***MN502 Network Security***

**Laboratory 5**: l Packet Filtering Firewalls (IPTABLES)

**Description**:

This manual is for running experiments on IPTABLES firewall. In this lab you will set up a Linux-based system as a firewall, according to a network security policy provided to you. The implementation will be done using an Oracle Virtual Box.

* In the first experiment, you will use IPTABLES firewall on a Kali in order to be familiar with IPTABLES firewalls. Students will also explore the functionalities of iptables, the Linux firewall
* In the second experiment, students will perform some basic commands, followed by some exercises.

**Pre-work:**

1. **Launch Virtual Box**
2. **Launch your Kali**
3. **Do all your exercises in the VM**

**Exercises 1:** You are required to read, install, set up, configure, and test your iptables firewall

The goal of this exercise is to explore the functionalities of iptables, the Linux firewall, in a simple network setting. To facilitate the deployment of the network setting out of the laboratory, we will use only Kali virtual machines for implementing it.

IPTABLES is a command line utility for configuring Linux kernel [firewall](https://wiki.archlinux.org/index.php/Firewall) implemented within the [Netfilter](https://en.wikipedia.org/wiki/Netfilter" \o "wikipedia:Netfilter) project. The term iptables is also commonly used to refer to this kernel-level firewall. Iptables is a Linux command line firewall that allows system administrators to manage incoming and outgoing traffic via a set of configurable table rules.

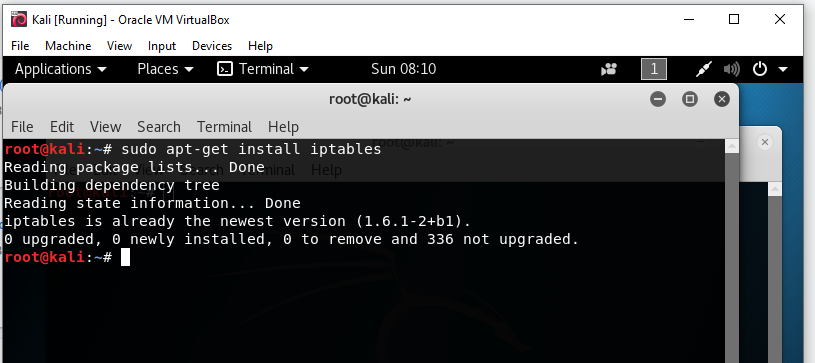
Iptables uses a set of tables which have chains that contain set of built-in or user defined rules. Thanks to them a system administrator can properly filter the network traffic of his system.

Per iptables manual, there are currently 3 types of tables:

* 1. FILTER – this is the default table, which contains the built in chains for:
     1. INPUT  – packages destined for local sockets
     2. FORWARD – packets routed through the system
     3. OUTPUT – packets generated locally

1. Install iptables (if not installed already).

Sudo apt-get install iptables



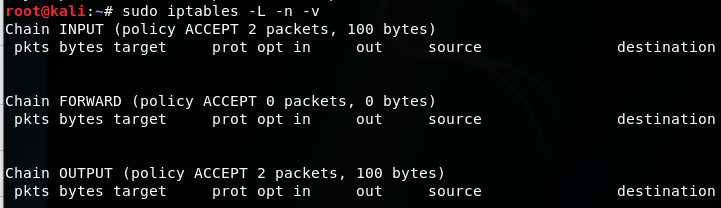
1. Access to the iptables manual page

Man iptables



1. If you want to check your existing rules, use the following command

Sudo iptables –L –n –v



Text

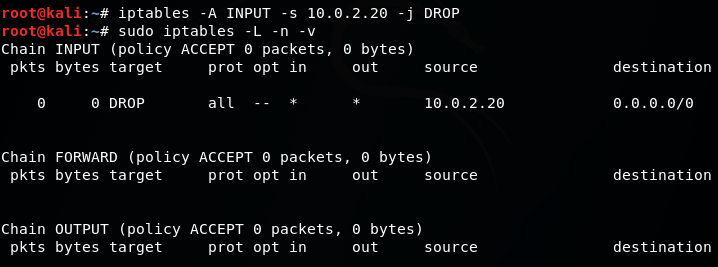
Description automatically generated

**Exercises 2:** Explore the functionalities of iptables, the Linux firewall and write some rules

1. Block Specific IP Address in IPtables Firewall



For example

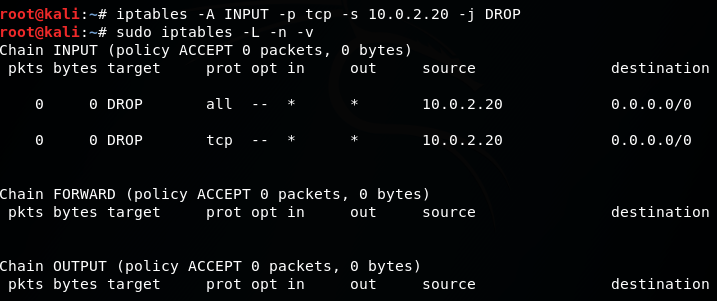


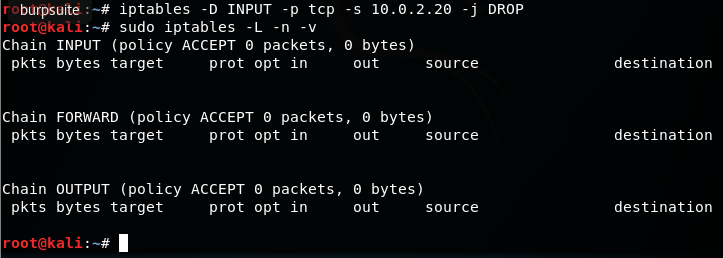
Where you need to change "xxx.xxx.xxx.xxx" with the actual IP address. Be very careful when running this command as you can accidentally block your own IP address. The -A option appends the rule in the end of the selected chain.

In case you only want to block TCP traffic from that IP address, you can use the -p option that specifies the protocol. That way the command will look like this:



For example

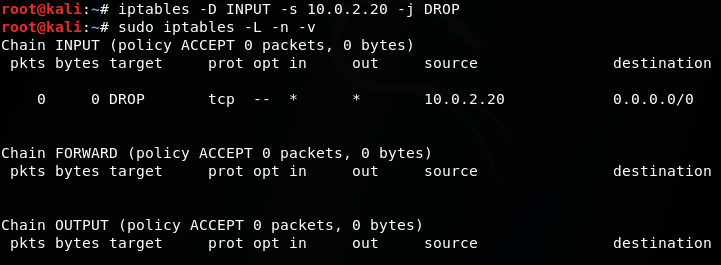




1. Unblock IP Address in IPtables Firewall

If you have decided that you no longer want to block requests from specific IP address, you can delete the blocking rule with the following command:





Need perms?

Text

Description automatically generated

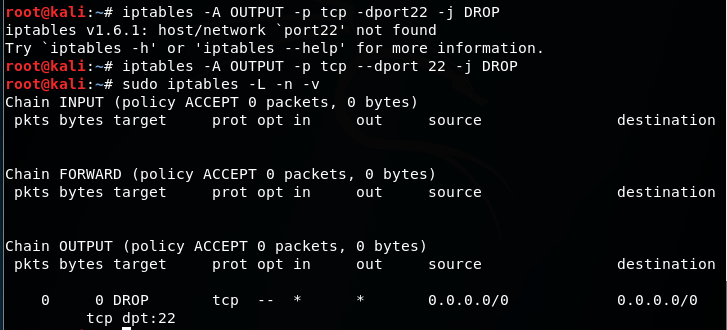
The **-D** option deletes one or more rules from the selected chain. If you prefer to use the longer option you can use **--delete**.

1. Block Specific Port on IPtables Firewall

Sometimes you may want to block incoming or outgoing connections on a specific port. It’s a good security measure and you should really think on that matter when setting up your firewall.

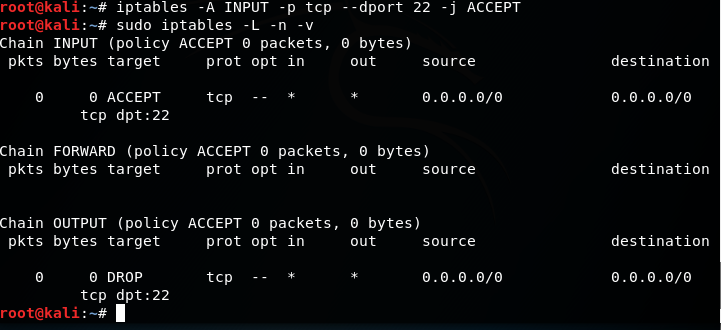
To block outgoing connections on a specific port use





To allow incoming connections use:





In both examples change **"xxx"** with the actual port you wish to allow. If you want to block UDP traffic instead of TCP, simply change **"tcp"** with **"udp"** in the above iptables rule.

1. Allow Multiple Ports on IPtables using Multiport

You can allow multiple ports at once, by using multiport, below you can find such rule for both incoming and outgoing connections:



1. Block Facebook on IPtables Firewall

First find the IP addresses used by Facebook



You can then block that Facebook network with:



Keep in mind that the IP address range used by Facebook may vary in your country.

1. Block Network Flood on Apache Port with IPtables



The above command limits the incoming connections from per minute to 100 and sets a limit burst to 200. You can edit the limit and limit-burst to your own specific requirements

1. Block Incoming Ping Requests on IPtables

Some system administrators like to block incoming ping requests due to security concerns. While the threat is not that big, it’s good to know how to block such request:



1. Allow loopback Access

Loopback access (access from 127.0.0.1) is important and you should always leave it active:



1. Keep a Log of Dropped Network Packets on IPtables

If you want to log the dropped packets on network interface eth0, you can use the following command:



1. Block Access to Specific MAC Address on IPtables

You can block access to your system from specific MAC address by using:



Of course, you will need to change "00:00:00:00:00:00" with the actual MAC address that you want to block

1. Limit the Number of Concurrent Connections per IP Address

If you don’t want to have too many concurrent connection established from single IP address on given port you can use the command below:



The above command allows no more than 3 connections per client. Of course, you can change the port number to match different service. Also the --connlimit-above should be changed to match your requirement.

1. Search within IPtables Rule

Once you have defined your iptables rules, you will want to search from time to time and may need to alter them. An easy way to search within your rules is to use:



In the above example, you will need to change $table with the actual table within which you wish to search and $string with the actual string for which you are looking for.

Here is an example:



1. Flush IPtables Firewall Chains or Rules

If you want to flush your firewall chains, you can use



You can flush chains from specific table with



You can change "nat" with the actual table which chains you wish to flush.

1. Disable Outgoing Mails through IPTables

### If your system should not be sending any emails, you can block outgoing ports on SMTP ports. For example you can use this

### 

**Description of the laboratory exercise**:

In this laboratory will explore the Linux firewall iptables. The first part of the lab will describe iptables, then basic commands, followed by some exercises.

**Task 1**: **Running iptables**

After completing all the steps in the manual for exercise one answer the following questions:

1. Write a report on your understanding on how iptables works.

**An administrator tool for filtering ports and IP’s over a network.**

1. Discuss how the computer system could be secured using iptables firewall.

**Since the admin person using iptables can put in certain rules and policies by blocking and letting in only certain pots and IP addresses.**

**Task 2**: Explore the functionalities of iptables, the Linux firewall and write some rules

After completing all the steps in the manual for exercise two answer the following questions:

1. Reject all **ssh** packets.

# Set default chain policies

iptables -P INPUT DROP

iptables -P FORWARD DROP

iptables -P OUTPUT ACCEPT

# Accept on localhost

iptables -A INPUT -i lo -j ACCEPT

iptables -A OUTPUT -o lo -j ACCEPT

# Allow established sessions to receive traffic

iptables -A INPUT -m conntrack --ctstate ESTABLISHED,RELATED -j ACCEPT

1. Allow **ssh** remote connections.

iptables -A OUTPUT -p tcp --sport 22 -j ACCEPT

1. Deny **ping**.

iptables -D INPUT -p icmp --icmp-type echo-request -j REJECT

1. Reject all traffic coming to port 80.

iptables -A INPUT -p tcp -i eth1 ! -s x.x.x.x. --dport 80 -j DROP

1. Block incoming traffic connection to your IP address of your virtual machine.

iptables -I INPUT -p tcp --dport 22 -m mac --mac-source 1E:2B:65:48:54:AD -j REJECT

1. Allow traffic coming to port 80 (inbound) but reject traffic going out (outbound) through port 80.
2. What is the command to block the following subnet: 192.168.2.x?

$ sudo iptables -A INPUT -s 192.168.2/24 -j DROP

1. Describe what the following rules do:
   * + 1. iptables -A INPUT -i lo -p all -j ACCEPT
       2. iptables -A INPUT -p all -s localhost -i eth0 -j DROP
       3. iptables -A INPUT -s 0/0 -i eth0 -d 192.168.1.1 -p TCP -j ACCEPT
2. A SYN flood is a form of denial-of-service attack in which an attacker sends a succession of SYN requests to a target's system. This is a well-known type of attack and is generally not effective against modern networks. It works if a server allocates resources after receiving a SYN, but before it has received the ACK. If Half-open connections bind resources on the server, it may be possible to take up all these resources by flooding the server with SYN messages. Syn flood is common attack and it can be blocked with iptable rules.

Can you craft iptable rules that can block SYN flooding attacks? Explain your work and rationale.

iptables -A INPUT -p tcp --syn -m limit --limit 1/s --limit-burst 3 -j RETURN

Limits the amount of connections that can come in

1. Port knocking is a stealth method to externally open ports that, by default, the firewall keeps closed. It works by requiring connection attempts to a series of predefined closed ports. When the correct sequence of port "knocks" (connection attempts) is received, the firewall opens certain port(s) to allow a connection.

The benefit is that, for a regular port scan, it may appear as the service of the port is just not available. This article shows how to use port knocking with either a daemon or with iptables only.

1. Can you create a port knocking system only with iptable commands that will open up port 22 (SSH) when the remote host “knocks” on ports 2222, 3333, 4444 within 30 seconds?
2. Can an attacker defeat the port knocking system? How?