Debian

- Apt-usage
- Create and restore apt-package-list
- Hardening
- Install Docker on Debian Trixie

Apt-usage

Install package

`apt-get install <packagename>`

search package

`apt-cache search <packagename> or <packagedescription>`

remove package

`apt-get remove <packagename>` to remove but letting configfiles untouched

`apt-get purge <packagename>` to remove with configfiles and without checking dependencies

`apt-get autoremove` remove unneccesary Packages

Update Package-Cache

`apt-get update`

Upgrade Packages

and (without installing additional) dependencies

`àpt-get upgrade`

can lead to breaking. To install needed dependencies do:

```
apt-get install -f
```

Upgrade whole OS

```
apt-get dist-upgrade
```

Create and restore apt-package-list

you can use this list for example to install all your apt packages on another machine, for example if you migrage or want to start over on a fresh installation

```
dpkg --get-selections | awk '$2 == "install" {print $1}' > packages.list.save
```

Restore your apt-packages

```
xargs -a "packages.list.save" sudo apt-get install
```

Don't forget to save your /etc/apt/sources.list and /etc/apt/sources.list.d/*, too!

Hardening

Apply a layered hardening methodology for Debian/Ubuntu systems. Includes firewalling, automatic updates, intrusion prevention, malware scanning, rootkit detection, and auditing.

Audit the current system configuration:

```
sudo lynis audit system
```

Configure a restrictive firewall with UFW:

```
sudo ufw default deny incoming && sudo ufw default allow outgoing && sudo ufw enable
```

Verify firewall status:

```
sudo ufw status verbose
```

Install and enable automatic security updates:

```
sudo apt install unattended-upgrades && sudo dpkg-reconfigure unattended-upgrades
```

Verify automatic updates are running:

```
systemctl status unattended-upgrades
```

Disable Avahi network discovery services:

```
sudo systemctl disable --now avahi-daemon
```

Verify Avahi is disabled:

```
systemctl is-enabled avahi-daemon
```

Disable the CUPS printing service:

```
sudo systemctl disable --now cups
```

Verify CUPS is disabled:

```
systemctl is-enabled cups
```

Install and enable Fail2Ban intrusion prevention:

```
sudo apt install fail2ban && sudo systemctl enable --now fail2ban
```

Copy default config file to ensure upgrades don't overwrite changes: 0

```
sudo cp /etc/fail2ban/jail.conf /etc/fail2ban/jail.local
```

Show Fail2Ban status:

```
sudo fail2ban-client status
```

Show SSH jail activity:

```
sudo fail2ban-client status sshd
```

View Banned IPs:

```
sudo fail2ban-client get sshd banip
```

Install ClamAV antivirus:

```
sudo apt install clamav clamav-daemon
```

Update ClamAV Virus Definitions:

```
sudo systemctl stop clamav-freshclam && sudo freshclam && sudo systemctl start clamav-freshclam
```

Scan the home directory for infected files:

```
clamscan -r --bell -i ~
```

Scan the entire system excluding /sys:

```
sudo clamscan -r / --exclude-dir="^/sys"
```

Install and update RKHunter:

```
sudo apt install rkhunter && sudo rkhunter --update
```

Run a non-interactive RKHunter scan:

```
sudo rkhunter --check --sk
```

Install and run Chkrootkit:

```
sudo apt install chkrootkit && sudo chkrootkit
```

Run a quieter Chkrootkit scan:

```
sudo chkrootkit -q
```

Install Needrestart to identify services using outdated libraries:

```
sudo apt install needrestart
```

Check which services require restarting:

```
sudo needrestart
```

Install Debsums for package integrity verification:

```
sudo apt install debsums
```

Show packages with modified files:

```
sudo debsums -c
```

Show only checksum mismatches:

```
sudo debsums -ca
```

Install and enable Linux auditing:

```
sudo apt install auditd audispd-plugins && sudo systemctl enable --now auditd
```

Verify auditd status:

```
sudo systemctl status auditd
```

Show audit events from today:

```
sudo ausearch -ts today
```

Generate an audit summary report:

```
sudo aureport --summary
```

Show login activity from audit logs:

```
sudo aureport --login
```

Install needrestart debsums apt-cacher apt-listchanges and apt-show-versions:

```
sudo apt install needrestart debsums apt-cacher apt-listchanges apt-show-versions
```

Credits:

https://github.com/DouglasFreshHabian/Cheatsh33ts/blob/main/TLDR/fortress.md

Install Docker on Debian Trixie

Install using the `apt` repository

Before you install Docker Engine for the first time on a new host machine, you need to set up the Docker `apt` repository. Afterward, you can install and update Docker from the repository.

Set up Docker's `apt` repository.

```
# Add Docker's official GPG key:
sudo apt update
sudo apt install ca-certificates curl
sudo install -m 0755 -d /etc/apt/keyrings
sudo curl -fsSL https://download.docker.com/linux/debian/gpg -o /etc/apt/keyrings/docker.asc
sudo chmod a+r /etc/apt/keyrings/docker.asc

# Add the repository to Apt sources:
sudo tee /etc/apt/sources.list.d/docker.sources <<EOF
Types: deb
URIs: https://download.docker.com/linux/debian
Suites: $(. /etc/os-release && echo "$VERSION_CODENAME")
Components: stable
Architectures: $(dpkg --print-architecture)
Signed-By: /etc/apt/keyrings/docker.asc
EOF

sudo apt update
```

Install the Docker packages.

To install the latest version, run:

```
sudo apt install docker-ce docker-ce-cli containerd.io docker-buildx-plugin docker-compose-plugin
```

Verify that the installation is successful by running the `hello-world` image:

```
sudo docker run hello-world
```

Full instructions here