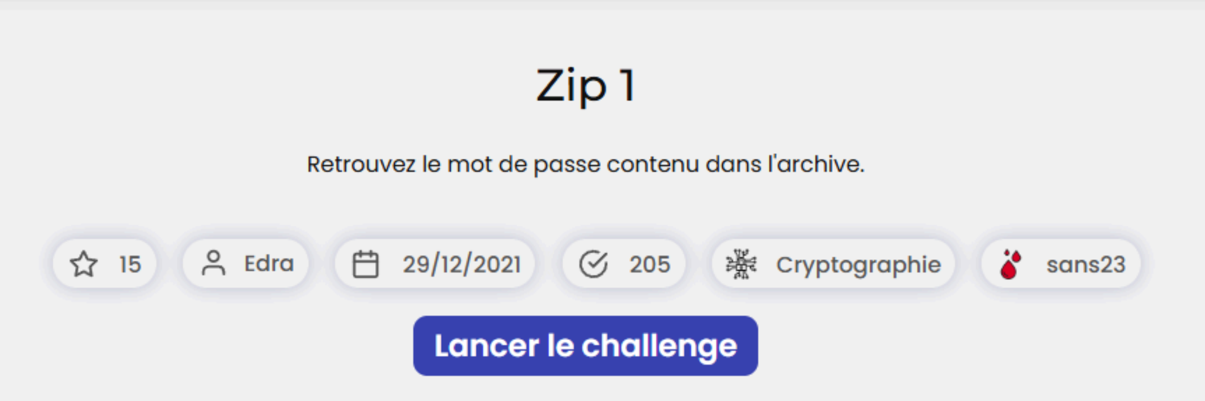
Zip1 CTF Writeup

This document is a walkthrough on one way to solve the **Zip1 CTF** on **TheBlackSide**.
The objective is to explain how I was able to solve this CTF to my future self.

General Information

- *Difficulty:* Medium
- *Category:* Cryptography
- *Link:* [Zip1 - TheBlackSide]()

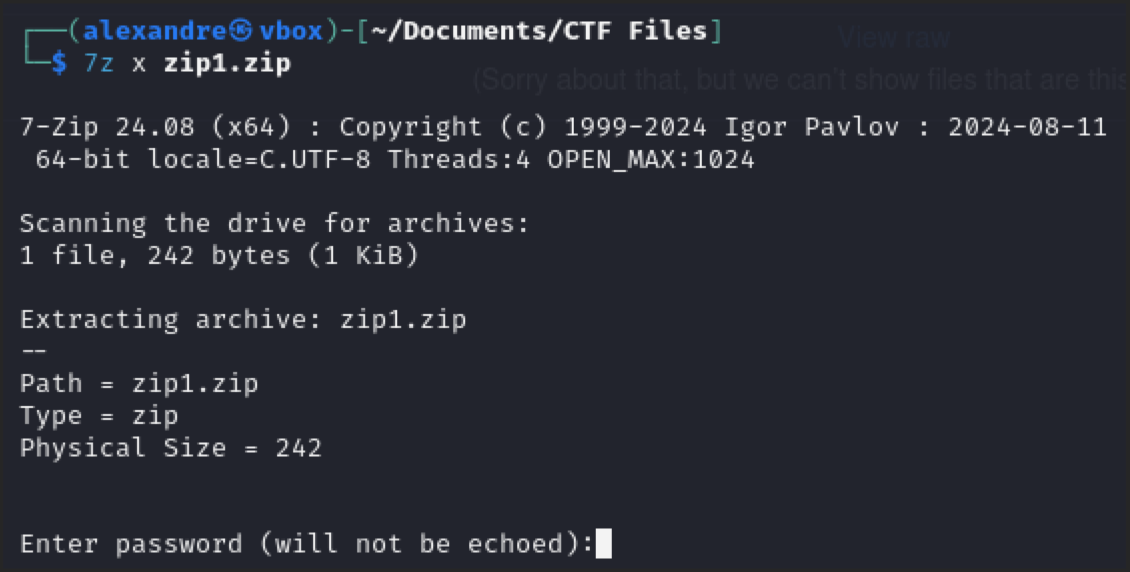
Solution



Zip 1

Retrouvez le mot de passe contenu dans l'archive.

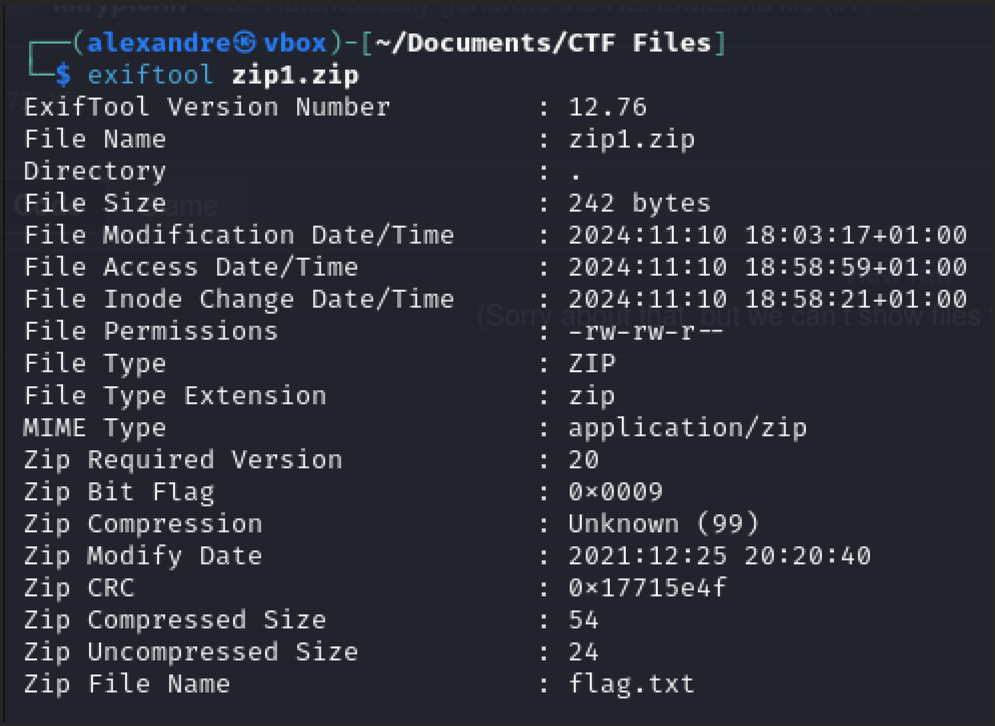
We're given a zip file, and are prompted to find the password to access it.
It doesn't seem to support the **"unzip"** command on Linux, so we'll use **7zip**.



```
┌──(alexandre㉿vbox)-[~/Documents/CTF Files]
└─$ 7z x zip1.zip

7-Zip 24.08 (x64) : Copyright (c) 1999-2024 Igor Pavlov : 2024-08-11
 64-bit locale=C.UTF-8 Threads:4 OPEN_MAX:1024

Scanning the drive for archives:
1 file, 242 bytes (1 KiB)

Extracting archive: zip1.zip
--
Path = zip1.zip
Type = zip
Physical Size = 242


Enter password (will not be echoed):
```

However, we notice that the zip file is password-protected.

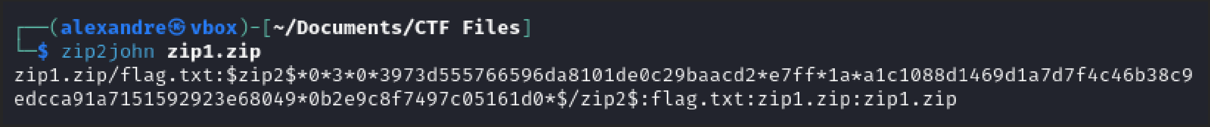
Moreover, it appears that a **“flag.txt”** file is in the archive, so it appears that the main objective of this challenge is to find the password that unlocks access to the file.



```
┌──(alexandre㉿vbox)-[~/Documents/CTF Files]
└─$ exiftool zip1.zip
ExifTool Version Number         : 12.76
File Name                       : zip1.zip
Directory                       : .
File Size                       : 242 bytes
File Modification Date/Time     : 2024:11:10 18:03:17+01:00
File Access Date/Time           : 2024:11:10 18:58:59+01:00
File Inode Change Date/Time     : 2024:11:10 18:58:21+01:00
File Permissions                : -rw-rw-r--
File Type                       : ZIP
File Type Extension             : zip
MIME Type                       : application/zip
Zip Required Version            : 20
Zip Bit Flag                    : 0x0009
Zip Compression                 : Unknown (99)
Zip Modify Date                 : 2021:12:25 20:20:40
Zip CRC                         : 0x17715e4f
Zip Compressed Size             : 54
Zip Uncompressed Size           : 24
Zip File Name                   : flag.txt
```

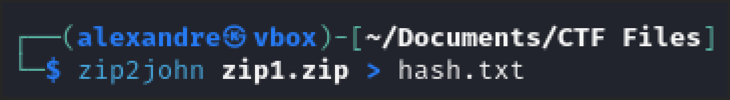
After a bit of searching, we don’t notice any way of getting the password, the zip file is all there is.
With no other options left, it’s time to unleash the ultimate hacking weapon: **BRUTE-FORCE**
In order words, we’re going to **guess** the password.

Luckily, the password is stored in its encrypted format, a hash.
We can get it by using a tool called **John The Ripper:**

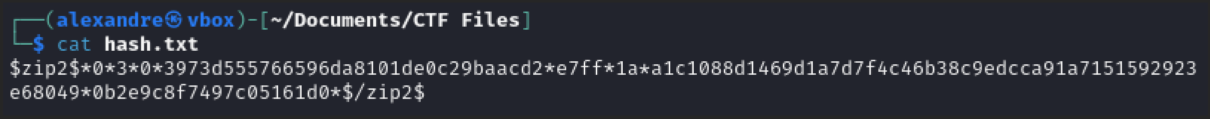


```
┌──(alexandre㉿vbox)-[~/Documents/CTF Files]
└─$ zip2john zip1.zip
zip1.zip/flag.txt:$zip2$*0*3*0*3973d555766596da8101de0c29baacd2*e7ff*1a*a1c1088d1469d1a7d7f4c46b38c9
edcca91a7151592923e68049*0b2e9c8f7497c05161d0*$/zip2$:flag.txt:zip1.zip:zip1.zip
```

We’re going to save that output to a new file **hash.txt,** and then keep only the parts that interest us using the **nano** command to edit the previously-mentioned text file.



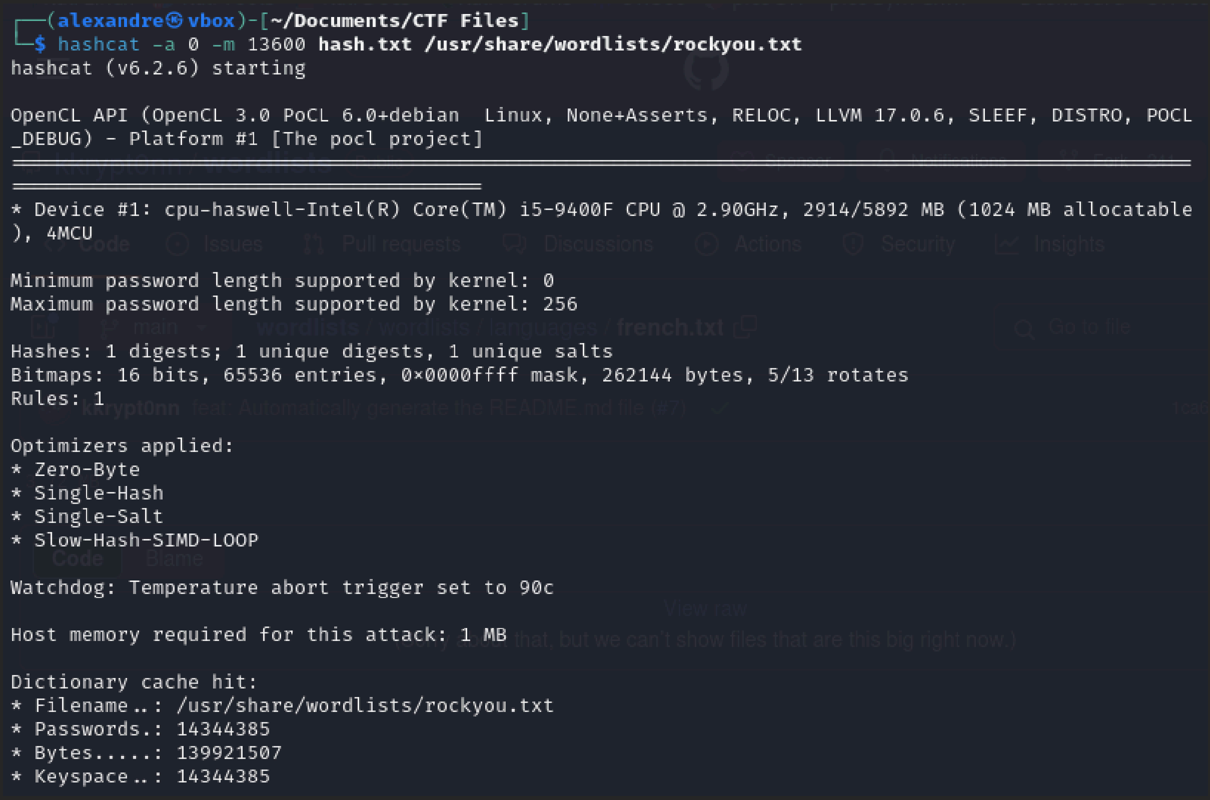
```
┌──(alexandre㉿vbox)-[~/Documents/CTF Files]
└─$ zip2john zip1.zip > hash.txt
```



```
┌──(alexandre㉿vbox)-[~/Documents/CTF Files]
└─$ cat hash.txt
$zip2$*0*3*0*3973d555766596da8101de0c29baacd2*e7ff*1a*a1c1088d1469d1a7d7f4c46b38c9edcca91a7151592923
e68049*0b2e9c8f7497c05161d0*$/zip2$
```

Now that our hash file is ready, it’s time to crack it !

Using a tool called **Hashcat**, notorious for cracking hashes, we'll type the following command:

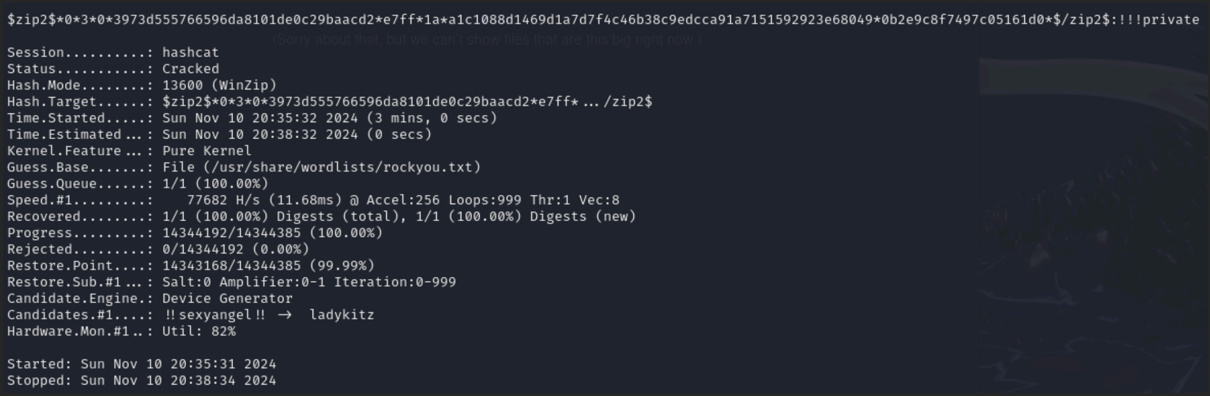


```
┌──(alexandre㉿vbox)-[~/Documents/CTF Files]
└─$ hashcat -a 0 -m 13600 hash.txt /usr/share/wordlists/rockyou.txt
hashcat (v6.2.6) starting

OpenCL API (OpenCL 3.0 PoCL 6.0+debian  Linux, None+Asserts, RELOC, LLVM 17.0.6, SLEEF, DISTRO, POCL
_DEBUG) - Platform #1 [The pocl project]
==================================================================================================
========================================

* Device #1: cpu-haswell-Intel(R) Core(TM) i5-9400F CPU @ 2.90GHz, 2914/5892 MB (1024 MB allocatable
), 4MCU

Minimum password length supported by kernel: 0
Maximum password length supported by kernel: 256

Hashes: 1 digests; 1 unique digests, 1 unique salts
Bitmaps: 16 bits, 65536 entries, 0x0000ffff mask, 262144 bytes, 5/13 rotates
Rules: 1

Optimizers applied:
* Zero-Byte
* Single-Hash
* Single-Salt
* Slow-Hash-SIMD-LOOP

Watchdog: Temperature abort trigger set to 90c

Host memory required for this attack: 1 MB

Dictionary cache hit:
* Filename..: /usr/share/wordlists/rockyou.txt
* Passwords.: 14344385
* Bytes.....: 139921507
* Keyspace..: 14344385
```

-m 13600 is to tell hashcat that the zip file was encrypted with WinZip

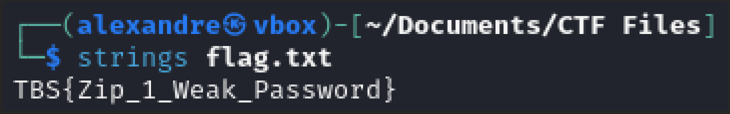
/usr/share/wordlists/rockyou.txt is the location of a text file on our system that Hashcat will use for password cracking. The rockyou.txt file is well-known for its extensive list, containing over 14 million password combinations

After some time (about 3 minutes in my case), hashcat returns this



```
$zip2$*0*3*0*3973d555766596da8101de0c29baacd2*e7ff*1a*a1c1088d1469d1a7d7f4c46b38c9edcca91a7151592923e68049*0b2e9c8f7497c05161d0*$/zip2$:!!!private

Session..........: hashcat
Status...........: Cracked
Hash.Mode........: 13600 (WinZip)
Hash.Target......: $zip2$*0*3*0*3973d555766596da8101de0c29baacd2*e7ff*.../zip2$
Time.Started.....: Sun Nov 10 20:35:32 2024 (3 mins, 0 secs)
Time.Estimated...: Sun Nov 10 20:38:32 2024 (0 secs)
Kernel.Feature...: Pure Kernel
Guess.Base.......: File (/usr/share/wordlists/rockyou.txt)
Guess.Queue......: 1/1 (100.00%)
Speed.#1.........:    77682 H/s (11.68ms) @ Accel:256 Loops:999 Thr:1 Vec:8
Recovered........: 1/1 (100.00%) Digests (total), 1/1 (100.00%) Digests (new)
Progress.........: 14344192/14344385 (100.00%)
Rejected.........: 0/14344192 (0.00%)
Restore.Point....: 14343168/14344385 (99.99%)
Restore.Sub.#1...: Salt:0 Amplifier:0-1 Iteration:0-999
Candidate.Engine.: Device Generator
Candidates.#1....: !!sexyangel!! -> ladykitz
Hardware.Mon.#1..: Util: 82%

Started: Sun Nov 10 20:35:31 2024
Stopped: Sun Nov 10 20:38:34 2024
```

The password is: **!!!private**



```
┌──(alexandre㉿vbox)-[~/Documents/CTF Files]
└─$ strings flag.txt
TBS{Zip_1_Weak_Password}
```

Finally, inputting this as the zip file's password, we're able to extract the flag.txt file and get the following flag: **TBS{Zip_1_Weak_Password}**