Application Note AN2010-10

Securely Configuring a Linux® Firewall

Jason Kraft

INTRODUCTION

This application note describes how enabling and tuning the network firewall can help prevent unauthorized access to a Linux-based operating system. Linux uses iptables as the basis for configuring its firewall. Iptables has been integrated into the Linux kernel since Version 2.4 (the year 2000).

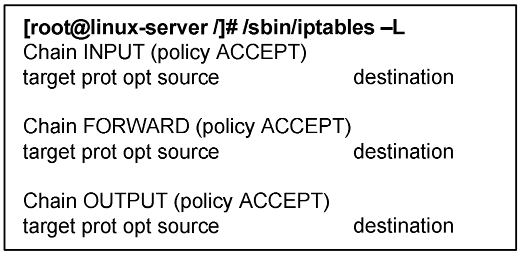
PROBLEM

The default firewall configuration for most Linux distributions permits both inbound and outbound traffic. Having this setup provides no security because all traffic is permitted. The firewall is usually set up this way to allow end users to configure rules that are applicable to their environment.

Firewalls are typically configured in one of two ways. The first is to permit all traffic and then have exceptions to deny certain network traffic. This method is usually used when a network attack is occurring and you need to deny traffic for a very specific reason. The second, and preferable method, is to deny all traffic by default and then have exceptions that allow certain network traffic. This requires you to know exactly what network traffic should be permitted on your system.

SOLUTION

The first step is to determine what rules are currently applied to your system. This can be accomplished by running the **iptables** command as the root user. Figure 1 shows the firewall rules currently configured on the system.



```
[root@linux-server /]# /sbin/iptables –L
Chain INPUT (policy ACCEPT)
target prot opt source              destination

Chain FORWARD (policy ACCEPT)
target prot opt source              destination

Chain OUTPUT (policy ACCEPT)
target prot opt source              destination
```

Figure 1 List the firewall rules

In this example, both inbound (INPUT) and outbound (OUTPUT) traffic are set to ACCEPT, which permits all traffic. The third traffic type is called FORWARD, which is typically used for port forwarding or Network Address Translation (NAT) applications, and is set to permit all traffic.

The next step is to deny all inbound network traffic by default. **Caution**: if you perform this step over a Telnet or Secure Shell (SSH) connection, you will drop your current incoming connections. This step is best performed over a console or serial connection. Also be aware that if there are remote computers sending data to this server, the data will be blocked until a permit rule has been written.

```
[root@linux-server /]# /sbin/iptables –P INPUT DROP
[root@linux-server /]# /sbin/iptables –P FORWARD DROP
[root@linux-server /]# /sbin/iptables –A INPUT –m conntrack –ctstate ESTABLISHED,RELATED –j ACCEPT
[root@linux-server /]# /sbin/iptables –A INPUT –m limit –limit 5/min –j LOG –log-prefix “iptables inbound-denied:”
```

Figure 2 Drop inbound traffic by default

At this point, all inbound and forward traffic is blocked per the first two rules. The third rule is used to allow incoming traffic where an outbound connection has already been established. An example is when you make an outbound SSH connection from the Linux server. If you did not have this rule, the Transmission Control Protocol (TCP) three-way handshake would fail and you would not be able to make an outbound SSH connection. The fourth and most useful rule is for logging all denied traffic. Logging can be an invaluable resource for both detecting unauthorized access attempts and troubleshooting firewall rules. By enabling the logging functionality, you see all traffic that is blocked—by default, the log messages are saved in the **/var/log/messages** file.

At this point, all inbound traffic is blocked, and all outbound traffic is permitted. The next step is to allow inbound traffic that is necessary to operate the Linux server. Figure 3 shows a rule that allows inbound SSH for engineering remote access for a single client Internet Protocol (IP) address. After this rule has been applied, it is no longer necessary to apply rules from the console or serial connection. Instead, you can use SSH to connect to the server and run these commands over a secure channel.

```
[root@linux-server /]# /sbin/iptables –A INPUT –p tcp –s 192.168.1.100 – –dport 22 –m state – –state NEW –j
ACCEPT
```

Figure 3 Accept inbound SSH traffic from a single IP address

As you can see from Figure 3, the rules can be adapted to be specific so that the firewall is very restrictive. If you wanted to permit an IP range instead of a single IP address, you could replace **192.168.1.100** with **192.168.1.0/24** to permit an entire block of IP addresses.

You can test the rule either by using an SSH client such as PuTTY [1] to connect to the server or by using Telnet [2] and specifying Port 22 as an option, which is the standard port for SSH. The two command prompts in Figure 4 show the typical output for a success and failure. If you see the second command prompt, it means either the iptables firewall is misconfigured or a network-based firewall is denying access.

```
C:\Windows> telnet linux-server 22
SSH-2.0-OpenSSH_4.3

C:\Windows> telnet linux-server 22
Connecting to linux-server...Cound not open connection to the host, on port 22: Connect failed
```

Figure 4 Test the rule from the client computer

As you can see, the Linux firewall is powerful and rich in features. With a few commands, you can harden your Linux device to prevent unauthorized access via IP networks. Please consult your security advisor for further information on configuring your firewall for your environment.

For more information, please visit the following web locations:

- Ubuntu Documentation, *Iptables How To*, Available: https://help.ubuntu.com/community/IptablesHowTo, last accessed on November 19, 2010.
- Linux Home Networking, *Quick HOWTO : Ch 14 : Linux Firewalls Using iptables*, Available: https://www.linuxhomenetworking.com/wiki/index.php/Quick_HOWTO_:_Ch14_:_Linux_Firewalls_Using_iptables, last accessed on November 19, 2010.

References

[1] PuTTY: A Free Telnet/SSH Client. Available: http://www.chiark.greenend.org.uk/~sgtatham/putty/.

[2] Telnet: Frequently Asked Questions. Available: http://windows.microsoft.com/en-US/windows-vista/Telnet-frequently-asked-questions.

SCHWEITZER ENGINEERING LABORATORIES, INC.
2350 NE Hopkins Court • Pullman, WA 99163-5603 USA
Tel: +1.509.332.1890 • Fax: +1.509.332.7990
www.selinc.com • info@selinc.com