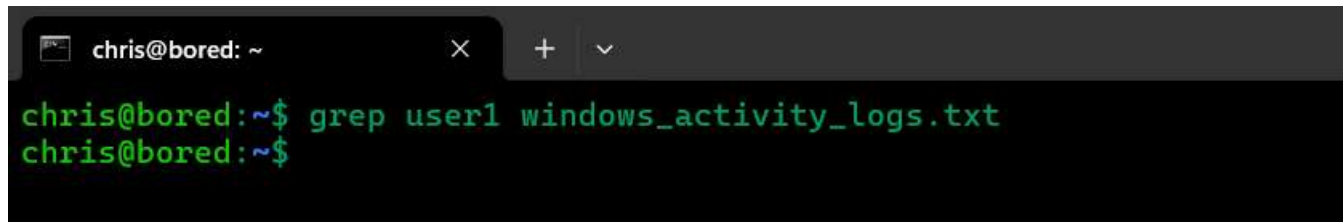


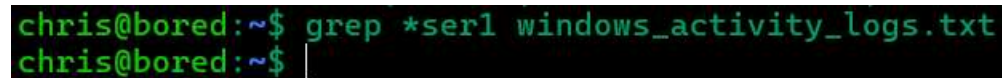
Logs and Grep

Lets start with looking for user1
windows_activity_logs.

A terminal window with a dark background. The title bar shows 'chris@bored: ~' and window control buttons. The terminal text shows a green prompt 'chris@bored:~\$' followed by the command 'grep user1 windows_activity_logs.txt'. A second green prompt 'chris@bored:~\$' is shown on the next line, indicating the command has executed without output.

```
chris@bored: ~  
chris@bored:~$ grep user1 windows_activity_logs.txt  
chris@bored:~$
```

Nothing returned but I know user1
exists. Lets try to wildcard it with *

A terminal window showing a green prompt 'chris@bored:~\$' followed by the command 'grep *ser1 windows_activity_logs.txt'. A second green prompt 'chris@bored:~\$' is shown on the next line with a vertical cursor, indicating the command has executed without output.

```
chris@bored:~$ grep *ser1 windows_activity_logs.txt  
chris@bored:~$ |
```

Again, nothing returned. Lets keep
going and see if we can figure it out.

I know there is a user1 so lets try to ignore case sensitive using -i

```
chris@bored:~$ grep -i user1 windows_activity_logs.txt
2024-02-29 11:32:09, User1, password failure, failed,
2024-02-23 05:40:33, User1, login, success,
2024-02-28 16:41:19, User1, password failure, failed,
2024-03-06 10:46:41, User1, open file, success, Presentation.pptx
2024-02-27 08:04:18, User1, login, success,
2024-03-06 01:53:30, User1, password failure, failed,
2024-02-14 05:19:10, User1, logout, success,
2024-02-11 00:56:20, User1, logout, success,
2024-02-19 22:15:55, User1, edit file, success, Document1.docx
2024-02-27 02:56:50, User1, open file, success, Spreadsheet.xlsx
2024-02-08 07:36:31, User1, open file, success, Presentation.pptx
2024-02-23 22:41:59, User1, password failure, failed,
2024-02-14 10:20:58, User1, open file, success, Spreadsheet.xlsx
2024-02-07 22:44:09, User1, edit file, success, Presentation.pptx
2024-02-08 21:30:30, User1, password failure, failed,
```

Ah, that solved it. User1 does exist but we have to watch case sensitivity.

Why though did * not work...

Don't forget invoking grep switches you out of bash and into regex. Remembering this lets try again but this time lets use "." as the wildcard

```
chris@bored:~$ grep .ser1 windows_activity_logs.txt
2024-02-29 11:32:09, User1, password failure, failed,
2024-02-23 05:40:33, User1, login, success,
2024-02-28 16:41:19, User1, password failure, failed,
2024-03-06 10:46:41, User1, open file, success, Presentation.pptx
2024-02-27 08:04:18, User1, login, success,
2024-03-06 01:53:30, User1, password failure, failed,
2024-02-14 05:19:10, User1, logout, success,
2024-02-11 00:56:20, User1, logout, success,
2024-02-19 22:15:55, User1, edit file, success, Document1.docx
2024-02-27 02:56:50, User1, open file, success, Spreadsheet.xlsx
2024-02-08 07:36:31, User1, open file, success, Presentation.pptx
2024-02-23 22:41:59, User1, password failure, failed,
2024-02-14 10:20:58, User1, open file, success, Spreadsheet.xlsx
2024-02-07 22:44:09, User1, edit file, success, Presentation.pptx
```

Success!

Ok, lets switch to User3 and see if we have logs.

```
chris@bored:~$ grep User3 windows_activity_logs.txt
2024-03-05 08:44:39, User3, edit file, success, Presentation.pptx
2024-02-12 20:35:58, User3, logout, success,
2024-02-29 16:40:38, User3, open file, success, Report.pdf
2024-03-05 13:59:37, User3, login, success,
2024-02-29 05:15:46, User3, logout, success,
2024-02-26 20:02:08, User3, open file, success, Spreadsheet.xlsx
2024-02-23 18:55:26, User3, delete file, success, Report.pdf
2024-02-21 21:27:15, User3, edit file, success, Presentation.pptx
2024-03-01 12:03:40, User3, delete file, success, Presentation.pptx
2024-02-25 19:33:19, User3, open file, success, Report.pdf
2024-03-06 13:12:11, User3, delete file, success, Presentation.pptx
```

Easy day and we do have logs for User3

Next, lets check for document1

```
chris@bored:~$ grep document1 windows_activity_logs.txt
chris@bored:~$ grep document1.docx windows_activity_logs.txt
chris@bored:~$ grep .document1 windows_activity_logs.txt
chris@bored:~$ grep .ocument1 windows_activity_logs.txt
2024-02-10 00:01:56, User4, open file, success, Document1.docx
2024-02-19 21:09:17, User2, open file, success, Document1.docx
2024-02-21 09:50:04, User2, edit file, success, Document1.docx
2024-02-07 22:35:42, User2, edit file, success, Document1.docx
2024-02-21 01:46:12, User4, edit file, success, Document1.docx
2024-02-19 22:15:55, User1, edit file, success, Document1.docx
```

Notice, it took a couple tries to get a return. Initially what went wrong? I should have used -i to avoid a case sensitive issue.

Knowing this I was curious what about .document? Again, case sensitive got me and .Document1 would have worked. Again, .ocument1 worked similar to .ser1 that we did in the beginning.

Now, lets make a copy of log1 and call it log2 using “cp”. Making a copy can be useful because you may need it in the future depending what you find.

A word to think about here is “immutable” or unchanging.

```
chris@bored:~$ ls -l
total 33620
-rw-rw-r-- 1 chris chris 174392 Jan 17 20:38 alice2.txt
-rw-rw-r-- 1 chris chris 174392 Jan 17 17:13 alice.txt
-rw-rw-r-- 1 chris chris 22 Feb 18 02:14 class.txt
-rw-rw-r-- 1 chris chris 306132 Feb 26 20:38 dvl.txt
-rwxrw-r-- 1 chris chris 37 Feb 9 16:42 hello_world.sh
-rw-rw-r-- 1 chris chris 12203080 Feb 26 17:24 lanscan
-rwxrw-r-- 1 chris chris 183 Feb 9 17:17 main.sh
-rw-rw-r-- 1 chris chris 1253948 Jan 17 20:25 moby.txt
-rw-rw-r-- 1 chris chris 20159777 Jan 19 16:59 passwords.txt
-rw-rw-r-- 1 chris chris 5 Jan 17 16:43 rope.txt
-rw-rw-r-- 1 chris chris 69497 Feb 26 17:09 scanme.nmap.org_vulnscan
-rw-rw-r-- 1 chris chris 56075 Mar 7 02:25 windows_activity_logs.txt
chris@bored:~$ cp windows_activity_logs.txt windows_activity_logs2.txt
chris@bored:~$ ls -l
total 33676
-rw-rw-r-- 1 chris chris 174392 Jan 17 20:38 alice2.txt
-rw-rw-r-- 1 chris chris 174392 Jan 17 17:13 alice.txt
-rw-rw-r-- 1 chris chris 22 Feb 18 02:14 class.txt
-rw-rw-r-- 1 chris chris 306132 Feb 26 20:38 dvl.txt
-rwxrw-r-- 1 chris chris 37 Feb 9 16:42 hello_world.sh
-rw-rw-r-- 1 chris chris 12203080 Feb 26 17:24 lanscan
-rwxrw-r-- 1 chris chris 183 Feb 9 17:17 main.sh
-rw-rw-r-- 1 chris chris 1253948 Jan 17 20:25 moby.txt
-rw-rw-r-- 1 chris chris 20159777 Jan 19 16:59 passwords.txt
-rw-rw-r-- 1 chris chris 5 Jan 17 16:43 rope.txt
-rw-rw-r-- 1 chris chris 69497 Feb 26 17:09 scanme.nmap.org_vulnscan
-rw-rw-r-- 1 chris chris 56075 Mar 7 02:25 windows_activity_logs2.txt
-rw-rw-r-- 1 chris chris 56075 Mar 7 02:25 windows_activity_logs.txt
chris@bored:~$
```

Thinking about immutable, how can we prove this? Exactly, check the hash using “md5sum”.

```
chris@bored:~$ md5sum windows_activity_logs*  
8c21974b8df2c0771ba8854b25f20b33 windows_activity_logs2.txt  
8c21974b8df2c0771ba8854b25f20b33 windows_activity_logs.txt  
chris@bored:~$ |
```

Here we see they match, meaning successful copy. Another way to check would be “diff”

```
chris@bored:~$ diff windows_activity_logs*  
chris@bored:~$ |
```

No return = no differences

Though, unlike md5sum this doesn't give us any output. Did it even work?

Lets append logs and add “test” to the end. Remember append using “>>” and then lets use “diff” one more time. We can also use “*” here to simplify it as well.

```
chris@bored:~$ echo test >> windows_activity_logs.txt
chris@bored:~$ diff windows_activity_logs*
1000a1001
> test
chris@bored:~$ |
```

Success, and we see there is a difference. The output is telling us 1000 lines vs 1001 and the word test. This all makes sense as we appended the document with the word test.

One last check and lets use md5sum

```
chris@bored:~$ md5sum windows_activity_logs*
8c21974b8df2c0771ba8854b25f20b33  windows_activity_logs2.txt
0d1581006e42b6acc6453245680adcc5  windows_activity_logs.txt
chris@bored:~$ |
```

Easy confirmation through hash comparison

Next, lets search for Spreadsheet.xlsx in both logs. We will grep for Spreadsheet and then wildcard the logs to search across both files.

```
chris@bored:~$ grep Spreadsheet.xls windows_activity_logs*
windows_activity_logs2.txt:2024-02-26 13:08:59, User4, open file, success, Spreadsheet.xlsx
windows_activity_logs2.txt:2024-02-27 02:56:50, User1, open file, success, Spreadsheet.xlsx
windows_activity_logs2.txt:2024-02-08 13:41:26, User4, open file, success, Spreadsheet.xlsx
windows_activity_logs2.txt:2024-02-14 10:20:58, User1, open file, success, Spreadsheet.xlsx
windows_activity_logs2.txt:2024-02-15 03:37:17, User2, open file, success, Spreadsheet.xlsx
```

```
windows_activity_logs2.txt:2024-02-08 22:58:00, User1, edit file, success, Spreadsheet.xlsx
windows_activity_logs2.txt:2024-02-24 03:34:50, User1, delete file, success, Spreadsheet.xlsx
windows_activity_logs2.txt:2024-02-21 01:08:08, User2, edit file, success, Spreadsheet.xlsx
windows_activity_logs2.txt:2024-02-14 02:08:45, User1, edit file, success, Spreadsheet.xlsx
windows_activity_logs2.txt:2024-03-02 00:11:47, User1, delete file, success, Spreadsheet.xlsx
windows_activity_logs.txt:2024-02-26 13:08:59, User4, open file, success, Spreadsheet.xlsx
windows_activity_logs.txt:2024-02-27 02:56:50, User1, open file, success, Spreadsheet.xlsx
windows_activity_logs.txt:2024-02-08 13:41:26, User4, open file, success, Spreadsheet.xlsx
windows_activity_logs.txt:2024-02-14 10:20:58, User1, open file, success, Spreadsheet.xlsx
```

And...success

Next, lets look for User2 **OR** open file in the log and send it to log1.txt

Use grep to search and “-E” for the or function.

```
chris@bored:~$ grep -E 'User2|open file' windows_activity_logs.txt > log1.txt
```

Quick check using tail and we see open file is there for all users and everything for User2 is there.

```
chris@bored:~$ tail log1.txt
2024-02-22 06:30:07, User2, login, success,
2024-02-06 04:02:09, User4, open file, success, Document1.docx
2024-03-02 07:35:07, User3, open file, success, Report.pdf
2024-02-24 10:41:54, User1, open file, success, Document1.docx
2024-02-07 10:43:29, User1, open file, success, Report.pdf
2024-02-21 05:33:42, User2, delete file, success, Report.pdf
2024-02-21 01:08:08, User2, edit file, success, Spreadsheet.xlsx
2024-02-24 23:22:59, User4, open file, success, Report.pdf
2024-02-06 01:24:00, User4, open file, success, Report.pdf
2024-02-13 23:48:47, User2, edit file, success, Report.pdf
chris@bored:~$ |
```

Lets add onto log1.txt but first lets grab hash (md5sum)
Now, lets search for User1 **AND** failed and add it to log1.txt

```
chris@bored:~$ md5sum log1.txt
6bdc1f877c50c4b6f1797e17a6d1b633  log1.txt
chris@bored:~$ grep 'User1' windows_activity_logs.txt | grep 'failed' >> log1.txt
chris@bored:~$ tail log1.txt
2024-02-29 04:35:01, User1, password failure, failed,
2024-02-24 15:48:03, User1, password failure, failed,
2024-02-10 03:42:13, User1, password failure, failed,
2024-02-07 05:40:37, User1, password failure, failed,
2024-02-06 02:14:42, User1, password failure, failed,
2024-02-11 20:50:46, User1, password failure, failed,
2024-02-18 19:57:14, User1, password failure, failed,
2024-02-24 12:17:08, User1, password failure, failed,
2024-03-06 02:59:47, User1, password failure, failed,
2024-02-11 16:35:41, User1, password failure, failed,
chris@bored:~$ md5sum log1.txt
39853875c3425486dae6fbf5a08d5e68  log1.txt
chris@bored:~$ |
```

Quick check and we see it worked because the hash is
different

Lastly, lets search for User1 and Failed in the windows_activity_logs. Lets wildcard this a bit as well. We will use “.” in grep (regex) and “*” for the filename (bash)

```
chris@bored:~$ grep '.ser1' windows_activity_logs* | grep '.ailed' >> log2.txt
chris@bored:~$
```

Quick check using tail and looks like success.

```
chris@bored:~$ tail log2.txt
2024-02-29 04:35:01, User1, password failure, failed,
2024-02-24 15:48:03, User1, password failure, failed,
2024-02-10 03:42:13, User1, password failure, failed,
2024-02-07 05:40:37, User1, password failure, failed,
2024-02-06 02:14:42, User1, password failure, failed,
2024-02-11 20:50:46, User1, password failure, failed,
2024-02-18 19:57:14, User1, password failure, failed,
2024-02-24 12:17:08, User1, password failure, failed,
2024-03-06 02:59:47, User1, password failure, failed,
2024-02-11 16:35:41, User1, password failure, failed,
chris@bored:~$
```