Cryptography

Module 18

Cryptography

Cryptography is the study and art of hiding meaningful information in an unreadable format.

ICON KEY

- Valuable information
- Test your knowledge
- Web exercise
- Workbook review

Lab Scenario

With the increasing adoption of Internet–World Wide Web use for business and personal communication, securing sensitive information such as credit-card numbers, personal identifiable information, bank account numbers, secret messages, and so on is becoming increasingly more important. Today's information-based organizations extensively use Internet for e-commerce, market research, customer support, and a variety of other activities. Data security is critical to online business and privacy of communication.

The ability to protect and secure information is vital to the growth of electronic commerce and to the growth of the Internet itself. Many people need or want to use communications and maintain data security. The encryption of data plays a major role in doing so. For example, banks all over the world use encryption methods to process financial transactions involving the transfer of huge amounts of money. They also use encryption methods to protect their customers' ID numbers at bank automated teller machines. There are many companies and even shopping malls selling anything from flowers to wine over the Internet, and these transactions are made by the use of credit cards and secure Internet browsers that include encryption. Internet customers want to know that their credit-card information and other financial details will remain private and secure. But this can only be accomplished by the use of strong and impenetrable encryption methods.

As part of a security assessment, you have to suggest to your target organization that it use proper encryption techniques to protect data, both in storage and during transmission. The labs in this module demonstrate how you can use encryption to protect information systems.

Tools demonstrated in this lab are available in D:\CEH-Tools\CEHv9 Module 18 Cryptography

Lab Objectives

This lab will show you how to use encryption tools to encrypt data. It will teach you how to:

- Use encrypting/decrypting techniques
- Generate Hashes and checksum files

Lab Environment

To complete this lab, you will need:

- A computer running Window Server 2012
- A computer running Windows 8.1 in virtual machine
- A computer running Windows Server 2008 in virtual machine
- A computer running Kali Linux in virtual machine

- A Web browser with Internet access
- Administrative privileges to run the tool

Lab Duration

Time: 115 Minutes

Overview of Cryptography

Cryptography is the practice and study of hiding information. Modern cryptography intersects the disciplines of mathematics, computer science, and electrical engineering. Cryptology before the modern age was almost synonymous with encryption, the conversion of information from a readable state to one apparently without sense. It helps in securing data from interception and compromise during online transmissions. This module provides a comprehensive understanding of different crypto systems and algorithms, one-way hash functions, public-key infrastructure (PKI), and the different ways cryptography can help in ensuring privacy and security of online communication. The module also covers various cryptography tools used to encrypt sensitive data.

Lab Tasks

TASK 1

Overview

Recommended labs to assist you in cryptography are:

- Calculating MD5 Hashes and Verifying File Integrity Using **Quick Checksum Verifier**
- Calculating One-way Hashes Using **HashCalc**
- Calculating MD5 Hashes Using **MD5 Calculator**
- Understanding File and Text Encryption Using **CryptoForge**
- Basic Data Encryption Using **Advanced Encryption Package**
- Encrypting and Decrypting the Data Using **BCTextEncoder**
- Exploiting **OpenSSL Heartbleed** Vulnerability on a Https website
- Creating and Using **Self-Signed Certificates**
- Basic Disk Encryption Using **VeraCrypt**
- Basic Data Encrypting Using **Rohos Disk Encryption**
- Basic Data Encryption Using **CrypTool**

Lab Analysis

Analyze and document the results related to this lab exercise. Provide your opinion on your target's security posture and exposure.

PLEASE TALK TO YOUR INSTRUCTOR IF YOU HAVE QUESTIONS RELATED TO THIS LAB.

Lab 1

Calculating MD5 Hashes and Verifying File Integrity Using Quick Checksum Verifier

Checksum Verifier generates and checks file integrity by secure time proven algorithms like MD5 and SHA-1. You can easily create checksums (the digital fingerprints) of files and later verify their integrity. The operation is very easy—just two steps: load the file, and paste the predefined checksum.

ICON KEY

- Valuable information
- Test your knowledge
- Web exercise
- Workbook review

Tools demonstrated in this lab are available in D:\CEH-Tools\CEHv9 Module 18 Cryptography

Lab Scenario

A checksum, or hash sum, is a small datum from a block of digital data for detecting errors, which may have been introduced during its transmission or storage. Checksum Verifier generates and checks file integrity by secure time proven algorithms like MD5 and SHA-1. You can easily create checksums (the digital fingerprints) of files and verify their integrity. As an Expert Ethical Hacker and Penetration Tester, you will need to use hashes and checksum verifiers at every stage of your assessment to ensure the integrity of data collected.

Lab Objectives

This lab will show you how to check file integrity:

- Generate Hashes and checksum files

Lab Environment

To complete this lab, you will need:

- Quick Checksum Verifier located at **D:\CEH-Tools\CEHv9 Module 18 Cryptography\MD5 Hash Calculators\Quick Checksum Verifier**
- You can also download the latest version of Quick Checksum Verifier from the link http://www.bitdreamers.com/en/products/checksum-verifier
- If you decide to download the latest version, then screenshots shown in the lab might differ
- Follow the wizard driven installation instructions
- Run this tool in Windows Server 2012
- Administrative privileges to run tools

Lab Duration

Time: 5 Minutes

Overview of Lab

A *checksum* is a count of the number of bits in a transmission unit that is included with the unit so that the receiver can check to see whether the same number of bits arrived. If the counts match, it is assumed that the complete transmission was received.

Lab Tasks

TASK 1

View Sample.txt File

1. In this lab, we are going verify the MD5 checksum values before editing a file and after editing of the file.
2. Already, we have created a **Sample.txt** file and placed it in D:\CEH-Tools\CEHv9 Module 18 Cryptography\MD5 Hash Calculators\Quick Checksum Verifier.
3. Open Sample.txt and check the text. Don't edit or manipulate the information in the Sample.txt file.

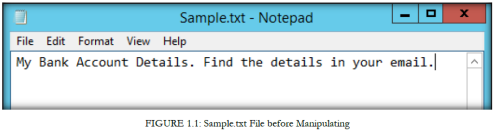


FIGURE 1.1: Sample.txt File before Manipulating

4. Now install and launch Quick Checksum Verifier. If you are launching it for the first time, it will ask you to choose a language; do so and click **Next**.

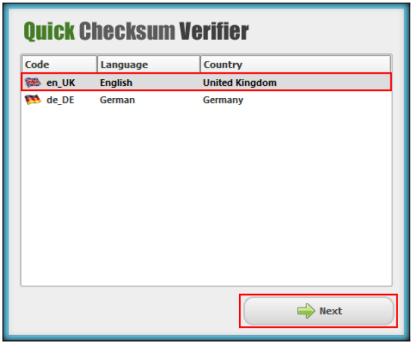


FIGURE 1.2: Quick Checksum Verifier Language

5. The Quick Checksum Verifier "thank you" window appears; click **Next** to continue.

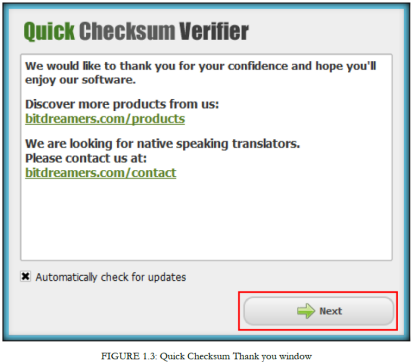


FIGURE 1.3: Quick Checksum Thank you window

TASK 2

Observe MD5 Checksum Value

6. The Quick Checksum Verifier main window appears; under **Calculate Checksum,** choose **MD5 (Message-Digest Algorithm)** from the drop down list.

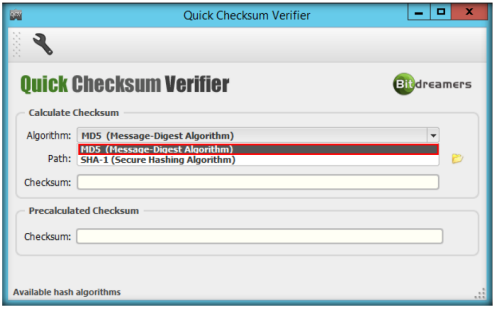


FIGURE 1.4: Quick Checksum Main Window

7. Now, click on **Browse for folder** icon next to the **Path** field to open the file you need to verify the Checksum value.

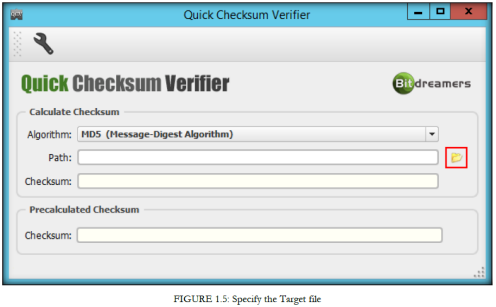


FIGURE 1.5: Specify the Target file

8. Navigate to **D:\CEH-Tools\CEHv9 Module 18 Cryptography\MD5 Hash Calculators\Quick Checksum Verifier**, select **Sample.txt**, and click **Open**.

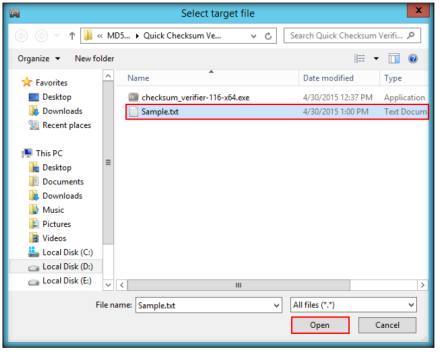


FIGURE 1.6: Standard Windows Browse for Folder

9. Once you have provided the path, Quick Checksum Verifier will automatically calculate the Checksum of the required file.
10. Make a note of the Checksum value, once it is generated.

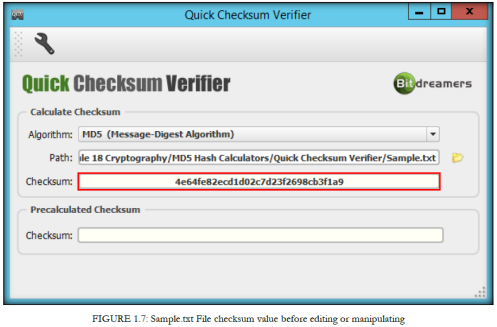


FIGURE 1.7: Sample.txt File checksum value before editing or manipulating

TASK 3

Observe MD5 Checksum Value

11. Navigate to **D:\CEH-Tools\CEHv9 Module 18 Cryptography\MD5 Hash Calculators\Quick Checksum Verifier**; open **Sample.txt,** alter the file's text, and save the file in the same location.

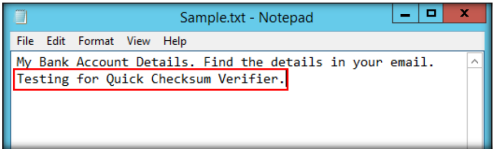


FIGURE 1.8: Sample.txt File Manipulated Sample.txt file

12. Follow **steps 6–8**, then compare the Checksum values of the file before and after manipulating its text.

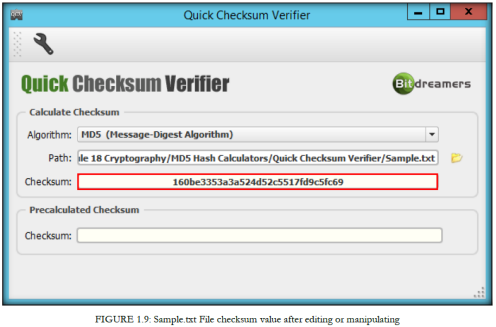


FIGURE 1.9: Sample.txt File checksum value after editing or manipulating

13. In real time, when a person sends sensitive information to another person, the sender will calculate its hashes and send the information (along with the hash value) through a medium (e.g., email). When the person on the other side receives the mail, he/she will note the hash value, copy the message, and calculate its value. If the calculated value and the hash value noted earlier tally, it means that the received data hasn't been modified by a third party during transit and is thus legitimate.
14. Hash calculation is mainly performed to check data integrity.

Lab Analysis

Document all Hash, MD5, and CRC values for further references.

PLEASE TALK TO YOUR INSTRUCTOR IF YOU HAVE QUESTIONS RELATED TO THIS LAB.

Internet Connection Required	
☐ Yes	☑ No
Platform Supported	
☑ Classroom	☑ iLabs

Lab 2

Calculating One-Way Hashes Using HashCalc

HashCalc enables you to compute multiple hashes, checksums and HMACs for files, text and hex strings. It supports MD2, MD4, MD5, SHA1, SHA2 (SHA256, SHA384, SHA512), RIPEMD160, PANAMA, TIGER, CRC32, ADLER32, and the hash used in eDonkey and eMule tools.

ICON KEY

- Valuable information
- Test your knowledge
- Web exercise
- Workbook review

Lab Scenario

Message digests or one-way hash functions distill the information contained within a file (small or large) into a single fixed-length number, typically between 128 and 256 bits in length. If any given bit of the function's input is changed, every output bit has a 50% chance of changing. Given an input file and its corresponding message digest, it should be nearly impossible to find another file with the same message digest value, as it is computationally unfeasible to have two files with the same message digest value.

Hash algorithms are widely used in a wide variety of cryptographic applications, and is useful for digital signature applications, file integrity checking, and storing passwords.

Lab Objectives

This lab will show you how to encrypt data and how to use it. It will teach you how to:

- Use encrypting/decrypting command
- Generate Hashes and checksum files

Tools demonstrated in this lab are available in D:\CEH-Tools\CEHv9 Module 18 Cryptography

Lab Environment

To complete this lab, you will need:

- HashCalc located at **D:\CEH-Tools\CEHv9 Module 18 Cryptography\MD5 Hash Calculators\HashCalc**

- You can also download the latest version of HashCalc from the link http://www.slavasoft.com/hashcalc/
- If you decide to download the latest version, then screenshots shown in the lab might differ
- Follow the wizard driven installation instructions
- Run this tool in Windows Server 2012
- Administrative privileges to run tools

Lab Duration

Time: 5 Minutes

Overview of Hash

HashCalc is a fast and easy-to-use calculator that allows computing message **digests, checksums,** and **HMACs for files**, as well as for **text and hex strings**. It offers a choice of 13 of the most popular hash and checksum algorithms for calculations.

Lab Tasks

TASK 1

Calculate the Hash

1. Launch **HashCalc** application from **Apps** screen.

FIGURE 2.1: Launching HashCalc application

HashCalc simple dialog-size interface dispenses with glitz to plainly list input and results.

2. The main window of **HashCalc** appears; select the type of **Data format** (here, **Text string**) from dropdown list.

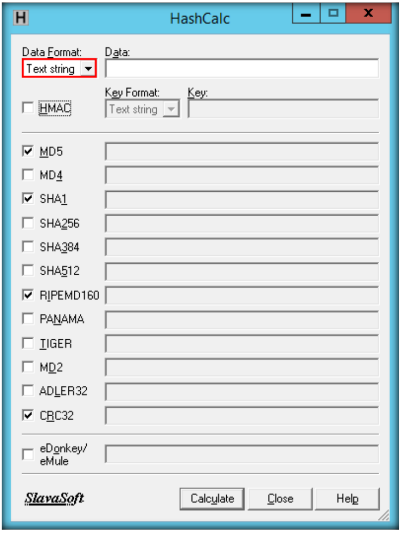


FIGURE 2.2: HashCalc main window

Hash algorithms support three input data formats: file, text string, and hexadecimal string.

3. As you are specifying the data format as Text string, the application accepts text strings and converts them to their respective hashes.

4. **Enter** data which you would like to **calculate**.
5. Choose the appropriate **Hash algorithms** by selecting their respective checkboxes.
6. In this lab, **MD5**, **SHA1**, **RIPEMD160** and **CRC32** hash algorithms have been selected.
7. Now, click **Calculate**.

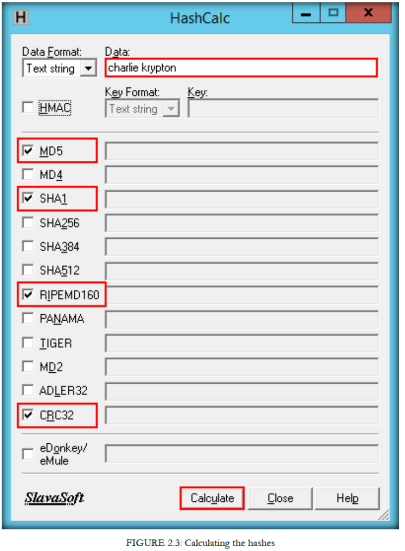


FIGURE 2.3: Calculating the hashes

You can also download HashCalc from http://www.slavasoft.com.

8. The application calculates the hashes and displays them, as shown in the screenshot:

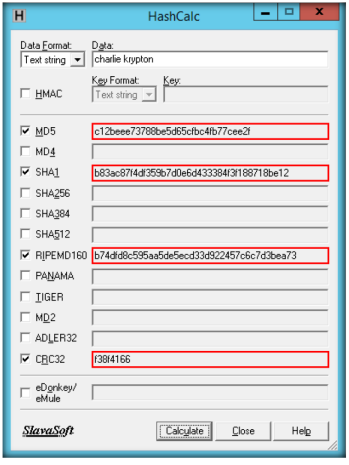


FIGURE 2.4: Hash is generated for chosen hash string

HashCalc is used to generate crypting text.

9. Hash calculation is mainly performed to check data integrity.

Lab Analysis

Document all Hash, MD5, and CRC values for further references.

PLEASE TALK TO YOUR INSTRUCTOR IF YOU HAVE QUESTIONS RELATED TO THIS LAB.

Internet Connection Required	
☐ Yes	☑ No
Platform Supported	
☑ Classroom	☑ iLabs

Lab 3

Calculating MD5 Hashes Using MD5 Calculator

MD5 Calculator is a simple application that calculates the MD5 hash of a given file. It can be used with big files (e.g., multiple gigabytes). It features a progress counter and a text field from which the final MD5 hash can be easily copied to the clipboard.

ICON KEY

- Valuable information
- Test your knowledge
- Web exercise
- Workbook review

Lab Scenario

There has been a need to protect information from "prying eyes." In the electronic age, information that could otherwise benefit or educate a group or individual can also be used against such groups or individuals. Industrial espionage among highly competitive businesses often requires that extensive security measures be put into place. And those who wish to exercise their personal freedom, outside oppressive governments, may also wish to encrypt certain information to avoid suffering the penalties of going against the wishes of those who attempt to control it. Still, the methods of data encryption and decryption are relatively straightforward; algorithms are used to encrypt the data and store system information files safely, away from prying eyes. To be an Expert Ethical Hacker and Penetration Tester, you must understand data encryption using encrypting algorithms.

Lab Objectives

This lab will give you experience on encrypting data and show you how to do it. It will teach you how to:

- Use encrypting/decrypting command
- Calculate the MD5 value of the selected file

Tools demonstrated in this lab are available in D:\CEH-Tools\CEHv9 Module 18 Cryptography

Lab Environment

To complete this lab, you will need:

- MD5 Calculator located at **D:\CEH-Tools\CEHv9 Module 18 Cryptography\MD5 Hash Calculators\MD5 Calculator**
- You can also download the latest version of MD5 Calculator from the link http://www.bullzip.com/products/md5/info.php
- If you decide to download the latest version, then screenshots shown in the lab might differ
- Run this tool in Windows Server 2012
- Administrative privileges to run tools

Lab Duration

Time: 5 Minutes

Overview of MD5 Calculator

MD5 Calculator is a bare-bones program for calculating and comparing MD5 files. While its layout leaves something to be desired, its results are fast and simple.

Lab Tasks

TASK 1

Calculate MD5 Checksum

1. Navigate to **D:\CEH-Tools\CEHv9 Module 18 Cryptography\MD5 Hash Calculators\MD5 Calculator**, double-click **md5calc(1.0.0.0).msi** and follow the installation steps to install MD5 Calculator.
2. To find MD5 Hash of any file, right-click on the specific file (here, **md5calc(1.0.0.0).msi**), and Select **"MD5 Calculator"** from the context menu.

MD5 checksum is used to generate MD5 hash.

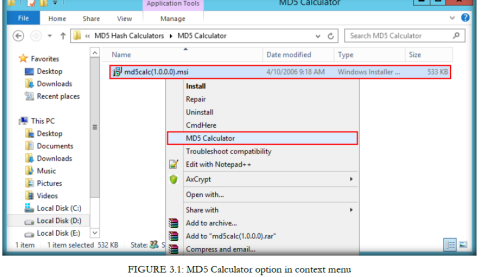


FIGURE 3.1: MD5 Calculator option in context menu

3. **MD5 Calculator** shows the MD5 digest of the selected file.

Note: Alternatively, you can browse any file to calculate the MD5 hash and click on the **Calculate** button to calculate the MD5 hash of the file.

MD5 hash (or checksum) functions as a compact digital fingerprint of a file.

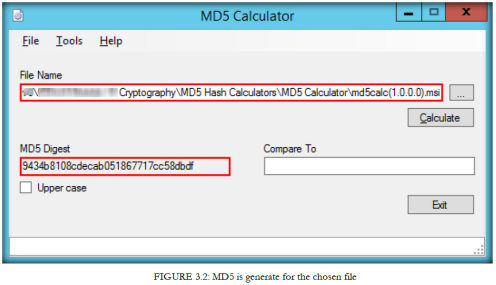


FIGURE 3.2: MD5 is generate for the chosen file

4. MD5 calculator is used to check the integrity of a file.
5. If a person wants to send a file to another person via a medium, he/she will calculate its hashes and sends the file (along with the hash value) to the intended person. When the person on the other side receives the mail, he/she will download the file and calculates its value using MD5 Calculator.
6. Then, the person compares the generated hash value with the hash value that was sent through mail. If both the hash values tally, it is evident that the person obtained the file without any modifications by a third person.

Lab Analysis

Analyze and document the results related to this lab exercise.

PLEASE TALK TO YOUR INSTRUCTOR IF YOU HAVE QUESTIONS RELATED TO THIS LAB.

Internet Connection Required	
☐ Yes	☑ No
Platform Supported	
☑ Classroom	☑ iLabs

Lab 4

Understanding File and Text Encryption Using CryptoForge

CryptoForge is a file encryption software for personal and professional data security. It allows you to protect the privacy of sensitive files, folders, or email messages, by encrypting them with strong encryption algorithms.

ICON KEY

- Valuable information
- Test your knowledge
- Web exercise
- Workbook review

Lab Scenario

CryptoForge allows you to protect the privacy of sensitive files, folders, or email messages, by encrypting them with up to four strong encryption algorithms. Once the information has been encrypted, it can be stored on insecure media or transmitted on an insecure network—like the Internet—and remain secret. Later, the information can be decrypted into its original form.

Lab Objectives

This lab will show you how to encrypt files and text.

Tools demonstrated in this lab are available in D:\CEH-Tools\CEHv9 Module 18 Cryptography

Lab Environment

To complete this lab, you will need:

- CryptoForge located at **D:\CEH-Tools\CEHv9 Module 18 Cryptography\Cryptography Tools\CryptoForge**
- You can also download the latest version of CryptoForge from the link http://www.cryptoforge.com/download
- If you decide to download the latest version, then screenshots shown in the lab might differ
- Follow the wizard driven installation instructions
- Windows Server 2012 running as a host machine
- Windows 8.1 running as a virtual machine
- Administrative privileges to run tools

Lab Duration

Time: 15 Minutes

Overview of the Lab

The lab demonstrates basic encryption methodology used to encrypt files and text messages and share them with the intended person/people.

Lab Tasks

TASK 1

Encrypt a File

1. Navigate to **D:\CEH-Tools\CEHv9 Module 18 Cryptography\Cryptography Tools\CryptoForge**, double-click **CryptoForge.exe** and follow the steps to install the application.
2. Once done with the installation, log in to Windows 8.1 virtual machine, navigate to **Z:\CEHv9 Module 18 Cryptography\Cryptography Tools\CryptoForge**, double-click **CryptoForge.exe** and follow the steps to install the application.

Note: If a **User Account Control** pop-up appears, click **Yes**. If a **Windows Security dialog**-box appears, enter the credentials of Windows Server 2012 machine, and click **OK**.

3. Now, switch to **Windows Server 2012** machine, navigate to **D:\CEH-Tools\CEHv9 Module 18 Cryptography\Cryptography Tools\CryptoForge**, right-click Confidential.txt, and select **Encrypt** from the context menu.

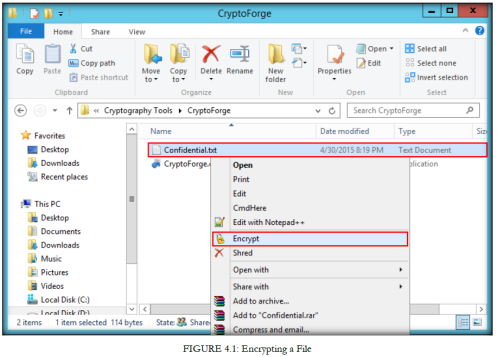


FIGURE 4.1: Encrypting a File

4. The **Enter Passphrase - CryptoForge Files** dialog-box appears; type a password in the **Passphrase** field, retype it in the **Confirm** field, and click **OK**. The password used in this lab is **qwerty@123**.

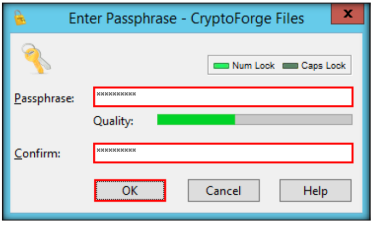


FIGURE 4.2: Enter Passphrase - CryptoForge Files Dialog-Box

5. Now, the file will be encrypted in the same location, and the old file will be deleted automatically, as shown in the screenshot:

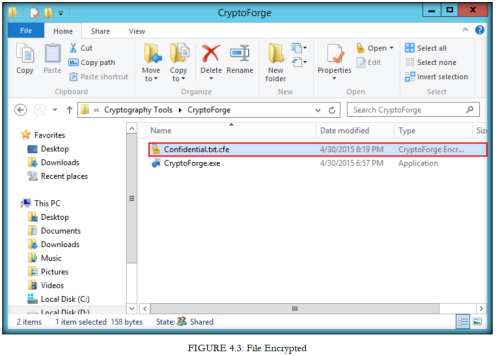


FIGURE 4.3: File Encrypted

TASK 2

Decrypt the Encrypted File

6. No one can access this file unless he/she provides the password for the encrypted file. You will have to share the password with him/her through message, mail, or another means.
7. Let us assume that you shared this file through shared network drive.

8. Now, switch to Windows 8.1 virtual machine, navigate to **Z:\CEHv9 Module 18 Cryptography\Cryptography Tools\CryptoForge**. You will observe the encrypted file in this location.

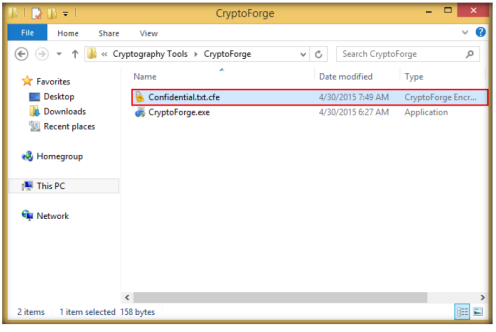


FIGURE 4.4: Viewing the Encrypted File

9. Now, double-click the encrypted file to decrypt it and view its contents.

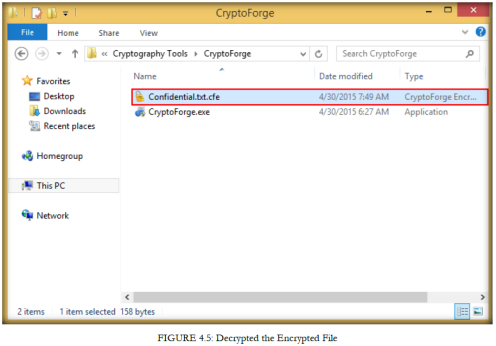


FIGURE 4.5: Decrypted the Encrypted File

10. The **Enter Passphrase - CryptoForge Files** dialog-box appears; enter the password that you have provided to encrypt the file, and click **OK**.

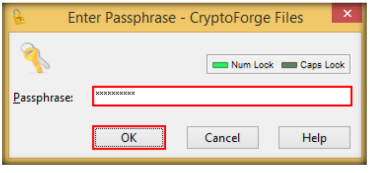


FIGURE 4.6: Enter Passphrase - CryptoForge Files Dialog-Box

11. On entering the password, the file will be successfully decrypted. You may now double-click the file to view its contents.

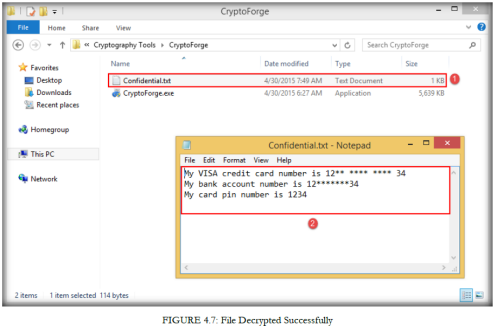


FIGURE 4.7: File Decrypted Successfully

12. So far, we have seen how to encrypt a file and share it with the intended user. Now, let us see how to share an encrypted message with a user.

TASK 3

Encrypt a Message

13. Switch to **Windows Server 2012** machine, go to the **Apps** screen, and click **CryptoForge Text** to launch the application.

14. **CryptoForge Text** window appears, type a message, and click **Encrypt** from the toolbar.

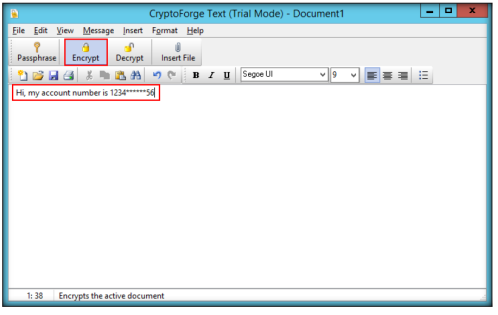


FIGURE 4.8: Encrypting a Text Message

15. The **Enter Passphrase - CryptoForge Text** dialog-box appears; type a password in the **Passphrase** field, retype it in the **Confirm** field, and click **OK**. The password used in this lab is **test@123**.

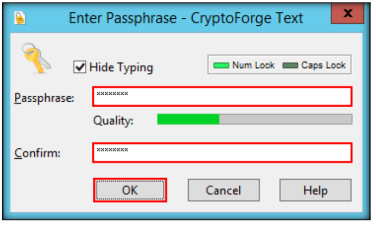


FIGURE 4.9: Enter Passphrase - CryptoForge Text Dialog-Box

16. The message you type will be encrypted, as shown in the screenshot:

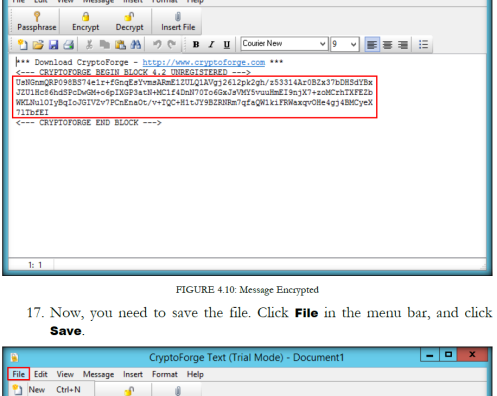


FIGURE 4.10: Message Encrypted

17. Now, you need to save the file. Click **File** in the menu bar, and click **Save**.

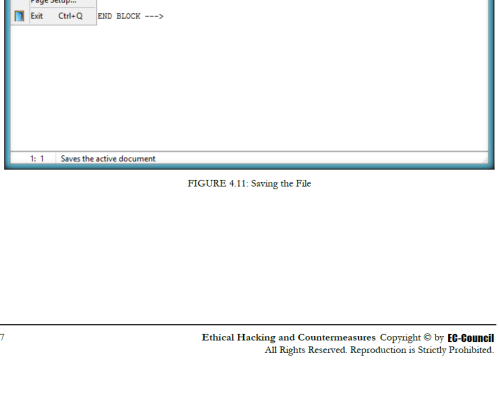


FIGURE 4.11: Saving the File

18. The Save As window appears; navigate to **D:\CEH-Tools\CEHv9 Module 18 Cryptography\Cryptography Tools\CryptoForge**, specify the file name as **Credentials.cfd** and click **Save**.

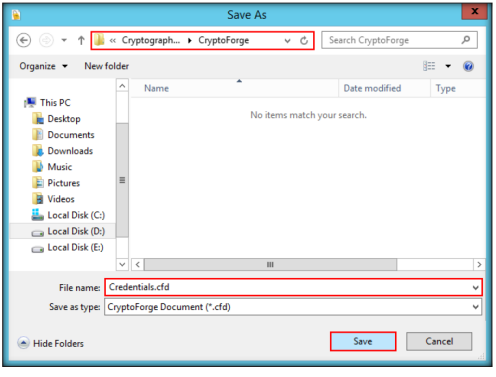


FIGURE 4.12: Saving the File

19. Close the **CryptoForge Text** window.

20. Now, let us assume that you shared the file and through mapped network drive, and shared the password to decrypt the file in an email message or some other means.

TASK 4

Decrypt the Encrypted Message

21. Switch to Windows 8.1 virtual machine, and navigate to navigate to **Z:\CEHv9 Module 18 Cryptography\Cryptography Tools\CryptoForge**. Observe the encrypted file in this location; double-click.

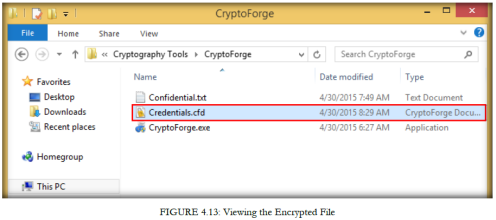


FIGURE 4.13: Viewing the Encrypted File

22. The **CryptoForge Text** window appears, displaying the message in encrypted format. Click **Decrypt** to decrypt it.

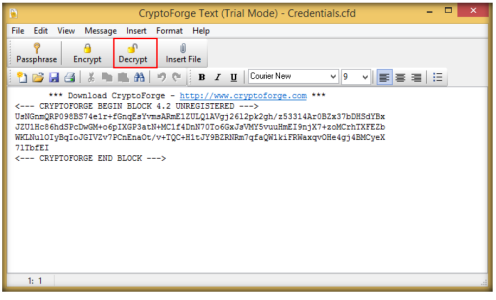


FIGURE 4.14: Decrypting the Encrypted File

23. The **Enter Passphrase - CryptoForge Text** dialog-box appears; enter the password you used to encrypt the message in the **Passphrase** field, and click **OK**.

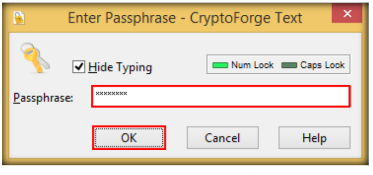


FIGURE 4.15: Enter Passphrase - CryptoForge Text Dialog-Box

24. The **CryptoForge Text** window appears, displaying the message in plain-text format, as shown in the screenshot:

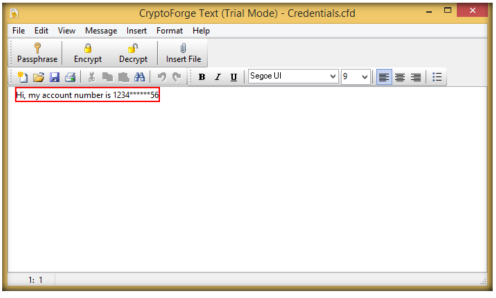


FIGURE 4.16: Message Decrypted Successfully

25. Thus, you have used CryptoForge tool to encrypt and share files and messages with the intended person.
26. In real time, you may share sensitive information through email by encrypting data using CryptoForge.

Lab Analysis

Analyze and document the results related to this lab exercise. Provide your opinion of your target's security posture and exposure through public and free information.

PLEASE TALK TO YOUR INSTRUCTOR IF YOU HAVE QUESTIONS RELATED TO THIS LAB.

Internet Connection Required	
☐ Yes	☑ No
Platform Supported	
☑ Classroom	☑ iLabs

Lab 5

Basic Data Encryption Using Advanced Encryption Package

Advanced Encryption Package is most noteworthy for its flexibility: not only can you encrypt files for your own protection, but you can easily create "self-decrypting" versions of your files that others can run without needing this or any other software.

ICON KEY

- Valuable information
- Test your knowledge
- Web exercise
- Workbook review

Lab Scenario

Data encryption and decryption operations require major security applications to secure data. Most systems uses block ciphers, such as public AES standard. However, implementations of block ciphers such as AES, as well as other cryptographic algorithms, are subject to side-channel attacks. These attacks allow adversaries to extract secret keys from devices by passively monitoring the power consumption of other side channels. Countermeasures are required for applications to which side-channel attacks are a threat. These include several military and aerospace applications in which program information, classified data, algorithms, and secret keys reside on assets that may not always be physically protected. To be an Expert Ethical Hacker and Penetration Tester, you must understand file data encryption.

Lab Objectives

This lab will give you experience regarding data encryption and show you the techniques to do it. It will teach you how to:

- Use encrypting/decrypting command
- Calculate the encrypted value of the selected file

Tools demonstrated in this lab are available in D:\CEH-Tools\CEHv9 Module 18 Cryptography

Lab Environment

To complete this lab, you will need:

- Advanced Encryption Package located at **D:\CEH-Tools\CEHv9 Module 18 Cryptography\Cryptography Tools\Advanced Encryption Package**
- You can also download the latest version of Advanced Encryption Package from the link http://www.secureaction.com/encryption_pro/
- If you decide to download the latest version, then screenshots shown in the lab might differ
- Administrative privileges to run tools
- Run this tool in Windows Server 2012

Lab Duration

Time: 10 Minutes

Overview of Advanced Encryption Package

Advanced Encryption Package includes a file shredder that wipes out the contents of your original files. It also integrates nicely with Windows Explorer, allowing you to use Explorer's context menus and avoid having another window clutter your screen.

Lab Tasks

TASK 1

Encrypting a File

1. Navigate to **D:\CEH-Tools\CEHv9 Module 18 Cryptography\Cryptography Tools\Advanced Encryption Package**, double-click **aep.msi** and follow the steps to install the application.
2. On completing the installation, launch **Advanced Encryption Package** application from the **Apps** screen.

Advanced Encryption Package is a symmetric-key encryption comprising three block ciphers, AES-128, AES-192 and AES-256.

FIGURE 5.1: Launching Advanced Encryption Package application from the Apps screen

3. The **Advanced Encryption Package 2014 - License Manager** window appears displaying the **License Manager** section. Select **Start free 30-day trial** radio button, and click **Next**.

You can also download Advance Encryption Package from http://www.secureaction.com

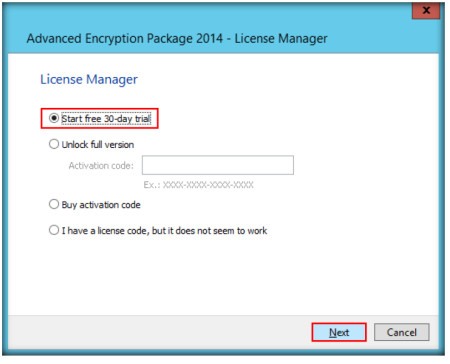


FIGURE 5.2: License Manager window

4. The **Activating** step appears; click **Next**.

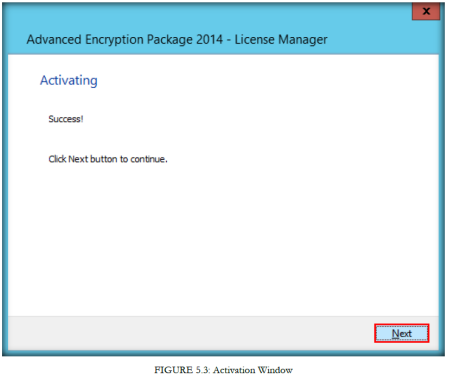


FIGURE 5.3: Activation Window

5. Leave all the options set to default in **License Information** step, and click **Finish**.

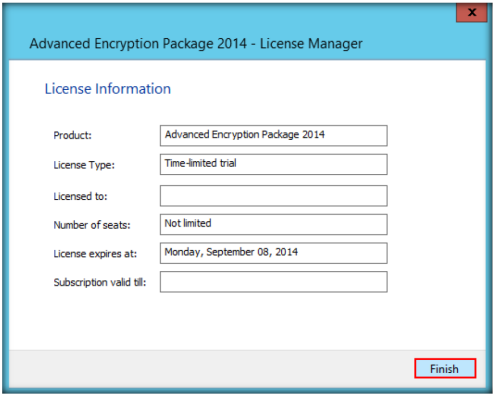


FIGURE 5.4: License Information section

6. The main window of **Advanced Encryption Package** appears.

Advance Encryption Package is easy to use for novices.

7. A sample file named **Sample.docx** is provided at **D:\CEH-Tools\CEHv9 Module 18 Cryptography\Cryptography Tools\Advanced Encryption Package**. Select the sample file, and click **Encrypt** in the toolbar.

Tools demonstrated in this lab are available in D:\CEH-Tools\CEHv9 Module 18 Cryptography

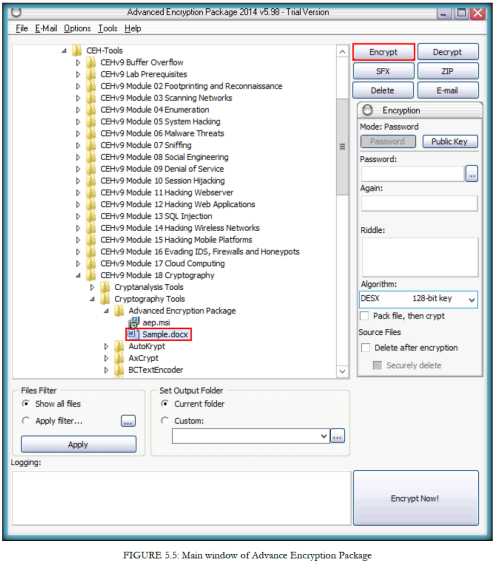


FIGURE 5.5: Main window of Advance Encryption Package

8. You need to provide a password for encryption. Enter the password in **Password** field, retype it in the **Again** field, and click **Encrypt Now!**.
9. In this lab, the password is **test@123**.

It creates encrypted self-extracting files to send as email attachments.

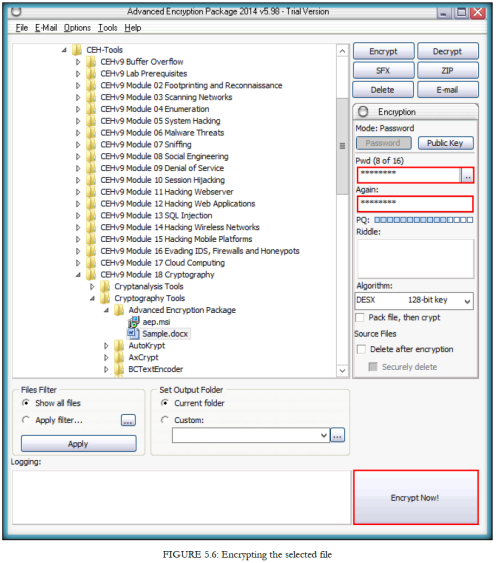


FIGURE 5.6: Encrypting the selected file

10. The encrypted Sample File appears in the same location as the original file (i.e., **D:\CEH-Tools\CEHv9 Module 18 Cryptography\Cryptography Tools\Advanced Encryption Package**).

11. To **decrypt** the file, first select the encrypted file, and click on **Decrypt**.

Note: Navigate to **D:\CEH-Tools\CEHv9 Module 18 Cryptography\Cryptography Tools\Advanced Encryption Package** and delete the unencrypted source file, as conflicts might occur while decrypting the encrypted file in the same location.

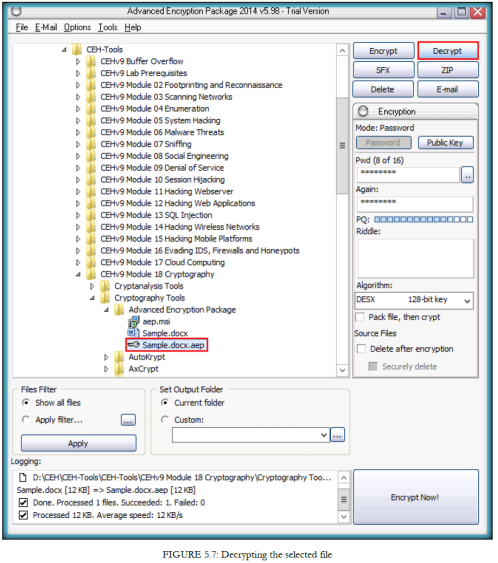


FIGURE 5.7: Decrypting the selected file

12. It will prompt you to enter the password.
13. Because the unencrypted source file is already present in the same location, click **Leave it alone,** under **Source file(s),** and click **Decrypt Now!**

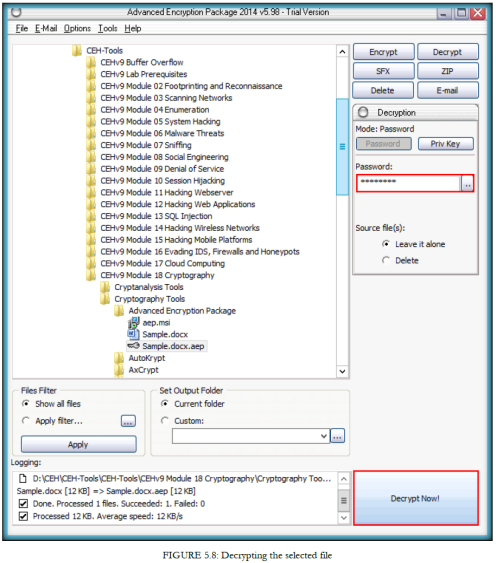


FIGURE 5.8: Decrypting the selected file

14. The decrypted file appears in the same location shown in the screenshot:

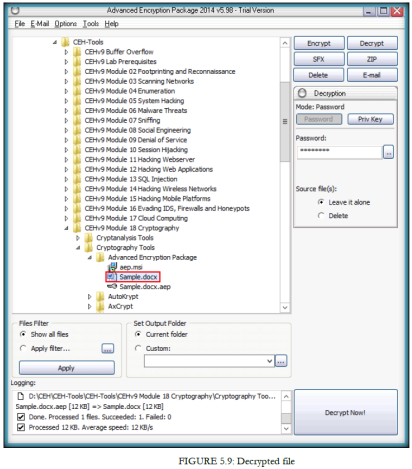


FIGURE 5.9: Decrypted file

15. In real time, network administrators or ethical hackers use this tool to encrypt files and send it to the intended persons to safeguard the integrity of the files.

Lab Analysis

Analyze and document the results related to this lab exercise.

PLEASE TALK TO YOUR INSTRUCTOR IF YOU HAVE QUESTIONS RELATED TO THIS LAB.

Internet Connection Required	
☑ Yes	☐ No
Platform Supported	
☑ Classroom	☐ iLabs

Lab 6

Encrypting and Decrypting the Data Using BCTextEncoder

BCTextEncoder simplifies encoding and decoding text data. Plain text data are compressed, encrypted and converted to text format, which can then be easily copied to the clipboard or saved as a text file.

ICON KEY

- Valuable information
- Test your knowledge
- Web exercise
- Workbook review

Lab Scenario

To be an expert ethical hacker and penetration tester, you must have knowledge of cryptography functions.

Lab Objectives

This lab will give you experience on encrypting data and show you how to do it. It will teach you how to:

- Use Encode/decode text data encrypted with a password

Lab Environment

To complete this lab, you will need:

- BCTextEncoder located at **D:\CEH-Tools\CEHv9 Module 18 Cryptography\Cryptography Tools\BCTextEncoder**
- You can also download the latest version of BCTextEncoder from the link https://www.jetico.com/products/free-security-Tools/bctextencoder
- If you decide to download the latest version, then screenshots shown in the lab might differ
- Run this tool on Windows Server 2012 host machine
- Administrative Privileges to run the tool

Tools demonstrated in this lab are available in D:\CEH-Tools\CEHv9 Module 18 Cryptography

Lab Duration

Time: 10 Minutes

Overview of BCTextEncoder

BCTextEncoder uses public key encryption methods, as well as password-based encryption. This utility software uses strong and approved symmetric and public key algorithms for data encryption.

Lab Tasks

TASK 1

Encrypt the Data

1. Navigate to **D:\CEH-Tools\CEHv9 Module 18 Cryptography\Cryptography Tools\BCTextEncoder** and double-click **BCTextEncoder.exe**.
2. The main window of BCTextEncoder appears as show in the following screenshot:

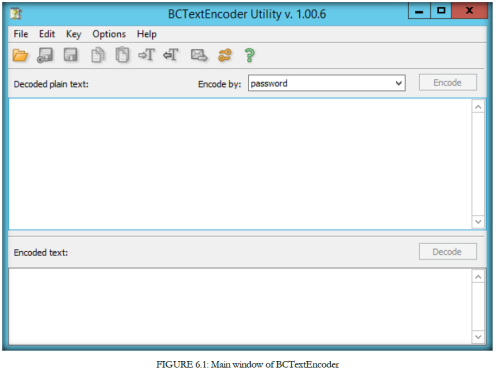


FIGURE 6.1: Main window of BCTextEncoder

BCTextEncoder utilizes the following encryption algorithms:

- ZLIB compression algorithm
- AES (Rijndael) encryption algorithm for password based encryption
- RSA asymmetric encryption algorithm for public key encryption

3. To encrypt the text, type the text in the **clipboard.** Or, select the secret data, and paste it to the clipboard by pressing **Ctrl+V** and clicking **Encode**.

BCTextEncoder is intended for fast encoding and decoding text data

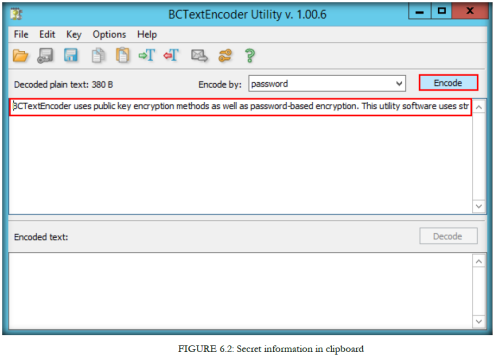


FIGURE 6.2: Secret information in clipboard

4. The **Enter Password** dialog-box appears; set the **password** (**qwerty@123**), and **confirm** it in the respective field.
5. Click **OK**.

The main advantage of BCTextEncoder is support of public-key encryption.

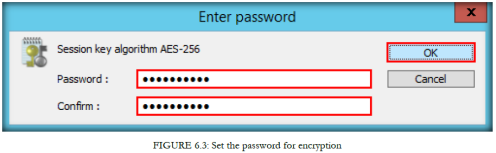


FIGURE 6.3: Set the password for encryption

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

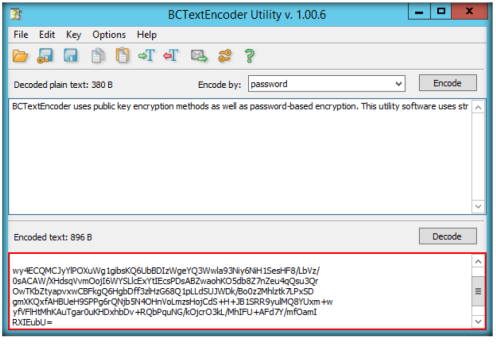


FIGURE 6.4: Encoded text

TASK 2

Decrypt the Data

7. To **decrypt** the data, first you need to clean the **Decoded plain text** in the clipboard.
8. Click **Decode**.

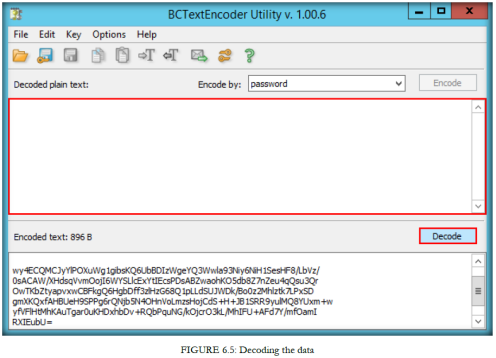


FIGURE 6.5: Decoding the data

BCArchive includes the BC Key Manager utility to manage your own public/secret key pair as well as public keys you have received from other people

9. **Enter password for encoding text** dialog-box appears; enter the **password** (**qwerty@123**) in password field, and click **OK.**

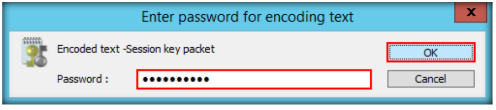


FIGURE 6.6: Enter the password for decoding

10. Decoded plain text appears, as shown in the screenshot:

BCTextEncoder not only encrypts, but also compresses the data

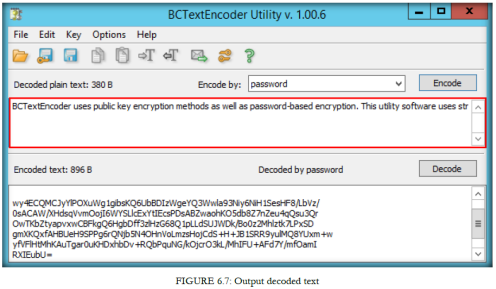


FIGURE 6.7: Output decoded text

11. This way, you need to encode the text while sending it to the intended user along with the password used for encryption. The user for whom the text is intended should have the BCTextEncoder application installed on his/her machine.
12. He/she will have to paste the encoded text in the Encoded text section and use the password you shared, to decode it to plain text.

Lab Analysis

Analyze and document the results related to this lab exercise.

PLEASE TALK TO YOUR INSTRUCTOR IF YOU HAVE QUESTIONS RELATED TO THIS LAB.

Internet Connection Required	
☐ Yes	☑ No
Platform Supported	
☑ Classroom	☑ iLabs

Lab 7

Exploiting OpenSSL Heartbleed Vulnerability on a HTTPS Website

ICON KEY

- Valuable information
- Test your knowledge
- Web exercise
- Workbook review

The Heartbleed Bug is vulnerability in the popular OpenSSL cryptographic software library. This weakness allows stealing the information protected, under normal conditions, by the SSL/TLS encryption used to secure the Internet.

Lab Scenario

HTTPS websites provide encryption to the data flow. Though hackers attempt to intercept the data, it is encrypted and cannot be easily decrypted/decoded. However, due to vulnerabilities found in the recent versions of OpenSSL, attackers can easily intercept the data flow and obtain it in plain-text. 1.0.2-beta and lower versions of OpenSSL are vulnerable to Heartbleed exploit and return sensitive information in clear text.

As an expert Security Professional and Penetration Tester you should be familiar with these exploits and take certain security measures to avoid websites in your organization from being exploited.

Tools demonstrated in this lab are available D:\CEH-Tools\CEHv9 Module 18 Cryptography\Heartbleed

Lab Objectives

The objective of this lab is to help students learn how to penetrate into a Heartbleed vulnerable website.

In this lab, you will learn to:

- Test Heartbleed vulnerability in a https website
- Exploit Heartbleed vulnerability and obtain passwords and certificate information in plain-text

Lab Environment

To complete this lab, you will need:

- **ownCloud**, **Microsoft Visual C++ 2010** and **WAMP Server** located at **D:\CEH-Tools\CEHv9 Module 18 Cryptography\Heartbleed**

- You can download the latest version of WAMP Server from http://www.Wampserver.com/en/ and Microsoft Visual C++ 2010 from http://www.microsoft.com/en-in/download/details.aspx?id=5555
- If you decide to download the latest version, screenshots and steps might differ in your lab environment.
- Run this lab in Window Server 2008 and Kali Linux virtual machines
- Administrative privileges to run the tool
- A web browser with Internet access in both the machines

Lab Duration

Time: 15 Minutes

Overview of Heartbleed

SSL/TLS provides communication security and privacy over the Internet for applications such as web, email, instant messaging (IM), and some virtual private networks (VPNs). The Heartbleed bug allows anyone on the Internet to read the memory of the systems protected by the vulnerable versions of the OpenSSL software. This compromises the secret keys used to identify the service providers and to encrypt the traffic, the names and passwords of the users, and the actual content. This allows attackers to eavesdrop on communications, steal data directly from the services and users, and impersonate services and users.

Lab Tasks

Note: Before running this lab, log into **Windows Server 2008,** and ensure that you stop IIS admin service and World Wide Web Publishing Service (if you have the service installed on the machine.). To stop the service, go to **Start → Administrative Tools → Services**, right-click **IIS Admin Service** and click **Stop**, right-click **World Wide Web Publishing Service** and click **Stop**. Also ensure that you stop Internet Information Services (IIS) Manager and Internet Information Services (IIS) 6.0 Manager. To stop Internet Information Services (IIS) Manager, go to **Start → Administrative Tools → Internet Information Services (IIS) Manager**, right-click on the server name in the left pane and click **Stop** to stop the manager. To stop Internet Information Services (IIS) 6.0 Manager, go to **Start → Administrative Tools → Internet Information Services (IIS) 6.0 Manager**, right-click on the server name in the left pane, and click **Disconnect** to disconnect the manager.

In this lab, we are featuring OpenSSL 1.0.1c (vulnerable to Heartbleed) for demonstration purpose.

Note: Make sure that you delete all the cookies in the browser in which you will be hosting **ownCloud,** and make sure that WAMPServer is kept online throughout this lab.

1. Before beginning the lab, click Start at the lower-left corner of the screen, and then click start WampServer to launch WampServer.

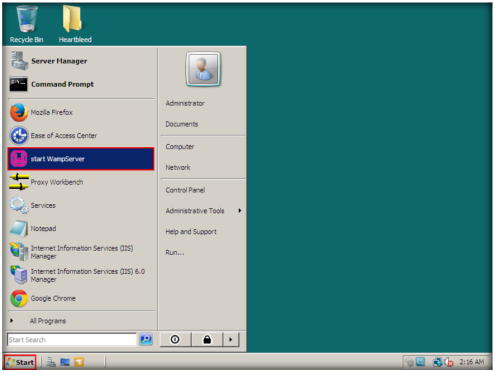


FIGURE 7.1: Starting WAMPServer

TASK 1

Test for Heartbleed vulnerability

2. Log into the **Kali - Linux** virtual machine, and open a command-line terminal.
3. To check whether the website is vulnerable to Heartbleed, launch a command prompt, issue the command **nmap -p 443 --script ssl-heartbleed 10.0.0.3** and press **Enter**.

Note: In this lab, **10.0.0.3** is the IP address of **Windows Server 2008** which is hosting the ownCloud server. This IP address may differ in your lab environment.

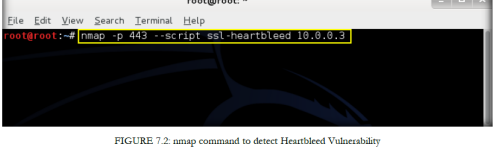


FIGURE 7.2: nmap command to detect Heartbleed Vulnerability

4. This initiates nmap and the tool begins to scan the server for Heartbleed vulnerability.
5. Nmap returns result stating that the server is Heartbleed vulnerable, as shown in the screenshot:

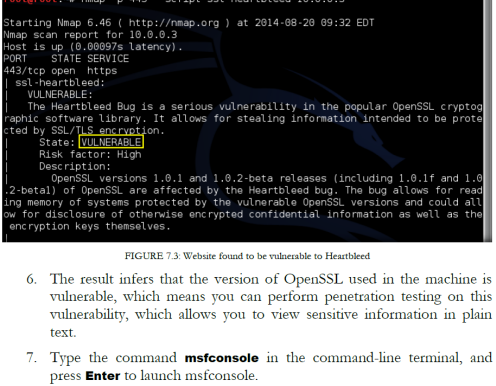


FIGURE 7.3: Website found to be vulnerable to Heartbleed

6. The result infers that the version of OpenSSL used in the machine is vulnerable, which means you can perform penetration testing on this vulnerability, which allows you to view sensitive information in plain text.
7. Type the command **msfconsole** in the command-line terminal, and press **Enter** to launch msfconsole.

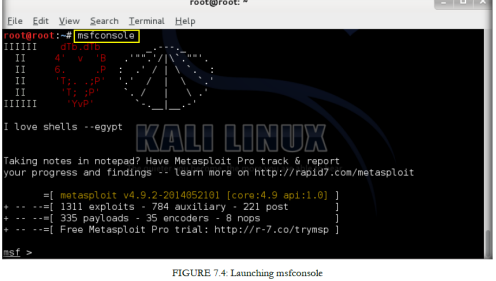


FIGURE 7.4: Launching msfconsole

8. Type **use auxiliary/scanner/ssl/openssl_heartbleed** and press **Enter**. This launches the **openssl_heartbleed** auxiliary module, as shown in the screenshot:

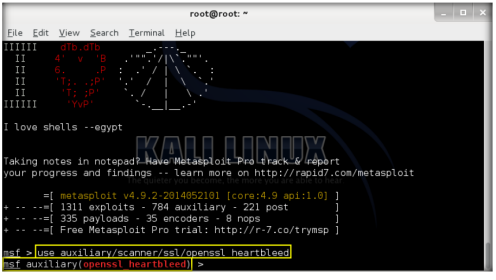


FIGURE 7.5: Using heartbleed exploit in msfconsole

TASK 2

Exploit the vulnerability

9. Issue the following commands:
   a. **set RHOSTS 10.0.0.3**
   b. **set RPORT 443**
   c. **set VERBOSE true**

Note: **10.0.0.3** is the IP address of Windows Server 2008 virtual machine on which ownCloud is configured.

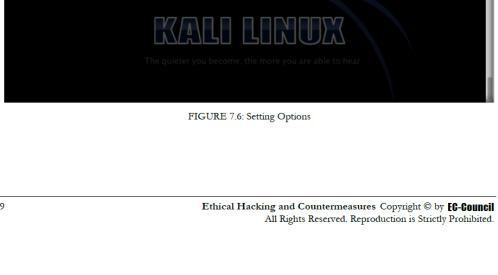


FIGURE 7.6: Setting Options

10. Now, assume that you are a user who wants to login to the ownCloud application through the **Windows Server 2008** machine. Browse the ownCloud login page. Enter the username **shane** and password **florida@123** and click **Log in**.

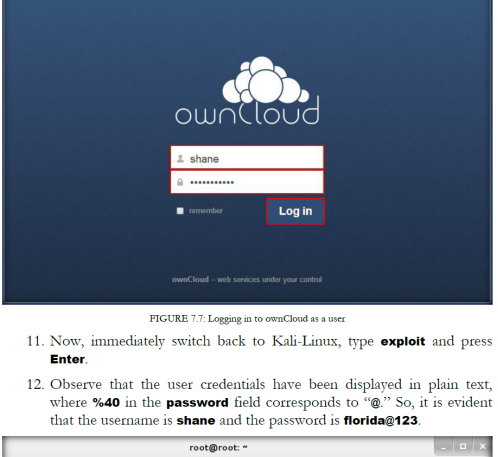


FIGURE 7.7: Logging in to ownCloud as a user

11. Now, immediately switch back to Kali-Linux, type **exploit** and press **Enter**.

12. Observe that the user credentials have been displayed in plain text, where **%40** in the **password** field corresponds to "**@**." So, it is evident that the username is **shane** and the password is **florida@123**.

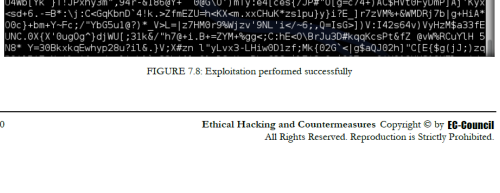


FIGURE 7.8: Exploitation performed successfully

13. Along with this, extra information such as data incorporated in the certificate request is displayed, which is shown in the screenshot:

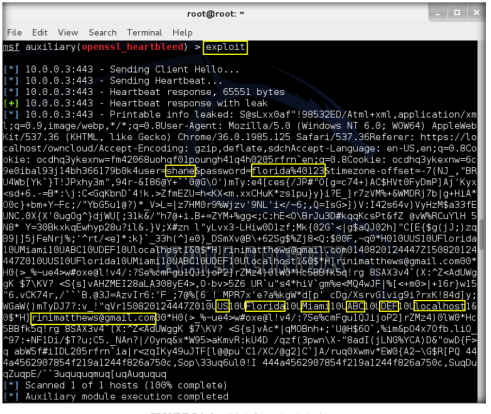


FIGURE 7.9: Sensible information leaked out

14. In real time, an attacker issues the "**exploit**" command continuously, to obtain more and more information.

Note: This exploit works only for OpenSSL versions **1.0.1** to **1.0.2-beta**.

Lab Analysis

Analyze and document the results related to this lab exercise. Provide your opinion of your target's security posture and exposure through public and free information.

PLEASE TALK TO YOUR INSTRUCTOR IF YOU HAVE QUESTIONS RELATED TO THIS LAB.

Internet Connection Required	
☐ Yes	☑ No
Platform Supported	
☑ Classroom	☑ iLabs

Lab 8

Creating and Using Self-Signed Certificate

SSL is an essential part of securing your IIS 7.0 site. Creating a self-signed certificate in IIS 7 is much easier to do than in previous versions. SSL certificates enable the encryption of all traffic sent to and from your IIS web site, preventing others from viewing sensitive information. It uses public-key cryptography to establish a secure connection. This means that anything encrypted with a public key (the SSL certificate) can only be decrypted with a private key and vice-versa.

ICON KEY

- Valuable information
- Test your knowledge
- Web exercise
- Workbook review

Lab Scenario

A self-signed certificate is an identity certificate signed by the same entity whose identity it certifies. In general, self-signed certificates are widely used for testing servers.

Lab Objectives

This lab will give you experience on how to create self-signed certificates.

Lab Environment

To complete this lab, you will need:

- Windows Server 2012
- Administrative privileges required to perform this lab

Tools demonstrated in this lab are available in D:\CEH-Tools\CEHv9 Module 18 Cryptography

Lab Duration

Time: 10 Minutes

Overview of Lab

In cryptography and computer security, a self-signed certificate is an identity certificate signed by the same entity whose identity it certifies. However, the term has nothing to do with the identity of the person or organization that actually performed the signing procedure.

TASK 1

Verifying Self-Signed Certificate

Lab Tasks

1. Before we start the lab, first we will check with our local sites whether they include a self-signed certificate.
2. Launch a web browser, type https://www.goodshopping.com in the address bar, and press **Enter**. In this lab, we are using Google Chrome.

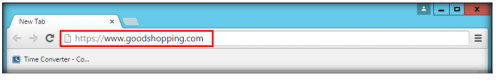


FIGURE 8.1: www.goodshopping.com before adding Certificate

3. As we are using an https channel to browse, it displays a page stating that the connection is not private. Click **Advanced** to proceed.

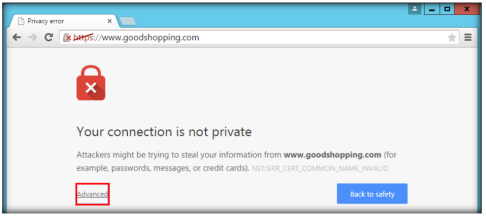


FIGURE 8.2: Connection is not Private

Self-signed certificate is an identity certificate that is signed by the same entity whose identity it certifies. This term has nothing to do with the identity of the person or organization that actually performed the signing procedure.

4. Click **Proceed to** the www.goodshopping.com (unsafe) link.

In technical terms a self-signed certificate is one signed with its own private key.

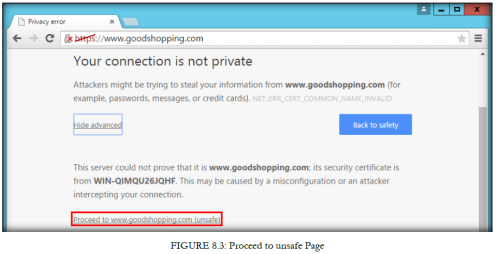


FIGURE 8.3: Proceed to unsafe Page

5. As the site does not have a self-signed certificate, it displays a Not Found page, as shown in the screenshot.

FIGURE 8.4: HTTP Error 404 Page

TASK 2

Launch IIS Manager

6. Launch **Start** menu by hovering the mouse cursor over the lower-left corner of the desktop.
7. Click on the down arrow to view the Apps screen, and click **Internet Information Services**.

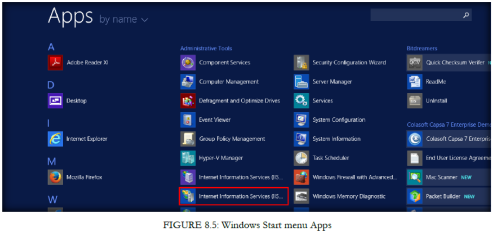


FIGURE 8.5: Windows Start menu Apps

8. If the **Do you want to get started with Microsoft Web Platform ...** pop-up appears, click **Cancel**.

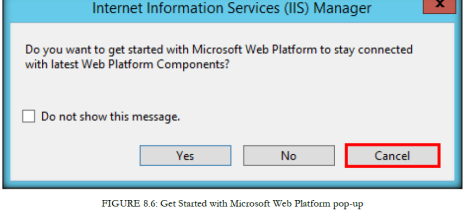


FIGURE 8.6: Get Started with Microsoft Web Platform pop-up

TASK 3

Configure Server Certificates

9. The Internet Information Services (IIS) Manager window appears; click the **Machine** name in the Connections pane, and double-click Server Certificates.

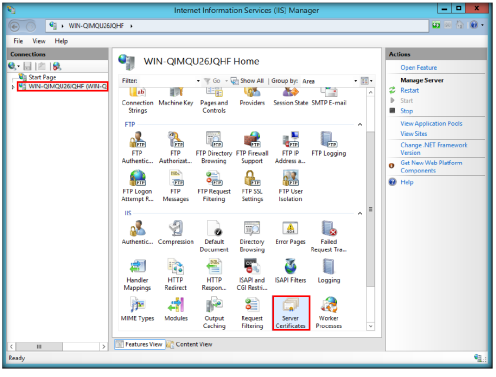


FIGURE 8.7: IIS Manager Server Certificates

In typical public key infrastructure (PKI) arrangements, a digital signature from a certificate authority (CA) attests that a particular public key certificate is valid (i.e., contains correct information). Users or their software on their behalf, check that the private key used to sign some certificate matches the public key in the CA's certificate.

TASK 4

Create Self-Signed Certificate

10. In the **Server Certificates** wizard, click **Create Self-Signed Certificate** in the **Actions** pane.

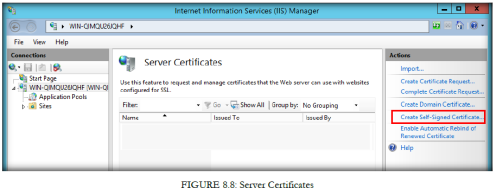


FIGURE 8.8: Server Certificates

Since CA certificates are often signed by other, "higher-ranking," CAs, there must necessarily be a highest CA, which provides the ultimate in attestation authority in that particular PKI scheme.

11. The Create Self-Signed Certificate wizard appears; type a name in the **Specify a friendly name for the certificate** field.

12. Choose **Personal** in the **Select a certificate store for the new certificate** field drop-down list, and click **OK**.

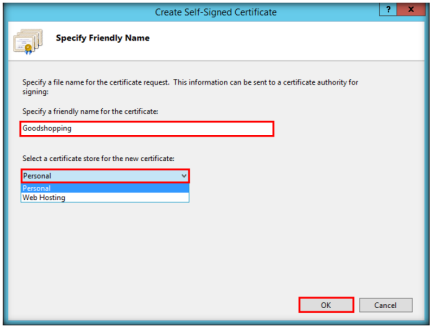


FIGURE 8.9: Specify Friendly Name

Obviously, the highest-ranking CA's certificate can't be attested by some other higher CA (there being none), and so that certificate can only be "self-signed." Such certificates are also termed root certificates.

13. The New Self-Signed Certificate will display in the Server Certificates pane, as shown in screenshot.

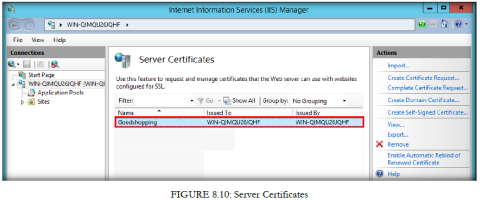


FIGURE 8.10: Server Certificates

TASK 5

Edit Bindings

14. Expand the **Sites** node, and select **Goodshopping** in the **Connections** pane, and click **Bindings** in the **Actions** pane.

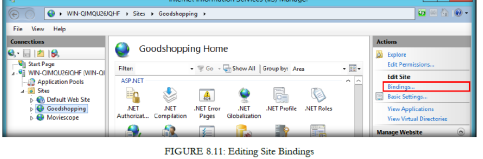


FIGURE 8.11: Editing Site Bindings

15. The Site Bindings wizard appears; click **Add**.

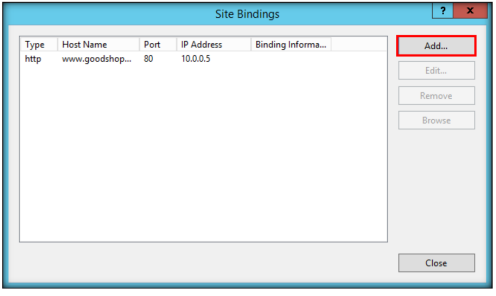


FIGURE 8.12: Site Bindings Wizard

16. The Add Site Binding window appears; choose **https** from the **Type:** field drop-down list.

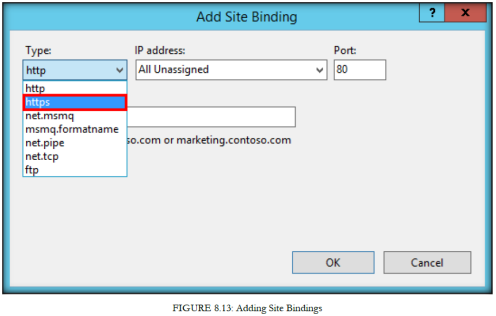


FIGURE 8.13: Adding Site Bindings

17. Once you choose the https channel in the Port field, it will automatically changes to **443** (the channel on which HTTPS runs).
18. Choose the IP address in which the site is hosted, or leave the default setting.

In a web of trust certificate scheme there is no central CA, and so identity certificates for each user can be self-signed. In this case, however, it has additional signatures from other users which are evaluated to determine whether a certificate should be accepted as correct.

A certificate serves two essential purposes: distributing the public key and verifying the identity of the server so visitors know they aren't sending their information to the wrong person.

19. Specify the **Host name** www.example.com. In this lab, we are applying certificate for the **Goodshopping** site.

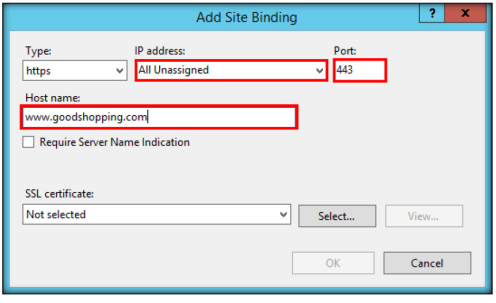


FIGURE 8.14: Adding Site Bindings-Host Name

It can only properly verify the identity of the server when it is signed by a trusted third party because any attacker can create a self-signed certificate and launch a man-in-the-middle attack.

20. In the **SSL certificate** field, choose Goodshopping from the drop-down list, and click **OK**.

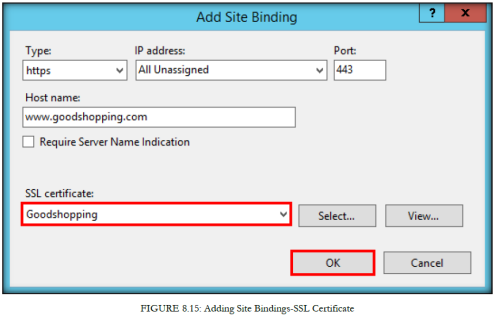


FIGURE 8.15: Adding Site Bindings-SSL Certificate

If a user just accepts a self-signed certificate, an attacker could eavesdrop on all the traffic or try to set up an imitation server to phish additional information out of the user.

An Intranet. When clients only have to go through a local Intranet to get to the server, there is virtually no chance of a man-in-the-middle attack.

21. In the Site Bindings wizard, the newly created SSL certificate is added, as shown in the screenshot. Click **Close**.

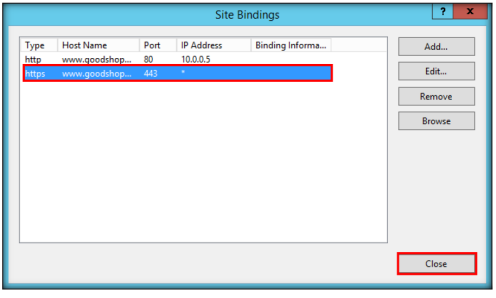
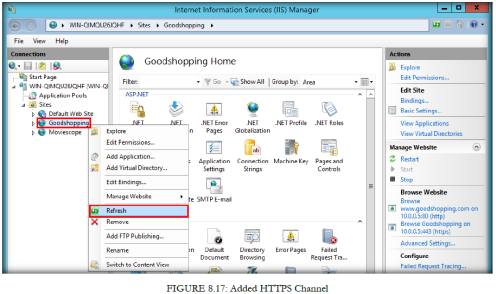


FIGURE 8.16: Added HTTPS Channel

22. Now, right-click the name of the site for which you have created the self-signed certificate, and click **Refresh** from the context menu.



A development server. There is no need to spend extra cash buying a trusted certificate when you are just developing or testing an application.

FIGURE 8.17: Added HTTPS Channel

23. Open a browser, type https://www.goodshopping.com in the address bar, and press **Enter**.

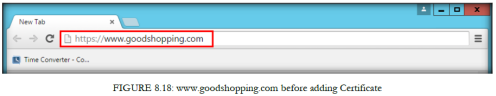


FIGURE 8.18: www.goodshopping.com before adding Certificate

Personal sites with few visitors. If you have a small personal site that transfers non-critical information, there is very little incentive for someone to attack the connections.

24. As we are using an https channel to browse, it displays a page stating that the connection is not private; click **Advanced** to proceed.

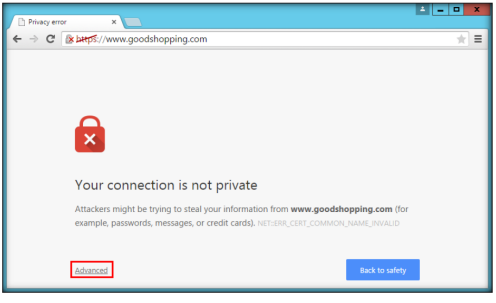


FIGURE 8.19: Connection is not Private

25. Click **Proceed to www.goodshopping.com (unsafe)**.

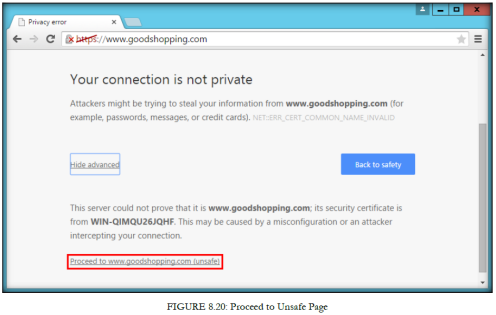


FIGURE 8.20: Proceed to Unsafe Page

Creating a self-signed certificate in IIS 7 is much easier to do than in previous versions of IIS. IIS now provides a simple interface for generating a self-signed certificate. One drawback is that the common name of the certificate is always the servers name instead of the site name.

26. Now you can display the Goodshopping webpage, as shown in the screenshot.

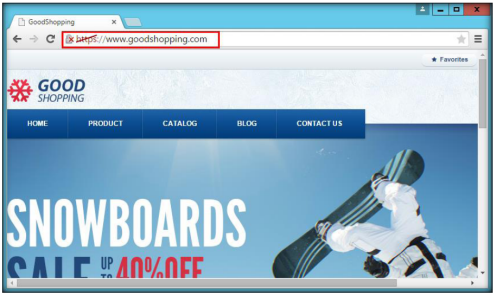


FIGURE 8.21: Self-Signed Certificate Page

If SSL utilizes public key cryptography to encrypt the data stream traveling over the Internet, why is a certificate necessary? The technical answer to that question is that a certificate is not really necessary - the data is secure and cannot easily be decrypted by a third party.

Lab Analysis

Analyze and document the results related to this lab exercise.

PLEASE TALK TO YOUR INSTRUCTOR IF YOU HAVE QUESTIONS RELATED TO THIS LAB.

Internet Connection Required	
☐ Yes	☑ No
Platform Supported	
☑ Classroom	☑ iLabs

Lab 9

Basic Disk Encryption Using VeraCrypt

VeraCrypt adds enhanced security to the algorithms used for system and partitions encryption, making it immune to new developments in brute-force attacks.

ICON KEY

- Valuable information
- Test your knowledge
- Web exercise
- Workbook review

Lab Scenario

Disk encryption encrypts all data on a system, including files, folders, and the operating system. This is most appropriate when the physical security of the system is not assured. Examples include traveling laptops or desktops that are not in a physically secured area. When properly implemented, encryption provides an enhanced level of assurance that the data, while encrypted, cannot be viewed or otherwise discovered by unauthorized parties in the event of theft, loss, or interception.

Lab Objectives

This lab will give you experience in encrypting data and show you how to do so. It will teach you how to:

- Create a virtual encrypted disk with a file

Tools demonstrated in this lab are available in D:\CEH-Tools\CEHv9 Module 18 Cryptography

Lab Environment

To complete this lab, you will need:

- VeraCrypt located at **D:\CEH-Tools\CEHv9 Module 18 Cryptography\Disk Encryption Tools\VeraCrypt**
- You can also download the latest version of VeraCrypt from the link https://veracrypt.codeplex.com/
- If you decide to download the latest version, then screenshots shown in the lab might differ
- Run this tool in Windows Server 2012
- Follow the wizard driven installation instructions

- Administrative privileges to run tools

Lab Duration

Time: 10 Minutes

Overview of VeraCrypt

VeraCrypt is a software application used for on-the-fly encryption (OTFE). It is distributed without cost, and the source code is available. It can create a virtual encrypted disk within a file, or encrypt a partition or entire storage device.

TASK 1

Create a Volume

Lab Tasks

1. Open the **Start** menu by hovering the mouse cursor to the lower-left corner of the desktop.

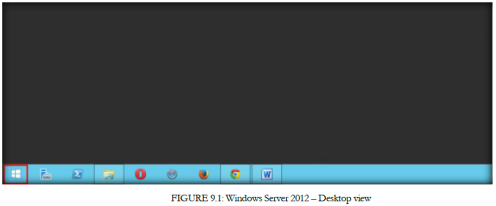


FIGURE 9.1: Windows Server 2012 – Desktop view

2. Click on the **down arrow** to view installed apps.

FIGURE 9.2: Windows Server 2012 – More Apps

3. Click **VeraCrypt** to launch the application.

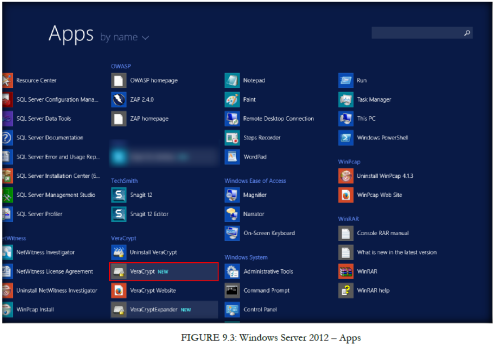


FIGURE 9.3: Windows Server 2012 – Apps

VeraCrypt is a software application used for on-the-fly encryption (OTFE). It is distributed without cost and the source code is available.

4. The VeraCrypt **main window** appears; click **Create Volume**.

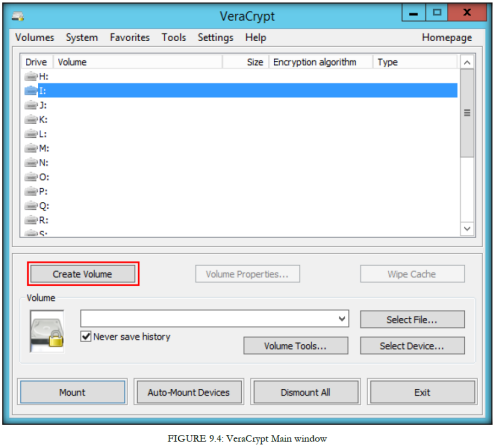


FIGURE 9.4: VeraCrypt Main window

VeraCrypt have the ability to create and run a hidden encrypted operating system whose existence may be denied.

5. The **VeraCrypt Volume Creation Wizard** window appears.
6. Select **Create an encrypted file container** to create a file containing a virtual, encrypted disk.

IMPORTANT: Note that VeraCrypt will not encrypt any existing files (when creating a VeraCrypt file container). If you select an existing file in this step, it will be overwritten and replaced by the newly created volume (so the overwritten file will be lost, not encrypted). You will be able to encrypt existing files (later on) by moving them to the VeraCrypt volume that we are creating now.

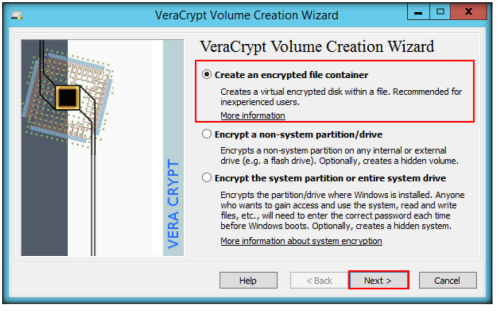


FIGURE 9.5: VeraCrypt Volume Creation Wizard

7. In the **Volume Type** wizard, select **Standard VeraCrypt volume**. This creates a **normal** VeraCrypt volume.
8. Click **Next** to proceed.

Note: After you copy existing unencrypted files to a VeraCrypt volume, you should securely erase (wipe) the original unencrypted files. There are software tools that can be used for the purpose of secure erasure (many of them are free).

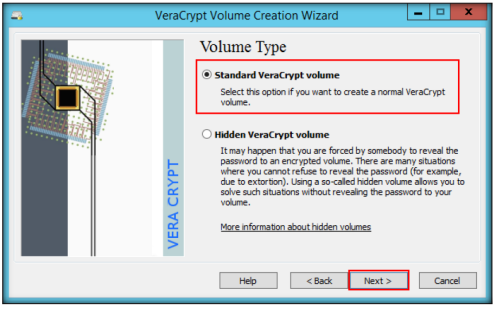


FIGURE 9.6: VeraCrypt Volume Creation Wizard-Volume Type

9. In the **Volume Location** wizard, click **Select File...**.

FIGURE 9.7: VeraCrypt Volume Creation Wizard-Volume Location

VeraCrypt supports a concept called plausible deniability.

10. The **Specify Path and File Name** window appears; navigate to the desired location (here, **Desktop**), provide the File name as **MyVolume,** and click **Save**.

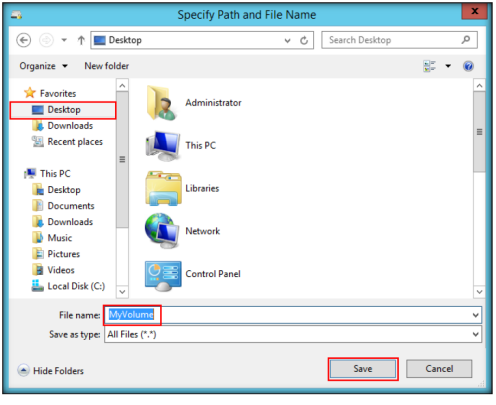


FIGURE 9.8: Windows Standard-Specify Path and File Name Window

The mode of operation used by VeraCrypt for encrypted partitions, drives, and virtual volumes is XTS.

11. After **saving** the file, the location of file containing the **VeraCrypt** volume is set; click **Next**.

VeraCrypt volumes do not contain known file headers and their content is indistinguishable from random data.

VeraCrypt currently supports the following hash algorithms:

- RIPEMD-160
- SHA-512
- Whirlpool

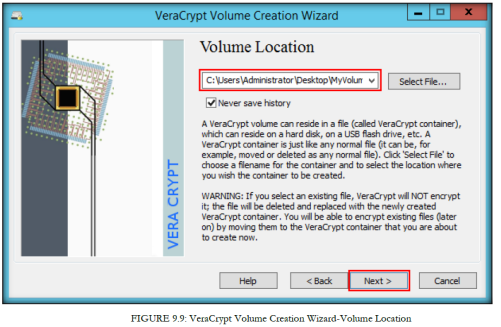


FIGURE 9.9: VeraCrypt Volume Creation Wizard-Volume Location

12. In the **Encryption Options** wizard, select the **AES** Encryption Algorithm and **SHA-512** Hash Algorithm, and click **Next**.

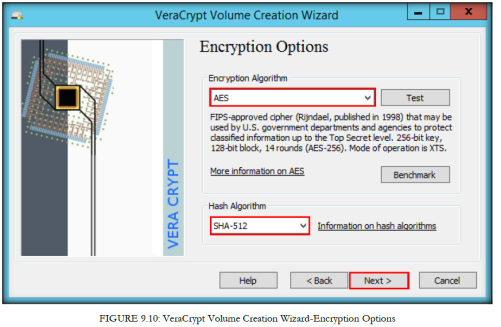


FIGURE 9.10: VeraCrypt Volume Creation Wizard-Encryption Options

13. In the **Volume Size** wizard, specify the size of the VeraCrypt container as **2 megabyte,** and click **Next**.

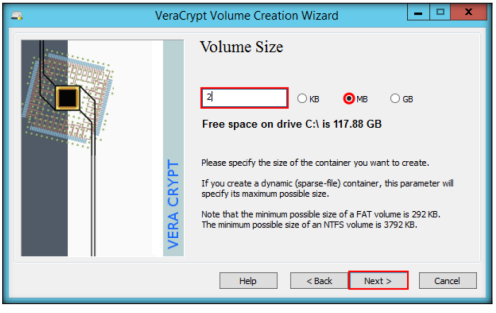


FIGURE 9.11: VeraCrypt Volume Creation Wizard-Volume Size

Note: The button "Next" will be disabled until passwords in both input fields are the same.

14. The **Volume Password** wizard appears; provide a **good password** in the **Password** field, retype it in the **Confirm** field, and click **Next**.
15. In this lab, the password used is **qwerty@123**.

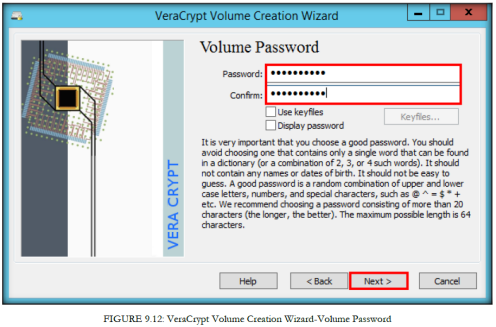


FIGURE 9.12: VeraCrypt Volume Creation Wizard-Volume Password

The longer you move the mouse, the better. This significantly increases the cryptographic strength of the encryption keys.

Note: A **VeraCrypt Volume Creation Wizard** warning pop-up appears; click **Yes**.

16. The Volume Format option appears. Select **FAT Filesystem**, and set the cluster to **Default**.

17. Move your mouse as **randomly** as possible within the Volume Creation Wizard window for at least **30 seconds**.
18. Click **Format**.

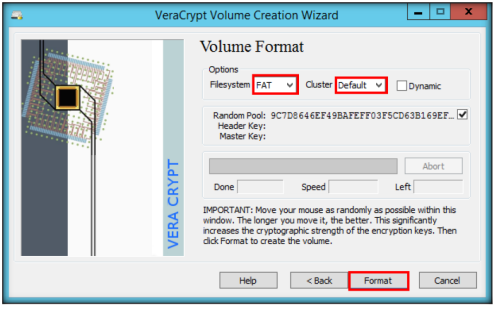


FIGURE 9.13: VeraCrypt Volume Creation Wizard-Volume Format

VeraCrypt volumes have no "signature" or ID strings. Until decrypted, they appear to consist solely of random data.

19. After clicking **Format,** VeraCrypt will create a file called **MyVolume** in the provided folder. This file depends on the VeraCrypt container (it will contain the encrypted VeraCrypt volume).
20. Depending on the **size of the volume**, it may take some time for volume creation.
21. Once the volume is created, a **VeraCrypt Volume Creation Wizard** dialog-box appears; click **OK**.

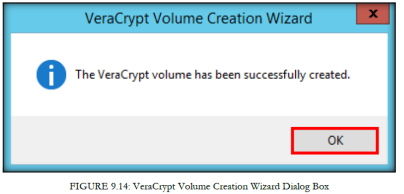


FIGURE 9.14: VeraCrypt Volume Creation Wizard Dialog Box

Free space on each VeraCrypt volume is filled with random data when the volume is created.

22. Click **OK** to close the dialog box.
23. You have **successfully** created a **VeraCrypt volume** (file container).

24. In the VeraCrypt Volume Creation wizard window, click **Exit**.

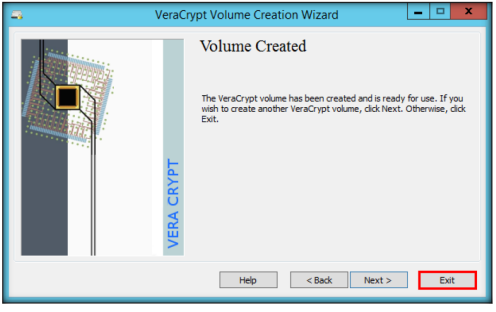


FIGURE 9.15: VeraCrypt Volume Creation Wizard-Volume Created

VeraCrypt is unable to secure data on a computer if an attacker physically accessed it and VeraCrypt is used on the compromised computer by the user again.

TASK 2

Mount a Volume

25. The **VeraCrypt** main window appears; select a drive (here, **I:**), and click **Select File...**.

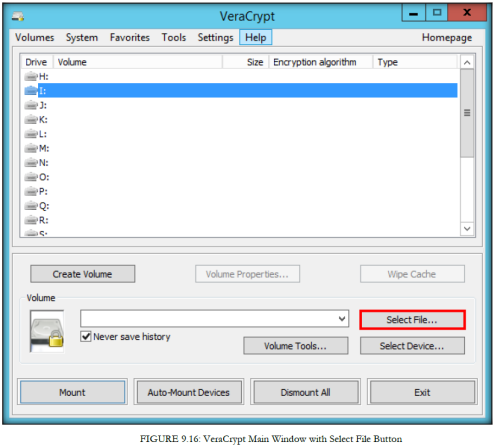


FIGURE 9.16: VeraCrypt Main Window with Select File Button

Mount options affect the parameters of the volume being mounted. The Mount Options dialog can be opened by clicking on the Mount Options button in the password entry dialog box.

26. The **Select a VeraCrypt Volume** window appears; navigate to **C:\Users\Administrator\Desktop**, click **MyVolume,** and click **Open**.

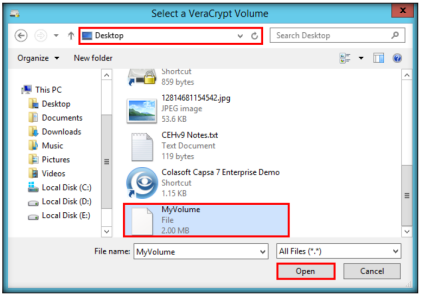


FIGURE 9.17: Windows Standard File Selector Window

Default mount options can be configured in the main program preferences (Settings → Preferences).

27. The window **closes** and you are returned to the **VeraCrypt** window. Click **Mount**.

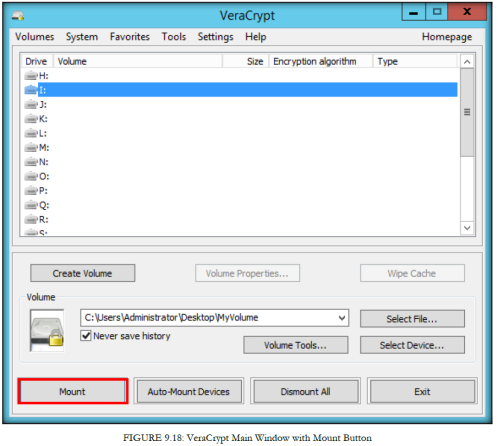


FIGURE 9.18: VeraCrypt Main Window with Mount Button

This option can be set in the password entry dialog so that it will apply only to that particular mount attempt. It can also be set as default in the Preferences.

28. The **Enter Password** dialog-box appears; type the password you specified earlier for this volume (in this lab, **qwerty@123**) in the **Password** input field, and click **OK**.

When a correct password is cached, volumes are automatically mounted after you click Mount. If you need to change mount options for a volume being mounted using a cached password, hold down the Control (Ctrl) key while clicking Mount, or select Mount with Options from the Volumes menu.

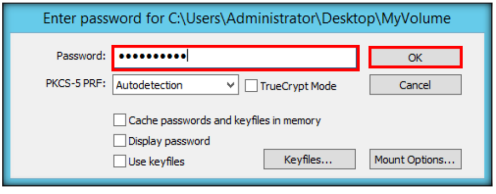


FIGURE 9.19: VeraCrypt Password Window

29. After the password is **verified**, VeraCrypt will **mount the volume,** as shown in the screenshot:

No data stored on an encrypted volume can be read (decrypted) without using the correct password or correct encryption key.

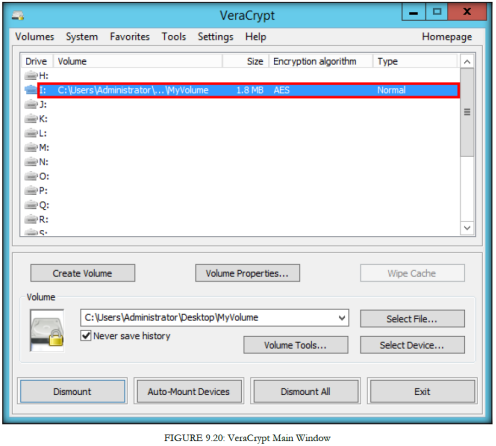


FIGURE 9.20: VeraCrypt Main Window

30. **MyVolume** has **successfully** mounted the container as a virtual disk (**I:**).
31. The virtual disk is entirely **encrypted** (including file names, allocation tables, free space, etc.) and behaves like a **real disk**.
32. You can copy or move files to this virtual disk to encrypt them.

33. Create a text document on the **Desktop** and name it **Test**.
34. Open the text document, and enter some text in it.
35. Click **File** in the menu bar, and click **Save**.

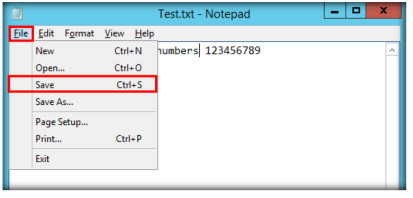


FIGURE 9.21: VeraCrypt Main Window with Dismount Button

VeraCrypt cannot automatically dismount all mounted VeraCrypt volumes on system shutdown/restart.

36. Copy the file from the **Desktop,** and paste it in **I:**. **Close** the window.

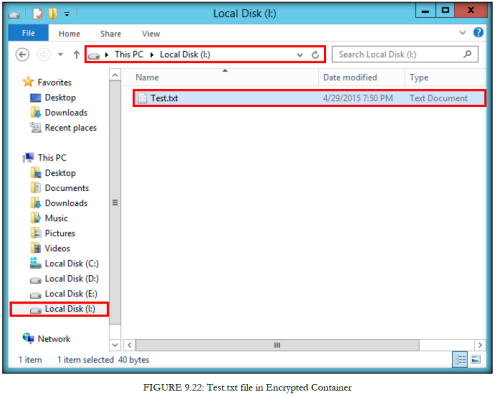


FIGURE 9.22: Test.txt file in Encrypted Container

37. Switch to **VeraCrypt** window, click **Dismount** and then click **Exit**.

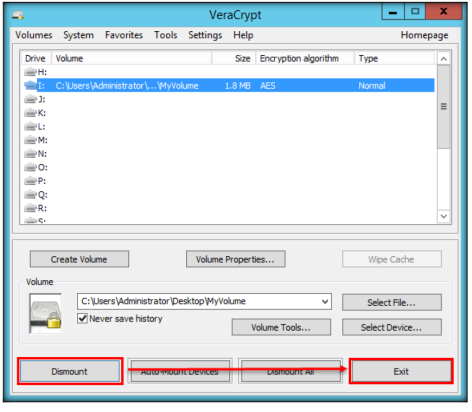


FIGURE 9.23: VeraCrypt Main Window with Dismount Button

38. The **I:** located in **This PC** disappears. This lab is used to demonstrate that, in cases of system hacks, if an attacker manages to gain remote access or complete access to the machine, he/she cannot find the encrypted volume—including its files—unless he/she is able to obtain the password. Thus, all sensitive information located on the encrypted volume is safeguarded.

Lab Analysis

Analyze and document the results related to this lab exercise.

PLEASE TALK TO YOUR INSTRUCTOR IF YOU HAVE QUESTIONS RELATED TO THIS LAB.

Internet Connection Required	
☐ Yes	☑ No
Platform Supported	
☑ Classroom	☑ iLabs

Lab 10

Basic Data Encrypting Using Rohos Disk Encryption

Rohos Disk is a program used to create hidden and protected partitions on a computer or USB flash drive, that password-protects/locks access to your Internet applications.

Lab Scenario

ICON KEY

- Valuable information
- Test your knowledge
- Web exercise
- Workbook review

Disk encryption works in a manner similar to text-message encryption. By using an encryption program for the user's disk, the user can safeguard all information burned onto the disk and save it from falling into the wrong hands. Disk-encryption software scrambles the information on the disk into an illegible code. The information must be decrypted to be read and used. To be an expert ethical hacker and penetration tester, you must have knowledge of these cryptography functions.

Lab Objectives

This lab will give you experience on encrypting data and show you how to do it. It will teach you how to:

- Create an encrypted drive for Windows
- Create a virtual encrypted drive for an external USB

Lab Environment

To complete this lab, you will need:

Tools demonstrated in this lab are available in D:\CEH-Tools\CEHv9 Module 18 Cryptography

- Rohos Disk Encryption located at **D:\CEH-Tools\CEHv9 Module 18 Cryptography\Disk Encryption Tools\Rohos Disk Encryption**
- You can also download the latest version of Rohos Disk Encryption from the link http://www.rohos.com/products/rohos-disk-encryption/
- If you decide to download the latest version, then screenshots shown in the lab might differ
- Follow the wizard driven installation instructions

- Windows Server 2012 running in host machine
- Administrative Privileges to run the tool

Lab Duration

Time: 15 Minutes

Overview of Rohos Disk Encryption

Rohos Disk Encryption creates hidden and password-protected partitions on a computer or USB flash drive. Rohos Disk uses an NIST-approved AES encryption algorithm with 256-bit encryption key length. Encryption is automatic and on-the-fly.

Lab Tasks

Note: Plug in a USB device to your machine before performing this lab.

TASK 1

Install Rohos Disk Encryption

1. To install Rohos Disk Encryption, navigate to **D:\CEH-Tools\CEHv9 Module 18 Cryptography\Disk Encryption Tools\Rohos Disk Encryption**.
2. Double-click **rohos.exe**; the **Select Setup Language** dialog box appears.
3. Select the language as **English,** and click **OK**.

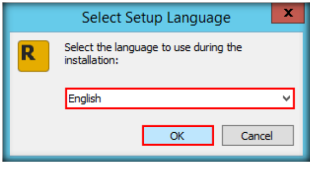


FIGURE 10.1: Select the Language

You can also download Rohos from http://www.rohos.com.

4. The **Setup** window appears; read the instruction, and click **Next**.

FIGURE 10.2: Rohos setup wizard

Portable Rohos Disk Browser allows to use encrypted partition on any PC without Admin rights, without install.

5. The **License Agreement** window appears; read the agreement carefully, select **I accept the agreement,** and click **Next**.

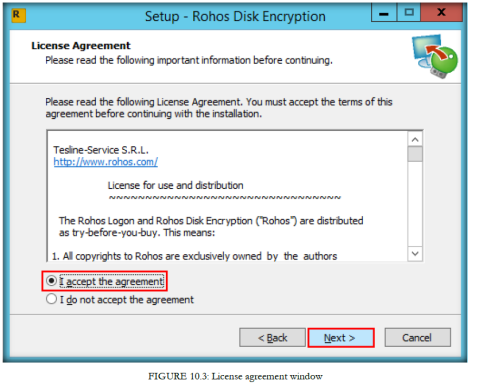


FIGURE 10.3: License agreement window

Encryption is automatic and on-the-fly. AES 256 bit key length. Using NIST compliant encryption standards.

6. Select the location in which you want the program to place the shortcut.

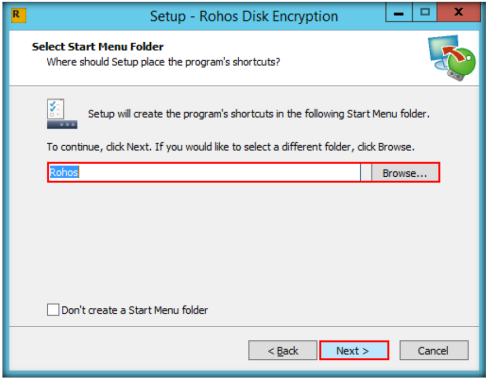


FIGURE 10.4: Select the destination folder

File Virtualization: prevents secret data leak outside encrypted disk on TEMP folders, Registry, Recent documents list, etc.

7. Check **Create a desktop,** and click **Next**.

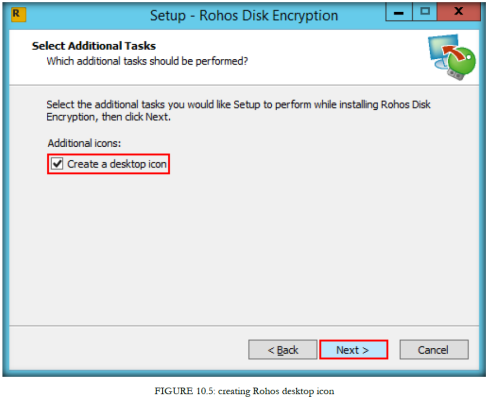
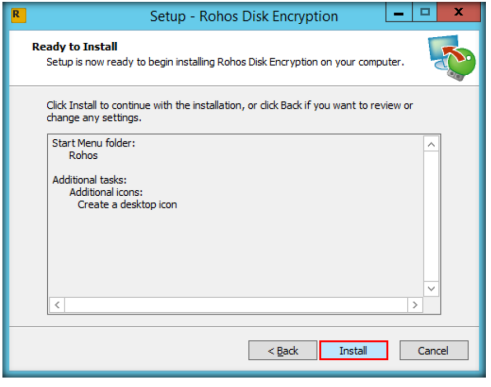


FIGURE 10.5: creating Rohos desktop icon

Any file or folder can be easily moved into Encrypted Rohos Disk with shredding afterwards.

8. Click **Install** to begin installation.



Secured virtual keyboard - protect encrypted disk password from a keylogger.

FIGURE 10.6: Rohos disk encryption installation

9. On completion of installation, click **Finish**.

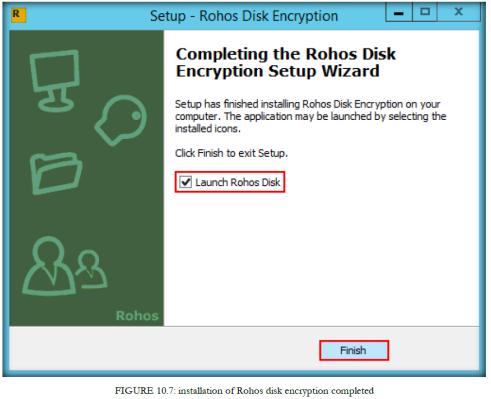


FIGURE 10.7: installation of Rohos disk encryption completed

TASK 2

Create an encrypted disk for Local Machine

10. The **Rohos Get Ready Wizard** window appears, displaying the **Disk Encryption** step. Specify the password (**qwerty@123**) in the respective fields, and click **Next**.
11. Alternatively, you can launch it from the **Start** menu apps of **Windows Server 2012**.

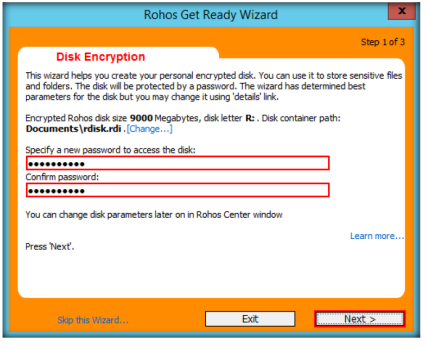


FIGURE 10.8: Select password for accessing the disk

Rohos disk uses NIST approved AES encryption algorithm, 256-bit encryption key length.

12. The **Setup a key** step appears; click **Next**.

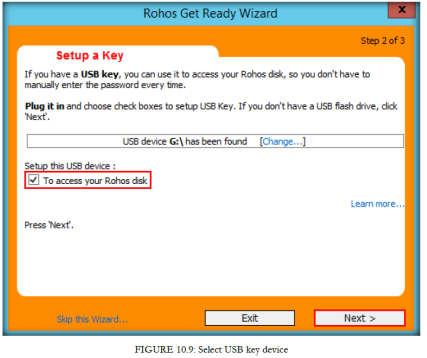


FIGURE 10.9: Select USB key device

Rohos cares about usability. Your first Encrypted Drive can be turned on with a single click or automatically on system startup.

13. The **Rohos update** section appears; click **Finish**.

FIGURE 10.10: Rohos disk encryption updates window

Partition password reset option allows creating a backup file to access your secured disk if you forgot your password or lost USB key.

14. Wait until the encrypted volume is created.

This option brings affordable and AES 256 strength encryption solution to improve security issues by preventing unauthorized access to your Internet apps, such as Google Chrome, Firefox.

FIGURE 10.11: Disk creation in progress

15. On creating the encrypted volume, a new **9000 MB** (8.78 GB) drive (**R:**\) appears in This PC, as shown in the screenshot:

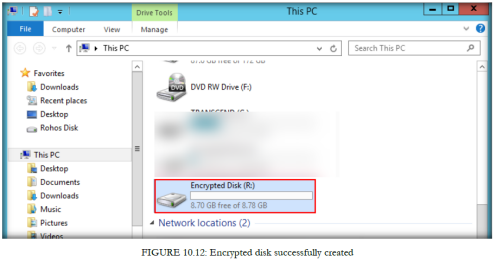


FIGURE 10.12: Encrypted disk successfully created

16. This drive appears only when you are connected to Rohos Disk Encryption, and disappears when you exit it.
17. So, when you want to hide any important files/directories from anyone accessing your system, you can place them in this drive and access them whenever required (by launching Rohos and entering the password).

TASK 3

Create an encrypted disk USB Disk

18. To create an encrypted USB drive, click **Encrypt USB drive** in the Rohos Disk Encryption GUI.

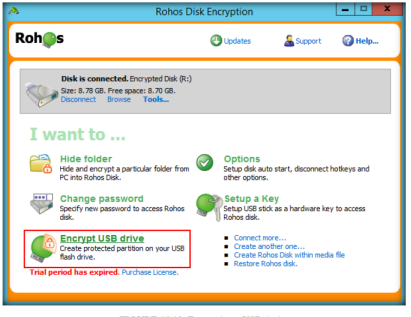


FIGURE 10.13: Encrypting a USB device

19. The **Encrypt USB drive** dialog box appears; click **Change...** in the Encrypted partition properties section.

FIGURE 10.14: Encrypt USB drive dialog-box

20. The **Disk details** window appears; choose the Disk letter **M:**, set the disk size to **60,** and click **OK**.

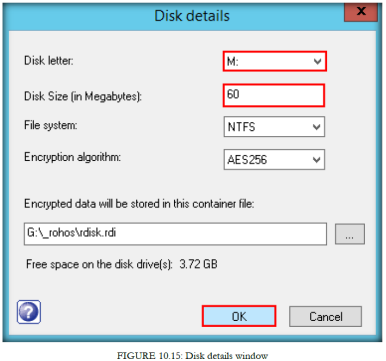


FIGURE 10.15: Disk details window

21. This creates an Encrypted USB drive (**M:**) of 60 MB.
22. You need to apply a password for the disk, so that whenever someone wants to access the drive, they need to specify the password.
23. Specify the password (here, **test@123**) in both fields, and click **Create disk**.

FIGURE 10.16: Encrypt USB drive window

24. Wait until the disk is created.

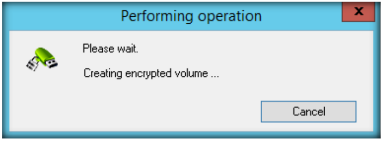


FIGURE 10.17: Disk creation in progress

25. On successful creation of the disk, a **Rohos Disk Encryption** dialog box appears; click **OK**.

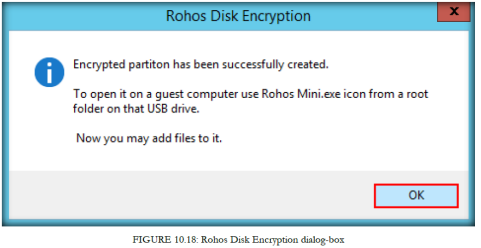


FIGURE 10.18: Rohos Disk Encryption dialog-box

26. The **Encrypted disk** (here, **M:**) of **60 MB** is created successfully, as shown in the screenshot:

You can open or Save your protected documents right from MS Word (Excel) by clicking on the personal disk icon.

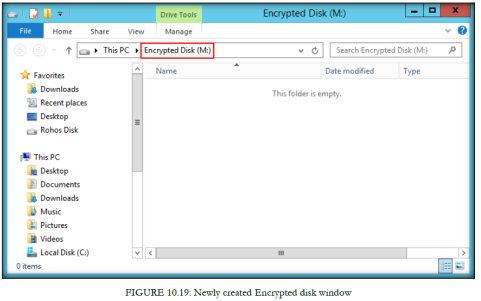


FIGURE 10.19: Newly created Encrypted disk window

27. The files you place in this drive will automatically be placed in the external USB.
28. In this lab, we are copying the folder **Rohos Disk Encryption** from **D:\CEH-Tools\CEHv9 Module 18 Cryptography\Disk Encryption Tools** to **M:**.

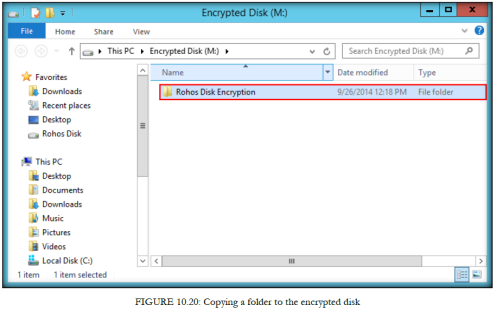


FIGURE 10.20: Copying a folder to the encrypted disk

TASK 4

Access Files in the Encrypted Disk

29. Now, if you want to access this file, open the external USB drive which has been connected to your computer, and double-click **Rohos Mini Drive (Portable).exe**.

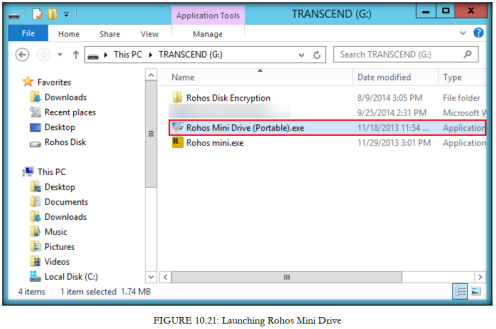


FIGURE 10.21: Launching Rohos Mini Drive

30. A **Rohos** dialog box appears asking you to enter the password. You need to enter the password which you specified at the time of creating the encrypted USB disk (**M:**).

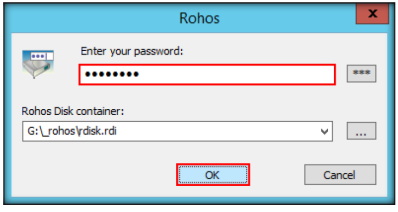


FIGURE 10.22: Rohos dialog-box

31. A **Rohos Disk Browser** window appears, displaying the folder that was placed in **M:**, as shown in the screenshot:

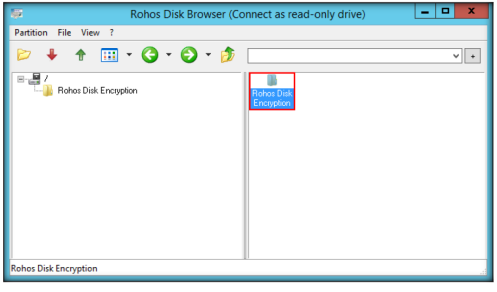


FIGURE 10.23: Rohos Disk Browser window containing the file placed in M:\

32. When you want to share sensible information with someone via USB, you can use this application to store the files in an encrypted disk, and share the password with that person.

33. The person with whom you want to share the files can access them only after entering the correct password.

34. This way, you can protect the files from being viewed by a third person and thereby safeguard them.

Lab Analysis

Analyze and document the results related to this lab exercise.

PLEASE TALK TO YOUR INSTRUCTOR IF YOU HAVE QUESTIONS RELATED TO THIS LAB.

Internet Connection Required	
☑ Yes	☐ No
Platform Supported	
☑ Classroom	☐ iLabs

Lab 11

Basic Data Encryption Using CrypTool

CrypTool is a freeware program that enables you to apply and analyze cryptographic mechanisms. It has the typical look and feel of a modern Windows application. CrypTool includes every state-of-the-art cryptographic function and allows you to learn and use cryptography within the same environment.

ICON KEY

- Valuable information
- Test your knowledge
- Web exercise
- Workbook review

Lab Scenario

Most security initiatives are defensive strategies aimed at protecting the perimeter of the network. But these efforts may ignore a crucial vulnerability sensitive data stored on networked servers are at risk from attackers who only need to find one way inside the network to access this confidential information. Additionally, perimeter defenses like firewalls cannot protect stored sensitive data from the internal threat employees with the means to access and exploit this data. Encryption can provide strong security for sensitive data stored on local or network servers. To be an expert ethical hacker and penetration tester, you must have knowledge of cryptography functions.

Lab Objectives

This lab will give you experience on encrypting data and show you how to do so. It will teach you how to:

- Use encrypting/decrypting command
- Visualize several algorithms
- Calculate hash values and analysis

Tools demonstrated in this lab are available in D:\CEH-Tools\CEHv9 Module 18 Cryptography

Lab Environment

To complete this lab, you will need:

- CrypTool located at **D:\CEH-Tools\CEHv9 Module 18 Cryptography\Cryptanalysis Tools\CrypTool**

- You can also download the latest version of CrypTool from the link http://www.cryptool.org/en/download-ct1-en
- If you decide to download the latest version, then screenshots shown in the lab might differ

CrypTool is a free e-learning application for Windows.

- Windows Server 2012 running on host machine
- Windows 8.1 running on virtual machine
- Administrative Privileges to run the tool

Lab Duration

Time: 15 Minutes

Overview of CrypTool

CrypTool is a free, open-source e-learning application used in the implementation and analysis of cryptographic algorithms. It was originally designed for internal business application for information security training.

Lab Tasks

TASK 1

Encrypting the Data

1. Navigate to **D:\CEH-Tools\CEHv9 Module 18 Cryptography\Cryptanalysis Tools\CrypTool**, double-click **SetupCrypTool_1_4_31_Beta6b_r3670_VS2008_en.exe,** and follow the wizard driven installation steps to install the application.
2. On completing the installation, launch **CrypTool** application from the **Apps** screen.

You can also download CrypTool from http://www.cryptool.org.

FIGURE 11.1: Launching CrypTool from Apps screen

3. The **How to Start** dialog box appears; check **Don't show this message again,** and click **Close**.

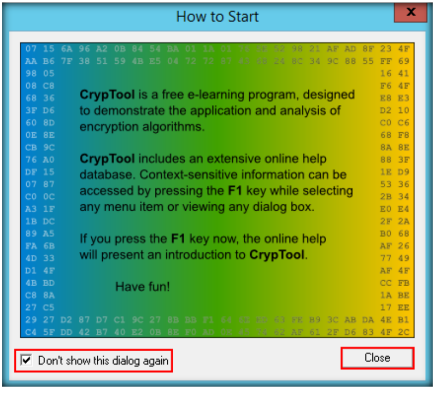


FIGURE 11.2: How to Start Dialog box

CrypTool Online provides an exciting insight into the world of cryptology with a variety of ciphers and encryption methods.

4. The main window of **CrypTool** appears; close the **startingexample-en.txt** window.

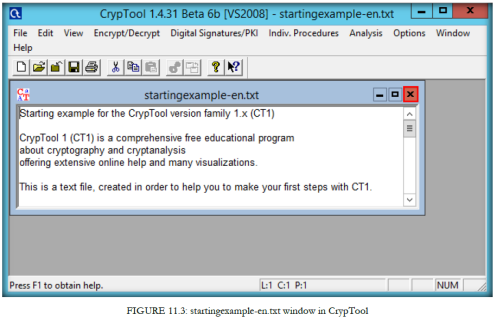


FIGURE 11.3: startingexample-en.txt window in CrypTool

5. To **encrypt** data, click the **File** option from the menu bar, and select **New**.

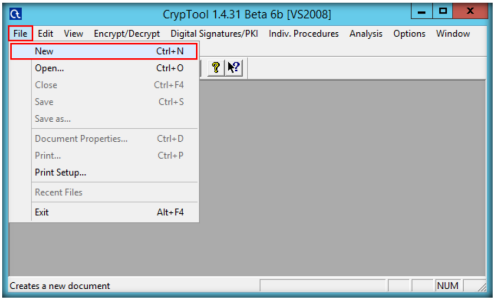


FIGURE 11.4: Choosing a new file to crypt

CrypTool was originally designed for internal business application for information security.

6. Type some content in the opened **Unnamed1** Notepad of CrypTool. You will be encrypting this content.
7. Select **Encrypt/Decrypt → Symmetric (modern) → RC2...** in the Menu bar.

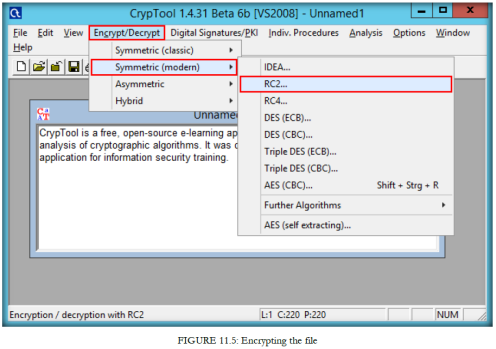


FIGURE 11.5: Encrypting the file

8. The **Key Entry: RC2** dialog box appears; select **Key length** (here, **8 bits**) from the drop-down list.
9. Enter the key using hexadecimal characters (**05**), and click **Encrypt**.

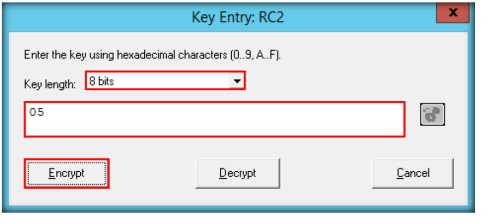


FIGURE 11.6: Encrypting the file

10. The RC2 encryption of **Unnamed1** notepad displays, as shown in the screenshot:

CrypTool includes every state-of-the-art cryptographic function and allows you to learn and use cryptography within the same environment.

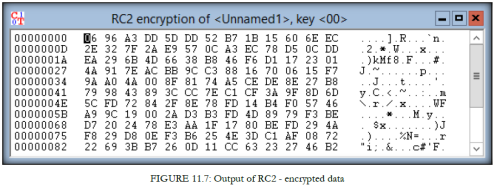


FIGURE 11.7: Output of RC2 - encrypted data

11. To save the file, click **File** in the menu bar, and select **Save**.

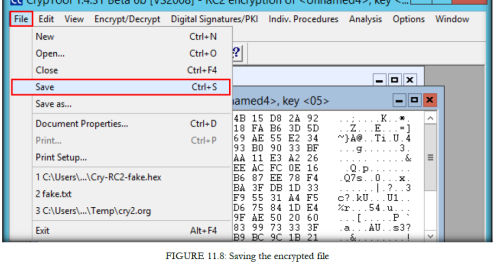


FIGURE 11.8: Saving the encrypted file

12. The **Save As** dialog-box appears; choose a location where you want to save the file (**Desktop**), specify a file name (**Cry-RC2-Unnamed4.hex**), and click **Save**.

Note: The file name may differ in your lab environment.

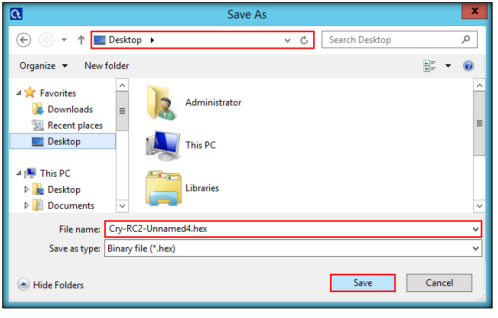


FIGURE 11.9: Saving the encrypted file

CrypTool Online provides an exciting insight into the world of **cryptology** with a variety of ciphers and encryption methods.

13. Now, you can send this file to the intended person by email or any other means and provide him/her with the hex value, which will be used to decrypt the file.
14. To share the file, you may copy the encrypted file from the Desktop to **D:\CEH-Tools\CEHv9 Module 18 Cryptography\Cryptanalysis Tools\CrypTool**.
15. Assume that you are the intended recipient (working on Windows 8.1) of the Crypted file through the shared network drive.

TASK 2

Decrypting the Data

16. Log into **Windows 8.1** virtual machine, navigate to **Z:\CEHv9 Module 18 Cryptography\Cryptanalysis Tools\CrypTool**, double-click **SetupCrypTool_1_4_31_Beta6b_r3670_VS2008_en.exe**, and follow the steps to install the application.
17. In the meanwhile, copy the Crypted hex file (**Cry-RC2-Unnamed4.hex**) from **Z:\CEHv9 Module 18 Cryptography\Cryptanalysis Tools\CrypTool**, and save it to the **Desktop**.
18. Launch the **CrypTool** application.
19. The **How to Start** dialog box appears; check **Don't show this message again,** and click **Close**.

20. The main window of **CrypTool** appears; close the **startingexample-en.txt** window.

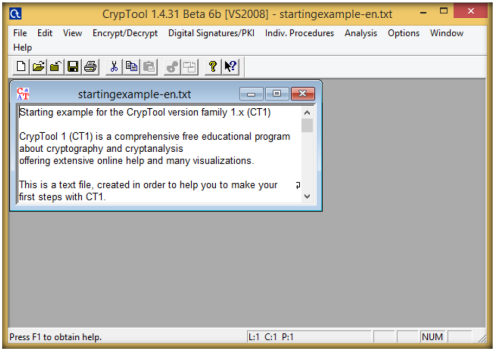


FIGURE 11.10: startingexample-en.txt window in CrypTool

21. To **decrypt** data, click **File** in the menu bar, and select **Open...**.

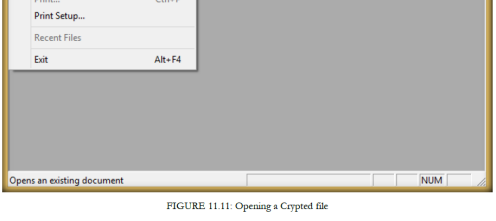


FIGURE 11.11: Opening a Crypted file

22. The **Open** dialog-box appears; select **All files** from the drop-down list, navigate to the location of the file (**Desktop**), select it, and click **Open**.

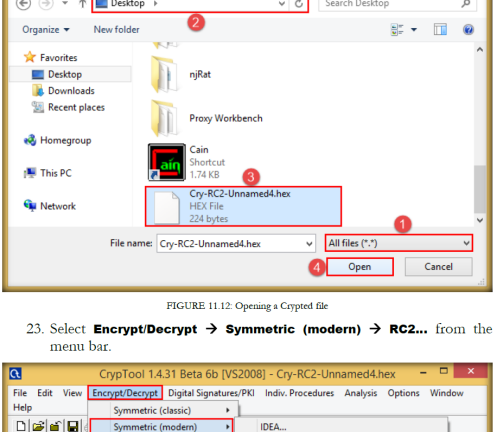


FIGURE 11.12: Opening a Crypted file

23. Select **Encrypt/Decrypt → Symmetric (modern) → RC2...** from the menu bar.

FIGURE 11.13: Select the RC2 Encryption algorithm

24. The **Key Entry: RC2** dialog-box appears; select **Key length** (here, **8 bits**) from the drop-down list.
25. Enter the hexadecimal key (**05**) that was used to encrypt the file, and click **Decrypt**.

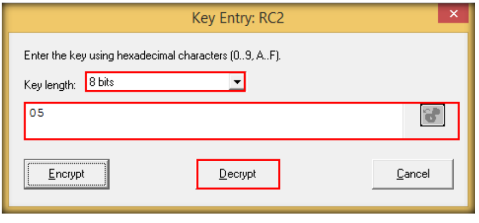


FIGURE 11.14: Decrypting the file

26. The decrypted text appears, as shown in the screenshot:

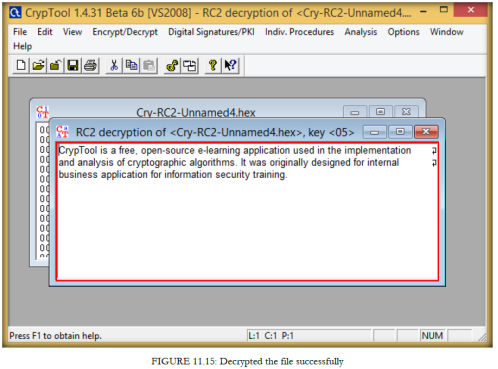


FIGURE 11.15: Decrypted the file successfully

27. This way, files can be encrypted using CrypTool and shared with an individual in a secure manner, so that no one can intercept its data.

Lab Analysis

Analyze and document the results related to this lab exercise.

PLEASE TALK TO YOUR INSTRUCTOR IF YOU HAVE QUESTIONS RELATED TO THIS LAB.

Internet Connection Required	
☐ Yes	☑ No
Platform Supported	
☑ Classroom	☑ iLabs