Cracking Passwords with Hashcat:

Link to challenge: https://academy.hackthebox.com/module/20

(log in required)

Class: Tier II | Medium | Offensive

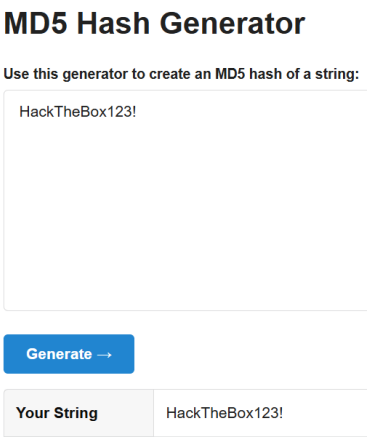
Introduction

Hashing vs. Encryption:

Question: Generate an MD5 hash of the password 'HackTheBox123!'.

Answer: 87946d0585ba62c0671b734cada46b41

Method: https://www.md5hashgenerator.com/

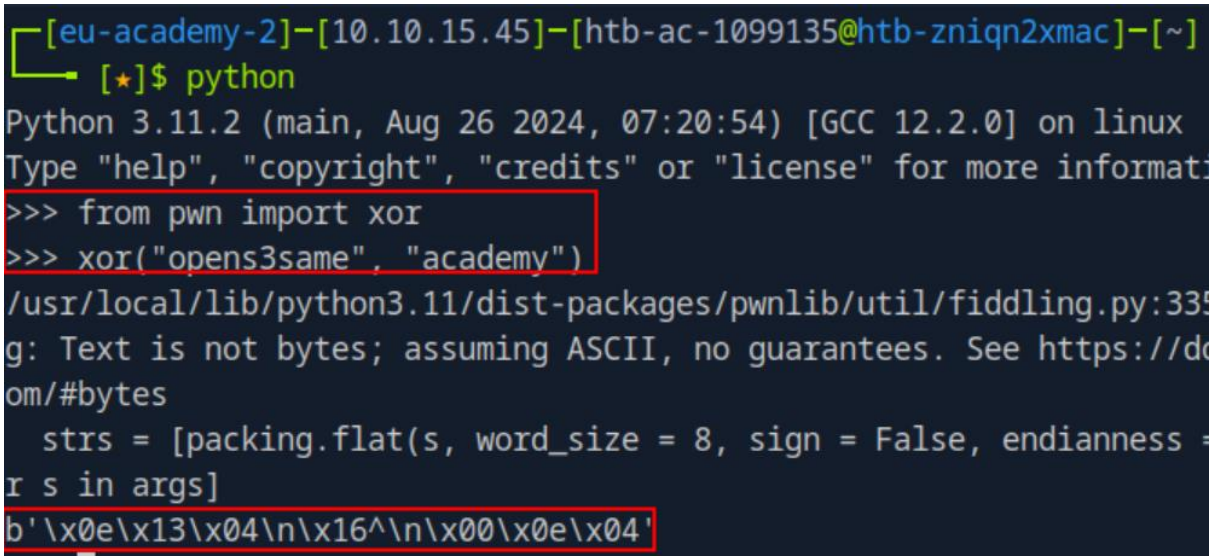


Question: Create the XOR ciphertext of the password 'opens3same' using the key 'academy'. (Answer format: \x00\x00\x00\....)

Answer: \x0e\x13\x04\n\x16^\n\x00\x0e\x04

Method: we use a simple python script:

```
from pwn import xor

xor("opens3same", "academy")
```



Identifying Hashes:

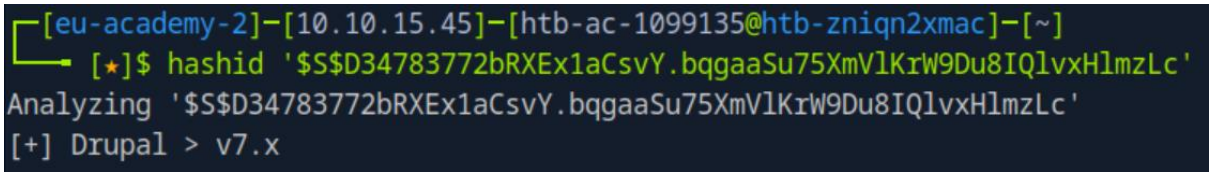
Question: Identify the following hash: SD34783772bRXEx1aCsvY.bqgaaSu75XmVlKrW9Du8IQlvxHlmzLc

Answer: Drupal > v7.x

Method: we use the tool 'hashid':

*note – if needed we install it with pip ('pip install hashid')

```
hashid
'$S$D34783772bRXEx1aCsvY.bqgaaSu75XmVlKrW9Du8IQlvxHlmzLc'
```



Hashcat Overview:

Question: What is the hash mode of the hash type Cisco-ASA MD5?

Answer: 2410

Method: we find it with the command:

```
hashcat --example-hashes | grep 'Cisco-ASA MD5' -B 1
```

to look for examples hashes, but filter for 'Cisco-ASA MD5', and present the line which contains that keyword, and the line above it (-B 1):

Hashcat Attack Types

Dictionary Attack:

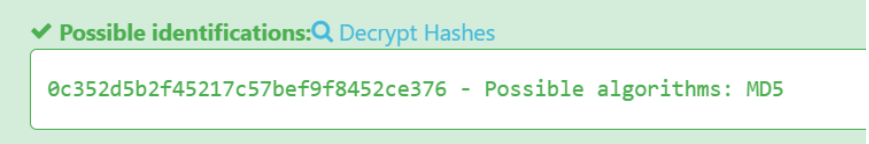
Question: Crack the following hash using the rockyou.txt wordlist:
0c352d5b2f45217c57bef9f8452ce376

Answer: cricket1

Method: First, lets start with downloading the rockyou.txt wordlist to the pwnbox:

```
wget https://github.com/brannondorsey/naive-hashcat/releases/download/data/rockyou.txt -q
```

now, according to this hash analyzer, the hash we need to crack is MD5:



Lets put the hash in a 'hash.txt' file

```
echo 0c352d5b2f45217c57bef9f8452ce376 > hash.txt
```

and begin to crack, to crack unsalted MS5 hash, we use the hashcat code 0:

```
hashcat -m 0 hash.txt rockyou.txt
```

```
┌─[eu-academy-2]─[10.10.15.45]─[htb-ac-1099135@htb-yrvxi8n455]─[~]
└──╼ [★]$ hashcat -m 0 hash.txt rockyou.txt
hashcat (v6.2.6) starting

OpenCL API (OpenCL 3.0 PoCL 3.1+debian  Linux, None+Asserts, RELOC, SPIR, LLVM 1
5.0.6, SLEEF, DISTRO, POCL_DEBUG) - Platform #1 [The pocl project]
```

* *

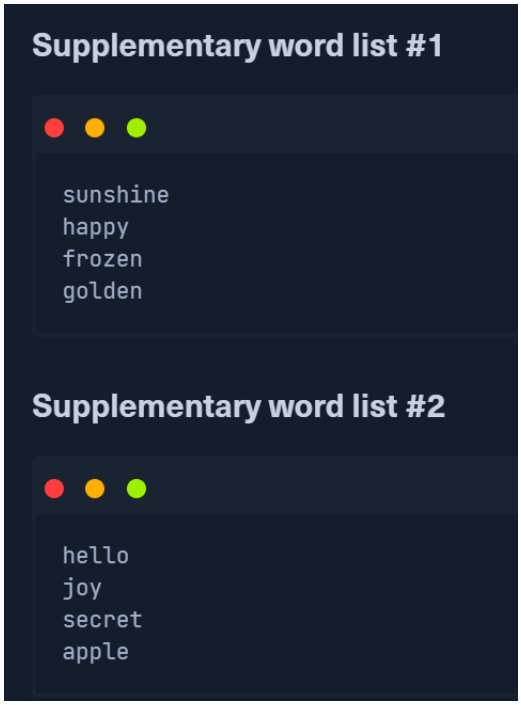
```
Runtime...: 1 sec

0c352d5b2f45217c57bef9f8452ce376:cricket1

Session..........: hashcat
Status...........: Cracked
Hash.Mode........: 0 (MD5)
Hash.Target......: 0c352d5b2f45217c57bef9f8452ce376
Time.Started.....: Sat Nov 16 09:01:10 2024 (0 secs)
```

Combination Attack:

Question: Using the Hashcat combination attack find the cleartext password of the following md5 hash: 19672a3f042ae1b592289f8333bf76c5. Use the supplementary wordlists shown at the end of this section.

Answer: frozenapple

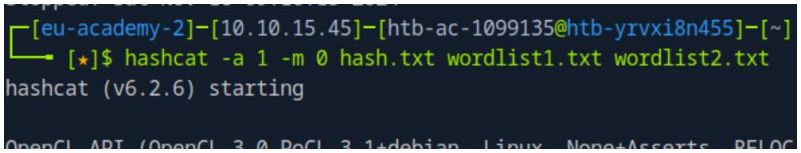
Method: for this question we are provided with 2 wordlists:



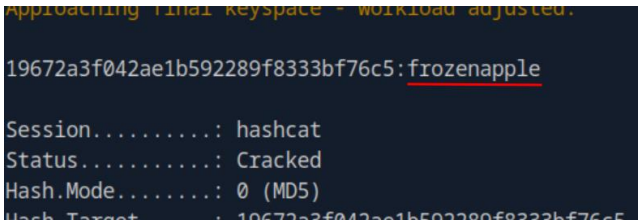
We will put them in 'wordlist1.txt' and 'wordlist2.txt' respectively.

Then we will proceed to replace the old hash with the new hash in 'hash.txt. and then to crack the hash using the combined list using the command:

```
hashcat -a 1 -m 0 hash.txt wordlist1.txt wordlist2.txt
```



* *



Mask Attack:

Question: Crack the following MD5 hash using a mask attack: 50a742905949102c961929823a2e8ca0. Use the following mask: -1 02 'HASHCAT?l?l?l?l?l20?1?d'

Answer: HASHCATqrstu2020

Method: we update the hash in 'hash.txt', and run the command:

```
hashcat -a 3 -m 0 hash.txt HASHCAT?l?l?l?l?l20?1?d -1 02
```

```
┌─[eu-academy-2]─[10.10.15.45]─[htb-ac-1099135@htb-yrvxi8n455]─[~]
└──╼ [★]$ hashcat -a 3 -m 0 hash.txt HASHCAT?l?l?l?l?l20?1?d -1 02
hashcat (v6.2.6) starting

OpenCL API (OpenCL 3.0 PoCL 3.1+debian  Linux, None+Asserts, RELOC
```

* *

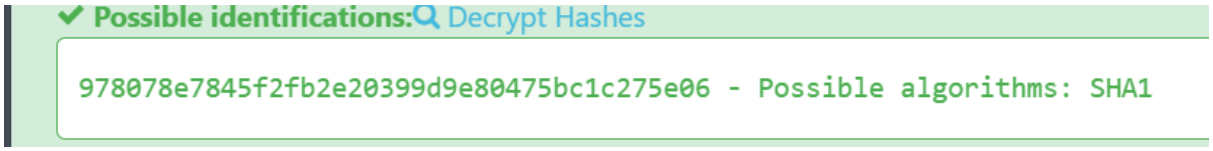
```
Host memory required for this attack: 1 MB

50a742905949102c961929823a2e8ca0:HASHCATqrstu2020

Session..........: hashcat
Status...........: Cracked
Hash.Mode........: 0 (MD5)
Hash.Target......: 50a742905949102c961929823a2e8ca0
Time.Started.....: Sat Nov 16 09:20:24 2024 (7 secs)
Time.Estimated...: Sat Nov 16 09:20:31 2024 (0 secs)
Kernel.Feature...: Pure Kernel
Guess.Mask.......: HASHCAT?l?l?l?l?l20?1?d [16]
Guess.Charset....: -1 02, -2 Undefined, -3 Undefined, -4 Undefined
Guess.Queue......: 1/1 (100.00%)
```

Hybrid Mode:

Question: Crack the following hash: 978078e7845f2fb2e20399d9e80475bc1c275e06 using the mask ?d?s.

Answer: hybridmaster9$

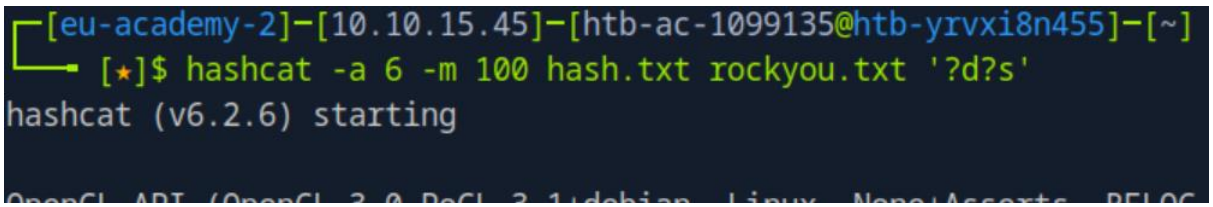
Method: using the [this hash analyzer] – the hash is SHA1 hash:



So we will use the command:

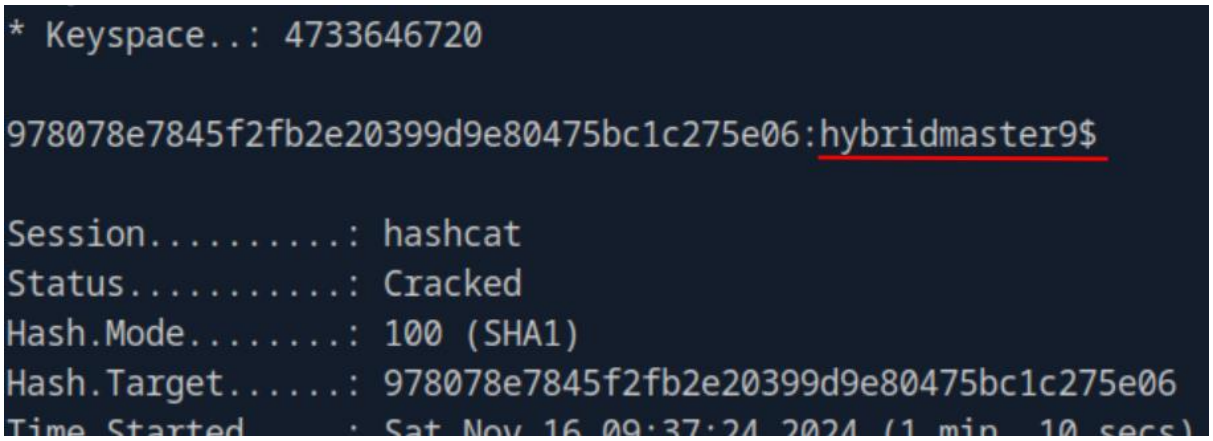
```
hashcat -a 6 -m 100 hash.txt rockyou.txt '?d?s'
```

where '-m 100' is for sha1:



*

*



Working with Wordlists

Working with Rules:

Question: Crack the following SHA1 hash using the techniques taught for generating a custom rule: 46244749d1e8fb99c37ad4f14fccb601ed4ae283. Modify the example rule in the beginning of the section to append 2020 to the end of each password attempt.

Answer: R@c3c@r2020

Method: having the hash in 'hash.txt' and the rockyou.txt – first we create the rule:

```
echo 'c so0 si1 se3 ss5 sa@ $2 $0 $2 $0' > rule.txt
```

```
┌─[eu-academy-2]─[10.10.15.45]─[htb-ac-1099135@htb-hlggj3znfb]─[~]
└──╼ [★]$ echo 'c so0 si1 se3 ss5 sa@ $2 $0 $2 $0' > rule.txt
```

Then – we crack:

```
hashcat -a 0 -m 100 hash.txt rockyou.txt -r rule.txt
```

```
┌─[eu-academy-2]─[10.10.15.45]─[htb-ac-1099135@htb-hlggj3znfb]─[~]
└──╼ [★]$ hashcat -a 0 -m 100 hash.txt rockyou.txt -r rule.txt
hashcat (v6.2.6) starting

OpenCL API (OpenCL 3.0 PoCL 3.1+debian  Linux, None+Asserts, RELOC
```

*

*

```
* Keyspace..: 14344384

46244749d1e8fb99c37ad4f14fccb601ed4ae283:R@c3c@r2020

Session..........: hashcat
Status...........: Cracked
Hash.Mode........: 100 (SHA1)
Hash.Target......: 46244749d1e8fb99c37ad4f14fccb601ed4ae283
Time.Started.....: Sat Nov 16 15:39:42 2024 (0 secs)
```

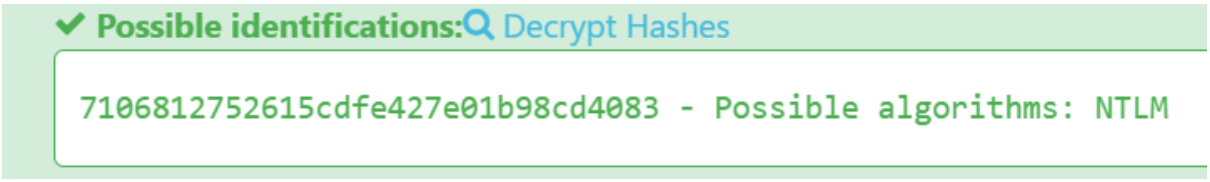
Cracking

Cracking Common Hashes:

Question: Crack the following hash: 7106812752615cdfe427e01b98cd4083

Answer: Password22$

Method: using the hash identifier:

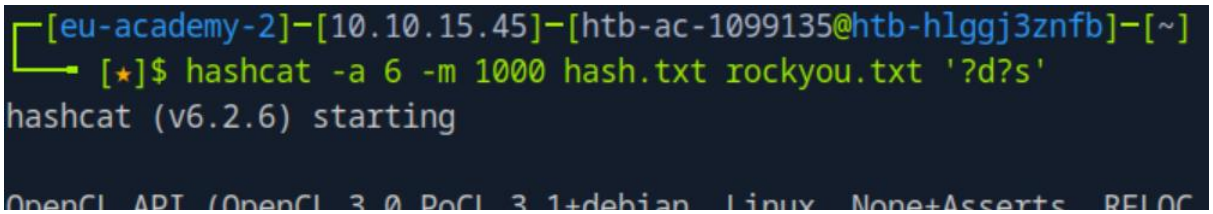


The hash is NTLM hash

Then we proceed to crack with the same rule from 'Hybrid Mode' section:

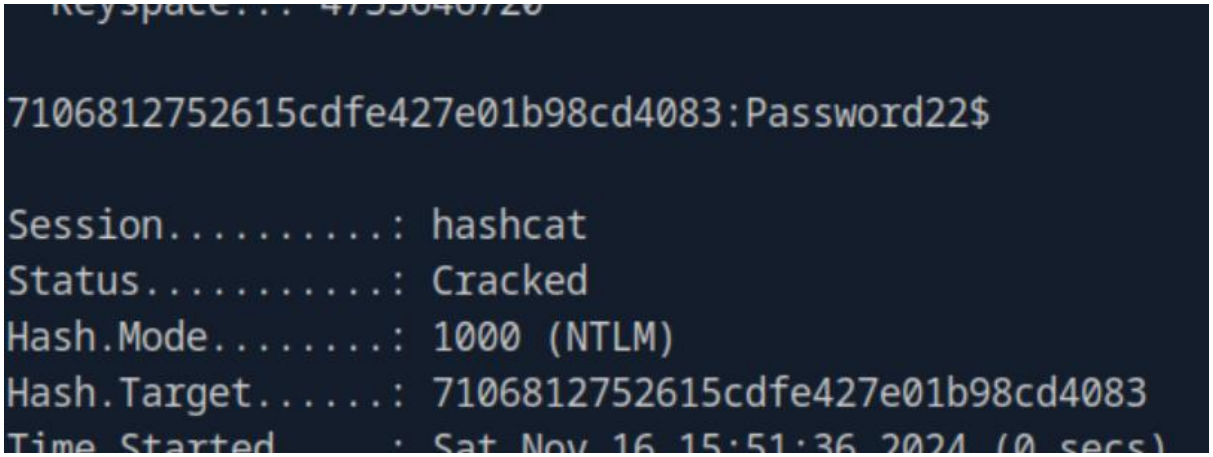
```
hashcat -a 6 -m 1000 hash.txt rockyou.txt '?d?s'
```

using '-m 1000' for NTLM cracking:



*

*



Cracking Common Hashes:

Question: Extract the hash from the attached 7-Zip file, crack the hash, and submit the value of the flag.txt file contained inside the archive.

Answer: 3c0e87a0396cb26d5b80dc03eeef8ea0

Method: first, we will download the attached file and unzip it:

```
wget
https://academy.hackthebox.com/storage/modules/20/Misc_hashes.zip -q

unzip Misc_hashes.zip
```

```
┌[eu-academy-2]-[10.10.15.45]-[htb-ac-1099135@htb-rjclugbbhr]-[~]
└─[★]$ wget https://academy.hackthebox.com/storage/modules/20/Misc_hashes.zip -q
```

* *

```
┌[eu-academy-2]-[10.10.15.45]-[ht
└─[★]$ unzip Misc_hashes.zip
Archive:  Misc_hashes.zip
 extracting: hashcat.7z
```

To get the 'hashcat.7z' file.

Upon extraction attempt, we will be required a password:

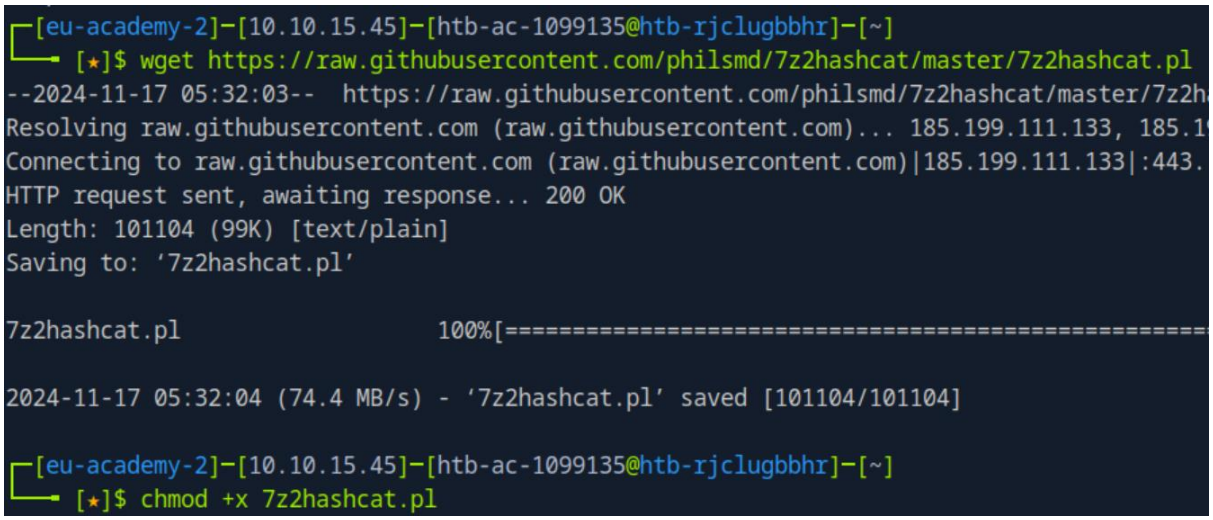
```
┌[eu-academy-2]-[10.10.15.45]-[htb-ac-1099135@htb-rjclugbbhr]-[~]
└─[★]$ 7z x hashcat.7z

7-Zip [64] 16.02 : Copyright (c) 1999-2016 Igor Pavlov : 2016-05-21
p7zip Version 16.02 (locale=en_US.UTF-8,Utf16=on,HugeFiles=on,64 bits,128 CPUs AMD EPYC 7543 32-Core Processor
 (A00F11),ASM,AES-NI)

Scanning the drive for archives:
1 file, 230 bytes (1 KiB)

Extracting archive: hashcat.7z

Enter password (will not be echoed):
ERROR: hashcat.7z
Can not open encrypted archive. Wrong password?
```

So we need to extract the hash and crack it. We will use the tool '7z2hashcat'. We download it:

```
wget
https://raw.githubusercontent.com/philsmd/7z2hashcat/master/7z2hashcat.pl
```

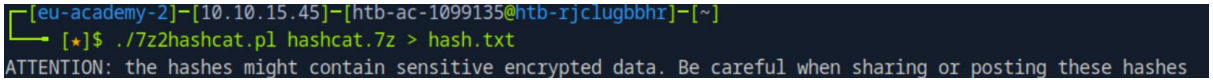
and then give it execution permissioms:

```
chmod +x 7z2hashcat.pl
```



Once the tool is ready, lets run run on the hashcat.7z to extract the hash from it:

```
./7z2hashcat.pl hashcat.7z > hash.txt
```

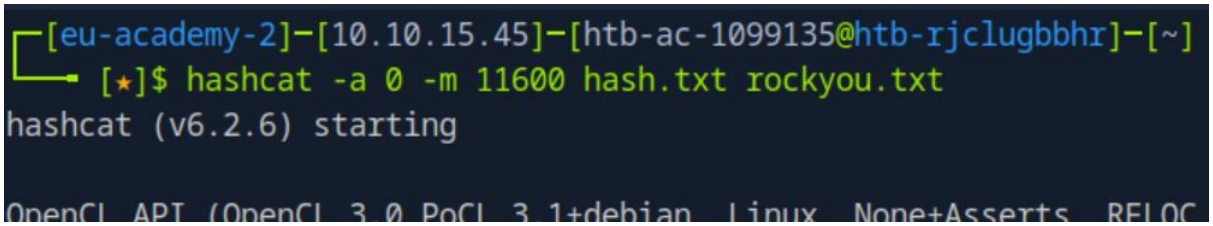


To an output file 'hash.txt'

We will proceed to crack the hash using the wordlist rockyou, and the command:

```
hashcat -a 0 -m 11600 hash.txt rockyou.txt
```

where -m 11600 is for 7z hashes:



*

*

```
* Keyspace..: 14344384
* Runtime...: 1 sec

$7z$0$19$0$$8$9c7684c204c437fa0000000000000000$1098215690$112$106$7395978cad9ad8b1
d69a1f37978e5df0179860d0fe4754721ae3cbbee1b558d93cd27e0b2959efe44a00305f982527d195
3d504fc3063744d081db1492ea1cdef7a9b983:123456789a

Session..........: hashcat
Status...........: Cracked
Hash.Mode........: 11600 (7-Zip)
Hash.Target......: $7z$0$19$0$$8$9c7684c204c437fa0000000000000000$1098...a9b983
```

The ‘hashcat.7z’ password is 123456789a.

We can now extract the hashcat.7z for the flag:

```
┌─[eu-academy-2]─[10.10.15.45]─[htb-ac-1099135@htb-rjclugbbhr]─[~]
└──╼ [★]$ 7z x hashcat.7z

7-Zip [64] 16.02 : Copyright (c) 1999-2016 Igor Pavlov : 2016-05-21
p7zip Version 16.02 (locale=en_US.UTF-8,Utf16=on,HugeFiles=on,64 bits,12
 (A00F11),ASM,AES-NI)

Scanning the drive for archives:
1 file, 230 bytes (1 KiB)

Extracting archive: hashcat.7z

Enter password (will not be echoed):
--
Path = hashcat.7z
Type = 7z
Physical Size = 230
Headers Size = 182
Method = LZMA2:12 7zAES
```

→

```
┌─[eu-academy-2]─[10.10.15.45]─[htb
└──╼ [★]$ cat flag.txt
3c0e87a0396cb26d5b80dc03eeef8ea0
```

Cracking Wireless (WPA/WPA2) Handshakes with Hashcat:

Question: Perform MIC cracking using the attached .cap file.

Answer: 1212312121

Method: First, we will download and extract the 'corp_question1-01.cap' file:

```
wget
https://academy.hackthebox.com/storage/modules/20/Hashcat_wi
reless1.zip -q

unzip Hashcat_wireless1.zip
```

```
┌[eu-academy-2]-[10.10.15.45]-[htb-ac-1099135@htb-rjclugbbhr]-[~]
└─► [★]$ wget https://academy.hackthebox.com/storage/modules/20/Hashcat_wireless1.zip -q
┌[eu-academy-2]-[10.10.15.45]-[htb-ac-1099135@htb-rjclugbbhr]-[~]
└─► [★]$ unzip Hashcat_wireless1.zip
Archive:  Hashcat_wireless1.zip
  inflating: corp_question1-01.cap
```

Now we will compile the tool 'hcxpcapngtool' (pre installed in the pwnbox) to extract the hash from the 'corp_question-01.cap':

```
hcxpcapngtool -o mic_to_crack.22000 corp_question1-01.cap
```

```
┌[eu-academy-2]-[10.10.15.45]-[htb-ac-1099135@htb-rjclugbbhr]-[~]
└─► [★]$ hcxpcapngtool -o mic_to_crack.22000 corp_question1-01.cap
hcxpcapngtool 6.2.7 reading from corp_question1-01.cap...
failed to read pcap packet header for packet 95415

summary capture file
--------------------
file name.............................: corp_question1-01.cap
version (pcap/cap)....................: 2.4 (very basic format without any additional information)
timestamp minimum (GMT)...............: 16.07.2020 17:07:54
timestamp maximum (GMT)              : 16.07.2020 17:12:43
```

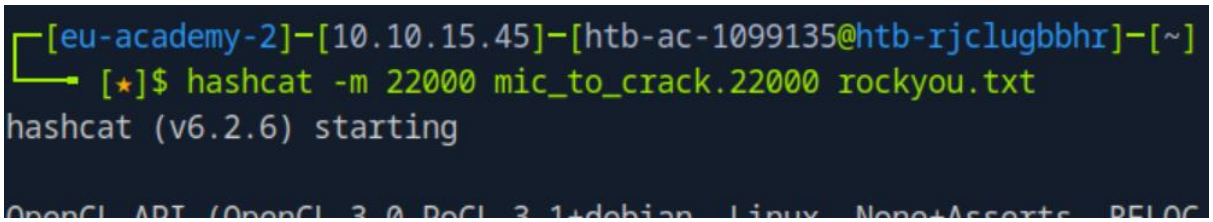
*

*

```
---------------
processed cap files...................: 1
```

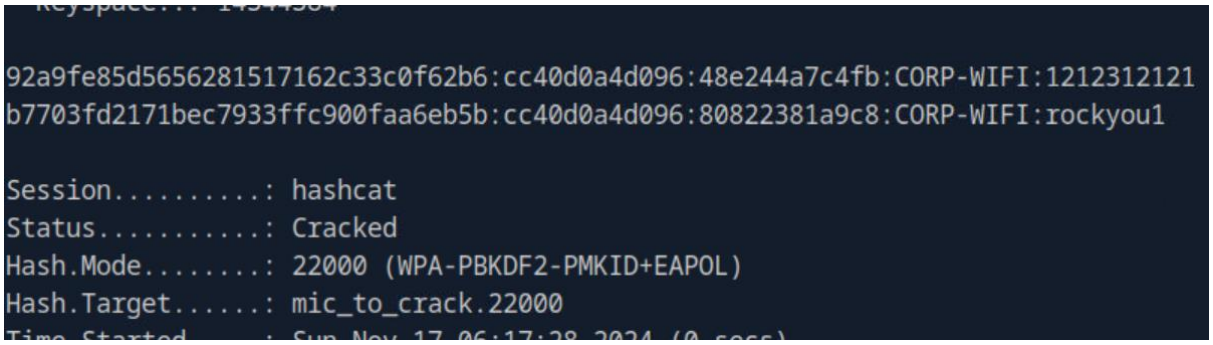
Now we can crack the output hash file ‘mic_to_crack.22000’:

```
hashcat -m 22000 mic_to_crack.22000 rockyou.txt
```



*

*



Question: Extract the PMKID hash from the attached .cap file and crack it.

Answer: 1password

Method: the cracking process is identical to the previous question, only here we use the commands:

```
hcxpcapngtool -o mic_to_crack2.22000
cracking_pmkid_question2.cap

hashcat -m 22000 mic_to_crack2.22000 rockyou.txt
```

(the original file is ‘cracking_pmkid_question2.cap’, then the hash is extracted to ‘mic_to_crack2.22000’)

Skills Assessment

Skills Assessment - Hashcat:

Question: What type of hash did your colleague obtain from the SQL injection attack?

Answer: SHA-1

Method: while we can use the hash identifier website, for exact answer format we will use the tool 'hashid':

```
hashid 0c67ac18f50c5e6b9398bfe1dc3e156163ba10ef
```

```
┌[eu-academy-2]-[10.10.15.45]-[htb-ac-1099135@htb-etkidqpnd6]-[~]
└─[★]$ hashid 0c67ac18f50c5e6b9398bfe1dc3e156163ba10ef
Analyzing '0c67ac18f50c5e6b9398bfe1dc3e156163ba10ef'
[+] SHA-1
[+] Double SHA-1
```

Question: What is the cleartext password for the hash obtained from SQL injection in example 1?

Answer: flower1

Method: we crack the hash with rockyou wordlist with '-m 100' hsahcat code for sha1 hashes:

```
hashcat -m 100 hash.txt rockyou.txt
```

where the hash is put in 'hash.txt':

```
┌[eu-academy-2]-[10.10.15.45]-[htb-ac-1099135@htb-etkidqpnd6]-[~]
└─[★]$ hashcat -m 100 hash.txt rockyou.txt
hashcat (v6.2.6) starting

OpenCL API (OpenCL 3.0 PoCL 3.1+debian  Linux  None+Asserts  RELOC
```

* *

```
0c67ac18f50c5e6b9398bfe1dc3e156163ba10ef:flower1

Session..........: hashcat
Status...........: Cracked
Hash.Mode........: 100 (SHA1)
Hash.Target......: 0c67ac18f50c5e6b9398bfe1dc3e156163ba10ef
Time.Started.....: Mon Nov 18 04:17:06 2024 (0 secs)
```

Question: What is the cleartext password value for the NetNTLMv2 hash?

Answer: bubbles1

Method: we put the NTLMv2 hash in a file 'hash2.txt', and proceed the same as the hash before it, only in here we use the hashcat code '-m 5600' for NTLMv2 hashes:

```
hashcat -m 1000 hash2.txt rockyou.txt
```

```
┌[eu-academy-2]-[10.10.15.45]-[htb-ac-1099135@htb-etkidqpnd6]-[~]
└─► [★]$ hashcat -m 5600 hash2.txt rockyou.txt
hashcat (v6.2.6) starting

OpenCL API (OpenCL 3.0 PoCL 3.1+debian  Linux  None+Asserts  RELOC  S
```

* *

```
BJONES::INLANEFREIGHT:699f1e768bd69c00:5304b6db9769d974a8f24c4f4309b6bc:0101000000000000c0653150de09d2010409df59f277926e000000
0002000800530040042003300001001e00570049004e002d0050005200480034003900320052005100410046005600040014005300 4d00420033002e006c00
6f00630061006c0003003400570049004e002d0050005200480034003900320052005100410046005600 2e0053004d00420033002e006c006f00630061006c
0005001400530040042003300 2e006c006f00630061006c0007000800c0653150de09d2010600040002000000080030003000000000000000000000030
0000b14866125d55255dd82c994c0d8ac3d9ff1a3efdaecbe908f1f91c7bd4b05cf50a0010000000000000000000000000000000000000090028006300690066
0073002f00310039003200 2e003100360038002e003100390035002e003100320039000000000000000000000000:bubbles1

Session..........: hashcat
Status...........: Cracked
Hash.Mode........: 5600 (NetNTLMv2)
Hash.Target......: BJONES::INLANEFREIGHT:699f1e768bd69c00:5304b6db9769...000000
```

Question: Crack the TGS ticket obtained from the Kerberoasting attack.

Answer: p@ssw0rdadmin

Method: we put the hash in 'hash3.txt' file, and run the hashcat for Kerberos hash – '-m 13100':

```
hashcat -m 13100 hash3.txt rockyou.txt
```

```
┌[eu-academy-2]-[10.10.15.45]-[htb-ac-1099135@htb-etkidqpnd6]-[~]
└─► [★]$ hashcat -m 13100 hash3.txt rockyou.txt
hashcat (v6.2.6) starting

OpenCL API (OpenCL 3.0 PoCL 3.1+debian  Linux  None+Asserts  RELOC  S
```

* *

```
7c452c0077efdea2a6c00704a8bee28326b5e554e1faa48a33963ce2c2e0e2446b4504a05d541bbaf531e1644ad92a2feae5b2eb8851b067e7bd8d7d23d82e
63d368983ba44f52901cba7e05cfa35e832ec445a7de50eca670fa90:p@ssw0rdadmin

Session..........: hashcat
Status...........: Cracked
Hash.Mode........: 13100 (Kerberos 5, etype 23, TGS-REP)
Hash.Target......: $krb5tgs$23$*sql_svc$INLANEFREIGHT.LOCAL$mssql/inla...70fa90
```

Question: What is the cleartext password value for the MS Cache 2 hash?

Answer: welcome1

Method: we put the hash in 'hash4.txt' file, and run the hashcat for with code '-m 2100':

```
hashcat -m 2100 hash4.txt rockyou.txt
```

```
┌[eu-academy-2]-[10.10.15.45]-[htb-ac-1099135@htb-etkidqpnd6]-[~]
└─[★]$ hashcat -m 2100 hash4.txt rockyou.txt
hashcat (v6.2.6) starting

OpenCL API (OpenCL 3.0 PoCL 3.1+debian  Linux, None+Asserts, RELOC
```

* *

```
* Keyspace..: 14344384

$DCC2$10240#backup_admin#62dabbde52af53c75f37df260af1008e:welcome1

Session..........: hashcat
Status...........: Cracked
Hash.Mode........: 2100 (Domain Cached Credentials 2 (DCC2), MS Cache 2)
Hash.Target......: $DCC2$10240#backup_admin#62dabbde52af53c75f37df260af1008e
Time.Started.....: Mon Nov 18 04:38:19 2024 (0 secs)
```

Question: After cracking the NTLM password hashes contained in the NTDS.dit file, perform an analysis of the results and find out the MOST common password in the INLANEFREIGHT.LOCAL domain.

Answer: freight1

Method: First, we will extract the 'DC01.inlanefreight.local.ntds' file from the provided zip file they give.

In it, are the various hashes of the various domain users:

```
┌[eu-academy-2]-[10.10.15.45]-[htb-ac-1099135@htb-ujm1rn3umw]-[~]
└─[★]$ head -40 DC01.inlanefreight.local.ntds
Administrator:500:aad3b435b51404eeaad3b435b51404ee:cf3a5525ee9414229e66279623ed5c58:::
Guest:501:aad3b435b51404eeaad3b435b51404ee:31d6cfe0d16ae931b73c59d7e0c089c0:::
krbtgt:502:aad3b435b51404eeaad3b435b51404ee:2a7537f27442d2aa20e26068e52faba8:::
DefaultAccount:503:aad3b435b51404eeaad3b435b51404ee:31d6cfe0d16ae931b73c59d7e0c089c0:::
sqladmin:1002:aad3b435b51404eeaad3b435b51404ee:55a6c7d7376319de86f83f28112b07e1:::
bjones:1108:aad3b435b51404eeaad3b435b51404ee:fdd42a89151b5d7d204a8bc07c972186:::
sql_svc:1109:aad3b435b51404eeaad3b435b51404ee:711fde548f259ffceb76fbd3d4605575:::
backup_admin:1110:aad3b435b51404eeaad3b435b51404ee:a3a685f89364d4a5182b028fbe79ac38:::
INLANEFREIGHT\Maureen.Woods:1114:aad3b435b51404eeaad3b435b51404ee:cf3a5525ee9414229e66279623ed5c58:::
INLANEFREIGHT\Margaret.Harris:1115:aad3b435b51404eeaad3b435b51404ee:d80f17f5a6bebf61c6c0c290782ce3fa:::
```

Where the NTLM hash is the right-most hash.

We will use the following python script to determine which NTLM hash appears the most:

```
from collections import Counter

# File path
file_path = 'DC01.inlanefreight.local.ntds'

# Read and process the file
with open(file_path, 'r') as file:
    ntlm_hashes = []
    for line in file:
        parts = line.strip().split(':')
        if len(parts) > 3:  # Ensure there are enough parts
            ntlm_hash = parts[3]  # The NTLM hash is the 4th
field
            ntlm_hashes.append(ntlm_hash)

# Count occurrences of each NTLM hash
hash_counts = Counter(ntlm_hashes)

# Find the most common NTLM hash
most_common_hash = hash_counts.most_common(1)

if most_common_hash:
    print("\nMost Common NTLM Hash:")
    print(f"{most_common_hash[0][0]}:
{most_common_hash[0][1]} occurrences")
```

```
┌[eu-academy-2]-[10.10.15.45]-[htb-ac-1099135@htb-ujm1rn3umw]-[~]
└─► [★]$ cat hash_counter.py
from collections import Counter

# File path
file_path = 'DC01.inlanefreight.local.ntds'

# Read and process the file
with open(file_path, 'r') as file:
    ntlm_hashes = []
    for line in file:
        parts = line.strip().split(':')
        if len(parts) > 3:  # Ensure there are enough parts
            ntlm_hash = parts[3]  # The NTLM hash is the 4th field
            ntlm_hashes.append(ntlm_hash)

# Count occurrences of each NTLM hash
hash_counts = Counter(ntlm_hashes)

# Find the most common NTLM hash
most_common_hash = hash_counts.most_common(1)


if most_common_hash:
    print("\nMost Common NTLM Hash:")
    print(f"{most_common_hash[0][0]}: {most_common_hash[0][1]} occurrences")
```

lets see this in action:

```
┌[eu-academy-2]-[10.10.15.45]-[htb-ac-1099135@htb-ujm1rn3umw]-[~]
└─► [★]$ python hash_counter.py

Most Common NTLM Hash:
db3a9af5e74be03220d213b47ef25b53: 43 occurrences
```

The hash ‘db3a9af5e74be03220d213b47ef25b53’ appears the most – 43 times.

All we are left to do is to put the hash in the file ‘hash.txt’ and crack it:

```
hashcat -m 1000 hash.txt rockyou.txt
```

```
db3a9af5e74be03220d213b47ef25b53:freight1

Session..........: hashcat
Status...........: Cracked
Hash.Mode........: 1000 (NTLM)
Hash.Target......: db3a9af5e74be03220d213b47ef25b53
Time.Started.....: Mon Nov 18 06:52:30 2024 (0 secs)
```