ESCOLA
POLITÉCNICA

Using firewall iptables in CORE Emulator

Redes de Computadores - Prof. Dr. Tiago Ferreto (tiago.ferreto@pucrs.br)
Student - Ângelo Crestani (angelo.crestani@edu.pucrs.br)

Agenda

1 Introduction

2 Firewall

3 Iptables

Agenda

4 NAT

5 Core Emulator

6 References

Introduction

The firewall systems are born at the end of the ’80s, with the necessity of protecting the computer networks of unwelcome accesses.

Introduction

A firewall is a **network security device** that monitors incoming and outgoing network traffic and decides whether to allow or block specific traffic based on a defined set of security rules.

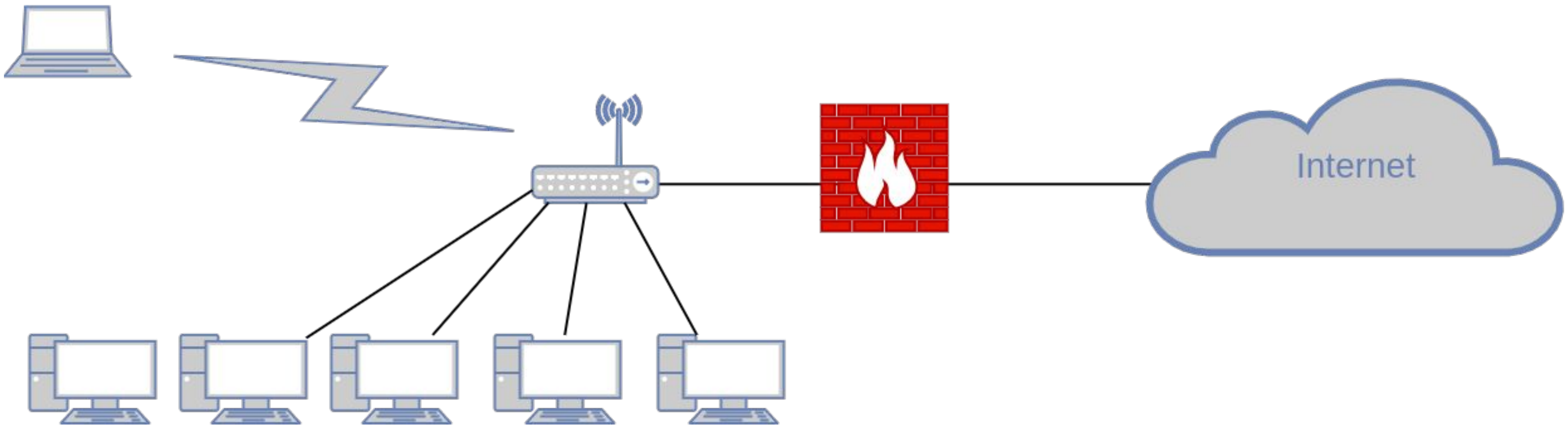
- Cisco

Firewall

Firewalls establish a barrier between secured and controlled internal networks that can be trusted and untrusted outside networks, such as the Internet.

A firewall can be **hardware**, **software**, or **both**.

Firewall



Firewall

The Linux kernel has packet filter since version 1.1

- Ipfw
- Ipfwadm (Kernel 2.0)
- Ipchains (Kernel 2.2)
- iptables (Kernel 2.4)

Iptables

Netfilter is a packet filtering framework inside the Linux Kernel that provides firewall functions.

Netfilter framework is controlled by the **iptables**.

Iptables

Iptables/Ip6tables is an administration tool for IPv4/IPv6 packet filtering and NAT.

Iptables and **ip6tables** are used to set up, maintain, and inspect the tables of IPv4 and IPv6 packet filter rules in the Linux kernel.

Iptables

Iptables organizes its rules into a structure that contains **tables** and **chains**. Tables are a grouping of chains at a higher level.

Several different tables may be defined. Each table contains a number of **built-in chains** and may also contain **user-defined chains**.

Iptables

Structure:

- **Tables**: Each table contains a number of built-in chains and may also contain user-defined chains.
  - **Chains**: Each chain is a list of rules which can match a set of packets.
    - **Rules**: Each rule specifies what to do with a packet that matches
      - **Targets**: Specifies what action is taken on packets matching the above rule.

Iptables

There are currently five independent **tables** in iptables.

- **Filter**: This is the default table for handling network packets.
- **NAT**: This table is consulted when a packet that creates a new connection is encountered, and to redirect connections to NAT.
- **Mangle**: This table is used for specialized packet alteration, such as modifying a packet's IP header options.
- **Raw**: This table is used mainly for configuring exceptions from connection tracking.
- **Security**: This table is used for Mandatory Access Control (MAC) networking rules.

Iptables

Existing **chains** currently in iptables.

- **INPUT**: Applies the rules to incoming network packets server.
- **OUTPUT**: Applies the rules to locally-generated network packets.
- **FORWARD**: Applies the rules for packets being routed through the firewall.
- **PREROUTING**: Chain for altering packets as soon as they come in.
- **POSTROUTING**: Chain for altering packets as they are about to go out.

Iptables

Tables with their respective **chains**.

- **Filter**
  - INPUT, FORWARD, OUTPUT.
- **NAT**
  - PREROUTING, INPUT, OUTPUT, POSTROUTING.
- **Mangle**
  - PREROUTING, INPUT, OUTPUT, FORWARD, POSTROUTING.
- **Raw**
  - PREROUTING, OUTPUT.
- **Security**
  - INPUT, OUTPUT, FORWARD.

Iptables

Some of the existing iptables **targets** .

- **ACCEPT:** Accepts the package
- **DROP:** Drops the package
- **REJECT:** Rejects the package
- **DNAT:** Rewrite destination address
- **SNAT:** Rewrite source address

Iptables

Commands

- Only root users can execute iptables commands
- Case-sensitive

```
iptables [-t table_name] COMMAND CHAIN_NAME matches -j TARGET
```

Iptables

Commands

Table	Command	Chain	Matches	Target/Jump
filter (default)	**-L** (list)	INPUT	**-4** (ipv4)	ACCEPT
NAT	**-S** (list_rules)	OUTPUT	**-6** (ipv6)	DROP
mangle	**-A** (append)	FORWARD	**-s** (source_ip)	REJECT
…	**-I** (insert)	PREROUTING	**-d** (destination_ip)	DNAT
	-D (delete_rule)	POSTROUTING	**-p** (protocol)	SNAT
	-F (flush)	USER_DEFINED	**-j** (jump_target)	…
	-R (replace)		**-i** (in_interface)	
	-P (policy)		**-o** (out_interface)	
	-N (new_chain)		**-v** (verbose)	
	-X (delete_chain)		**-n** (numeric)	
	…		**--line-numbers**	
			…	

Iptables

Commands examples:

- **iptables -A INPUT -p tcp --dport 80 -j DROP**
  Table filter
  -A append
  INPUT chain
  -p tcp protocol tcp
  --dport 80 destination port 80 (HTTP)
  -j DROP target DROP

Iptables

Commands examples:

- **iptables -I INPUT 3 -p udp --dport 69 -j DROP**
  Table filter
  -A insert
  INPUT chain
  3 rule position
  -p udp protocol udp
  --dport 69 destination port 69 (TFTP)
  -j DROP target DROP

Iptables

Commands examples:

- **iptables -F INPUT**
  Table filter
  -A flush
  INPUT chain

  This is equivalent to deleting all the rules one by one.

NAT

Network **A**ddress **T**ranslation is a method of rewrite one IP address that passes through a firewall or router, allowing a computer on a LAN to have the access to Internet.

NAT

Since **N**etwork **A**ddress **T**ranslation is also configured from the packet filter ruleset, **iptables** is used for this, too.

References

- Cisco
  - https://www.cisco.com
- Netfilter
  - https://netfilter.org
- Red Hat Enterprise Linux 4: Reference Guide
  - http://web.mit.edu/rhel-doc/4/RH-DOCS/rhel-rg-en-4/ch-iptables.html

Thanks!

Does anyone have any questions?

angelo.crestani@edu.pucrs.br