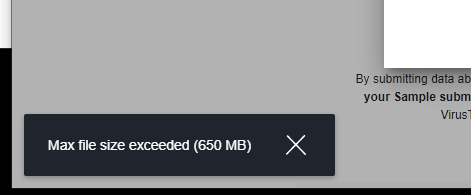
We will now be looking at dirty tricks used by malicious actors who try their best to get a user to click on a malicious program. For this discussion topic, we will use all the tools we have learned.

Answer the following questions using the Lecture 12 file:

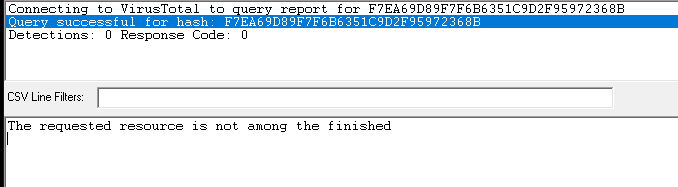
Answer the following questions:

**Upload the file to**[**http://www.VirusTotal.com/**](http://www.virustotal.com/)**. Does the file match any existing antivirus signatures?**

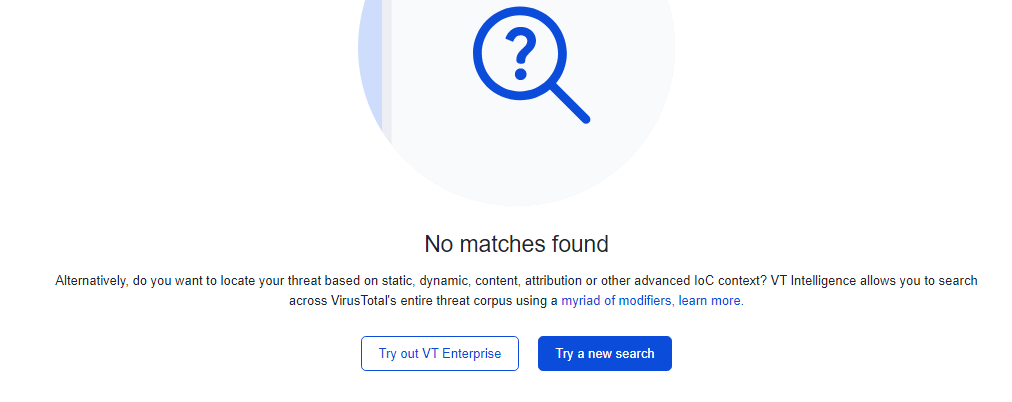
No. It does not. The file could not be uploaded to VirusTotal due to exceeding the maximum allowed size.



However, with FlareVM you can right-click on the file and select “Submit to VirusTotal”. Although it didn’t finish, there weren’t any detections.



The hash was submitted to VirusTotal and there weren’t any matches found.

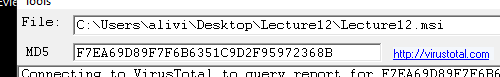


**What is this file known for?**

It is unknown as to what this file is known for as the file is too large.

**What is the hash of the file?**

MD5: F7EA69D89F7F6B6351C9D2F95972368B



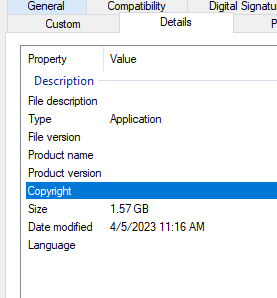
**What complications did you run into or what issues would prevent you from uploading this to Virus Total?**

It was too large and VirusTotal did not even recognize the hash. This is most likely due to the file being too large and therefore could not be effectively submitted to be compared against security vendor signatures.

**Using the tools we discussed so far, answer the below.**

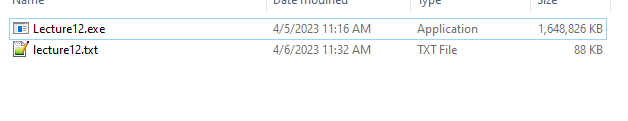
**When was this file compiled?**

PEview nor DIE could detect a compilation date for this file. This didn’t change even when this file’s extension was changed from .msi to .exe. The only data related to some sort of timestamp is examining the properties of the file and seeing a last modified date/time stamp of 5 April 2023 at 11:16AM MST.

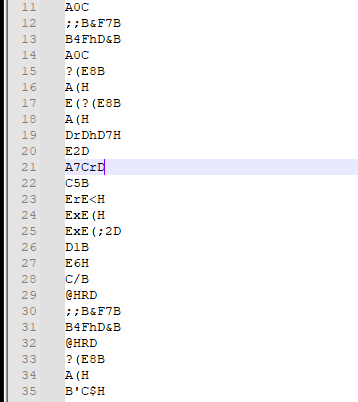


**Are there any indications that this file is packed or obfuscated? If so, what are the indicators?**

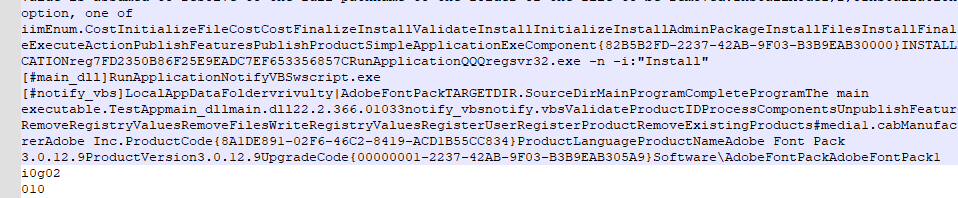
Yes. The file shows that it about 1.6 GB large, But the strings analysis only yielded 88KB of data. For a file of this size, there should much more data that is gleaned. It was originally set as the .msi file extension indicating that it is an installer of some sort.



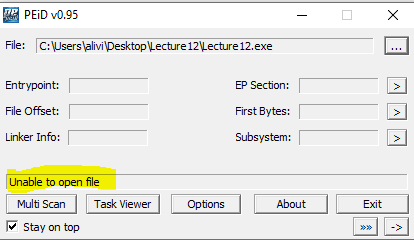
In the strings, there is a lot of data representative of encoded hex values.



There is a singular long string within the strings output that indicates an executable file. There are also instructions of “-n -i: Install” which could indicate some command line arguments parameters to install another program. There is also a reference to VBScript, a Visual Basic programming language.



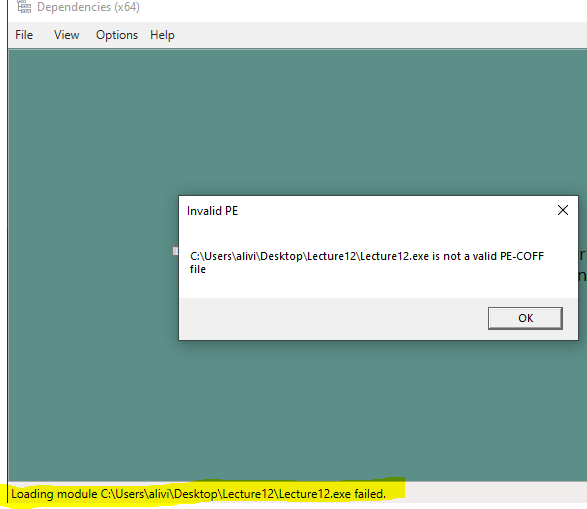
PEiD could not open the file to detect any packing algorithms.



Based on the size of this file combined with the limited strings and other data, as well as the fact that it is a .msi installer file, this file is both packed and obfuscated in order to install another program. Most likely this program is stored within other data regions of the file.

**Do any imports hint at what this malware does? If so, which imports are they?**

When trying to load the file into Dependencies, it is shown as not being a valid PE file.



As mentioned previously, there were two .exe files that were detected in the strings analysis that could potentially also be used as host-based indicators (to answer the next question). There were the two following .exe files:

RunApplicationNotifyVBSwscript.exe

and:

regsvr32.exe

regsvr32.exe is a command-line utility that is used to register and unregister .dll files and ActiveX controls in the Windows Registry. Running this utility along with a specified .dll or ActiveX parameter can register or unregister the file. It is typically used when installing or removing a program that requires these (un)registrations. It also performs tasks related to the registry.

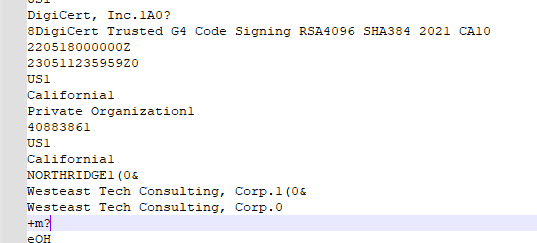
Therefore, we can expect to see some registry changes and although this file has not yet been positively-identified as malware, this is normal for a .msi file. We can expect to see regsvr32.exe and the VBS .exe file to run when this file is run.

**Are there other files or host-based indicators you could look for on infected systems?**

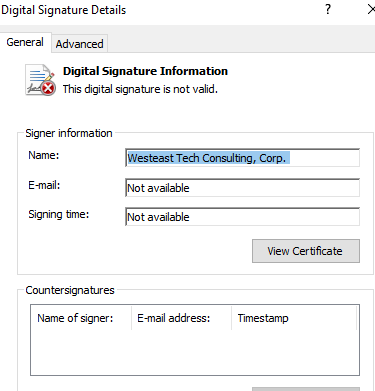
See above.

**BEFORE you run this malware, would you consider this file malware based on your findings? "Exclude your virus total findings." Would you have accidentally activated this malware before your research?**

I would not consider this malware. There are relatively standard imports this file has for being an installer file. Additionally, it appears that this installer is digitally signed by Westeast Tech Consulting located in Northridge California. This is a real business and is located in that part of CA. However, it does appear that a lot of the identifying information is appended with the integer of “1” which is slightly odd, but nothing too much to raise eyebrows.



But when clicking on the Digital Signatures tab when opening the file’s properties, the signature is not valid, does not have an email, signing time, nor countersigners. Based on this information alone, I would not trust this file but not necessarily consider it malware.



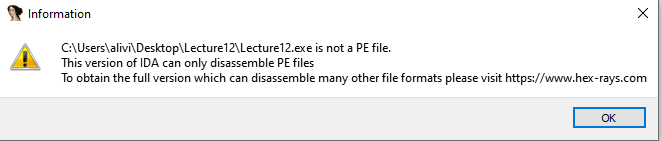
* \****ONLY IN THE VM\****

**TURN OFF NETWORKING!!!!!**

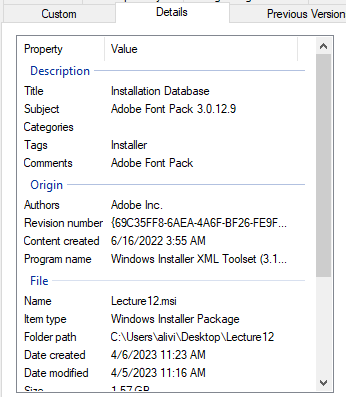
**Using the tools we discussed, answer the questions below and provide screenshots.**

**Based on the Imports and strings, what does the program appear to do?**

See above. IDA also could not open the file, even when the extension was changed to .exe.



Why does this happen? Obviously, the file is not a PE and IDA could not be tricked into thinking it was a PE file. The strings analysis clearly indicated that this is an installer file and it even says so within the Details tab of the file properties, containing relevant information.



**Open the Strings window within IDA pro and provide an analysis of which ones are interesting and why.**

See above.

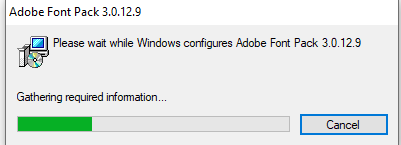
**Open the file in IDA pro and identify the following:**

**What does IDA identify as the START of the program**

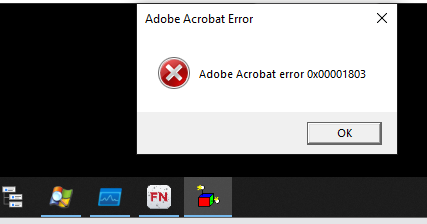
See above.

**Run the file and record your observations. This malware opens up a dialog box, what information is being displayed?**

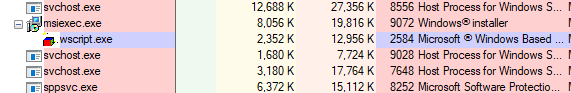
There appeared to be an installer window displayed.



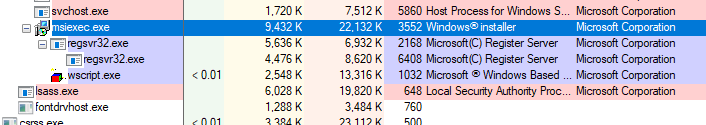
After the installer window went away, there was an error window. This error window was associated with a colorful box that looked like it has sparklers on it.



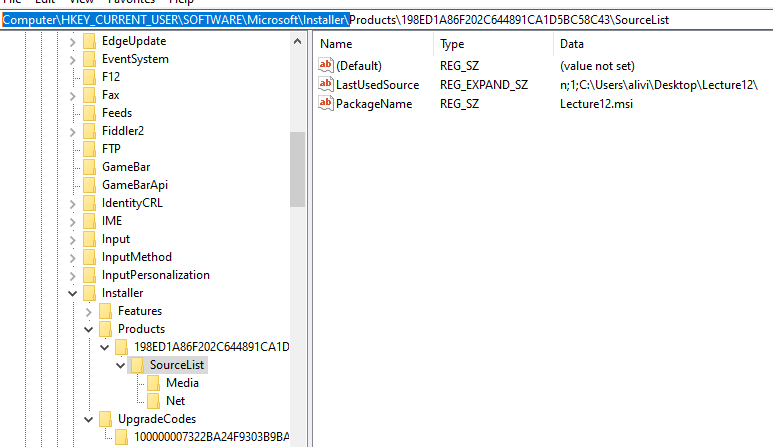
Within Process Explorer, we see that this icon is associated with wscript.exe, a child process of msiexec.exe.



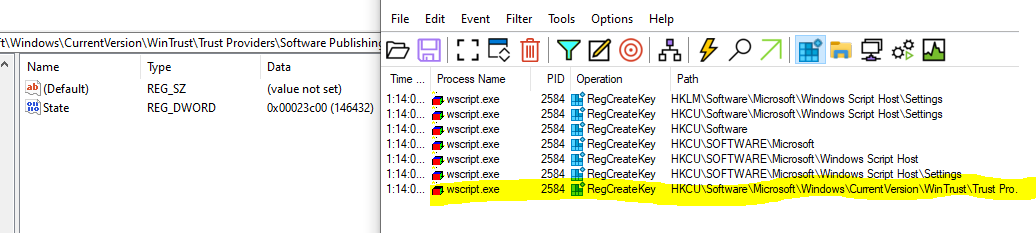
There was also a brief moment, even after the error window was closed, that msiexec.exe, a child process of wininit.exe > services.exe contained the previously-identified process of regsvr32.exe. Eventually, all of these processes disappeared from Process Explorer.



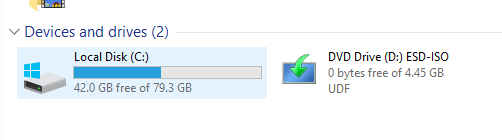
Msiexec.exe appeared to create some registry keys within HKCU\SOFTWARE\ Microsoft\Installer\. This is normal for an installer program.

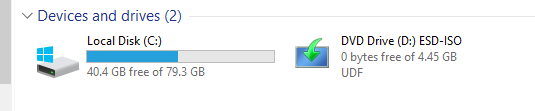


The wscript.exe process created a registry key within HKCU\Software\Microsoft\ Windows\CurrentVersion\WinTrust\Trust Providers\Software Publishing, adding itself to a trusted list of software publishers.

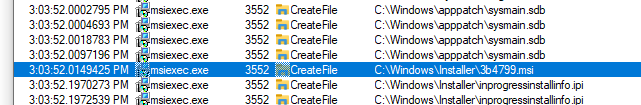


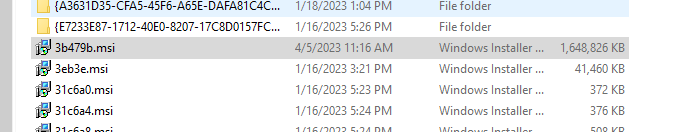
There was an additional 1.6 GB chunk of space that was used after the process was run. Below are before and after pictures of the C drive.



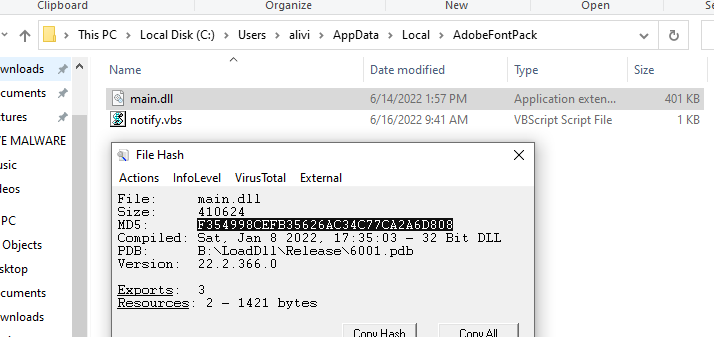


This change occurred within the C:\Windows\Installer directory and made a similarly-sized of 3b4799.msi.

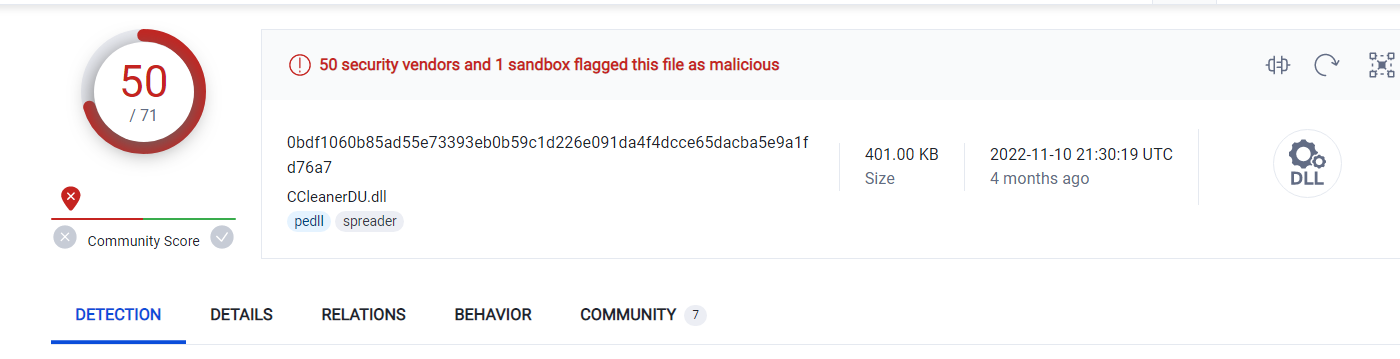




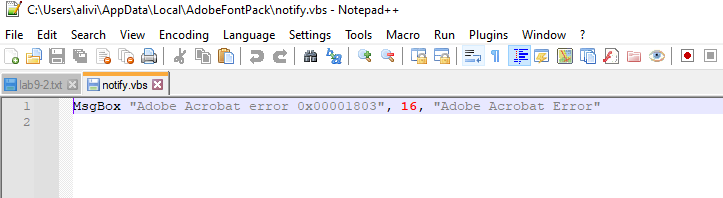
There also seemed to be a .dll file created within AppData\Local\AdobeFontPack. This directory should contain font-related files for use by Adobe software. The md5 hash for this file is F354998CEFB35626AC34C77CA2A6D808. The hash for the .vbs file is 0308AA2C8DAB8A69DE41F5D16679BB9B.



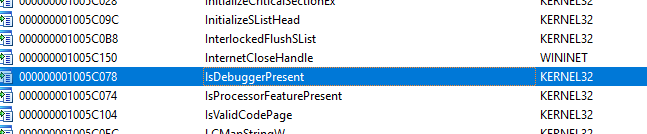
When these hashes were searched for in VirusTotal, the first one returned 50 malicious signatures. The .vbs did not return any malicious signatures.



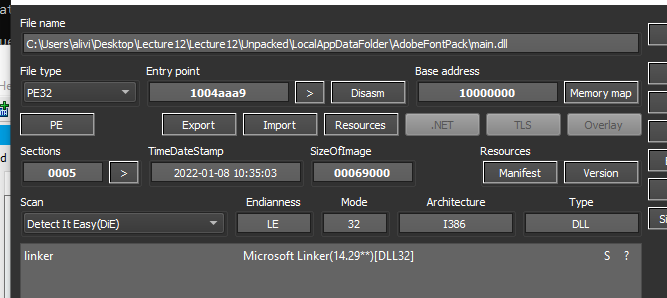
However, the .vbs file included the same error that was printed out on the screen.



This .dll file was loaded into IDA pro and there were some interesting items found. First, there was an import for IsDebuggerPresent which indicates anti-debugging capability.

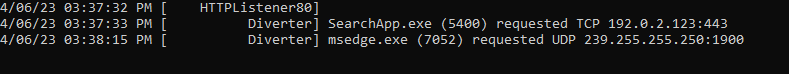


This would have been a lot easier if I just used UniExtract in the first place. The modified compilation timestamp is 2022-01-08 at 10:35:03 local time.

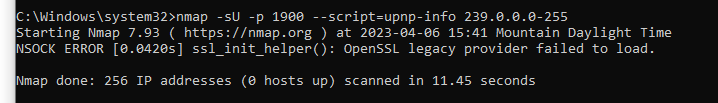


**What network-based indicators could be used to find this malware on infected machines?**

There were attempted UDP connections over port 1900 to IP address 239.255.255.250 through the use of the edge browser.



Although this was an attempted connection, I used nmap to scan the IP range over port 1900, which yielded no results.



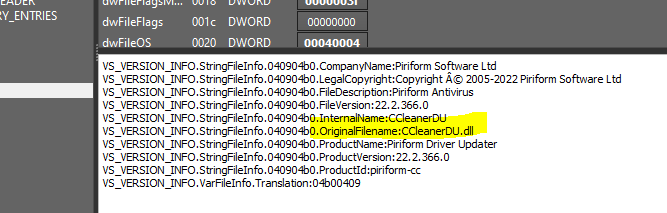
**What host-based indicators could be used to find this malware on infected machines?**

The existence of main.dll within AppData\Local\AdobeFontPack is clearly malicious.

**What did this malware do?**

This malware was disguised to look like an installer program and to prevent analysis on VirusTotal. When it was run, the malware made a copy of itself within the C:\Windows\Installer directory as a red herring in order to install the true malicious program of main.dll within AppData\Local\AdobeFontPack. This also included the .vbs script which was used to print the error message.

The original name of the file is CCleanerDU.dll instead of main.dll, most likely to disguise it as a type of antivirus software .dll file.



However, VirusTotal reports that this file as the Matanbuchus trojan. It is a Malware-as-a-Service and allows third-parties to use it to drop malware files from a C2 node over secure shell. More information can be found here: https://www.cyberark.com/resources/threat-research-blog/inside-matanbuchus-a-quirky-loader

This is most likely why FakeNet is reporting attempted connections by Microsoft edge. The malware is using Edge to try and connect to the C2 node to download additional payloads.

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

After restarting the machine, I noticed that there was still UDP traffic attempting to connect over port 1900. This means that this malware established persistence and I wouldn’t feel comfortable running this machine with networking enabled.

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

We need to disconnect this machine and stop generating revenue. Quarantining the machine away from the enterprise network, I would be interested to see what the malware does when it is allowed to download the intended file from the C2 node.

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

I would need to find and delete all of the new services it created in order to prevent persistence.

**Go to two other students' posts and observe their findings. Post if you agree or disagree with the results.**

<https://d2l.arizona.edu/d2l/le/1243099/discussions/threads/9791250/View>

<https://d2l.arizona.edu/d2l/le/1243099/discussions/threads/9791731/View>

**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?**

Yes, I believe it did. After extracting the main.dll and placing it into VirusTotal, the signature was immediately recognized as a third-party malware installer which could download additional malware to compromise enterprise systems.

**Did they find something using a new technique, and if so, would you use this next time?**

Not really, but I should have unpacked the file first so I wasn’t spinning my wheels with the .msi file. I was so far along in my analysis when I decided to use UniExtract that the information I gathered from it was enough to make a determination.

**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?**

Adam Braun did an excellent job and was more efficient with analyzing the file than I was. I would choose his analysis over mine, simply because I spent too much time analyzing the .msi file whereas he was able to unpack it and do the analysis on main.dll.