Exemplar: Decrypt an encrypted a message

Activity overview

Previously, you learned about cryptography and how encryption and decryption can be used to secure information online. You were also introduced to the Caesar cipher, one of the earliest cryptographic algorithms used to protect people's privacy.

As a security analyst, it's important that you understand the role of encryption to secure data online and that you're familiar with the right security controls to do so.

In this lab activity, you'll be guided through some basic cryptographic activities using Linux commands to decrypt files and reveal hidden messages.

This exemplar is a walkthrough of the previous Qwiklab activity, including detailed instructions and solutions. You may use this exemplar if you were unable to complete the lab and/or you need extra guidance in competing lab tasks. You may also refer to this exemplar to prepare for the graded quiz in this module.

Scenario

In this scenario, all of the files in your home directory have been encrypted. You'll need to use Linux commands to break the Caesar cipher and decrypt the files so that you can read the hidden messages they contain.

Here's how you'll do this task: **First**, you'll explore the contents of the home directory and read the contents of a file. **Next**, you'll find a hidden file and decrypt the Caesar cipher it contains. **Finally**, you'll decrypt the encrypted data file to recover your data and reveal the hidden message.

OK, it's time to decrypt some messages in Linux!

Note: *The lab starts with you logged in as user* **analyst***, with your home directory,* **/home/analyst***, as the current working directory.*

Task 1. Read the contents of a file

The lab starts in your home directory, **/home/analyst**, as the current working directory.

In this task, you need to explore the contents of your home directory and read the contents of a file to get further instructions.

1. Use the **ls** command to list the files in the current working directory.

The command to complete this step:

```
ls /home/analyst
```

Two files, **Q1.encrypted** and **README.txt**, and a subdirectory, **caesar**, are listed:

```
Q1.encrypted  README.txt caesar
```

The **README.txt** file contains an important message with instructions you need to follow.

2. Use the **cat** command to list the contents of the **README.txt** file.

The command to complete this step:

```
cat README.txt
```

This will display the following output:

```
Hello,
All of your data has been encrypted. To recover your data, you will need to solve a cipher.
```

The message in the **README.txt** file advises that the **caesar** subdirectory contains a hidden file.

In the next task, you'll need to find the hidden file and solve the Caesar cipher that protects it. The file contains instructions on how to recover your data.

Task 2. Find a hidden file

In this task, you need to find a hidden file in your home directory and decrypt the Caesar cipher it contains. This task will enable you to complete the next task.

1. First, use the **cd** command to change to the **caesar** subdirectory of your home directory:

```
1   cd caesar
```

2. Use the **ls -a** command to list all files, including hidden files, in your home directory.

The command to complete this step:

```
1   ls -a
```

This will display the following output:

```
1   .  ..  .leftShift3
```

Hidden files in Linux can be identified by their name starting with a period (.).

3. Use the **cat** command to list the contents of the **.leftShift3** file.

The command to complete this step:

```
1   cat .leftShift3
```

The message in the **.leftShift**3 file appears to be scrambled. This is because the data has been encrypted using a Caesar cipher. This cipher can be solved by shifting each alphabet character to the left or right by a fixed number of spaces. In this example, the shift is three letters to the left. Thus "d" stands for "a", and "e" stands for "b".

4. You can decrypt the Caesar cipher in the **.leftshift3** file by using the following command:

```
cat .leftShift3 | tr "d-za-cD-ZA-C" "a-zA-Z"
```

Note: *The* **tr** *command translates text from one set of characters to another, using a mapping. The first parameter to the* **tr** *command represents the input set of characters, and the second represents the output set of characters. Hence, if you provide parameters "abcd" and "pqrs", and the input string to the* **tr** *command is "ac", the output string will be "pr".*

This will display the following output:

```
In order to recover your files you will need to enter the following command:

openssl aes-256-cbc -pbkdf2 -a -d -in Q1.encrypted -out Q1.recovered -k ettubrute
```

In this case, the command **tr "d-za-cD-ZA-C" "a-zA-Z"** translates all the lowercase and uppercase letters in the alphabet back to their original position. The first character set, indicated by **"d-za-cD-ZA-C"**, is translated to the second character set, which is **"a-zA-Z"**.

Note: *The output provides you with the command you need to solve the next task!*

You don't need to copy the command revealed in the output. It will be provided in the next task.

5. Now, return to your home directory before completing the next task:

```
cd ~
```

Task 3. Decrypt a file

Now that you have solved the Caesar cipher, in this task you need to use the command revealed in **.leftshift3** to decrypt a file and recover your data so you can read the message it contains.

1. Use the exact command revealed in the previous task to decrypt the encrypted file:

```
openssl aes-256-cbc -pbkdf2 -a -d -in Q1.encrypted -out Q1.recovered -k ettubrute
```

Although you don't need to memorize this command, to help you better understand the syntax used, let's break it down.

In this instance, the **openssl** command reverses the encryption of the file with a secure symmetric cipher, as indicated by **AES-256-CBC**. The **-pbkdf2** option is used to add extra security to the key, and **-a** indicates the desired encoding for the output. The **-d** indicates decrypting, while -in specifies the input file and **-out** specifies the output file. The **-k** specifies the password, which in this example is **ettubrute**.

2. Use the **ls** command to list the contents of your current working directory again.

The command to complete this step:

```
ls
```

The new file **Q1.**recovered in the directory listing is the decrypted file and contains a message.

3. Use the **cat** command to list the contents of the **Q1.recovered** file.

The command to complete this step:

```
1   cat Q1.recovered
```

This will display the following output: