COMP 8006

Assignment 2

Tao Yuan & Xinghua Wei

Design Work

Objectives

For This assignment, the objective is to design, implement and test a standalone Linux firewall and packet filter. The constrains are:

- Set the initial default policies
- Get user specified parameters (see constraints) and create a set of rules that will implement the firewall requirements. Specifically, the firewall will control:
    - Inbound/Outbound TCP packets on allowed ports.
    - Inbound/Outbound UDP packets on allowed ports.
    - Inbound/Outbound ICMP packets based on type numbers.
    - All packets that fall through to the default rule will be dropped.
    - Drop all packets destined for the firewall host from the outside.
    - Do not accept any packets with a source address from the outside matching your internal network.
    - You must ensure the you reject those connections that are coming the “wrong” way (i.e., inbound SYN packets to high ports).
    - Accept fragments
    - Accept all TCP packets that belong to an existing connection (on allowed ports).
    - Drop all TCP packets with the SYN and FIN bit set.
    - Do not allow Telnet packets at all.
    - Block all external traffic directed to ports 32768 – 32775, 137 – 139, TCP ports 111 and 515.
    - For FTP and SSH services, set control connections to "Minimum Delay" and FTP data to "Maximum Throughput"
- Design a test procedure that will test all your firewall rules and print the results of the test to a file. Make sure that someone reading the file contents will know exactly which rule worked and which rule failed.
- The machines in the lab are equipped with two Ethernet cards. One of them is already configured and operational. You will have to enable and configure the other one for use as the gateway to your “internal” network.

- The firewall/packet filter must be designed and implemented using Netfilter.
- Your firewall script must have two sections: a "User Configurable Section" and the "Implementation Section".
- The user configuration section will allow a user to set at least the following parameters:
- Only allow NEW and ESTABLISHED traffic to go through the firewall. In other words you are doing stateful filtering.
- You must ensure that you reject those connections that are coming the "wrong" way, meaning inbound connection requests (unless of course it is to a permitted service).

Setting up network environment

For this assignment, we need three computers. One for external network, one for being firewall, and one for the client workstation. Also, we need to use a cross cable to connect firewall and workstation together so that external network will only be able to see the firewall instead of workstation.

To do that, after connecting two machines using psychical cable, we need to configure system setup

For firewall machine:

- Input 1 to /proc/sys/net/ipv4/ip_forward
- Ifconfig local net card local gateway ip up
- Ip route add net dev net card
- Route add default gw net gateway ip
- Route add -net local net gw local gateway ip

For client workstation:

- Make sure nameserver in /etc/resolv.conf is the same as the one in firewall machine
- Close eno1
- Close enp2s0
- Bring enp2s0 up and sign a new ip for this machine
- Route add default gw local gateway ip

Then the new wired connection should be active, but workstation would not have internet access. Therefore, we need to route internet to client machine by doing POSTROUTING and PREROUTING:

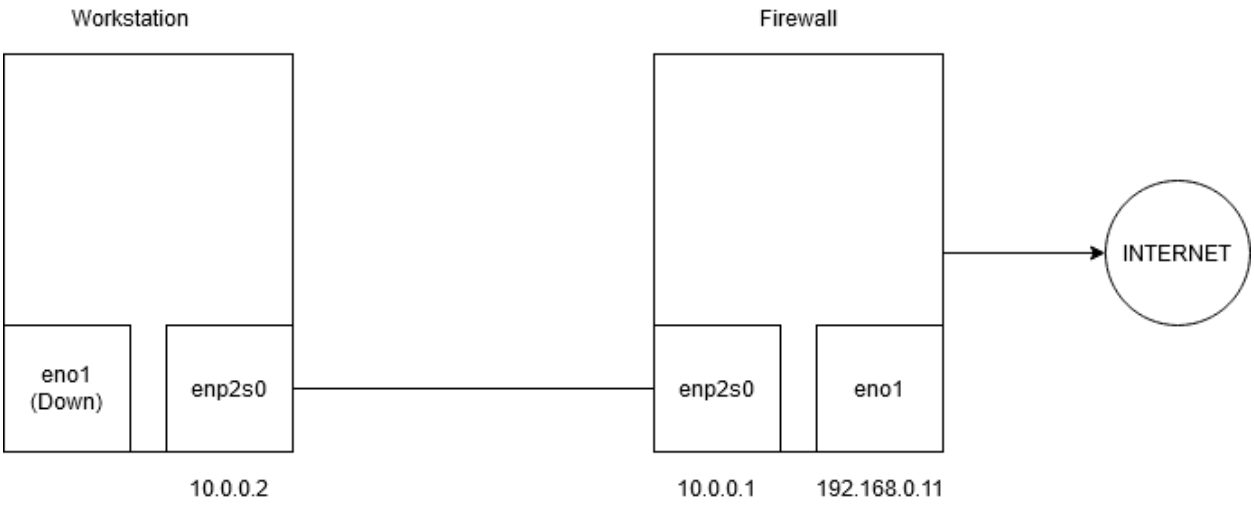
- POSTROUTING
    - POSTROUTING using iptables with SNAT to source ip of internet ip
- PREROUTING
    - PREROUTING using iptables with DNAT to destination of client ip

After all configuration set, we can start build firewalls.

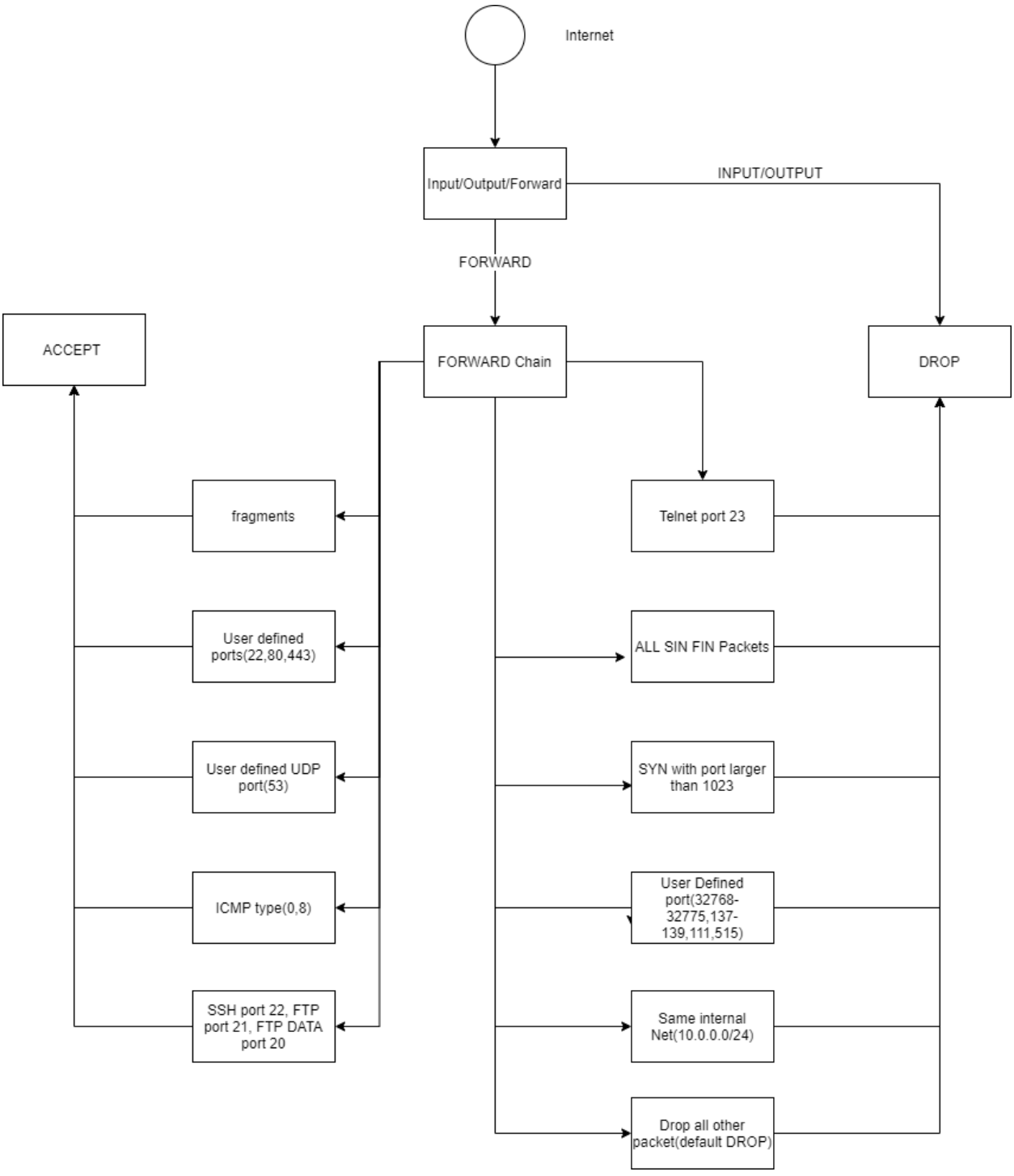
Firewall

Because for this assignment, firewall does not need internet access and only forward all the packets to client workstation, therefore, there is no need for firewall to have input chain or output chain to accept anything. We only need forward chain.

Overall principle:



- Shut eno1 of workstation down
- Set up enp2s0 with new ip
- Set ip new enp2s0 ip in firewall side
- Eno1 in firewall side have internet access
- Firewall will filter and forward packets through enp2s0 to workstation



Firewall Pseudocode

First, we put all the DROP rules before ACCEPT rules

- Drop all telnet
    - Drop TCP with destination port 23

- DROP all SIN FIN packets
    - Flag with SIN and FIN set together will be dropped
- Drop incoming SYN packets to high ports
    - Drop flags with SYN when source port is larger than 1023
- DROP user defined TCP ports
    - Drop TCP packets coming to network with port that user defined
- DROP UDP ports
    - Drop UDP packets coming to network with port that user defined
- Drop packets to firewall from outside when having same internal network
    - DROP packets to network card from source ip of local netmask
- DROP all packets destined for the firewall from the outside
    - Set default INPUT, OUTPUT, FORWARD rule to DROP

Then we set our ACCEPT rules,

- ACCEPT fragments
    - ACCEPT iptables with -f
- ACCEPT TCP on user defined port
    - Only NEW or ESTABLISHED can access in this user defined TCP ports
- ACCEPT UDP on user defined port
    - Only NEW or ESTABLISHED can access in this user defined UDP ports
- Allow user defined ICMP type
    - Only NEW or ESTABLISHED can access in this user defined ICMP ports
- ACCEPT SSH, FTP, and FTP DATA
    - SSH, PREROUTING on port 22, set to minimize-delay
    - FTP, PREROUTING on port 21, set to minimize-delay
    - FTP DATA, PREROUTING on port 20, set to maximize-throughput

Test

For Testing, we will write each rule a test case and write a test script and output result to a file.

- Using Hping3 to test user defined inbound TCP port
- Using Hping3 to test user defined outbound TCP port
- Using Hping3 to test user defined inbound UDP port
- Using Hping3 to test user defined outbound UDP port

- Using Hping3 to test user defined ICMP type
- Using Hping3 to test user defined outbound ICMP type
- Using Hping3 to test default packet
- Using Hping3 to test external network matching internal network
- Using Hping3 to test accepting fragments
- Using Hping3 to test SIN FIN
- Using Hping3 to test SIN with high ports
- SSH into another machine
- Using Ncat to test FTP
- Using Hping3 to test Telnet(port 20)