Docker Lab 10 — Docker Runtime Security with Falco

Author: Dr. Sandeep Kumar Sharma

Lab Description

Lab 10 introduces **runtime container security** using **Falco**, a CNCF project used to detect unexpected container behavior, anomalies, and security threats. You will configure Falco, generate suspicious activity, and observe real-time alerts.

Topics Covered

- What is Falco?
- Why runtime security is required
- Installing Falco
- Detecting suspicious container behavior
- Custom Falco rules

Learning Objectives

- Understand how Falco monitors syscalls
- Detect real-time attacks or anomalies inside containers
- Write custom Falco rules for your environment

Learning Outcomes

- Ability to monitor Docker runtime security
- Ability to configure alerts and custom security detection logic

---

Section 1 — Install Falco

```
curl -s https://falco.org/install.sh | sudo bash
```

Verify installation:

```
sudo systemctl status falco
```

Section 2 — Run a Test Container

```
docker run -it --name sandeep-falco-test ubuntu bash
```

Inside container:

```
apt update && apt install -y curl
```

This will trigger Falco rules.

Section 3 — View Falco Logs

```
sudo falco -o time_format_iso_8601=true
```

Look for alerts like:

```
Notice: curl executed inside container (sandeep-falco-test)
```

Section 4 — Create a Custom Rule

Edit Falco rules:

```
sudo nano /etc/falco/falco_rules.local.yaml
```

Add:

```
- rule: Detect File Changes in /etc
  desc: Alert when files inside /etc are modified
  condition: evt.type=open and fd.name startswith /etc and container.id != host
  output: "ETC modification detected in container (user=%user.name)"
  priority: WARNING
  tags: [filesystem]
```

Restart Falco:

```
sudo systemctl restart falco
```

Section 5 — Trigger Custom Rule

Inside container:

```
echo "test" > /etc/testfile
```

Falco logs should show the alert.

Section 6 — Cleanup

```
sudo systemctl stop falco
```

```
docker rm -f sandeep-falco-test
```

Summary

You successfully installed Falco, observed runtime security alerts, and wrote custom detection rules to secure Docker workloads.