EC-Council

codered
FROM EC-COUNCIL

CHAPTER 14

CRYPTOGRAPHY

CERTIFIED CYBERSECURITY TECHNICIAN

INDEX

Chapter 14:
Cryptography

INDEX

LAB SCENARIO

With the increasing adoption of the Internet (World Wide Web) for business and personal communication, securing sensitive information such as credit card details, PINs, bank account numbers, and private messages has become increasingly important, albeit more difficult to achieve. Today's information-based organizations extensively use the Internet for e-commerce, market research, customer support, and a variety of other activities. Data security is critical to online business and communication privacy.

A security professional must have the required knowledge of data security and cryptographic algorithms to enable secure transactions, communications, and other processes performed within and outside the organizations.

LAB OBJECTIVE

The objective of this lab is to provide expert knowledge on various cryptographic concepts. This includes knowledge of the following tasks:

- Calculating One-way hashes and MD5 hashes using tools such as HashCalc, MD5 Calculator and HashMyFiles
- Encrypting and decrypting data using tools such as BCTextEncoder
- Creating and using self-signed certificates
- Create and managing certificates using OpenSSL
- Performing image stegnography using OpenStego

OVERVIEW OF CRYPTOGRAPHY

Cryptography" is derived from the Greek words kryptos, which means "concealed, hidden, veiled, secret, or mysterious," and graphia, which means "writing"; thus, cryptography is "the art of secret writing." Thus, cryptography is the practice of concealing information by converting plaintext (readable format) into ciphertext (unreadable format) using a key or encryption scheme. It is the process of converting data into a scrambled code that is encrypted and sent across a private or public network. Cryptography protects confidential data such as email messages, chat sessions, web transactions, personal data, corporate data, e-commerce applications, and many other types of communication. Encrypted messages can, at times, be decrypted by cryptanalysis (code breaking), although modern encryption techniques are virtually unbreakable.

LAB TASKS

A cyber security professional or a security professional use numerous tools and techniques to implement data security using cryptographic algorithms. The recommended labs that will assist you in learning the implementation of data security include:

01 Calculate One-way Hashes using HashCalc

02 Calculate MD5 Hashes using MD5 Calculator

03 Calculate MD5 Hashes using HashMyFiles

04 Encrypt and Decrypt Data using BCTextEncoder

05 Create and Use Self-signed Certificates

06 Create and Manage Certificates using OpenSSL

07 Image Steganography using OpenStego

Note: Turn on **PfSense Firewall** virtual machine and keep it running throughout the lab exercises.

EXERCISE 1: **CALCULATE ONE-WAY HASHES USING HASHCALC**

Message digests are also called as one-way hash functions because they cannot be reversed.

LAB SCENARIO

A security professional must have the required knowledge to calculate One-way hashes using tools such as HashCalc.

LAB OBJECTIVE

This lab demonstrates how to calculate one-way hashes using HashCalc and further check the integrity of a file by comparing hash values.

OVERVIEW OF ONE-WAY HASH

Message digest (One-way Hash) functions distil the information contained in a file (small or large) into a single fixed-length number, typically between 128 and 256 bits. Message digests are also known as one-way hash functions because they cannot be reversed.

Note: Ensure that **PfSense Firewall** virtual machine is running.

1. Turn on the **Admin Machine-1** virtual machine.
2. Log in with the credentials **Admin** and **admin@123.**

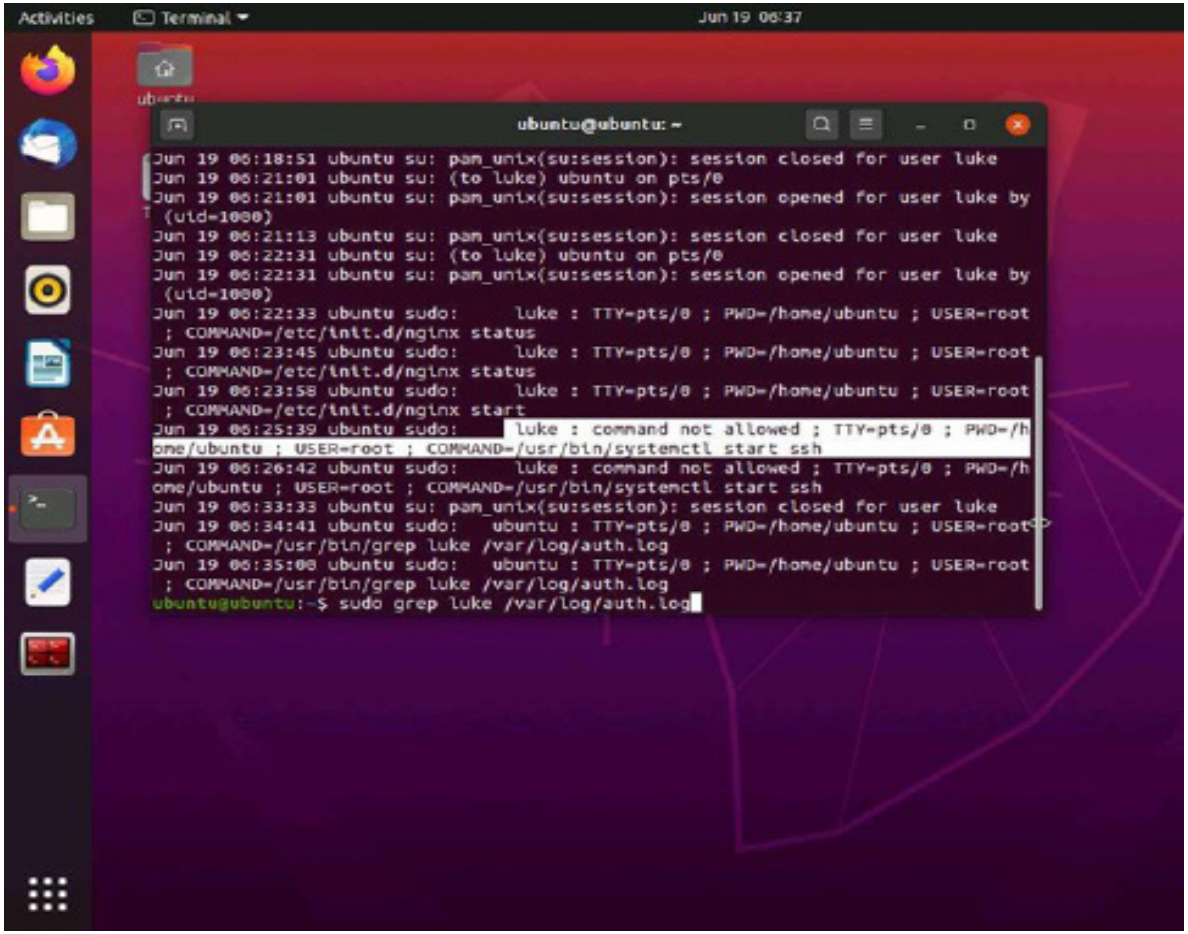
   Note: If the network screen appears, click **Yes.**
3. Navigate to the **Z:\CCT-Tools\CCT Module 14 Cryptography\MD5 and MD6 Hash Calculators\HashCalc** and double click **setup.exe.**

   Note: If the **User Account Control** pop-up appears, click **Yes.**
4. In the **Setup - HashCalc** window, click **Next.**

5. Follow the installation wizard to install **HashCalc** with all default settings.

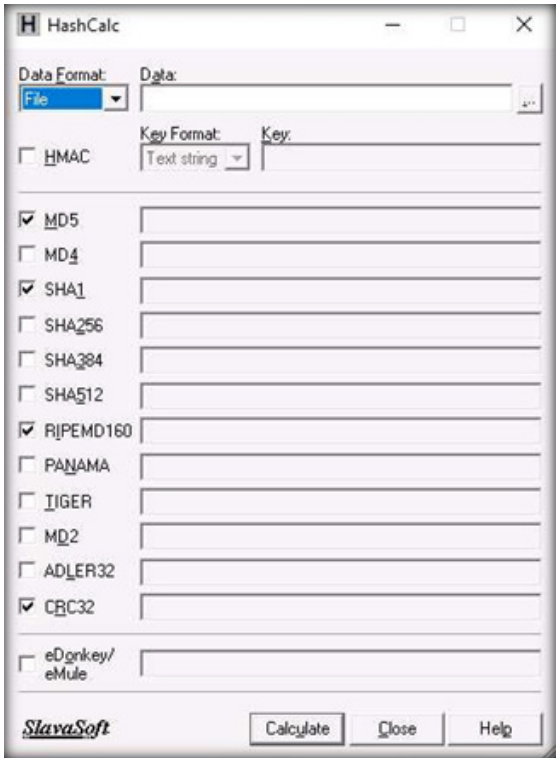
6. After the completion of installation, a **Completing the HashCalc Setup Wizard** appears. Uncheck the **View the README file** checkbox and click **Finish.**

EXERCISE 1: CALCULATE ONE-WAY HASHES USING HASHCALC



7. The **HashCalc** main window appears, as shown in the screenshot below.

EXERCISE 1: CALCULATE ONE-WAY HASHES USING HASHCALC



8. Minimize the **HashCalc** window. Navigate to **Desktop**, right-click on the **Desktop** window, and navigate to **New** → **Text Document** to create a new text file.

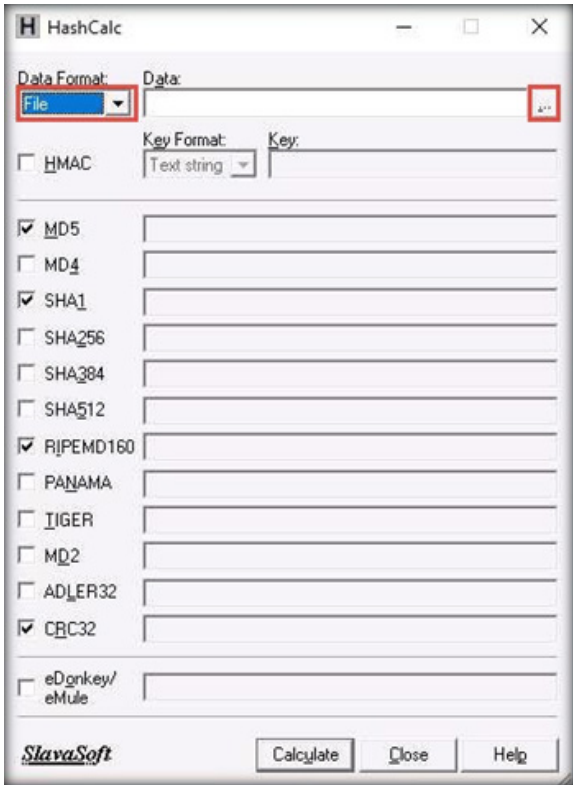
   Note: You can create a text file at any location of your choice.

9. A newly created text file appears; rename it to **Test.txt** and open it. Write some text in it (here, **Hello World** !!) and press **Ctrl+S** to save the file. Now, close the text file.

EXERCISE 1: CALCULATE ONE-WAY HASHES USING HASHCALC

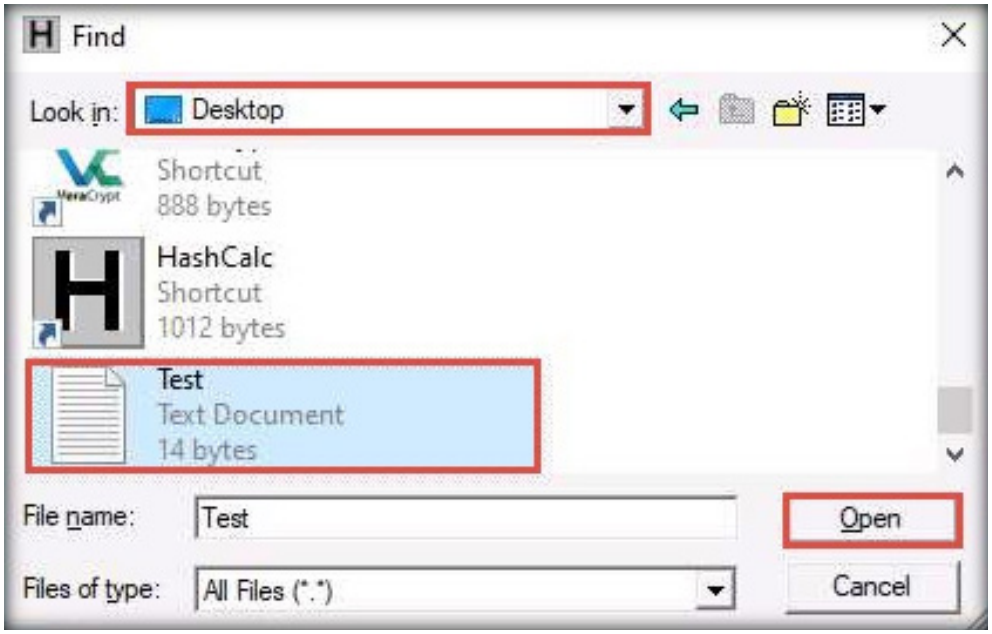
10. Now, switch back to the **HashCalc** window. Ensure that the **File** option is selected in the **Data Format** field and click on the ellipsis icon under the **Data** field.

EXERCISE 1: CALCULATE ONE-WAY HASHES USING HASHCALC



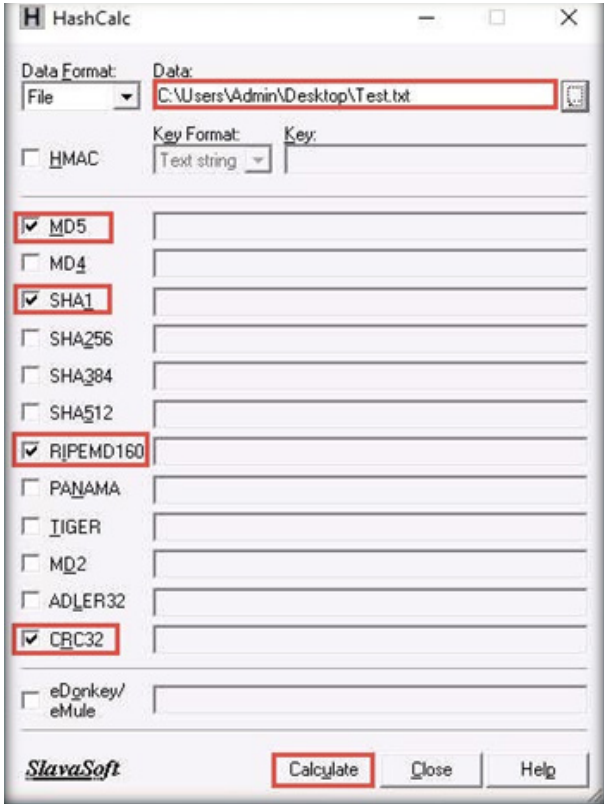
11. In the **Find** window that appears, navigate to the location where you saved the **Test.txt** file (here, **Desktop**) and click **Open.**

EXERCISE 1: CALCULATE ONE-WAY HASHES USING HASHCALC



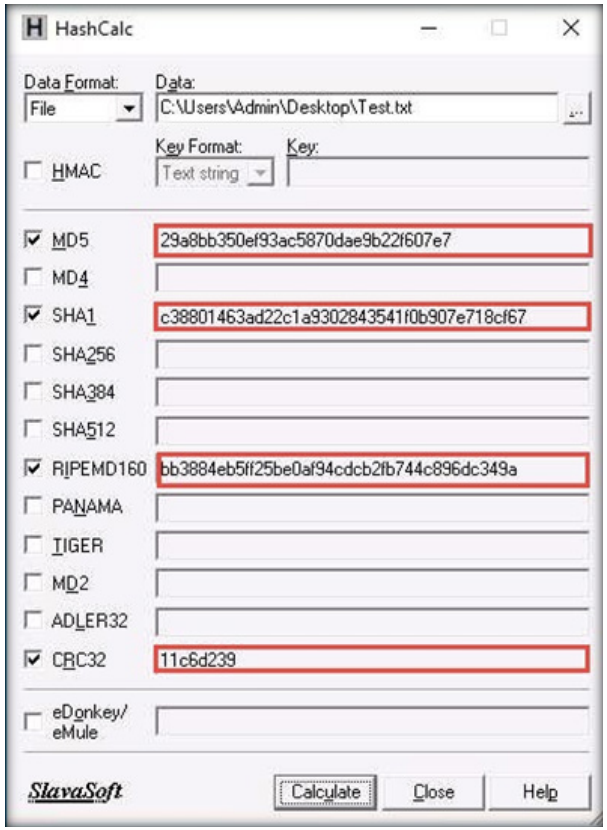
12. The path of the selected file **(Test.txt)** appears under the **Data** field. Ensure that the **MD5, SHA1, RIPEMD160,** and **CRC32** hash functions are selected. Click the **Calculate** button.

EXERCISE 1: CALCULATE ONE-WAY HASHES USING HASHCALC



13. The calculated hash values of the **Test.txt** file appears, as shown in the screenshot below.

EXERCISE 1: CALCULATE ONE-WAY HASHES USING HASHCALC

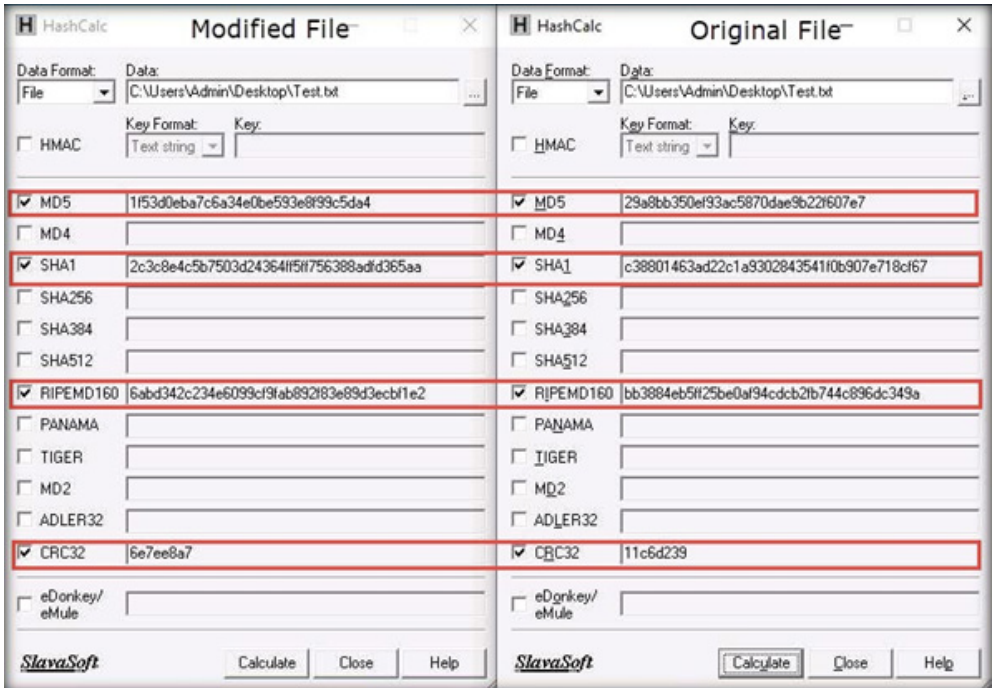


14. Minimize the **HashCalc** window, navigate to **Desktop,** and double-click the **Test.txt** file to open it. Modify the file content by writing some text (here, **Modified File...!!!**) and press **Ctrl+S** to save it. Now, close the text file.

EXERCISE 1: CALCULATE ONE-WAY HASHES USING HASHCALC

15. Now, double-click on the **HashCalc** shortcut from **Desktop** to launch another HashCalc window.
16. In the new **HashCalc** window that appears, perform **Steps #10-13.**
17. Now, maximize the first **HashCalc** window and place it beside the second **HashCalc** window. You can observe changes in the hash values of the text file **(Test.txt)** before and after the modification, as shown in the screenshot below.

EXERCISE 1: CALCULATE ONE-WAY HASHES USING HASHCALC



Note: In real-time, the HashCalc tool is used to check the integrity of a file when the changes in hash values indicate the file content has been modified.

18. This concludes the demonstration showing how to calculate one-way hashes using HashCalc.
19. Close all open windows and document all acquired information.

EXERCISE 1: CALCULATE ONE-WAY HASHES USING HASHCALC

EXERCISE 2: **CALCULATE MD5 HASHES USING MD5 CALCULATOR**

MD5 Calculator is a simple application that calculates the MD5 hash of a given file.

LAB SCENARIO

A security professional must have the required knowledge to calculate MD5 hashes using tools such as MD5 Calculator to check the integrity of a given file.

LAB OBJECTIVE

This lab demonstrates how to calculate MD5 hashes of a given file using MD5 Calculator.

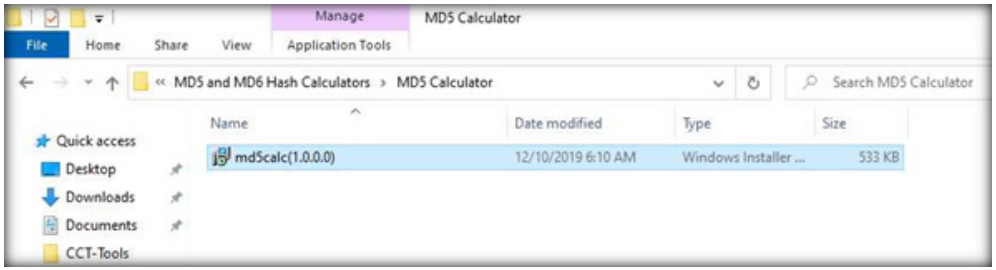
OVERVIEW OF ONE-WAY HASH

MD5 Calculator can be used with large files (e.g., several gigabytes in size). It features a progress counter and a text field from which the final MD5 hash can be easily copied to the clipboard. MD5 Calculator can be used to check the integrity of a file.

Note: Ensure that **Admin Machine-1** and **PfSense Firewall** virtual machines are running.

1. In the **Admin Machine-1** virtual machine, navigate to **Z:\CCT-Tools\CCT Module 14 Cryptography\MD5 and MD6 Hash Calculators\MD5 Calculator** and double-click **md5calc(1.0.0.0).msi.**

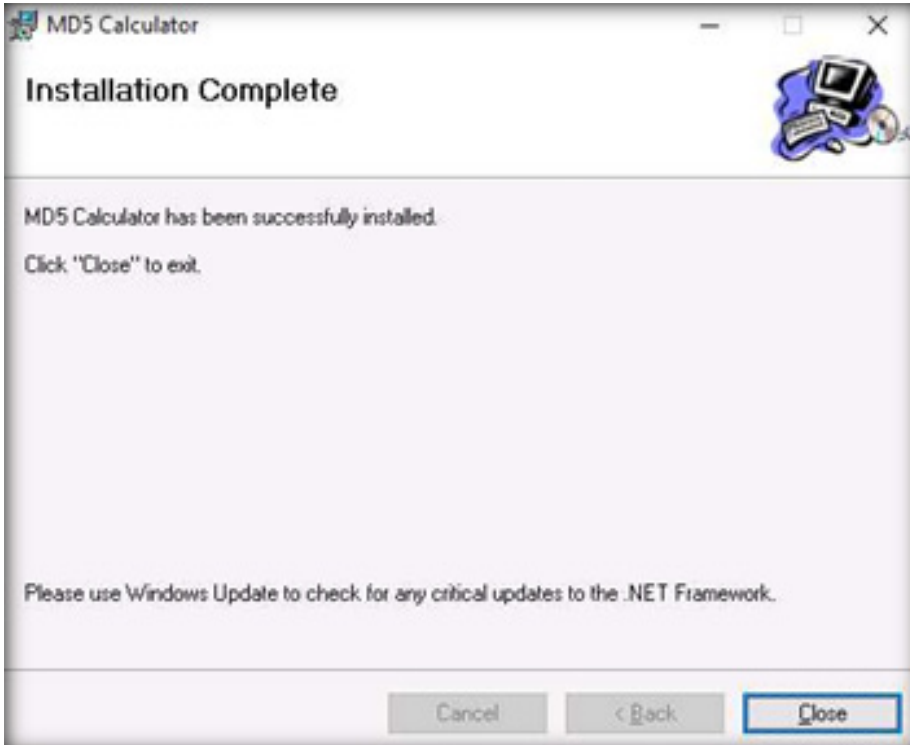
EXERCISE 2: CALCULATE MD5 HASHES USING MD5 CALCULATOR



2. The **MD5 Calculator** setup window appears; click **Next.**
3. Follow the installation wizard to install the **MD5 Calculator** using all default settings.

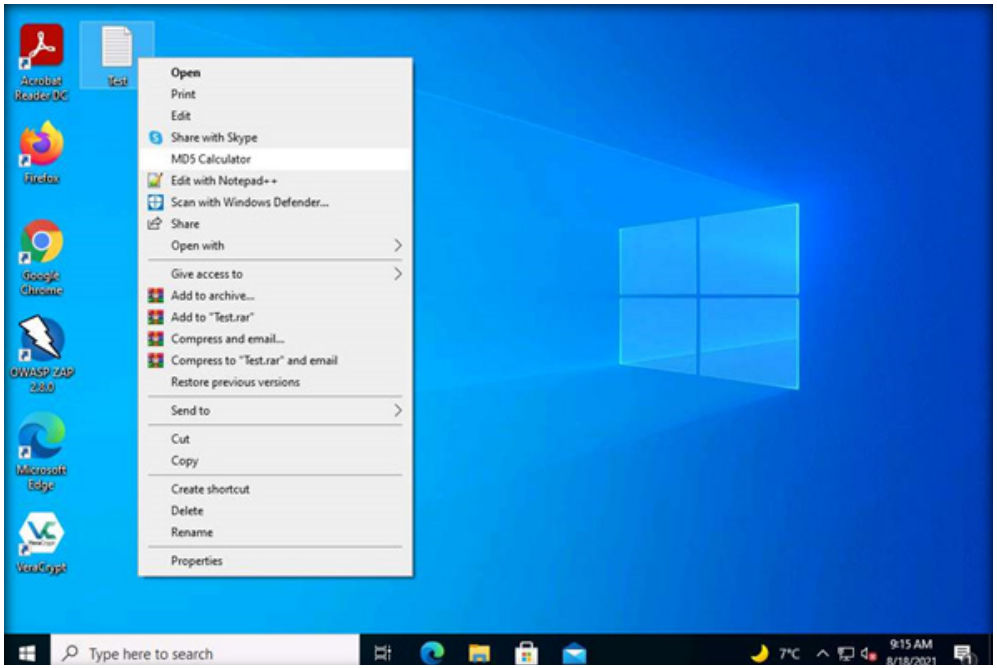
   Note: If a **User Account Control** pop-up appears, click Yes.
4. After the completion of the installation, the **Installation Complete** wizard appears; click **Close.**

EXERCISE 2: CALCULATE MD5 HASHES USING MD5 CALCULATOR



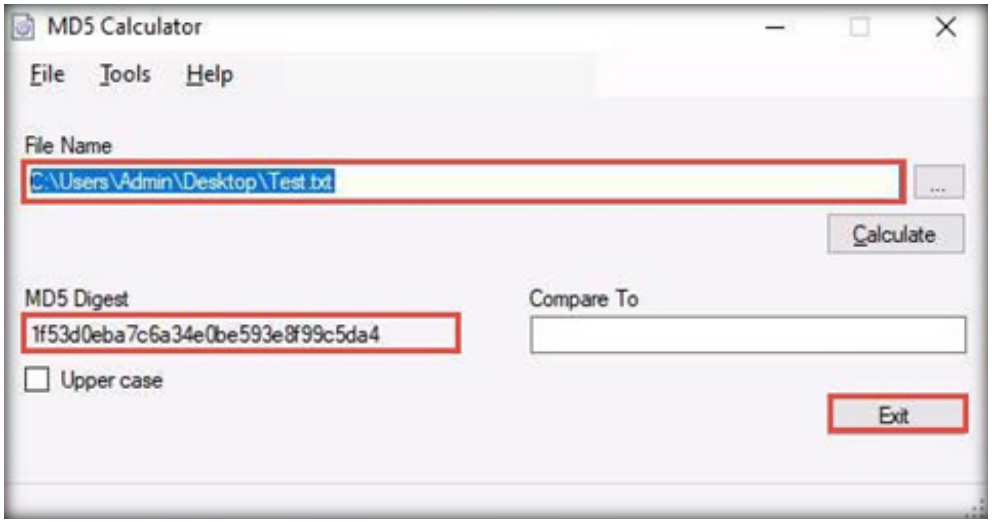
5. Navigate to **Desktop,** right-click on the text file **(Test.txt)** that we created in the previous task and click **MD5 Calculator** from the context menu to calculate the MD5 hash of the file.

EXERCISE 2:
CALCULATE MD5
HASHES USING MD5
CALCULATOR



6. The **MD5 Calculator** window appears, with the file path under the **File Name** field and MD5 hash value under the **MD5 Digest** field, as shown in the screenshot below.
7. Copy the MD5 hash value from the **MD5 Digest** field and click on **Exit** to close the **MD5 Calculator.**

EXERCISE 2: CALCULATE MD5 HASHES USING MD5 CALCULATOR

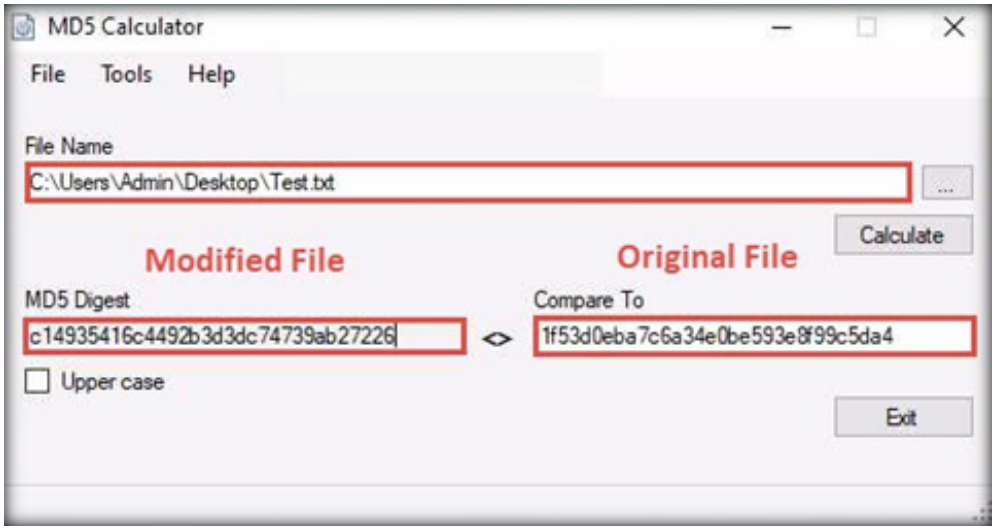


8. Now, double-click on the **Test.txt** file from **Desktop** to open it and change its content of the file by inserting some text in it (here, **Hello World...!!!**). Now, save and close the **Test.txt** file.

EXERCISE 2: CALCULATE MD5 HASHES USING MD5 CALCULATOR

9. After changing the file content, right-click on the text file **(Test.txt)** again and click **MD5 Calculator** from the context menu to calculate its MD5 hash.
10. A new **MD5 Calculator** window appears, with the MD5 hash value under the **MD5 Digest** field. In the **Compare To** field, paste the copied MD5 hash value of the file before modification.
11. The symbol (<>) between the **MD5 Digest** and **Compare** To fields indicates that the MD5 hash values of the file before modification is not equal to the MD5 hash value of the file after modification.

EXERCISE 2: CALCULATE MD5 HASHES USING MD5 CALCULATOR



Note: If a person wants to send a file to another person via a medium (e.g., email), they will calculate its hashes and send the file (along with the hash value) to the intended person. When the intended person receives the email, they will download the file and calculate its hash value using the MD5 Calculator.

Note: The recipient will then compare the generated hash value with the hash value that was sent through email. If both hash values match, it is evident that the file was received without any modifications by a third person and that its integrity is intact.

12. This concludes the demonstration showing how to calculate MD5 hashes using MD5 Calculator.
13. Close all open windows and document all the acquired information.

EXERCISE 2: CALCULATE MD5 HASHES USING MD5 CALCULATOR

EXERCISE 3: **CALCULATE MD5 HASHES USING HASHMYFILES**

MD5 is a message digest algorithm used in digital signature applications to compress a document securely before the system signs it with a private key.

LAB SCENARIO

A security professional must have the required knowledge to calculate MD5 hashes using tools such as HashMyFiles.

LAB OBJECTIVE

This lab demonstrates how to calculate MD5 hashes of a given file using HashMyFiles.

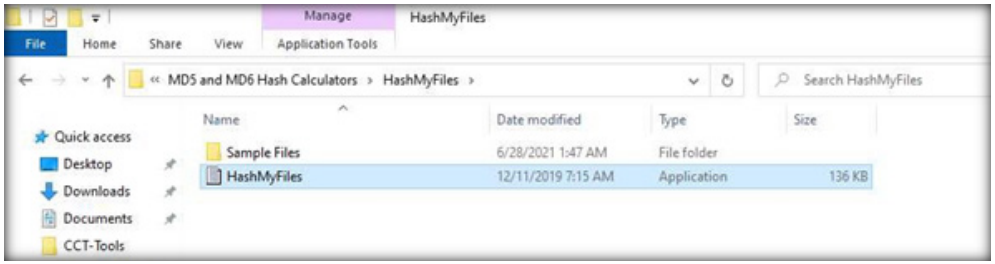
OVERVIEW OF MD5 HASHES

MD5 is a widely used cryptographic hash function that takes a message of arbitrary length as input and outputs a 128-bit (16-byte) fingerprint or message digest. MD5 can be used in a wide variety of cryptographic applications and is useful for digital signature applications, file integrity checking, and storing passwords. However, MD5 is not collision resistant; therefore, it is better to use the latest algorithms, such as MD6, SHA-2, and SHA-3.

Note: Ensure that **Admin Machine-1** and **PfSense Firewall** virtual machines are running.

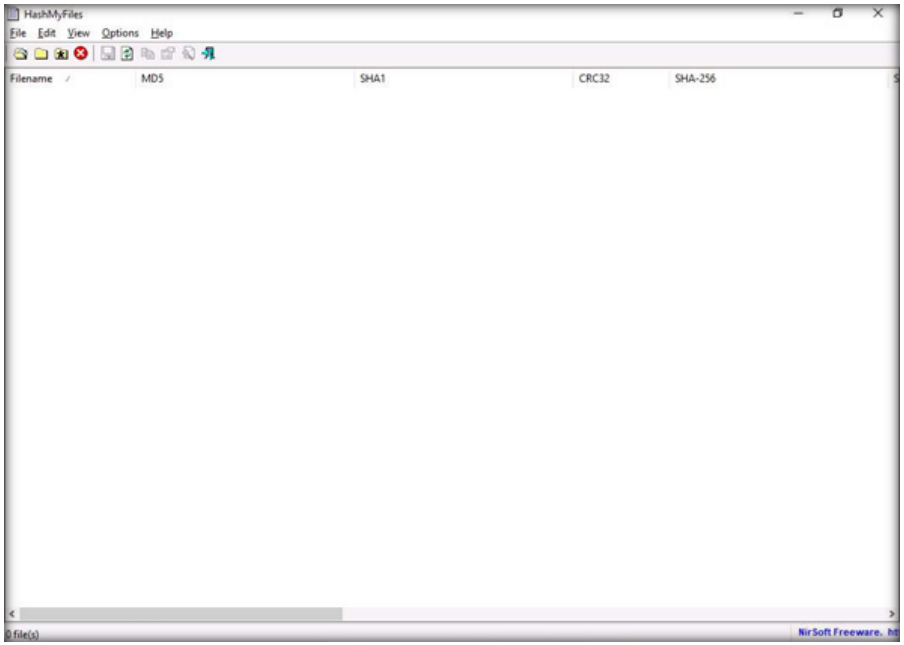
1. In the **Admin Machine-1** virtual machine, navigate to **Z:\CCT-Tools\CCT Module 14 Cryptography\MD5 and MD6 Hash Calculators\HashMyFiles** and double-click **HashMyFiles.exe.**

EXERCISE 3: CALCULATE MD5 HASHES USING HASHMYFILES



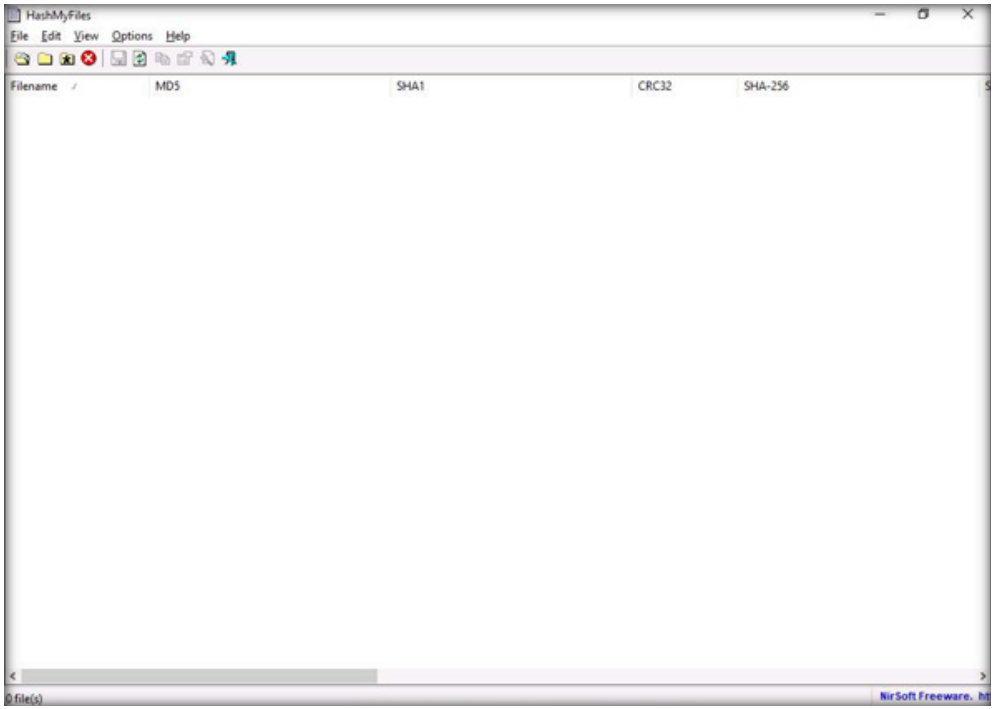
2. The **HashMyFiles** main window appears, as shown in the screenshot below.

EXERCISE 3:
CALCULATE MD5
HASHES USING
HASHMYFILES



2. The **HashMyFiles** main window appears, as shown in the screenshot below.

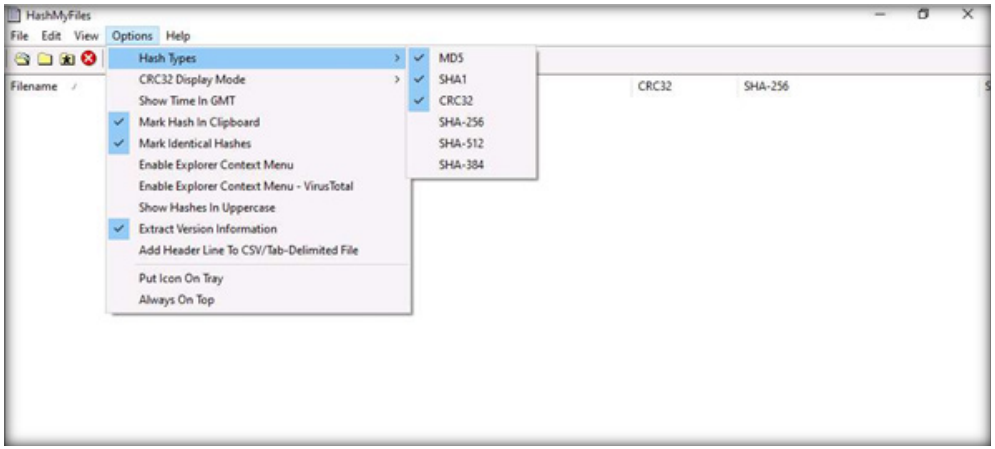
EXERCISE 3:
CALCULATE MD5
HASHES USING
HASHMYFILES



3. In the **HashMyFiles** window, click **Options** from the menu bar and choose **Hash Types** from the options.
4. You can observe a list of hash functions. From the list, untick **SHA-256, SHA-512** and **SHA-384 Hash Types.**

   Note: Here, we will calculate **MD5, SHA1** and **CRC32 Hash Types.**

EXERCISE 3: CALCULATE MD5 HASHES USING HASHMYFILES



5. In the **HashMyFiles** window, click **File** from the menu bar. From the drop-down list, click the **Add Folder** option.

   Note: You can also use the **Add Files** option to add multiple files.

EXERCISE 3: CALCULATE MD5 HASHES USING HASHMYFILES

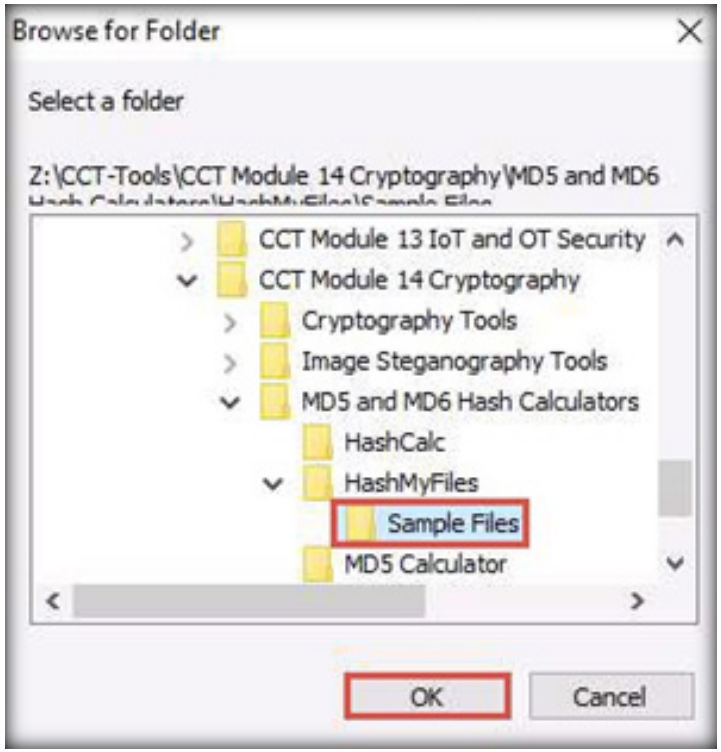
6. In the **Select Folder** pop-up appears; click on the ellipsis icon to select the folder you want to encrypt.

EXERCISE 3:
CALCULATE MD5
HASHES USING
HASHMYFILES

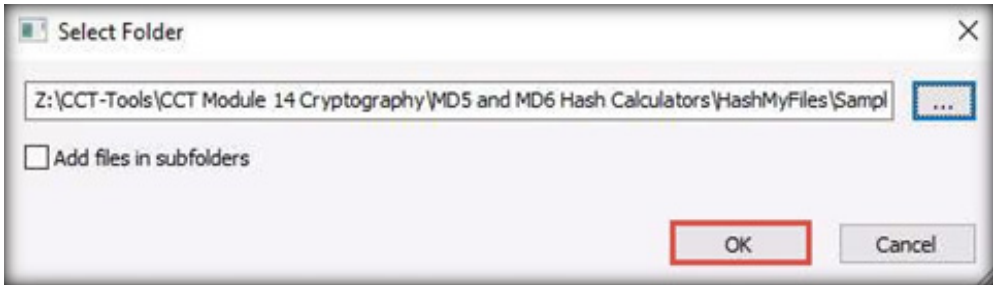
7. The **Browse for Folder** window appears; navigate to **Z:\CCT-Tools\CCT Module 14 Cryptography\MD5 and MD6 Hash Calculators\HashMyFiles** and select the **Sample Files** folder; then, click **OK.**

Note: You can select any folder of your choice that you wish to encrypt.

EXERCISE 3: CALCULATE MD5 HASHES USING HASHMYFILES



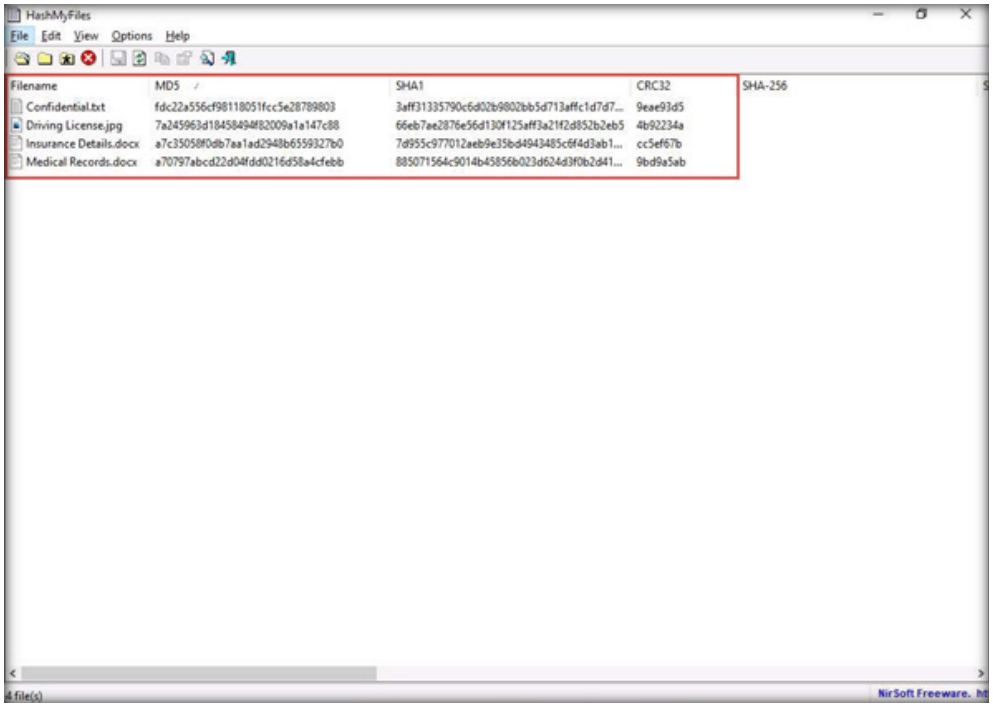
8. The location of the selected folder appears in the field; click **OK.**



EXERCISE 3:
CALCULATE MD5
HASHES USING
HASHMYFILES

9. A list of files contained in the folder appears, along with their various hash values such as **MD5, SHA1, CRC32,** etc.

EXERCISE 3: CALCULATE MD5 HASHES USING HASHMYFILES



Note: In real-time, you may share confidential information in the folder in an encrypted form to maintain its integrity.

10. This concludes the demonstration showing how to calculate MD5 hashes using HashMyFiles.
11. Close all open windows and document all the acquired information.

EXERCISE 3: CALCULATE MD5 HASHES USING HASHMYFILES

EXERCISE 4: **ENCRYPT AND DECRYPT DATA USING BCTEXTENCODER**

BCTextEncoder simplifies encoding and decoding text data.

LAB SCENARIO

A security professional must have the required knowledge to encrypt and decrypt organization's sensitive data to maintain its confidentiality and integrity.

LAB OBJECTIVE

This lab demonstrates how to perform data encryption and decryption using tools such as BCTextEncoder.

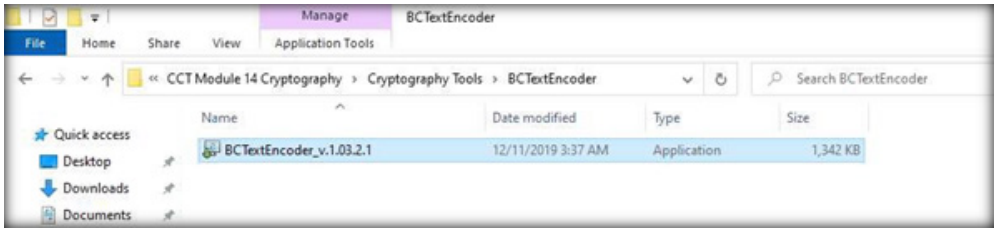
OVERVIEW OF BCTEXTENCODER

BCTextEncoder compresses, encrypts, and converts plaintext data into text format, which the user can then copy to the clipboard or save as a text file. It uses public key encryption methods as well as password-based encryption. Furthermore, it uses strong and approved symmetric and public-key algorithms for data encryption.

Note: Ensure that **Admin Machine-1** and **PfSense Firewall** virtual machines are running.

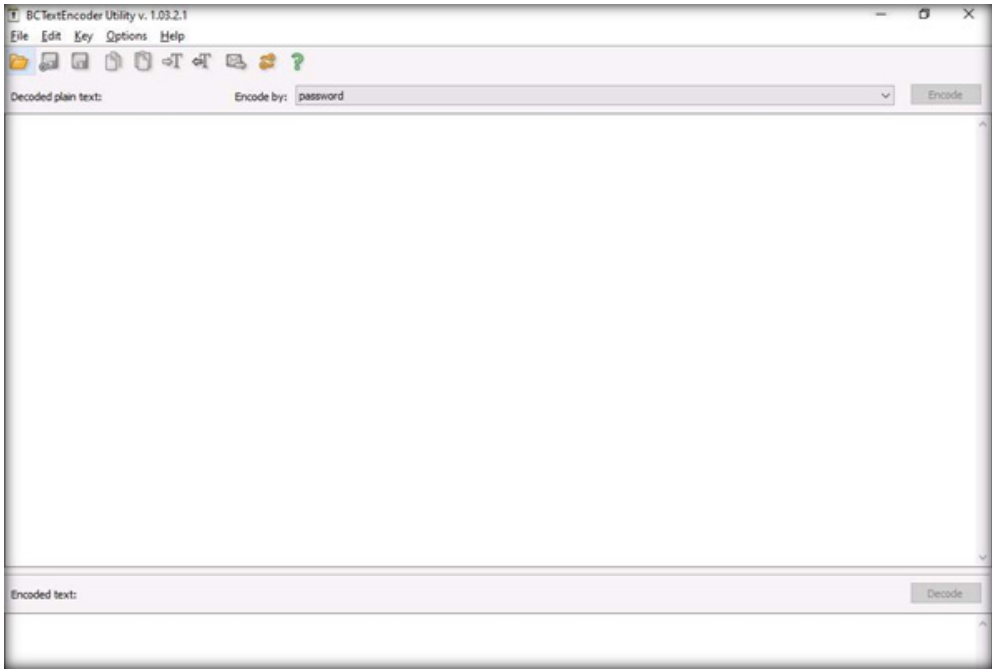
1. In the **Admin Machine-1** virtual machine, navigate to **Z:\CCT-Tools\CCT Module 14 Cryptography\Cryptography Tools\BCTextEncoder** and double click **BCTextEncoder_v.1.03.2.1.exe.**

EXERCISE 4: ENCRYPT AND DECRYPT DATA USING BCTEXTENCODER



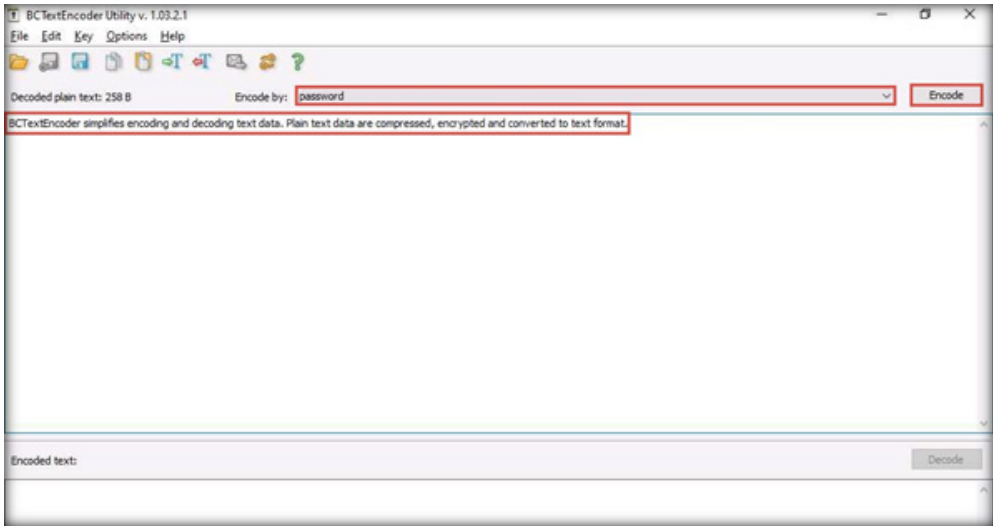
2. The **BCTextEncoder Utility** window appears, as shown in the screenshot below.

EXERCISE 4: ENCRYPT AND DECRYPT DATA USING BCTEXTENCODER



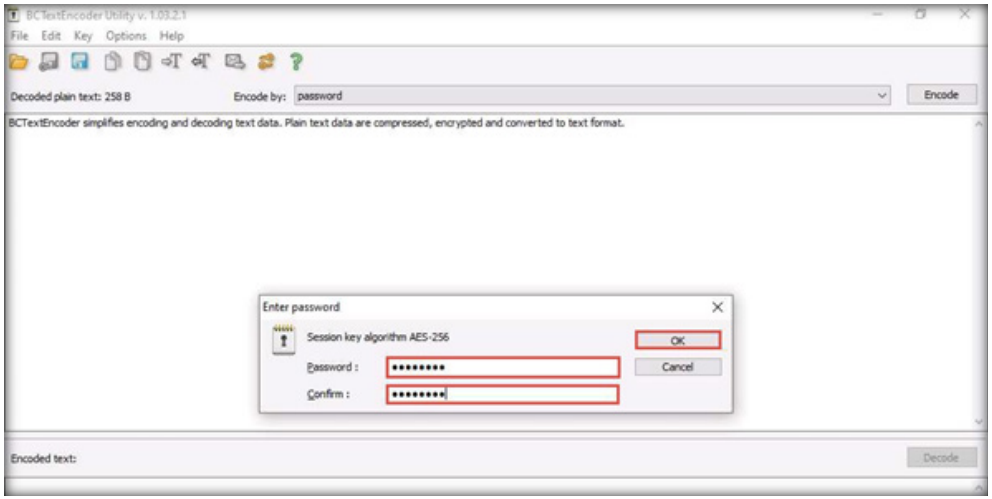
3. To encrypt text, insert some text in the clipboard.
   Or
   Select the data you want to encode and paste it to the clipboard by pressing **Ctrl+V.**
4. Ensure that the **password** option is selected in the **Encode by** field and click **Encode.**

EXERCISE 4: ENCRYPT AND DECRYPT DATA USING BCTEXTENCODER



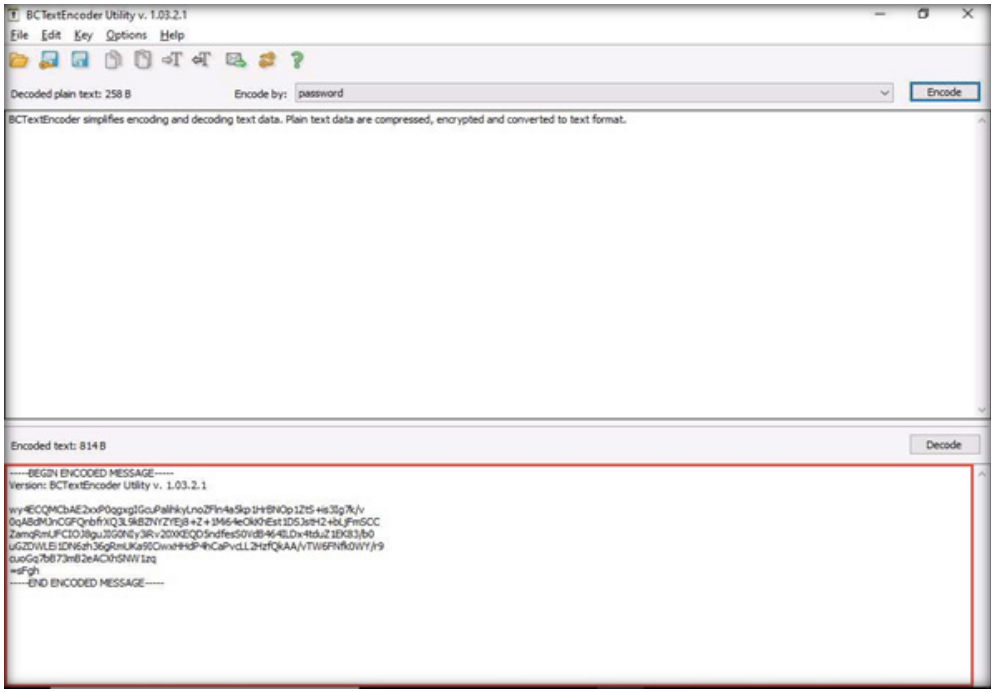
5. In the Enter password pop-up appears; enter the password into the Password field and retype it in the Confirm field; then, click OK. (Here, we use the password test@123).

EXERCISE 4: ENCRYPT AND DECRYPT DATA USING BCTEXTENCODER



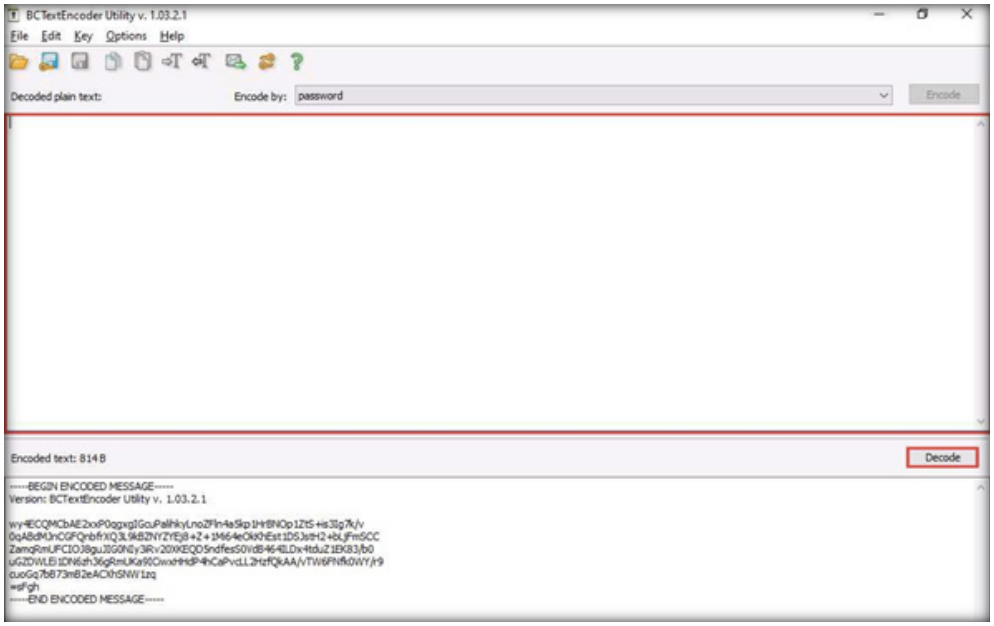
6. BCTextEncoder encodes the text and displays it in under the Encoded text section, as shown in the screenshot below.

EXERCISE 4:
ENCRYPT AND DECRYPT DATA USING BCTEXTENCODER



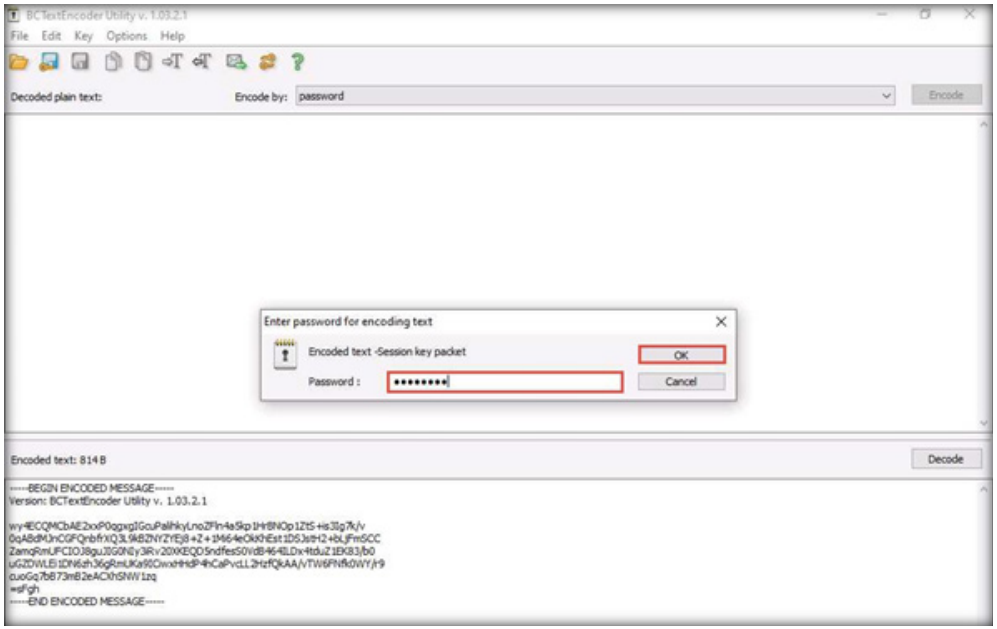
7. To decrypt the data, you need to clean the Decoded plain text in the clipboard first, and then click the Decode button.

EXERCISE 4: ENCRYPT AND DECRYPT DATA USING BCTEXTENCODER



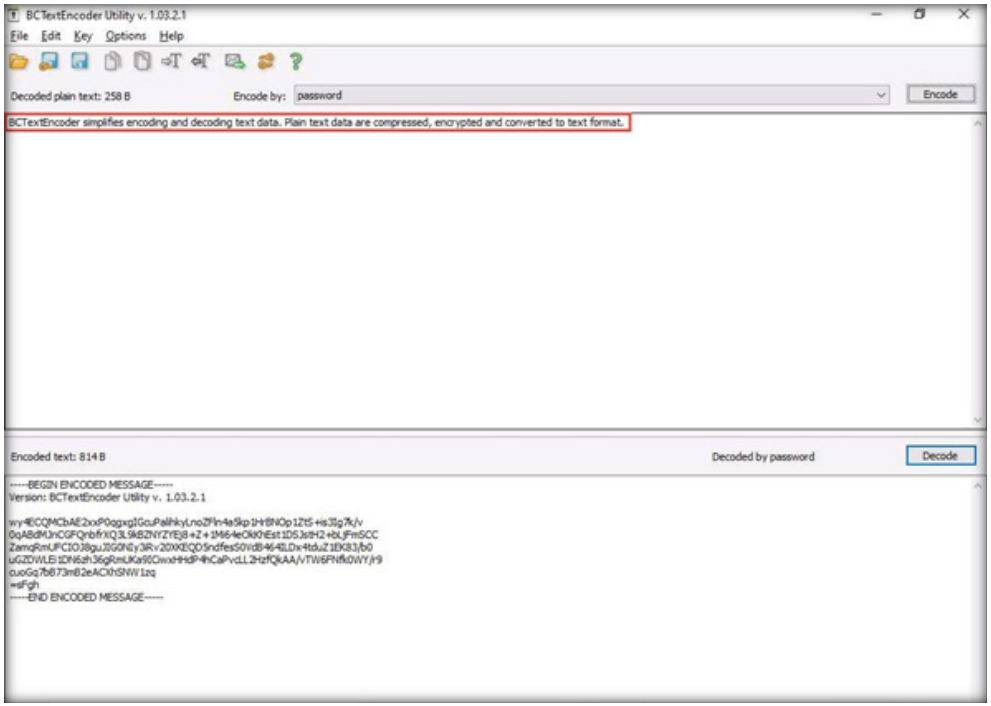
8. In the Enter password for encoding text dialog-box appears; enter the Password (here, test@123) into the password field and click OK.

EXERCISE 4: ENCRYPT AND DECRYPT DATA USING BCTEXTENCODER



9. The decoded plain text appears under the Decoded plain text section, as shown in the screenshot below.

EXERCISE 4:
ENCRYPT AND DECRYPT DATA USING BCTEXTENCODER



Note: In real-time, you can use this procedure to encode the text while sending it to the intended user along with the password used for its encryption. The user for whom the text is intended should have the BCTextEncoder application installed on his/her machine. They will have to paste the encoded text into the Encoded text section and use the shared password, to decode it to plain text.

10. This concludes the demonstration showing how to encrypt and decrypting the data using BCTextEncoder.
11. Close all open windows and document all the acquired information.
12. Turn off the Admin Machine-1 virtual machine.

EXERCISE 4: ENCRYPT AND DECRYPT DATA USING BCTEXTENCODER

EXERCISE 5: **CREATE AND USE SELF-SIGNED CERTIFICATES**

Self-signed certificates are widely used for testing servers.

LAB SCENARIO

A security professional must possess a proper knowledge of creating this certificate as it validates the public key contained within the certificate belonging to the person, company, server, or other entity mentioned.

LAB OBJECTIVE

This lab demonstrates the creation of a self-signed certificate.

OVERVIEW OF SELF-SIGNED CERTIFICATE

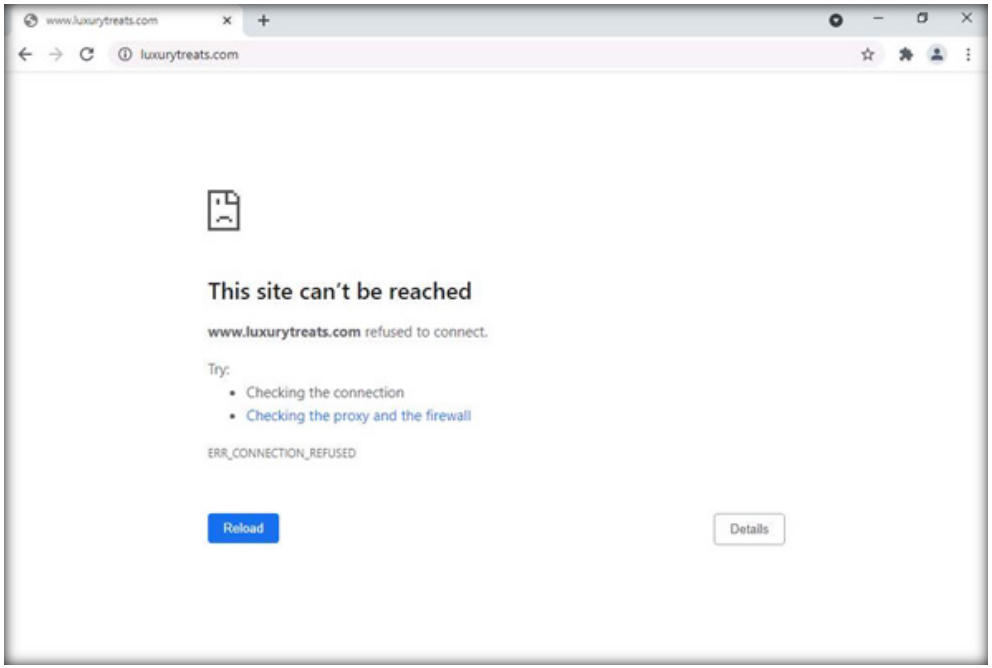
In self-signed certificates, a user creates a pair of public and private keys using a certificate creation tool such as Adobe Acrobat Reader, Java's keytool and Apple's Keychain and signs the document with the public key. The recipient requests the private key from the sender in order to verify the certificate. However, certificate verification rarely occurs due to the necessity to disclose the private key. This makes self-signed certificates useful only in a self-controlled testing environment.

Note: Ensure that the PfSense Firewall virtual machine is running.

1. Turn on the Web Server virtual machine.
2. Log in with the credentials Administrator and admin@123.

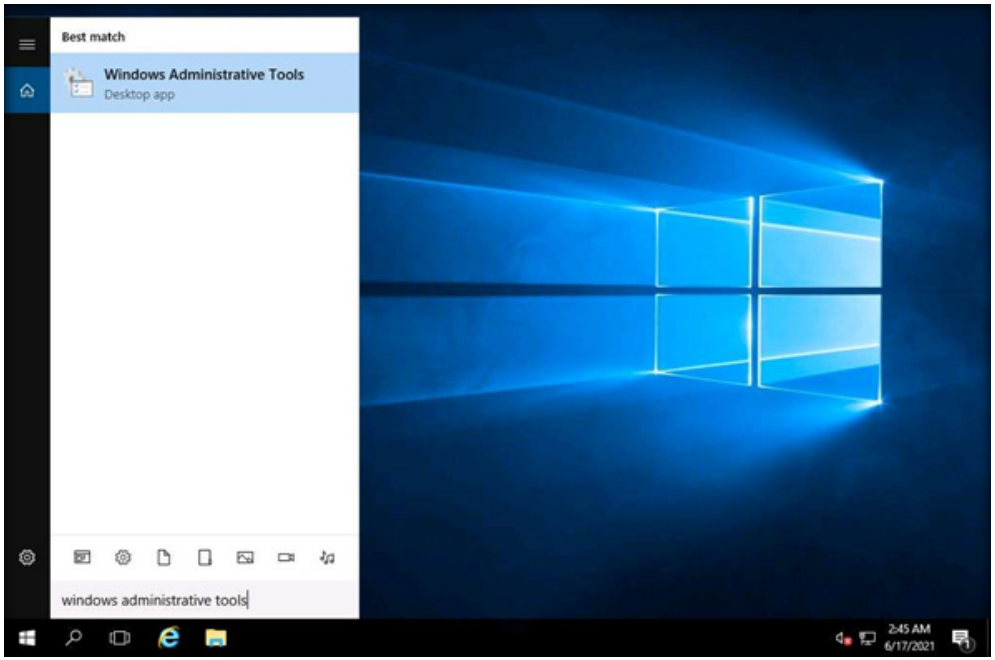
3. Before you start this task, you will need to check whether your local sites include a self-signed certificate.
4. Launch any web browser (here, Google Chrome), place the cursor in the address bar and type on https://www.luxurytreats.com, and press Enter.
5. As you are using an https channel to browse the website, a page displaying a This site can't be reached message will open.
6. As the site does not have a self-signed certificate, it displays a connection refused message, as shown in the screenshot below. Close the web browser.

EXERCISE 5: CREATE AND USE SELF-SIGNED CERTIFICATES



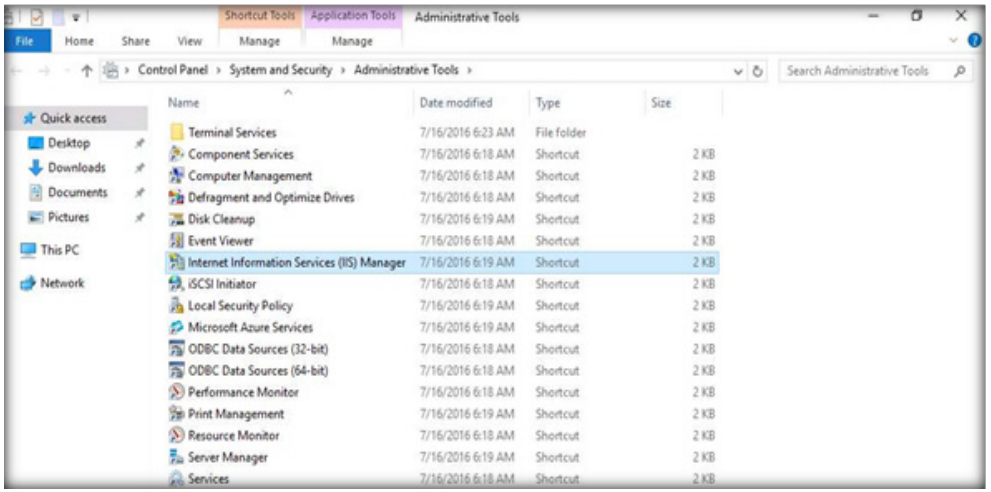
7. Click the Type here to search icon present at the bottom-left of Desktop and type windows administrative tools. Then, select Windows Administrative Tools from the results.

EXERCISE 5: CREATE AND USE SELF-SIGNED CERTIFICATES



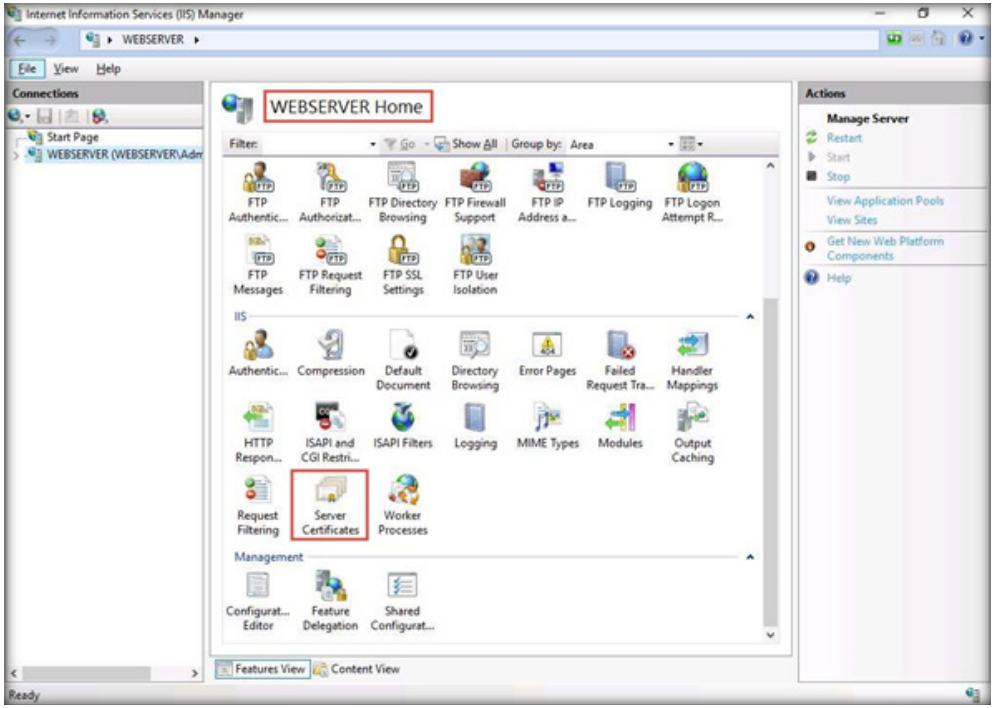
EXERCISE 5:
CREATE AND USE
SELF-SIGNED
CERTIFICATES

8. Double click on Internet Information Services (IIS) Manager from Administrative Tools.



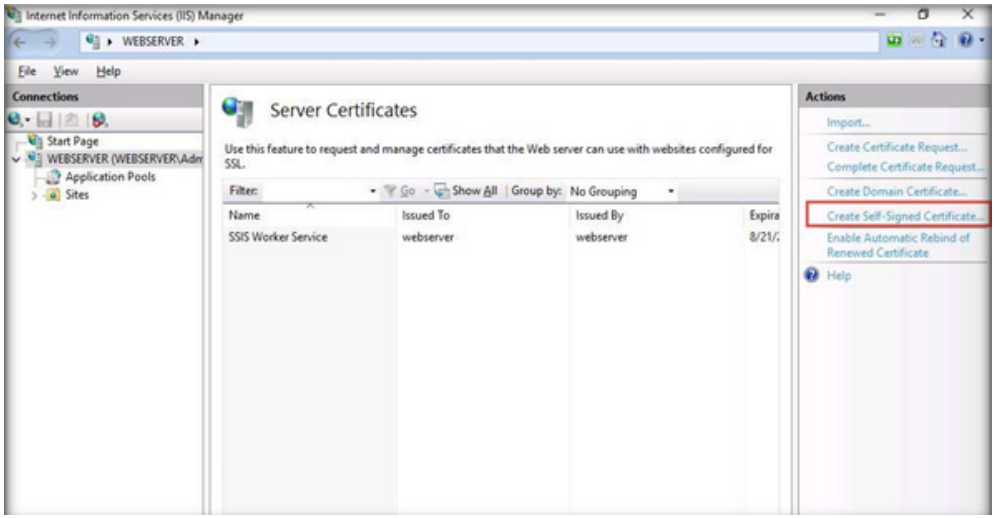
9. In the Internet Information Services (IIS) Manager window appears; click the machine name (WEBSERVER (WEBSERVER\Administrator)) under the Connections section from the left pane.
10. In WEBSERVER Home, double-click Server Certificates in the IIS section.

EXERCISE 5: CREATE AND USE SELF-SIGNED CERTIFICATES



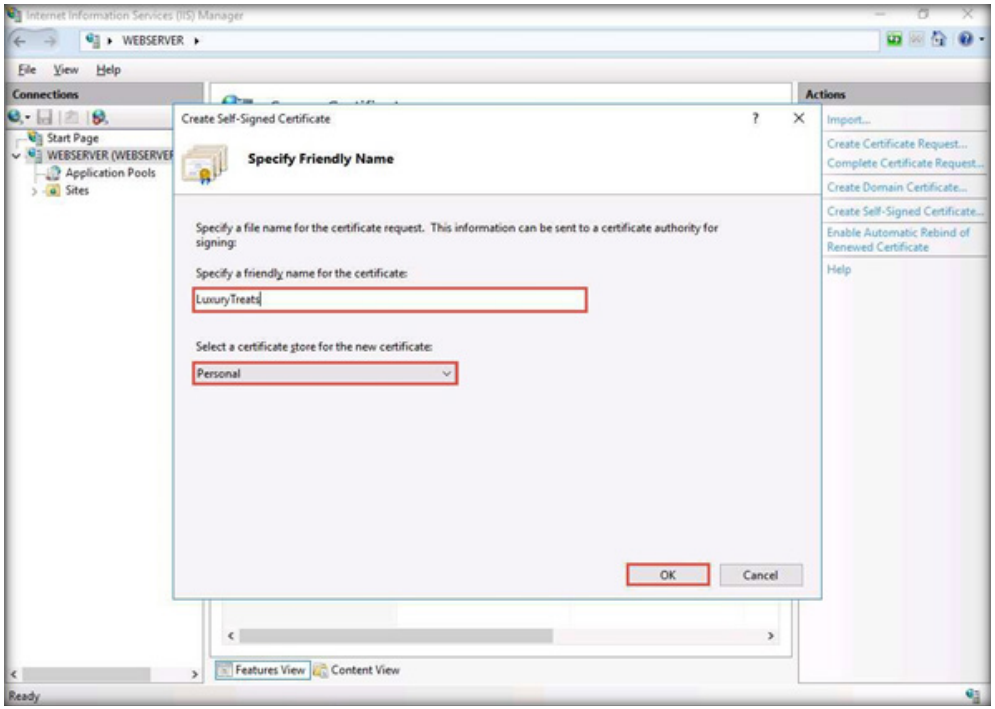
11. The Server Certificates wizard appears; click on Create Self-Signed Certificate... from the right pane in the Actions section.

EXERCISE 5: CREATE AND USE SELF-SIGNED CERTIFICATES



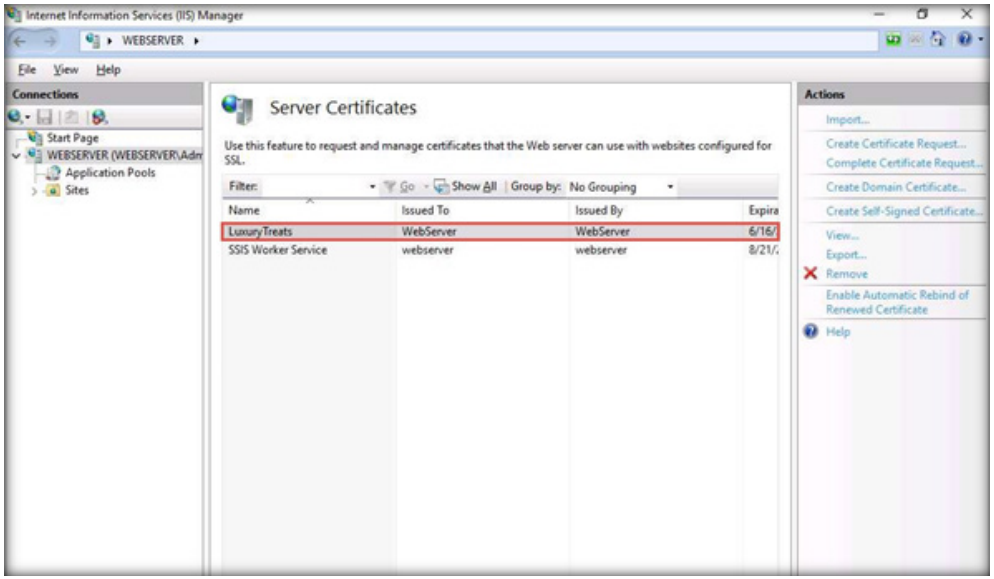
12. The Create Self-Signed Certificate window appears now. Type LuxuryTreats in the Specify a friendly name for the certificate field. Ensure that the Personal option is selected in the Select a certificate store for the new certificate field; then, click OK.

EXERCISE 5:
CREATE AND USE
SELF-SIGNED
CERTIFICATES



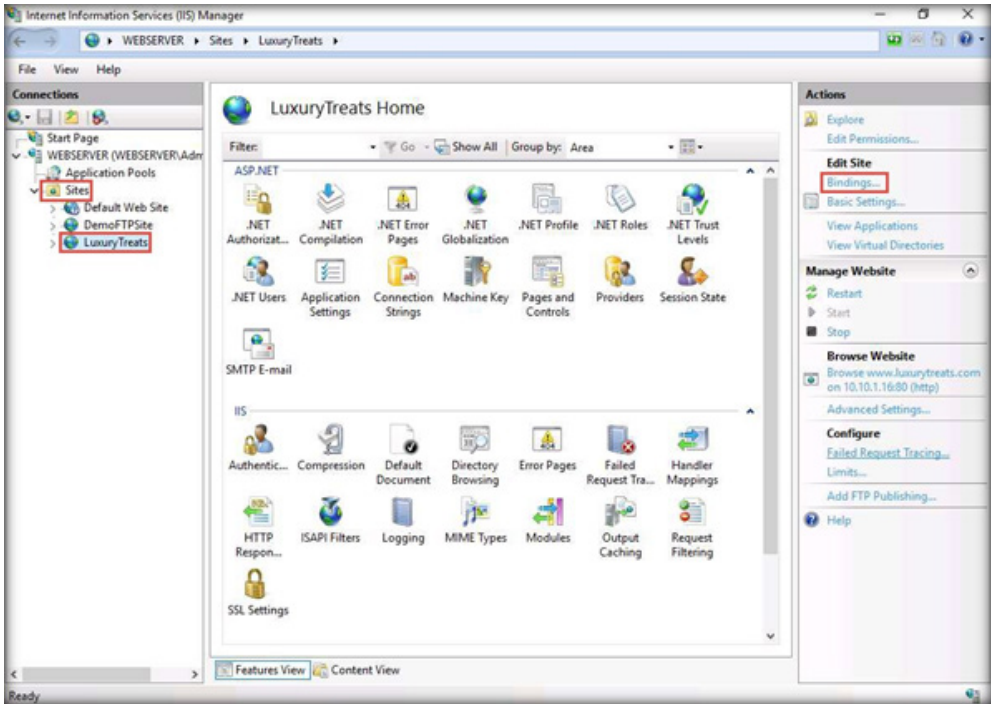
13. A newly created self-signed certificate will be displayed in the Server Certificates pane, as shown in the screenshot below.

EXERCISE 5:
CREATE AND USE
SELF-SIGNED
CERTIFICATES



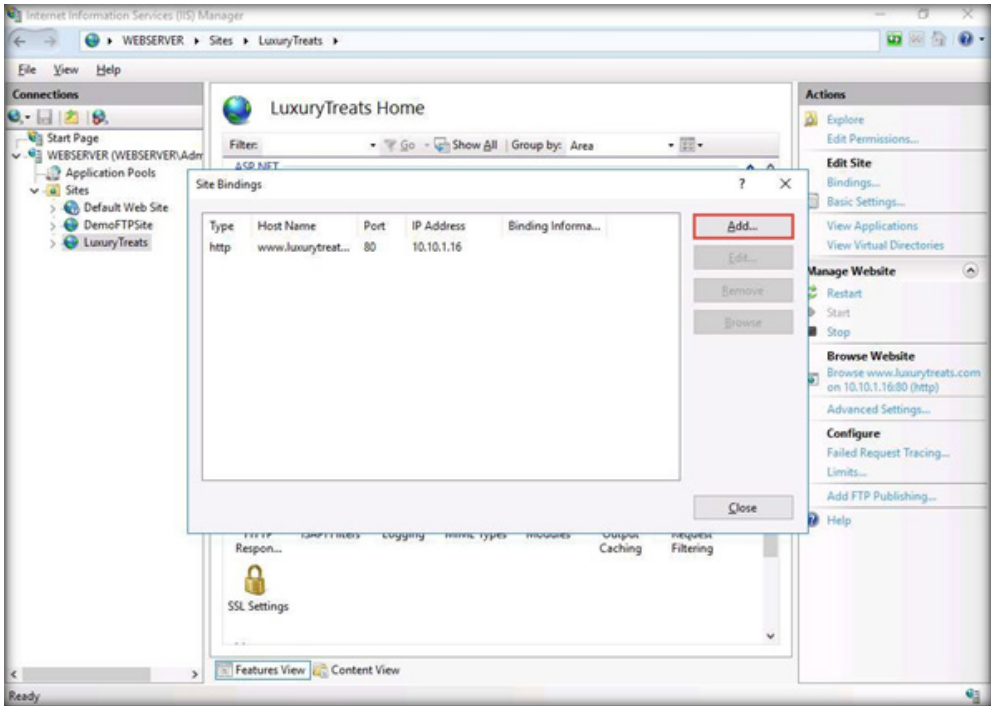
14. Expand the Sites node from the left pane, and select LuxuryTreats from the available sites. Then, click Bindings... from the right pane in the Actions section.

EXERCISE 5:
CREATE AND USE
SELF-SIGNED
CERTIFICATES



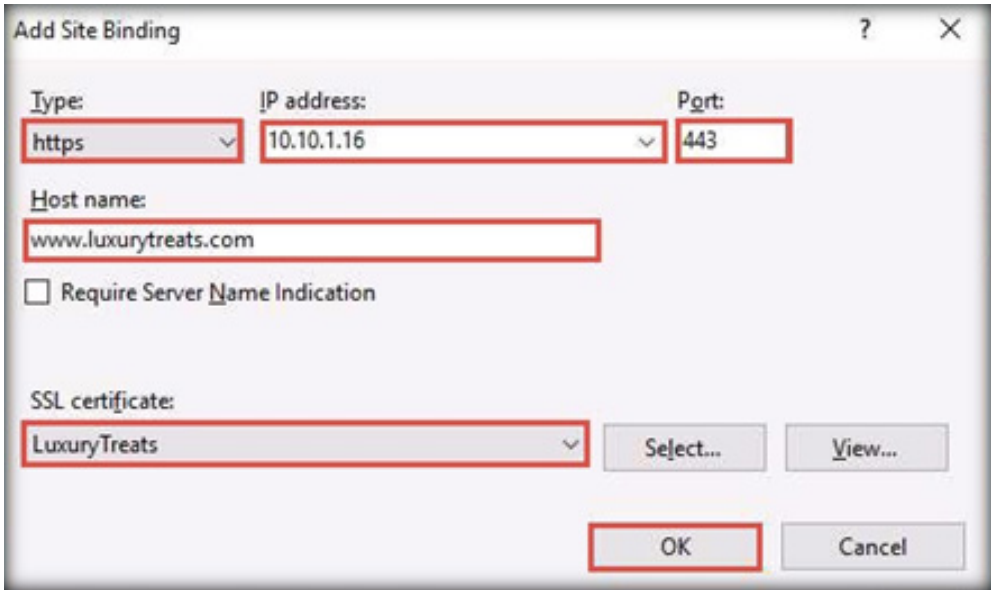
15. The Site Bindings window appears; click Add....

EXERCISE 5: CREATE AND USE SELF-SIGNED CERTIFICATES



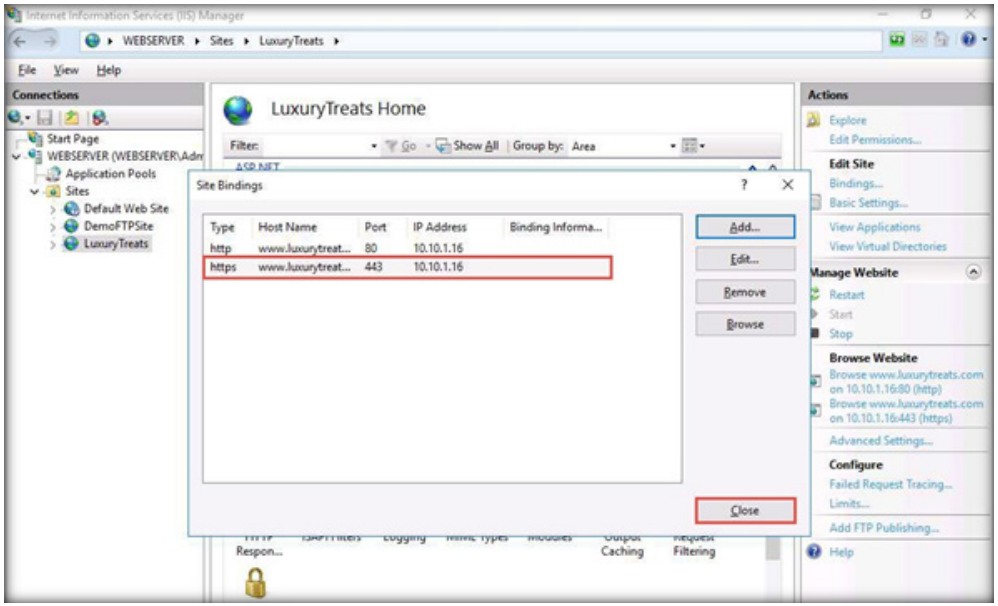
16. The Add Site Binding window appears; choose https from the Type field drop-down list. Once you choose the https type, the port number in the Port field automatically changes to 443 (the channel on which HTTPS runs).
17. Choose the IP address on which the site is hosted (here, 10.10.1.16).
18. Under the Host name field, type www.luxurytreats.com. Under the SSL certificate field, select LuxuryTreats from the drop-down list, and

EXERCISE 5:
CREATE AND USE
SELF-SIGNED
CERTIFICATES



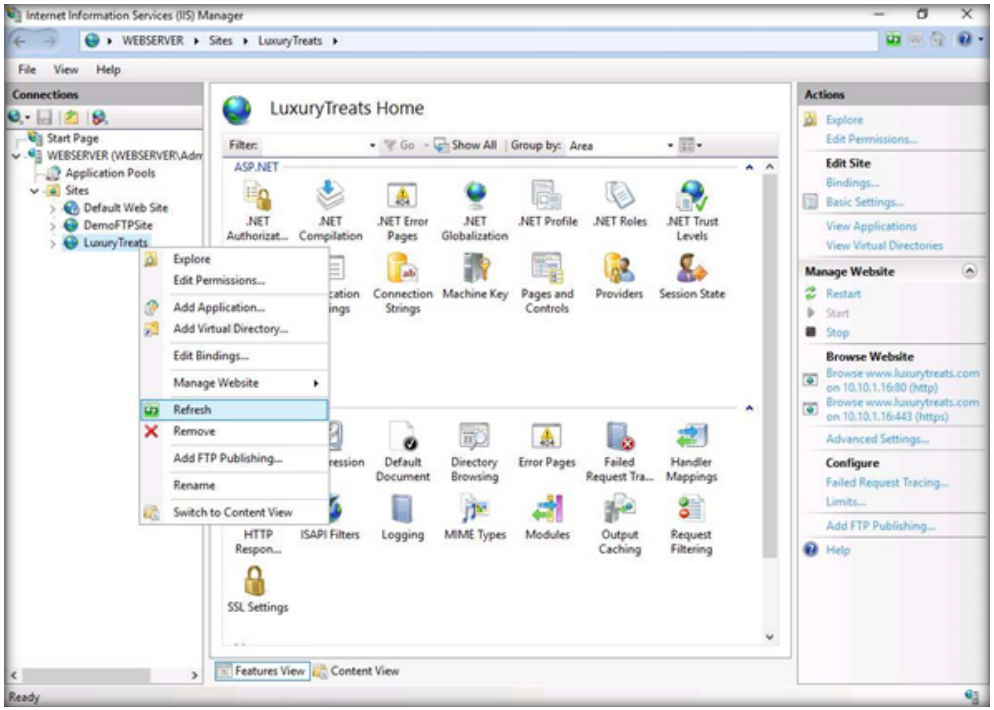
19. The newly created SSL certificate is added to the Site Bindings window; then, click Close.

EXERCISE 5: CREATE AND USE SELF-SIGNED CERTIFICATES



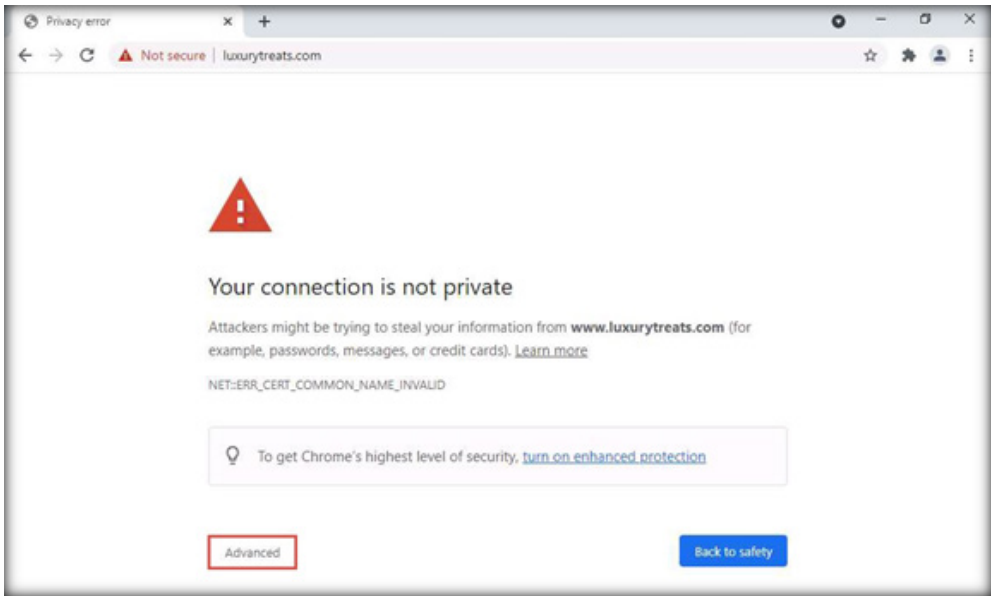
20. Now, right-click the name of the site for which you have created the self-signed certificate (here, LuxuryTreats) and click Refresh from the context menu.

EXERCISE 5:
CREATE AND USE
SELF-SIGNED
CERTIFICATES



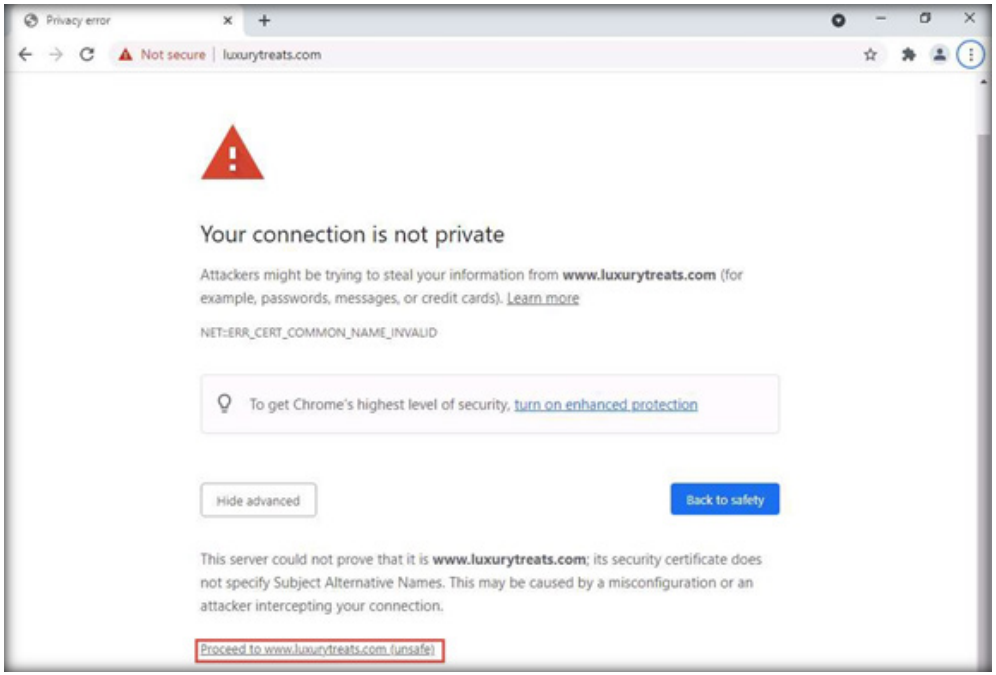
21. Minimize the Internet Information Services (IIS) Manager window.
22. Open the Google Chrome browser place the cursor in the address bar and type on https://www.luxurytreats.com, and press Enter.
23. The Your connection is not private message appears, click ADVANCED to proceed.

EXERCISE 5: CREATE AND USE SELF-SIGNED CERTIFICATES



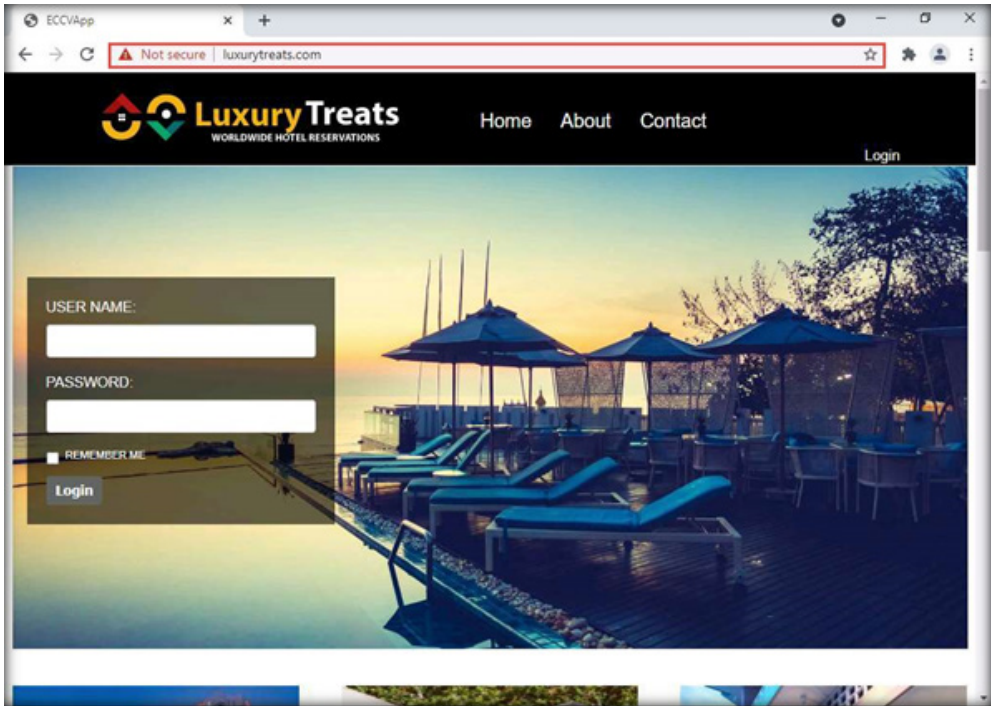
24. Click Proceed to www.luxurytreats.com (unsafe).

EXERCISE 5: CREATE AND USE SELF-SIGNED CERTIFICATES



25. Now you can see luxurytreats webpage with ssl certificate assigned to it, as shown in the screenshot below.

EXERCISE 5:
CREATE AND USE
SELF-SIGNED
CERTIFICATES



26. This concludes the demonstration showing how to create and use a self-signed certificate.
27. Close all open windows and document all acquired information.
28. Turn off the Web Server virtual machine.

EXERCISE 5: CREATE AND USE SELF-SIGNED CERTIFICATES

EXERCISE 6: **CREATE AND MANAGE CERTIFICATES USING OPENSSL**

Digital certificates allow a secure exchange of information between a sender and a receiver.

LAB SCENARIO

A trusted intermediary solution is used for securing public keys, where the public key is bound with the name of its owner. Owners need to acquire their public keys certified from the intermediary; the intermediary then issues certificates called digital certificates to the owners, which they can use to send the public key to a number of users. The sender applies for a digital certificate from the certificate authority (CA). Along with the encrypted message and the public key, the CA provides other identity validating information. The receiver accepts the encrypted message and uses the CA's public key to decode the digital certificate. This allows the receiver to identify the digital signature and obtain the sender's public key and other identification details.
A security professional must have a proper knowledge about creating and managing certificates using various tools such as OpenSSL in order to prevent unauthorized access to the organization's website and resources. In this task, we will create a private key, a certificate and we will further convert the certificate file format.

LAB OBJECTIVE

This lab demonstrates the creation and management of certificates using OpenSSL.

OVERVIEW OF DIGITAL CERTIFICATES

A digital certificate can hold information such as the name of the sender who applied for the certificate, expiration date, and a copy of the sender's public key digital signature of the CA. The receivers of the digital certificate can check its validity using the signature attached from the approved authorities using their private key. Each OS and web browser carries authorized certificates from the CA which enables easy validation. The main aim of implementing a digital certificate is to ensure nonrepudiation.

Note: Ensure that the PfSense Firewall virtual machine is running.

1. Turn on the Attacker Machine-2 virtual machine.
2. In the login page, the attacker username will be selected by default. Enter password as toor in the Password field and press Enter to log in to the machine.

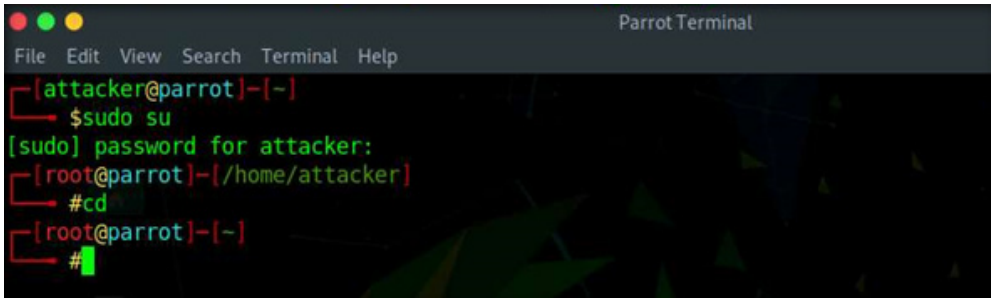
Note: If a Parrot Updater pop-up appears at the top-right corner of Desktop, ignore and close it.

Note: If a Question pop-up window appears asking you to update the machine, click No to close the window.

3. Click the MATE Terminal icon at the top of the Desktop window to open a Terminal window.
4. A Parrot Terminal window appears. In the terminal window, type sudo su and press Enter to run the programs as a root user.
5. In the [sudo] password for attacker field, type toor as a password and press Enter.

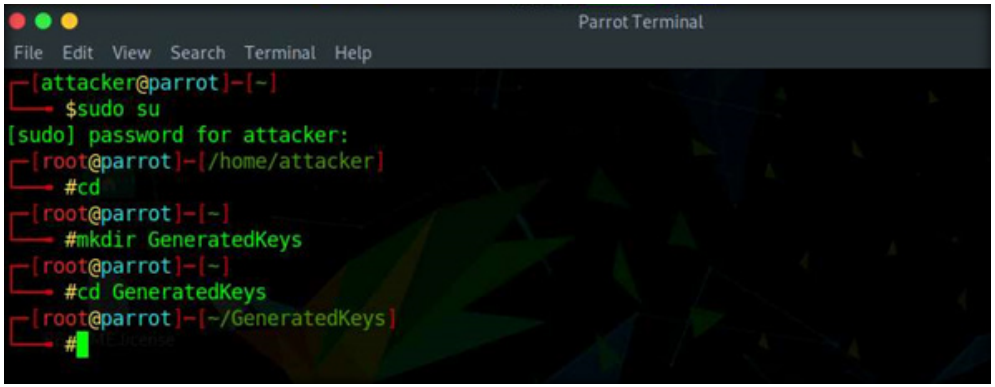
   Note: The password that you type will not be visible.
6. Now, type cd and press Enter to jump to the root directory.

EXERCISE 6: CREATE AND MANAGE CERTIFICATES USING OPENSSL



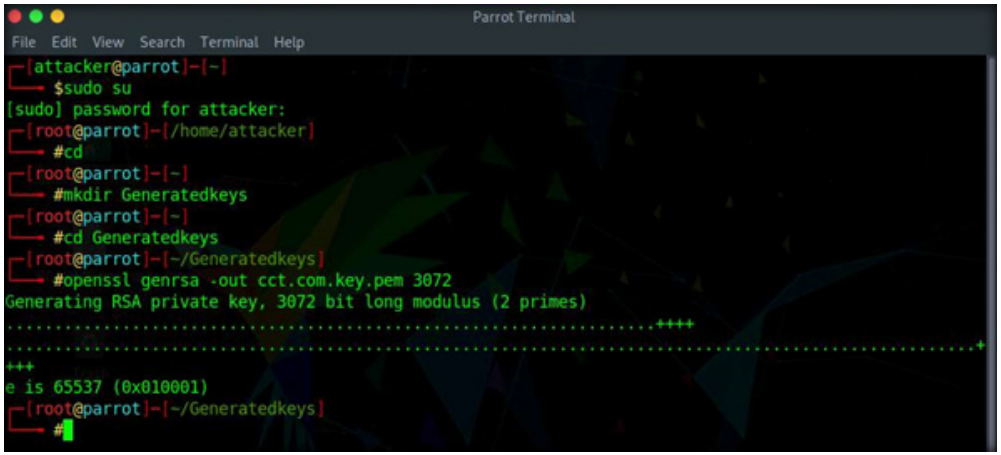
7. Type mkdir GeneratedKeys and press Enter to create a new directory (GeneratedKeys) inside the root directory.
8. Type cd GeneratedKeys and press Enter to navigate to the GeneratedKeys directory.

EXERCISE 6:
CREATE AND MANAGE CERTIFICATES USING OPENSSL



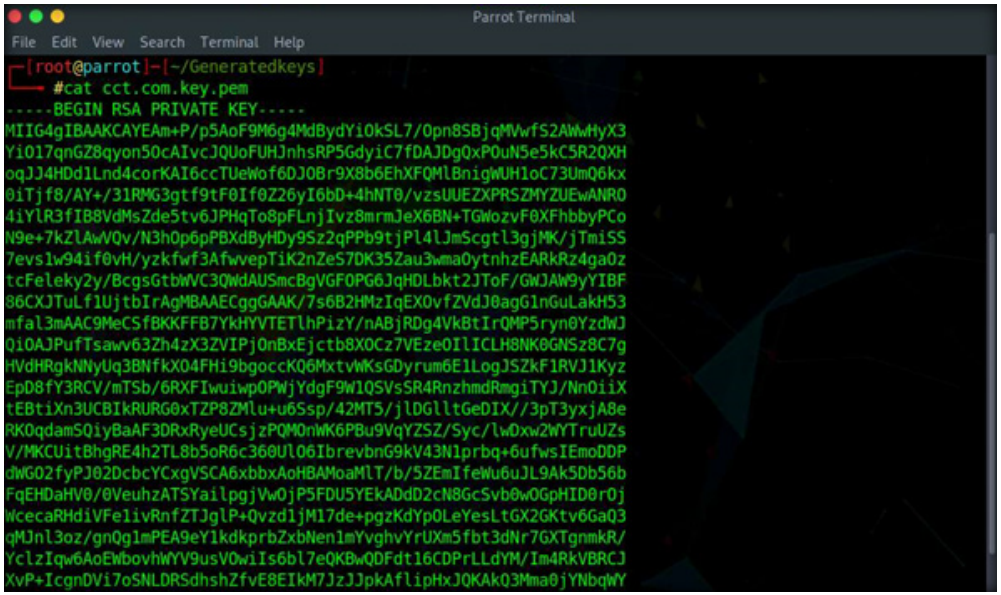
9. Type openssl genrsa -out cct.com.key.pem 3072 and press Enter to generate a private key.
   Note:
   - genrsa: specifies generating key with RSA algorithm.
   - out: specifies the location to output the certificate file itself.
   - 3072: specifies key size; the higher the value, the better the security it will provide. (2048 bit is the minimum key length, but 3072 is recommended)

EXERCISE 6:
CREATE AND MANAGE CERTIFICATES USING OPENSSL



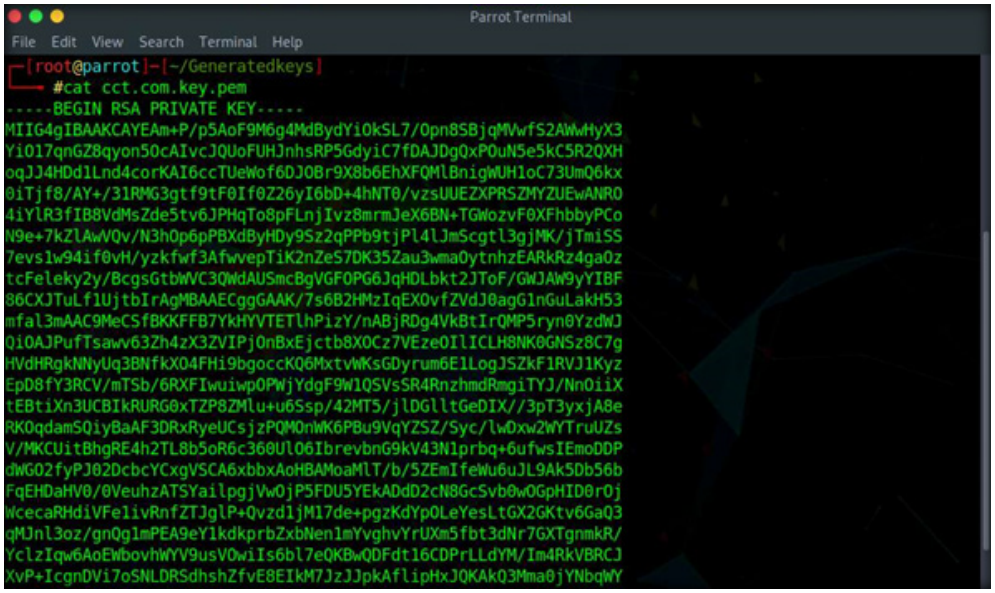
10. Type cat cct.com.key.pem and press Enter to view the generated private key.
    Note: Here, we have saved the private key with a pem file extension which is used in Linux systems.

EXERCISE 6:
CREATE AND MANAGE CERTIFICATES USING OPENSSL



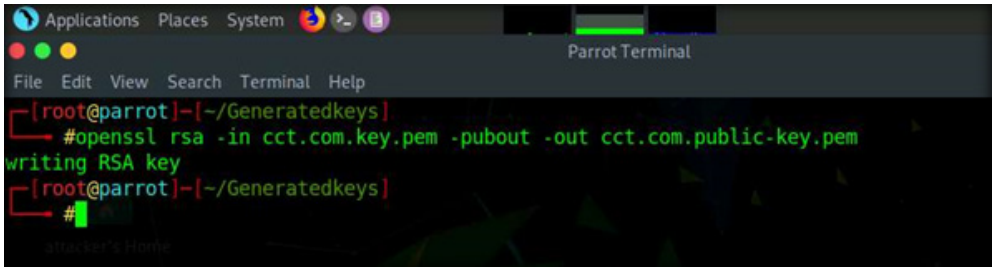
10. Type cat cct.com.key.pem and press Enter to view the generated private key.
    Note: Here, we have saved the private key with a pem file extension which is used in Linux systems.

EXERCISE 6:
CREATE AND MANAGE CERTIFICATES USING OPENSSL



11. Type openssl rsa -in cct.com.key.pem -pubout -out cct.com.public-key.pem and press Enter to export the public key.
    Note:
    - rsa: specifies RSA algorithm.
    - in: specifies the private key (here, cct.com.key.pem).
    - pubout: specifies exporting public key.
    - out: specifies the location to output the certificate file.

EXERCISE 6: CREATE AND MANAGE CERTIFICATES USING OPENSSL



12. You can view the public key using cat cct.com.public-key.pem command.
13. Here, we do not have a Certificate Authority (CA). Therefore, we will skip sending the CSR (Certificate Signing Request) to CA. Instead, we will sign the certificate by ourselves by generating a self-signed certificate.
14. Type openssl req -new -x509 -key cct.com.key.pem -out cct.com.cert.pem -days 360 and press Enter to create a self-signed certificate.

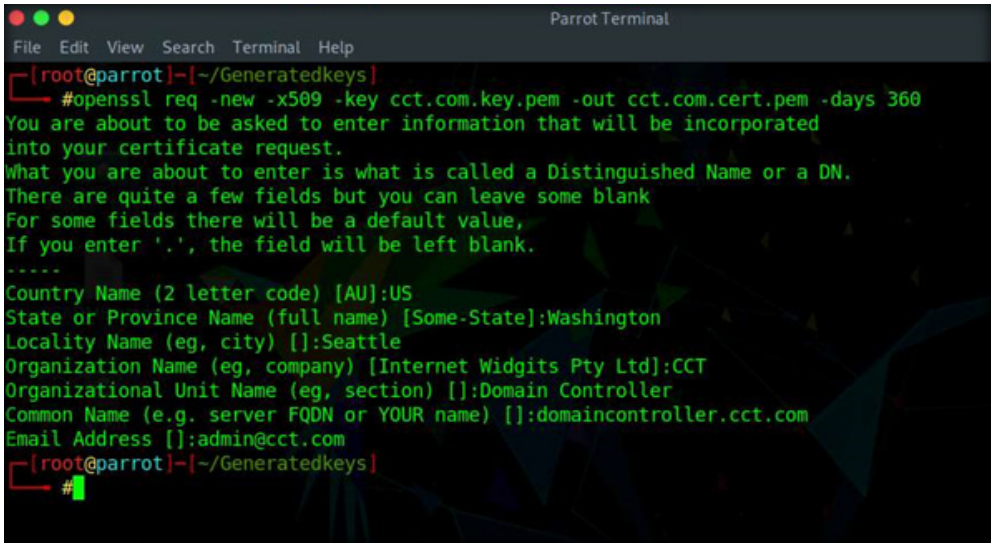
Note:

- req: specifies creating and processing certificate requests.
- new: specifies generating a new certificate.
- key: specifies the private key.
- out: specifies the location to output the certificate file.
- days: specifies the validity of a certificate (here, 360 days).

EXERCISE 6: CREATE AND MANAGE CERTIFICATES USING OPENSSL

15. Enter the additional information as follows:
    - Country Name: your country
    - State or Province Name: your state or province
    - Locality Name: your city
    - Organization Name: CCT
    - Organizational Unit Name: Domain Controller
    - Common Name: domaincontroller.cct.com
    - Email Address: admin@cct.com

EXERCISE 6: CREATE AND MANAGE CERTIFICATES USING OPENSSL



16. Now, we will convert the generated certificate and key files from the previous steps into a format allowed by Windows machine. In this case, we will convert PEM format into PKCS format.

Note:

- PEM Format: A format typically understood by Linux systems and Apache. File extensions: .pem, .key, .csr, .cert .
- PKCS Format: A format typically understood by Windows machines. File extensions: .pkcs, .p12, .p7b, .pfx.

17. Type openssl pkcs12 -export -inkey cct.com.key.pem -in cct.com.cert.pem -out cct.com.cert.pfx and press Enter to combine the exiting key and certificate files from PEM format to PFX file format.

EXERCISE 6: CREATE AND MANAGE CERTIFICATES USING OPENSSL

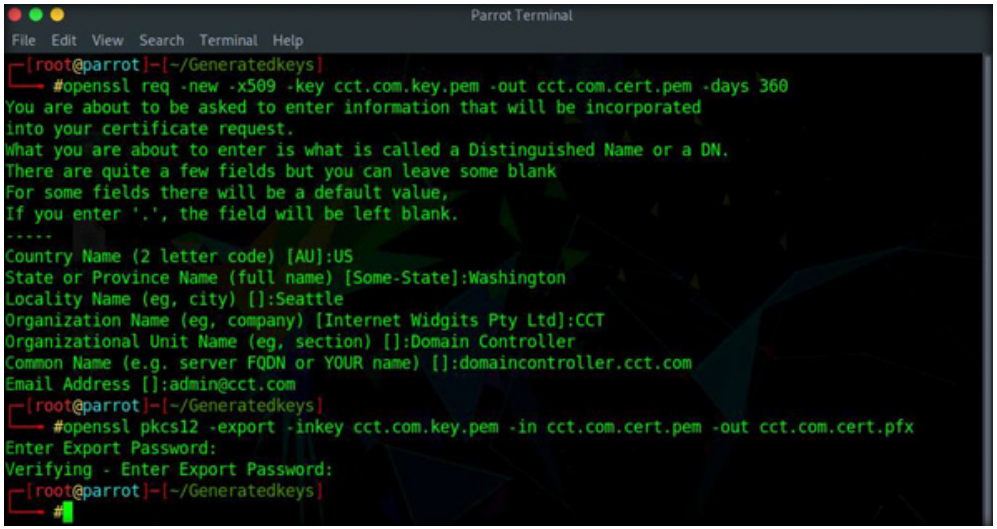
Note:

- export: specifies exporting private key.
- inkey: specifies private key (here, cct.com.key.pem).
- in: specifies exporting certificate (here, cct.com.cert.pem).
- out: specifies the location to output the certificate file.

18. In the Enter Export Password field, enter test@123 and press Enter. Similarly, enter test@123 in the Verifying - Enter Export Password field and press Enter.

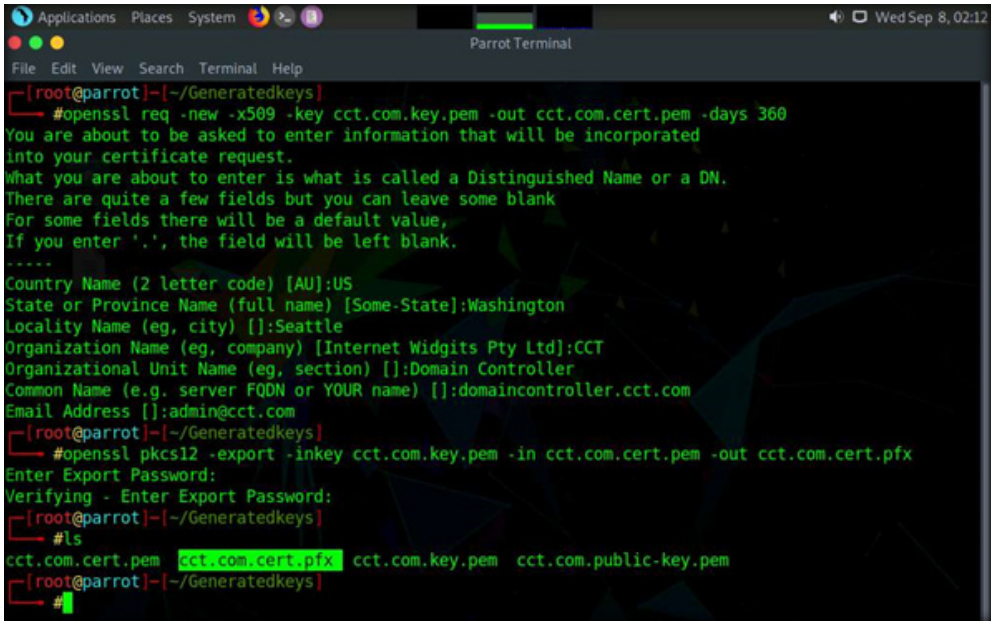
    Note: You can set a password of your choice.

EXERCISE 6: CREATE AND MANAGE CERTIFICATES USING OPENSSL



19. Type ls and press Enter to view the folder content.
20. Now, you can observe that, there are four files in the folder and a new file with a pfx extension has been created (cct.com.cert.pfx), as shown in the screenshot below.

EXERCISE 6:
CREATE AND MANAGE CERTIFICATES USING OPENSSL



21. This concludes the demonstration showing how to manage and create certificates using OpenSSL.
22. Close all open windows.
23. Turn off the Attacker Machine-2 virtual machine.

EXERCISE 6:
CREATE AND MANAGE CERTIFICATES USING OPENSSL

EXERCISE 7: **IMAGE STEGANOGRAPHY USING OPENSTEGO**

Image steganography allows you to conceal your secret message within an image.

LAB SCENARIO

A security professional must have a proper knowledge about stegnography and its types and how to conceal a secret message in a image file.

LAB OBJECTIVE

This lab demonstrates how to perform image steganography using tools such as OpenStego.

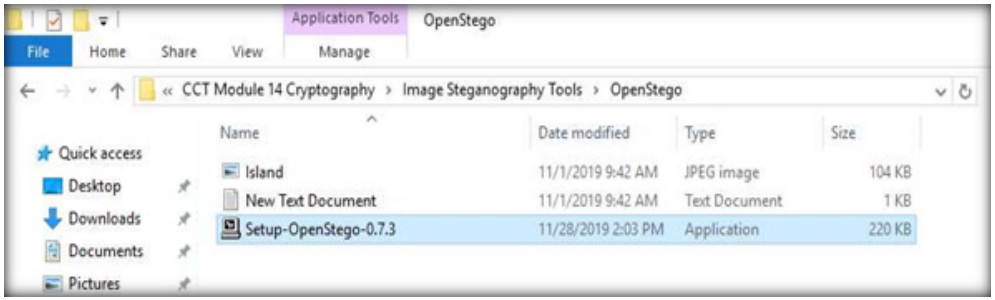
OVERVIEW OF DIGITAL CERTIFICATES

The redundant bits of an image can be exploited to conceal your message within it. These redundant bits are those parts of an image that have an insignificant effect on it if altered. The detection of this alteration is not easy. You can conceal your information within images of different formats (e.g., .PNG, .JPG, .BMP).

Note: Ensure that the PfSense Firewall virtual machine is running.

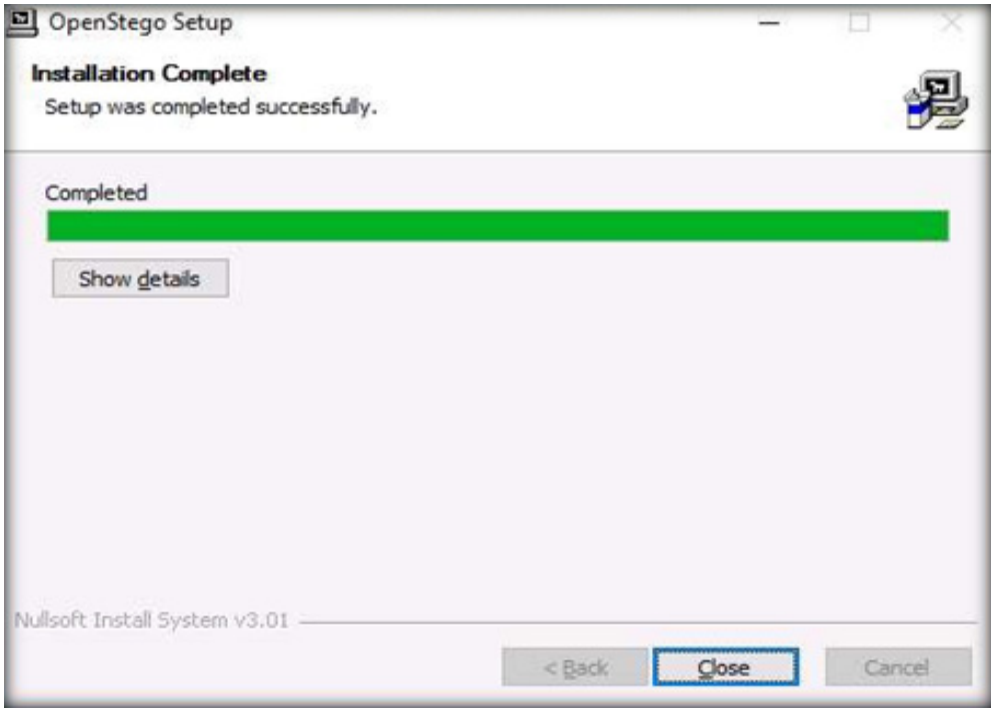
1. Turn on the Web Server virtual machine.
2. Log in with the credentials Administrator and admin@123.
3. Navigate to Z:\CCT Module 14 Cryptography\Image Steganography Tools\OpenStego and double-click Setup-OpenStego-0.7.3.exe.
   Note: If an Open File-Security Warning pop-up appears, click on Run.

EXERCISE 7: IMAGE STEGANOGRAPHY USING OPENSTEGO



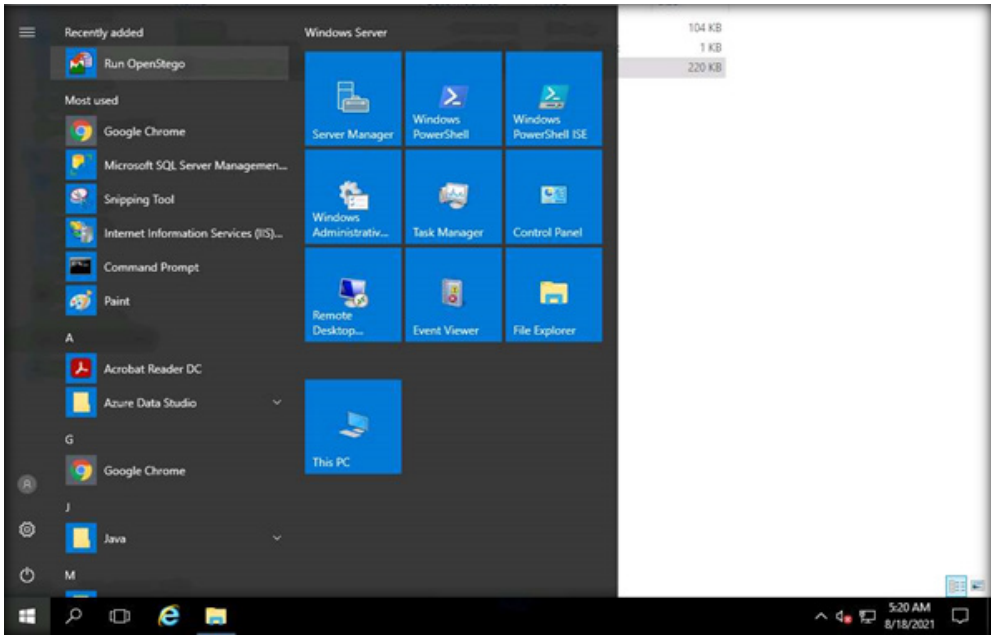
4. In the OpenStego Setup window appears; click I Agree. Follow the installation wizard and install the tool with the default settings.
5. In the Installation Complete wizard, click Close.

EXERCISE 7: IMAGE STEGANOGRAPHY USING OPENSTEGO



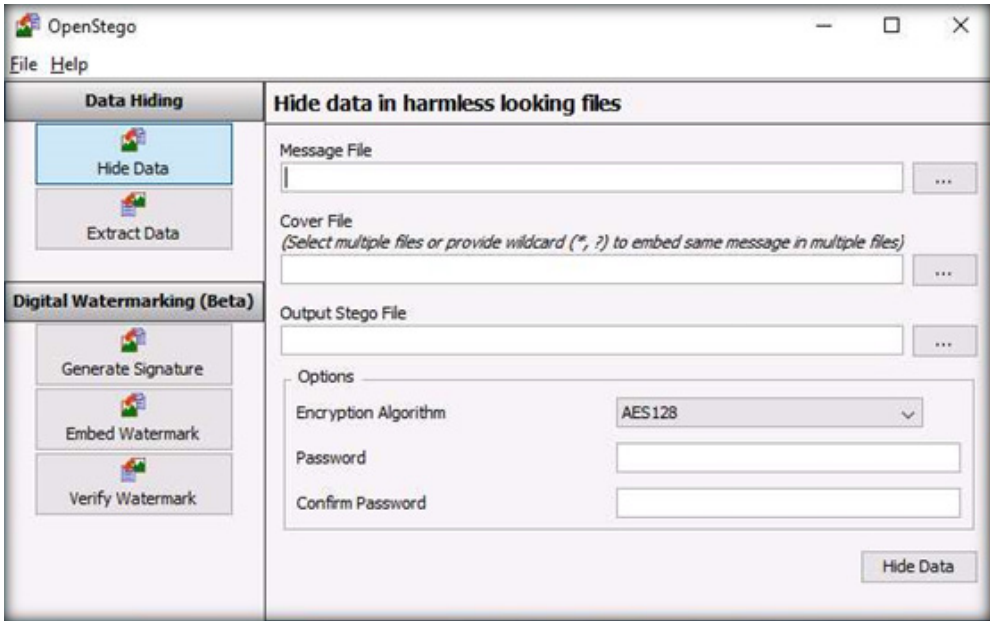
6. Click on the Start menu at the bottom-left corner of Desktop. Then, click Run OpenStego from the applications list to launch OpenStego.

EXERCISE 7:
IMAGE STEGANOGRAPHY USING OPENSTEGO



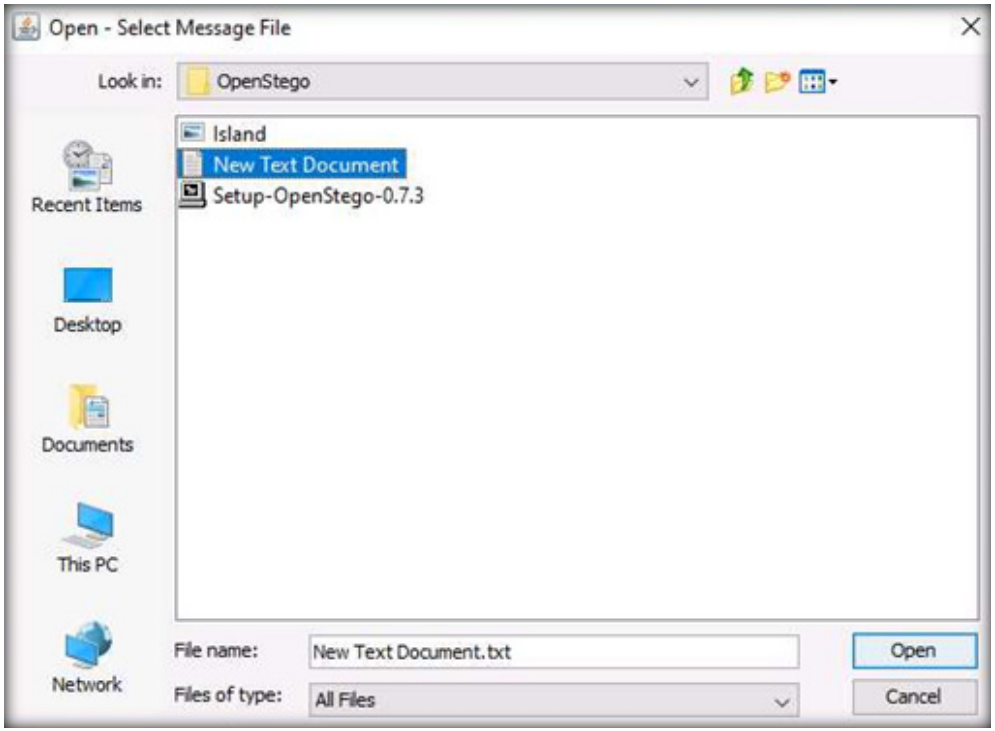
7. The OpenStego main window appears now, as shown in the screenshot below.

EXERCISE 7: IMAGE STEGANOGRAPHY USING OPENSTEGO



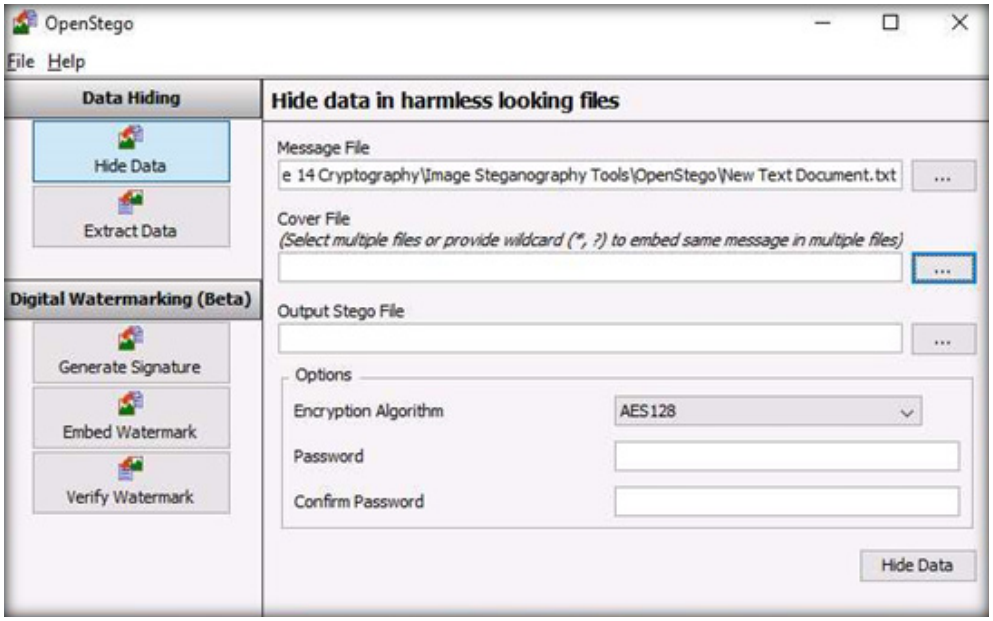
8. Click the ellipsis button next to the Message File section.
9. The Open - Select Message File window appears. Navigate to Z:\CCT-Tools\Module 14 Cryptography\Image Steganography Tools\ OpenStego, select New Text Document.txt, and click Open. Assume the text file contains sensitive information such as credit card and pin numbers.

EXERCISE 7: IMAGE STEGANOGRAPHY USING OPENSTEGO



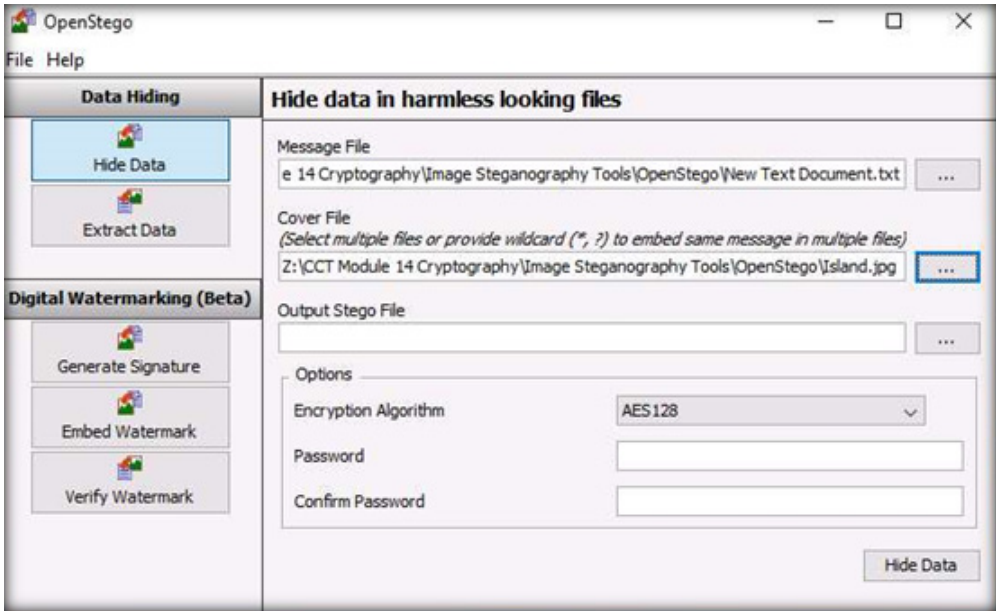
10. The location of the selected file appears in the Message File field.
11. Click on the ellipsis button next to Cover File.

EXERCISE 7: IMAGE STEGANOGRAPHY USING OPENSTEGO



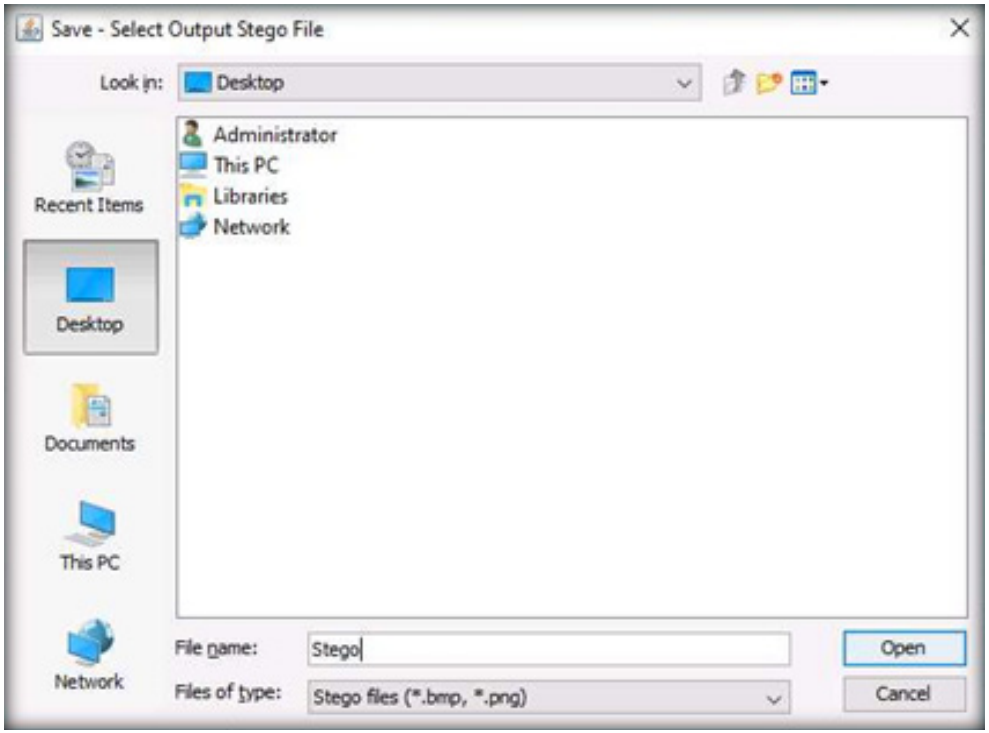
12. The Open - Select Cover File window appears. Navigate to Z:\CCT-Tools\Module 14 Cryptography\Image Steganography Tools\OpenStego, select Island.jpg, and click Open.
13. Now, both Message File and Cover File are uploaded. By performing steganography, the message file will be hidden in the designated cover file.

EXERCISE 7: IMAGE STEGANOGRAPHY USING OPENSTEGO



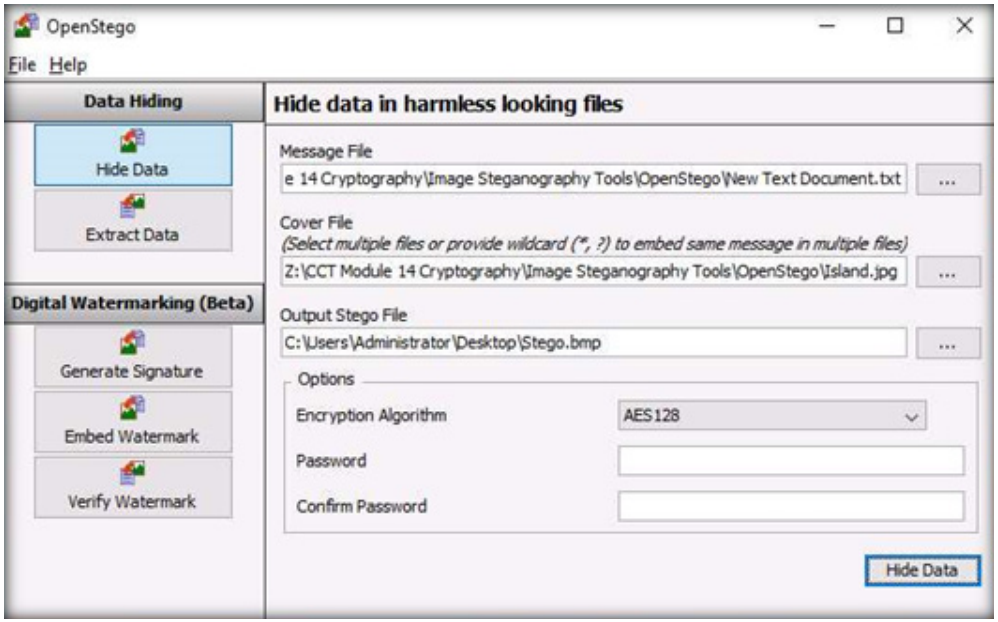
14. Click the ellipsis button next to Output Stego File.
15. The Save - Select Output Stego File window appears. Choose the location where you want to save the file. In this lab, the chosen location is Desktop.
16. Provide the file name as Stego and then, click Open.

EXERCISE 7: IMAGE STEGANOGRAPHY USING OPENSTEGO



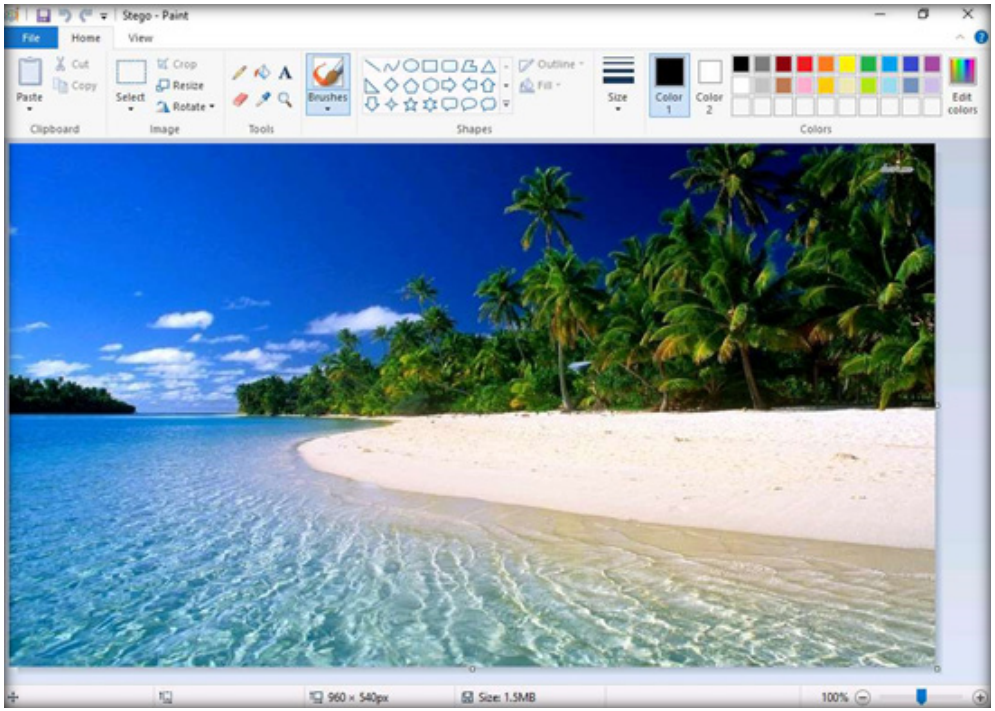
17. In the OpenStego window, click the Hide Data button.

EXERCISE 7: IMAGE STEGANOGRAPHY USING OPENSTEGO



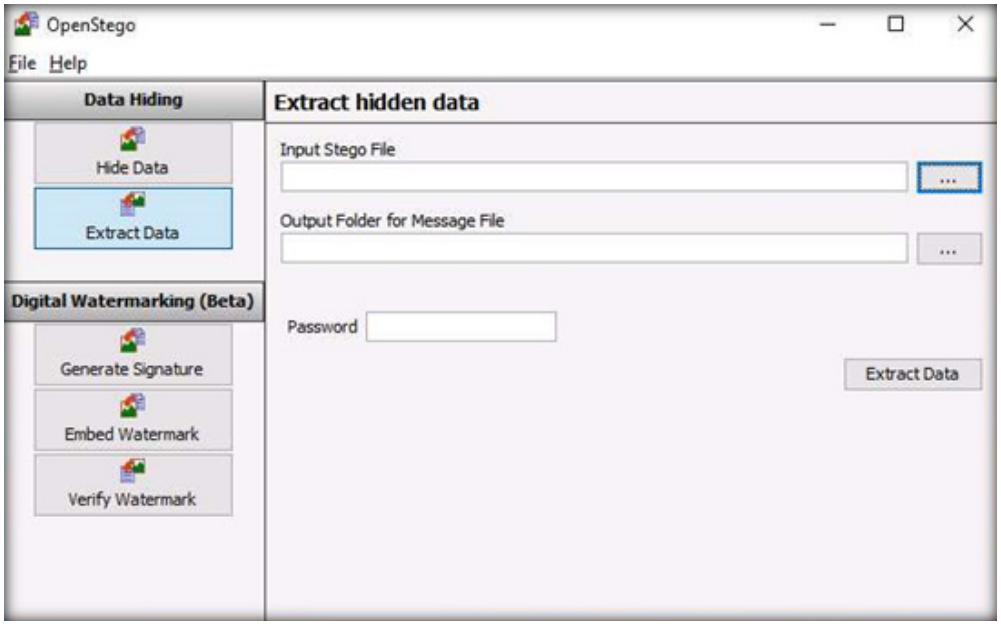
18. A Success pop-up appears, stating that the message has been successfully embedded. Then, click OK.
19. Minimize the OpenStego window. The image containing the secret message appears on Desktop. Double-click the image file (Stego .bmp) to view it.
20. You will see the image, but not the contents of the message (text file) embedded in it, as shown in the screenshot below.

EXERCISE 7: IMAGE STEGANOGRAPHY USING OPENSTEGO



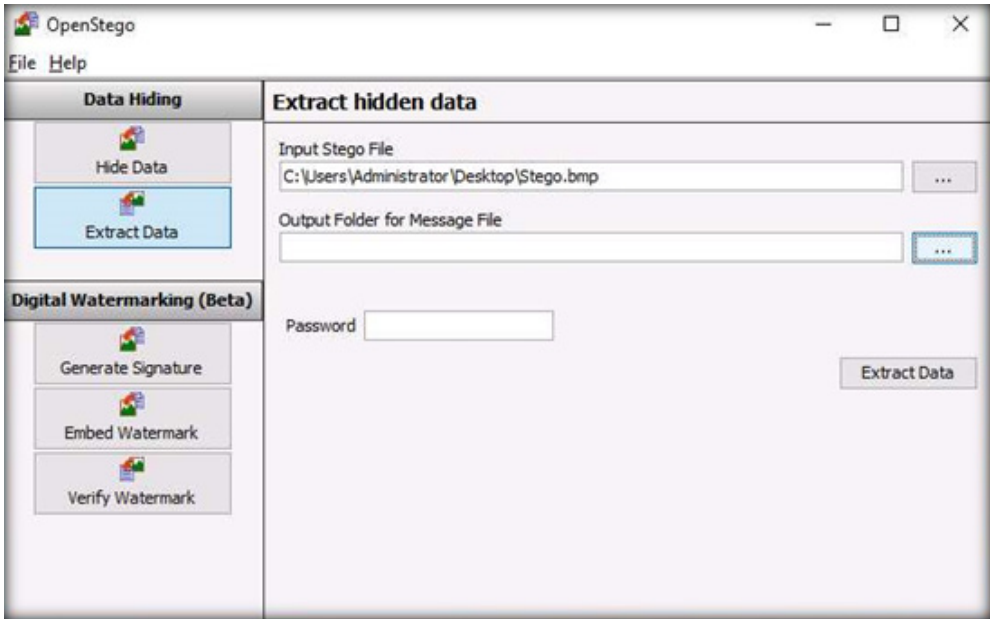
21. Close the Photos viewer window, switch to the OpenStego window, and click Extract Data in the left-pane.
22. Click the ellipsis button next to Input Stego File.

EXERCISE 7: IMAGE STEGANOGRAPHY USING OPENSTEGO



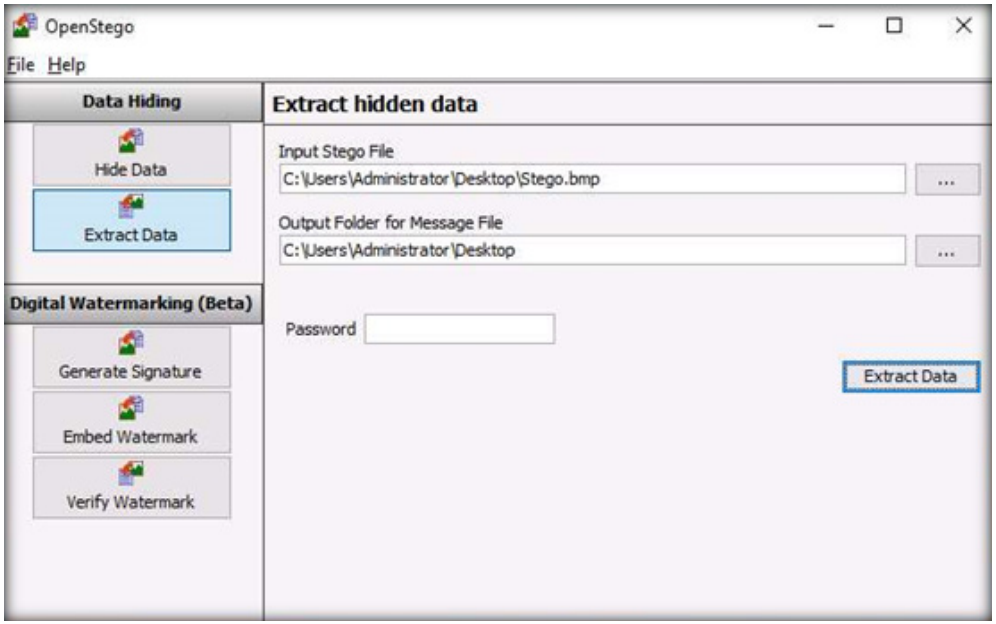
23. The Open - Select Input Stego File window appears. Navigate to Desktop, select Stego.bmp, and click Open.
24. Click the ellipsis button next to Output Folder for Message File.

EXERCISE 7: IMAGE STEGANOGRAPHY USING OPENSTEGO



25. The Select Output Folder for Message File window appears. Choose a location to save the message file (here, Desktop) and click Open.
26. In the OpenStego window, click the Extract Data button. This will extract the message file from the image and save it to Desktop.

EXERCISE 7: IMAGE STEGANOGRAPHY USING OPENSTEGO



27. A Success pop-up appears, stating that the message file has been successfully extracted from the cover file. Then, click OK.
28. The extracted image file (New Text Document.txt) is displayed on Desktop.
29. Close the OpenStego window, navigate to Desktop, and double-click New Text Document.txt.
30. The file displays all information contained in the text document, as shown in the screenshot below.

    Note: In real-time, an attacker might scan for images that contain hidden information and use steganography tools to decrypt their hidden information.

EXERCISE 7: IMAGE STEGANOGRAPHY USING OPENSTEGO

31. This concludes the demonstration showing how to perform image steganography using OpenStego.
32. Close all open windows and document all the acquired information.
33. Turn off Web Server and PfSense Firewall virtual machines.

EXERCISE 7: IMAGE STEGANOGRAPHY USING OPENSTEGO


EC-Council