Security & Firewall (Plus)

Linux Commands Course · Section 19

IDSchool

Linux Security Layers

Linux security operates on multiple levels:

1. **Discretionary Access Control (DAC):** standard file permissions and ownership.
2. **Capabilities:** fine-grained privileges for executables.
3. **Mandatory Access Control (MAC):** enforced security frameworks (SELinux, AppArmor).
4. **Network Firewall:** traffic filtering with ufw, firewalld, or nftables.

File Capabilities — getcap, setcap

Traditionally, privileged actions required root (UID 0).
Capabilities allow splitting root powers into smaller permissions.

List file capabilities:

```
sudo getcap /bin/ping
```

Output example:

```
/bin/ping = cap_net_raw+ep
```

This means ping can use raw network sockets without being setuid root.

Assign a capability:

```
sudo setcap cap_net_bind_service=+ep /usr/bin/nginx
```

Mandatory Access Control (MAC)

Beyond standard ownership and permissions, Linux can enforce additional security through **SELinux** or **AppArmor**.

SELinux (Security-Enhanced Linux)

Developed by the NSA, SELinux enforces strict policy rules for processes and files.

Check mode:

```
getenforce
```

Possible modes:

- Enforcing – policy actively blocks violations
- Permissive – logs violations but allows actions
- Disabled – inactive

Temporarily change mode (root only):

```
sudo setenforce 0   # switch to Permissive
sudo setenforce 1   # back to Enforcing
```


View logs:

AppArmor (Ubuntu and Debian)

AppArmor provides per-application confinement via security profiles.

Check AppArmor status:

```
sudo aa-status
```

Output example:

```
apparmor module is loaded.
26 profiles are loaded.
22 profiles are in enforce mode.
```

List profiles and modes:

```
sudo aa-status | grep enforce
```

Host Firewalls — Overview

Linux firewalls filter traffic using the **netfilter** framework.
There are several user-friendly frontends built on top of it.

UFW (Uncomplicated Firewall)

Simplified interface (Ubuntu and derivatives).

Check status:

```
sudo ufw status
```

Enable the firewall:

```
sudo ufw enable
```

Allow or deny rules:

```
sudo ufw allow 22/tcp
sudo ufw deny 23/tcp
```

Delete a rule:

firewalld and firewall-cmd

Used by RHEL, Fedora, and openSUSE.

Start and enable service:

```
sudo systemctl enable --now firewalld
```

Check active zones:

```
sudo firewall-cmd --get-active-zones
```

Allow a service in the default zone:

```
sudo firewall-cmd --add-service=http --permanent
sudo firewall-cmd --reload
```

Add a custom port:

nftables and iptables (Conceptual Overview)

nftables is the **modern packet filter** replacing iptables.

- iptables — legacy interface (still widely used)
- nftables — unified replacement for IPv4/IPv6

Check active rules:

```
sudo nft list ruleset
```

Example nftables rule snippet:

```
table inet filter {
  chain input {
    type filter hook input priority 0;
    policy drop;
    iif "lo" accept
    ct state established,related accept
    tcp dport {22,80,443} accept
  }
```

When to Use Which

Tool	Recommended for	Notes
ufw	Simple desktop/server setups	Easy syntax
firewalld	Enterprise systems (RHEL/Fedora)	Zone-based rules
nftables	Advanced configurations	Modern standard
iptables	Legacy compatibility	Being replaced

Recap

- **File capabilities:** getcap, setcap (fine-grained privileges)
- **MAC systems:** SELinux (getenforce, setenforce), AppArmor (aa-status)
- **Firewalls:** ufw, firewall-cmd, nftables, iptables
- Defense layers work together — never rely on just one.
