Networking Academy

Nama : Andi Raqueela Magistra Azrika
NIM : 22/494745/SV/20904
Kelas : RI4B2

Lab - Encrypting and Decrypting Data using a Hacker Tool

Objectives

Part 1: Create and Encrypt Files

Part 2: Recover Encrypted Zip File Passwords

Background / Scenario

What if you work for a large corporation that had a corporate policy regarding removable media? Specifically, it states that only encrypted zipped documents can be copied to portable USB flash drives.

In this scenario, the Chief Financial Officer (CFO) is out-of-town on business and has contacted you in a panic with an emergency request for help. While out-of-town on business, he attempted to unzip important documents from an encrypted zip file on a USB drive. However, the password provided to open the zip file is invalid. The CFO contacted you to see if there was anything you could to do.

Note: The provided scenario is simple and only serves as an example.

There may some tools available to recover lost passwords. This is especially true in situations such as this where the cybersecurity analyst could acquire pertinent information from the CFO. The pertinent information could be the length of the password and an idea of what it could be. Knowing pertinent information dramatically helps when attempting to recover passwords.

Examples of password recovery utilities and programs include hashcat, John the Ripper, Lophtcrack, and others. In our scenario, we will use **fcrackzip** which is a simple Linux utility to recover the passwords of encrypted zip files.

Consider that these same tools can be used by cybercriminals to discover unknown passwords. Although they would not have access to some pertinent information, with time, it is possible to discover passwords to open encrypted zip files. The amount of time required depends on the password strength and the password length. Longer and more complex passwords (mix of different types of characters) are more secure.

In this lab, you will:

- Create and encrypt sample text files.
- Decrypt the encrypted zip file.

Note: This lab should be used for instructional purposes only. The methods presented here should NOT be used to secure truly sensitive data.

Required Resources

- CyberOps Workstation virtual machine

Instructions

Part 1: Create and Encrypt Files

In this part, you will create a few text files that will be used to created encrypted zip files in the next step.

Step 1: Create text files.

a. Start the CyberOps Workstation VM.

b. Open a terminal window. Verify that you are in the analyst home directory. Otherwise, enter **cd ~** at the terminal prompt.

```
[analyst@secOps ~]$ cd -
/home/analyst
```

c. Create a new folder called Zip-Files using the **mkdir Zip-Files** command.

```
[analyst@secOps ~]$ mkdir Zip-Files
[analyst@secOps ~]$ ls
Desktop  Downloads  lab.support.files  second_drive  Zip-Files
```

d. Move into that directory using the **cd Zip-Files** command.

```
[analyst@secOps ~]$ cd Zip-Files/
```

e. Enter the following to create three text files.

```
[analyst@secOps Zip-Files]$ echo This is a sample text file > sample-1.txt
[analyst@secOps Zip-Files]$ echo This is a sample text file > sample-2.txt
[analyst@secOps Zip-Files]$ echo This is a sample text file > sample-3.txt
```

```
[analyst@secOps Zip-Files]$ echo This is a sample text file > sample-1.txt
[analyst@secOps Zip-Files]$ echo This is a sample text file > sample-2.txt
[analyst@secOps Zip-Files]$ echo This is a sample text file > sample-3.txt
```

f. Verify that the files have been created, using the **ls** command.

```
[analyst@secOps Zip-Files]$ ls -l
total 12
-rw-r--r-- 1 analyst analyst 27 May 13 10:58 sample-1.txt
-rw-r--r-- 1 analyst analyst 27 May 13 10:58 sample-2.txt
-rw-r--r-- 1 analyst analyst 27 May 13 10:58 sample-3.txt
```

```
[analyst@secOps Zip-Files]$ ls -l
total 12
-rw-r--r-- 1 analyst analyst 27 May  1 00:42 sample-1.txt
-rw-r--r-- 1 analyst analyst 27 May  1 00:43 sample-2.txt
-rw-r--r-- 1 analyst analyst 27 May  1 00:43 sample-3.txt
```

```
[analyst@secOps Zip-Files]$ cat sample-1.txt
This is a sample text file
```

Step 2: Zip and encrypt the text files.

Next, we will create several encrypted zipped files using varying password lengths. To do so, all three textfiles will be encrypted using the **zip** utility.

a. Create an encrypted zip file called **file-1.zip** containing the three text files using the following command:

```
[analyst@secOps Zip-Files]$ zip –e file-1.zip sample*
```

```
[analyst@secOps Zip-Files]$ zip -e file-1.zip sample*
```

b. When prompted for a password, enter a one-character password of your choice. In the example, the letter **B** was entered. Enter the same letter when prompted to verify.

```
[analyst@secOps Zip-Files]$ zip -e file-1.zip sample-*
Enter password:
Verify password:
adding: sample-1.txt (stored    0%)
adding: sample-2.txt (stored    0%)
adding: sample-3.txt (stored    0%)
```

```
[analyst@secOps Zip-Files]$ zip -e file-1.zip sample*
Enter password:
Verify password:
  adding: sample-1.txt (stored 0%)
  adding: sample-2.txt (stored 0%)
  adding: sample-3.txt (stored 0%)
```

c. Repeat the procedure to create the following 4 other files

- **file-2.zip** using a 2-character password of your choice. In our example, we used **R2**.
- **file-3.zip** using a 3-character password of your choice. In our example, we used **0B1**.
- **file-4.zip** using a 4-character password of your choice. In our example, we used **Y0Da**.
- **file-5.zip** using a 5-character password of your choice. In our example, we used **C-3P0**.

```
[analyst@secOps Zip-Files]$ zip -e file-2.zip sample*
Enter password:
Verify password:
  adding: sample-1.txt (stored 0%)
  adding: sample-2.txt (stored 0%)
  adding: sample-3.txt (stored 0%)
[analyst@secOps Zip-Files]$ zip -e file-3.zip sample*
Enter password:
Verify password:
  adding: sample-1.txt (stored 0%)
  adding: sample-2.txt (stored 0%)
  adding: sample-3.txt (stored 0%)
```

```
[analyst@secOps Zip-Files]$ zip -e file-4.zip sample*
Enter password:
Verify password:
  adding: sample-1.txt (stored 0%)
  adding: sample-2.txt (stored 0%)
  adding: sample-3.txt (stored 0%)
[analyst@secOps Zip-Files]$ zip -e file-5.zip sample*
Enter password:
Verify password:
  adding: sample-1.txt (stored 0%)
  adding: sample-2.txt (stored 0%)
  adding: sample-3.txt (stored 0%)
```

d. Verify that all zipped files have been created using the **ls -l f*** command.

```
[analyst@secOps Zip-Files]$ ls -l f*
-rw-r--r-- 1 analyst  Analyst 643  May  13 11:01 file-1.zip
-rw-r--r-- 1 analyst  analyst 643  May  13 11:02 file-2.zip
-rw-r--r-- 1 analyst  analyst 643  May  13 11:03 file-3.zip
-rw-r--r-- 1 analyst  analyst 643  May  13 11:03 file-4.zip
-rw-r--r-- 1 analyst  analyst 643  May  13 11:03 file-5.zip
```

```
[analyst@secOps Zip-Files]$ ls -l f*
-rw-r--r-- 1 analyst analyst 643 May  1 00:53 file-1.zip
-rw-r--r-- 1 analyst analyst 643 May  1 00:55 file-2.zip
-rw-r--r-- 1 analyst analyst 643 May  1 00:56 file-3.zip
-rw-r--r-- 1 analyst analyst 643 May  1 00:57 file-4.zip
-rw-r--r-- 1 analyst analyst 643 May  1 00:58 file-5.zip
```

e. Attempt to open a zip using an incorrect password as shown.

```
[analyst@secOps Zip-Files]$ unzip file-1.zip
Archive:  file-1.zip
[file-1.zip] sample-1.txt password:
password incorrect--reenter:
password incorrect--reenter:
   skipping: sample-1.txt            incorrect password
[file-1.zip] sample-2.txt password:
password incorrect--reenter:
password incorrect--reenter:
   skipping: sample-2.txt            incorrect password
[file-1.zip] sample-3.txt password:
password incorrect--reenter:
password incorrect--reenter:
   skipping: sample-3.txt            incorrect password
```

```
[analyst@secOps Zip-Files]$ unzip file-1.zip
Archive:  file-1.zip
[file-1.zip] sample-1.txt password:
password incorrect--reenter:
password incorrect--reenter:
   skipping: sample-1.txt            incorrect password
[file-1.zip] sample-2.txt password:
password incorrect--reenter:
password incorrect--reenter:
   skipping: sample-2.txt            incorrect password
[file-1.zip] sample-3.txt password:
password incorrect--reenter:
password incorrect--reenter:
   skipping: sample-3.txt            incorrect password
```

Part 2: Recover Encrypted Zip File Passwords

In this part, you will use the **fcrackzip** utility to recover lost passwords from encrypted zipped files. Fcrackzip searches each zip file given for encrypted files and tries to guess the password using brute-force methods.

The reason we created zip files with varying password lengths was to see if password length influences the time it takes to discover a password.

Step 1: Introduction to fcrackzip

From the terminal window, enter the **fcrackzip –h** command to see the associated command options.

```
[analyst@secOps Zip-Files]$ fcrackzip -h

fcrackzip version 1.0, a fast/free zip password cracker
written by Marc Lehmann <pcg@goof.com> You can find more info on
http://www.goof.com/pcg/marc/

USAGE: fcrackzip
         [-b|--brute-force]           use brute force algorithm
         [-D|--dictionary]            use a dictionary
         [-B|--benchmark]             execute a small benchmark
         [-c|--charset characterset]  use characters from charset
         [-h|--help]                  show this message
         [--version]                  show the version of this program
         [-V|--validate]              sanity-check the algortihm
         [-v|--verbose]               be more verbose
         [-p|--init-password string]  use string as initial password/file
         [-l|--length min-max]        check password with length min to max
         [-u|--use-unzip]             use unzip to weed out wrong passwords
         [-m|--method num]            use method number "num" (see below)
         [-2|--modulo r/m]            only calculcate 1/m of the password
         file...                    the zipfiles to crack

methods compiled in (* = default):

 0: cpmask
 1: zip1
*2: zip2, USE_MULT_TAB
```

In our examples, we will be using the **–v**, **-u**, and **-l** command options. The -l option will be listed last because it specifies the possible password length. Feel free to experiment with other options.

Step 2: Recovering Passwords using fcrackzip

a. Now attempt to recover the password of the **file-1.zip** file. Recall, that a one-character password was used to encrypt the file. Therefore, use the following **fcrackzip** command:

```
[analyst@secOps Zip-Files]$ fcrackzip -vul 1-4 file-1.zip
found file 'sample-1.txt', (size cp/uc  39/  27, flags 9, chk 5754) found
file 'sample-2.txt', (size cp/uc  39/  27, flags 9, chk 5756) found file
'sample-3.txt', (size cp/uc  39/  27, flags 9, chk 5757)


PASSWORD FOUND!!!!: pw == B
```

Note: The password length could have been set to less than 1 – 4 characters.

```
[analyst@secOps Zip-Files]$ fcrackzip -vul 1-4 file-1.zip
found file 'sample-1.txt', (size cp/uc     39/    27, flags 9, chk 0558)
found file 'sample-2.txt', (size cp/uc     39/    27, flags 9, chk 0562)
found file 'sample-3.txt', (size cp/uc     39/    27, flags 9, chk 0569)


PASSWORD FOUND!!!!: pw == B
```

How long does it take to discover the password?
= Durasi yang dibutuhkan untuk menemukan kata sandi dari file-1.zip hanya membutuhkan waktu kurang dari satu detik. Hal ini disebabkan karena kata sandi yang digunakan disini hanya terdiri dari 1 karakter saja.

b. Now attempt to recover the password of the **file-2.zip** file. Recall, that a two-character password was used to encrypt the file. Therefore, use the following **fcrackzip** command:

```
[analyst@secOps Zip-Files]$ fcrackzip -vul 1-4 file-2.zip
found file 'sample-1.txt', (size cp/uc  39/  27, flags 9, chk 5754) found
file 'sample-2.txt', (size cp/uc  39/  27, flags 9, chk 5756) found file
'sample-3.txt', (size cp/uc  39/  27, flags 9, chk 5757)


PASSWORD FOUND!!!!: pw == R2
```

```
[analyst@secOps Zip-Files]$ fcrackzip -vul 1-4 file-2.zip
found file 'sample-1.txt', (size cp/uc     39/    27, flags 9, chk 0558)
found file 'sample-2.txt', (size cp/uc     39/    27, flags 9, chk 0562)
found file 'sample-3.txt', (size cp/uc     39/    27, flags 9, chk 0569)


PASSWORD FOUND!!!!: pw == R2
```

How long does it take to discover the password?

= Sama seperti file-1.zip, durasi waktu yang dibutuhkan untuk menemukan kata sandi dari file-2.zip juga hanya membutuhkan waktu kurang dari 1 detik. Hal ini disebabkan karena kata sandi yang digunakan disini hanya terdiri dari 2 karakter dan tidak menggunakan kombinasi yang terlalu rumit, sehingga tidak akan memakan waktu yang terlalu lama untuk memecahkan kata sandinya.

c. Repeat the procedure and recover the password of the **file-3.zip** file. Recall, that a three-character password was used to encrypt the file. Time to see how long it takes to discover a 3-letter password. Use the following **fcrackzip** command:

```
[analyst@secOps Zip-Files]$ fcrackzip -vul 1-4 file-3.zip
found file 'sample-1.txt', (size cp/uc  39/  27, flags 9, chk 5754) found
file 'sample-2.txt', (size cp/uc  39/  27, flags 9, chk 5756) found file
'sample-3.txt', (size cp/uc  39/  27, flags 9, chk 5757)


PASSWORD FOUND!!!!: pw == 0B1
```

```
[analyst@secOps Zip-Files]$ fcrackzip -vul 1-4 file-3.zip
found file 'sample-1.txt', (size cp/uc     39/    27, flags 9, chk 0558)
found file 'sample-2.txt', (size cp/uc     39/    27, flags 9, chk 0562)
found file 'sample-3.txt', (size cp/uc     39/    27, flags 9, chk 0569)


PASSWORD FOUND!!!!: pw == 0B1
```

How long does it take to discover the password?

= Durasi waktu yang dibutuhkan untuk menemukan kata sandi dari file-3.zip hanya membutuhkan waktu sekitar 1-2 menit saja. Hal ini disebabkan karena kata sandi yang digunakan disini hanya terdiri dari 3 karakter saja dan tidak menggunakan kombinasi karakter yang terlalu rumit (hanya angka dan huruf), sehingga tidak akan memakan waktu yang terlalu lama untuk memecahkan kata sandinya.

d. How long does it take to crack a password of four characters? Repeat the procedure and recover the password of the **file-4.zip** file. Time to see how long it takes to discover the password using the following **fcrackzip** command:

```
[analyst@secOps Zip-Files]$ fcrackzip -vul 1-4 file-4.zip
found file 'sample-1.txt', (size cp/uc  39/  27, flags 9, chk 5754) found
file 'sample-2.txt', (size cp/uc  39/  27, flags 9, chk 5756) found file
'sample-3.txt', (size cp/uc 39/ 27, flags 9, chk 5757) checking pw X9M~

PASSWORD FOUND!!!!: pw == Y0Da
```

```
[analyst@secOps Zip-Files]$ fcrackzip -vul 1-4 file-4.zip
found file 'sample-1.txt', (size cp/uc     39/    27, flags 9, chk 0558)
found file 'sample-2.txt', (size cp/uc     39/    27, flags 9, chk 0562)
found file 'sample-3.txt', (size cp/uc     39/    27, flags 9, chk 0569)
checking pw X9M~

PASSWORD FOUND!!!!: pw == Y0Da
```

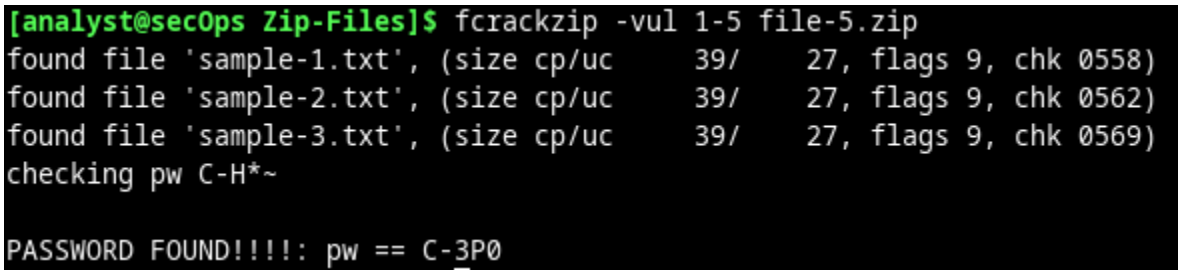
How long does it take to discover the password?

= Sama seperti file sebelumnya, durasi waktu yang dibutuhkan untuk menemukan kata sandi dari file-4.zip juga hanya membutuhkan waktu sekitar 1-2 menit saja. Hal ini disebabkan karena kata sandi yang digunakan disini hanya terdiri dari 4 karakter dan tidak menggunakan kombinasi karakter yang terlalu rumit, sehingga tidak akan memakan waktu yang terlalu lama untuk memecahkan kata sandinya.

e. How long does it take to crack a password of five characters? Repeat the procedure and recover the password of the **file-5.zip** file. The password length is five characters, so we need to set the **-l** command option to **1-5**. Again, time to see how long it takes to discover the password using the following **fcrackzip** command:

```
[analyst@secOps Zip-Files]$ fcrackzip -vul 1-5 file-5.zip
found file 'sample-1.txt', (size cp/uc  39/  27, flags 9, chk 5754) found
file 'sample-2.txt', (size cp/uc  39/  27, flags 9, chk 5756) found file
'sample-3.txt', (size cp/uc 39/ 27, flags 9, chk 5757) checking pw C-H*~


PASSWORD FOUND!!!!: pw == C-3P0
```



How long does it take to discover the password?

= Durasi waktu yang dibutuhkan untuk menemukan kata sandi dari file-5.zip hanya membutuhkan waktu sekitar 2-3 menit saja. Hal ini disebabkan karena kata sandi yang digunakan disini hanya terdiri dari 5 karakter dan tidak menggunakan kombinasi karakter yang terlalu rumit, sehingga tidak akan memakan waktu yang terlalu lama untuk memecahkan kata sandinya.

f. Recover a 6 Character Password using fcrackzip

It appears that longer passwords take more time to discover and therefore, they are more secure. However, a 6 character password would not deter a cybercriminal.

How long do you think it would take fcrackzip to discover a 6-character password?
= Waktu yang diperlukan oleh fcrackzip untuk menemukan sandi dengan 6 karakter bisa bervariasi tergantung pada berbagai faktor seperti kecepatan CPU, maupun kompleksitas sandi. Jika di dalam 6 karakter digit kata sandi tersebut menggunakan kombinasi karakter yang bervariasi, maka kata sandi tersebut dapat ditemukan dalam kurun waktu yang cukup lama (hingga beberapa jam).

To answer that question, create a file called **file-6.zip** using a 6-character password of your choice. In our example, we used **JarJar**.

```
[analyst@secOps Zip-Files]$ zip -e file-6.zip sample*
```

```
[analyst@secOps Zip-Files]$ zip -e file-6.zip sample*
Enter password:
Verify password:
  adding: sample-1.txt (stored 0%)
  adding: sample-2.txt (stored 0%)
  adding: sample-3.txt (stored 0%)
```

g. Repeat the procedure to recover the password of the **file-6.zip** file using the following **fcrackzip** command:

[analyst@secOps Zip-Files]$ **fcrackzip –vul 1-6 file-6.zip**

- ***(mulai discover password)***

21:42
01/05/2024

```
[analyst@secOps Zip-Files]$ fcrackzip -vul 1-6 file-6.zip
found file 'sample-1.txt', (size cp/uc     39/    27, flags 9, chk 0558)
found file 'sample-2.txt', (size cp/uc     39/    27, flags 9, chk 0562)
found file 'sample-3.txt', (size cp/uc     39/    27, flags 9, chk 0569)
checking pw eGUM~
```

- ***(password ditemukan)***

23:02
01/05/2024

```
[analyst@secOps Zip-Files]$ fcrackzip -vul 1-6 file-6.zip
found file 'sample-1.txt', (size cp/uc     39/    27, flags 9, chk 0558)
found file 'sample-2.txt', (size cp/uc     39/    27, flags 9, chk 0562)
found file 'sample-3.txt', (size cp/uc     39/    27, flags 9, chk 0569)
sh: -c: line 0: unexpected EOF while looking for matching `"'
sh: -c: line 1: syntax error: unexpected end of file
sh: -c: line 0: unexpected EOF while looking for matching `"'
sh: -c: line 1: syntax error: unexpected end of file
sh: -c: line 0: unexpected EOF while looking for matching `"'
sh: -c: line 1: syntax error: unexpected end of file
sh: -c: line 0: unexpected EOF while looking for matching `"'
sh: -c: line 1: syntax error: unexpected end of file
sh: -c: line 0: unexpected EOF while looking for matching `"'
sh: -c: line 1: syntax error: unexpected end of file
sh: -c: line 0: unexpected EOF while looking for matching `"'
sh: -c: line 1: syntax error: unexpected end of file
sh: -c: line 0: unexpected EOF while looking for matching `"'
sh: -c: line 1: syntax error: unexpected end of file
sh: -c: line 0: unexpected EOF while looking for matching `"'
sh: -c: line 1: syntax error: unexpected end of file
sh: -c: line 0: unexpected EOF while looking for matching `"'
sh: -c: line 1: syntax error: unexpected end of file
sh: -c: line 0: unexpected EOF while looking for matching `"'
sh: -c: line 1: syntax error: unexpected end of file
sh: -c: line 0: unexpected EOF while looking for matching `"'
sh: -c: line 1: syntax error: unexpected end of file
checking pw Jarc6~

PASSWORD FOUND!!!!: pw == JarJar
```


How long does it take fcrackzip to discover the password?
= Durasi waktu yang dibutuhkan untuk menemukan kata sandi dari file-6.zip jauh lebih lama dari file-file sebelumnya. Hal ini disebabkan karena pada file ini kita menggunakan kata sandi yang lebih panjang yaitu sebanyak 6 karakter (terdiri dari uppercase dan lowercase), sehingga waktu untuk menemukan kata sandinya juga otomatis menjadi lebih lama tergantung pada jumlah dan jenis karakter yang digunakan. Pada kasus ini, saya menggunakan kata sandi sesuai dengan yang diperintahkan di modul yaitu “JarJar” dan durasi yang dibutuhkan untuk menemukan kata sandi ini adalah selama kurang lebih 1 jam 30 menit mulai dari jam 21.42 hingga jam 23.02.

The simple truth is that longer passwords are more secure because they take longer to discover.
How long would you recommend a password needs to be for it to be secure?
= Agar kata sandi dianggap aman dan dapat membantu mengamankan file yang terenkripsi, maka kita perlu menggunakan kata sandi yang lebih panjang. Hal ini disebabkan karena kata sandi yang lebih panjang membutuhkan waktu lebih lama untuk ditemukan. Kata sandi disarankan memiliki panjang minimal setidaknya 12 karakter. Selain itu, juga disarankan untuk mempertimbangkan kompleksitas kata sandi dengan menggunakan kombinasi karakter yang unik, seperti huruf besar/kecil, angka, maupun simbol untuk membantu mengurangi resiko pembobolan kata sandi oleh penyerang. Semakin panjang dan kompleks kata sandi yang digunakan, maka akan semakin baik pula.