Security & Firewall (Plus)

Linux Commands Course · Section 19

IDSchool

Goal

Learn about **security controls and firewalls** in Linux — how privileges are managed, how access control systems work, and how to configure basic network protection.

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
```

Remove it:

```
sudo setcap -r /usr/bin/nginx
```

View all files with capabilities:

```
sudo getcap -r / 2>/dev/null
```

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:

```
sudo cat /var/log/audit/audit.log | grep denied
```

Permanent configuration is in /etc/selinux/config.

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
```

Enable or disable specific profiles:

```
sudo aa-enforce /etc/apparmor.d/usr.bin.firefox
sudo aa-disable /etc/apparmor.d/usr.bin.firefox
```

AppArmor is easier to manage than SELinux but provides similar isolation.

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:

```
sudo ufw delete allow 22/tcp
```

View detailed numbered list:

```
sudo ufw status numbered
```

Disable firewall:

```
sudo ufw disable
```

Reset to default:

```
sudo ufw reset
```

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:

```
sudo firewall-cmd --add-port=8080/tcp --permanent
sudo firewall-cmd --reload
```

List all active rules:

```
sudo firewall-cmd --list-all
```

---

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
  }
}
```

iptables equivalent (legacy):

```
sudo iptables -L -n -v
```

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

Best Practices for Security

- Keep system and packages updated (`apt upgrade`, `dnf update`).
- Limit SSH access; use key authentication instead of passwords.
- Disable unnecessary services.
- Use `ufw` or `firewalld` to restrict inbound ports.
- Monitor logs for unusual activity (`journalctl -p err`, `/var/log/auth.log`).
- Review file capabilities periodically.
- For servers, use `fail2ban` to block brute-force attacks.

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.

Practice

1. View capabilities of `/bin/ping`.
2. Enable `ufw` and allow SSH while denying Telnet.
3. Check current firewall rules.
4. Check whether your system uses SELinux or AppArmor.
5. List loaded AppArmor profiles or SELinux mode.
6. View firewall rules using `nft list ruleset`.

Next Up

Backups & Data Moves (Plus) — archiving, syncing, and transferring efficiently.