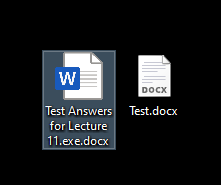
We will now be looking at dirty tricks used by malicious actors who try their best to get a user to click on a malicious program. For this discussion topic, we will use all the tools we have learned.

Answer the following questions using the Lecture 11 file:

Answer the following questions:

**Observe the file first and describe your first impressions**

At first, it looks like a word document that contains the answers to this discussion post. However, having this icon would indicate that Microsoft Work is installed on my VM, which it is not as a REAL .docx file is next to it for comparison. Additionally, there is a “.exe” extension on the file, indicating that this is a PE. The .docx icon that the file has also appears to be of low quality when compared to a real .docx file on my PC proper. The name for the file also allowed the extensions to be hidden unless the file was single-clicked on.



**Describe what looks odd / off about this file**

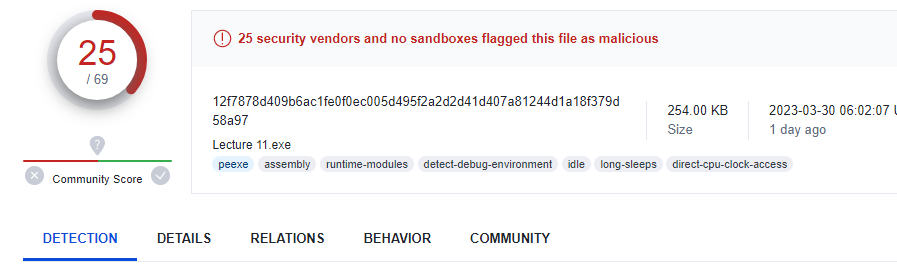
See above.

**You or a normal user may activate this file by accident. What techniques would you deploy to make this file more detectable?**

First and foremost would be training users to keep an eye out for .exe file extensions within file names that appear to be normal. They need to be trained to NOT open them. Other techniques would be to implement security software within an organization that scans file names for multiple extension types, keeping an especially-keen eye for .exe file types. Any file with a .exe extension should be flagged immediately and blocked within the organization. There should also be access controls implemented for users to not be able to run unauthorized .exe files or applications.

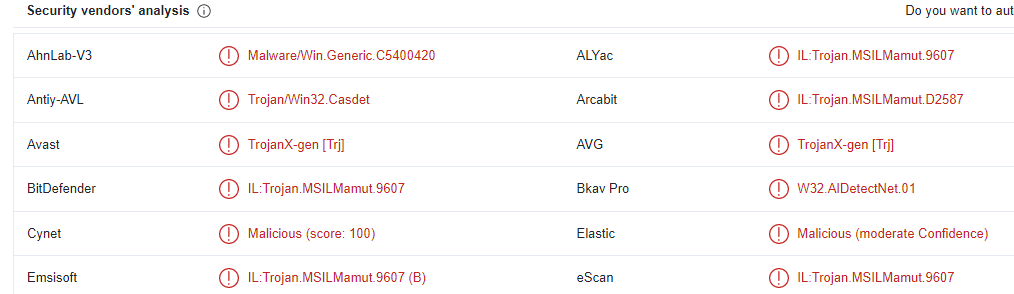
**Upload the file to**[**http://www.VirusTotal.com/**](http://www.virustotal.com/)**. Does the file match any existing antivirus signatures?**

Yes. It matches 25 of 69 security vendor signatures.



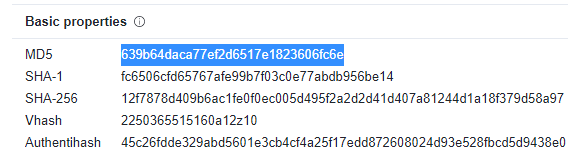
**What is this file known for?**

Multiple security vendors label the file as MSILMamut trojan. MSIL means it was programmed in Microsoft Intermediate Language, a component of the .NET framework. Other than that, no more information was found on MSILMamut.



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

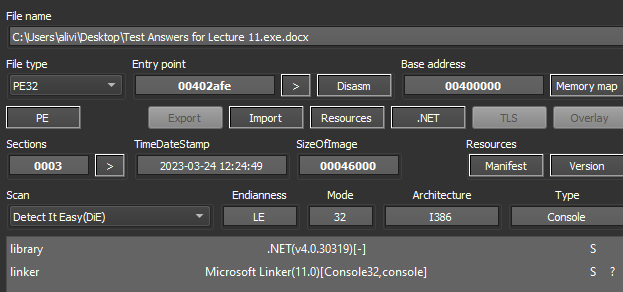
MD5: 639b64daca77ef2d6517e1823606fc6e



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

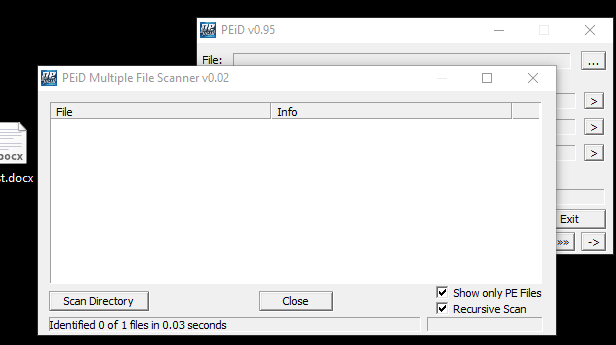
**When was this file compiled?**

DIE states that it was compiled on 2023-03-24 at 12:24:49. However, DIE only shows local time so the correct UTC time is 18:24:49.

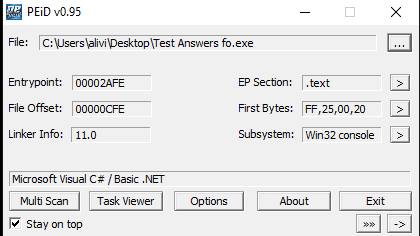


**Are there any indications that this file is packed or obfuscated? If so, what are the indicators?**

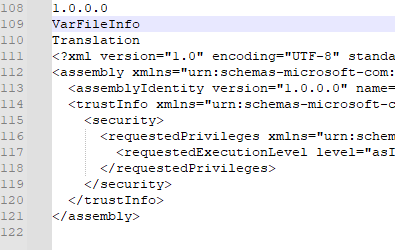
The screenshot above shows that there wasn’t any indication of packing. However, PEiD appeared to be confused by the multiple file extensions and popped up a new window upon loading.



However, removing the .docx extension from the end shows that there wasn’t any packing detected by PEiD.



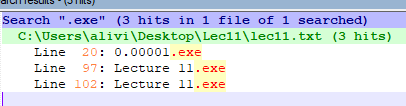
Running strings on the file, there were only 122 lines of code.



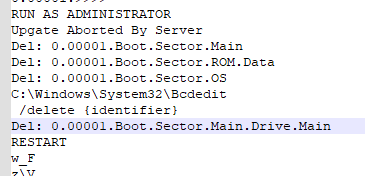
Based on these observations, the file is not packed but has definitely obfuscated its intentions due to the lack of code. Additionally, there were multiple imports observed that were not within the strings (see below).

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

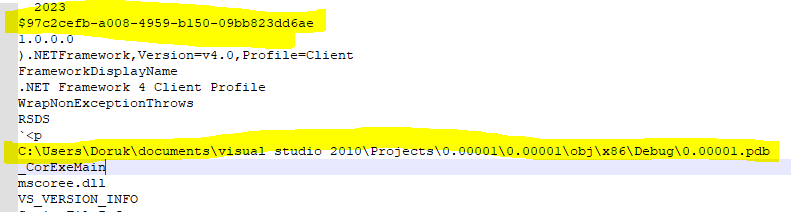
There were three .exe references in the strings output.



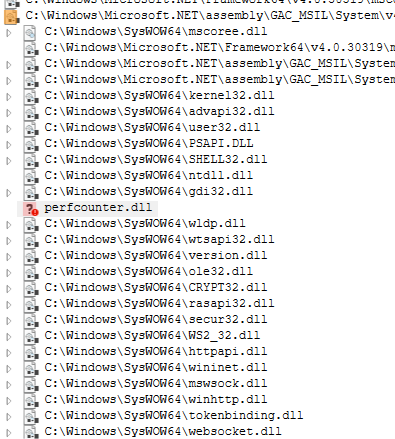
The most concerning import was what appears to be cmd.exe running the Boot Configuration Data utility. It appears that it uses this tool through 0.00001.exe to delete important boot sectors of the hard drive. It then calls Restart, which will most likely cause the computer to become a brick.



There also appears to be a unique identifier within the strings as well as a .pdb (program database) file stored in a new user titled “Doruk” within the documents folder.

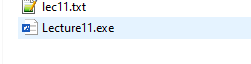


There are a lot of .dll imports detected by Dependencies for this file. Notably, there are network-related .dll imports as well as cryptographic imports.



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

Other than what was mentioned above, I did not find anything else. The main indicators would be the low-quality icon and the multiple file extensions within the file name. A more noticeable example of this low-quality icon can be seen below.



**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?**

Yes, I would consider this malware. I noticed it was a suspicious file off the bat with the multiple file extensions and the low-quality icon. Plus, a real .docx file would not have all of those strings in it. It is tough to say if I would have accidentally activated this malware. If I downloaded it from a trusted source, then yes, I probably would have. Although the low-quality icon would be the first indicator that this was malicious which would prompt me to examine the file name itself.

\****ONLY IN THE VM\****  **TURN OFF NETWORKING!!!!!**

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

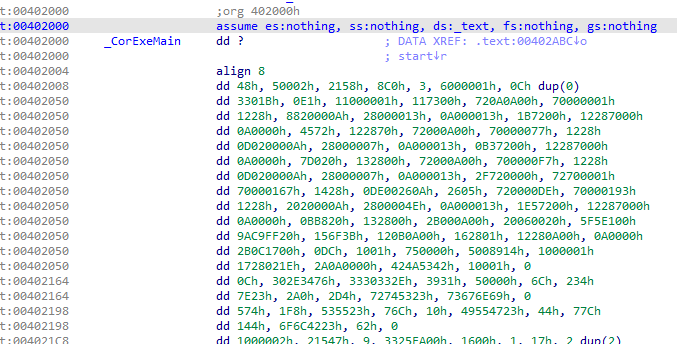
**Based on the Imports and strings, what does the program appear to do?**

Based on the imports and strings, I suspect that this file likely connects to an external website to download the file 0.00001.exe which will execute on the machine and cause it to become bricked. It is possible that the file creates a new user named Doruk, which potentially also provides a backdoor into the machine. Since the user cannot access the machine, they have free reign to steal anything they please.

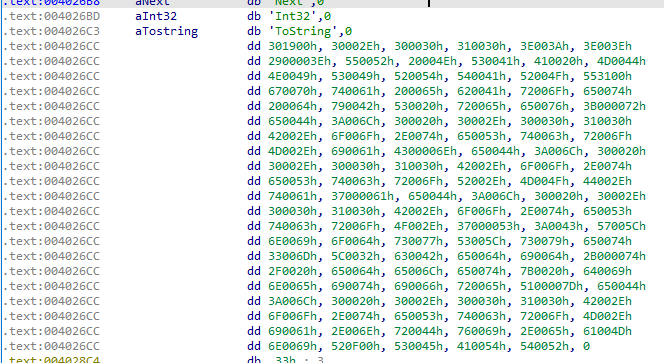
**Open the file in IDA pro and identify the following:**

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

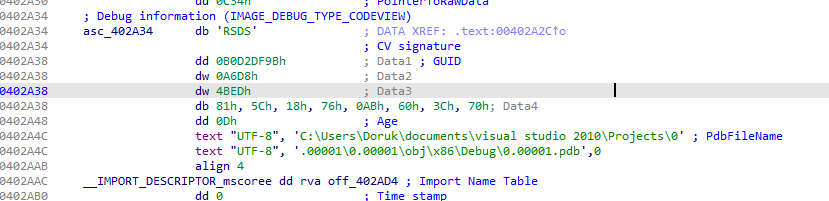
The strings window shows much of what was in the strings analysis. One string that is interesting is \_CorExeMain, similar to what was seen in Lecture10.exe which is jumped to from what IDA identifies as the start of the program. It appears to be a lot of random hex data.



Following CoreExeMain are a lot of string variables. After the variable “aTostring”, there are a lot of other hexadecimal data.

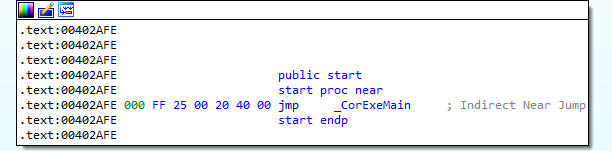


When clicking on the string that shows the path to the .pdb file, there appears to be a function that defines some variables as CV Signature and GUID, intended for parsing debugging information.



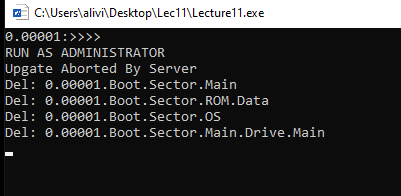
**What does IDA identify as the START of the program**

The function at 0x00402AFE.

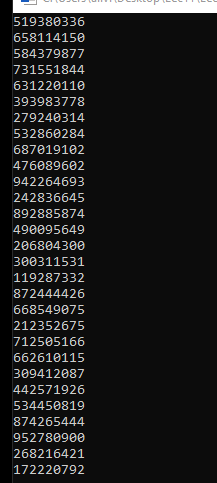


**Run the file and record your observations. This malware opens up a dialog box, what information is being displayed?**

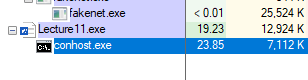
Initially, there were messages displayed that were expected by 0.00001.exe within cmd.exe. However, 0.00001.exe was not found within procmon nor Process Explorer.



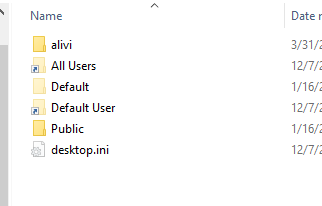
The box then printed “RESTART” (as expected) but then endlessly cycled through random numbers. ChatGPT could not find a discernable pattern in the numbers.



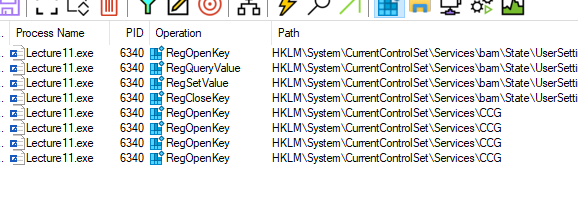
The console window was a child process of the file.



A new user was not created.



No registry activity was detected for persistence. However, the Lecture11 program did modify some startup services values. It also attempted to find the CCG service, which is a background service installed on some Dell computers.



When the VM was instructed to restart, it did so without issue contrary to my initial prediction.

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

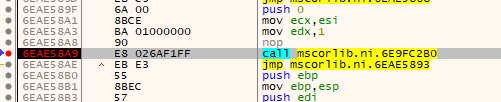
No network-based indicators were detected on FakeNet.

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

Aside from the console screen and the program running, I cannot think of any additional host-based indicators.

**What did this malware do?**

Within x32dbg, there were numerous locations found that indicated calls to mscorlib.ni. The function call every time a message was printed was mscorlib.ni.<memoryLocation>. These functions are part of the mscorlib.ni.dll file which is what I assume the program is going through. What is known is that whatever the program is doing it’s calling a function that returns a value that is translated into a 9-digit number.



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

All the time. The malware didn’t appear to alter the functionality of my VM in any way.

**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?**

As I am not 100% sure what this program is doing, I would suggest to management that we clean the system as we go. The VirusTotal findings suggested some backdoor functionality, but there wasn’t any network activity detected. I would present the option that whatever machine this file was detected on that we continue to quarantine it for analysis (just in case).

The obvious courses of action would also be to block the file signature on our IDS and IPS systems. I would also recommend a brief remedial cyber awareness training session for the staff and use this file as an example. Pointing out the poor icon quality in conjunction with the multiple file extensions within the name would be the focal point of the training for users to identify malware.

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

I would simply close the program and then delete the file.

**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/9769520/View>

<https://d2l.arizona.edu/d2l/le/1243099/discussions/threads/9761428/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 that it did. I walked through the debugging process to be absolutely sure that there wasn’t any data being stolen but it is still concerning that a file which was labeled and made to look like a .docx was a PE file.

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

From the posts I read, we all appeared to use the same techniques.

**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?**

I feel as if I was more thorough in my analysis than some other posts, but we all achieved the same results. It really depends on the priority that the company places on the analysis of this file. I do think that since Virus Total returned potential network activity that it would behoove the analyst to go in-depth into this file and see if there was potentially any intellectual property stolen.