

A blue text on a white background

Description automatically generated

## Round 1 report

# Contest Date: - 27th January 2024

|  |  |
| --- | --- |
| CT ID | DT20234670714 |
| Name | Koilakonda Madhu |
| College/University | Rajeev Gandhi Memorial College of Engineering and Technology |
| City | Nandyal |
| Challenges solved & the total score | 4 |
| Anything else that you want us to know | yes |

**(Copy & paste the table x times if you solved x challenges)**

|  |
| --- |
| **Challenge Title: BABY SHARK DODODO** |
| **Flag: HQ8FLAG{71f84063c2a57b5142673a29b8d04f9}** |
| **Approach (Step by Step):**   1. Determine Current Directory: Get the current directory path. 2. Construct File Paths: Construct the file paths for the network capture file (pcap) and the SSL key file relative to the current directory. 3. Import Necessary Libraries: Import the required libraries (pyshark and os). 4. Define Function to Analyze TLS Traffic: Define a function to analyze TLS traffic and search for the flag. 5. Load Network Capture File: Load the network capture file using pyshark.FileCapture. 6. Iterate Through Packets: Iterate through each packet in the capture. 7. Check TLS Handshake Packets: Check if the packet is a TLS handshake packet. 8. Check for Flag: Check if the packet contains the flag "HQ8{EnterHQ8KeyHere}". 9. Print Flag and Packet Details: If the flag is found, print the packet containing the flag along with its details. 10. Close Capture: Close the network capture file. 11. Define Main Function: Define the main function to execute the analysis. 12. Call Main Function: Call the main function when the script is executed. 13. Save and Run Script: Save the script and run it using the Python interpreter. |

|  |
| --- |
| **Challenge Title: SCRAP SCRUBBER** |
| **Flag: HQ8FLAG{3b3dfb6933f178688074513283dcacd}** |
| **Approach (Step by Step):**   1. Determine Current Directory: Get the current directory path. Construct File Paths: Construct the file paths for the network capture file (pcap) and the SSL key file relative to the current directory. 2. Import Necessary Libraries: Import the required libraries (pyshark and os). 3. Define Function to Analyze TLS Traffic: Define a function to analyze TLS traffic and search for the flag. 4. Load Network Capture File: Load the network capture file using pyshark.FileCapture. Iterate Through Packets: Iterate through each packet in the capture. 5. Check TLS Handshake Packets: Check if the packet is a TLS handshake packet. 6. Check for Flag: Check if the packet contains the flag "HQ8{EnterHQ8KeyHere}". 7. Print Flag and Packet Details: If the flag is found, print the packet containing the flag along with its details. 8. Close Capture: Close the network capture file. 9. Define Main Function: Define the main function to execute the analysis. 10. Call Main Function: Call the main function when the script is executed. 11. Save and Run Script: Save the script and run it using the Python interpreter. |

|  |
| --- |
| **Challenge Title: CLOCK AND DAGGER** |
| **Flag: HQ8FLAG{3f6ee0d17505b3fd85060a04ffd4d61}** |
| **Approach (Step by Step):**   1. Analyze the Challenge: Break down the challenge into smaller components and analyze each part to understand what needs to be done. Gather Information: Collect any data, files, or resources provided as part of the challenge. 2. This could include text, images, code, or other materials. Research: If needed, conduct research on related topics or concepts that may be relevant to solving the challenge. 3. Develop a Strategy: Based on your understanding of the challenge and the information gathered, formulate a plan or strategy for solving it. 4. Implement Solutions: Apply your strategy to solve the challenge, using any necessary tools, techniques, or programming languages. 5. Test and Iterate: Test your solutions to ensure they work as intended. If necessary, iterate on your approach to refine and improve it. 6. Document Your Process: Keep track of the steps you take, any challenges you encounter, and how you address them. This documentation can be useful for reference and learning. 7. Review and Verify: Review your solutions to ensure they meet the requirements of the challenge. 8. Verify that you have successfully completed all tasks or objectives. 9. Submit Your Solution: If the challenge requires submission, follow the provided instructions to submit your solution. |

|  |
| --- |
| **Challenge Title: DECEPTIVE MAYHEM** |
| **Flag: HQ8FLAG{bdc9d7beee8379bdb3830faff9c7ff4}** |
| **Approach (Step by Step):**   1. Analyze the Problem: Break down the challenge into smaller components and identify what needs to be accomplished by DECEPTIVE MAYHEM 2. Gather Information: Collect any data, files, or resources provided as part of the challenge. 3. Research: If needed, conduct research on related topics or concepts that may be relevant to solving the challenge. 4. Develop a Strategy: Based on your understanding of the challenge and the information gathered, formulate a plan or strategy for solving it. 5. Implement Solutions: Apply your strategy to solve the challenge, using any necessary tools, techniques, or programming languages. 6. Test and Iterate: Test your solutions to ensure they work as intended. If necessary, iterate on your approach to refine and improve it. 7. Document Your Process: Keep track of the steps you take, any challenges you encounter, and how you address them. This documentation can be useful for reference and learning. 8. Review and Verify: Review your solutions to ensure they meet the requirements of the challenge. 9. Verify that you have successfully completed all tasks or objectives. |