Skills for Hire Atlantic

Cybersecurity – Assignment 2

Overview

This assignment consists of **two main questions**, each with specific topics. You are required to complete **one** question **only** and the total Marks is 20

The options are:

Option 1: Question 1a: Cryptography **and** Question 1b: Malware Analysis
OR
Option 2: Question 2: Log File Analysis

You may either complete both Question 1a **and** Question 1b, **or** you can choose to complete only Question 2.

Please ensure that you understand the requirements and topics covered in each option before making your choice.

Instructions

- Create a Google Colab or Jupyter notebook for each question attempted.
- Label each notebook with the corresponding title as mentioned in the assignment.
- Write clean, well-documented code with comments that explain your approach and logic.
- The breakdown of marks for each question is indicated between square brackets.

Note: This assignment will count for 40% of your Assignments grade.

Due date: September 8th, 2024

Submission

Submit through: https://formesign.com/sm/WrHCnP8gO

Submission Requirements

File format: **A compressed ZIP (.zip) file** containing your Python Notebook (.ipynb) file File name:
CS_Group_Assignment2_FirstName_LastName_EmailAddress.ZIP

MAKE SURE TO REPLACE Group with your Group color, FirstName with your first name, LastName with your last name and EmailAddress with your email address.

Hints

Start early. There are many parts to this assignment and it would be very difficult if left to the last minute.
Don’t reinvent the wheel. Feel free to use the examples covered in class. If you get stuck, reach out to your TA for help! Do not spend hours without asking for help. Good luck!

Question 1a: Cryptography. [10]

File for question 1a – **crypto.txt**

i. Create a Google Colab notebook titled "**Cryptography**" and install the necessary cryptographic modules. [1]
ii. Generate an RSA key pair (public and private keys) with a key size of **2048** bits. [2]
iii. Encrypt the contents of the crypto.txt file using **AES** encryption with a randomly generated AES key (**16 bytes**). [2]
iv. Encrypt the randomly generated AES key using **RSA** encryption with the generated public key. [2]
v. Generate a **digital signature** for the encrypted data using the **RSA private key**. [2]
vi. Save the **encrypted data**, **encrypted AES key**, and the **digital signature** into separate files and name the files as "crypto_encrypted.txt", "aes_key_encrypted.bin", and "signature.bin" respectively. [1]

Question 1b: Malware Analysis. [10]

File for question 1b – **yara.txt**

i. Create a Google Colab notebook titled "**Malware**" and install the **YARA python library**. [1]
ii. Write a YARA rule to match the string "**malware**" anywhere in the file. Save the rule as **malware_rule.yar**. [2]
iii. Write a YARA rule to match the string "**virus**" anywhere in the file. Save the rule as **virus_rule.yar**. [2]
iv. Compile each rule individually and **verify** the number of matches for each rule. [2]
v. **Count** the **occurrences** of the “**malware**” and “**virus**” in yara.txt file. [3]

OR

Question 2: Log File Analysis. [20]

File for question 2 – **log_files.txt** Below is the explanation of the log file fields.

Column	Explanation
Client IP	The IP address of the client (user or device) making the request.
Identity	Often used to indicate the identity of the user determined by HTTP authentication. It is usually a dash (-) if not used or unavailable.
User ID	Typically represents the user ID of the person making the request. This is often not used, represented by a dash (-).
Timestamp	The date and time when the request was made, formatted as day/month/year:hour:minute:second timezone.
HTTP Method	The HTTP method used for the request, such as GET, POST, PUT, DELETE, etc.
Resource	The path to the resource requested by the client.
HTTP Version	The version of the HTTP protocol used for the request.
Status Code	The HTTP status code returned by the server in response to the request.
Bytes Transferred	The size of the response returned by the server, measured in bytes.

a. Create a Google Colab notebook titled "**LogAnalysis**" and install the necessary modules. [1]
b. Define the **regex** pattern for the log entries. [2]
c. **Parse** the log file and print only the **first 5** entries. [2]
d. Import **defaultdict** from collections module to handle counting of status codes. [1]
e. Define a function that reads log entries from the specified file and extracts the **HTTP status code** from each entry. [2]
f. Define a function to **count the occurrences** of each status code (200, 404, 500). [2]
g. Plot the above counts in a **bar chart**. [2]
h. Define a function to get the **percentage of successful requests.** [2] *Hint: 200 status code denotes successful requests.*
i. Define a function to get the **top 3 URLs** and their respective **count**. [3]
j. Define a function to get the **IP address** and the **number of requests** made by the **most active client.** [3]