

Cyber Security Report

CMP5329 – Coursework Log format

Jaspreet Singh

Student ID: 19150299

Lab 1: Encrypting and Decrypting with OpenSSL

This lab use various mechanisms can be the cryptography and cryptographically secure hash functions, the digital signatures, certificates and PKI (Public Key Infrastructure), specifically, OpenSSL will be used.

The SSL and TLS are important certificates that keep internet connection secure and safeguard any sensitive data that is being sent between two systems.

Cryptography is the process of converting between readable text, and an unreadable form, is used to protect confidentiality of information other than protect it during the transmission,

Cryptographic techniques involve different methods, made specific by the use of keys.

- Algorithms that use a shared key are known as symmetric algorithms.
- Algorithms that use public and private key pairs are known as asymmetric algorithms.

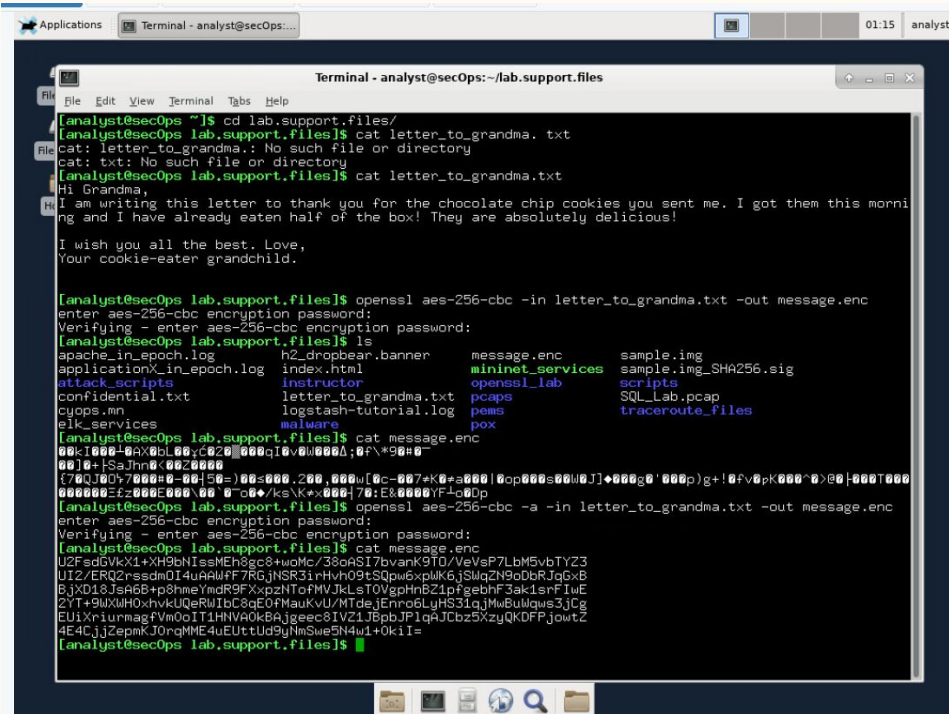
One of the biggest disadvantages is that you have to share the secret key somehow, however if an attacker finds out what the secret key is. Then files that were encrypted with that secret key are now compromised.

Lab 1: Encrypting and Decrypting with OpenSSL

Open the directory with the txt file containing the message to encrypt with AES-256 command with the “cyberops” password and then the file won't appear readable but in fact encrypted.

Now the file is encoded by running the OpenSSL command again, but this time by adding the -a option to tell the system to encode the file as base64.

After this step the message is displayed again with the cat command.

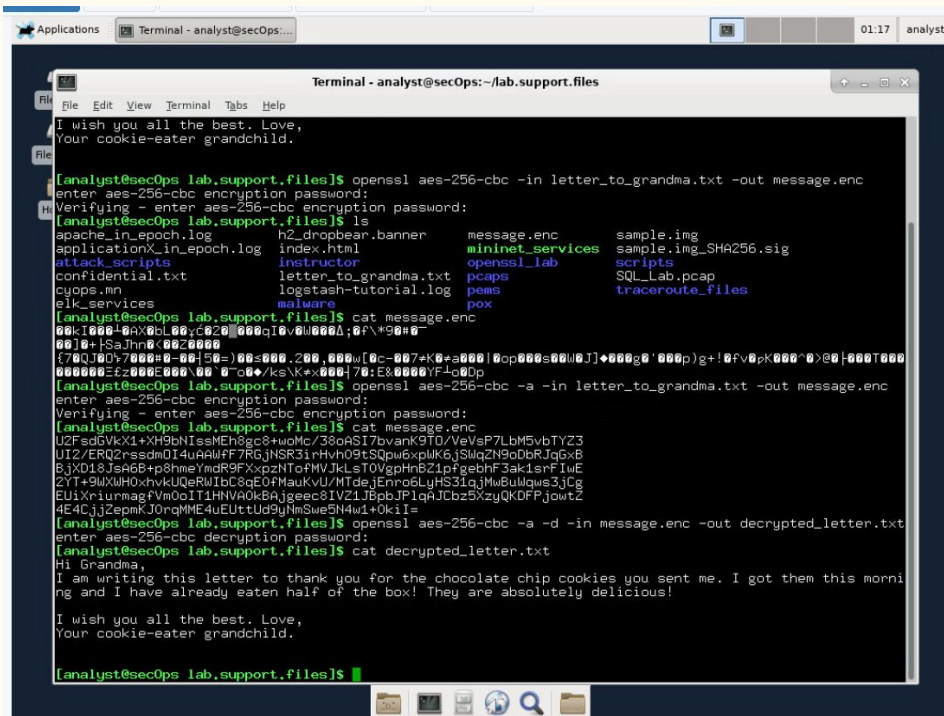


```
Applications Terminal - analyst@secOps:~
Terminal - analyst@secOps:~/lab.support.files
[analyst@secOps ~]$ cd lab.support.files/
[analyst@secOps lab.support.files]$ cat letter_to_grandma.txt
cat: letter_to_grandma.: No such file or directory
cat: txt: No such file or directory
[analyst@secOps lab.support.files]$ cat letter_to_grandma.txt
Hi Grandma,
I am writing this letter to thank you for the chocolate chip cookies you sent me. I got them this mornin
g and I have already eaten half of the box! They are absolutely delicious!

I wish you all the best. Love,
Your cookie-eater grandchild.

[analyst@secOps lab.support.files]$ openssl aes-256-cbc -in letter_to_grandma.txt -out message.enc
enter aes-256-cbc encryption password:
Verifying - enter aes-256-cbc encryption password:
[analyst@secOps lab.support.files]$ ls
apache_in_epoch.log      h2_dropbear.banner      message.enc              sample.img
applicationX_in_epoch.log index.html               openssl_lab              sample.img_SHA256.sig
attack_scripts           instructor               scripts                  SQL_Lab.pcap
confidential.txt         letter_to_grandma.txt   pcaps                   traceroute_files
cyops.mm                logstash-tutorial.log  pems
elk.services            malware
[analyst@secOps lab.support.files]$ cat message.enc
00kI000-0AX0bl00yC020_000qI0V0000A;0f\*00#0=
(70QJ00+7000#0-00;50=)00s000.200.0000[0c-007*K0+a000|0op000s0000J|+000g0'000p)g!0fv0pk000'0>00|000T000
000000Ez000E000\00 0~0c0/ks\K*x000|70:E&0000YF+c00p
[analyst@secOps lab.support.files]$ openssl aes-256-cbc -a -in letter_to_grandma.txt -out message.enc
enter aes-256-cbc encryption password:
Verifying - enter aes-256-cbc encryption password:
[analyst@secOps lab.support.files]$ cat message.enc
U2FsdGVkX1+XH9bNissMEh8gc8+woMc/38oASi7bvank9T0/VeVsP7LbM5vbTYZ3
UI2/ERQ2rssd0I4uAAUff7RGjNSR3iRhVh09tSQpw6xpMK6jSMqZN9oDbRJg6xB
BjXD18Jsa6B+p8hmeYndR9FXxpzNTofMVJkLsTOVgphnBZ1pfgebhf3ak1srFIwE
2YT+9WxUHOxhvkUQeRWIbC8qE0fMauKvU/MTdeJEnro6LyHS31qjMwBuWqws3jCg
EU1XrTurmagfVn0oIT1HNVA0kBAJgeecS1VZ1JBpbJP1qAJCzbz5XzyQKDFPjowtZ
4E4CjjZepmkJ0p0qWME4eUsttu09yhmSwe5M4u1+0k1i=
[analyst@secOps lab.support.files]$
```

Lab 1: Encrypting and Decrypting with OpenSSL



```
Applications Terminal - analyst@secOps... 01:17 analyst

Terminal - analyst@secOps: ~/lab.support.files
File Edit View Terminal Tabs Help

I wish you all the best. Love,
Your cookie-eater grandchild.

[analyst@secOps lab.support.files]$ openssl aes-256-cbc -in letter_to_grandma.txt -out message.enc
enter aes-256-cbc encryption password:
Verifying - enter aes-256-cbc encryption password:
[analyst@secOps lab.support.files]$ ls
apache_in_epoch.log      h2_dropbear.banner      message.enc              sample.img
applicationX_in_epoch.log index.html               mininet_services        sample.img_SHA256.sig
attack_scripts           instructor               openssl_lab             scripts
confidential.txt         letter_to_grandma.txt   pcaps                   SQL_Lab.pcap
cyops.mn                 logstash-tutorial.log  pems                    traceroute_files
elk_services            malware                 pox

[analyst@secOps lab.support.files]$ cat message.enc
00kI000+0AX0bL00;0200000qI0v00000A;0f\*90#0-
0000+ISaJhnn00000000
{70QJ00+700000-00;50-}00-000 200 0000[0--007+K0+a000!0op000:00W0J]+000g0'000p)g+!0fv0pK000'0>00!000T000
0000000=2000000000'0000+/ksKx000!70;E0.00000Pia0p
[analyst@secOps lab.support.files]$ openssl aes-256-cbc -a -in letter_to_grandma.txt -out message.enc
enter aes-256-cbc encryption password:
Verifying - enter aes-256-cbc encryption password:
[analyst@secOps lab.support.files]$ cat message.enc
U2FsdGVkX1+XH9bNissMEh8gc8+woMc/38oASi7bvank9TD/VeVsP7LbM5vbTYZ3
U12/ERQ2rssdmD14uAAHFF7R6jNJSR31rHvh09tSQpw6xplWk6jSWqZn9oDbRjGqxB
6X0D1Sjsa5Bp8hmeYndR9FXp2NToPwJLsTOVgphnBZlpgeBhP3aklsrFiwE
2YtA9uXW0hvxUQeRwI0c8q0VManukvU/NT0ejJenn06LyHS31qJMu8Ulwqs3Jc3
EUlXrTurmagfVmd0oIT1HNVA0kBAjgeec8IVZ1JBpbIP1qAJCbz5XyZkQDFPjowtZ
4E4CijZepmKJ0rqMME4uEUttUd9yNmSwe5N4w1+0kiI=
[analyst@secOps lab.support.files]$ openssl aes-256-cbc -a -d -in message.enc -out decrypted_letter.txt
enter aes-256-cbc decryption password:
[analyst@secOps lab.support.files]$ cat decrypted_letter.txt
Hi Grandma,
I am writing this letter to thank you for the chocolate chip cookies you sent me. I got them this morni
ng and I have already eaten half of the box! They are absolutely delicious!

I wish you all the best. Love,
Your cookie-eater grandchild.

[analyst@secOps lab.support.files]$
```

The message will now be decrypted and will ask for the same password used during the encryption process.

Now when the message is displayed it will be readable as same as it was before the encryption.

The Base64 could be easily missed causing confusion and errors.

Is important to select a strong password.

2. Lab 3: Access Control

Access Control refers to the control over access to system resources after a user's account credentials have been authenticated and access to the system has been granted to it.

For example, a particular user, or group of users, might only be permitted access to certain files after logging into a system, while at the same time being denied access to some other resources.

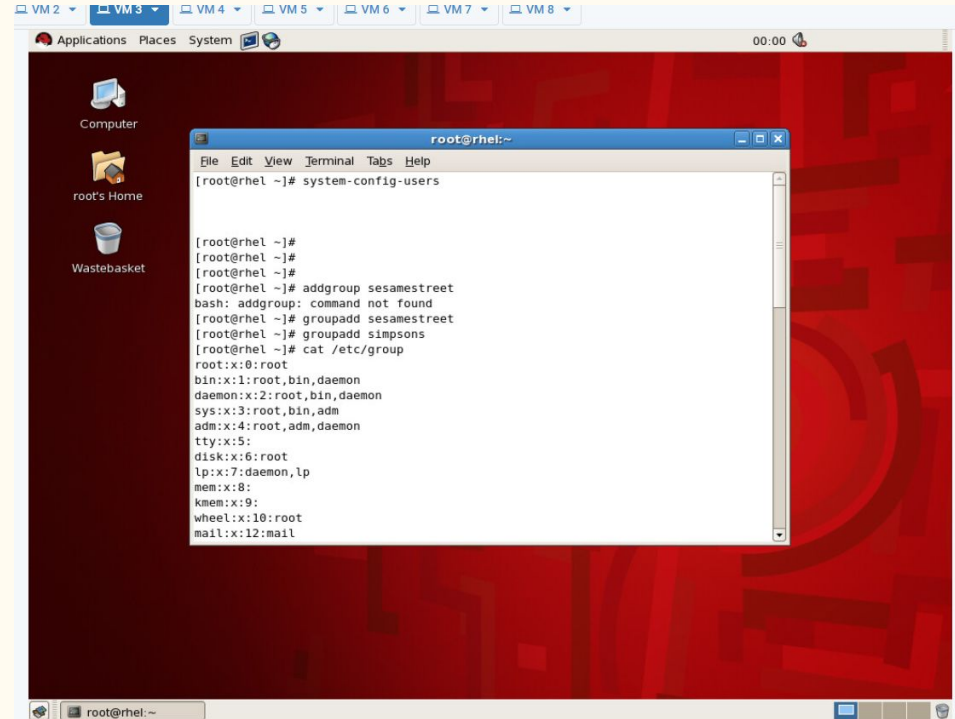
There are various types of access control:

- Discretionary Access Control (DAC)
- Mandatory Access Control (MAC)
- Role-based Access Control (RBAC)
- Rule-based Access Control

2. Lab 3: Access Control

Open Red Hat Linux virtual machine login and open the terminal to reach the user manager windows that can be used to see all the groups and users present in the system.

Once closed we can move to start adding new groups with the addgroup command and then display them with another command.



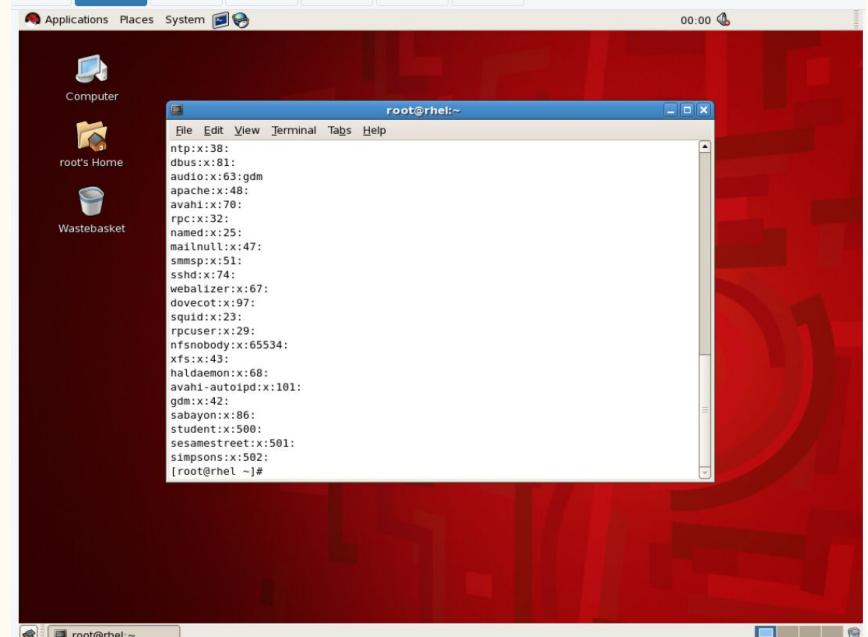
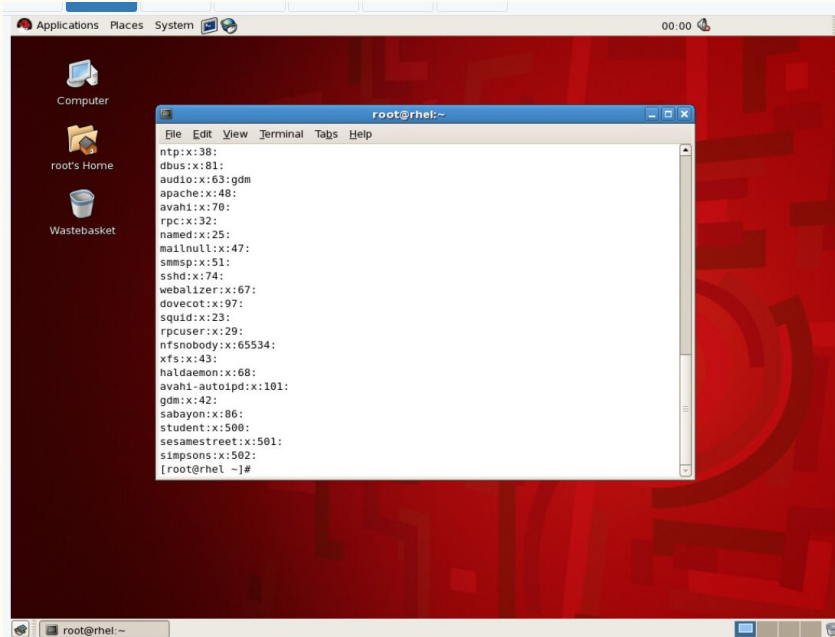
The screenshot shows a Red Hat Linux virtual machine desktop environment. The desktop has a red background with a geometric pattern. On the left side, there are icons for 'Computer', 'root's Home', and 'Wastebasket'. A terminal window titled 'root@rhel:~' is open in the center. The terminal shows the following commands and output:

```
root@rhel:~# system-config-users

[root@rhel ~]#
[root@rhel ~]#
[root@rhel ~]#
[root@rhel ~]# addgroup sesamestreet
bash: addgroup: command not found
[root@rhel ~]# groupadd sesamestreet
[root@rhel ~]# groupadd simpsons
[root@rhel ~]# cat /etc/group
root:x:0:root
bin:x:1:root,bin,daemon
daemon:x:2:root,bin,daemon
sys:x:3:root,bin,adm
adm:x:4:root,adm,daemon
tty:x:5:
disk:x:6:root
lp:x:7:daemon,lp
mem:x:8:
kmem:x:9:
wheel:x:10:root
mail:x:12:mail
```

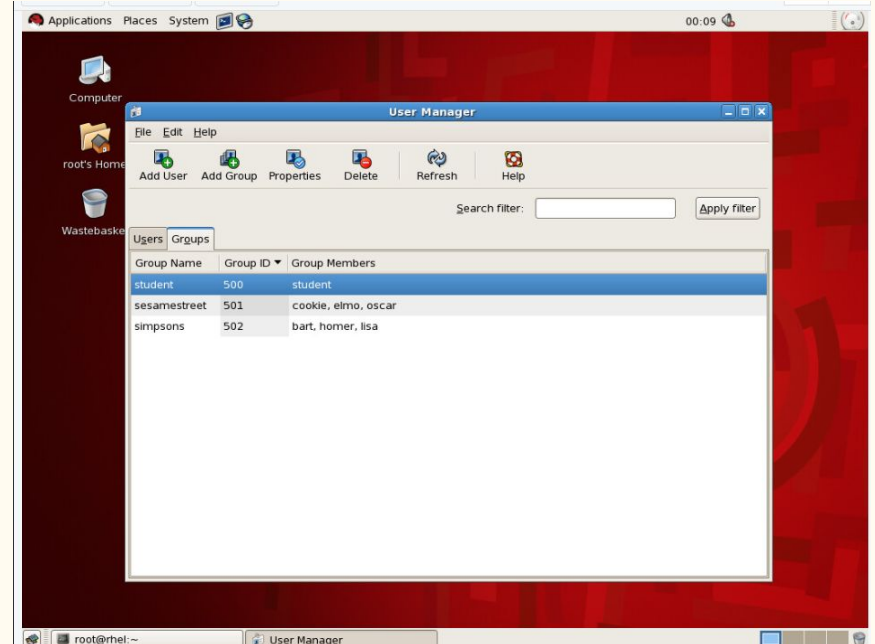
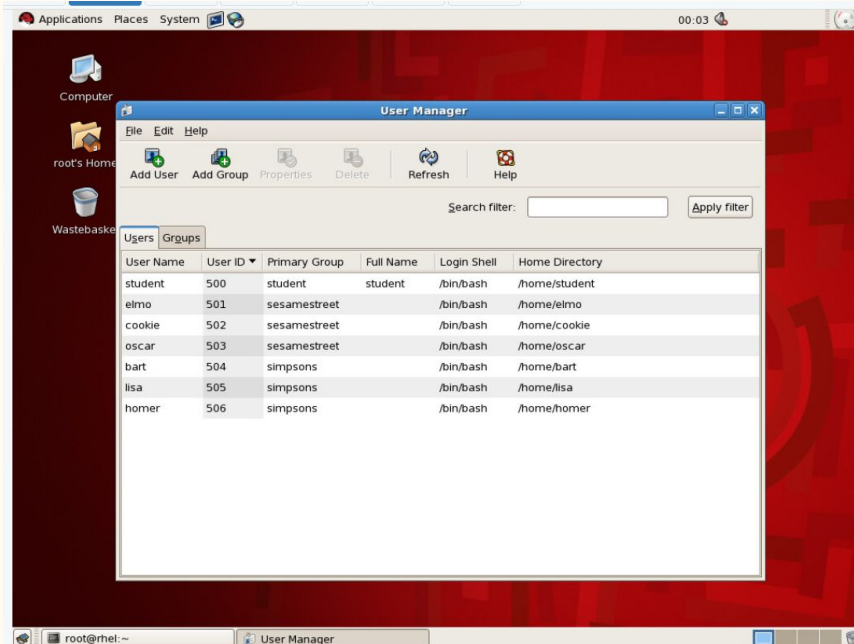
2. Lab 3: Access Control

Time to add the users with the useradd command to each group.



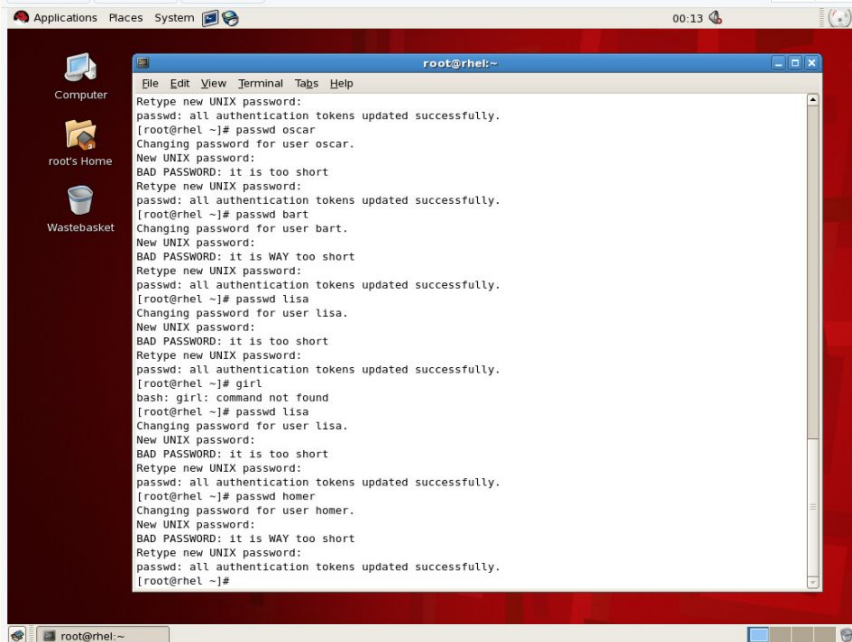
2. Lab 3: Access Control

After adding the various users the user manager is used to control if all the groups and users have been correctly inserted.

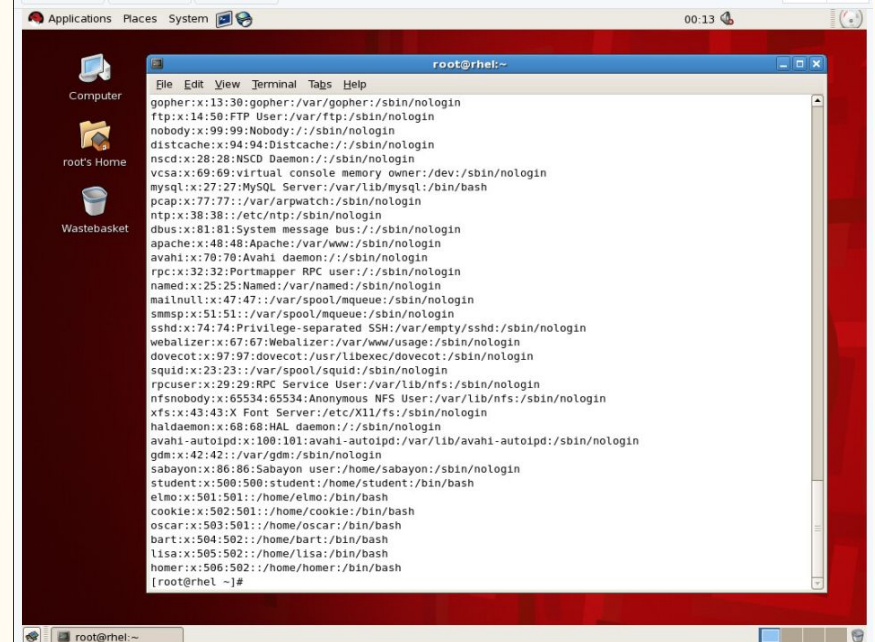


2. Lab 3: Access Control

All the passwords for the various users are created with the passwd command and it shows the users in the passwd file where we can see their directory too.

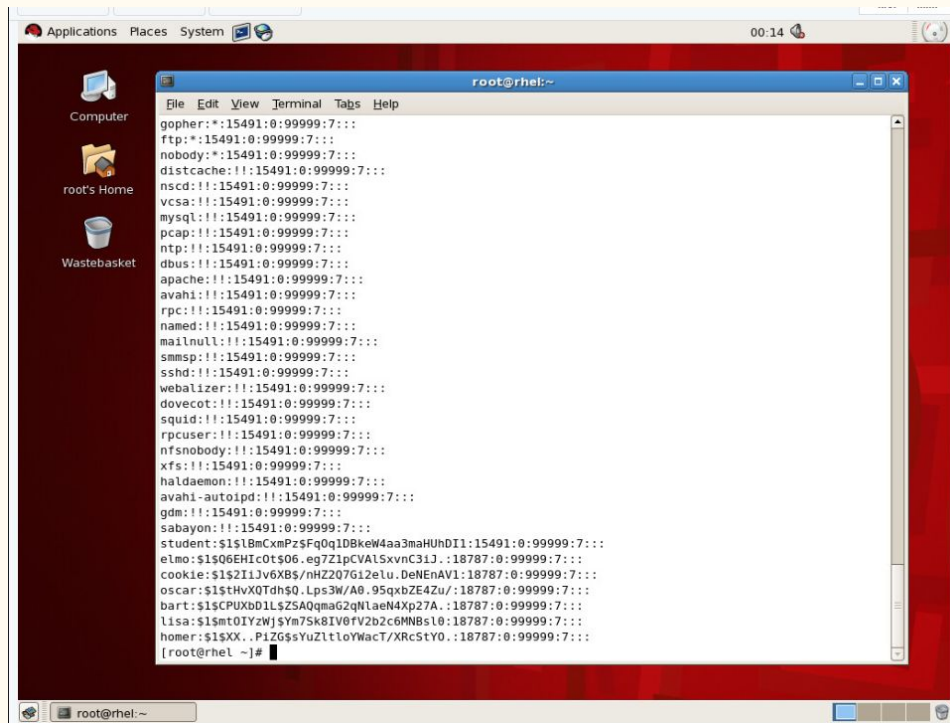


```
root@rhel:~  
Retype new UNIX password:  
passwd: all authentication tokens updated successfully.  
[root@rhel ~]# passwd oscar  
Changing password for user oscar.  
New UNIX password:  
BAD PASSWORD: it is too short  
Retype new UNIX password:  
passwd: all authentication tokens updated successfully.  
[root@rhel ~]# passwd bart  
Changing password for user bart.  
New UNIX password:  
BAD PASSWORD: it is WAY too short  
Retype new UNIX password:  
passwd: all authentication tokens updated successfully.  
[root@rhel ~]# passwd lisa  
Changing password for user lisa.  
New UNIX password:  
BAD PASSWORD: it is too short  
Retype new UNIX password:  
passwd: all authentication tokens updated successfully.  
[root@rhel ~]# passwd lisa  
Changing password for user lisa.  
New UNIX password:  
BAD PASSWORD: it is too short  
Retype new UNIX password:  
passwd: all authentication tokens updated successfully.  
[root@rhel ~]# passwd homer  
Changing password for user homer.  
New UNIX password:  
BAD PASSWORD: it is WAY too short  
Retype new UNIX password:  
passwd: all authentication tokens updated successfully.  
[root@rhel ~]#
```



```
root@rhel:~  
gopher:x:13:30:gopher:/var/gopher:/sbin/nologin  
ftp:x:14:50:FTP User:/var/ftp:/sbin/nologin  
nobody:x:99:99:Nobody:/sbin/nologin  
distcache:x:94:94:Distcache:/sbin/nologin  
nscd:x:28:28:NSCD Daemon:/sbin/nologin  
vcsa:x:69:69:virtual console memory owner:/dev:/sbin/nologin  
mysql:x:27:27:MySQL Server:/var/lib/mysql:/bin/bash  
pcap:x:77:77:/var/arpwatch:/sbin/nologin  
ntp:x:38:38:/etc/ntp:/sbin/nologin  
dbus:x:81:81:System message bus:/sbin/nologin  
apache:x:48:48:Apache:/var/www:/sbin/nologin  
avahi:x:70:70:Avahi daemon:/sbin/nologin  
rpc:x:32:32:Portmapper RPC user:/sbin/nologin  
named:x:25:25:Named:/var/named:/sbin/nologin  
mailnull:x:47:47:/var/spool/queue:/sbin/nologin  
smmsp:x:51:51:/var/spool/queue:/sbin/nologin  
sshd:x:74:74:Privilege-separated SSH:/var/empty/ssh:/sbin/nologin  
webalizer:x:67:67:Webalizer:/var/www/usage:/sbin/nologin  
dovecot:x:97:97:dovecot:/usr/libexec/dovecot:/sbin/nologin  
squid:x:23:23:/var/spool/squid:/sbin/nologin  
rpcuser:x:29:29:RPC Service User:/var/lib/nfs:/sbin/nologin  
nfsnobody:x:65534:65534:Anonymous NFS User:/var/lib/nfs:/sbin/nologin  
xfs:x:43:43:X Font Server:/etc/X11/fs:/sbin/nologin  
haldaemon:x:68:68:HAL daemon:/sbin/nologin  
avahi-autoipd:x:100:101:avahi-autoipd:/var/lib/avahi-autoipd:/sbin/nologin  
gdm:x:42:42:/var/gdm:/sbin/nologin  
sabayon:x:86:86:Sabayon user:/home/sabayon:/sbin/nologin  
student:x:500:500:student:/home/student:/bin/bash  
elmo:x:501:501:/home/elmo:/bin/bash  
cookie:x:502:501:/home/cookie:/bin/bash  
oscar:x:503:501:/home/oscar:/bin/bash  
bart:x:504:502:/home/bart:/bin/bash  
lisa:x:505:502:/home/lisa:/bin/bash  
homer:x:506:502:/home/homer:/bin/bash  
[root@rhel ~]#
```

2. Lab 3: Access Control



The screenshot shows a Linux desktop with a red background. A terminal window titled 'root@rhel:~' is open, displaying the contents of the /etc/shadow file. The desktop has a sidebar with icons for 'Computer', 'root's Home', and 'Wastebasket'. The terminal window has a menu bar with 'File', 'Edit', 'View', 'Terminal', 'Tabs', and 'Help'. The output of the 'cat /etc/shadow' command is as follows:

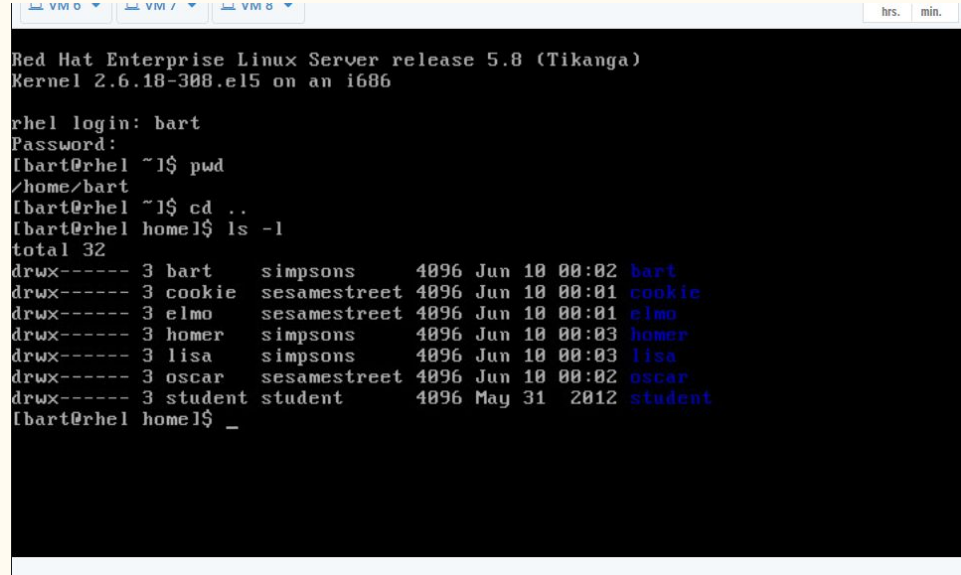
```
gopher:!:15491:0:99999:7:::
ftp:!:15491:0:99999:7:::
nobody:!:15491:0:99999:7:::
distcache:!:15491:0:99999:7:::
nscd:!:15491:0:99999:7:::
vcsa:!:15491:0:99999:7:::
mysql:!:15491:0:99999:7:::
pcap:!:15491:0:99999:7:::
ntp:!:15491:0:99999:7:::
dbus:!:15491:0:99999:7:::
apache:!:15491:0:99999:7:::
avahi:!:15491:0:99999:7:::
rpc:!:15491:0:99999:7:::
named:!:15491:0:99999:7:::
mailnull:!:15491:0:99999:7:::
smmsp:!:15491:0:99999:7:::
sshd:!:15491:0:99999:7:::
webalizer:!:15491:0:99999:7:::
dovecot:!:15491:0:99999:7:::
squid:!:15491:0:99999:7:::
rpcuser:!:15491:0:99999:7:::
nfsnobody:!:15491:0:99999:7:::
xfs:!:15491:0:99999:7:::
haldaemon:!:15491:0:99999:7:::
avahi-autoipd:!:15491:0:99999:7:::
gdm:!:15491:0:99999:7:::
sabayon:!:15491:0:99999:7:::
student:$1$L8mCxnPz$Fq0qID8keW4aa3maHUnDI1:15491:0:99999:7:::
elmo:$1$Q6EHicOtS06.eg7ZlpcVAL5xvnC31J.:18787:0:99999:7:::
cookie:$1$2I1Jv6XBs/nHZ207G12eLu.DeNeAV1:18787:0:99999:7:::
oscar:$1$tHvX0Tdh$0.Lps3W/A8.95qxbZE4Zu/:18787:0:99999:7:::
bart:$1$CPuXb0Ll$Z5AQqmaG2qNlaeN4Xp27A.:18787:0:99999:7:::
lisa:$1$mt0IYzWj$Ym7Sk8IV0fVZb2c6MNBs10:18787:0:99999:7:::
homer:$1$XX..P1Z6$YUzLtl0YMacT/XRcStY0.:18787:0:99999:7:::
[root@rhel ~]#
```

Now with the command `cat /etc/shadow` we can create the shadow file disponible to the root with all the details of the users created before.

2. Lab 3: Access Control

Now we can login in the accounts and experiment with their permissions but first the system should be restated with the init 6 command.

After logging in with the “bart” account we can go to it’s directory using the command `pwd` and then go back with the `cd ..` command and then display the list of the various directories present with the permissions on front for each user’s directory.



```
VM 0 VM 7 VM 8 hrs. min.
Red Hat Enterprise Linux Server release 5.8 (Tikanga)
Kernel 2.6.18-308.el5 on an i686

rhel login: bart
Password:
[bart@rhel ~]$ pwd
/home/bart
[bart@rhel ~]$ cd ..
[bart@rhel home]$ ls -l
total 32
drwx----- 3 bart      simpsons    4096 Jun 10 00:02 bart
drwx----- 3 cookie    sesamestreet 4096 Jun 10 00:01 cookie
drwx----- 3 elmo      sesamestreet 4096 Jun 10 00:01 elmo
drwx----- 3 homer     simpsons    4096 Jun 10 00:03 homer
drwx----- 3 lisa      simpsons    4096 Jun 10 00:03 lisa
drwx----- 3 oscar     sesamestreet 4096 Jun 10 00:02 oscar
drwx----- 3 student   student     4096 May 31 2012 student
[bart@rhel home]$ _
```

2. Lab 3: Access Control

When using the command `cd` to enter into another user's directory from the `bart` user the terminal shows that it doesn't have the permissions to do so and it shows permission denied.

Once logged in with the `lisa` account the `chmod` command is used to add special permissions to the `lisa`'s group and it is now possible to see the permissions of the `lisa`'s directory is now changed.

```
VM 6 VM 7 VM 8 hrs. min.
Red Hat Enterprise Linux Server release 5.8 (Tikanga)
Kernel 2.6.18-308.el5 on an i686

rhel login: bart
Password:
[bart@rhel ~]$ pwd
/home/bart
[bart@rhel ~]$ cd ..
[bart@rhel home]$ ls -l
total 32
drwx----- 3 bart simpsons 4096 Jun 10 00:02 bart
drwx----- 3 cookie sesamestreet 4096 Jun 10 00:01 cookie
drwx----- 3 elmo sesamestreet 4096 Jun 10 00:01 elmo
drwx----- 3 homer simpsons 4096 Jun 10 00:03 homer
drwx----- 3 lisa simpsons 4096 Jun 10 00:03 lisa
drwx----- 3 oscar sesamestreet 4096 Jun 10 00:02 oscar
drwx----- 3 student student 4096 May 31 2012 student
[bart@rhel home]$ cd lisa
-bash: cd: lisa: Permission denied
[bart@rhel home]$ _
```

```
rhel login: lisa
Password:
[lisa@rhel ~]$ pwd
/home/lisa
[lisa@rhel ~]$ cd ..
[lisa@rhel home]$ ls -l
total 32
drwx----- 3 bart simpsons 4096 Jun 10 00:29 bart
drwx----- 3 cookie sesamestreet 4096 Jun 10 00:01 cookie
drwx----- 3 elmo sesamestreet 4096 Jun 10 00:01 elmo
drwx----- 3 homer simpsons 4096 Jun 10 00:03 homer
drwx----- 3 lisa simpsons 4096 Jun 10 00:03 lisa
drwx----- 3 oscar sesamestreet 4096 Jun 10 00:02 oscar
drwx----- 3 student student 4096 May 31 2012 student
[lisa@rhel home]$ chmod g+rxw lisa
[lisa@rhel home]$ ls -l
total 32
drwx----- 3 bart simpsons 4096 Jun 10 00:29 bart
drwx----- 3 cookie sesamestreet 4096 Jun 10 00:01 cookie
drwx----- 3 elmo sesamestreet 4096 Jun 10 00:01 elmo
drwx----- 3 homer simpsons 4096 Jun 10 00:03 homer
drwxrwx--- 3 lisa simpsons 4096 Jun 10 00:03 lisa
drwx----- 3 oscar sesamestreet 4096 Jun 10 00:02 oscar
drwx----- 3 student student 4096 May 31 2012 student
[lisa@rhel home]$ _
```

2. Lab 3: Access Control

Now if the bart account in the same group as lisa try to access the directory it will show it without the permission denied message.

If another group's account, in this case elmo, tries to access Lisa's directory, the system will still show the permission denied message.

```
Red Hat Enterprise Linux Server release 5.8 (Tikanga)
Kernel 2.6.18-308.el5 on an i686
```

```
rhel login: bart
Password:
Last login: Thu Jun 10 00:27:00 on tty1
[bart@rhel ~]$ cd ..
[bart@rhel home]$ cd lisa
[bart@rhel lisa]$ whoami && pwd
bart
/home/lisa
[bart@rhel lisa]$ exit_
```

```
Red Hat Enterprise Linux Server release 5.8 (Tikanga)
Kernel 2.6.18-308.el5 on an i686
```

```
rhel login: elmo
Password:
[elmo@rhel ~]$ pwd
/home/elmo
[elmo@rhel ~]$ cd ..
[elmo@rhel home]$ ls -l
total 32
drwx----- 3 bart    simpsons    4096 Jun 10 00:29 bart
drwx----- 3 cookie  sesamestreet 4096 Jun 10 00:01 cookie
drwx----- 3 elmo    sesamestreet 4096 Jun 10 00:01 elmo
drwx----- 3 homer   simpsons    4096 Jun 10 00:03 homer
drwxrwx--- 3 lisa    simpsons    4096 Jun 10 00:33 lisa
drwx----- 3 oscar   sesamestreet 4096 Jun 10 00:02 oscar
drwx----- 3 student student    4096 May 31 2012 student
[elmo@rhel home]$ cd lisa
-bash: cd: lisa: Permission denied
[elmo@rhel home]$ exit_
```

2. Lab 3: Access Control

By using the command `chmod 707` Lisa's directory is now available for the other accounts as well. For this lab is required attention to the various access control permissions using the absolute and symbolic symbols.

Passwords can't be weak as if someone can access the root permission of the system could easily manipulate all the access of all the users and a possible malicious hacker could easily lock the users out of the system altogether.

```
Password:
Last login: Thu Jun 10 00:30:02 on tty1
[lisa@rhel ~]$ pwd
/home/lisa
[lisa@rhel ~]$ cd ..
[lisa@rhel home]$ ls -l
total 32
drwx----- 3 bart    simpsons    4096 Jun 10 00:29 bart
drwx----- 3 cookie  sesamestreet 4096 Jun 10 00:01 cookie
drwx----- 3 elmo    sesamestreet 4096 Jun 10 00:49 elmo
drwx----- 3 homer   simpsons    4096 Jun 10 00:03 homer
drwxrwx--- 3 lisa     simpsons    4096 Jun 10 00:33 lisa
drwx----- 3 oscar   sesamestreet 4096 Jun 10 00:02 oscar
drwx----- 3 student student    4096 May 31 2012 student
[lisa@rhel home]$ chmod 707 lisa
[lisa@rhel home]$ ls -l
total 32
drwx----- 3 bart    simpsons    4096 Jun 10 00:29 bart
drwx----- 3 cookie  sesamestreet 4096 Jun 10 00:01 cookie
drwx----- 3 elmo    sesamestreet 4096 Jun 10 00:49 elmo
drwx----- 3 homer   simpsons    4096 Jun 10 00:03 homer
drwx--rwx 3 lisa     simpsons    4096 Jun 10 00:33 lisa
drwx----- 3 oscar   sesamestreet 4096 Jun 10 00:02 oscar
drwx----- 3 student student    4096 May 31 2012 student
[lisa@rhel home]$ exit_
```

```
Red Hat Enterprise Linux Server release 5.8 (Tikanga)
Kernel 2.6.18-308.el5 on an i686

rhel login: elmo
Password:
Last login: Thu Jun 10 00:48:07 on tty1
[elmo@rhel ~]$ cd ..
[elmo@rhel home]$ cd lisa
[elmo@rhel lisa]$ whoami && pwd
elmo
/home/lisa
[elmo@rhel lisa]$ exit_
```

3. Lab 4: Reconnaissance with NMAP

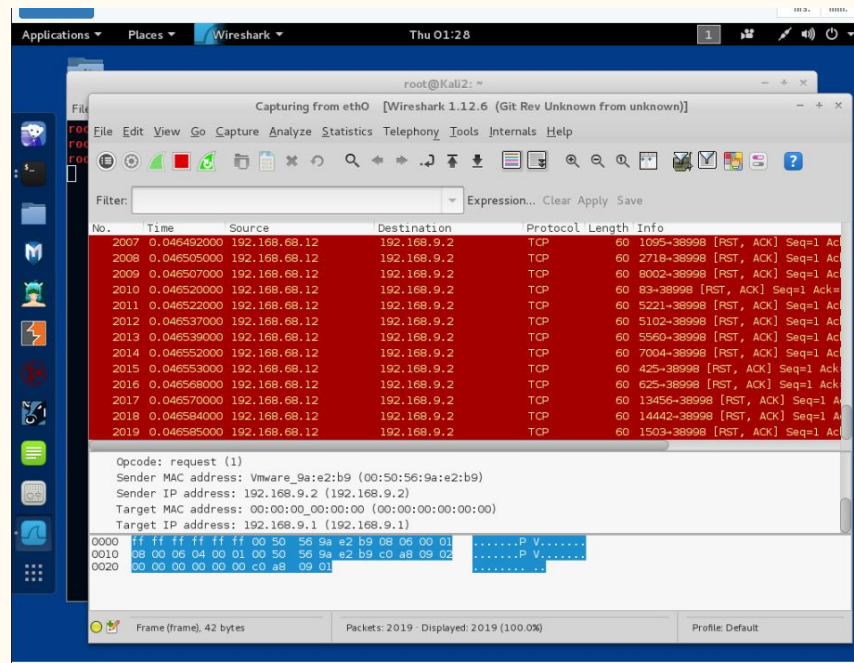
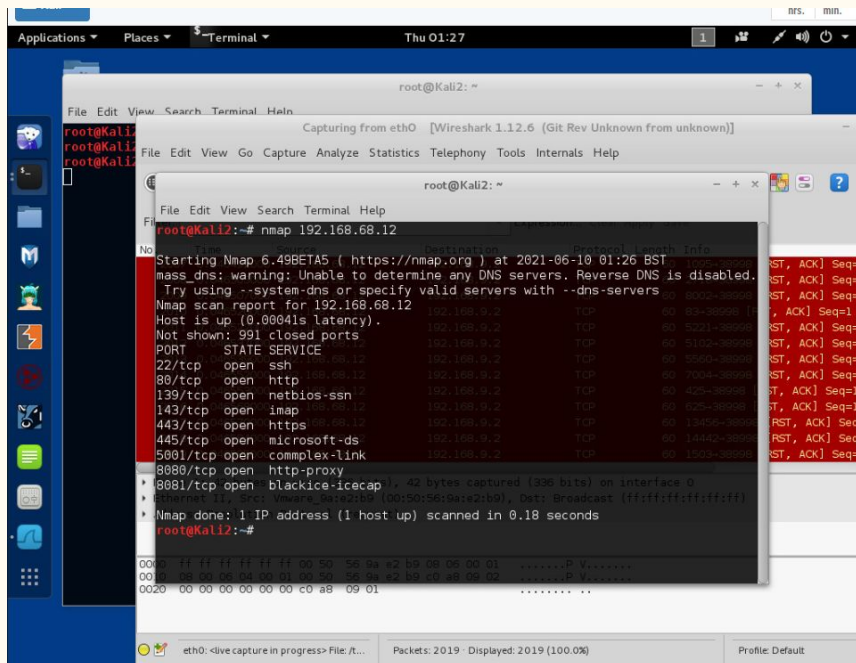
Network security is a very important subject that has to be highly maintained safe from possible malicious attacks or possible data breaches that could imply an important risk for the holder of the data, an example are the details of millions of accounts of a bank.

In this lab (Lab 1: Reconnaissance with Nmap & Amap) will be used the NMAP and AMAP commands to map a network to determine the status of the various ports and the possible breaches in the security of the network itself.

These commands are perfectly fine to use on an owned network but on the other hand they can easily be interpreted as a malicious attack by the admin of a company's network and the subject performing them can be prosecuted legally.

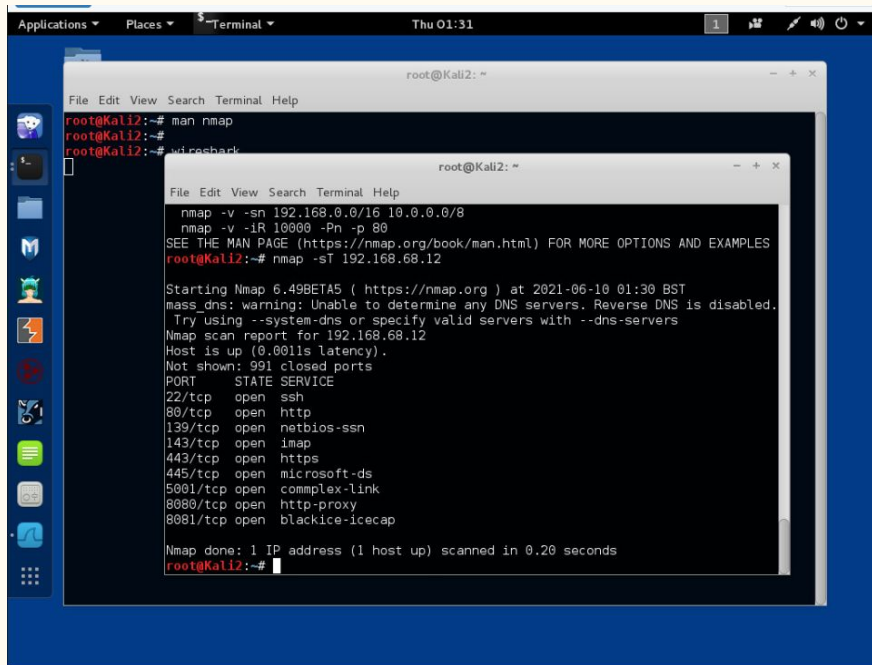
3. Lab 4: Reconnaissance with NMAP

The possible commands of nmap are displayed with the `man nmap` command. After seeing the various commands it is time to map the network with the IP using the command `nmap` and see the most used 1000 ports and the details of the requests using `wireshark`.



3. Lab 4: Reconnaissance with NMAP

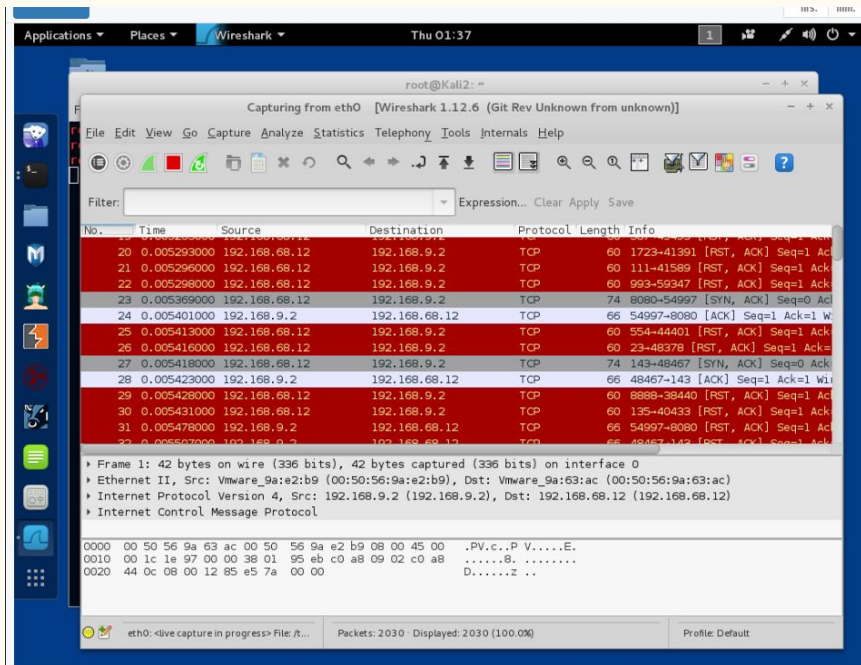
Now with the command `nmap -sT` a TCP connection scan will be done and some port will reply.



```
root@Kali2:~# man nmap
root@Kali2:~# nmap -v -sT 192.168.0.0/16 10.0.0.0/8
root@Kali2:~# nmap -v -sT 192.168.68.12

Starting Nmap 6.49BETA5 ( https://nmap.org ) at 2021-06-10 01:30 BST
mass_dns: warning: Unable to determine any DNS servers. Reverse DNS is disabled.
Try using --system-dns or specify valid servers with --dns-servers
Nmap scan report for 192.168.68.12
Host is up (0.001s latency).
Not shown: 991 closed ports
PORT      STATE SERVICE
22/tcp    open  ssh
80/tcp    open  http
139/tcp   open  netbios-ssn
143/tcp   open  imap
443/tcp   open  https
445/tcp   open  microsoft-ds
5001/tcp  open  complex-link
8080/tcp  open  http-proxy
8081/tcp  open  blackice-icecap

Nmap done: 1 IP address (1 host up) scanned in 0.20 seconds
root@Kali2:~#
```



Wireshark 1.12.6 (Git Rev Unknown from unknown)

Filter: Expression... Clear Apply Save

No.	Time	Source	Destination	Protocol	Length	Info
20	0.005293000	192.168.68.12	192.168.9.2	TCP	60	1723->41391 [RST, ACK] Seq=1 Ack=1
21	0.005296000	192.168.68.12	192.168.9.2	TCP	60	111->41589 [RST, ACK] Seq=1 Ack=1
22	0.005298000	192.168.68.12	192.168.9.2	TCP	60	993->59347 [RST, ACK] Seq=1 Ack=1
23	0.005369000	192.168.68.12	192.168.9.2	TCP	74	8080->54997 [SYN, ACK] Seq=0 Ack=1
24	0.005401000	192.168.9.2	192.168.68.12	TCP	66	54997->8080 [ACK] Seq=1 Ack=1 W.
25	0.005413000	192.168.68.12	192.168.9.2	TCP	60	554->44401 [RST, ACK] Seq=1 Ack=1
26	0.005416000	192.168.68.12	192.168.9.2	TCP	60	23->48378 [RST, ACK] Seq=1 Ack=1
27	0.005418000	192.168.68.12	192.168.9.2	TCP	74	143->48467 [SYN, ACK] Seq=0 Ack=1
28	0.005423000	192.168.9.2	192.168.68.12	TCP	66	48467->143 [ACK] Seq=1 Ack=1 W.
29	0.005428000	192.168.68.12	192.168.9.2	TCP	60	8888->38440 [RST, ACK] Seq=1 Ack=1
30	0.005431000	192.168.68.12	192.168.9.2	TCP	60	135->40433 [RST, ACK] Seq=1 Ack=1
31	0.005478000	192.168.9.2	192.168.68.12	TCP	66	54997->8080 [RST, ACK] Seq=1 Ack=1
32	0.005507000	192.168.9.2	192.168.68.12	TCP	66	48467->143 [RST, ACK] Seq=1 Ack=1

Frame 1: 42 bytes on wire (336 bits), 42 bytes captured (336 bits) on interface 0
• Ethernet II, Src: Vmware_9a:e2:b9 (00:50:56:9a:e2:b9), Dst: Vmware_9a:63:ac (00:50:56:9a:63:ac)
• Internet Protocol Version 4, Src: 192.168.9.2 (192.168.9.2), Dst: 192.168.68.12 (192.168.68.12)
• Internet Control Message Protocol

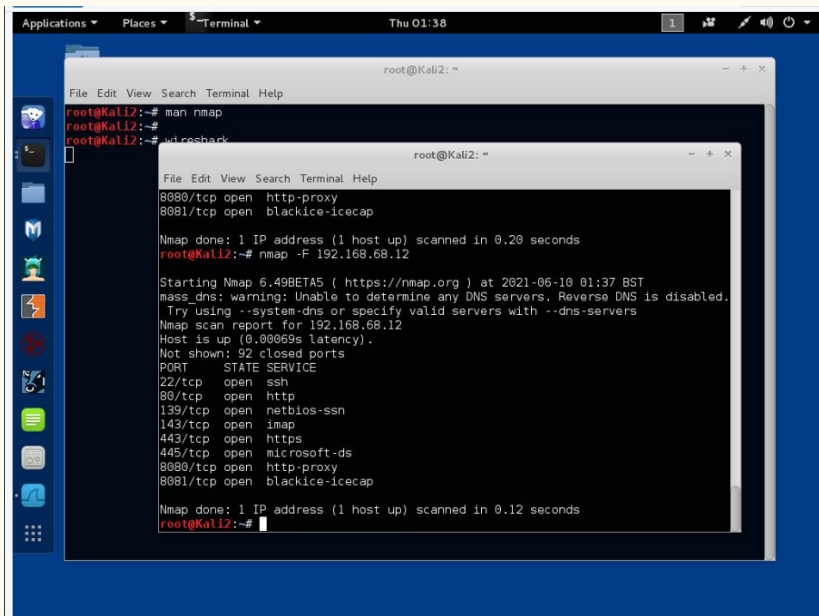
0000 00 50 56 9a 63 ac 00 50 56 9a e2 b9 08 00 45 00 .PV.c..P V.....E.
0010 00 1c 1a 97 00 00 38 01 95 eb c0 a8 09 02 c0 a8B.
0020 44 0c 08 00 12 85 e5 7a 00 002 ..

eth0: <live capture in progress> File: /... Packets: 2030 Displayed: 2030 (100.00%) Profile: Default

3. Lab 4: Reconnaissance with NMAP

Now with nmap -F command the scan will be quicker and only the first 100 ports will be scanned.

Following this scan the nmap -A scan is used and the received information increased greatly with the OS detection, version detection, script scanning and traceroute.

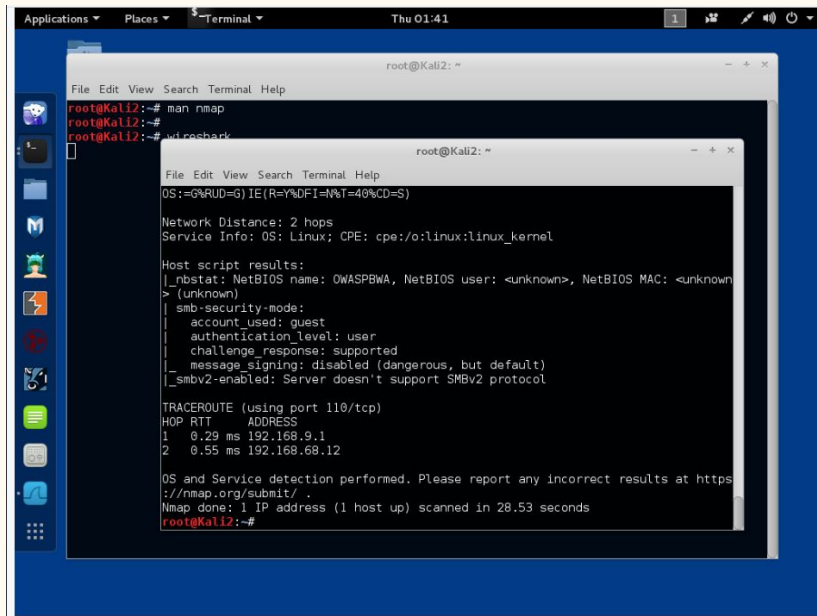


A terminal window on a Kali Linux desktop environment. The terminal shows the execution of the nmap -F command on the IP address 192.168.68.12. The output indicates that the scan was completed in 0.20 seconds and that 92 ports were closed. A list of open ports and their corresponding services is displayed.

```
root@Kali2:~# man nmap
root@Kali2:~# wireshark
root@Kali2:~# nmap -F 192.168.68.12

Nmap done: 1 IP address (1 host up) scanned in 0.20 seconds
root@Kali2:~#
```

PORT	STATE	SERVICE
22/tcp	open	ssh
80/tcp	open	http
139/tcp	open	netbios-ssn
143/tcp	open	imap
443/tcp	open	https
445/tcp	open	microsoft-ds
8080/tcp	open	http-proxy
8081/tcp	open	blackice-icecap



A terminal window on a Kali Linux desktop environment. The terminal shows the execution of the nmap -A command on the IP address 192.168.68.12. The output provides detailed information, including OS detection (Linux), network distance (2 hops), and a list of open ports with their services. It also includes a traceroute and a warning about DNS servers.

```
root@Kali2:~# man nmap
root@Kali2:~# wireshark
root@Kali2:~# nmap -A 192.168.68.12

OS: Linux 3.10 (Ubuntu 12.04 LTS)
Network Distance: 2 hops
Service Info: OS: Linux; CPE: cpe:/o:linux:linux_kernel

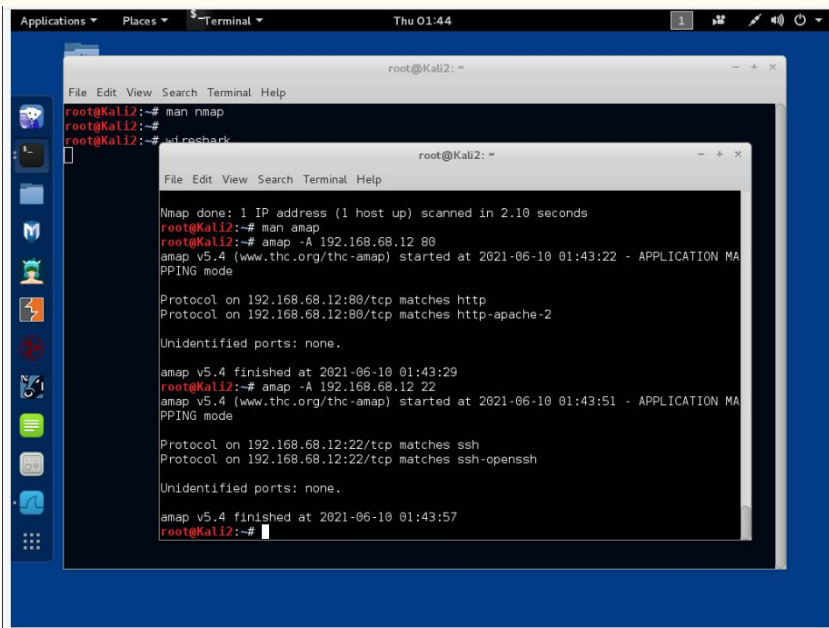
Host script results:
|_ nbstat: NetBIOS name: OWASPBWA, NetBIOS user: <unknown>, NetBIOS MAC: <unknown>
|_ smb-security-mode:
|   account_used: guest
|   authentication_level: user
|   challenge_response: supported
|_ message_signing: disabled (dangerous, but default)
|_ smb2-enabled: Server doesn't support SMB2 protocol

TRACEROUTE (using port 110/tcp)
HOP RTT ADDRESS
1 0.29 ms 192.168.9.1
2 0.55 ms 192.168.68.12

OS and Service detection performed. Please report any incorrect results at https://nmap.org/submit/
Nmap done: 1 IP address (1 host up) scanned in 28.53 seconds
root@Kali2:~#
```


3. Lab 4: Reconnaissance with NMAP

Finally we can get more information about any disponible port such as the version of the protocol or of the SHH used. (Use of Anap -A/-B command)



```
root@Kali2:~# man nmap
root@Kali2:~#
root@Kali2:~# wd rosback
root@Kali2:~#
Nmap done: 1 IP address (1 host up) scanned in 2.10 seconds
root@Kali2:~# man amap
root@Kali2:~# amap -A 192.168.68.12 80
amap v5.4 (www.thc.org/thc-amap) started at 2021-06-10 01:43:22 - APPLICATION MAPPING mode

Protocol on 192.168.68.12:80/tcp matches http
Protocol on 192.168.68.12:80/tcp matches http-apache-2

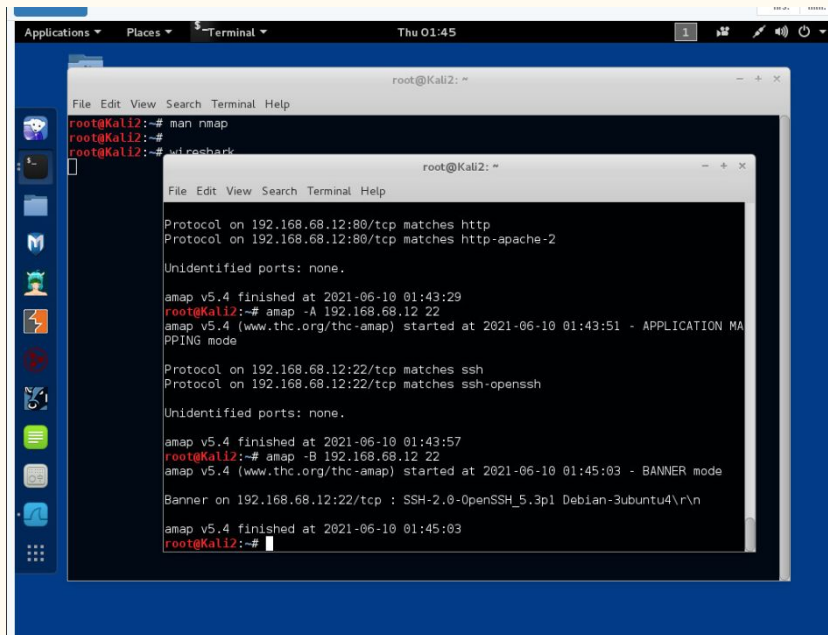
Unidentified ports: none.

amap v5.4 finished at 2021-06-10 01:43:29
root@Kali2:~# amap -A 192.168.68.12 22
amap v5.4 (www.thc.org/thc-amap) started at 2021-06-10 01:43:51 - APPLICATION MAPPING mode

Protocol on 192.168.68.12:22/tcp matches ssh
Protocol on 192.168.68.12:22/tcp matches ssh-openssh

Unidentified ports: none.

amap v5.4 finished at 2021-06-10 01:43:57
root@Kali2:~#
```



```
root@Kali2:~# man nmap
root@Kali2:~#
root@Kali2:~# wd rosback
root@Kali2:~#
Protocol on 192.168.68.12:80/tcp matches http
Protocol on 192.168.68.12:80/tcp matches http-apache-2

Unidentified ports: none.

amap v5.4 finished at 2021-06-10 01:43:29
root@Kali2:~# amap -A 192.168.68.12 22
amap v5.4 (www.thc.org/thc-amap) started at 2021-06-10 01:43:51 - APPLICATION MAPPING mode

Protocol on 192.168.68.12:22/tcp matches ssh
Protocol on 192.168.68.12:22/tcp matches ssh-openssh

Unidentified ports: none.

amap v5.4 finished at 2021-06-10 01:43:57
root@Kali2:~# amap -B 192.168.68.12 22
amap v5.4 (www.thc.org/thc-amap) started at 2021-06-10 01:45:03 - BANNER mode

Banner on 192.168.68.12:22/tcp : SSH-2.0-OpenSSH_5.3p1 Debian-3ubuntu4\r\n

amap v5.4 finished at 2021-06-10 01:45:03
root@Kali2:~#
```

3. Lab 4: Reconnaissance with NMAP

Nmap gives you the ability to explore any devices connected to a network, finding information like which applications are listening on open ports and the operating system a device is running. This information lets a hacker design an attack that perfectly suits the target environment.

These commands should be used wisely as if a hacker wants to remain anonymous it could easily use a VPN and utilize the `amap -F` command to be less visible as the requests sent are just a fraction compared to the other commands.

Hackers can also use these scanning commands to find a bunch of services running on open ports, this can be a huge benefit, if one of them is vulnerable to possible breaches.

4. Lab 5: Password cracking with John the Ripper

Cybersecurity Compliance involves meeting various controls that are enacted by a regulatory authority, law, or industry to protect the confidentiality, integrity, and availability of data stored. Most cybersecurity compliance requirements require a risk and vulnerability assessment. These determinate what's your organization's most critical security flaws, as well as how efficient are the controls that are already in place.

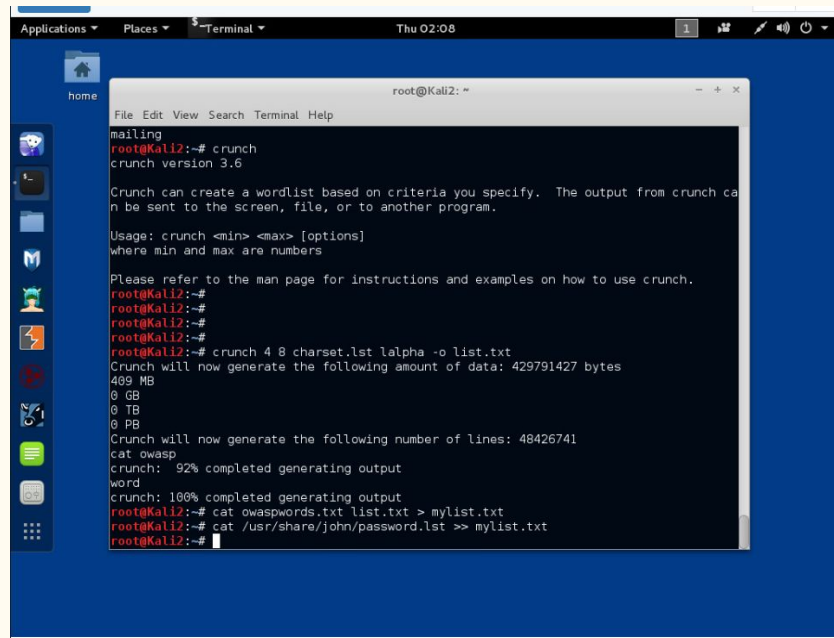
Cybersecurity isn't just about software and hardware. Having policies and procedures in place to reduce risk is also important for both safety and compliance. That's why companies have mandatory cybersecurity training for their employees and they conduct various risk and vulnerability assessments.

The most famous ways to crack passwords of a system are by using:

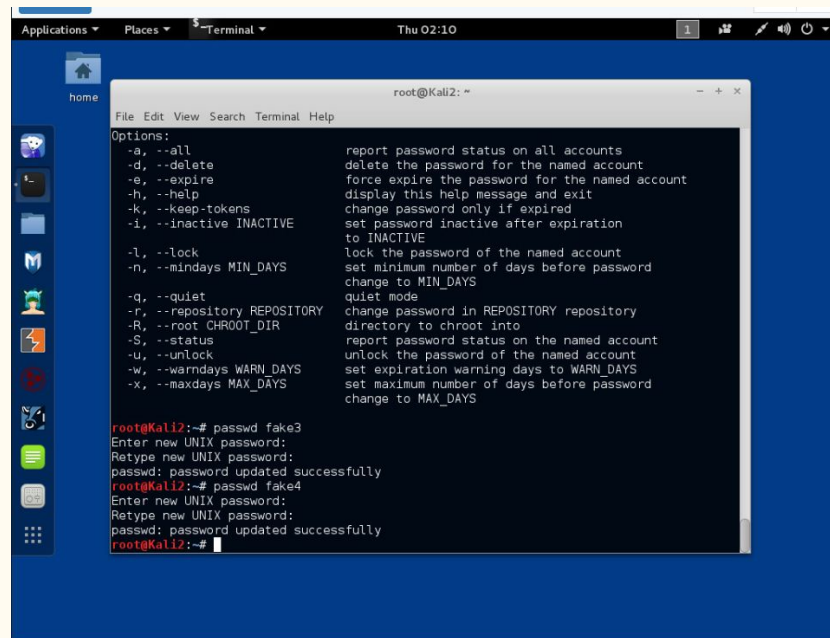
- John the Ripper
- Hashcat

4. Lab 5: Password cracking with John the Ripper

The cewl command is used to copy contents of the attacked virtual machine to a text file then the crunch command is used to create the passwords. Finally these files are united with the other list created by john the ripper. Then using the adduser and passwd command we create the fake account's details.



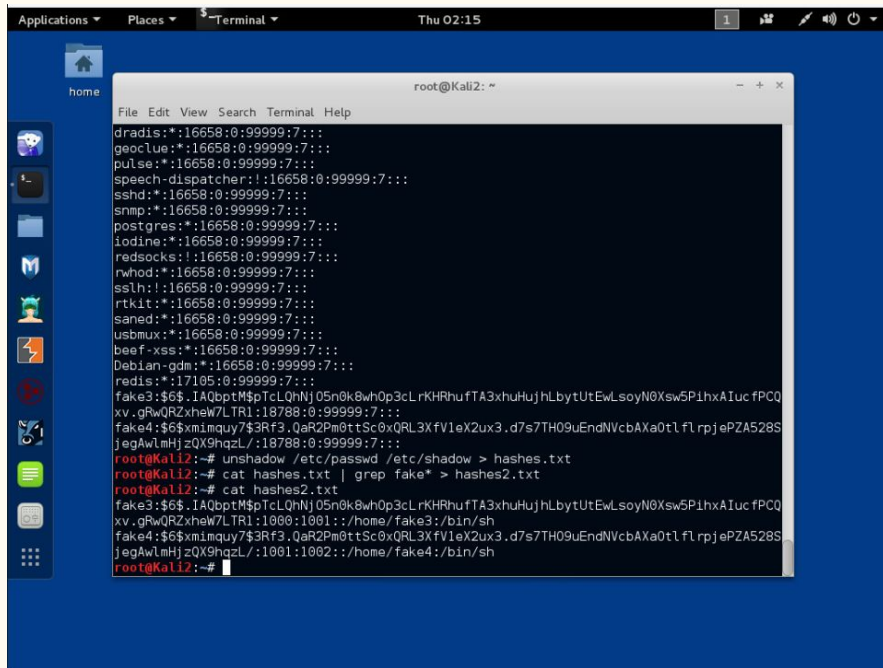
```
root@Kali2: ~  
mailing  
root@Kali2:~# crunch  
crunch version 3.6  
  
Crunch can create a wordlist based on criteria you specify. The output from crunch can be sent to the screen, file, or to another program.  
  
Usage: crunch <min> <max> [options]  
where min and max are numbers  
  
Please refer to the man page for instructions and examples on how to use crunch.  
root@Kali2:~#  
root@Kali2:~#  
root@Kali2:~#  
root@Kali2:~# crunch 4 8 charset.lst lalpha -o list.txt  
Crunch will now generate the following amount of data: 429791427 bytes  
489 MB  
0 GB  
0 TB  
0 PB  
Crunch will now generate the following number of lines: 48426741  
cat owasp  
crunch: 92% completed generating output  
word  
crunch: 100% completed generating output  
root@Kali2:~# cat owaspwords.txt list.txt > mylist.txt  
root@Kali2:~# cat /usr/share/john/password.lst >> mylist.txt  
root@Kali2:~#
```



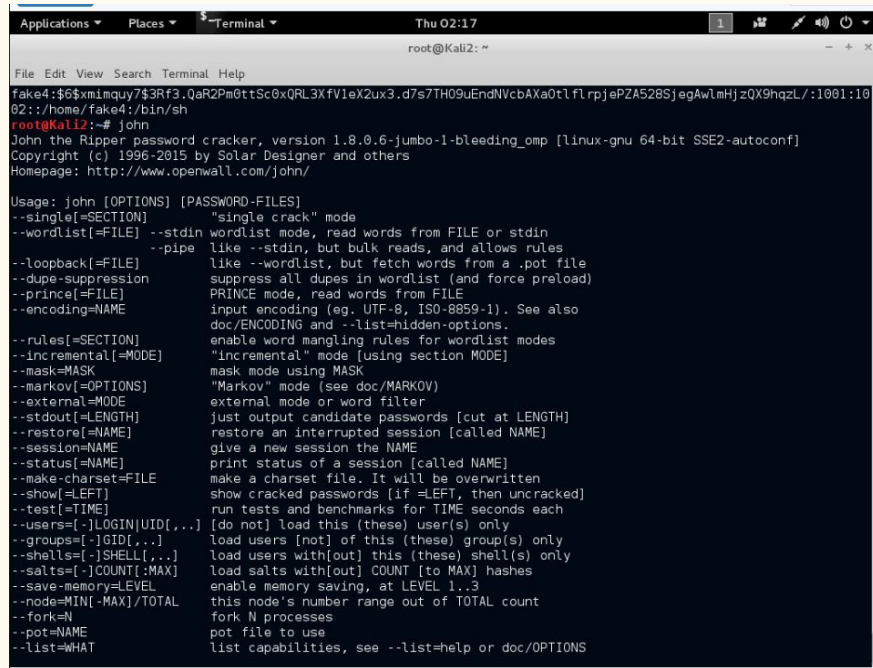
```
Options:  
-a, --all report password status on all accounts  
-d, --delete delete the password for the named account  
-e, --expire force expire the password for the named account  
-h, --help display this help message and exit  
-k, --keep-tokens change password only if expired  
-i, --inactive INACTIVE set password inactive after expiration to INACTIVE  
-l, --lock lock the password of the named account  
-n, --mindays MIN_DAYS set minimum number of days before password change to MIN_DAYS  
-q, --quiet quiet mode  
-r, --repository REPOSITORY change password in REPOSITORY repository  
-R, --root CHROOT_DIR directory to chroot into  
-S, --status report password status on the named account  
-u, --unlock unlock the password of the named account  
-w, --warndays WARN_DAYS set expiration warning days to WARN_DAYS  
-x, --maxdays MAX_DAYS set maximum number of days before password change to MAX_DAYS  
  
root@Kali2:~# passwd fake3  
Enter new UNIX password:  
Retype new UNIX password:  
passwd: password updated successfully  
root@Kali2:~# passwd fake4  
Enter new UNIX password:  
Retype new UNIX password:  
passwd: password updated successfully  
root@Kali2:~#
```


4. Lab 5: Password cracking with John the Ripper

Now the updated files of the new user's details are united in the hash.txt file then it's narrowed down to the 2 fake accounts and shown in the terminal. Then we open john the ripper to see the possible options.



```
root@kali2: ~  
File Edit View Search Terminal Help  
dradis:*:16658:0:99999:7:::  
geoclue:*:16658:0:99999:7:::  
pulse:*:16658:0:99999:7:::  
speech-dispatcher:*:16658:0:99999:7:::  
sshd:*:16658:0:99999:7:::  
snmp:*:16658:0:99999:7:::  
postgres:*:16658:0:99999:7:::  
iodine:*:16658:0:99999:7:::  
redsocks:*:16658:0:99999:7:::  
rwhod:*:16658:0:99999:7:::  
sslsht:*:16658:0:99999:7:::  
rtkit:*:16658:0:99999:7:::  
saned:*:16658:0:99999:7:::  
usbmux:*:16658:0:99999:7:::  
beef-xss:*:16658:0:99999:7:::  
Debian-gdm:*:16658:0:99999:7:::  
redis:*:17105:0:99999:7:::  
fake3:$6$IA0bptM$ptTcLQHnJ05n0k8wh0p3cLrKHRhufTA3xhuHujhLbytUtEwLsoyN0$Xsw5P1hxAIucfPCQ  
xv.gRwQR2xheW7Lr1:18788:0:99999:7:::  
fake4:$6$xmimquy7$3Rf3.QaR2Pm0ttSc0xQRL3xfV1eX2ux3.d7s7TH09uEndNVcbAXa0t1flrpjEPZA528S  
jegAwlmHjzQX9hqzL/:18788:0:99999:7:::  
root@kali2:~# unshadow /etc/passwd /etc/shadow > hashes.txt  
root@kali2:~# cat hashes.txt | grep fake* > hashes2.txt  
root@kali2:~# cat hashes2.txt  
fake3:$6$IA0bptM$ptTcLQHnJ05n0k8wh0p3cLrKHRhufTA3xhuHujhLbytUtEwLsoyN0$Xsw5P1hxAIucfPCQ  
xv.gRwQR2xheW7Lr1:1000:1001::/home/fake3:/bin/sh  
fake4:$6$xmimquy7$3Rf3.QaR2Pm0ttSc0xQRL3xfV1eX2ux3.d7s7TH09uEndNVcbAXa0t1flrpjEPZA528S  
jegAwlmHjzQX9hqzL/:1001:1002::/home/fake4:/bin/sh  
root@kali2:~#
```



```
root@kali2: ~  
File Edit View Search Terminal Help  
fake4:$6$xmimquy7$3Rf3.QaR2Pm0ttSc0xQRL3xfV1eX2ux3.d7s7TH09uEndNVcbAXa0t1flrpjEPZA528SjegAwlmHjzQX9hqzL/:1001:10  
02::/home/fake4:/bin/sh  
root@kali2:~# john  
John the Ripper password cracker, version 1.8.0.6-jumbo-1-bleeding_omp [linux-gnu 64-bit SSE2+autoconf]  
Copyright (c) 1996-2015 by Solar Designer and others  
Homepage: http://www.openwall.com/john/  
  
Usage: john [OPTIONS] [PASSWORD-FILES]  
--single[=SECTION] "single crack" mode  
--wordlist[=FILE] --stdin wordlist mode, read words from FILE or stdin  
--pipe like --stdin, but bulk reads, and allows rules  
--loopback[=FILE] --pipe like --wordlist, but fetch words from a .pot file  
--dupe-suppression suppress all dupes in wordlist (and force preload)  
--prince[=FILE] PRINCE mode, read words from FILE  
--encoding=NAME input encoding (eg. UTF-8, ISO-8859-1). See also  
doc/ENCODING and --list-hidden-options.  
--rules[=SECTION] enable word mangling rules for wordlist modes  
--incremental[=MODE] "Incremental" mode [using section MODE]  
--mask=MASK mask mode using MASK  
--markov[=OPTIONS] "Markov" mode (see doc/MARKOV)  
--external=MODE external mode or word filter  
--stdout[=LENGTH] just output candidate passwords [cut at LENGTH]  
--restore[=NAME] restore an interrupted session [called NAME]  
--session=NAME give a new session the NAME  
--status[=NAME] print status of a session [called NAME]  
--make-charset=FILE make a charset file. It will be overwritten  
--show[=LEFT] show cracked passwords [if =LEFT, then uncracked]  
--test[=TIME] run tests and benchmarks for TIME seconds each  
--users[=-]LOGIN|UID[,...] [do not] load this (these) user(s) only  
--groups[=-]GID[,...] load users [not] of this (these) group(s) only  
--shells[=-]SHELL[,...] load users with[out] this (these) shell(s) only  
--salts[=-]COUNT[:MAX] load salts with[out] COUNT [to MAX] hashes  
--save-memory=LEVEL enable memory saving, at LEVEL 1..3  
--node=MIN[:MAX]/TOTAL this node's number range out of TOTAL count  
--fork=N fork N processes  
--pot=NAME pot file to use  
--list=WHAT list capabilities, see --list=help or doc/OPTIONS
```


4. Lab 5: Password cracking with John the Ripper

Using john the ripper to crack the passwords in the hashes2.txt file and then showing them in the terminal. Now is the time to use hashcat and crack them in another way but first let's see the possible commands available.

```
Applications ▾ Places ▾ $Terminal ▾ Thu 02:21
root@Kali2: ~

File Edit View Search Terminal Help

--stdout[=LENGTH]      just output candidate passwords [cut at LENGTH]
--restore[=NAME]        restore an interrupted session [called NAME]
--session=NAME          give a new session the NAME
--status[=NAME]         print status of a session [called NAME]
--make-charset=FILE     make a charset file. It will be overwritten
--show[=LEFT]           show cracked passwords [if =LEFT, then uncracked]
--test[=TIME]           run tests and benchmarks for TIME seconds each
--users=[-]LOGIN|UID[,...] (do not) load this (these) user(s) only
--groups=[-]GID[,...]   load users [not] of this (these) group(s) only
--shells=[-]SHELL[,...] load users with[out] this (these) shell(s) only
--salts=[-]COUNT[:MAX] load salts with[out] COUNT [to MAX] hashes
--save-memory=LEVEL     enable memory saving, at LEVEL 1..3
--node=MIN[-MAX]/TOTAL this node's number range out of TOTAL count
--fork=N                fork N processes
--pot=NAME              pot file to use
--list=WHAT             list capabilities, see --list=help or doc/OPTIONS
--format=NAME           force hash of type NAME. The supported formats can
                        be seen with --list=formats and --list=subformats

root@Kali2:~# john -wordlist=/usr/share/john/password.lst hashes2.txt
Warning: detected hash type "sha512crypt", but the string is also recognized as "crypt"
Use the "--format=crypt" option to force loading these as that type instead
Using default input encoding: UTF-8
Loaded 2 password hashes with 2 different salts (sha512crypt, crypt(3) $6$ [SHA512 128/128 SSE2 2x])
Warning: OpenMP is disabled; a non-OpenMP build may be faster
Press 'q' or Ctrl-C to abort, almost any other key for status
password      (fake4)
123456        (fake3)
2g 0:00:00:02 DONE (2021-06-10 02:20) 0.8771g/s 224.5p/s 449.1c/s 449.1C/s 123456..crawlford
Use the "--show" option to display all of the cracked passwords reliably
Session completed
root@Kali2:~# john --show hashes2.txt
fake3:123456:1000:1001::/home/fake3:/bin/sh
fake4:password:1001:1002::/home/fake4:/bin/sh

2 password hashes cracked, 0 left
root@Kali2:~#
```

```
Applications ▾ Places ▾ $Terminal ▾ Thu 02:23
root@Kali2: ~

File Edit View Search Terminal Help

fake4:password:1001:1002::/home/fake4:/bin/sh

2 password hashes cracked, 0 left
root@Kali2:~# hashcat -help | more
hashcat, advanced password recovery

Usage: hashcat [options] hashfile [mask]wordfiles[directories]

=====
Options
=====

+ General:

-m, --hash-type=NUM      Hash-type, see references below
-a, --attack-mode=NUM    Attack-mode, see references below
-V, --version            Print version
-h, --help              Print help
-q, --quiet              Suppress output

+ Benchmark:

-b, --benchmark          Run benchmark

+ Misc:

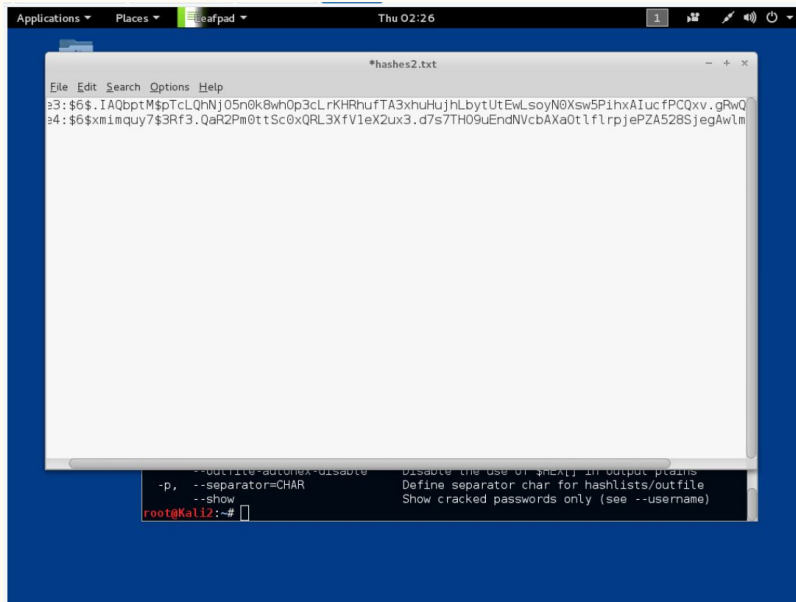
--hex-salt               Assume salt is given in hex
--hex-charset            Assume charset is given in hex
--runtime=NUM            Abort session after NUM seconds of runtime
--status                Enable automatic update of the status-screen
--status-timer=NUM       Seconds between status-screen update
--status-automat         Display the status view in a machine readable format

+ Files:

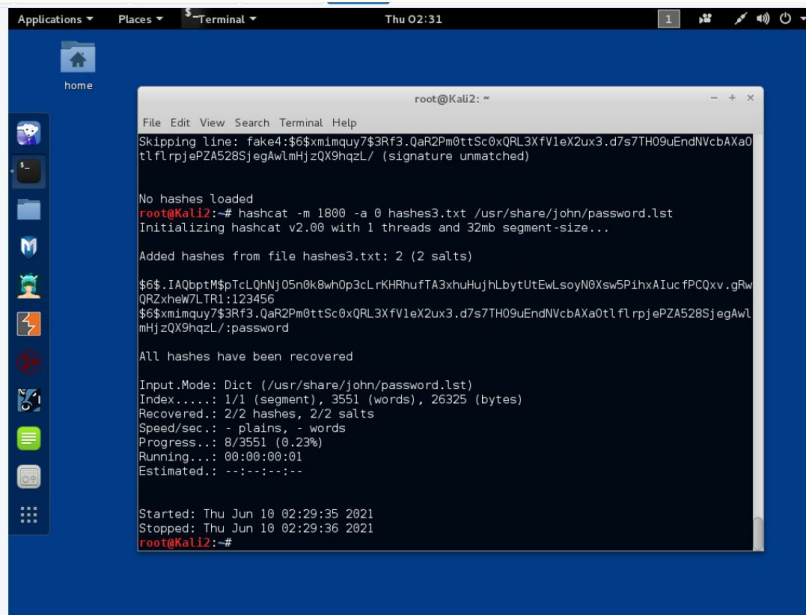
-o, --outfile=FILE       Define outfile for recovered hash
--outfile-format=NUM     Define outfile-format for recovered hash, see references below
```

4. Lab 5: Password cracking with John the Ripper

To use the hashcat, the file hashes3 is created with only the hash password and then the cracking command is used with the list of john the ripper and then the message is displayed showing the passwords.



```
File Edit Search Options Help
33: $6$.IAQbptM$ptcLQhNj05n0k8wh0p3cLrKHRhufTA3xhuHujhLbytUtEwLsoyN0Xsw5P1hxAIucfPCQxv.gRwQ
34: $6$mimquy7$3Rf3.QaR2Pm0ttSc0xQL3XfV1eX2ux3.d7s7TH09uEndNVcbAXa0t1flrpjePZA528SjegAwLmHjzQX9hqzL/
```



```
File Edit View Search Terminal Help
Skipping line: fake4:$6$mimquy7$3Rf3.QaR2Pm0ttSc0xQL3XfV1eX2ux3.d7s7TH09uEndNVcbAXa0t1flrpjePZA528SjegAwLmHjzQX9hqzL/ (signature unmatched)

No hashes loaded
root@Kali2:~# hashcat -m 1800 -a 0 hashes3.txt /usr/share/john/password.lst
Initializing hashcat v2.00 with 1 threads and 32mb segment-size...

Added hashes from file hashes3.txt: 2 (2 salts)

$6$.IAQbptM$ptcLQhNj05n0k8wh0p3cLrKHRhufTA3xhuHujhLbytUtEwLsoyN0Xsw5P1hxAIucfPCQxv.gRwQ
$6$mimquy7$3Rf3.QaR2Pm0ttSc0xQL3XfV1eX2ux3.d7s7TH09uEndNVcbAXa0t1flrpjePZA528SjegAwLmHjzQX9hqzL:password

All hashes have been recovered

Input.Mode: Dict (/usr/share/john/password.lst)
Index.....: 1/1 (segment), 3551 (words), 26325 (bytes)
Recovered.: 2/2 hashes, 2/2 salts
Speed/sec.: - plains, - words
Progress..: 0/3551 (0.23%)
Running...: 00:00:00:01
Estimated.: ---:---:---

Started: Thu Jun 10 02:29:35 2021
Stopped: Thu Jun 10 02:29:36 2021
root@Kali2:~#
```

4. Lab 5: Password cracking with John the Ripper

For this lab it is fundamental to understand that these commands are just for educational purposes and can easily be an infringement of the law.

John the Ripper operates in a different way from hashcat as it's more of a brute force attack to crack the passwords while hashcat is used to confront hashes till it finds the right password.

JtR can be more direct as it does not require to have the hash files of the passwords; they can still be recovered with a precise attack to the user.

Unfortunately most of the passwords are still easy combinations of words and not many users use a password that fully complies with the advised entropy.

References:

- <https://moodle.bcu.ac.uk/course/view.php?id=79458>
- <https://netlab.catcemea.org.uk/home.cgi>
- <https://www.reddit.com/>
- <https://help.yahoo.com/kb/SLN35642.html>

The End
