

Assignment – 6

Name : Ishan Aakash Patel

Student ID : 146151238

Course : CYT-245

YARA Malware analysis Tool

YARA is a powerful and versatile tool used in malware research and detection. It allows security researchers and analysts to create pattern-based descriptions of malware families or specific malicious files. These descriptions, called YARA rules, consist of textual or binary patterns along with logical conditions. YARA can scan files, processes, or even network traffic to identify content matching these rules. It's widely used in threat hunting, incident response, and malware analysis to detect known malicious patterns or behaviors. YARA's flexibility allows it to be integrated into various security tools and workflows, making it an essential component in many cybersecurity operations. Its rule-based approach enables rapid development and deployment of detection mechanisms for new threats, allowing security teams to quickly respond to emerging malware variants and attack techniques.

Step 1 : Installing yara

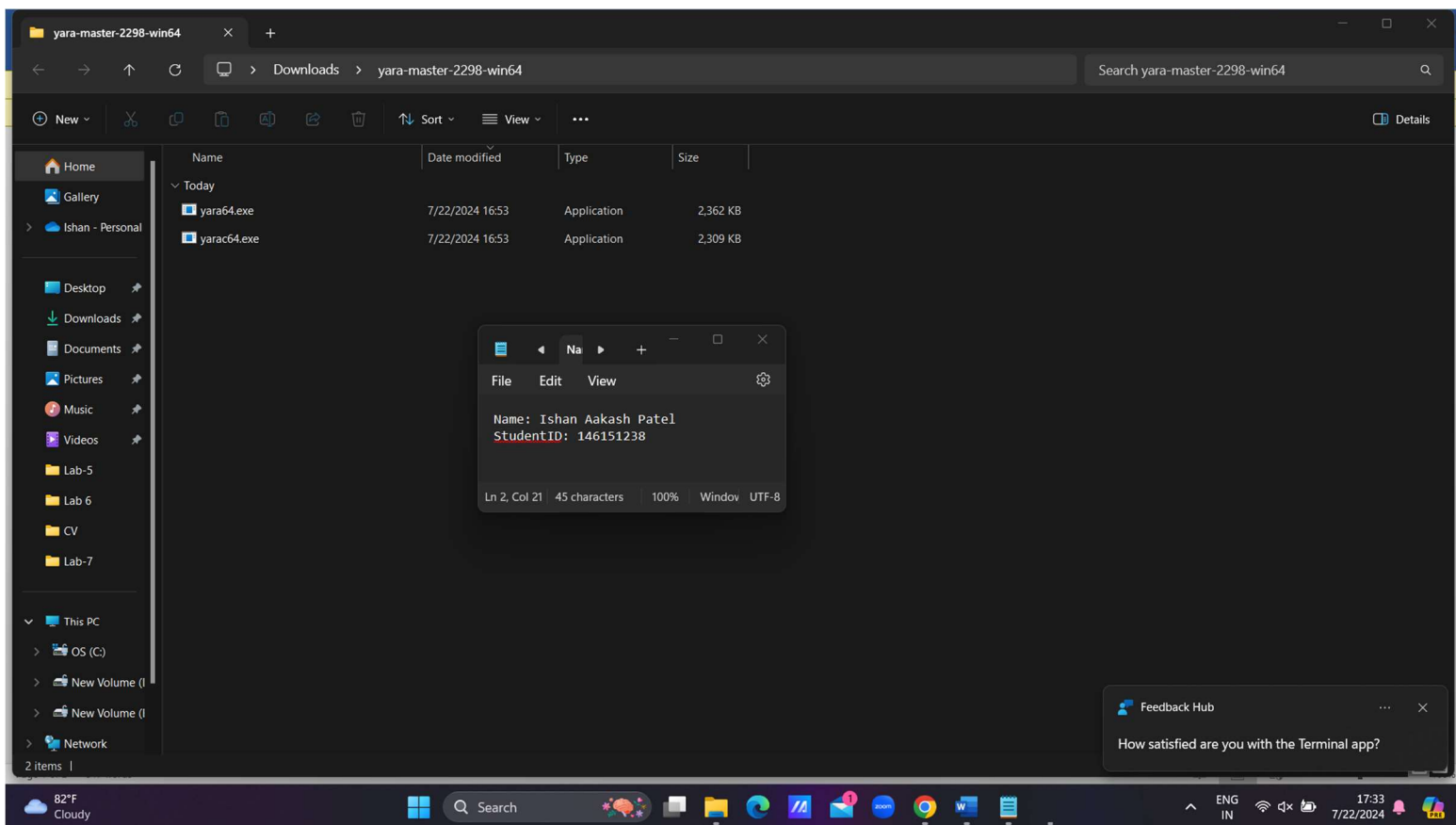


Figure 1 Downloaded the .exe files for windows

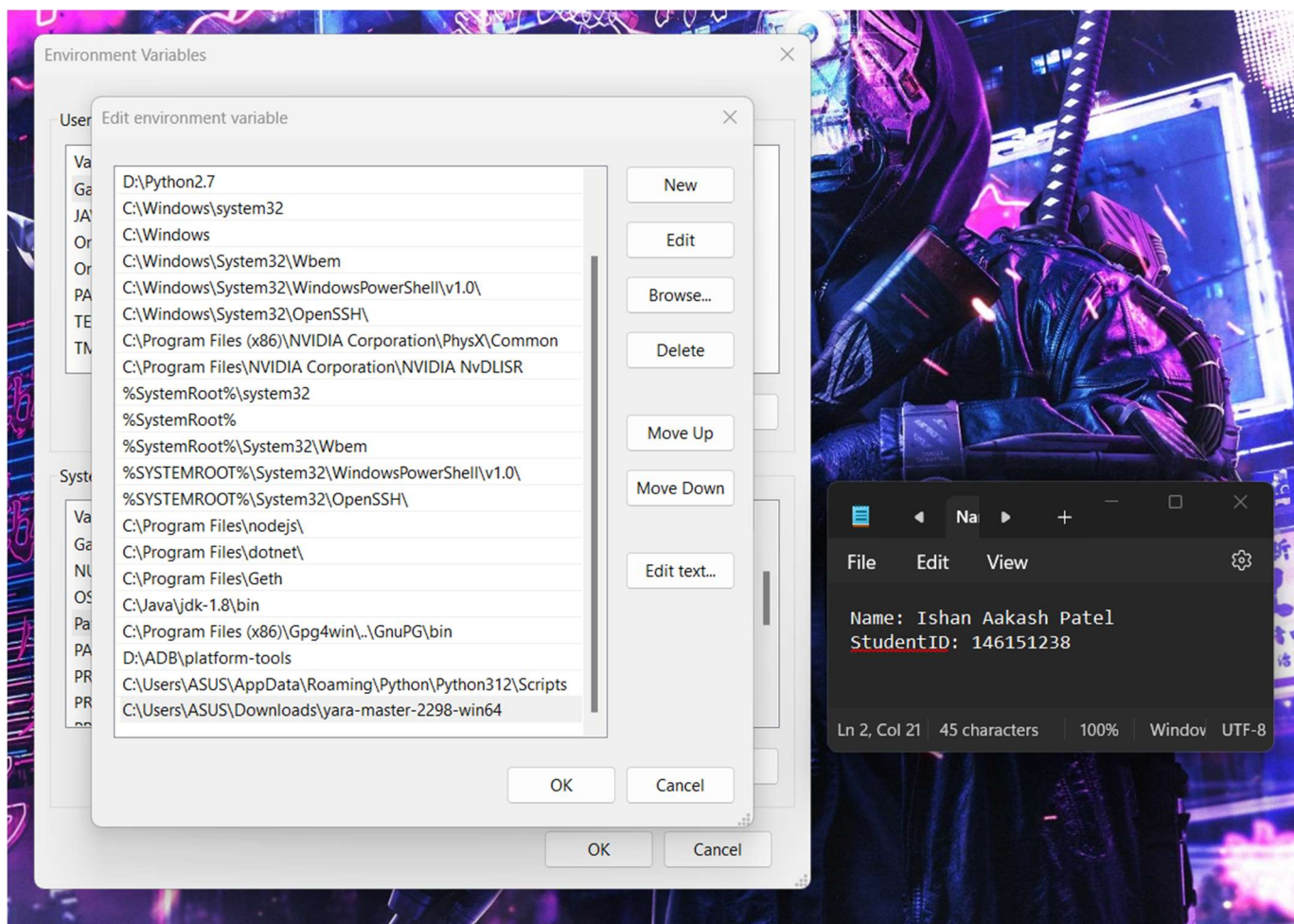


Figure 2 : Add the path to environment variable

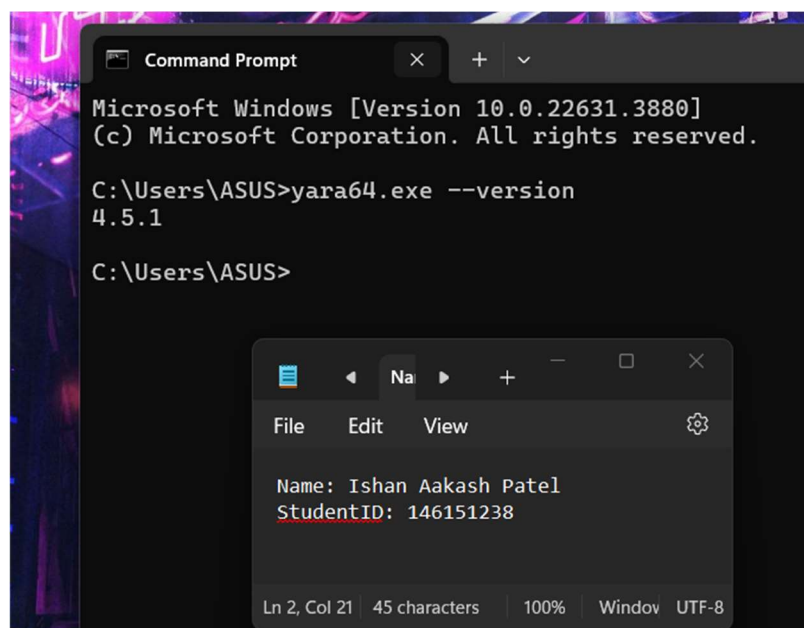


Figure 3 : Installation verification

Step 2 : Work on Yara rules available at github

Basics of YARA

YARA Overview: YARA is a tool used to identify and classify malware by creating rules based on patterns and characteristics observed in known samples. It is highly configurable and allows for both simple and complex matching.

YARA Rule Structure: A YARA rule typically includes the following sections:

1. Rule Name:

- A unique identifier for the rule.

Code : rule my_first_rule

2. Meta Section:

- Provides metadata about the rule, such as its description, author, and references.

Code:

meta:

description = "This is a test rule"

author = "John Doe"

date = "2024-07-22"

3. Strings Section:

- Defines the strings to look for in the target file. These can be text strings, hexadecimal patterns, or regular expressions.

Code:

strings:

\$text_string = "Hello, world!"

\$hex_string = { E2 34 A1 }

\$regex = /[a-zA-Z0-9]+/

4. Condition Section:

- Specifies the conditions under which the rule matches. This section can use logical operators, relational operators, and other YARA-specific functions.

Code:

condition:

\$text_string or \$hex_string or \$regex

1. Analysis of CVE-2018-4878.yar

This YARA rule, named "crime_ole_loadswf_cve_2018_4878", is designed to detect exploits targeting CVE-2018-4878, a vulnerability in Adobe Flash Player. Here's a detailed breakdown of the rule:

Purpose:

The rule aims to identify files potentially exploiting CVE-2018-4878, a remote code execution vulnerability in Adobe Flash Player.

Meta Information:

- The vulnerability allows for remote code execution through a use-after-free exploit.
- It affects Adobe Flash versions 28.0.0.137 and earlier.
- The exploit is typically weaponized by embedding it in Microsoft Office documents.
- It's associated with alleged North Korean threat actors.

Detection Mechanism:

The rule looks for several key indicators:

1. Flash Object Headers:

- Checks for "rdf:RDF" which is typically found in Flash object headers.
- Looks for "Adobe Flex" in the application type title.

2. Specific Strings Related to the Exploit:

- "URLRequest" and "URLLoader": Common ActionScript classes used for loading external content.
- "loadswf" and "myURLRequest": Likely custom function and variable names used in the exploit.

3. Development Artifacts:

- Checks for a specific PDB (Program Database) path, which could indicate the development environment of the exploit.

Conditions:

The rule triggers if either of these conditions are met:

1. The Flash header, title, and at least 3 of the loader strings are present.
2. The PDB path, Flash header, and at least one loader string are found.

This multi-condition approach allows for flexibility in detection, catching both obvious and slightly obfuscated variants of the exploit.

2. Analysis of EMAIL_Cryptowall.yar

This YARA rule set contains two rules: "CryptoWall_Resume_phish" and "docx_macro". Let's analyze each:

Rule: CryptoWall_Resume_phish

Purpose:

This rule is designed to detect phishing emails potentially delivering CryptoWall ransomware, specifically those disguised as job application emails with resumes.

Detection Mechanism:

The rule looks for combinations of three types of strings in email content:

1. Greeting:

- Checks for phrases like "my name is"

2. File References:

- Looks for mentions of attached resumes, using various phrasings

3. Salutations:

- Checks for common closing phrases used in job applications

Conditions:

The rule triggers if it finds at least one string from each category (greeting, file reference, and salutation).

This approach helps identify phishing emails that follow a common pattern used by CryptoWall distributors, mimicking job application emails to trick recipients into opening malicious attachments.

Rule: docx_macro

Purpose:

This rule is designed to detect Microsoft Word documents (.docx) that contain VBA macros, which are often used to deliver malware.

Detection Mechanism:

The rule checks for two key elements:

1. Document Header:

- Looks for "PK" at the beginning of the file, which is the signature for ZIP archives (DOCX files are essentially ZIP archives)

2. VBA Project:

- Checks for the presence of "word/vbaProject.bin", which indicates the document contains VBA macros

Conditions:

The rule triggers if both the correct header is present and the VBA project file is detected.

This rule is particularly useful for identifying potentially malicious Word documents that may be used as part of the CryptoWall distribution chain, complementing the email phishing detection.

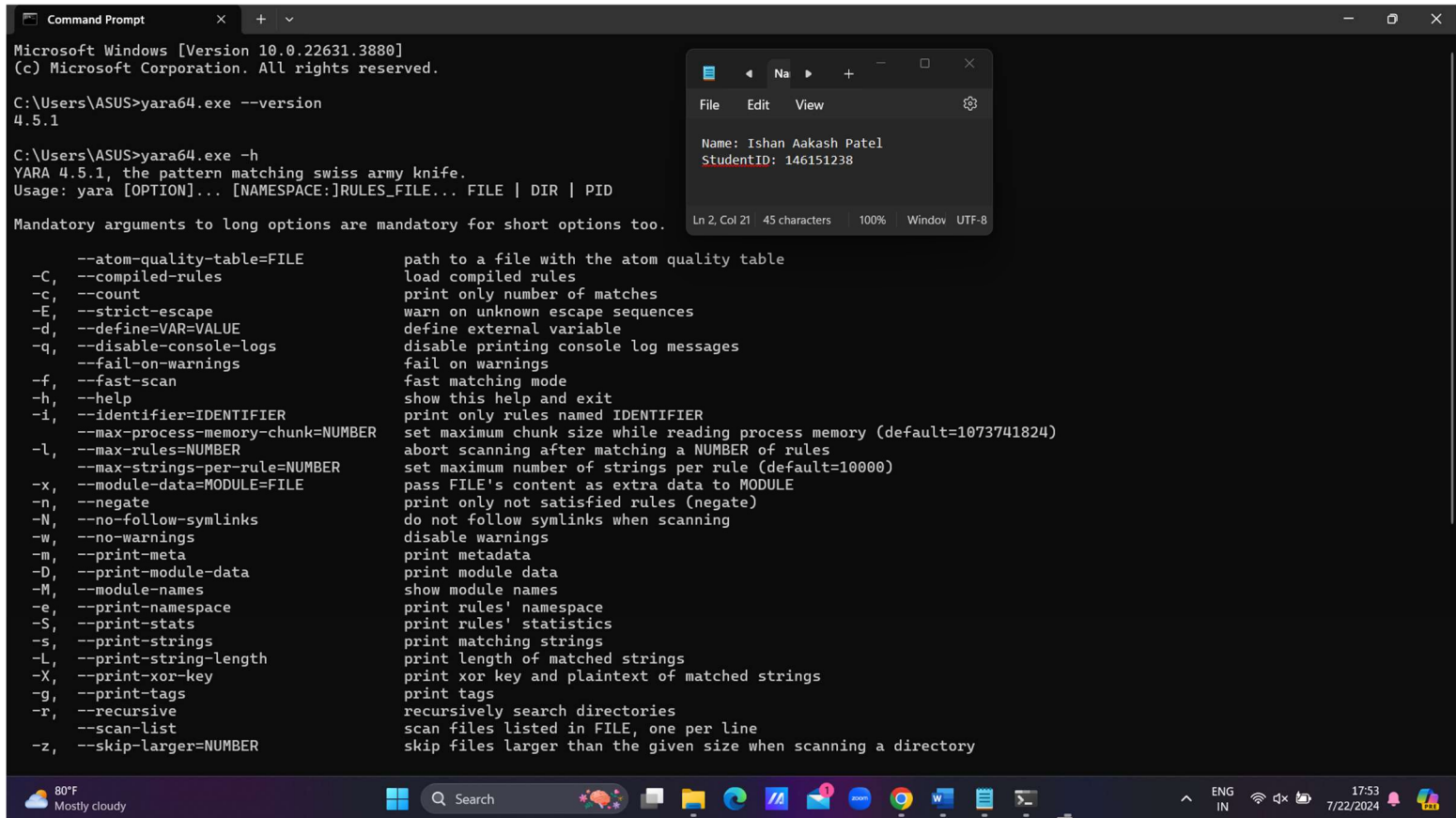
These detailed analyses provide a comprehensive understanding of how these YARA rules work to detect potential threats related to CVE-2018-4878 and CryptoWall phishing campaigns.

Inshort, **For CVE-2018-4878 Rule:** This rule is designed to detect remote code execution vulnerabilities in Adobe Flash objects embedded in Office documents. It checks for specific RDF and Adobe Flex markers, URL handling strings, and a particular file path used during Flash object creation.

For CryptoWall Rule: This rule targets phishing emails by looking for specific phrases commonly used in fake job application emails. It identifies introductory phrases, mentions of attached resumes, and polite sign-offs to detect malicious emails attempting to distribute CryptoWall ransomware.

Part – 3 : Work with Yara documentation to see in more details how to run the rules

Step 1: Open the specific directory where all the files of yara as well as sample file are present in command prompt.



```
Microsoft Windows [Version 10.0.22631.3880]
(c) Microsoft Corporation. All rights reserved.

C:\Users\ASUS>yara64.exe --version
4.5.1

C:\Users\ASUS>yara64.exe -h
YARA 4.5.1, the pattern matching swiss army knife.
Usage: yara [OPTION]... [NAMESPACE:]RULES_FILE... FILE | DIR | PID

Mandatory arguments to long options are mandatory for short options too.

  --atom-quality-table=FILE      path to a file with the atom quality table
  -C, --compiled-rules          load compiled rules
  -c, --count                   print only number of matches
  -E, --strict-escape           warn on unknown escape sequences
  -d, --define=VAR=VALUE        define external variable
  -q, --disable-console-logs    disable printing console log messages
  --fail-on-warnings            fail on warnings
  -f, --fast-scan              fast matching mode
  -h, --help                   show this help and exit
  -i, --identifier=IDENTIFIER  print only rules named IDENTIFIER
  --max-process-memory-chunk=NUMBER
                                set maximum chunk size while reading process memory (default=1073741824)
  -l, --max-rules=NUMBER       abort scanning after matching a NUMBER of rules
  --max-strings-per-rule=NUMBER
                                set maximum number of strings per rule (default=10000)
  -x, --module-data=MODULE=FILE
                                pass FILE's content as extra data to MODULE
  -n, --negate                 print only not satisfied rules (negate)
  -N, --no-follow-symlinks     do not follow symlinks when scanning
  -w, --no-warnings            disable warnings
  -m, --print-meta             print metadata
  -D, --print-module-data      print module data
  -M, --module-names          show module names
  -e, --print-namespace       print rules' namespace
  -S, --print-stats            print rules' statistics
  -s, --print-strings          print matching strings
  -L, --print-string-length    print length of matched strings
  -X, --print-xor-key          print xor key and plaintext of matched strings
  -g, --print-tags            print tags
  -r, --recursive             recursively search directories
  --scan-list                  scan files listed in FILE, one per line
  -z, --skip-larger=NUMBER    skip files larger than the given size when scanning a directory
```

Now save the CryptoWall_Resume_phish and also I have created a sample phishing email file.

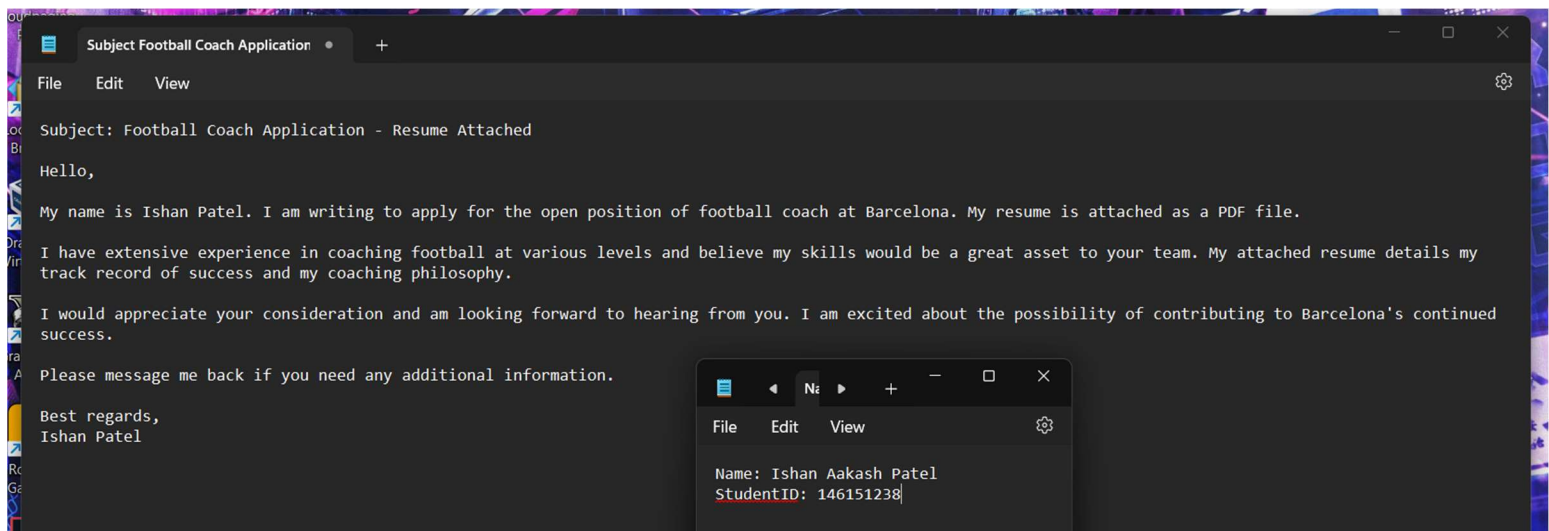


Figure 4 : Sample file

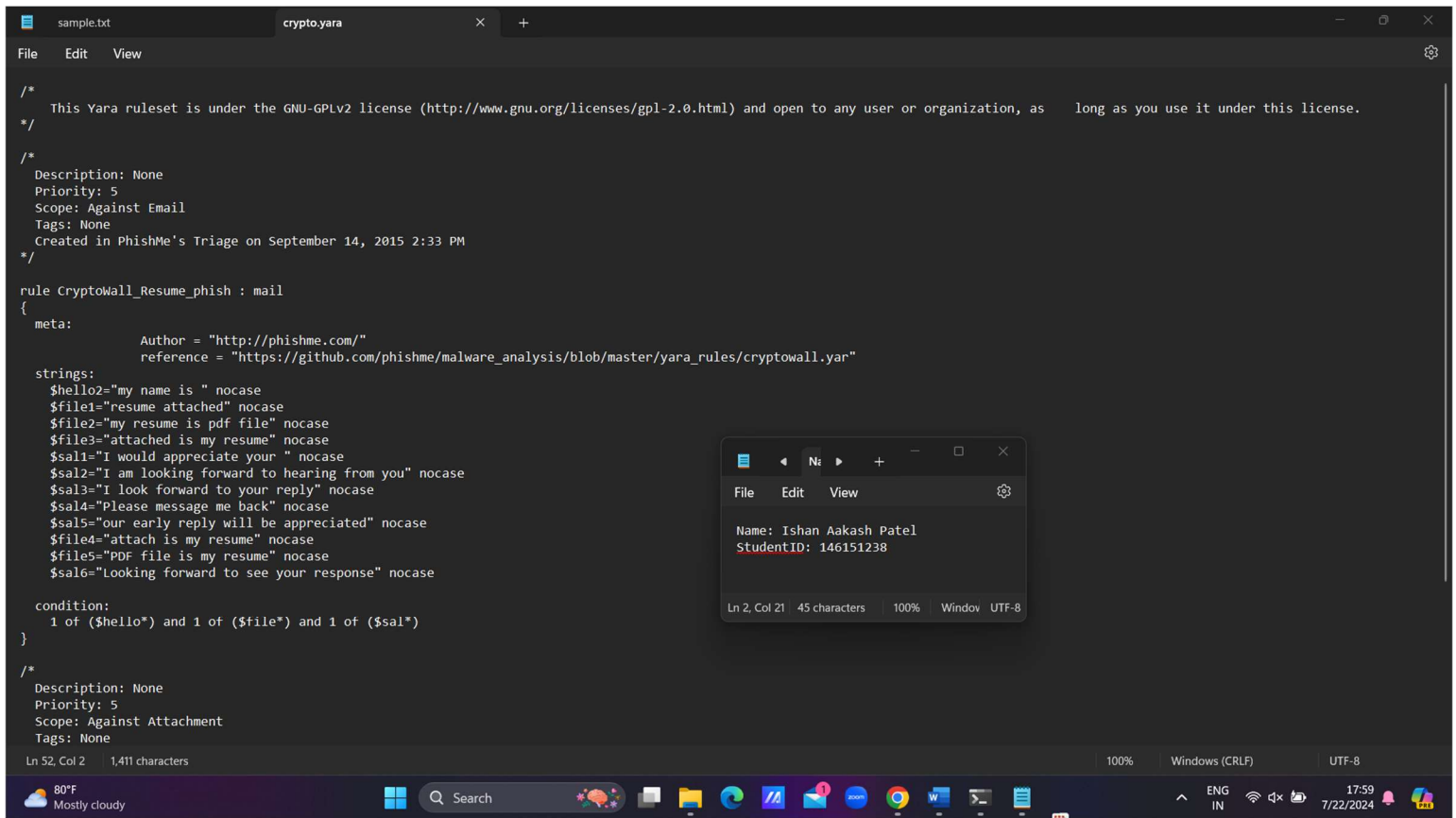
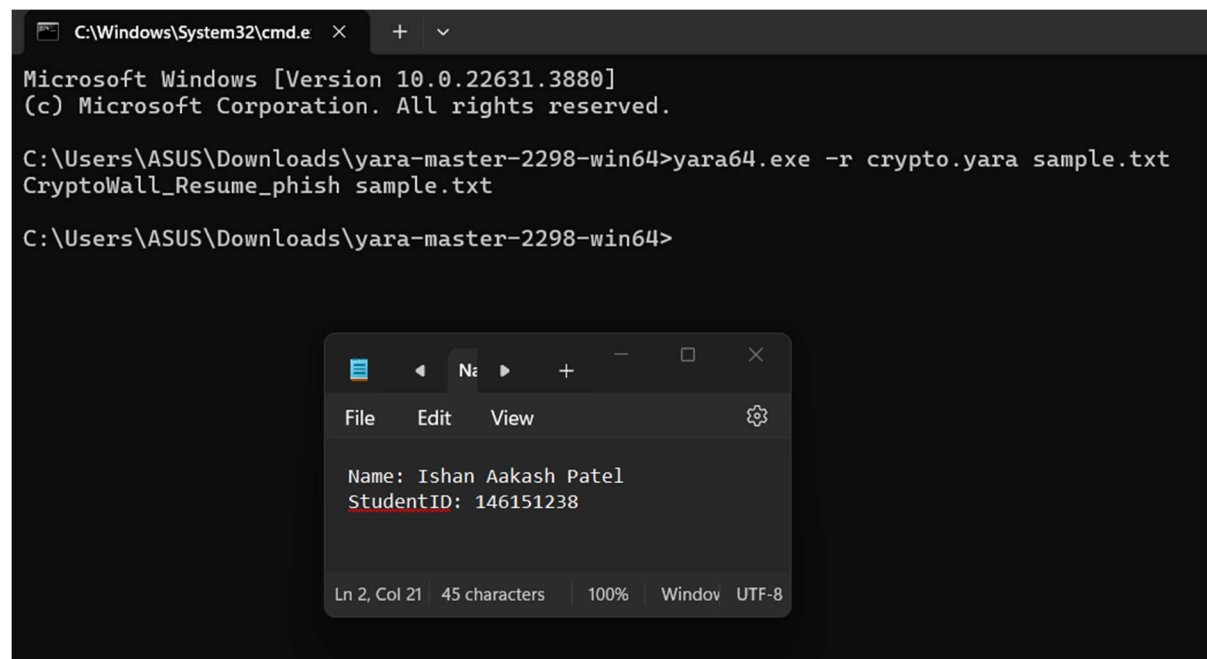


Figure 5 : The cryptowall_resume_phish yara file

Now run the following command to run yara on sample file to detect the phishing email...

Command : yara64.exe -r CryptoWall_Resume_phish.yar sample.txt



We can see in the output the name of the rule and detected file.

Here is another Command through which we can get specific strings which are matched from the rule...

Command : yara64.exe -s CryptoWall_Resume_phish.yar sample.txt

```
Send bug reports and suggestions to: vmalvarez@virustotal.com.

C:\Users\ASUS\Downloads\yara-master-2298-win64>yara64.exe -s crypto.yara sample.txt
CryptoWall_Resume_phish sample.txt
0x43:$hello2: My name is
0x26:$file1: Resume Attached
0x1a2:$sal1: I would appreciate your
0x24c:$sal4: Please message me back

C:\Users\ASUS\Downloads\yara-master-2298-win64>
```

File Edit View

Name: Ishan Aakash Patel
StudentID: 146151238

Ln 2, Col 21 45 characters 100% Window UTF-8

79°F Mostly cloudy

Search

File Explorer Edge VS Code Mail Zoom Chrome Word Calendar