Lab 4: Perform Cryptography using AI

Lab Scenario

AI-enhanced cryptography empowers ethical hackers with advanced tools for securing data through complex algorithms and neural networks. It enables robust encryption, decryption, and anomaly detection, ensuring privacy and integrity in digital security practices.

The labs in this exercise demonstrate how you can use AI to perform various cryptographic functions.

Lab Objectives

- Perform cryptographic techniques using ShellGPT

Overview of Cryptography using AI

AI enhances cryptography with advanced algorithms such as AES, RSA, and quantum-resistant methods. Machine learning aids in generating secure keys, detecting anomalies in encrypted data, and optimizing cryptographic protocols. Neural networks refine encryption speed and strength, protecting data from cyber threats. AI-driven cryptography evolves to safeguard sensitive information in an increasingly digital world.

Task 1: Perform Cryptographic Techniques using ShellGPT

ShellGPT augments cryptography through innovative techniques. It leverages shell commands to encrypt data, execute cryptographic operations, and manage key distribution securely. ShellGPT integrates with existing shell environments, enhancing encryption efficiency and reliability. Its adaptability and automation streamline cryptographic processes, fortifying data protection in diverse computing environments.

Here, we will use ShellGPT to perform various cryptographic techniques.

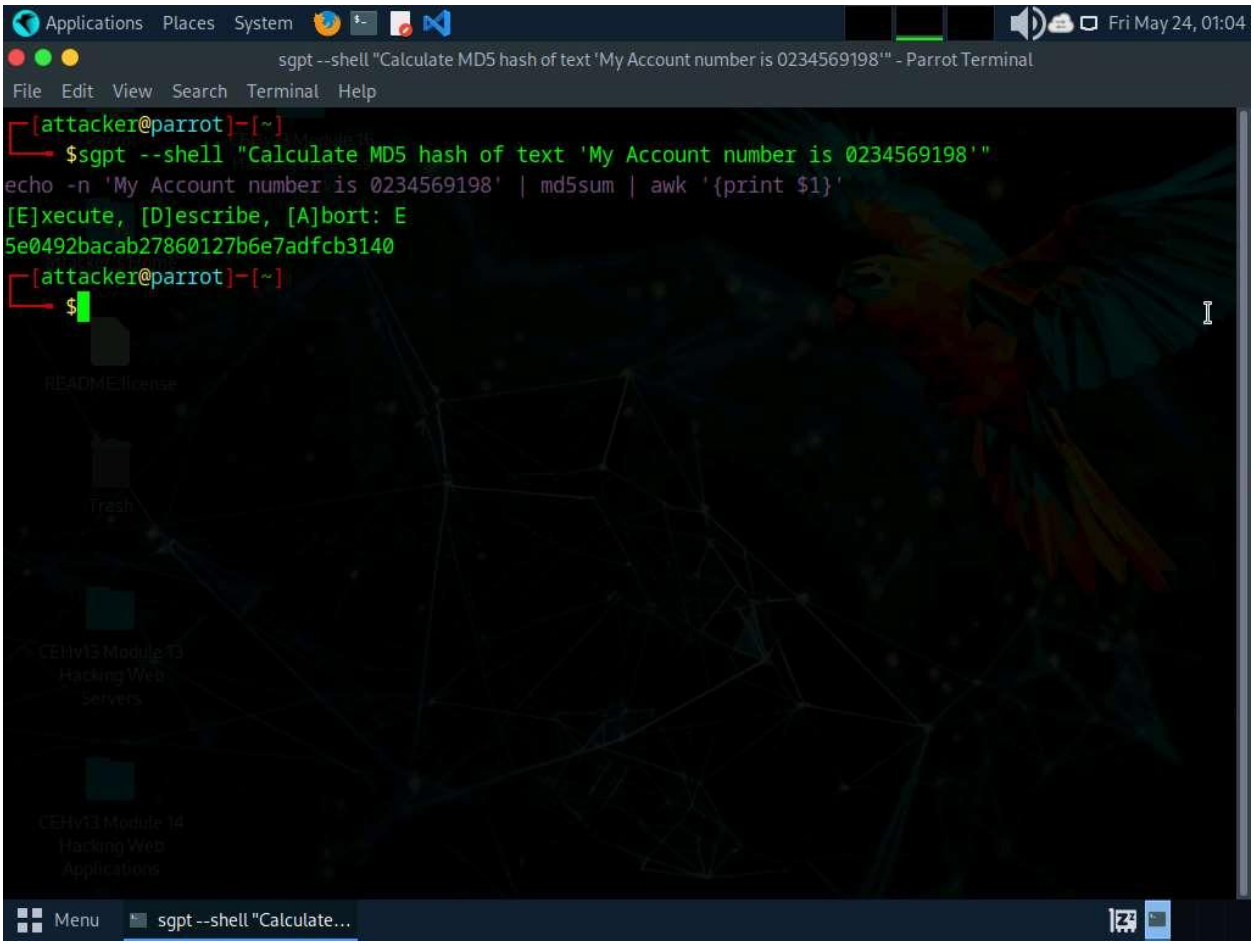
The commands generated by ShellGPT may vary depending on the prompt used and the tools available on the machine. Due to these variables, the output generated by ShellGPT might differ from what is shown in the screenshots. These differences arise from the dynamic nature of the AI's processing and the diverse environments in which it operates. As a result, you may observe differences in command syntax, execution, and results while performing this lab task.

1. Before starting this lab, click Parrot Security to switch to the **Parrot Security** machine and incorporate ShellGPT by following steps provided in Integrate ShellGPT in Parrot Security Machine.pdf.

Alternatively, you can follow the steps to integrate **ShellGPT** provided in **Module 00: Integrate ShellGPT in Parrot Security Machine**.

2. After incorporating the ShellGPT API in Parrot Security machine, run **sgpt --shell "Calculate MD5 hash of text 'My Account number is 0234569198'"** command to calculate MD5 value of the given text.

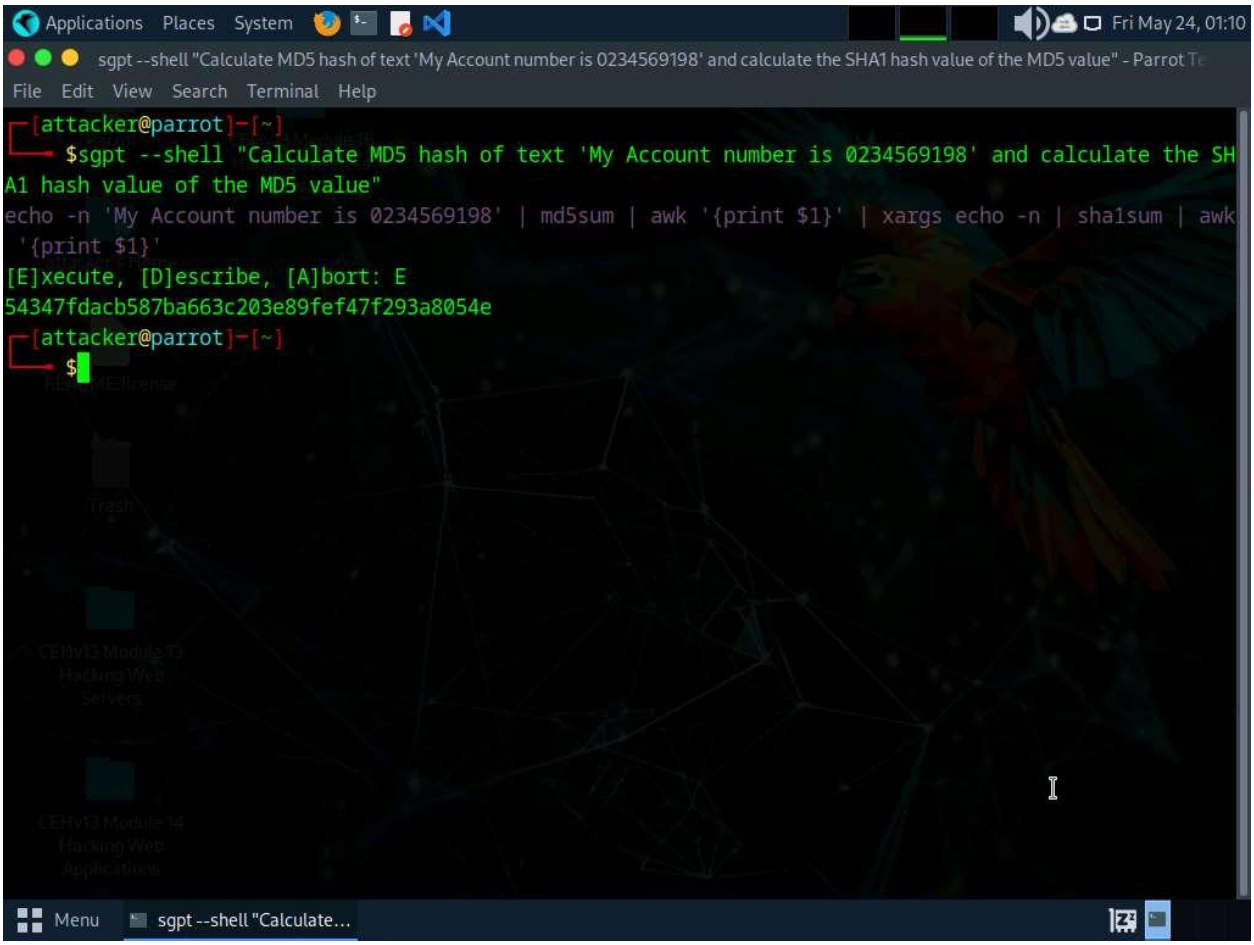
In the prompt type **E** and press **Enter** to execute the command.



MD5 (Message Digest Algorithm 5) is a widely-used cryptographic hash function producing a 128-bit hash value, typically rendered as a 32-character hexadecimal number. It is commonly used to verify data integrity.

3. Now, we will perform multi-layer hashing using ShellGPT to do so, run **sgpt --shell "Calculate MD5 hash of text 'My Account number is 0234569198' and calculate the SHA1 hash value of the MD5 value"** command.

In the prompt type **E** and press **Enter** to execute the command.

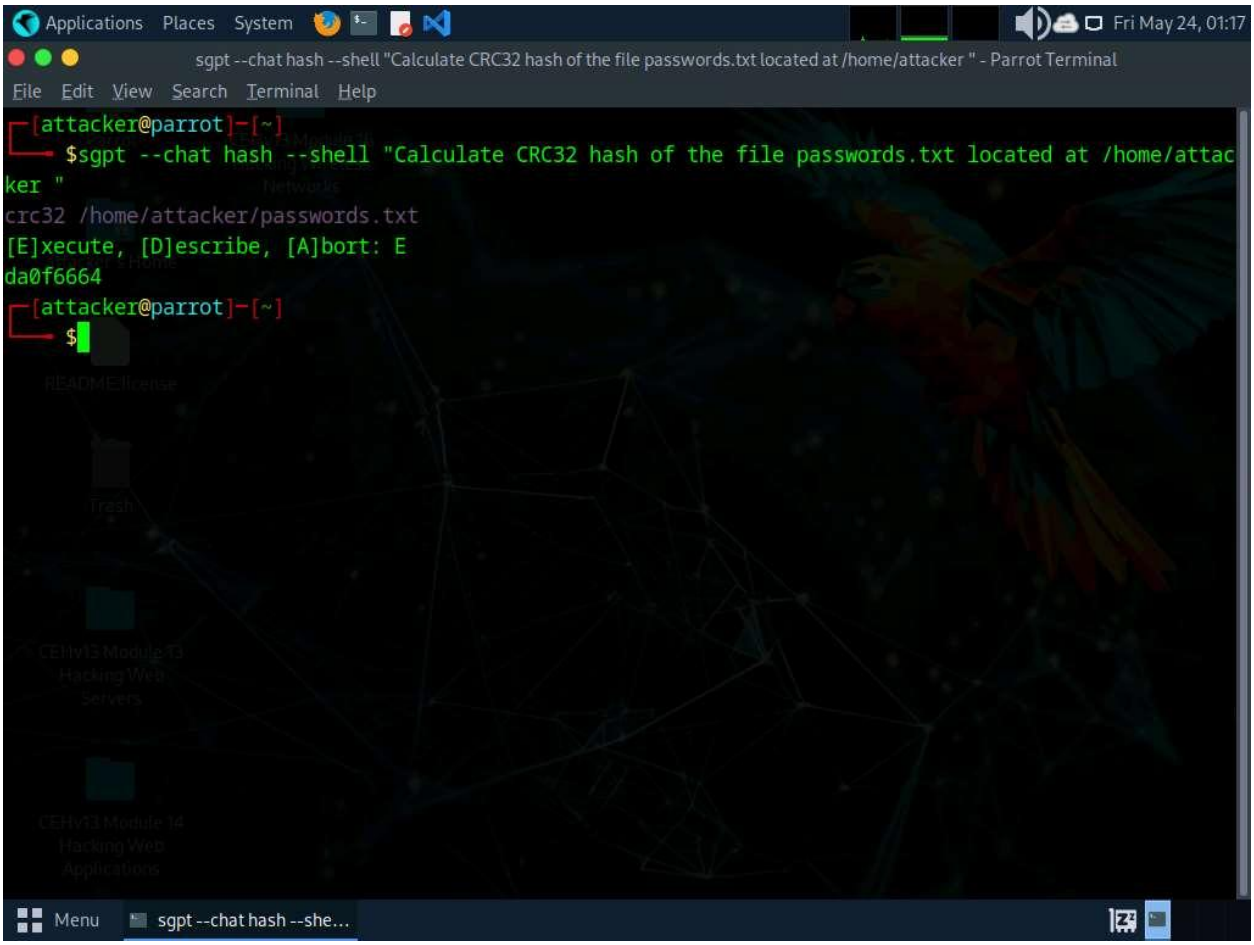


SHA-1 (Secure Hash Algorithm 1) is a cryptographic hash function that generates a 160-bit hash value, often represented as a 40-digit hexadecimal number.

4. Similarly, you can combine various hash functions in the ShellGPT prompt to perform multi-layer hashing.
5. We will now calculate hash of a file using ShellGPT, to do so, run **sgpt --chat hash --shell "Calculate CRC32 hash of the file passwords.txt located at /home/attacker"** command.

In the prompt type **E** and press **Enter** to execute the command.

You can use any file and hashing algorithm of your choice.



CRC32 (Cyclic Redundancy Check 32) is a widely used hash function to detect accidental changes to raw data. It generates a 32-bit checksum, providing a quick and efficient way to verify data integrity in storage and communication protocols like Ethernet, ZIP files, and various digital formats.

more...

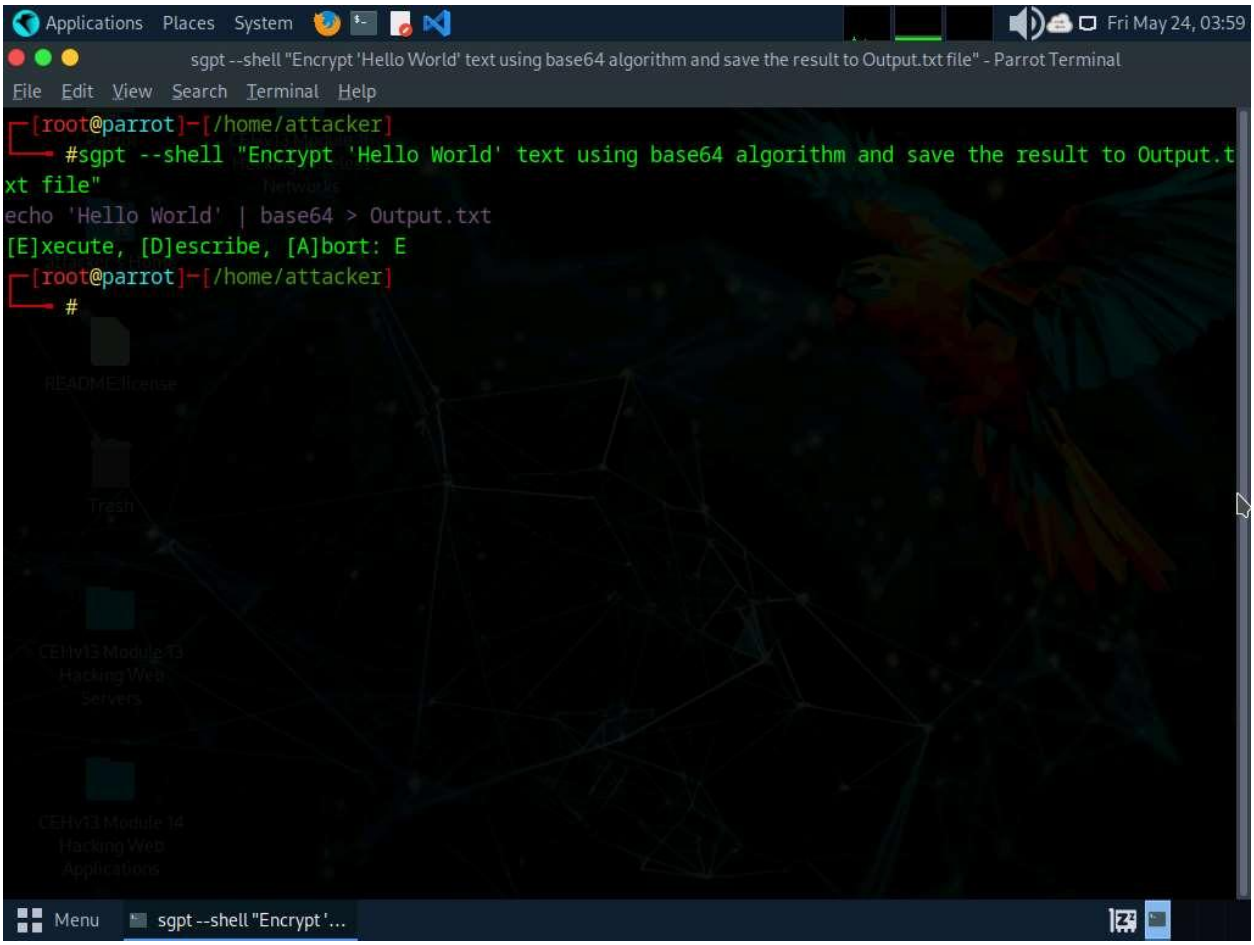
6. To perform basic encryption using ShellGPT, run the following command:

sgpt --shell "Encrypt 'Hello World' using the base64 algorithm without adding a newline character, and save the result to Output.txt file"

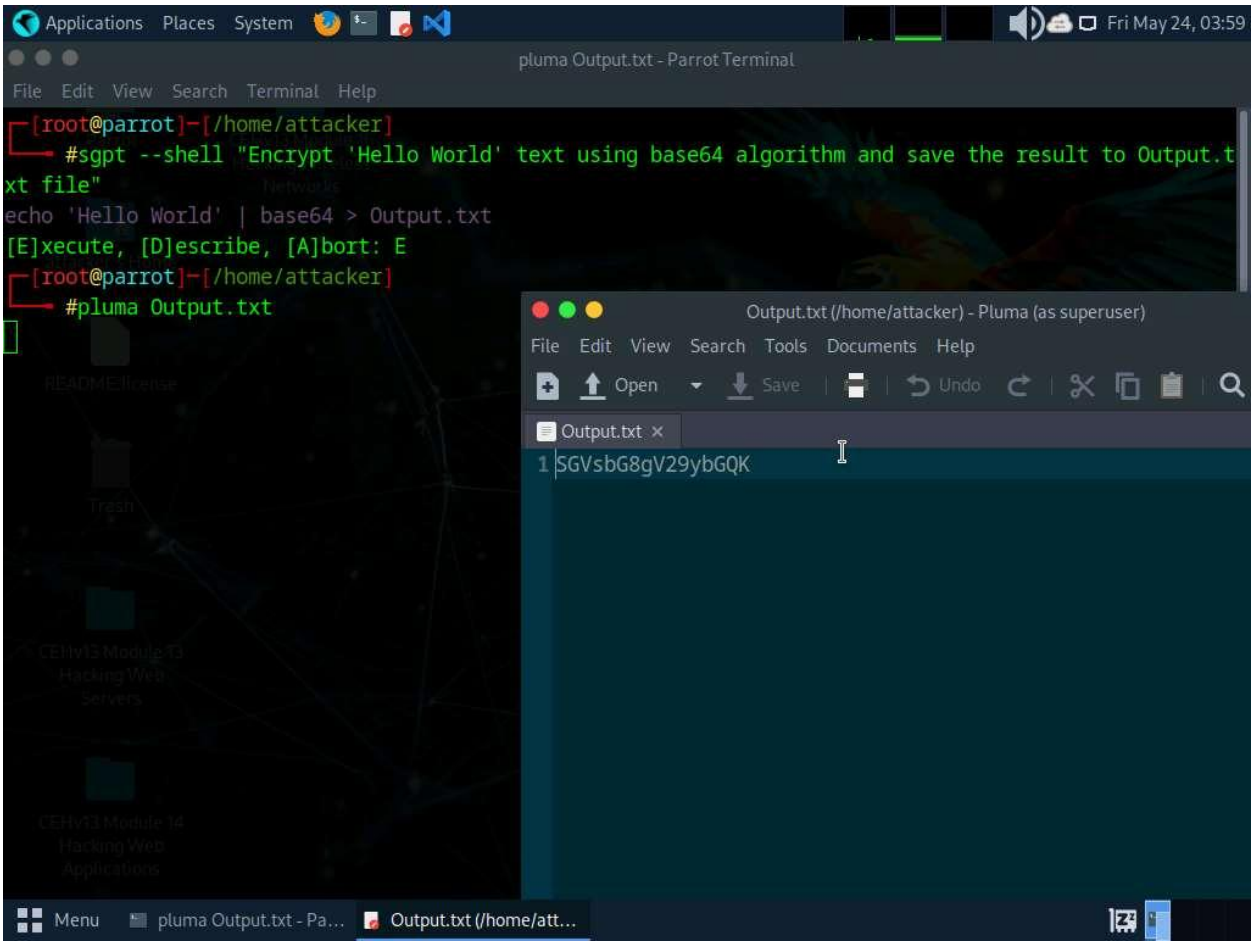
Ensure that the generated command includes echo -n "Hello World" instead of echo "Hello World", which would include a newline character and produce a different Base64 result.

The last character in the output should be the equal sign: =

If the output ends with a different character (such as **K**), that means a newline was included during encoding.

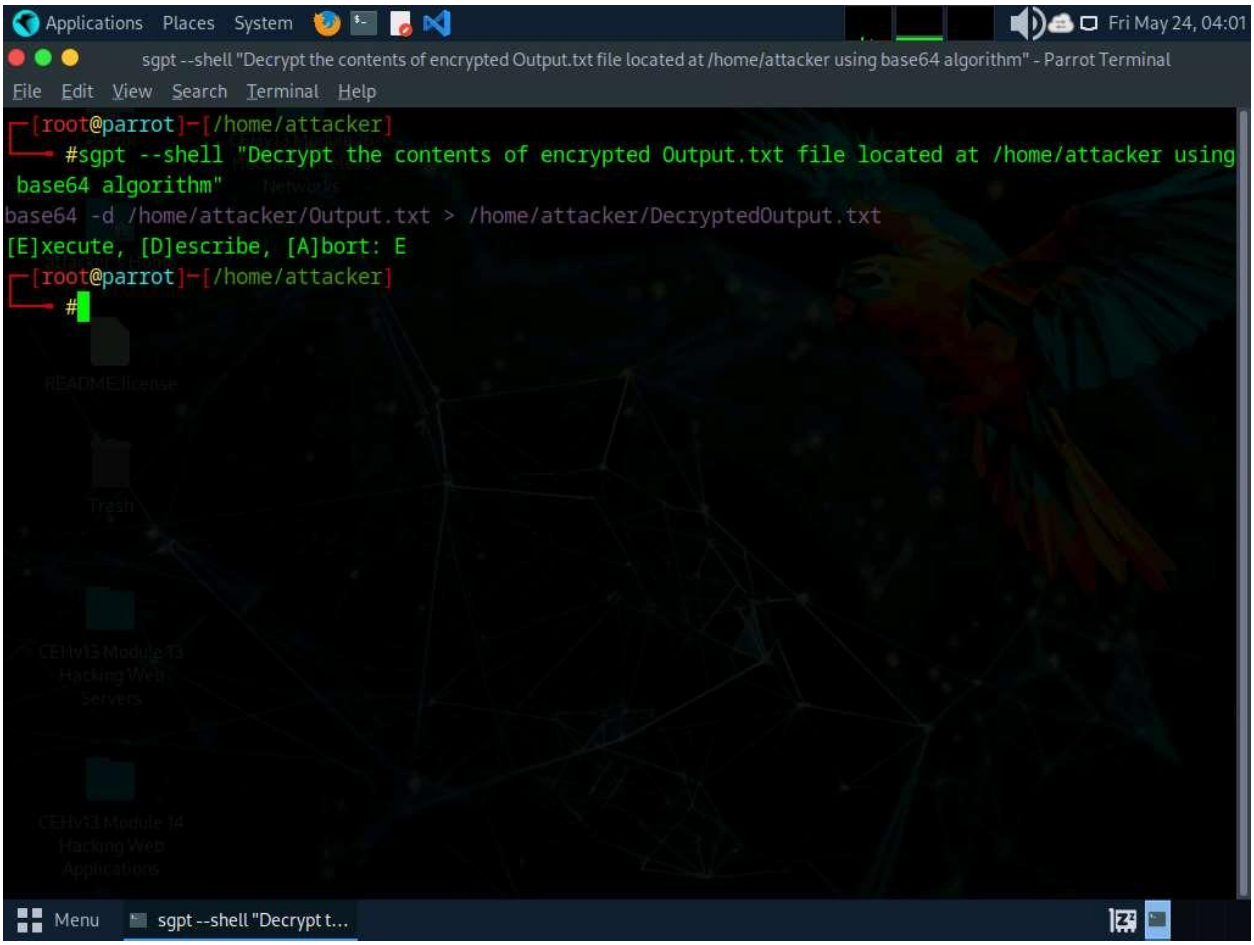


7. In the terminal run **pluma Output.txt** command to view the encrypted data.

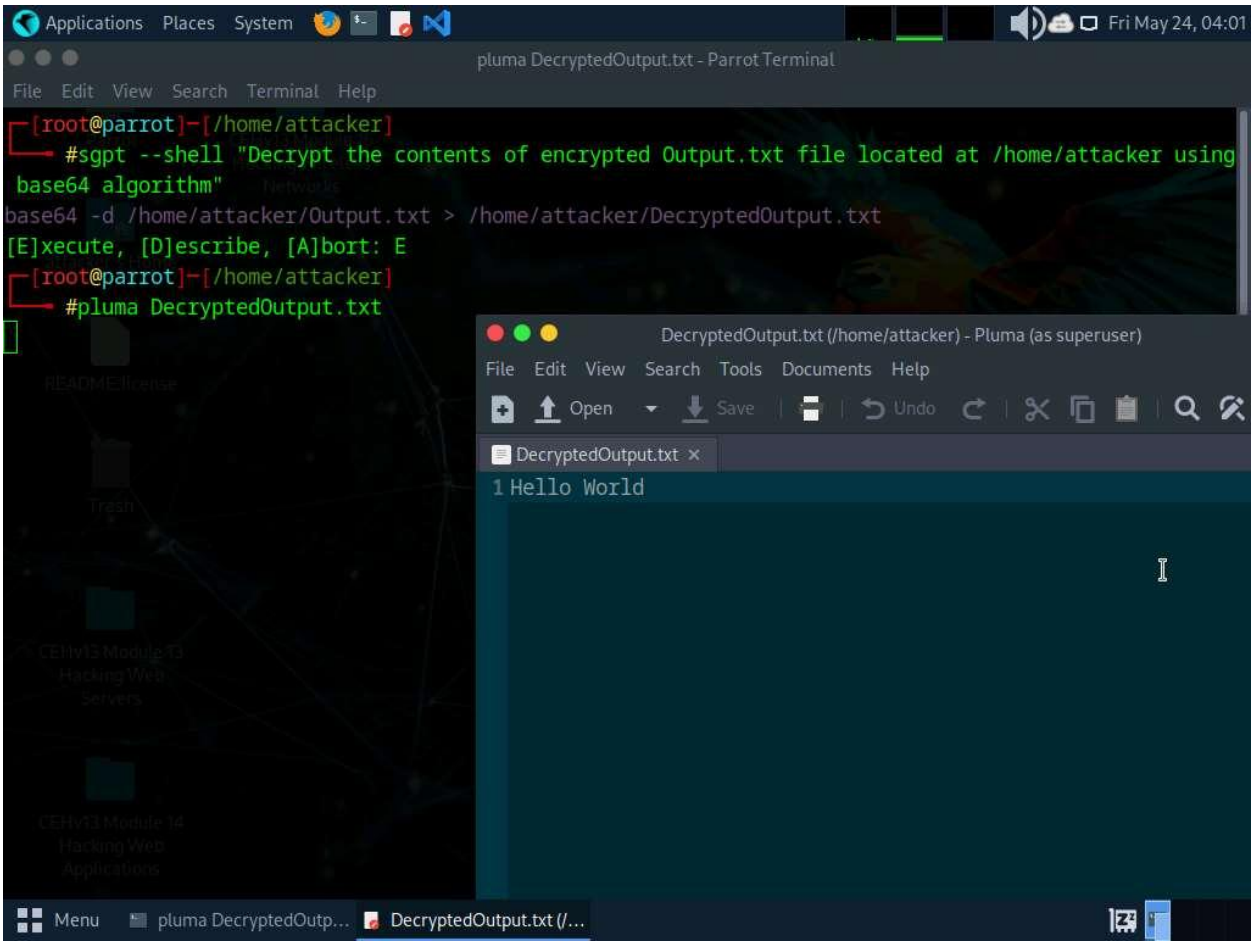


8. Close the text editor window.
9. Now we will decrypt the encrypted data using ShellGPT to do so, run **sgpt --shell "Decrypt the contents of encrypted Output.txt file located at /home/attacker using base64 algorithm"**.

In the prompt type **E** and press **Enter** to execute the command.



10. In the terminal run **pluma DecryptedOutput.txt** command to view the decrypted data.



11. Close the text editor window.
12. Apart from the aforementioned commands, you can further explore additional options within the ShellGPT tool and utilize various other tools to perform various cryptography techniques.
13. This concludes the demonstration of performing footprinting using the ShellGPT.
14. Close all open windows and document all the acquired information.

Question 20.4.1.1

Use ShellGPT on Parrot Security machine and write a prompt to encrypt Hello World text using base64 algorithm. Enter the base64 encoded form of Hello World.