IPTables for Industrial Control Systems (iptables-ics)

1 Overview

This Labtainer exercise illustrates the use of iptables to limit network access to a PLC component in an operational technology (OT) environment. This control is provided by a component serving as a firewall, as illustrated in Figure **??**.

When properly configured, the firewall will only allow the following traffic between the clients and the PLC:

- Client 1 can only access the PLC via SSH and HTTP (port 8080).
- Client 2 can only access the PLC via MODBUS TCP and HTTP (ports 80 and 8080).

1.1 Background

Industrial control systems often use IP based networks to communicate with other components. Just as is the case with many information technology (IT) networks, protection of assets may depend on limiting the types of network traffic permitted to flow between components. For example, web traffic (e.g., HTTP) might only be permitted to enter a given component if it originates from specific sources.

A variety of different techniques and products exist for the purpose of limiting IP traffic in IT and OT systems. In this lab, you will limit IP traffic through use of Linux iptables. The student is expected to have separately learned about the use of iptables to selectively block network traffic. The firewall component includes a few example firewall setting scripts that you can reference. The manpage for iptables can be viewed on the firewall component using:

```
man iptables
```

Students are expected to have a basic familiarity with the Linux command line, and the ability to edit files and run simple shell scripts. Some experience with Wireshark is presumed, e.g., performance of the wireshark-intro lab.

2 Lab Environment

This lab runs in the Labtainer framework, available at http://my.nps.edu/web/c3o/labtainers. That site includes links to a pre-built virtual machine that has Labtainers installed, however Labtainers can be run on any Linux host that supports Docker containers.

From your labtainer-student directory start the lab using:

```
labtainer iptables-ics
```

A link to this lab manual will be displayed.

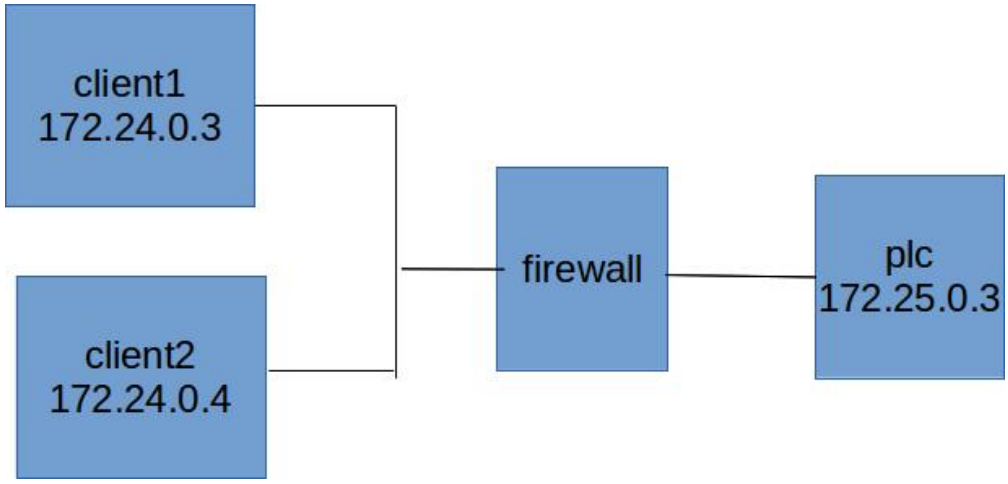

Figure 1: Network topology for the iptables-ics lab

3 Lab Tasks

3.1 Explore

The Wireshark utility is installed on the firewall. Use it to view network traffic through the firewall, and to debug your firewall rules. Start it from the firewall terminal:

```
wireshark &
```

Then select the eth0 interface.

On both the client1 and client2 terminals use the following applications to explore the services offered by PLC:

```
./mbtcp-simple.py
```

This starts a simple MODBUS client. Observe the network traffic using wireshark, making note of the destination TCP port number used by the client when connecting to the PLC. Then use `<ctrl C>` to terminate the program on each client.
Start firefox on each client:

```
firefox &
```

and point the browswer to the following two URLs:

```
http://plc:8080
http://plc
```

Observe the traffic in wireshark, making note the source IP addresses and the destination ports used by the clients when connecting to the PLC.
Finally, use SSH to connect to the PLC from each client:

```
ssh plc
```

There is no need to login. After observing the initial traffic, use `<ctrl C>` to exit from ssh.

3.2 Use iptables to limit traffic

The iptables utility is installed on the "firewall" component. Use it to prevent the firewall from forwarding any traffic to the PLC other than specified below.

1. Client 1 can only access the PLC via SSH and HTTP (with port 8080 only).
2. Client 2 can only access the PLC via MODBUS TCP and HTTP (with ports 8080 and 80).
3. No other network traffic is permitted to reach the PLC

You may reference and experiment with the example firewall scripts that are on the firewall component in the home directory. **NOTE:** The IP addresses in the example script may not correspond to the IP addresses of your system, and would therefore need to be changed. To run the `example_fw.sh` script, use:

```
sudo ./example_fw.sh
```

View the content of the scripts to understand what they do. Consider putting your iptables commands in a script so it is easy to test and reconfigure the iptables if you restart the lab. Note that the last line of `example_fw.sh` directs iptables to send log messages to a file that is at:

```
/var/log/iptables.log
```

If you include that directive in your configuration, you can observe when iptables drops filtered packets, e.g., by tailing the log from one of the firewall terminal tabs:

```
tail -f /var/log/iptables.log
```

After configuring iptables to meet the requirements, use the applications on Client1 and Client2 to demonstrate that the firewall only allows the desired traffic. Watch the traffic in wireshark to confirm the TCP handshake fails when attempting to connect to filtered ports.

When you believe you have the proper iptables configuration and have tested it, use the `stoplab` command from your Linux system. You are free to then restart the lab using `labtainer iptables-ics` and explore further without concern for the state of the system when you again stop it. Alternately, if the `checkwork` command is available on your distribution, you can use that command from the Linux system instead of `stoplab`, and then just continue working.

4 Submission

After finishing the lab, go to the terminal on your Linux system that was used to start the lab and type:

```
stoplab
```

When you stop the lab, the system will display a path to the zipped lab results on your Linux system. Provide that file to your instructor, e.g., via the Sakai site.