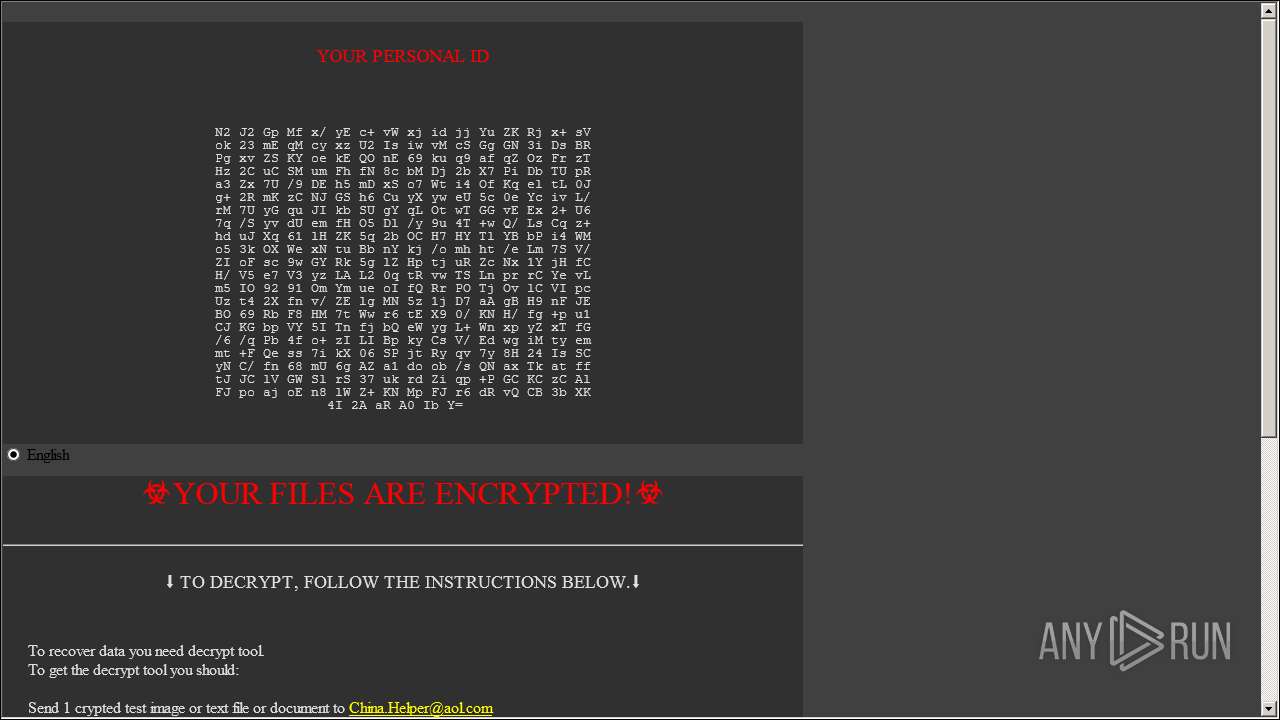


There is also some suspicious imports when the file is viewed in BinText. There is an XML command that states uiAccess = ‘false’ as well as an executable titled “HOW TO BACK YOUR FILES.exe”. An additional suspicious import is “globeimposter-alpha865qqz” which shows after a simple Google search that it is a type of ransomware. These imports suggests that this piece of malware is a form of ransomware.

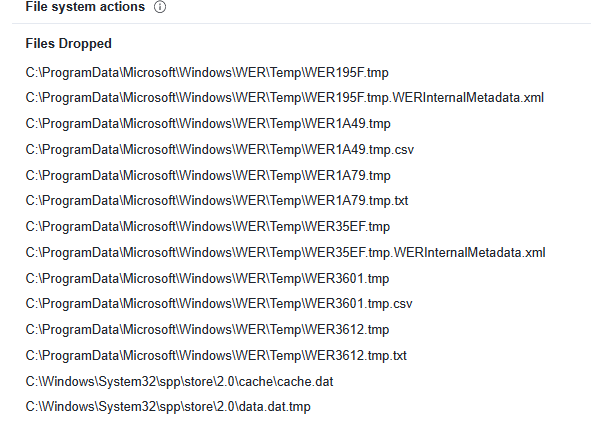


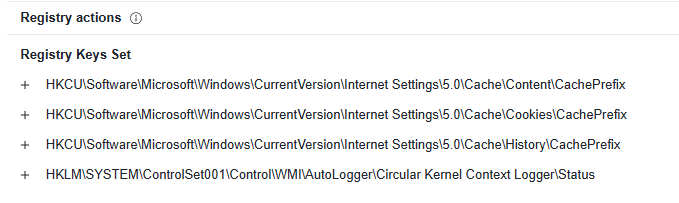
**File or Host-Based Indicators**

Since this program leans heavily towards being ransomware malware, it would be self-evident that the computer is infected. This is because malware normally takes the form of locking a user out of their system and displaying a warning message that warns the user their files are encrypted unless a ransom is paid. This was confirmed by running a an analysis of the malware on app.any.run.



VirusTotal also indicated that the malware drops a number of files that could be looked for on an infected machine as well as some registry key alterations.

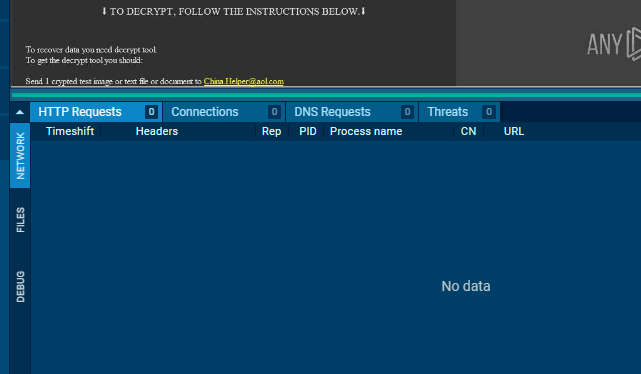




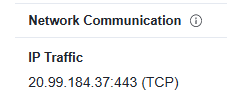
**DYNAMIC ANALYSIS**

**Network-Based Indicators**

App.any.run did not pick up any network-based indicators when it ran this malware and because the malware locked me out of my sandbox environment, I was unable to use FakenetNG or other application-based tools to monitor network traffic.

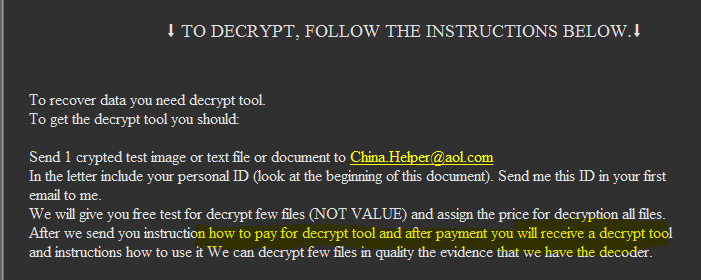


However, VirusTotal did show one network-based indicator that could be used to find this malware or other variations of it. It shows a TCP connection over port 443 to 20.99.184.37. If the variation of this malware is the bank account credential-harvesting type, then this IP address most likely is the host that is collecting that data and hosting the fake form.



**Purpose of File**

It was readily-apparent from the beginning that this was ransomware. The entire point of ransomware is to scare a user into paying an outrageous sum of money to unencrypt or unlock their machine. In fact, the ransom message this file displays explicitly states that the user must pay for a decryption tool to unencrypt their files.



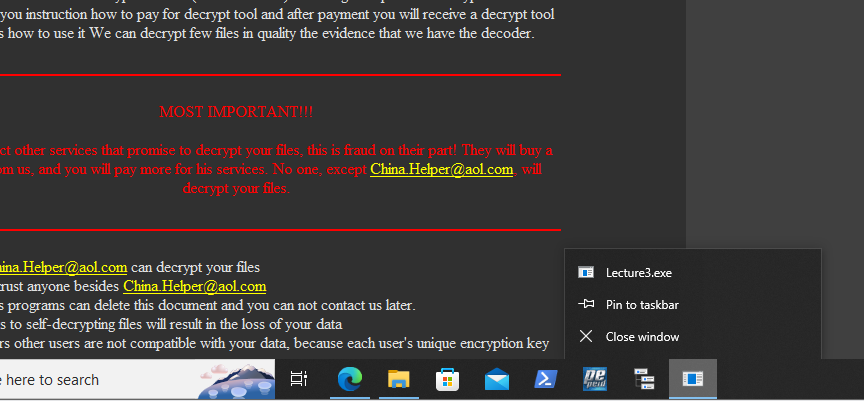
**Most Exciting Part**

It was interesting how quickly the file locked the machine (or so it seemed. Will get to that in a moment). The poor grammar, broken English, and [China.Helper@aol.com](mailto:China.Helper@aol.com) email address suggests that this malware was written by a non-native English speaker. What was also surprising was that the ransomware claimed that other services that promise to decrypt files is an attempt at fraud, which is hilarious. It’s almost like a mobster saying, “Damn, I know I absolutely destroyed your home here. But not to worry, I can fix it up for you for a modest fee. But don’t contact other contractors to help repair it because they will just scam you.” Rich, Mr. China Helper. Very rich.



**Now, how often would you be able to run this VM, excluding snapshots?**

I could run this machine very often without the use of snapshots. By just pressing “windows key + x”, I clicked on “desktop”, right-clicked the malware, and closed it. The supposed encryption of the files was bogus and this was just an application that ran in full-screen mode which hid the toolbar, much like a game.



**Recommendations to Management**

What would your recommendation be to Management? Do we need to stop generating revenue and clean, or can we go on and clean as we go?

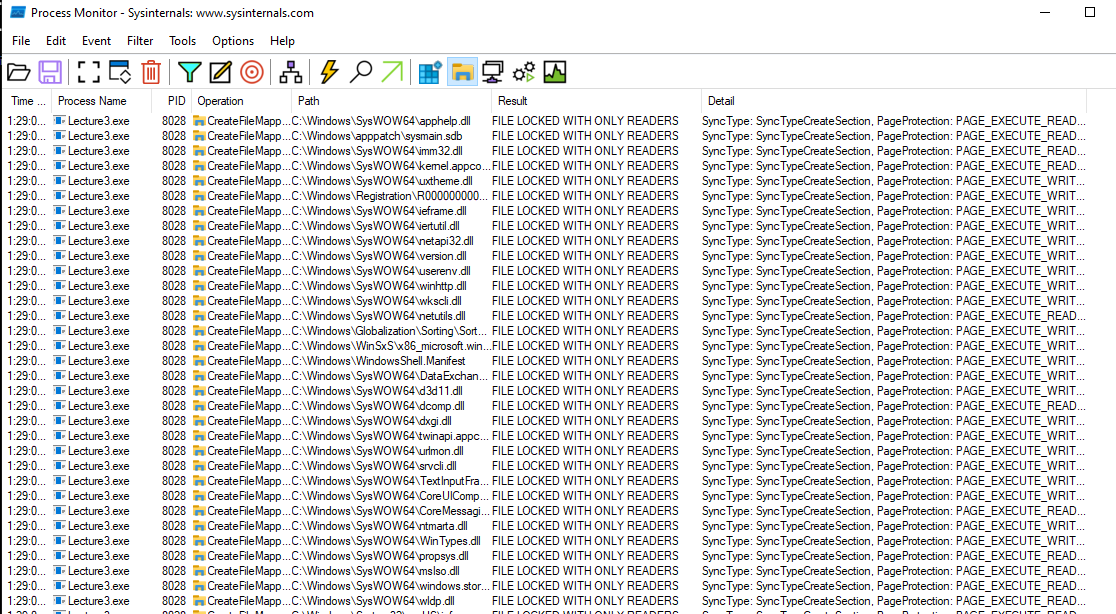
Can you clean the system, and if so, how would you do it?

After discovering that “win + x” can close the file, I re-ran the malware with procmon running and did not find any network activity so other computers on the network are most likely safe. However, the machine that this file was executed on did show numerous .dll file alterations that changed permissions to “FILE LOCKED WITH ONLY READERS”, effectively making those dlls have read-only permissions.

To err on the side of caution, I would certainly follow incident response steps to contain the affected machine and segregate it from the rest of the enterprise network. and collect necessary forensic evidence in the proper order of volatility to preserve the evidence. I would not yet reimage the machine until a proper forensic analysis was conducted. The user who ran the malware would be issued a new machine and I would call for immediate remedial cybersecurity training for not only the user, but for all enterprise employees.

More likely than not, the malware that ran did not make significant changes to the machine that would prevent it from performing its intended function. If I were to clean the system, I would run multiple antivirus/anti spyware scanners on it to identify, quarantine, and eradicate signs of infection. I would then reinstall all Windows dlls after downloading them from the official Microsoft website. I would then ensure that this malware signature is added to the enterprise intrusion detection and intrusion prevention system’s list of signatures.

I would then add the IP address that Virus Total identified to a block list for all machines in the enterprise. However, if resources and time allowed, I would also create a honeypot in conjunction with a DNS sinkhole for that specific IP address in order to monitor the malicious actor’s TTPs.



**Chat GPT**

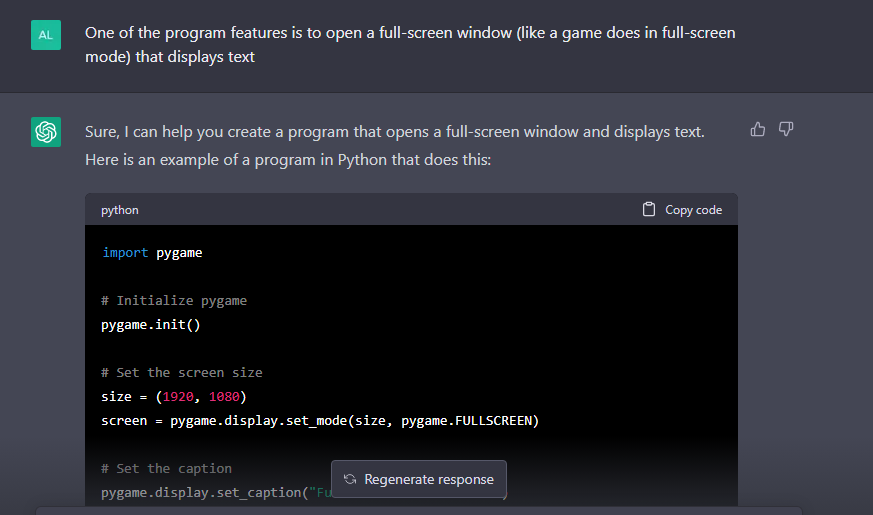
Ask the bot to recreate the program to the best of your abilities.

What prompts did you use?

What was the output?

Does it work and include the text output?

Since I realized that the “malware” was simply an application that created a full screen window, I wanted to know how to create that functionality in python. I first asked it to create a program with, “One of the program features is to open a full-screen window (like a game does in full-screen mode) that displays text ”.



ChatGPT gave me a good output that I tested and it opened up an application when I ran the debugger in WingIDE. I noticed that there wasn’t a “quit” button and the program had to be closed in the same manner that I had to close the malware that was analyzed above. From there, I started adding the strings that the malware outputted, giving them the same font and color. The most difficult part was getting the “Personal ID” section to print out where each character was aligned with each other from top-to-bottom. A simple font change from Times New Roman to Courier New accomplished this.

Every time I wanted to display the text in pygame in a similar fashion to the malware, I just asked ChatGPT how to do it. I avoided using phrases associated with malware, such as “lock the screen” because it would warn me that creating malware and using it is illegal.

I didn’t build the rest of the program or compile it into an executable due to time constraints, but it was a fun exercise to recreate it and learn some functionality of the pygame library which I was unfamiliar with.

**I have attached the .py file to this post if you want to play around with it.**

* 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?