



[CTF competition]

Name
Abdullah Saleh Ahmed Qanaan

Table of Contents

1.0 The Selected Question “Johnny Johnny Yes Papa”	3
1.1 Challenge Details	3
2.0 Write-Up: Step by Step	3
2.1 Step 1: Understanding the Challenge.....	3
2.1.1 What is Cracking?	3
2.2 Step 2: Download the ZIP File	4
2.3 Step 3: Move John’s file to Linux.....	4
2.4 Step 4: Navigating to the File in Linux.....	5
2.5 Step 5: Trying to Unzip the file	6
2.6 Step 6: Extracting the ZIP File Hash (zip2john)	6
2.6.1 What exactly does zip2john do?.....	6
2.6.2 The Command Used:	6
2.7 Step 7: Viewing hash.txt	7
2.8 Step 8: Cracking the Hash Using a Wordlist	7
2.9 Step 9: Displaying the Cracked Password.....	8
2.10 Step 10: Unzipping the File Using the Cracked Password	9
2.11 Step 11: Find & Submit the Flag	9
3.0 Alternative Solution to Get the Flag	10
3.1 What is fcrackzip Tool?.....	10
3.2 Step 1: Installing fcrackzip.....	10
3.3 Step 2: Running fcrackzip with a Wordlist	11
3.4 Step 3: Installing and Extracting the rockyou.txt Wordlist.....	11
3.5 Step 4: Cracking the ZIP File Password.....	12
3.6 Step 5: Unzipping the File with the Cracked Password	12
3.7 Step 6: Viewing the Flag	13
4.0 Tools / commands / scripts used.....	13
4.1 Tools Used:	13
4.2 Commands Used:	15

1.0 The Selected Question “Johnny Johnny Yes Papa”

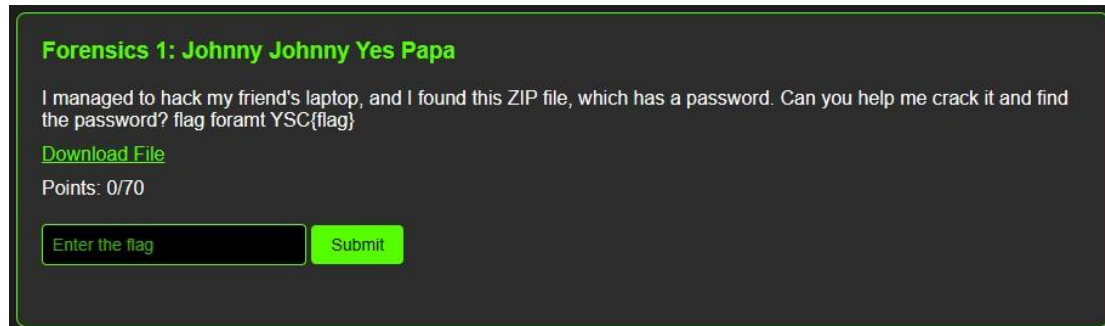


Figure 1

1.1 Challenge Details

Description:

The challenge “Johnny Johnny Yes Papa” is a forensics-based task that evaluates the candidate’s skill to analyze and extract data from a password-protected ZIP file. The challenge emulates an actual forensic investigation where an Investigator must access locked or encrypted data utilizing forensics, password-cracking methods, and investigative methods.

The participants are given a ZIP archive that holds a flag.txt file. The problem is that the ZIP file is password-protected, which means that the participant cannot access the flag. The objective is to recover the password, unlock the ZIP file, and extract the hidden flag. The challenge requires knowledge in cracking passwords, extracting hashes, and employing cybersecurity tools like zip2john, John the Ripper, fcrackzip, and Hashcat.

- Category: Forensics type
- Challenge Name: “Johnny Johnny Yes Papa”
- Difficulty Level: Hard (70 points)
- Hints: No hints for this challenge

2.0 Write-Up: Step by Step

2.1 Step 1: Understanding the Challenge

The title of the challenge is "Johnny Johnny Yes Papa" which is a forensics-based task. From the description, we can gather that the provided ZIP file is password-protected, and the goal is to crack the password to gain access to its contents.

2.1.1 What is Cracking?

Cracking is similar to trying to open a locked box without the key. In this scenario, the password serves as the "key" and the ZIP file is considered as the "box". Cracking includes utilizing methods or tools to guess or decipher the password in order to unlock and access the files inside; this is known as cracking. A

password cracker uses many methods to recover the password. An algorithm tool may be used to guess the password repeatedly or comparing a list of words to guess the right password. It's a bit like solving a puzzle to find the right combination (Maury, 2023).

2.2 Step 2: Download the ZIP File

First, we download the file and open the John's folder. Inside the folder, there is a file named "flag", after that, we immediately tried to click on the file to see the flag. And all this without extracting the file.

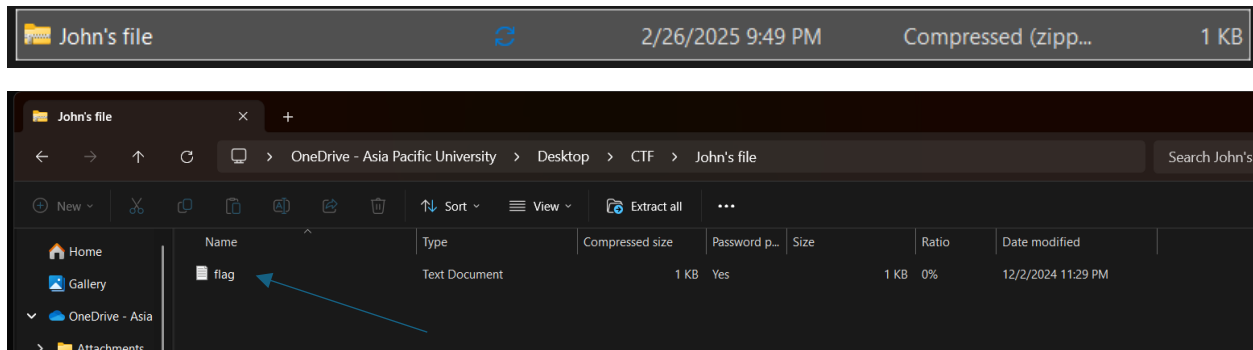


Figure 2

After we clicked on the file, we were immediately prompted with a password requirement to proceed with opening the file. The prompt displayed the following message as the figure below shows:

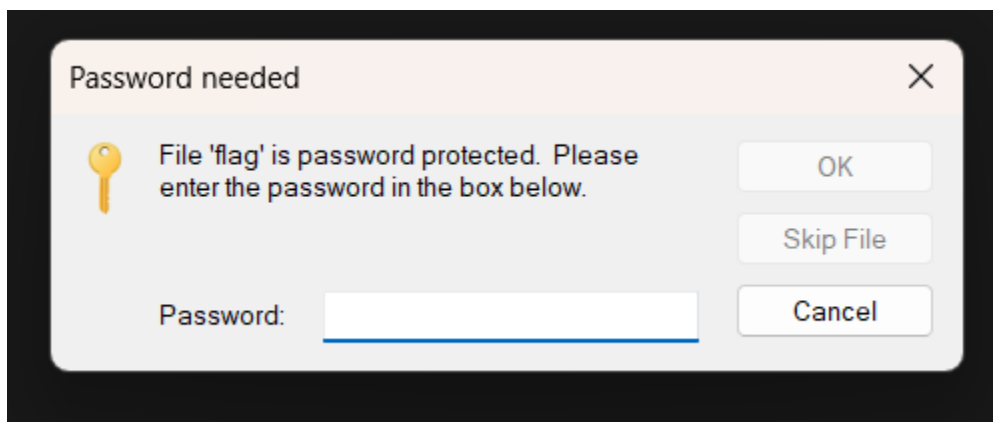


Figure 3

Without knowing the right password at this point, we understood that before proceeding any further, we needed to use forensic tools to recover or crack the password.

2.3 Step 3: Move John's file to Linux

We just need to drag the folder from our Windows to Kali Linux

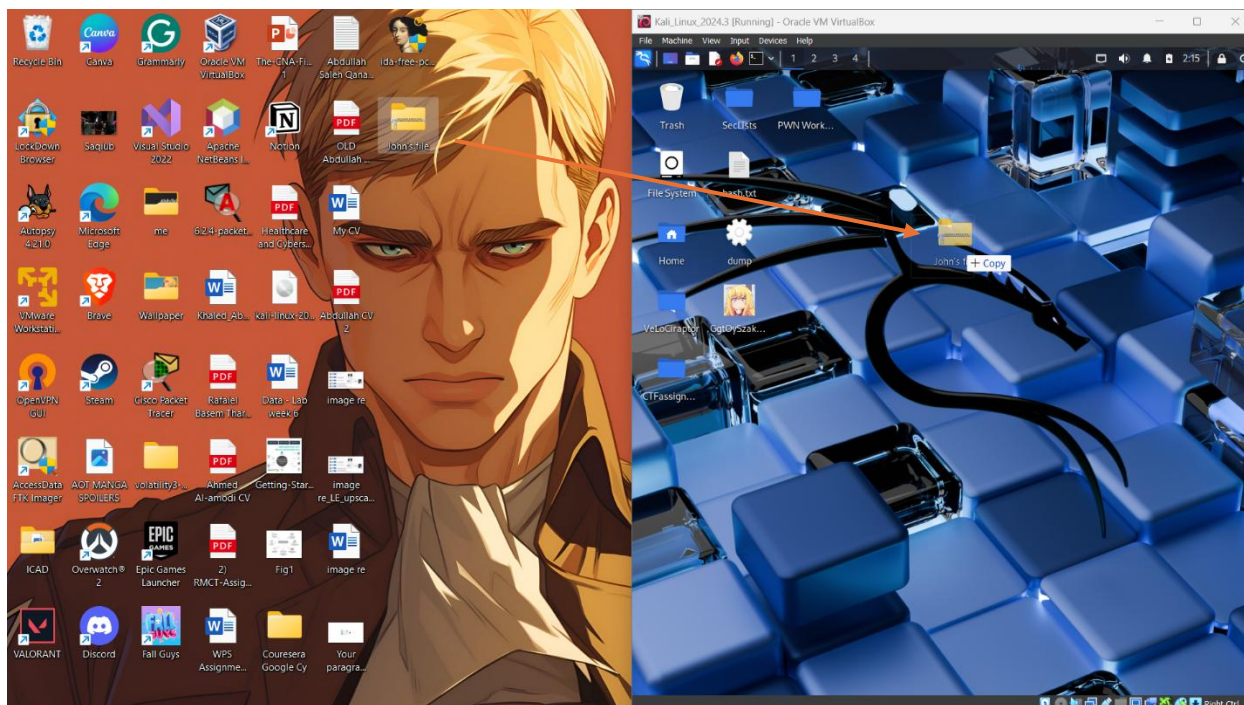


Figure 4

2.4 Step 4: Navigating to the File in Linux

Once the.zip file has been downloaded and moved to Linux, we need to open the terminal and go to the directory or folder containing the John's file by using the command "cd" with the path location. Type the command "ls" and hit enter to display a list of every file in that directory. This will show the directory's contents, and John's file.zip ought to be visible as the figure below displays:

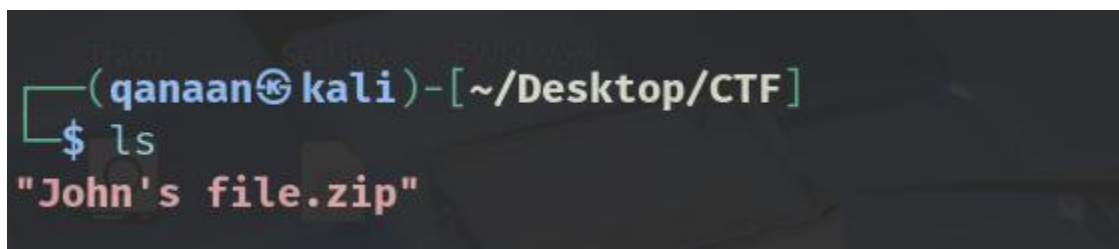


Figure 5

2.5 Step 5: Trying to Unzip the file

```
(qanaan@kali)-[~/Desktop/CTF]
$ unzip John\'s\ file.zip
Archive:  John's file.zip
[John's file.zip] flag.txt password:
password incorrect--reenter:
password incorrect--reenter:
      skipping: flag.txt                incorrect password
```

Figure 6

No matter what password we try, the system will display "Incorrect password" when we try to unzip the file unless we enter the exact right password. This means that in order to successfully unlock the file, we have to find the correct password.

2.6 Step 6: Extracting the ZIP File Hash (zip2john)

```
(qanaan@kali)-[~/Desktop/CTF]
$ zip2john John\'s\ file.zip > hash.txt
ver 1.0 efh 5455 efh 7875 John's file.zip/flag.txt PKZIP Encr: 2b chk, TS_chk,
```

Figure 7

2.6.1 What exactly does zip2john do?

Now zip2john helps us to “extract” the password from a ZIP archive. Let me explain: when a ZIP file gets locked with a password, the password gets encrypted and stored somewhere else. zip2john takes that information and saves it into a new file called (“hash”) so we can try to figure out the password later.

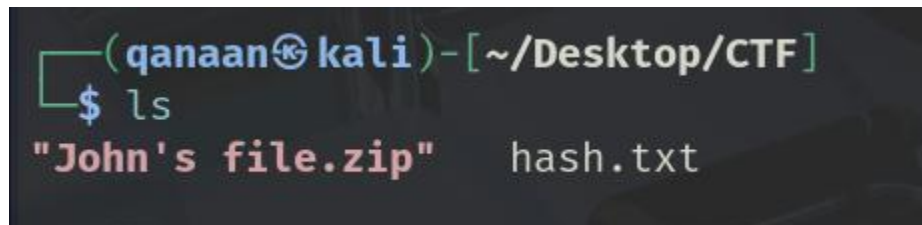
2.6.2 The Command Used:

zip2john john's\ file.zip > hash.txt

- **zip2john:** This is the tool we are using to extract the password details from the ZIP file.
- **john's\ file.zip:** This is the name of the ZIP file we want unlocked. (The \ is for spaces in the file name)
- **>:** This symbol instructs the computer to save the output (the extracted password details) in a new file.
- **hash.txt:** This is the new file name to which the details of the password will be saved as “hash.”

In simple terms, this command takes the locked ZIP file, extracts its password information, and saves it into hash.txt so we can work on cracking the password later.

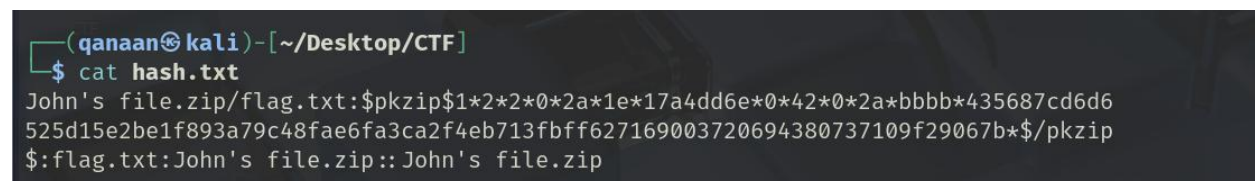
2.7 Step 7: Viewing hash.txt



```
(qanaan@kali)-[~/Desktop/CTF]
$ ls
"John's file.zip"  hash.txt
```

Figure 8

After we used the command “zip2john” a text file was created with the name “hash.txt” As we can see in the figure above

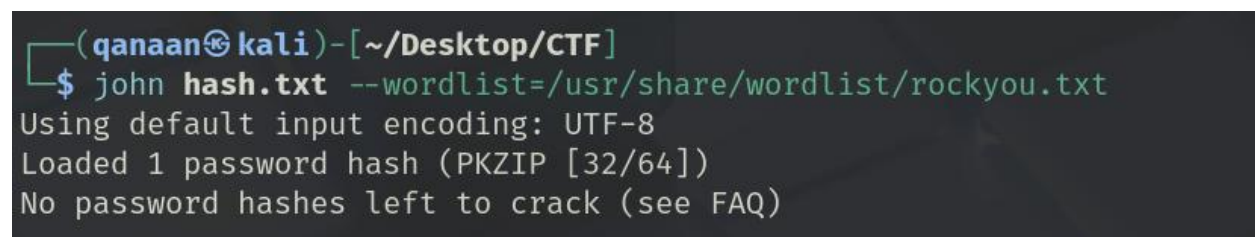


```
(qanaan@kali)-[~/Desktop/CTF]
$ cat hash.txt
John's file.zip/flag.txt:$pkzip$1*2*2*0*2a*1e*17a4dd6e*0*42*0*2a*bbbb*435687cd6d6
525d15e2be1f893a79c48fae6fa3ca2f4eb713fbff627169003720694380737109f29067b*$/$/pkzip
$:flag.txt:John's file.zip:: John's file.zip
```

Figure 9

We can view the contents of the hash.txt file after executing the command cat hash.txt. But the data within will appear as a "hash," which is a random combination of characters and numbers. Since it's a jumbled version of the password rather than the real one, it won't make sense to us. We must "decode" or crack this hash using a different tool or command in order get the real password. To determine the original password, this process involves trying many of different combinations or utilizing a password-cracking tool.

2.8 Step 8: Cracking the Hash Using a Wordlist



```
(qanaan@kali)-[~/Desktop/CTF]
$ john hash.txt --wordlist=/usr/share/wordlist/rockyou.txt
Using default input encoding: UTF-8
Loaded 1 password hash (PKZIP [32/64])
No password hashes left to crack (see FAQ)
```

Figure 10

Here we use the command:

`john hash.txt --wordlist=/usr/share/wordlists/rockyou.txt`

What Does this command do?

John the Ripper (or john) tool is configured in a way that it attempts to crack the password that is given the hash. Here is what happens step by step:

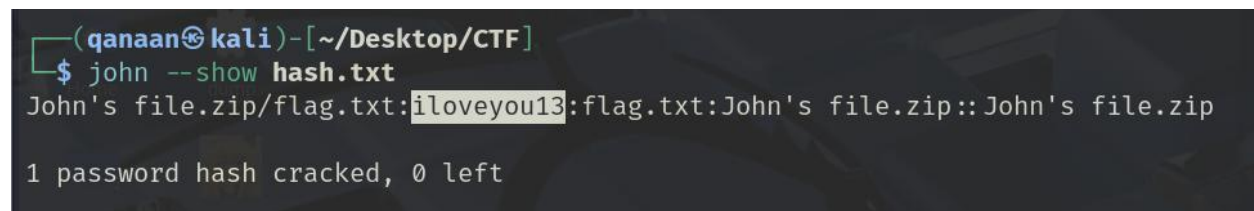
1. john: This is the tool we have chosen to crack passwords.
2. hash.txt: This is the file where we kept the password that we have scrambled (the hash) that we extracted earlier.
3. `--wordlist=/usr/share/wordlists/rockyou.txt`:
 - As a wordlist, we can consider it as password dictionary.
 - The rockyou.txt file is a very popular wordlist that contains millions of common passwords people use.
 - The tool will try to go through this word list and use each password, one by one, to see if it matches the hash.

To put it in another way:

This command asks the computer: "Start using the john tool to try to unlock hash.txt with the passwords present in the rockyou.txt wordlist."

As long as the rockyou.txt list has the password, we are guaranteed that the tool will find it and display it to us!

2.9 Step 9: Displaying the Cracked Password

A terminal window with a dark background. The prompt is `(qanaan@kali) - [~/Desktop/CTF]`. The user enters `$ john --show hash.txt`. The output is `John's file.zip/flag.txt:iloveyou13:flag.txt:John's file.zip::John's file.zip` followed by `1 password hash cracked, 0 left` on a new line.

```
(qanaan@kali) - [~/Desktop/CTF]
$ john --show hash.txt
John's file.zip/flag.txt:iloveyou13:flag.txt:John's file.zip::John's file.zip
1 password hash cracked, 0 left
```

Figure 11

As the figure above displays the command `john --show hash.txt`

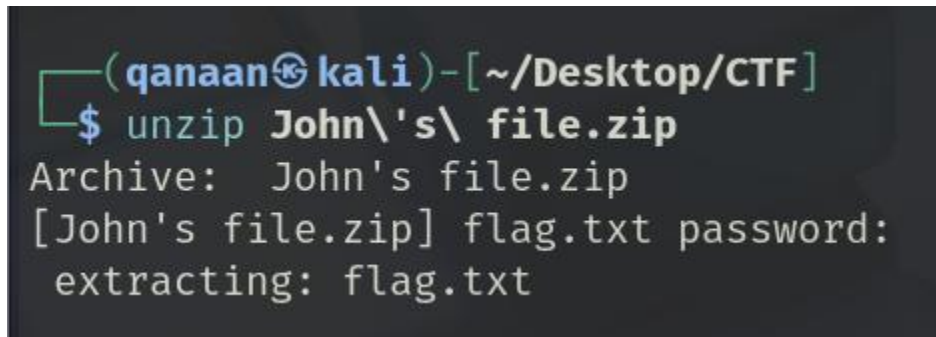
What Does It Do?

This command instructs the John tool to show any passwords from the hash.txt file that it has successfully cracked. John attempted passwords from the **rockyou.txt** wordlist using the previous command, and this command will display the outcomes.

The result:

The tool indicates that the password for the ZIP file is **"iloveyou13"** when we execute `john --show hash.txt`. We can now try to unlock the ZIP file and view its contents using this password.

2.10 Step 10: Unzipping the File Using the Cracked Password



```
(qanaan@kali)-[~/Desktop/CTF]
$ unzip John\'s\ file.zip
Archive:  John's file.zip
[John's file.zip] flag.txt password:
extracting: flag.txt
```

Figure 12

At this stage, we select the zip file to be extracted and enter the password 'iloveyou13', which we retrieved earlier. After entering the correct password, the system begins to extract the contents within the ZIP file.

The system showing this message indicate successful extraction of file **flag.txt** as you can see in the figure below.

This means that the ZIP file has been successfully extracted and we can now open the file's content.



```
(qanaan@kali)-[~/Desktop/CTF]
$ ls
"John's file.zip"  flag.txt  hash.txt
```

Figure 13

2.11 Step 11: Find & Submit the Flag

To view the contents of the extracted flag.txt file, we use the command "cat" in linux.

Simply type: "**cat flag.txt**" This will display the contents of the file on the screen. Once we see the flag "YSC{Zip2JoHn_ls_cr3CkIng_t0oL}", we can submit it to complete the challenge.

```
(qanaan@kali)-[~/Desktop/CTF]
$ cat flag.txt
YSC{ZiP2JoHn_ls_cr3CkIng_t0oL}
```

Figure 14

Here we can see the flag we got is correct as the figure below shows.

Forensics 1: Johnny Johnny Yes Papa

I managed to hack my friend's laptop, and I found this ZIP file, which has a password. Can you help me crack it and find the password? flag format YSC{flag}

[Download File](#)

Points: 70/70

Correct! Well done!

Figure 15

3.0 Alternative Solution to Get the Flag

3.1 What is fcrackzip Tool?

Fcrackzip: This is a quick password cracker tool that was created in part using an assembler. It can use dictionary based or brute force attacks to crack password-protected zip files, and it can optionally use unzip to confirm the results. Fcrackzip attempts to guess the password by searching each zip file for encryption keys. Every file needs to be encrypted with a password, and the more files you have, the better.

3.2 Step 1: Installing fcrackzip

The first step in our approach was to install **fcrackzip** by using the command "sudo apt install fcrackzip". This command installs the set package using the official repository. After installation, we were able to proceed on cracking the passwords.

```
(qanaan@kali)-[~/Desktop/CTF]
$ sudo apt install fcrackzip
[sudo] password for qanaan:
Installing:
fcrackzip
```

Figure 16

3.3 Step 2: Running fcrackzip with a Wordlist

```
(qanaan@kali)-[~/Desktop/CTF]
$ fcrackzip -v -u -D -p /usr/share/wordlists/rockyou.txt John's\ file.zip
found file 'flag.txt', (size cp/uc 42/ 30, flags 9, chk bbbb)
/usr/share/wordlists/rockyou.txt: No such file or directory
```

Figure 17

We tried using the popular rockyou.txt wordlist to crack the ZIP file after installing fcrackzip. This wordlist is a useful tool for brute-force attacks because it includes a vast collection of frequently used passwords. But we ran into a problem that said **rockyou.txt** did not exist in the system.

- -v: Enables verbose mode, allowing us to see the process clearly.
- -u: Ensures that fcrackzip only checks for valid passwords.
- -D: Specifies dictionary attack mode.
- -p /usr/share/wordlists/rockyou.txt: Uses rockyou.txt as the password dictionary.

3.4 Step 3: Installing and Extracting the rockyou.txt Wordlist

We had to manually install the wordlist in order to fix the missing issue. We used the following command to install the rockyou.txt wordlist, which is a component of the wordlists package as figure below displays:

```
(qanaan@kali)-[~/Desktop/CTF]
$ sudo apt install wordlists
wordlists is already the newest version (2023.2.0).
wordlists set to manually installed.
Summary:
  Upgrading: 0, Installing: 0, Removing: 0, Not Upgrading: 2043
```

Figure 18

After installing, we noticed that rockyou.txt was compressed as rockyou.txt.gz. This extracted the file, making it ready for use.

```
(qanaan@kali)-[~/Desktop/CTF]
$ sudo gzip -d /usr/share/wordlists/rockyou.txt.gz
```

Figure 19

3.5 Step 4: Cracking the ZIP File Password

We used the same command to run fcrackzip again after the wordlist had been installed correctly. This time, the password for the ZIP file was successfully cracked and as we can see in the figure below, the password "iloveyou13" was found in the wordlist and matched the ZIP file's encryption.

```
(qanaan@kali)-[~/Desktop/CTF]
$ fcrackzip -v -u -D -p /usr/share/wordlists/rockyou.txt John's\ file.zip
found file 'flag.txt', (size cp/uc 42/ 30, flags 9, chk bbbb)

PASSWORD FOUND!!!!: pw = iloveyou13
```

Figure 20

3.6 Step 5: Unzipping the File with the Cracked Password

Now after we cracked the password, we proceeded to extract the contents of the ZIP file. Entering the password "iloveyou13", the ZIP file extracted successfully, revealing a **flag.txt** file.

```
PWIN Work ...  
[qanaan@kali] - [~/Desktop/CTF]  
$ unzip John\'s\ file.zip  
Archive: John's file.zip  
[John's file.zip] flag.txt password:  
extracting: flag.txt
```

Figure 21

3.7 Step 6: Viewing the Flag

Lastly, we only need to use the command “cat” on the file flag.txt to display the flag. We had successfully completed the challenge and retrieved the flag.

```
[qanaan@kali] - [~/Desktop/CTF]  
$ cat flag.txt  
YSC{ZiP2JoHn_ls_cr3CkIng_t0oL}
```

Figure 22

4.0 Tools / commands / scripts used

4.1 Tools Used:



Figure 23 Kali Linux

Kali Linux

Purpose

- The primary operating system used for penetration testing.
- Includes pre-installed security tools such as **John the Ripper** and **zip2john**.

Usage

- The entire challenge was conducted in a Kali Linux environment
- Used for running password-cracking commands and handling files efficiently

zip2john

Purpose

- Extracts the hash from a password-protected ZIP file, allowing password cracking using **John the Ripper**.
- Converts ZIP password hashes into a format readable by password-cracking tools.

Usage

- Used to extract the encrypted password hash from the ZIP file.
- The extracted hash was then stored in a file called **hash.txt**.



Figure 24 John the Ripper

John the Ripper

Purpose

- A powerful password-cracking tool used to brute-force or perform dictionary attacks on password hashes.
- Can handle various hash types, including ZIP, MD5, SHA-1, and more.

Usage

- Used to crack the password extracted by **zip2john** using a wordlist attack.
- The output revealed the correct password needed to unlock the ZIP file.
- Allows brute-force and dictionary attacks against password hashes.
- Efficiently cracks weak passwords using pre-compiled wordlists.

rockyou.txt (wordlist)

Purpose

- A famous wordlist containing millions of real-world passwords from leaked databases.
- Used in dictionary attacks to guess weak passwords.

Usage

- Provided as input to **John the Ripper** to systematically try passwords until the correct one was found.
- Increases the success rate of password cracking
- Many users choose weak, commonly used passwords that are present in wordlists like **rockyou.txt**

Add table of summary tools used comparism should be 3 columns analysis also included.

4.2 Commands Used:

1. Navigating in Linux Terminal

```
cd /path/to/file  
ls
```

2. Extracting ZIP File Hash using zip2john

```
zip2john john's\ file.zip > hash.txt
```

3. Viewing the Hash File

```
cat hash.txt
```

4. Cracking the Hash Using a Wordlist with John the Ripper

```
john hash.txt --wordlist=/usr/share/wordlists/rockyou.txt
```

5. Displaying the Cracked Password

```
john --show hash.txt
```

6. Unzipping the File Using the Cracked Password

```
unzip john's\ file.zip
```

7. Viewing the Flag

```
cat flag.txt
```