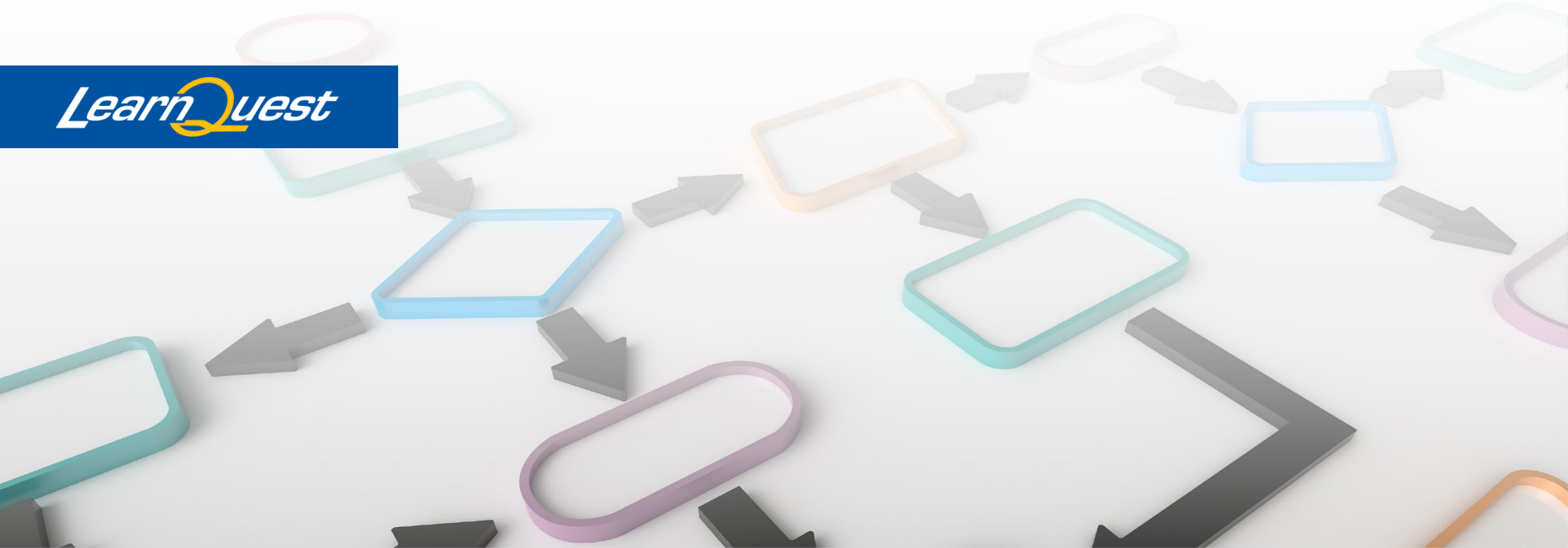
Securing Linux Systems

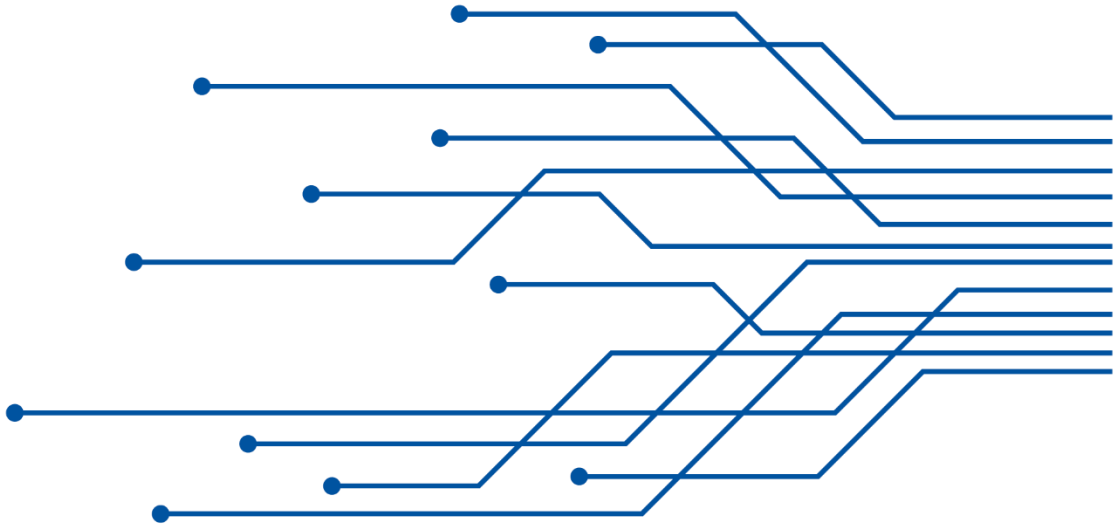
- 3rd Course in Linux Foundations Specialization



Linux Firewalls

In this module, we look at security at the boundaries of your network and your host Linux system.

4



Learning Objectives

Linux Firewalls

Upon completion of this module, learners will be able to:

- Implement ACL in the Linux Firewall
- List Linux Firewall Technologies
- Forward IP Packets
- Describe Intrusion Detection Systems

Lesson 1

ACL in the Firewalls

In this lesson, we look ACL in firewalls

Firewall ACL

Firewalls provide access control to your system or network.

A firewall identifies which network packets are allowed in or out.

This is referred to as packet filtering.

Firewall Data

A firewall ACL identifies a network packet by reviewing its control information along with other network data. This include the following information:

- Source address
- Destination address
- Network protocol
- Inbound port
- Outbound port
- Network state

Packet Filtering

Once a network packet is identified, the firewall's ACL rules decide what happens to that packet. The rules include the following actions:

- Accept – Allow packet
- Reject – Response sent to client
- Drop – No Response sent to client
- Log – Allow packet but write to log

Stateless vs Stateful

Firewalls can operate in a stateful or stateless manner

Stateless: Focuses on individual packets

Stateful:

- Treats packages as a collection
- Not vulnerable to attacks propagated over multiple packets
- Firewall tracks active connection's packets

Lesson 1 Review

Firewalls determine what packets get through to machine or network

Clients get a response on rejected packets

Clients do not get a response on dropped packets

Lesson 2

Firewall Technologies

In this lesson, we look at different Linux firewall technologies

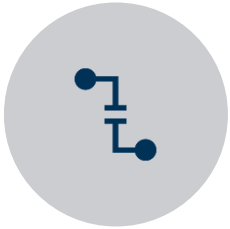
Netfilter

netfilter in embedded in the Linux kernel

Provides packet filter services to applications

Used by several programs including: iptables, firewalld, and UFW

Firewalld



Dynamic firewall can change an ACL rule without restarting the service

Zones: A predefined rule set for a network traffic group

Trust Level: A configuration file that defines a zone rule set

Default trust levels stored in /usr/lib/firewalld/zones/

Customized trust levels stored in /etc/firewalld/zones/

Predefined Firewalld Zones

drop - Drops all incoming network packets. Only outbound network connections allowed.

block - Accepts only network connections that originated on the system.

public - Accepts only selected incoming network connections. Typically used in a public setting, where other systems on network are not trusted.

external - Like public but is typically used on external networks, when masquerading is enabled for the local systems.

dmz - Like public but is used in a location's demilitarized zone, which is publicly accessible and has limited access to the internal network.

work - Accepts only selected incoming network connections. Typically used in a work setting, where other systems on the network are mostly trusted.

home - Like work but is used in a home setting, where other systems on the network are mostly trusted.

internal - Like work but is typically used on internal networks, where other systems on the network are mostly trusted.

trusted - Accepts all network connections.

IpTables Firewall

Uses a series process called chains to handle network packets

Chains determine the path each packet takes to reach the end application

Each chain contains tables to define rules:

filter applies rules to allow/block packets

mangle applies rules to change packet features

nat applies rules to change packet's address

raw applies NOTRACK to designate no tracking on packet

security applies mandatory access control rules

Iptables Command

View and alter the chains and filters in the iptables service

Example Usage:

- iptables –L
- iptables -P OUTPUT ACCEPT

Options:

- -a: Adds this new rule to a chain
- -D: Delete rule to a chain
- -P: Defines this default policy for a chain.

Ufw Command

Manage the Uncomplicated Firewall (UFW) service

Example Usage:

- ufw enable
- ufw status verbose
- ufw disable

Options:

- allow - Sets the rule identified by Identifiers to allow packets
- deny - Sets the rule identified by Identifiers to deny (drop) packets
- reject - Sets the rule identified by Identifiers to reject packets
- delete - Deletes the rule identified
- insert - Inserts the rule
- logging - Sets the logging level

Lesson 2 Review

Netfilter is embedded in the Linux kernel and used by tools.

Firewalld can change an ACL rule without restarting the service.

The Uncomplicated Firewall (UFW) is the default firewall service on Ubuntu distributions.

Lesson 3

Packet Forwarding

In this lesson, we look at how Linux allows packet forwarding

IP Packet Forwarding

IP forwarding is the ability for an operating system to accept incoming network packets on one interface, recognize that it is not meant for the system itself, and forwards it appropriately.

If the Linux server is acting as a firewall, router, or NAT device, it will need to be capable of forwarding packets that are meant for other destinations.

Sysctl and IP Forwarding

- **To enable that IP Forwarding, set the ip_forward entry for IPv4 or the forwarding entry for IPv6. The sysctl command can do this:**
  - sudo sysctl –w net.ipv4.ip_forward=1
  - sudo sysctl -w net.ipv6.conf.all.forwarding=1

IPset

Allows rule change without having to type whole
IP/MAC address repeatedly

Named set of

IP addresses	Network interfaces	Ports	MAC addresses

Lesson 3 Review

IP forwarding is built into Linux kernel

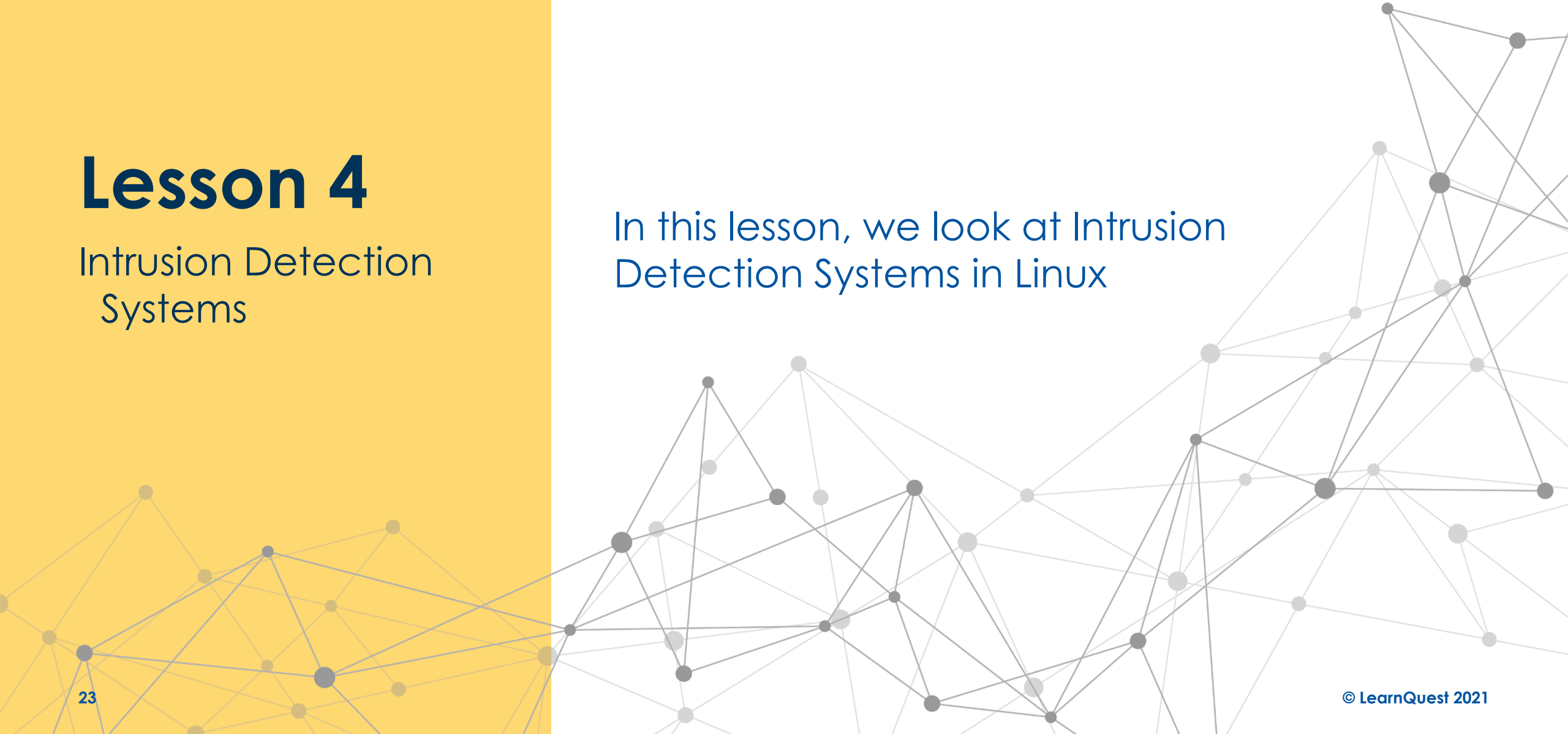
Sysctl is used to turn on IP forwarding

Ipset allows rule change without having to type whole address each time

Lesson 4

Intrusion Detection Systems

In this lesson, we look at Intrusion Detection Systems in Linux



Linux Intrusion Detection Systems

To protect your Linux system, you want to utilize software that monitors the network and applications running on the system, looking for suspicious behavior.

We call applications in this category intrusion detection systems (IDSs).

Some IDS applications allow you to dynamically change rules so that these attacks are blocked.

Two Linux IDS programs

- DenyHosts
- Fail2Ban

DenyHosts

Python script

Protects against brute-force attacks via OpenSSH

If the script sees repeated failed authentication attempts from same host it will add its IP address to /etc/hosts.deny file

Fail2ban

Monitors system logs looking for repeated failures from the same host

The fail2ban-client program monitors both system and application logs looking for problems

Fail2ban can also monitor individual application log files

/etc/fail2ban/jail.conf file contains the Fail2ban configuration

Configuration defines the applications to monitor, where their log files are located, and what actions to take if it detects a problem.

Lesson 4 Review

IDS software monitors the network/applications and dynamically changes firewall rules if it detects attacks

DenyHosts works with TCP Wrappers

Fail2ban works with many applications