

Bash-Fu

Cool things you can do on a Linux command line

File Querying/Manipulation

Unzip files into a directory named after the zip file

```
for file in *.zip; do [ -f "$file" ] || continue; unzip "$file" -d "${file%%.zip}"; done
for file in *.zip; do [ -f "$file" ] || continue; mkdir "${file%%.zip}" && bsdtar -xvf "$file" -C "${file%%.zip}"; done
for file in *.zip; do [ -f "$file" ] || continue; podman run --rm --workdir /data -it -v ./data docker.io/crazymax/7zip 7za x -aoa "$file" -o"${file%%.zip}"; done
```

Recursively count the number of files of any extension exist

```
find . -type f | sed -e 's/.*\\.//' | sed -e 's/.*\\///' | sort | uniq -c | sort -rn

# 14 zip
# 12 config
# 8 7z
```

Find which ports have been used by Podman/Docker containers

```
# Recursively search a directory for quadlet/compose files
find *.* -exec grep -oE '[0-9]{4,}:' '{}' \; | sort -h

# Search through running containers
podman ps --format "{{.Ports}}" | grep -oE ':[0-9]{4,}' | sort -h
```

Securely erase solid state storage

```
# Using NVME commands
nvme sanitize device -a start-crypto-erase
nvme sanitize device -a start-block-erase
nvme sanitize-log /dev/nvme0
```

```
blkdiscard --secure /dev/device
```

```
# Using in-place writing
```

```
shred --verbose --random-source=/dev/urandom -n1 --zero /dev/sdX
```

FFmpeg

Cut a clip

```
ffmpeg -ss 00:26:03 -to 00:27:05 -i <input file> -filter:v scale=720x480 -c:a copy <output file>
```

Recursively convert FLAC files into mp3

```
fdfind -t f -e flac -x ffmpeg -i "{}" -qscale:a 0 "{.}.mp3"
```

Networking

Interactively send a string over the network

```
# Note that this is unencrypted!
```

```
# Sender
```

```
read -rs i && echo $i | nc <dest_ip> <dest_port>
```

```
# Receiver
```

```
nc -l 5555 | wl-copy
```

Find the SHA256 fingerprint of a website's HTTPS certificate

```
echo 'Enter a URL: '; read -r url; echo 'Enter a port: '; read -r port && echo | openssl s_client -servername $url -connect $url:$port 2>/dev/null | openssl x509 -fingerprint -sha256 -noout
```

```
# This is particularly useful for verifying self-signed certificates in internal networks.
```

```
# After verifying a cert's authenticity, the browser should maintain trust unless the cert changes
```

Other

Create a WiFi QR code

```
qrencode -o wifi.png "WIFI:T:WPA;S:<SSID>;P:<PASSWORD>;;"
```

Create and open a two-factor LUKS device (password and keyfile)

```
# Format
echo 'Name of /dev/<device file>:'; read -r devicefile; echo 'Location of keyfile:'; read -r filepath; echo
'Password: '; read -rs passwd; fullpass="$(cat $filepath)$passwd"; echo $fullpass | sudo cryptsetup luksFormat
/dev/$devicefile -d -

# Unlock
echo 'Name of /dev/<device file>:'; read -r devicefile; echo 'Location of keyfile:'; read -r filepath; echo
'Password: '; read -rs passwd; fullpass="$(cat $filepath)$passwd"; echo $fullpass | sudo cryptsetup luksOpen
/dev/$devicefile cryptdisk -d -
```

Create a random password

```
# 24 digit alphaumeric
cat /dev/urandom | tr -dc '[:alnum:]' | head -c 24 | wl-copy

# see tr --help for more options
```

Revision #11

Created 23 July 2024 00:30:37 by GT

Updated 15 October 2024 23:59:40 by GT