

Logging Exercise 5....Come Grep Some!

Lets explore “grep” a little more.

First I have a passwords.txt file from <https://dazzlepod.com/uniqpass/>

Lets take a look using “cat”

```
chris@bored:~$ cat passwords.txt
12342008
12342009
123420249
12342040
123421
1234213
123422
12342200012
123422123422
12342222
1234222wkj
12342234
1234223432344234
123423
12342313
12342323
12342324
1234234
123423434
```



```
zzzzzzzzzz
zzzzzzzz95
zzzzzzzzzz
zzzzzzzzzz
zzzzzzzzzz
zzzzzzzzzz1
zzzzzzzzzzx
zzzzzzzzzz
zzzzzzzzzz
zzzzzzzzzzzz
zzzzzzzzzzzz
zzzzzzzzzzzzzz
zzzzzzzzzzzzzzzzzzzzzz
zzzzzzzzzzzzzzzzzzzzzz
chris@bored:~$ |
```

2 Hours Later

Well, that was a bit much. Other reader options are Head, Tail, Less, and More. If curious test them out and see how it goes.

Next, lets try a “grep” variation and use “fgrep” for 123456.

```
chris@bored:~$ fgrep "123456" passwords.txt |
```



```
zzw123456  
zzx123456  
zzx123456789  
zzxx123456  
zzy123456  
zzy123456789  
zzz123456  
zzz.123456  
ZZZ123456  
zzz123456789  
zzzz123456  
zzzzzz123456
```

By using fgrep we are now searching by string vs pattern. For instance “.” will no longer work as a wildcard.

## No results

```
chris@bored:~$ fgrep "123.56" passwords.txt
chris@bored:~$ |
```

## Results

```
chris@bored:~$ grep "123.56" passwords.txt
000000123456
00000123456
0000123456
000123456
0001234560
000123456123
0001234567
00012345678
000123456789
000xz123456
001123456
00123456
```

Ok, lets go and edit passwords.txt using “vim”.  
Then head to the end using “GG” and insert  
“1234567890\_from\_first\_file after the last “Z”.  
Once done we will save and quit.

```
chris@bored:~$ vim passwords.txt
```

```
ZZZZZZZZZZZZ  
ZZZZZZZZZZZZ  
ZZZZZZZZZZZZZ  
ZZZZZZZZZZZZZZ  
ZZZZZZZZZZZZZZZZZZZZZZ  
ZZZZZZZZZZZZZZZZZZZZ1234567890_from_first_file  
:wq|
```

Next, lets copy and name the new copy “passwords2.txt” and quick check with “ls”

```
chris@bored:~$ cp passwords.txt passwords2.txt
chris@bored:~$ ls -l pass*
-rw-rw-r-- 1 chris chris 20159803 Mar 20 02:03 passwords2.txt
-rw-rw-r-- 1 chris chris 20159803 Mar 20 02:02 passwords.txt
```

Now, lets repeat a little and use “vim” on passwords2.txt and change “first” to “second”.

```
chris@bored:~$ vim passwords2.txt
zzzzzzzzzzx
zzzzzzzzzzz
ZZZZZZZZZZZ
zzzzzzzzzzz
zzzzzzzzzzzz
zzzzzzzzzzzzz
zzzzzzzzzzzzz
zzzzzzzzzzzzzz
zzzzzzzzzzzzzzzzzzzzzz
ZZZZZZZZZZZZZZZZZZZZ1234567890_from_second_file
:wq|
```

Ok, lets fgrep 1234567890\_ passw\*

If you have returns from both files then congrats!

```
chris@bored:~$ fgrep 1234567890_ passw*
passwords2.txt:1234567890_
passwords2.txt:ZZZZZZZZZZZZZZZZZZZZ1234567890_from_second_file
passwords.txt:1234567890_
passwords.txt:ZZZZZZZZZZZZZZZZZZZZ1234567890_from_first_file
```

Ok, lets now pull the string “1234567890\_” from passwords.txt and redirect(>) it into “passwords3.txt” If successful, “cat” passwords3 and you should have two lines

```
chris@bored:~$ fgrep 1234567890_ passwords.txt > passwords3.txt
chris@bored:~$ cat passwords3.txt
1234567890_
ZZZZZZZZZZZZZZZZZZZZ1234567890_from_first_file
```

Next, lets try adding “-c” to our fgrep command:  
fgrep -c 1234567890\_ passw\*

```
chris@bored:~$ fgrep -c 1234567890_ passw*
passwords2.txt:2
passwords3.txt:2
passwords.txt:2
```

Based on the return it looks like -c = count and the output will let you know how many times it was found.

Ok, lets try -w and -n next and see what we get.

```
chris@bored:~$ fgrep -w 1234567890_ passw*  
passwords2.txt:1234567890_  
passwords3.txt:1234567890_  
passwords.txt:1234567890_
```

Looks like -w outputs the exact match of what we asked.

```
chris@bored:~$ fgrep -n 1234567890_ passw*  
passwords2.txt:126428:1234567890_  
passwords2.txt:2150838:ZZZZZZZZZZZZZZZZZZZZZZ1234567890_from_second_file  
passwords3.txt:1:1234567890_  
passwords3.txt:2:ZZZZZZZZZZZZZZZZZZZZZZ1234567890_from_first_file  
passwords.txt:126428:1234567890_  
passwords.txt:2150838:ZZZZZZZZZZZZZZZZZZZZZZ1234567890_from_first_file
```

-n tells you what line to find the match on.



Lets grep this time and use -E but lets try it with the regex we recently learned. Lets check the passwords.txt file and use the following:

```
grep -E "[0-9]{10}_from"
```

```
chris@bored:~$ grep -E "[0-9]{10}_from" passwords.txt  
ZZZZZZZZZZZZZZZZZZZZZZ1234567890_from_first_file
```

Success, and looks like -E allows us to use regex in the CLI.

Ok, once more but lets wildcard passwords.

```
chris@bored:~$ grep -E "[0-9]{10}_from" passwords*.txt  
passwords2.txt:ZZZZZZZZZZZZZZZZZZZZZZ1234567890_from_second_file  
passwords3.txt:ZZZZZZZZZZZZZZZZZZZZZZ1234567890_from_first_file  
passwords.txt:ZZZZZZZZZZZZZZZZZZZZZZ1234567890_from_first_file
```

Definitely useful options to know.

Alright, last one but lets just check a log file out that I happen to have using VIM.

```
chris@bored:~$ vim authy.log
```

```
Mar 10 00:00:55 server1 sshd[4422]: Received disconnect from 10.0.2.2 port 60950:11: disconnected by user
Mar 10 00:00:55 server1 sshd[4422]: Disconnected from user ajay 10.0.2.2 port 60950
Mar 10 00:00:55 server1 sshd[4366]: pam_unix(sshd:session): session closed for user ajay
Mar 10 00:00:55 server1 systemd-logind[710]: Session 13 logged out. Waiting for processes to exit.
Mar 10 00:00:55 server1 systemd-logind[710]: Removed session 13.
Mar 10 00:00:55 server1 sshd[3878]: Received disconnect from 10.0.2.2 port 61128:11: disconnected by user
Mar 10 00:00:55 server1 sshd[3878]: Disconnected from user ajay 10.0.2.2 port 61128
Mar 10 00:00:55 server1 sshd[3822]: pam_unix(sshd:session): session closed for user ajay
Mar 10 00:00:55 server1 systemd-logind[710]: Session 5 logged out. Waiting for processes to exit.
Mar 10 00:00:55 server1 sshd[4043]: Received disconnect from 10.0.2.2 port 62494:11: disconnected by user
Mar 10 00:00:55 server1 sshd[4043]: Disconnected from user ajay 10.0.2.2 port 62494
Mar 10 00:00:55 server1 sshd[3987]: pam_unix(sshd:session): session closed for user ajay
Mar 10 00:00:55 server1 systemd-logind[710]: Removed session 5.
Mar 10 00:00:55 server1 systemd-logind[710]: Session 7 logged out. Waiting for processes to exit.
Mar 10 00:00:55 server1 systemd-logind[710]: Removed session 7.
Mar 10 00:01:00 server1 systemd-logind[710]: Power key pressed.
Mar 10 00:01:00 server1 systemd-logind[710]: Powering Off...
Mar 10 00:01:00 server1 systemd-logind[710]: System is powering down.
Mar 10 00:23:02 server1 systemd-logind[714]: New seat seat0.
Mar 10 00:23:02 server1 sshd[764]: Server listening on 0.0.0.0 port 22.
Mar 10 00:23:02 server1 sshd[764]: Server listening on :: port 22.
Mar 10 00:23:02 server1 systemd-logind[714]: Watching system buttons on /dev/input/event0 (Power Button)
Mar 10 00:23:02 server1 systemd-logind[714]: Watching system buttons on /dev/input/event1 (Sleep Button)
```

Ok, looks like a bit more going on than passwords.txt

doesn't simply say website stopped and it just logs what has occurred.


So, after looking through the file nothing jumped out to me. We can always search with grep or inside VIM but for what?

What website service did we install? If you don't remember a quick google can provide you options. Some I found were Nginx, Apache, Lighttpd, and more.

I tend to forget about case sensitivity so first I will tell vim to ignore it using the cmd “set ic”

```
:set ic
```

Then I searched for Nginx (no luck) and next Apache and...success!

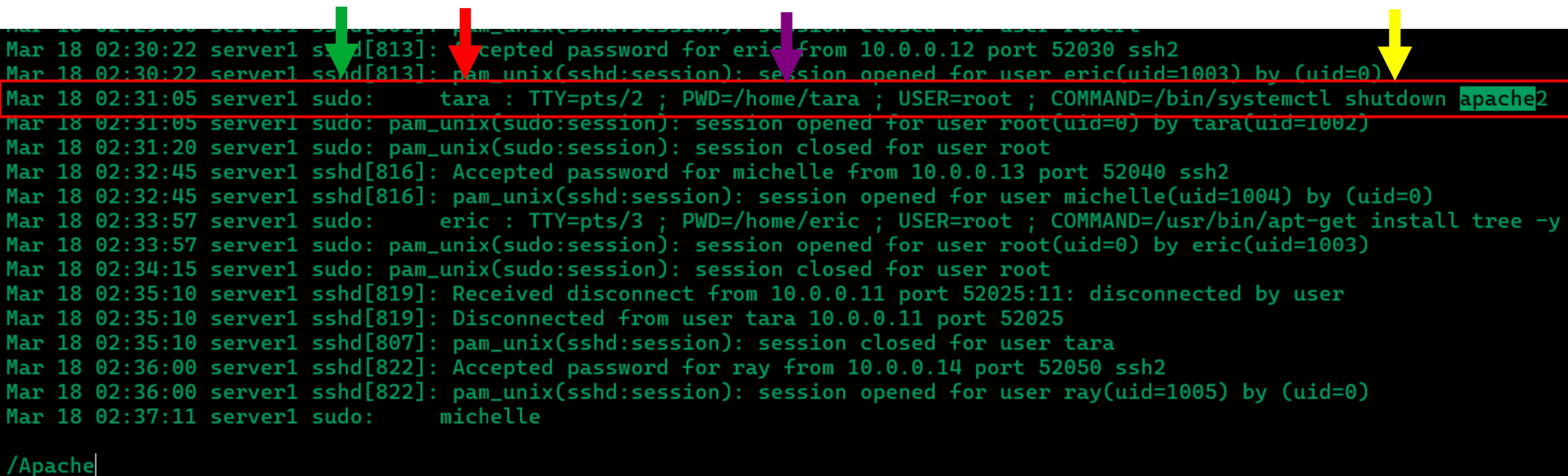


```
Mar 18 02:30:22 server1 sshd[813]: Accepted password for eric from 10.0.0.12 port 52030 ssh2
Mar 18 02:30:22 server1 sshd[813]: pam_unix(sshd:session): session opened for user eric(uid=1003) by (uid=0)
Mar 18 02:31:05 server1 sudo:      tara : TTY=pts/2 ; PWD=/home/tara ; USER=root ; COMMAND=/bin/systemctl shutdown apache2
Mar 18 02:31:05 server1 sudo: pam_unix(sudo:session): session opened for user root(uid=0) by tara(uid=1002)
Mar 18 02:31:20 server1 sudo: pam_unix(sudo:session): session closed for user root
Mar 18 02:32:45 server1 sshd[816]: Accepted password for michelle from 10.0.0.13 port 52040 ssh2
Mar 18 02:32:45 server1 sshd[816]: pam_unix(sshd:session): session opened for user michelle(uid=1004) by (uid=0)
Mar 18 02:33:57 server1 sudo:      eric : TTY=pts/3 ; PWD=/home/eric ; USER=root ; COMMAND=/usr/bin/apt-get install tree -y
Mar 18 02:33:57 server1 sudo: pam_unix(sudo:session): session opened for user root(uid=0) by eric(uid=1003)
Mar 18 02:34:15 server1 sudo: pam_unix(sudo:session): session closed for user root
Mar 18 02:35:10 server1 sshd[819]: Received disconnect from 10.0.0.11 port 52025:11: disconnected by user
Mar 18 02:35:10 server1 sshd[819]: Disconnected from user tara 10.0.0.11 port 52025
Mar 18 02:35:10 server1 sshd[807]: pam_unix(sshd:session): session closed for user tara
Mar 18 02:36:00 server1 sshd[822]: Accepted password for ray from 10.0.0.14 port 52050 ssh2
Mar 18 02:36:00 server1 sshd[822]: pam_unix(sshd:session): session opened for user ray(uid=1005) by (uid=0)
Mar 18 02:37:11 server1 sudo:      michelle
```

```
/Apache|
```

# Now we found the webservice but what happened...?

Looks like **Tara** shut down the **apache2** server from her home directory using **privilege escalation**



A terminal log snippet with four colored arrows pointing to specific lines: a green arrow to the first line, a red arrow to the second line, a purple arrow to the third line, and a yellow arrow to the fourth line. The fourth line is highlighted with a red background and the word 'apache2' is highlighted in green.

```
Mar 18 02:30:22 server1 sshd[813]: Accepted password for eric from 10.0.0.12 port 52030 ssh2
Mar 18 02:30:22 server1 sshd[813]: pam_unix(sshd:session): session opened for user eric(uid=1003) by (uid=0)
Mar 18 02:31:05 server1 sudo:      tara : TTY=pts/2 ; PWD=/home/tara ; USER=root ; COMMAND=/bin/systemctl shutdown apache2
Mar 18 02:31:05 server1 sudo: pam_unix(sudo:session): session opened for user root(uid=0) by tara(uid=1002)
Mar 18 02:31:20 server1 sudo: pam_unix(sudo:session): session closed for user root
Mar 18 02:32:45 server1 sshd[816]: Accepted password for michelle from 10.0.0.13 port 52040 ssh2
Mar 18 02:32:45 server1 sshd[816]: pam_unix(sshd:session): session opened for user michelle(uid=1004) by (uid=0)
Mar 18 02:33:57 server1 sudo:      eric : TTY=pts/3 ; PWD=/home/eric ; USER=root ; COMMAND=/usr/bin/apt-get install tree -y
Mar 18 02:33:57 server1 sudo: pam_unix(sudo:session): session opened for user root(uid=0) by eric(uid=1003)
Mar 18 02:34:15 server1 sudo: pam_unix(sudo:session): session closed for user root
Mar 18 02:35:10 server1 sshd[819]: Received disconnect from 10.0.0.11 port 52025:11: disconnected by user
Mar 18 02:35:10 server1 sshd[819]: Disconnected from user tara 10.0.0.11 port 52025
Mar 18 02:35:10 server1 sshd[807]: pam_unix(sshd:session): session closed for user tara
Mar 18 02:36:00 server1 sshd[822]: Accepted password for ray from 10.0.0.14 port 52050 ssh2
Mar 18 02:36:00 server1 sshd[822]: pam_unix(sshd:session): session opened for user ray(uid=1005) by (uid=0)
Mar 18 02:37:11 server1 sudo:      michelle

/Apache|
```



Don't want to mess around in VIM? That's ok. We just learned more about grep, so lets put it to use.

I'm going to just grep apache and feel confident that will work...

```
chris@bored:~$ grep apache authy.log  
Mar 18 02:31:05 server1 sudo:      tara : TTY=pts/2 ; PWD=/home/tara ; USER=root ; COMMAND=/bin/systemctl shutdown apache2
```

Boom, but don't forget there is always another way, so use what works best for you.

Thanks for playing and see you in the next...