CYT-250-Threat Investigation

Practical steps of using YARA by developing and running your own rules.

Elaborate by:

Leandro Delgado

Student Number: 114416241

Professor: Tatiana Outkina

CYT250 Lab 6 Winter 2025_Part2 - 3%

Individual work

Objectives: learn practical steps of using YARA by developing and running your own rules.

Pre-requisite: Lab6_Part1 is complete (Task1 and Task2). Proceed with the Task 3.

As usual, make screenshots to demonstrate your work and put them into MS Word document. Include Screen 0 and your comments.

Work on the AlienVault alert:

https://otx.alienvault.com/pulse/67d1748f2c0de9c0771afa40?utm_userid=tato12344&utm_medium=InProduct&utm_source=OTX&utm_content=Email&utm_campaign=new_pulse_from_following

The list of IOCs domain name, IP addresses, and file hash values which are associated with the threat. Use this data for your work. Do not forget that you can download IOCs as CSV file.

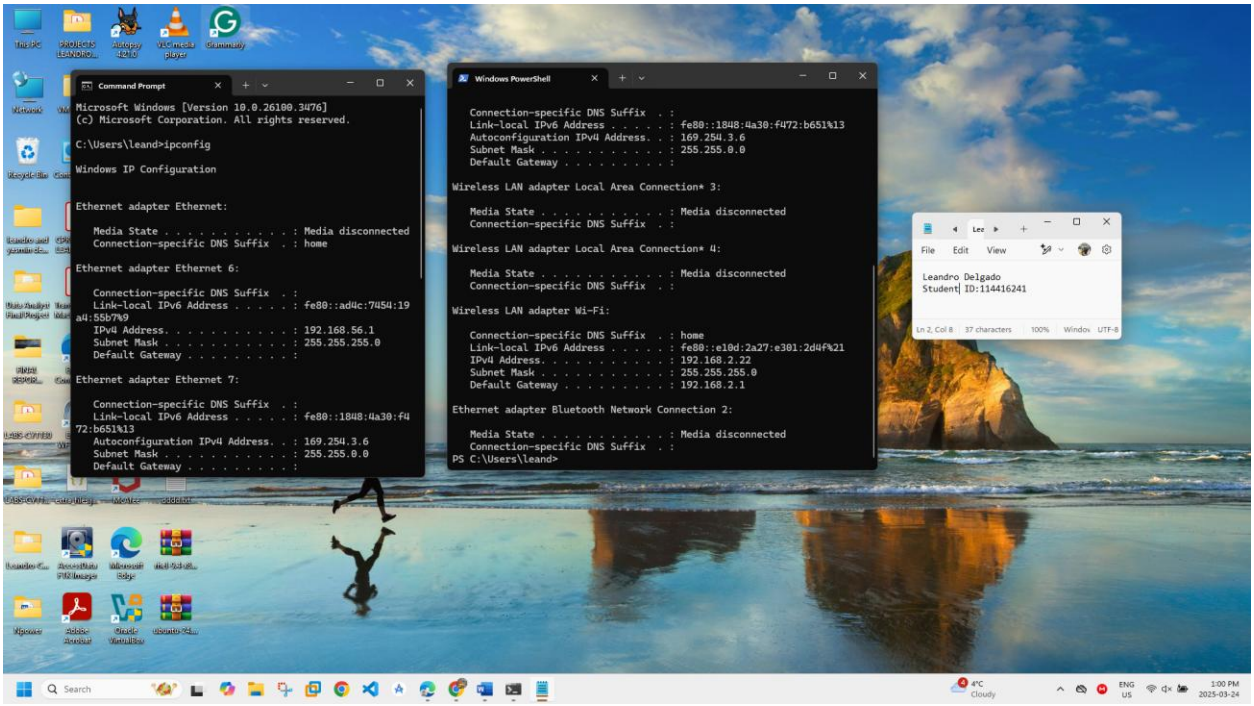

Figure 1. 0 Screen

So, your job is:

Step 1. Review the posted Alien Vault Alert.

Determine, what is the Yara Use Case in accordance with 4 known types (see the video below, it is not new, we watched it last week):

[YARA for Security Analyst | Crash Course]

There are 4 Use Cases presented in this video:

a) Identifying known malware signature
b) Detecting ransomware technique
c) Detecting packed executable
d) Matching obfuscated code

What Use Case apply to a given alert? Indicate and explain your choice.

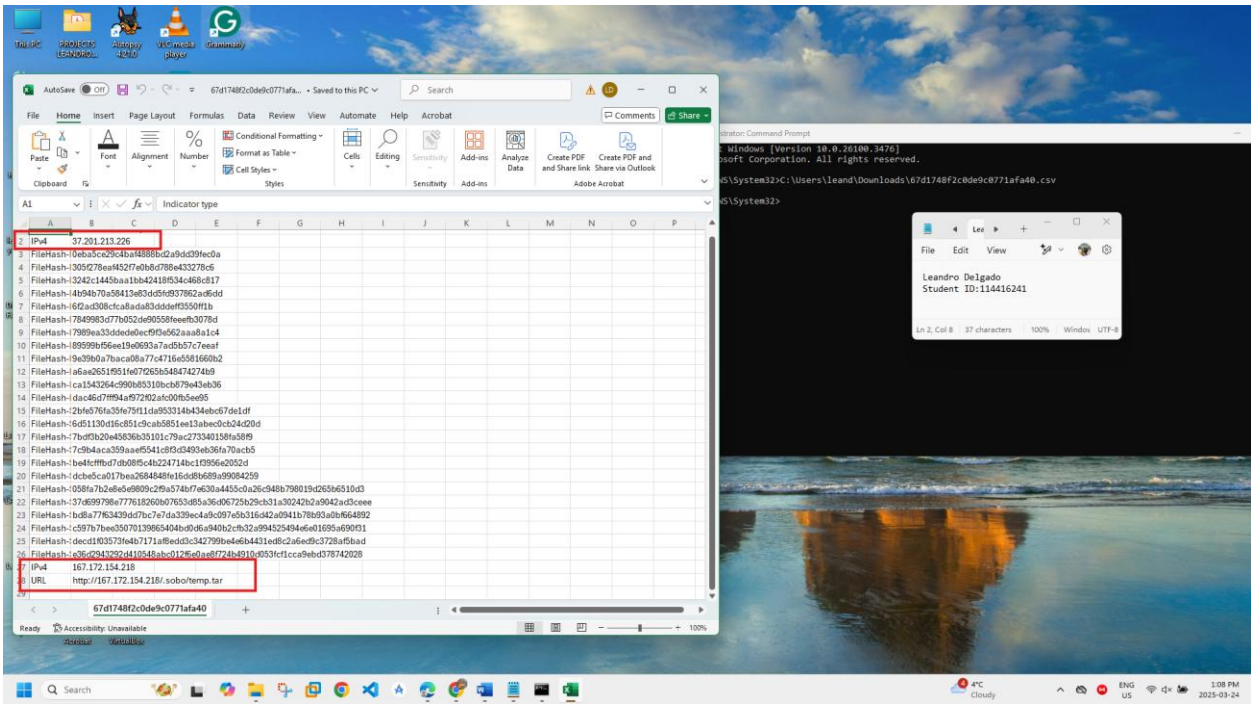

Figure 2. Cvs File Downloaded(Alient Vault site)

After downloading the CSV file, I set up a dedicated folder named *Rules* to organize and store the rules I will be creating throughout this lab. This will help keep my work structured and easily accessible as I progress through the completion of the lab.

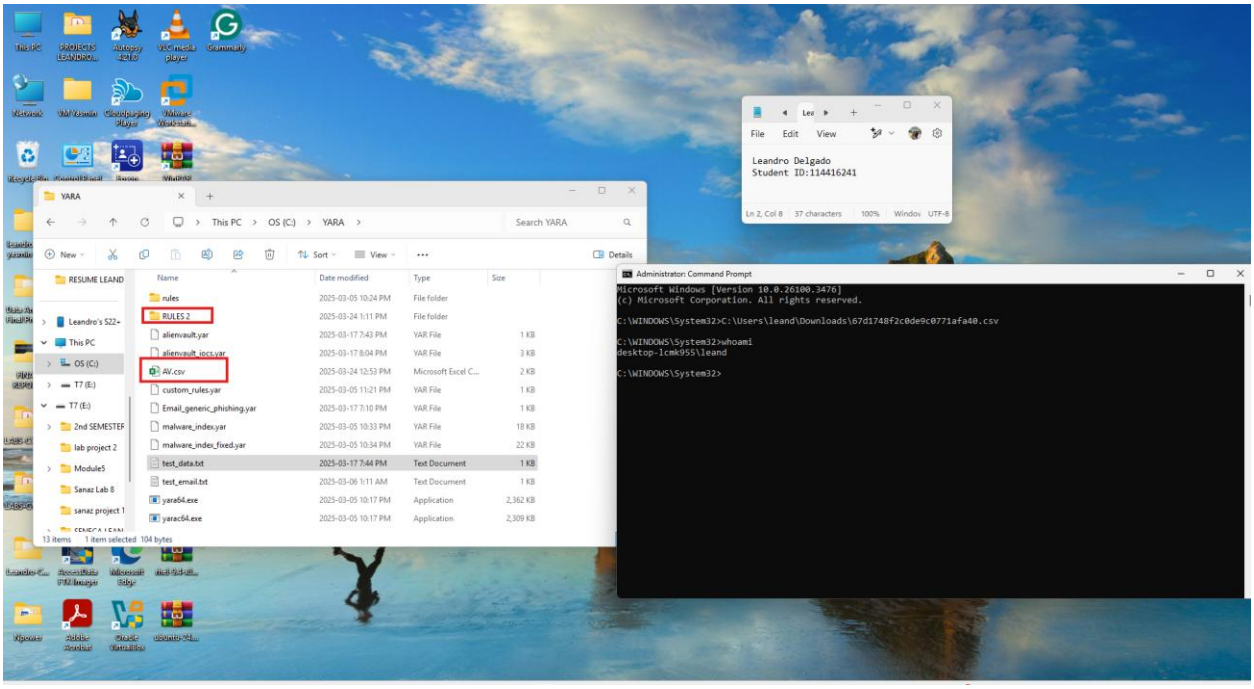

Figure 3. Folder rules created

I must first comprehend the scenario completely before setting a rule. My use case for this lab is Detecting Packed Executables. This will combine the use of Category 1: Rules for identifying malware on disk (packed) together with the dedicated rules to detect network IOCs since we are detecting file hashes and IP/URL addresses.The two-layer detection methodology is offered by the application of this rule. First, malicious packed executable files are looked for based on MD5, SHA1, and/or SHA256 hashes frequently associated with malware. Networking monitoring goes on in tandem with this, looking to detect known malicious IPs and URLs. Alerting is enabled if either condition matches so that early detection of both file and network-based threats can be achieved.

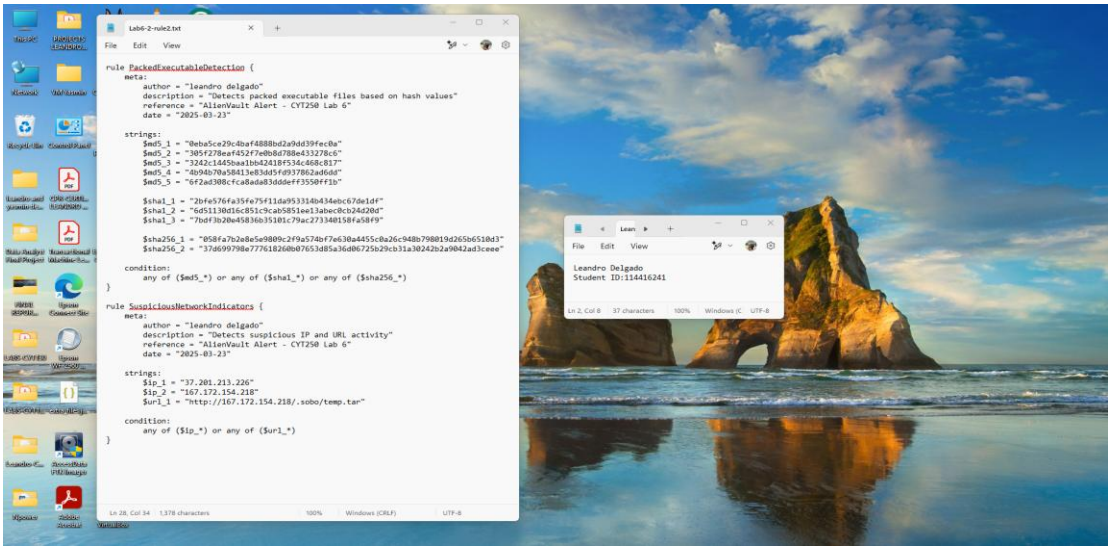

Figure 4.Personal rule Created

I prepared custom YARA rules for the detection of packed executables and suspicious network activities. My PackExDetection rule for packed executables uses MD5, SHA1 and SHA256 hashes to identify any malicious files. On the other hand, SuspiciousNetworkIndicators rule can able to flag known malicious IP addresses and URLs. I would formulate a rule with metadata, detection parameter and clearly established conditions to trigger alerts. I also put all these rules in a dedicated Rules folder for easy access and management of my work. I prepared custom YARA rules for the detection of packed executables and suspicious network activities. My PackExDetection rule for packed executables uses MD5, SHA1 and SHA256 hashes to identify any malicious files. On the other hand, SuspiciousNetworkIndicators rule can able to flag known malicious IP addresses and URLs. I would formulate a rule with metadata, detection parameter and clearly established conditions to trigger alerts. I also put all these rules in a dedicated Rules folder for easy access and management of my work.

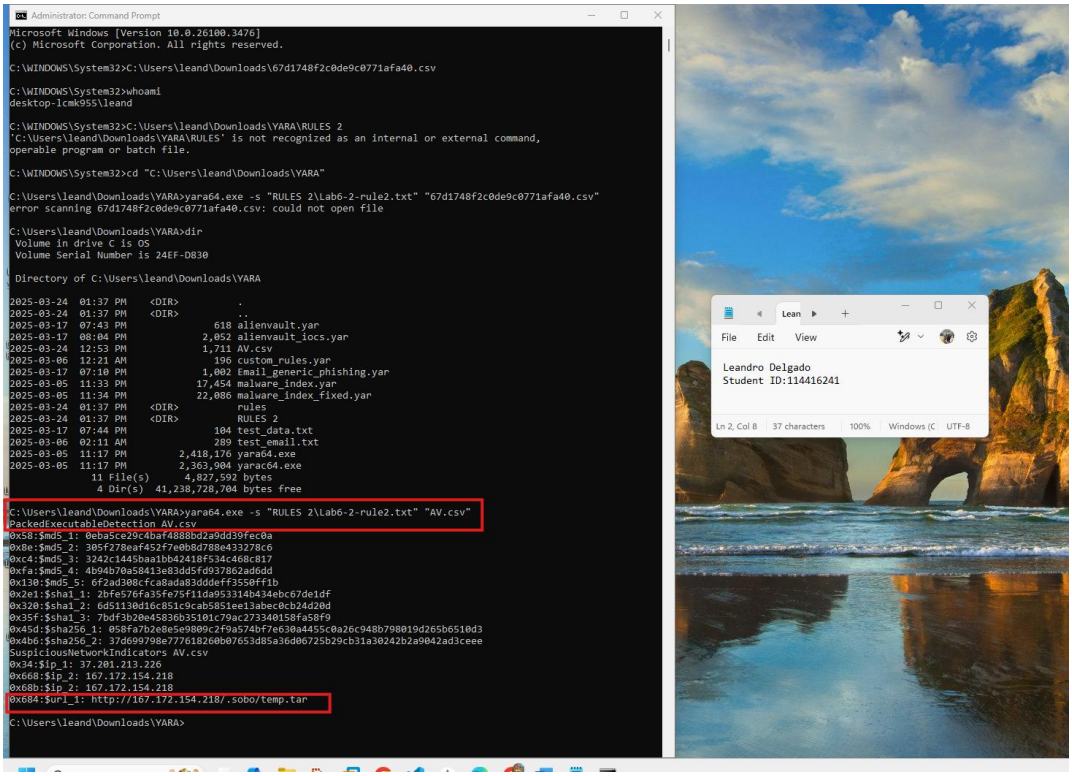

Figure 5, Full run of Rule

The command executed a YARA scan using the rule file Lab6-2-rule2.txt against AV.csv, successfully detecting potential threats. The PackedExecutableDetection rule flagged multiple file hashes (MD5, SHA1, and SHA256), indicating the presence of potentially packed or malicious executables. Meanwhile, the SuspiciousNetworkIndicators rule identified two suspicious IP addresses and a flagged URL (http://167.172.154.218/.sobo/temp.tar), suggesting possible malicious network activity. These results confirm that the implemented rules are effectively identifying both file-based and network-based threats, enhancing threat detection capabilities.

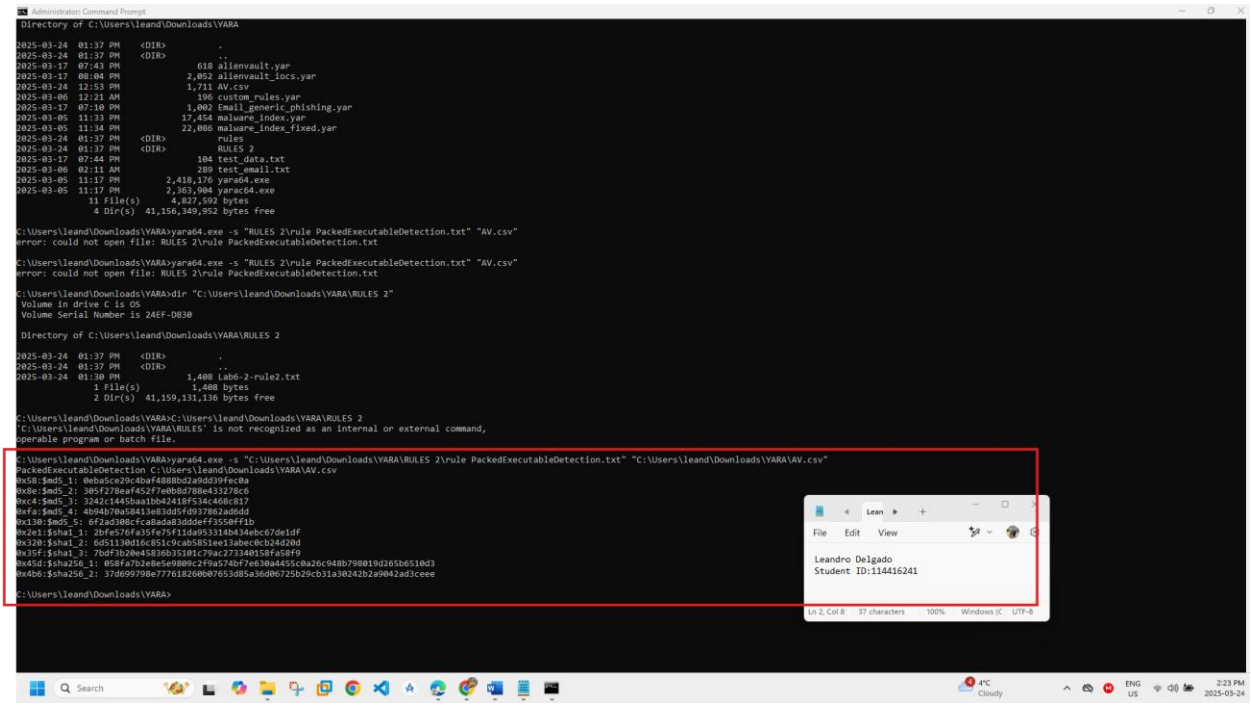

Figure 6`PackedExecutableDetection` rule

At first, I ran into an issue executing my **PackedExecutableDetection** rule because the system couldn't locate the rule file in the **RULES 2** directory. After troubleshooting, I realized it was likely a path-related issue and adjusted the command accordingly.

Once I corrected the path, the YARA scan executed successfully, detecting multiple hashes from **AV.csv**. The output displayed MD5, SHA1, and SHA256 values, confirming that the rule effectively identifies packed or potentially malicious executables. This test helped me validate that my rule is working as intended, correctly flagging files based on their hash signatures.

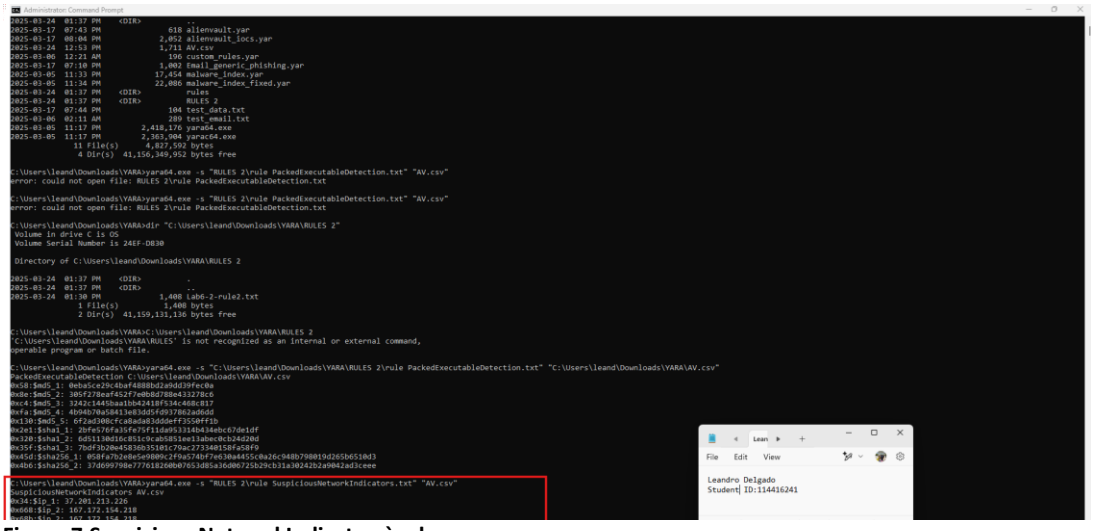

Figure 7.Suspicious NetworkIndicators` rule

I decided to run the Suspicious Network Indicators rule separately to see how it performs on its own. The execution was successful, detecting two flagged IP addresses (167.172.154.218 and 137.201.213.226) along with a suspicious URL (http://167.172.154.218/.sobo/temp.tar).

Seeing these results confirmed that my rule is working as expected, correctly identifying potential network threats based on known malicious indicators. Running each rule individually helped me better understand their behavior and ensured they trigger alerts when needed.

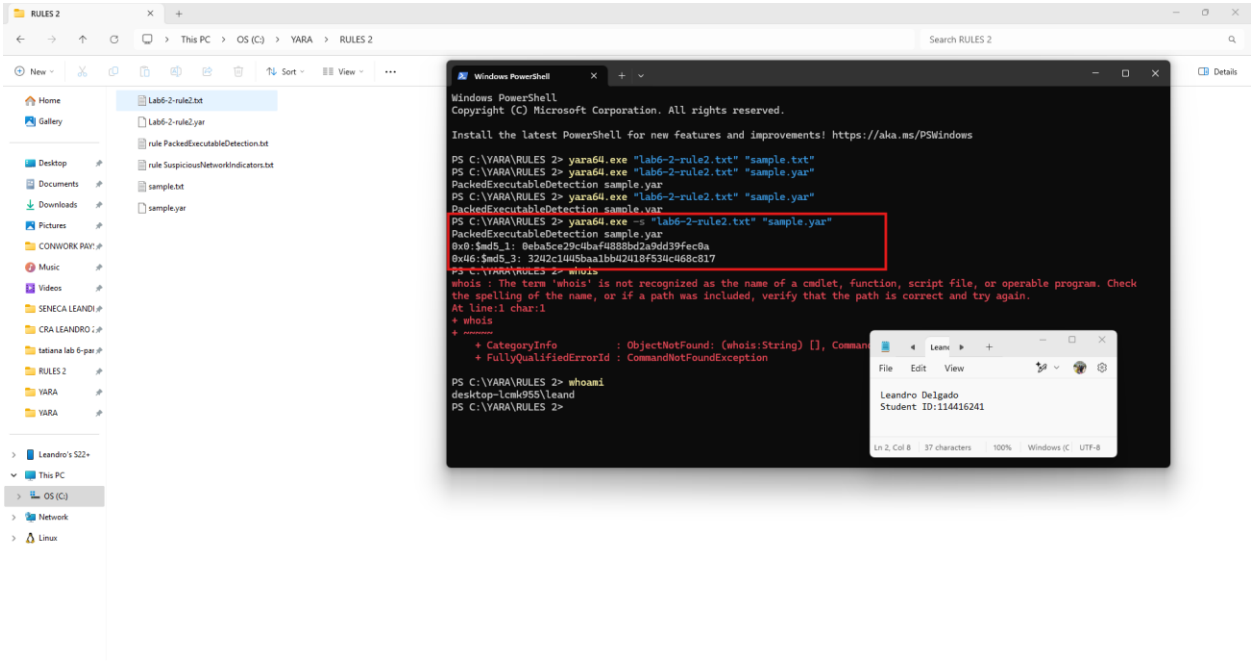

Figure 8. Creation of own sample to run.

In this test, I successfully ran my PackedExecutableDetection rule with YARA against a sample file, detecting specific hashes and thereby affirming the intended operation of the rule. I went ahead to create another sample file for the test for the further validation of my rules, thereby assuring the consistency of detection. Sadly, when trying to execute the whois command in PowerShell, I got an error stating that it is not recognized. It means likely that whois is not installed by default in Windows. To remedy the situation, I might use any of the following options: install any WHOIS client, use a Linux terminal, or find an online WHOIS lookup tool. The whois error notwithstanding, the testing of several sample files has helped me fine-tune and ascertain the working of my YARA rule.

Summary

This lab provided valuable hands-on experience in creating and testing YARA rules for detecting malicious files and suspicious network indicators. By structuring and executing my PackedExecutableDetection and SuspiciousNetworkIndicators rules separately, I was able to understand how each detection mechanism functions. Running tests with different sample files allowed me to refine my rules and confirm their accuracy in identifying known threats.

Additionally, troubleshooting errors, such as the missing whois command in PowerShell, reinforced the importance of using the right tools and environments for threat analysis. Overall, this lab enhanced my understanding of YARA rule creation, hash-based detection, and network-based threat identification, strengthening my skills in cybersecurity threat hunting.