12.6.1 Linux Firewalls

Click one of the buttons to take you to that part of the video.

Linux Firewalls 0:00-0:36

In this lesson, we're going to talk about firewalls. Generally, the term "firewall" refers to a barrier that keeps fire from spreading. In the computing world, it refers to a software- or hardware-based network security system that uses a set of rules to control incoming and outgoing network traffic.

A firewall basically establishes a barrier between the internal network, which is assumed to be secure and trusted, and the external network, which is usually the internet and is not secure or trusted. Most operating systems, including Linux, offer software-based firewalls to protect networks and systems from external threats.

Access Control Lists 0:37-1:42

As firewalls came into use, they functioned according to network access control lists, or ACLs. ACLs were the rules the firewall used to process IP packets. Today, Linux firewall technologies often use different methods to organize their configuration options. But ACL concepts are still at the heart of a firewall's design.

ACLs determine whether routed packets are accepted, rejected, or dropped. Accepted packets are forwarded on to their destinations. Rejected packets are blocked, and a message is sent back to the packet's sender. Dropped packets are also blocked, but no message is sent.

Firewalls are divided into two categories, stateless and stateful. ACLs are stateless firewall filters because they use the packet's source address, destination address, ports, and protocols to determine how to filter the packet. A stateful firewall looks at traffic patterns, tunneling, and encryption to determine how to filter packets.

For security and analysis, you can set ACLs to log each time they're used to filter a packet. Although it can be CPU-intensive, you can also set a logging option to capture packets that the ACL targets.

Netfilter 1:43-2:12

There are many third-party Linux firewalls, and a few of them are included in major Linux distributions. But before we get into that, let's talk about Netfilter.

Netfilter is part of the Linux kernel. It's used for network address translation and port translation. It also supplies the kernel's IP packet filtering functions, which are used by firewall applications.

This means that Linux firewall applications, even those included in Linux distributions, interface with Netfilter to perform firewall functions.

IPTables 2:13-3:47

IPTables is a firewall application that's pre-installed on most Linux distributions. It is a rule-based front-end tool that interfaces with Netfilter to decide which packets to filter.

Internally, IPTables consists of five pre-defined tables that contain chains. The kernel accesses each chain at a specific point while processing an IP packet, and each chain has its own purpose. IPTables is installed with pre-configured chains. You'll rarely need to create custom chains. Instead, you can add, delete, and customize the rules contained in each chain.

Each chain's rules are traversed in order. Each rule has fields that are matched against the IP packet. If a match is made, the action in the rule is taken, and then no more rules in the chain are checked. For example, the packet could be accepted, rejected, or dropped; any of these actions completes the process, and no more chains are checked.

If the packet doesn't match the rule, the rule is skipped, and the next rule is checked. This continues until a match is made. Normally, the last rule is configured with wildcards so that it matches any packet. In many cases, the action for the last rule is to reject the packet.

Although incredibly powerful, IPTables is complicated, especially for newer users. Fortunately, to configure a firewall, you only need to modify the chains in the filter table. There are three preconfigured chains, INPUT, FORWARD, and OUTPUT. The chain names indicate the

point in the IP packet processing when the chain is accessed. This means that you only need to modify three sets of rules to create an IPTables firewall.

Uncomplicated Firewall 3:48-4:05

Uncomplicated Firewall (UFW) provides a much more user-friendly framework for managing Netfilter and a command line interface for working with the firewall. On top of that, if you'd rather not deal with the command line, UFW has a few GUI tools that make working with the system incredibly simple.

The firewalld Firewall 4:06-5:02

firewalld is a third Linux firewall that's pre-installed on many Linux distributions. It's a front-end controller for IPTables. When you install it, you can still use the IPTables commands to configure your firewall. Its strong point is that it has both a command line and graphical interface.

The firewalld application uses zones and services instead of chains and rules. Zones are pre-constructed rulesets for various trust levels. They're similar to Microsoft Windows security zones. Different zones allow different network services, ports, protocols, and incoming traffic types, while denying everything else.

You can apply a zone to different network interfaces. For example, if your firewall connects two networks, you can allow DHCP on the internal zone and deny it on the external zone. You can also configure firewalld with rules to allow traffic for specific network services. And you can add custom service rules to any zone.

Linux IP Forwarding 5:03-6:03

There are some other topics that relate to firewalls. First, there's IP forwarding. IP forwarding is another name for routing. It's sometimes called kernel IP forwarding because it's a feature of the Linux kernel.

A firewall can be thought of as a special type of router. Normally, routers don't filter network traffic. If traffic comes in on one interface that matches the subnet of another network interface, the router forwards the traffic to the other network interface. Firewalls do the same thing, but only after checking whether a rule prevents forwarding.

You can enable IP forwarding by writing a 1 to the ip_forward file. To enable IPv4 forwarding, you write to the /proc/sys/net/ipv4/ip_forward file. To enable IPv6 forwarding, you write to the /proc/sys/net/ipv6/ip_forward file.

Be cautious about enabling IP forwarding without a firewall, especially if one of your interfaces connects to the internet or to a subnet you don't control.

Dynamic Rule Sets 6:04-6:35

Dynamic rule sets are another useful firewall tool. They automate the rules IPTables use to filter network traffic and prevent intrusions. There are two popular Python scripts that are classified as intrusion prevention software, DenyHosts and Fail2ban. Both monitor log files and react to common security problems, such as brute-force attacks, by adding or modifying firewall rules.

Another companion application to IPTables is IPset. This tool allows you to easily set firewall rules for a block of IP addresses.

Common Firewall Configurations 6:36-7:10

Finally, there are a couple of topics that are common to all firewalls. Many firewall applications read from the /etc/services file. This file is a list of well-known services and their port assignments. When you update firewall rules, consider updating this file with new services and ports.

Also, the ports from 1 to 1023 are privileged ports. Only the root account has access to these ports. This can give you confidence in internal networks, where only trusted individuals have passwords to the root account. Internal firewalls may be more tolerant when passing network traffic using these ports.

Summary 7:11-7:25

And that's it for this video. In this lesson, we talked about firewalls and ACLs. We discussed several Linux firewalls, including Netfilter, IPTables, Uncomplicated Firewall, and firewalld. We also covered several related topics: IP forwarding, dynamic rule sets, and common firewall configurations.
