Universität Zürich[UZH]

0690 - Communication Systems Lab Course

Exercise 3

1) Introduction
2) Preparation
3) NAT
4) Firewall

October 4, 2022

Assistants: Christian Killer, Eder John Scheid,
Jan von der Assen, Dr. Alberto Huertas

University of Zurich
Department of Informatics (IFI)
Binzmühlestrasse 14, CH—8050 Zürich, Switzerland

1 Introduction

In this lab, you will learn how to secure a machine using a Linux-based firewall. The firewall will be configured using iptables, which allows filtering of IP packets according to specified criteria, such as the source IP address or the destination port. Moreover, iptables will be used to enable Internet access to computers without routable IP addresses, using a NAT (Network Address Translation) approach. Therefore, this architecture is similar to residential internet service, where one public IP address is shared by many devices in a LAN.

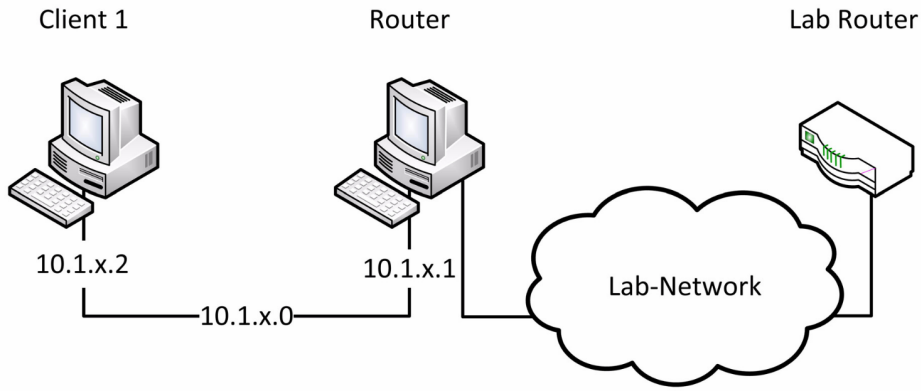

Figure 1: Network Topology

Group	Client 1	Router/ Firewall	Subnet
Group 1	10.1.1.2	10.1.1.1	10.1.1.0/24
Group 2	10.1.2.2	10.1.2.1	10.1.2.0/24
Group 3	10.1.3.2	10.1.3.1	10.1.3.0/24
Group 4	10.1.4.2	10.1.4.1	10.1.4.0/24
Group 5	10.1.5.2	10.1.5.1	10.1.5.0/24
Group 6	10.1.6.2	10.1.6.1	10.1.6.0/24
Group 7	10.1.7.2	10.1.7.1	10.1.7.0/24
Group 8	10.1.8.2	10.1.8.1	10.1.8.0/24
Group 9	10.1.9.2	10.1.9.1	10.1.9.0/24

Figure 2: Private Subnets

1.1 Preparation

1. Before starting to use `iptables`, you will have to (a) install the software required to successfully solve the exercises and (b) study the `iptables man` page. Use the `apt install` command to install the following tools on your machines. Note, that you will need to establish an uplink directly to the lab network first (e.g., connecting the PC to the wall and running `sudo dhclient enpXs0`).

   (a) **Install on PC1 (Router): apache2**, **nmap**

   (b) **Install on PC2 (Client): nmap**

2. Once you are done with that, you can set up the topology depicted in Figure 1 with the Addressing from Figure 2. Be careful to use the correct PC for the router and client. Since only the router needs the direct uplink, remove the cable from client to the wall and release its DHCP lease (e.g., `sudo dhclient -r enpXs0`) Now, take some time and study the iptables map (cf Figure 3):

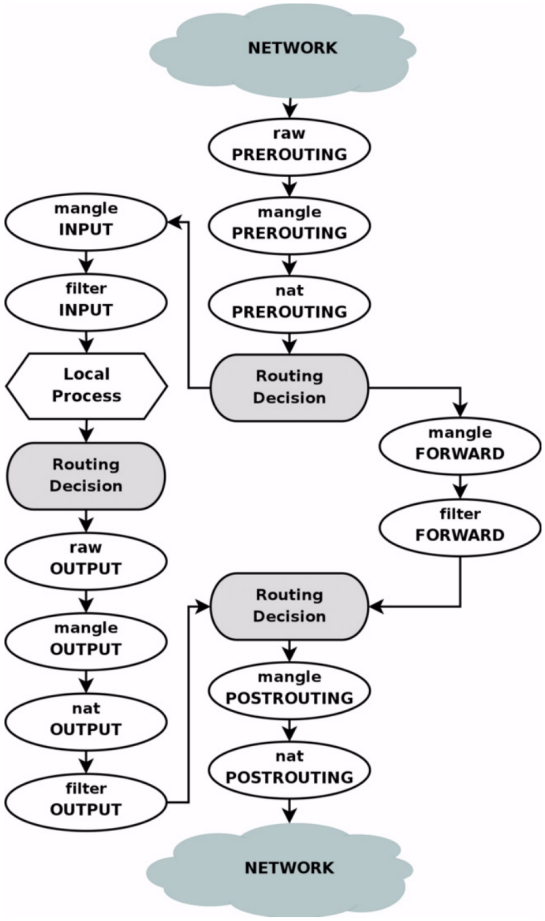

Figure 3: iptables NAT

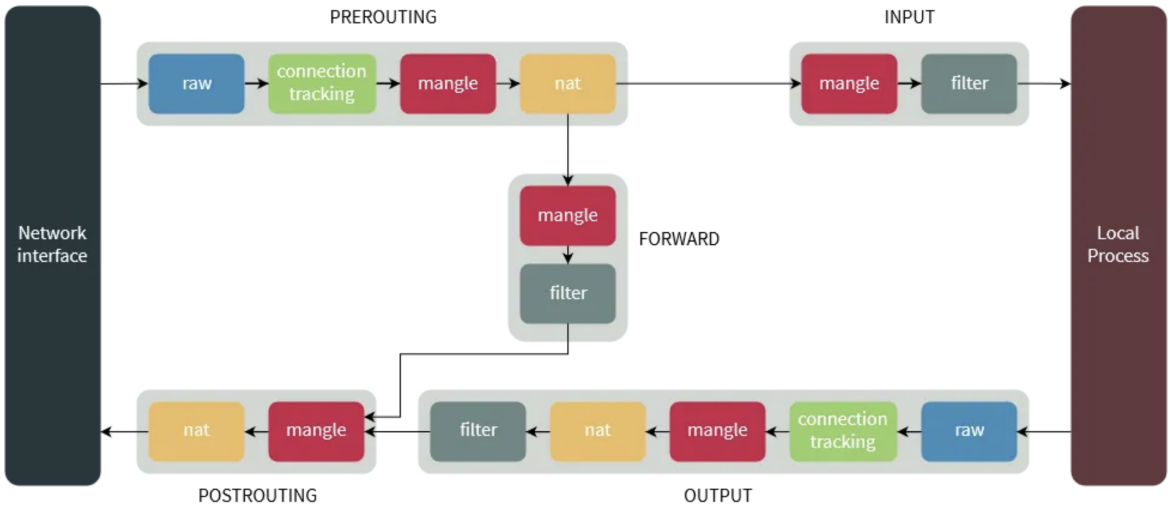

Figure 4: iptables chains and tables

The firewall configuration (which includes NAT) must be stored in a shell-script. Thus, create the file *firewall.sh* and save it in the home directory. Write `#!/bin/bash` in the first line of this script file. Also, make the file executable (`chmod +x firewall.sh`). Make a second script, or include rules in the beginning of the first script, which deletes all iptables entries from all tables and chains.

2 NAT

The following steps will guide you through the exercise. Collect the answers to the questions posed during the instructions in a text file and send them in an e-mail to vonderassen@ifi.uzh.ch and killer@ifi.uzh.ch at the end of the class. Keep all the iptables commands in the firewall script, and use comments (line starting with #) to indicate the instruction number for the commands, e.g.:

```
# 5 Block access from client
iptables -A FORWARD ......
```

Incoming packets first traverse the PREROUTING chain. Then, depending on the destination IP, the INPUT or FORWARD chains are executed. The OUTPUT chain is traversed only by packets which originate from the local machine.

1. As you already are an expert on LAN networks, configure the Router and the Client with IP addresses as described in Figure 2. Then, you should be able to exchange ping messages between them. Disconnect the client machine from the Internet (release DHCP if not done so already), such that the client can only access (the Internet via) the router.

2. In the router, activate the forwarding of IPv4 packets by executing the following line `sudo sh -c "echo 1 > /proc/sys/net/ipv4/ip_forward"`. Include this line in your firewall.sh script, and do not forget to include a comment describing what

each line of your script does. Is it possible to ping the lab main router or an outside external IP address from the client? Explain the result (monitoring the outgoing interface on the client and then on the router might help you find the answer).

3. Configure the **NAT** functionality in the router with iptables. This can be accomplished by adding a rule in the **POSTROUTING** chain, which performs the **MASQUERADE** action (like in the previous exercise). Include the rule into the firewall.sh script. Can you ping the lab main router or an outside address from the client? Explain the result.

4. Start to ping an outside host (e.g., 8.8.8.8) from the client. Start Wireshark on the router and monitor the outgoing interface. Which is the source IP address of the packets that the router is forwarding through the uplink?

3 Firewall

1. A firewall can be tested with the *nmap* port scanner. First, delete all existing entries (i.e., ”flush”) from all iptables tables. Scan the router from the client and write down which ports are opened. Which services might be running on this port?

2. For each packet, the firewall will evaluate each iptables rule from top to bottom, and the first matching rule will be applied. If there is not a matching rule, a default policy will be applied. By default, this policy is **ACCEPT**, what means that all packets are allowed, otherwise the packets are dropped. On the router, set DROP, as a default policy, for all three chains of the filter table (**INPUT**, **FORWARD**, **OUTPUT**). Also, the default policy commands (one for each chain) should be the first commands in your firewall.sh script. Try to ping an outside host (e.g. 8.8.8.8) from the client and the router.

3. Configure the firewall to, at first, only allow local (i.e., on the lab subnets) processes to communicate with each other. Thus, all packets which originate from the lab subnets (cf. Figure 2) should be allowed by the firewall. With that, the two hosts should be able to ping each other through the 10.1.X.0/24 network.

4. The **OUTPUT** chain controls which packets originate from the host and are allowed to exit the host. Add rules to allow all such packets, since it is assumed that only a trusted system administrator can access the firewall. Allow outgoing connections so that browsing websites is possible on the router but not on the client. Do not change the default policy (**DROP**). Do not forget to allow DNS queries, otherwise the client will not be able to resolve domains.

5. Allow the client to initiate ping (Hint: allow ICMP), HTTP, and HTTPS outgoing connections. All other type of connections must be dropped.

Do not forget to send your firewall script(s) and answers to the questions to vonderassen@ifi.uzh.ch and killer@ifi.uzh.ch.