Cheat-Sheet

Before you solely rely on this cheat-sheet, **try to gather own information** about the commands with the use of the man-pages or the "help" command (like this: **"<command> --help"** or **"man <command>"**)

Commands

ssh

```
ssh <username>@<ip_address>
```

> `-p <port>` Specify a port
> `-i <keyfile>` Use a keyfile for authentication

Ssh (SSH client) is a program for logging into a remote machine and for executing commands on a remote machine.

exit

```
exit
```

Closes the current shell or ssh connection.

task

```
task
```

Prints the task of the current level.

hint

```
hint
```

Gives a hint for the current level.

man

```
man <command>
```

Provides an in-depth description of a command and its usage.

ls

```
ls <directory_path>
```

> `-a` Get all files
> `-l` List properties

List information about the FILEs (the current directory by default).

cat

```
cat <file1> <file2> ...
```

Prints the content of a file to the terminal output.

cd

```
cd <directory_path>
```

> `~` Go to home directory
>
> `..` Go one directory up

With this command you can change directories.

find

```
find <directory_path>
```

> `-name <file/directory name>` Search for name
>
> `-size <size>[c,b,n]` Search for size [c: Bytes, d: 512-Byte Block, n: default]
>
> `-user <username>` Search for owner username
>
> `-group <groupname>` Search for owner group name

Searches the provided directory (and child directories) for matching parameters (size, owner, name, etc.).

grep

```
grep <pattern>
```

Searches for a pattern in a file or from the terminal output.

uniq

```
uniq
```

Filters adjacent matching lines from files or the terminal input. (Filters out multiple occurrences)

sort

```
sort
```

Sort terminal output/files (alphabetically, numerically, etc.).

base64

```
base64 <file>
```

> `-d` Decode

Encode/decode strings/files with base64.

tr

```
tr <from> <to>
```

Translate a string from terminal input/files with a given string. (E.g. use a custom alphabet)

john

```
john --wordlist=<wordlist> <hashfile>
```

> `--show <hashfile>` Show result of cracked hash.

Crack hashes using a brute force-attack.

strings

```
strings <binary_file>
```

Search for printable strings inside a binary.

nc

```
nc <address> <port>
```

Network tool to establish connections on a lower layer.

tar

```
tar <flags> <file/folder/archive>
```

> `-xf` Unpack archive
> `-cf` Pack archive

Create archive from files/folders or unpack archives to receive files.

gzip

```
gzip <file/folder/archive>
```

> `-d` Decompress archive

Compress/uncompress files/folders.

curl

```
curl <URL>
```

> `-I` Print just the HEADER
> `-i` Include the HEADER

Network tool for transferring data from/to servers using URLs.

MISC

Pipe

```
<command_1> | <command_2> | ...
```

A pipe “|” allows to redirect output from command_1 to command_2, so command_2 can process the output.

E.g: `cat <file> | grep <pattern>`

Output Redirection

```
[1,2]><output_path>
```

The terminal has multiple output-streams. That means e.g. “normal output” (1) and “errors” (2) can be separated.

E.g: `find / -name <name> 2>/dev/null` redirects the error to /dev/null (trash)

Renaming files

```
mv <file_1> <file_2>
```

To rename files, it is common practice to just use the move-command (mv).

E.g: `mv <old_filename> <new_filename>`