Iptables Firewall – A basic introduction

You probably heard of the term “IPtables Firewall” before. **Iptables** is used to set up, maintain, and inspect the tables of IP packet filter rules in the Linux kernel. This quick tutorial will teach you the basics about building a firewall by using iptables. First you need to know how a firewall handles packets leaving, entering, or passing through your server. Think about chains for each of the mentioned events. Any IP packet that is destined to your server has to go through the INPUT chain. Any packet that a process on your server sends out has to go through the OUTPUT chain. Any packet that wants to travel through your server to another host in the network has to got through the FORWARD chain. The chains represent the logic behind the whole iptables thing.

The way iptables work is by setting up certain rules for the chains. These rules allow the chains to inspect each incoming or outgoing package and then to apply the proper rules. For instance, if your server receives an incoming request for the website “exampledomain.com” the request would first be inspected by the chain for incoming traffic to your server. Let’s assume that the request comes from an IP address that should not have access to the website. The IP address is listed in the rules to be denied. The rule recognizes the IP of the requester in the rules and the IPtables firewall blocks the request from going through the firewall to reach the web server part on your server. The requestor would not get the website to see. As an example – you want to block all incoming traffic to your website from 10.1.1.25 (for the matter of this case we use a private IP address).

A very broad IPtables command would be:

```
iptables -A INPUT -p tcp -j ACCEPT
```

This rule would accept all tcp traffic. But this is a little too broad isn’t it? So, let’s work on being much more specific in regards to blocking incoming requests from the IP address specified.

Please be aware that ”-s" is used to specify a source IP or DNS name. So, for our example this would mean:

```
iptables -s 10.1.1.25
```

Now that we have specified the source IP address we need to tell the firewall of what to do if a request comes from that IP address. The “-j“ option is used to specify what happens to the incoming request from that IP address. The most common three settings are “ACCEPT“, “REJECT“, and “DROP“. “ACCEPT” would let traffic from the source IP address pass through the firewall. “REJECT” would send a message to the requestor that this server isn’t accepting connections. “DROP“ just ignores the incoming request and drops it. The requester would not get a response at all. For our example we would either use “DROP” or “REJECT” as the preferred option:

```
iptables -s 10.1.1.25 -j DROP
```

But we’re not done yet. Our server still won’t understand what we are trying to accomplish. We still need to specify the “INPUT” or “OUTPUT” chain. Since we want to deny access to the website from this specific requestor we would need to apply this setting to the “INPUT” chain.

```
iptables -A INPUT -s 10.1.1.25 -j DROP
```

This command would ignore every incoming request from 10.1.1.25 (with some exceptions, but we’ll get into that part later on). The order of the specified options doesn’t matter. The “-j DROP“ could go before “-s 10.1.1.25″. But you should use a consistent approach to avoid confusion down the road.

Sometimes you will need to be more specific when applying IPtables rules. Let’s modify the example to block only TELNET requests. We need to specify the protocol (here: TCP) and the port or service (here: TELNET).

```
iptables -A INPUT -s 10.1.1.25 -p tcp –destination-port telnet -j DROP
```

If you wanted to block all requests from a whole IP address subnet with a destination that is on a different host in the network you need to modify the command as follows:

```
iptables -A FORWARD-s 10.1.1.0/24 -p tcp –destination-port telnet -j DROP
```

Here is a list of some additional parameters that can be used when working with IPtables:

-j	Specifies the target (–jump)
-i	Specifies the input interface (–in-interface)
-o	Specifies the output interface (–out-interface)
-p	Specifies the protocol (–proto)
-s	Specifies the source address (–source)
-d	Specifies the destination address (–destination)
--source-ports	Specifies the source ports
--destination-ports	Specifies the destination ports
-P	Specifies the default policy of the chain (TARGET)
--mac-source	Specifies the mac address
!	Inverts the sense of the filter rule (match addresses NOT equal to)

Examples:

```
iptables -A INPUT -p tcp -j ACCEPT
iptables -s 10.1.1.25 -j DROP
iptables -A INPUT -s 10.1.1.25 -p tcp –destination-port 23 -j DROP
iptables -A INPUT -s 10.1.1.0/24 -p tcp –destination-port 23 -j DROP

iptables -A FORWARD -p tcp -s 192.168.10.0/255.255.255.0 --dport 80 -j REJECT
```