# Filter Contents

In the last section, we learned about the redirections we can use to redirect results from one program to another for processing. To read files, we do not necessarily have to use an editor for that. There are two tools called `more` and `less`, which are very identical. These are fundamental `pagers` that allow us to scroll through the file in an interactive view. Let us have a look at some examples.

## More

```
amit8986@htb[/htb]$ more /etc/passwd
```

After we read the content using `cat` and redirected it to `more`, the already mentioned `pager` opens, and we will automatically start at the beginning of the file.

```
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
<SNIP>
--More--
```

With the `[Q]` key, we can leave this `pager`. We will notice that the output remains in the terminal.

## Less

If we now take a look at the tool `less`, we will notice on the man page that it contains many more features than `more`.

```
amit8986@htb[/htb]$ less /etc/passwd
```

The presentation is almost the same as with `more`.

```
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
<SNIP>
:
```

When closing `less` with the `[Q]` key, we will notice that the output we have seen, unlike `more`, does not remain in the terminal.

## Head

Sometimes we will only be interested in specific issues either at the beginning of the file or the end. If we only want to get the `first` lines of the file, we can use the tool `head`. By default, `head` prints the first ten lines of the given file or input, if not specified otherwise.

```
amit8986@htb[/htb]$ head /etc/passwd

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
```

## Tail

If we only want to see the last parts of a file or results, we can use the counterpart of `head` called `tail`, which returns the `last` ten lines.

```
amit8986@htb[/htb]$ tail /etc/passwd

miredo:x:115:65534::/var/run/miredo:/usr/sbin/nologin
usbmux:x:116:46:usbmux daemon,,,:/var/lib/usbmux:/usr/sbin/nologin
rtkit:x:117:119:RealtimeKit,,,:/proc:/usr/sbin/nologin
nm-openvpn:x:118:120:NetworkManager OpenVPN,,,:/var/lib/openvpn/chroot:/usr/sbin/nologin
nm-openconnect:x:119:121:NetworkManager OpenConnect plugin,,,:/var/lib/NetworkManager:/usr/sbin/nologin
pulse:x:120:122:PulseAudio daemon,,,:/var/run/pulse:/usr/sbin/nologin
beef-xss:x:121:124::/var/lib/beef-xss:/usr/sbin/nologin
lightdm:x:122:125:Light Display Manager:/var/lib/lightdm:/bin/false
do-agent:x:998:998::/home/do-agent:/bin/false
user6:x:1000:1000:,,,:/home/user6:/bin/bash
```

## Sort

Depending on which results and files are dealt with, they are rarely sorted. Often it is necessary to sort the desired results alphabetically or numerically to get a better overview. For this, we can use a tool called `sort`.

```
amit8986@htb[/htb]$ cat /etc/passwd | sort

_apt:x:104:65534::/nonexistent:/usr/sbin/nologin
backup:x:34:34:backup:/var/backups:/usr/sbin/nologin
bin:x:2:2:bin:/bin:/usr/sbin/nologin
cry0l1t3:x:1001:1001::/home/cry0l1t3:/bin/bash
daemon:x:1:1:daemon:/usr/sbin:/usr/sbin/nologin
dnsmasq:x:107:65534:dnsmasq,,,:/var/lib/misc:/usr/sbin/nologin
dovecot:x:114:117:Dovecot mail server,,,:/usr/lib/dovecot:/usr/sbin/nologin
dovenull:x:115:118:Dovecot login user,,,:/nonexistent:/usr/sbin/nologin
ftp:x:113:65534::/srv/ftp:/usr/sbin/nologin
games:x:5:60:games:/usr/games:/usr/sbin/nologin
gnats:x:41:41:Gnats Bug-Reporting System (admin):/var/lib/gnats:/usr/sbin/nologin
htb-student:x:1002:1002::/home/htb-student:/bin/bash
<SNIP>
```

As we can see now, the output no longer starts with root but is now sorted alphabetically.

## Grep

More often, we will only search for specific results that contain patterns we have defined. One of the most used tools for this is `grep`, which offers many different features. Accordingly, we can search for users who have the default shell "`/bin/bash`" set as an example.

```
amit8986@htb[/htb]$ cat /etc/passwd | grep "/bin/bash"

root:x:0:0:root:/root:/bin/bash
mrb3n:x:1000:1000:mrb3n:/home/mrb3n:/bin/bash
cry0l1t3:x:1001:1001::/home/cry0l1t3:/bin/bash
htb-student:x:1002:1002::/home/htb-student:/bin/bash
```

Another possibility is to exclude specific results. For this, the option "`-v`" is used with `grep`. In the next example, we exclude all users who have disabled the standard shell with the name "`/bin/false`" or "`/usr/bin/nologin`".

```
amit8986@htb[/htb]$ cat /etc/passwd | grep -v "false\|nologin"

root:x:0:0:root:/root:/bin/bash
sync:x:4:65534:sync:/bin:/bin/sync
postgres:x:111:117:PostgreSQL administrator,,,:/var/lib/postgresql:/bin/bash
user6:x:1000:1000:,,,:/home/user6:/bin/bash
```

## Cut

Specific results with different characters may be separated as delimiters. Here it is handy to know how to remove specific delimiters and show the words on a line in a specified position. One of the tools that can be used for this is `cut`. Therefore we use the option "`-d`" and set the delimiter to the colon character (`:`) and define with the option "`-f`" the position in the line we want to output.

```
amit8986@htb[/htb]$ cat /etc/passwd | grep -v "false\|nologin" | cut -d":" -f1

root
sync
mrb3n
cry0l1t3
htb-student
```

## Tr

Another possibility to replace certain characters from a line with characters defined by us is the tool `tr`. As the first option, we define which character we want to replace, and as a second option, we define the character we want to replace it with. In the next example, we replace the colon character with space.

```
amit8986@htb[/htb]$ cat /etc/passwd | grep -v "false\|nologin" | tr ":" " "

root x 0 0 root /root /bin/bash
sync x 4 65534 sync /bin /bin/sync
mrb3n x 1000 1000 mrb3n /home/mrb3n /bin/bash
cry0l1t3 x 1001 1001  /home/cry0l1t3 /bin/bash
htb-student x 1002 1002  /home/htb-student /bin/bash
```

## Column

Since such results can often have an unclear representation, the tool `column` is well suited to display such results in tabular form using the "`-t`."

```
amit8986@htb[/htb]$ cat /etc/passwd | grep -v "false\|nologin" | tr ":" " " | column -t

root         x  0     0      root               /root        /bin/bash
sync         x  4     65534  sync               /bin         /bin/sync
mrb3n        x  1000  1000   mrb3n              /home/mrb3n  /bin/bash
cry0l1t3     x  1001  1001   /home/cry0l1t3     /bin/bash
htb-student  x  1002  1002   /home/htb-student  /bin/bash
```

## Awk

As we may have noticed, the user "`postgres`" has one row too many. To keep it as simple as possible to sort out such results, the (`g`)`awk` programming is beneficial, which allows us to display the first (`$1`) and last (`$NF`) result of the line.

```
amit8986@htb[/htb]$ cat /etc/passwd | grep -v "false\|nologin" | tr ":" " " | awk '{print $1, $NF}'

root /bin/bash
sync /bin/sync
mrb3n /bin/bash
cry0l1t3 /bin/bash
htb-student /bin/bash
```

## Sed

There will come moments when we want to change specific names in the whole file or standard input. One of the tools we can use for this is the stream editor called `sed`. One of the most common uses of this is substituting text. Here, `sed` looks for patterns we have defined in the form of regular expressions (regex) and replaces them with another pattern that we have also defined. Let us stick to the last results and say we want to replace the word "`bin`" with "`HTB`."

The "`s`" flag at the beginning stands for the substitute command. Then we specify the pattern we want to replace. After the slash (`/`), we enter the pattern we want to use as a replacement in the third position. Finally, we use the "`g`" flag, which stands for replacing all matches.

```
amit8986@htb[/htb]$ cat /etc/passwd | grep -v "false\|nologin" | tr ":" " " | awk '{print $1, $NF}' | sed 's/bin/HTB

root /HTB/bash
sync /HTB/sync
mrb3n /HTB/bash
cry0l1t3 /HTB/bash
htb-student /HTB/bash
```

## Wc

Last but not least, it will often be useful to know how many successful matches we have. To avoid counting the lines or characters manually, we can use the tool `wc`. With the "`-l`" option, we specify that only the lines are counted.

```
amit8986@htb[/htb]$ cat /etc/passwd | grep -v "false\|nologin" | tr ":" " " | awk '{print $1, $NF}' | wc -l

5
```

## Practice

It may be a bit overwhelming at first to deal with so many different tools and their functions if we are not familiar with them. Take your time and experiment with the tools. Have a look at the man pages (`man <tool>`) or call the help for it (`<tool> -h` / `<tool> --help`). The best way to become familiar with all the tools is to practice. Try to use them as often as possible, and we will be able to filter many things intuitively after a short time.

Here are a few optional exercises we can use to improve our filtering skills and get more familiar with the terminal and the commands. The file we will need to work with is the `/etc/passwd` file on our `target` and we can use any shown command above. Our goal is to filter and display only specific contents. Read the file and filter its contents in such a way that we see only:

1. A line with the username `cry0l1t3`.
2. The usernames.
3. The username `cry0l1t3` and his UID.
4. The username `cry0l1t3` and his UID separated by a comma (`,`).
5. The username `cry0l1t3`, his UID, and the set shell separated by a comma (`,`).
6. All usernames with their UID and set shells separated by a comma (`,`).
7. All usernames with their UID and set shells separated by a comma (`,`) and exclude the ones that contain `nologin` or `false`.
8. All usernames with their UID and set shells separated by a comma (`,`) and exclude the ones that contain `nologin` and count all lines of the filtered output.

VPN Servers

⚠ Warning: Each time you "Switch", your connection keys are regenerated and you must re-download your VPN connection file.

All VM instances associated with the old VPN Server will be terminated when switching to a new VPN server.

Existing PwnBox instances will automatically switch to the new VPN server.

eu-academy-1 ⌄

PROTOCOL

● UDP 1337 ○ TCP 443

DOWNLOAD VPN CONNECTION FILE

Start Instance

0 / 1 spawns left

Waiting to start...

## Questions

Cheat Sheet

Download VPN Connection File

Answer the question(s) below to complete this Section and earn cubes!

Target: 10.129.55.110

Life Left: 49 minutes +

SSH to 10.129.55.110 with user "htb-student" and password "HTB_@cademy_stdnt!"

+ 0 How many services are listening on the target system on all interfaces? (Not on localhost and IPv4 only)

Submit your answer here...

Submit

+ 0 Determine what user the ProFTPd server is running under. Submit the username as the answer.

Submit your answer here...

Submit

+ 1 Use cURL from your Pwnbox (not the target machine) to obtain the source code of the "https://www.inlanefreight.com" website and filter all unique paths of that domain. Submit the number of these paths as the answer.