🔥 Tutorial 9: Network Security - Firewalls

In this tutorial, we use the open-source, container-based network emulator Kathara to build a network of **two LANs and one WAN connection**. Our firewall is a **Linux router running `iptables`** as the firewall software. `iptables` doesn't have the full functionalities of a commercial firewall or an open source firewall, such as `pfSense`. However, it is sufficient to learn basic concepts of rule ordering and shadowing.

1. Prerequisites

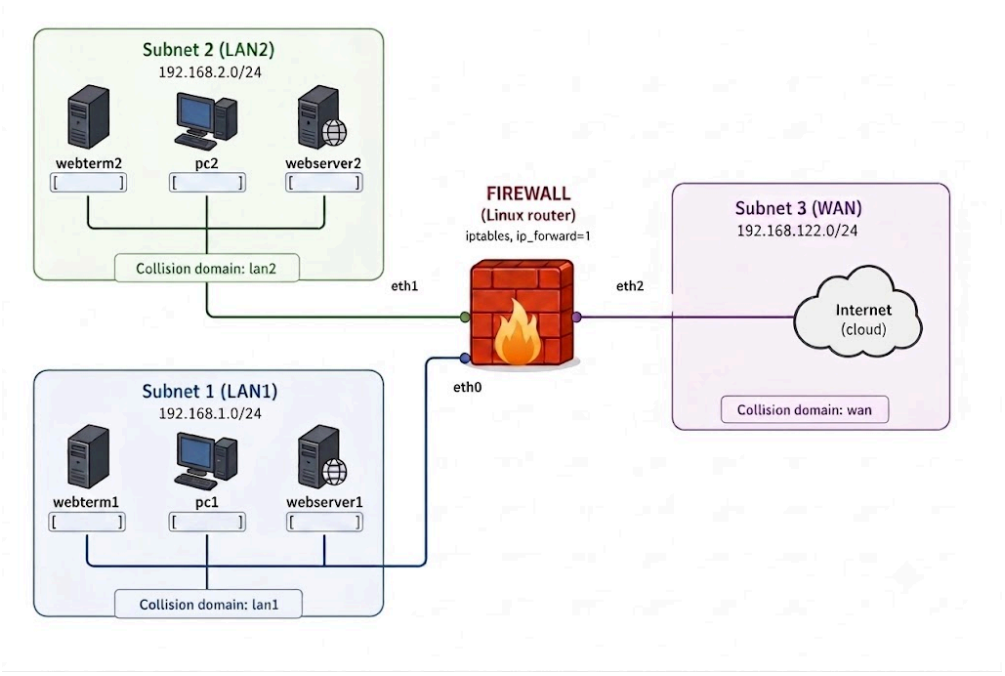
- Everything you need is pre-installed in the VM. Get the lab folder (`/home/ubuntu/Tutorial_9/kathara-lab/`) onto your machine. All commands from this tutorial must be run **inside that folder**.
- It is better to do this tutorial through an SSH session and in the Jupyter Notebook.

Check whether Kathara is running OK.

```
kathara check
```

2. Know the lab topology

The lab topology consists of two private networks and one public network within a firewall sitting between them. In contrast to the given figure, note that there is **no explicit switch device** in our simulated network. That is because in Kathara, a *collision domain* (here `lan1` , `lan2` , `wan`) is itself a shared Ethernet segment — that in fact is the switch.



Start the whole scenario:

`kathara lstart --noterminals`
This starts every device. You can use `kathara list` to list all the devices in the network. You can use `kathara exec` / `kathara connect` to execute commands or connect to various devices in the network.

```
kathara connect pc1                    # interactive shell on a pc1
```
Question (a): Use `ifconfig` and find out the `ip addresses` of all the devices. You can do either in the individual terminals of the devices or using `kathara connect <divice name>`, followed by `ifconfig`. In the firewall use the `ip -brief address` command.

If something goes wrong and you want to restart, use `kathara lclean` and restart issuing the above commands.

3. More on the topology

Question (b) — Why do the collision domains (lan1, lan2) / the "LAN switches" have no IP address?

Question (c) — What is the firewall's additional function, and what turns it on?**

4. Read the firewall rules

`kathara exec firewall -- iptables -L FORWARD -v -n --line-numbers`
Things to understand while reading it:

- The **`FORWARD` chain** holds the rules for traffic passing *through* the firewall between subnets. Rules are evaluated **top to bottom**; the first match wins.
- A rule with `-i eth0` matches traffic **entering from LAN1**; `-i eth1` matches traffic **entering from LAN2**. This is the equivalent of the pfSense "LAN1 interface" and "LAN2 interface" rule tabs.
- The first rule (`ctstate RELATED,ESTABLISHED`) makes the firewall **stateful**: replies to a connection the firewall already allowed out are let back automatically — exactly like pfSense.

Focus on what happens to **Ping (ICMP)** traffic, then **predict** the outcome of each of these before you test them:

- Ping `pc2` from `pc1`
- Ping `pc1` from `pc2`
- Ping `webserver1` from `pc1`
- Ping `pc2` from `webserver1`
- Ping `webserver1` from `pc2`

5. Verify your predictions

Open a shell on each device (`kathara connect pc1`) and ping, or run one-liners:

```
kathara exec pc1        -- ping -c3 192.168.2.2     # pc1  -> pc2
kathara exec pc2        -- ping -c3 192.168.1.3     # pc2  -> pc1
kathara exec pc1        -- ping -c3 192.168.1.4     # pc1  ->
webserver1
kathara exec webserver1 -- ping -c3 192.168.2.2     # webserver1 ->
pc2
kathara exec pc2        -- ping -c3 192.168.1.4     # pc2  ->
webserver1
```

Did they behave as you expected? Look again at the LAN2 rules and explain **why** the outcomes are asymmetric. Which LAN2 rule never gets used, and why?

6. Fix the rule shadowing

The system administrator's real objective was:

> **Allow ping (ICMP) between the two subnets — devices in LAN1 can ping devices in LAN2 and vice-versa — and nothing else.**

Edit the firewall rules to achieve this. You can experiment live on the firewall:

```
kathara connect firewall
# then use: iptables -L FORWARD --line-numbers
#           iptables -D FORWARD <num>          (delete a rule)
#           iptables -I FORWARD <num> ...      (insert at a
position)
#
```

7. Clean up

```
kathara lclean
```
