LinuxFest Northwest

2016

Digging through the logs

The scenario

'Someone's sending us tons of requests, it should all be coming from one IP, but which one???'

The solution

```
`cat output.txt | grep \\s[0-9]{3}\.
[0-9]{3}\\s -oE | sort | uniq -c`
```

- bye!

this is what you need to know

- most cool features of the tools here presented will go un-mentioned

- there are like 80 ways to do this, and if I accept mine is the worst, can we move on?

I hope you already know these things

- files and directories... exist

let's make a big text file

```
`ls -l /usr/bin > output.txt`
```

how big is it?

`ls -l`

works okay to do this, but let's use something that just gives a sense of size

`wc output.txt`

that gives us lines, words, and bytes

let's make it bigger

```
`ls -l /usr/bin > output.txt`
```

```
`wc output.txt`
```

That… didn't work

```
`ls -l /usr/bin >> output.txt`
```

```
`wc output.txt`
```

checkpoint

1. make new empty directory

`mkdir practice && cd practice`

2. create a long text file with the output of

`ls -l /usr/bin > output.txt`

3. append the same output to the file once or twice with

`>>`

4. get a sense of how big it is with

`wc`

read the file we made

```
`less output.txt`
```

less

return - go forward a line

space - page

q - quarrel with the program until it closes in frustration

let's search for something in our big list

```
`less output.txt | grep zip`
```

grep

- only print lines that match your search

- like all the other tools we've used, it's more powerful than it seems

> and | are different!

one sends output to a file >, the other pipes it to a second command |

```
cd /usr/bin
```

```
ls > less
```

If you're very unlucky, the output you're sending will overwrite the program.

just gimme a taste

```
`less output.txt | grep zip | tail -n
5`
```

- get the last five lines

- protip: up arrow to re-use your last command

what about this -n -l stuff?

- most commands can be run with options, usually given as `-n`

- to set two options, like to print n lines `-n` in reverse order `-r`, use `-nr`

checkpoint

1. view output.txt with

`less`

2. show only the lines containing 'zip'

`less output.txt | grep zip`

3. get just the last few lines with

`tail -n 5`

4. (if you have time) try changing the order of functions and see the result

aren't those duplicate lines annoying?

`less output.txt | grep zip | uniq`

- if you say it out loud it makes sense

We're done!

- I've given you all the tools to save your server

- bye!

back to our scenario

'Someone's sending us tons of requests, it should all be coming from one IP, but which one???'

okay, how do I have the tools to handle that?

- what's the most common year on the files in output.txt?

let's make grep find a year

- What would a year look like in our results?

- [audience participation]

grep's big secret: regex

- regex finds matches one character at a time

- regex doesn't do math

Find all the numbers that start with 1 or 2

- `less output.txt | grep less output.txt | grep [21]`

- just all the lines with a 1... or a 2... not super helpful

we want it to *start* with a 1 or 2

- `less output.txt | grep less output.txt | grep -E \\s[21]`

- `-E` for extended regex

- special regex symbol: `\s` for 'space'

- you might not need two slashes: `grep -E \s[21]`

then 9 or 0

`grep -E \\s[21][90]`

- not too surprising

do we haaaave to write out [1234567890]?

- nope!

```
`grep -E \\s[21][90][0-9]`
```

do we haaaaaaaave to write [0-9] again?

`grep -E \\s[21][90][0-9]{2}`

- {n} = 'match that last thing exactly n times'

what exactly is grep matching?

```
`grep -oE \\s[21][90][0-9]{2}`
```

checkpoint

1. filter results with `grep`

2. write an expression that matches 1990 - 2016 (extras ok)

3. require a space before and after your numbers `\s`

4. print **just** the matched part of the lines

```
`grep -oE \\s[21][90][0-9]{2}`
```

Time to learn about uniq's super power

```
`less output.exe | grep -oE \\s[21][90]
[0-9]{2} | uniq -c`
```

One quick tweak

- I am not 100% certain why, but you gotta sort this first, maybe uniq just hits neighbors? idk…

```
`less output.exe | grep -oE \\s[21][90][0-9]{2} | sort | uniq -c`
```

okay!

- *now* you have all the tools you need

Evaluate this session

“Digging Through the Logs” by Toby Fee

Session videos will be posted to the LinuxFest Northwest YouTube channel.

Thank you!