Linux Command Line Cheat Sheet

Abstract

The following examples may be typed in the Security511 Linux VM, and copy/paste will work fine (be sure to omit the prompt).

- To copy in Firefox: press CTRL-C
- To paste into a terminal: press SHIFT-CTRL-V (or Edit->Paste)

Many of these examples will use the "cat example.txt | command" syntax. This is safer than the equivalent syntax of "command < example.txt".

Why? Most everyone learning the Unix/Linux commandline has accidentally reversed the "<" sign (read) with the ">" sign (write), accidentally overwriting a file. The syntax of "cat example.txt | command" is therefore safer. Please feel free to use whatever syntax you are most comfortable with.

On a related note, "There is more than one way to do it," as Larry Wall once said. You may come up with different ways to perform the following, and perhaps better ways as well. Feel free to share your CLI Kung Fu with your instructor!

Where to Acquire

These tools are installed natively in most Unix/Linux distributions, as well as OS X.

Examples/Use Case

- awk
- checksum tools
- cut
- file
- grep
- head
- sed
- sort
- wc
- xxd

---

awk

Print the length of each line of a file (/etc/passwd in this case), followed by the line itself:

```
$ cat /etc/passwd | awk '{print length, $0;}'
```

Print the 2nd field from a file using the string 'Mozilla/' as a delimiter:

```
$ cat /var/log/apache2/access.log | awk -F "Mozilla/" '{print $2}'
```

Print the last period delimited field

```
$ cat domains.txt | awk -F "." '{print $(NF)}'
```

---

checksum tools

Generate the MD5 checksum of a file:

```
$ md5sum /etc/passwd
```

Generate the SHA1 checksum of a file. The three following commands are equivalent:

```
$ sha1sum /etc/passwd
$ shasum /etc/passwd
$ shasum -a1 /etc/passwd
```

Generate the SHA-256 checksum of a file:

```
$ shasum -a256 /etc/passwd
```

Generate the SHA-512 checksum of a file:

```
$ shasum -a512 /etc/passwd
```

---

cut

Cut the 2nd field from a file, using the space as a delimiter:

```
$ cat /var/log/dpkg.log | cut -d' ' -f2
```

Cut the 6th field from a file, using the colon as a delimiter:

```
$ cat /etc/passwd | cut -d: -f6
```

Cut the 2nd and 3rd field from a file, use the comma as a delimiter:

```
$ cat /labs/honeytokens/pilots.csv | cut -d, -f2-3
```

Cut beginning at the 7th field, to end of line, using the space as a delimiter:

```
$ cat /var/log/dpkg.log | cut -d' ' -f7-
```

Cut the 6th field, using the double-quote (“) as a delimiter, and escaping it to treat it as a literal character:

```
$  cat  /var/log/apache2/access.log  | cut -d\" -f6
```

Cut the beginning at the 11th character, to end of line:

```
$ ifconfig | cut -c11-
```

---

file

Determine the file type, using the file's magic bytes:

```
$ file /usr/local/bin/*
```

---

grep

Search for lines containing the string “bash”, case sensitive:

```
$ grep bash /etc/passwd
```

Search for lines containing the string “bash”, case insensitive:

```
$ grep -i bash /etc/passwd
```

Search for lines that do not contain the string “bash”, case insensitive:

```
$ grep -vi bash /etc/passwd
```

Search for lines containing the string “root”, case sensitive, plus print the next 5 lines:

```
$ grep -A5 root /etc/passwd
```

---

head

Print the first 10 lines of a file:

```
$ head -n 10 /etc/passwd
```

---

sed

grep for lines containing “Mozilla”, then change “Mozilla” to “MosaicKilla”:

```
$ grep Mozilla /var/log/apache2/access.log | sed "s/Mozilla/MosaicKilla/g"
```

grep for lines containing “Mozilla”, then delete all characters up to and including “Mozilla”:

```
$ grep Mozilla /var/log/apache2/access.log | sed "s/^.*Mozilla//g"
```

grep for lines containing “Mozilla”, then delete all characters that precede “Mozilla”:

```
$ grep Mozilla /var/log/apache2/access.log | sed "s/^.*Mozilla/Mozilla/g"
```

---

sort

The following examples will run strings on a file, search for user-agent (ignore case), and use various sort options

Simple alphabetic sort (may include duplicates)

```
$ strings /pcaps/fraudpack.pcap | grep -i user-agent | sort
```

Sort and unique lines. The two following sets of commands are equivalent:

```
$ strings /pcaps/fraudpack.pcap | grep -i user-agent | sort -u
$ strings /pcaps/fraudpack.pcap | grep -i user-agent | sort | uniq
```

Get a numeric count of each unique entry:

```
$ strings /pcaps/fraudpack.pcap | grep -i user-agent | sort | uniq -c
```

Get a numeric count of each unique entry, perform a numeric sort of that count:

```
$ strings /pcaps/fraudpack.pcap | grep -i user-agent | sort | uniq -c | sort -n
```

Sort and unique lines, print the length of each unique line followed by the line itself, perform a reverse numeric sort of that count:

```
$ strings /pcaps/fraudpack.pcap | grep -i user-agent | sort -u \
| awk '{print length, $0}' | sort -rn
```

Sort on the the second comma separated field

```
$ cat /bonus/alexa/top-1m.csv sort -t, -k2
```

---

wc

Determine number of lines in a file (the flag is the letter "ell", not the number one):

```
$ wc -l /etc/passwd
```

xxd

xxd creates a hexdump, or converts a hexdump into binary. A lot of malware hex-encodes web traffic or malicious payloads (such as DOS executables) in order to avoid signature matching. Useful hex patterns to look for are 4d5a90 (the magic bytes for a DOS executable: "MZ<90>"), and "DOS mode" (444f53206d6f6465, see commands below).

xxd cannot natively handle percent-encoded hex, such as "%63%67%69%2D%62%69%6E", but can if the percent signs are removed (see below).

Convert the string "DOS mode" to hex, grouped in sets of 4 hex characters (default):

```
$ echo -n "DOS mode" | xxd
0000000: 444f 5320 6d6f 6465                      DOS mode
```

Convert the string "DOS mode" to hex, ungrouped:

```
$ echo -n "DOS mode" | xxd -g0
0000000: 444f53206d6f6465                 DOS mode
```

Convert the hex string "444f53206d6f6465" to binary:

```
$ echo 444f53206d6f6465 | xxd -r -p
DOS mode
```

Use sed to remove the percent signs from the percent-encoded hex string "%63%67%69%2D%62%69%6E", then translate to binary:

```
echo "%63%67%69%2D%62%69%6E" | sed "s/\%//g" | xxd -r -p
cgi-bin
```

Additional Info