(https://linuxacademy.com/cp/dashboard)
(/cp)

(https://www.cloudassessments.com/p/#/dashboard)

(https://scaleyourcode.com)

19 🔔 Latest Updates (https://linuxacademy.com/blog)Refer A Friend (/cp/referFriend)Support ⚙ xe1phix 👤

≡
Navigation

←
Back to community

Fernando Panizza (/cp/socialize/profile/user/nandoz)

●

●

●

- Bookmark this post
- Sticky this post
- Follow this user
- Share!
- Spam!

How To Guide

Introduction to IPTables

3/8/2017

✹ Instructor Approved

Table of Contents

Introduction

This guide is meant as an introduction to iptables and covers basic packet filtering using this tool on Centos 7. The goal is to learn to work with some basic options so you can adapt them to your particular needs. For this reason we will not cover any iptables extensions in this guide.

Getting Started

The term iptables is used to define the Linux kernel firewall, part of the Netfilter project and the user space tool used to configure this firewall. The Netfilter framework provides a set of facilities that are used by iptables to hook functions that are designed to perform operations on packets. The Netfilter project defines it as *"a generic table structure for the definition of rulsets"*.

The structure of iptables is based on tables, chains and rules.

Installation

To install iptables in Centos 7 is very simple. Just run:

```
yum install iptables iptables-services
```

Starting and stopping the service

To start or stop the iptables service the following commands are used.

```
systemctl start iptables
```

```
systemctl stop iptables
```

or

```
service iptables start
```

```
service iptables stop.
```

Tables

Tables are used to group different functionalities and are separated in five:

- **filter**: used for packet filtering, is the default table if no table is specified.
- **nat**: used for network address translation (NAT)
- **raw**: used for configuring exemptions from connection tracking, and it is checked before any other table.
- **mangle**: used for specialized packet alteration
- **security**: used for Mandatory Access Control (MAC) networking rules

Tables are specified with the `-t (--table)` option.

Chains

Each table contains a set of chains that group the rules on different points of the process, these could be built-in chains or user defined. Since this guide will focus mainly on the filter table we will describe the built-in chains **INPUT**, **FORWARD** and **OUTPUT**.

- The **INPUT** chain handles packets that are destined to the local system.
- The **OUTPUT** chain is for packets that are locally generated.
- The **FORWARD** chain is used for packets that are routed through the system.

User defined chains are added with the `-N (--new-chain)` option and deleted with the `-X (--delete-chain)` option.

Rules

Rules are defined as a set of matches and a target. Rules are listed in chains and are followed by order until a match is found, then the packet is handled by the target specified by the rule.

Matches

A match is a condition to be met so that the packet can be processed by iptables. There are several matches that could be used, in this guide we will use the following:

`-s (--source)` : specifies the source of the packet, it can be an ip, a hostname or a network ip address.

`-d (--destination)` : specifies the destination, it can adopt any of the `--source` values.

`-p (--protocol)` : match the protocol used (e.g: tcp), the word all can be used to match all protocols and is the default if no protocol is specified.

`-i (--in-interface)` : match the interface that receives the packet `-o (--out-interface)` : match the output interface.

We can use the `!` operator to specify everything that's not in the match.

Targets

Targets are used to determine the action to be taken once a packet matches the rule and are specified with the `-j (--jump)` option. There are four built-in targets:

ACCEPT: allows the packet to continue without further checking.

DROP: refuse access for that packet without sending a response.

QUEUE: sends the packet to user space.

RETURN: returns the packet to the next rule in the previous chain, if this target is reached within a built-in chain the packet is handled by the chain policy.

Rules are added with the `-A (--append)` option and deleted with the `-D (--delete)` option along with the chain in which is contained, rules can be viewed with the `-L (--list)` or `-S (--list-rules)` options.

Default Policies

Default policies are used when a packet does not match any rule on the chain, in this case packets are handled by the target specified in the policy of that chain. A default policy is set with the `-P (--policy)` option passing the chain and the target as arguments.

There are two approaches when setting a policy, one is to accept everything by default and start adding rules to refuse access and the other is to refuse access by default and allow the necessary connections.

Practical Examples

Note: Before starting adding rules it is recommended to have local access to the system, if you're using ssh, configure ssh rules before changing the policies of the chains in the filter table. You can lock yourself out of the system! Iptables will start working once the module is loaded and in Centos 7 this occurs after a successful iptables command is typed (even when the service is stopped).

List rules:

The following command lists every rule in the INPUT chain in the filter table:

```
iptables -L INPUT
```

same as:

```
iptables -t filter -L INPUT
```

The use of the `-n (--numeric)` option prints ip addresses and ports in numeric format.

```
iptables -L -n INPUT
```

We can also add the `-v (--verbose)` option to get more detail.

Printing rules:

The following command print rules in the INPUT chain of the filter table in a usable format.

```
iptables -S INPUT
```

Setting a policy:

To accept packets by default:

```
iptables -t filter -P INPUT ACCEPT
```

To drop packets by default:

```
iptables -t filter -P INPUT DROP
```

Filtering IP addresses:

The following command sets a rule that drops every packet from the ip address 192.168.1.15 to 192.168.1.20. Since no protocol is specified it will assume `all` by default.

```
iptables -A INPUT -s 192.168.1.15 -d 192.168.1.20 -j DROP
```

In this case, we block everything outside the network 192.168.1.0/24 by the usage of the `!` operator.

```
iptables -A INPUT ! -s 192.168.1.0/24 -d 192.168.1.20 -j DROP
```

Note: If no source or destination is specified, the default route 0.0.0.0/0 is assumed.

Using protocols:

Protocols usually have their own set of matches that can be used. This can be viewed by using the `-h (--help)` option along with the `-p` protocol, it will print an iptables help message along with the options for the protocol.

Print possible matches for icmp protocol:

```
iptables -p icmp -h
```

Print possible matches for tcp protocol:

```
iptables -p tcp -h
```

For example, the following rule will allow tcp packets on port 22 on the interface eth0 by using the `--dport (destination port)` and `--sport (source port)` which are matches for the tcp protocol.

```
iptables -A INPUT -p tcp -i eth0 --dport 22 -j ACCEPT
```

```
iptables -A OUTPUT -p tcp -o eth0 --sport 22 -j ACCEPT
```

We can also specify a range of ports with these options, in the example below, we allow tcp packets to ports 22, 23, 24, and 25.

```
iptables -A INPUT -p tcp -i eth0 --dport 22:25 -j ACCEPT
```

```
iptables -A OUTPUT -p tcp -o eth0 --sport 22:25 -j ACCEPT
```

In the next example, we allow incoming icmp echo-requests (ping) from the network 192.168.1.0/24 to the interface eth0:

```
iptables -A INPUT -p icmp --icmp-type echo-request -s 192.168.1.0/24 -i eth0 -j ACCEPT
```

and outgoing echo-reply (pong) from eth0 to the network 192.168.1.0/24:

```
iptables -A OUTPUT -p icmp --icmp-type echo-reply -o eth0 -d 192.168.1.0/24 -j ACCEPT
```

Deleting and flushing rules

Rules can be flushed using the `-F (--flush)` option along with the name of the chain. If no chain is specified, it will flush the rules in all of the chains of the table.

The following deletes all rules in the raw table:

```
iptables -t raw -F
```

And this deletes all rules in the INPUT chain of the filter table:

```
iptables -F INPUT
```

To delete a particular rule, the `-D` option is used. This can be done in two ways:

1. Writing the rule that we want to delete.
2. Using the number of the rule along with the chain.

This deletes the rule we added in the previous example:

```
iptables -D OUTPUT -p icmp --icmp-type echo-reply -o eth0 -d 192.168.1.0/24 -j ACCEPT
```

This deletes the rule with the index 7:

```
iptables -D INPUT 7
```

To view the numbers of the rules, we can use the option `--line-numbers` along with the `-L` option

```
iptables -L --line-numbers
```

Persisting iptables rules

Persistence can be achieved through the use of `iptables-save` and `iptables-restore` commands. `iptables-save` dumps the rules to stdout so in order to save the rule to a file we need to use redirection. This command writes the current rules to a file named persistent.rules:

```
iptables-save > persistent.rules
```

To load those rules we can use iptables-restore. This command loads the rules contained in the file persistent.rules:

```
iptables-restore < persistent.rules
```

By default rules are stored in /etc/sysconfig/iptables, those are the rules that are loaded once the service is started or reloaded. Running the following command will save the rules so that they can be loaded every time the service start:

```
iptables-save > /etc/sysconfig/iptables
```

Another option is to use the command:

```
service iptables save
```

This command has the same result as the previous one.

Sources / Resources

- LPIC-2: Linux Engineer Exam 202 (https://linuxacademy.com/cp/modules/view/id/111)
- Linux+ LPIC Level 1 Exam 2 (https://linuxacademy.com/cp/modules/view/id/2)
- Linux Foundation Certified Systems Engineer (https://linuxacademy.com/cp/modules/view/id/58)
- Building a Firewall with IPTables (Nugget) (https://linuxacademy.com/cp/nuggets/view/id/86)
- www.netfilter.org. What is netfilter.org?. (http://www.netfilter.org/)
- www.netfilter.org. What is iptables?. (http://www.netfilter.org/projects/iptables/index.html)
- Archlinux wiki. iptables. (https://wiki.archlinux.org/index.php/Iptables)
- Centos Documentation. iptables. (https://www.centos.org/docs/5/html/Deployment_Guide-en-US/ch-iptables.html)

Rash, Michael. Linux Firewalls: Attack Detection and Response with iptables, psad, and fwsnort. No Starch Press © 2007.

🗨 12 12 + −

... Show Previous Comments

Nizam Uddin (/cp/socialize/profile/user/durjoy) 7/27/2017
Thank you, Fernando Panizza for an easy to follow, step by step tutorials.

❮ 0 ⊞ ⊟ ❗

Ashish Barot (/cp/socialize/profile/user/barotashish) 7/27/2017

Nice document Fernando. I like the method and step by step way to explain IPTable.
It is really helpful to any new user who is learning Linux security.

Good work, keep this module updated with some more day to day examples with IPtables.

Regards,
Ashish Barot.

❮ 0 ⊞ ⊟ ❗

Reply

^