

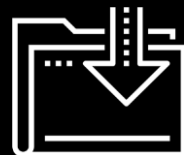


Firewall Policies

A computer lets you make more mistakes faster than any invention in human history—with the possible exceptions of handguns and tequila.

-Unknown

Cybersecurity
Linux 1 Day 3



Today's Objectives

By the end of class, you will be able to:



Develop firewall policies



Implement firewall rules with ufw



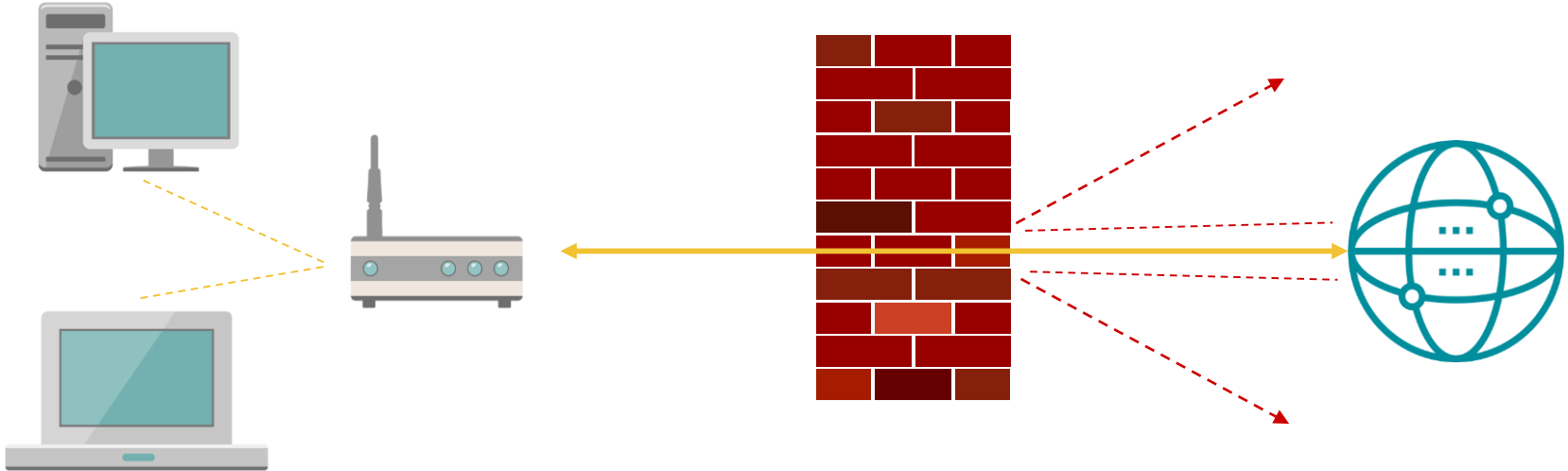
Interpret iptables rules



*Strong security requires
insight into the contents of
packets traveling on the
wire.*

Host Firewall: Filter packets entering / exiting a single *host* on a network.

Network Firewall: Filter packets entering / exit a network *from the public Internet*.



Firewalls

What firewalls can inspect:



The source/destination IP/MAC Addresses: does the packet comes from a device in a trusted subnet or is the response is to a request made from a trusted device



A packet's TCP flags: does the packet contain TCP flag settings.

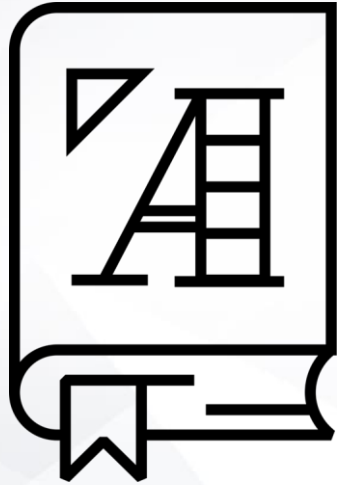


Whether the data in the packet is malformed or corrupted



The rate with which a given host is sending packets

Introducing iptables



Iptables are a standard firewall included in most Linux distributions.

Iptables

Iptables match each packet that crosses the networking interface against a set of rules, then decides an action.

Table 1 (Filter)

Chain 1 (Forward)

Rule 1

Rule 2

Rule 3

Chain 2

Rule 1

Rule 2

Rule 3

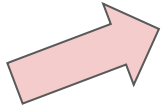
Rules: Conditions under which a server should accept or drop a packet.

Chain: A series of rules that a firewall applies to network packet.

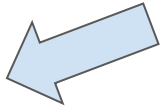
Tables: A collection of chains that serve a related purpose.

Input, Output, Forward

Iptables have three principle chains:



Input chains: determines whether the firewall accepts or drops incoming packets.



Output Chains: Determines whether the firewall accepts or drops outbound connections.



Forward chains: Determines whether the firewall will act as a router and forward the packet to its destination.

INPUT Chain

```
Iptables -A FORWARD -p tcp --syn --dport 80 -  
m state --state NEW -j ACCEPT
```

```
Iptables -A FORWARD -p tcp --syn --dport 443  
-m state --state NEW -j ACCEPT
```

```
Iptables -A FORWARD -p tcp --dport 4321 -i  
eth0 -s $INT_NET -j ACCEPT
```

INPUT Chain

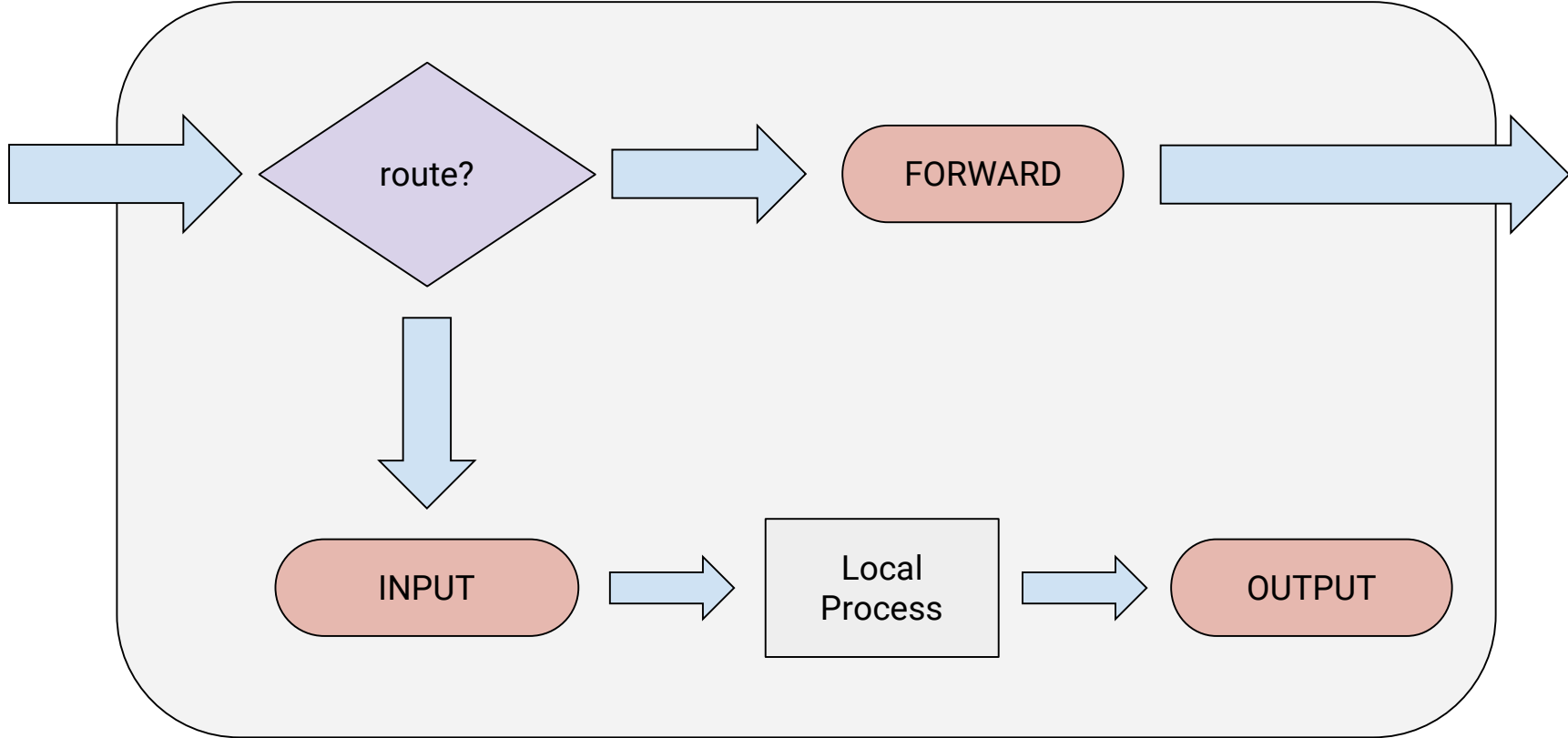
```
Iptables -A OUTPUT -p tcp --dport 21 --syn -m  
state --state NEW -j ACCEPT
```

```
Iptables -A OUTPUT -p tcp --dport 22 --syn -m  
state --state NEW -j ACCEPT
```

```
Iptables -A OUTPUT -p tcp --dport 25 --syn -m  
state --state NEW -j ACCEPT
```

Filter Table (Diagram)

KERNEL





Activity: Interpreting iptables Rules

In this activity, you will partner up to interpret common firewall rulesets, extracted from real configurations.

Activities/03_Par_Interpreting_iptables_Rules/Unsolved/REA
DME

Suggested Time:
15 Minutes





Times Up! Let's Review.

Interpreting iptables Rules

Introducing UFW

Ufw (Uncomplicated FireWall)

We'll use ufw for tasks like:



Deny incoming packets based on certain conditions, such as source IP, malformed TCP flags, etc.



Only allow the machine to send data for specific protocols, such as SSH.



Rate-limit communications from a source IP to a destination port to the host.



ufw is a firewall configuration utility found on many Linux machines.

ufw Demo

What we'll cover in the next demo:

01

Set “default” rules to deny all outgoing / incoming traffic from all ports

02

Set a specific rule to allow HTTP traffic, so we can browse the web.

03

Turn on logging to keep track of all packets that the firewall drops.

04

Verify firewall settings.

ufw commands

enable: Starts the firewall, and causes it to start on system startup.

disable: Stops the firewall.

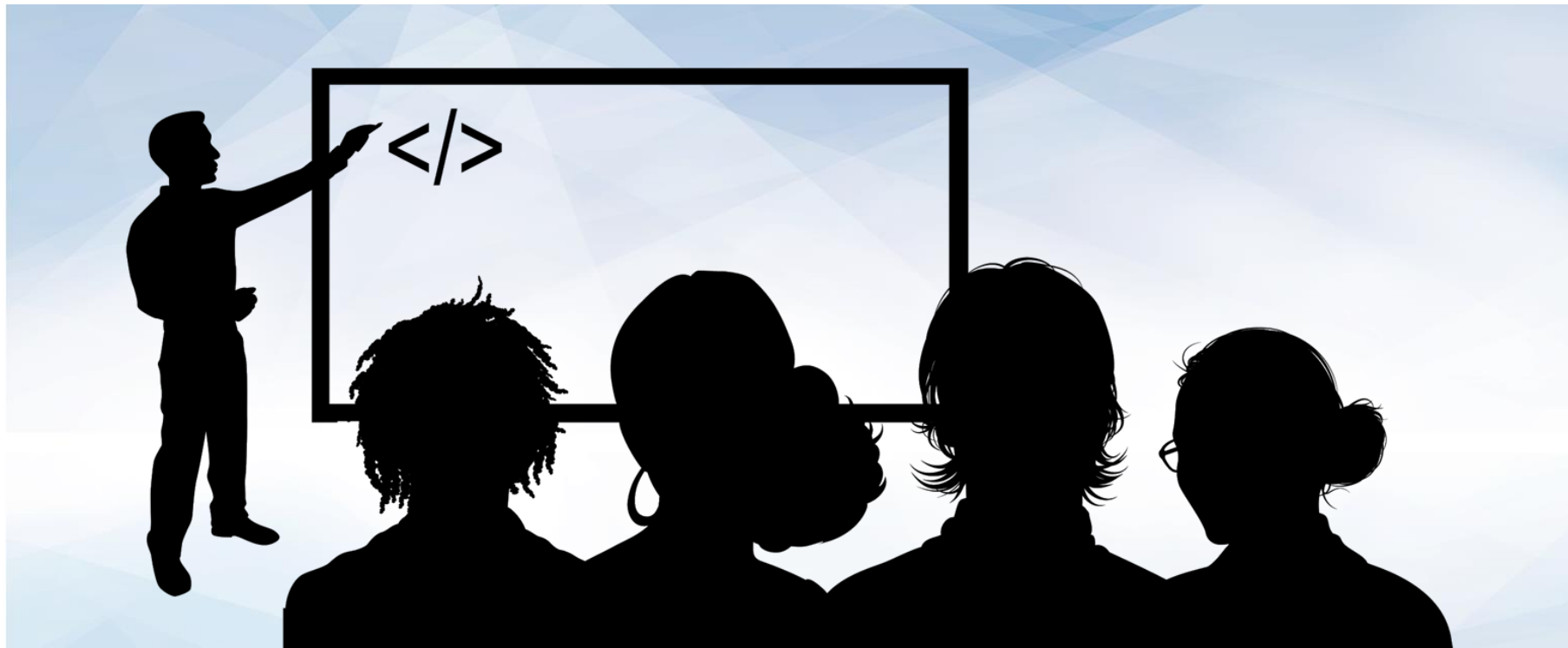
default: Allows administrators to set default rules, which apply to *all* packets except for those that have a specific rule set up for them already.

logging: Tells UFW how much information to log. When logging is enabled, UFW keeps information about every packet that it drops. The available levels are off, low, medium, and high. Increasing log levels causes UFW to save more information about packets that it drops.

allow: Used to set a rule to allow packets into/out of specific ports.

deny: Used to set a rule to allow packets into/out of specific ports. If UFW drops a packet due to a deny rule, it throws it away, *and sends no response to the source machine*.

reject: Reject rules are like deny rules, except they cause UFW to drop the packet *and* send an error message back to the source machine. This is somewhat less secure than a deny rule, because it reveals something about your firewall configuration to the source host.



Instructor Demonstration

Ufw

Firewall Design Principles

By default, well-configured firewalls should drop packets by default, only allowing **access** or **egress** if specified for particular ports and protocols.

1. Traffic should only be allowed to / from ports that are required by services the machine is expected to provide.
1. Document the services that the machine is expected to furnish, and implement rules clearing only the ports required.
1. Traffic should be allowed from as few hosts as possible. Ideally, this means allowing connections only from known IP address.
 - If configuring an intranet, restrict connections to machines on the local subnets.
 - If configuring a public server, you can't restrict connecting IP addresses as effectively, so it's important to harden the host thoroughly.

Firewall Design Principles

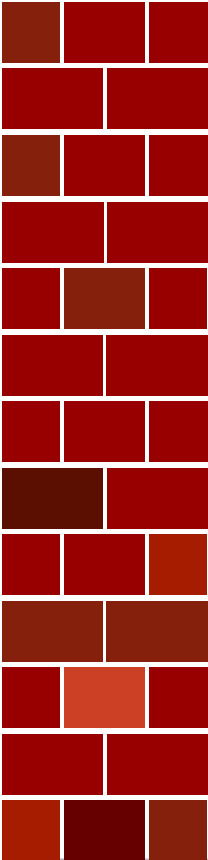
When configuring a new firewall, start from scratch.

A well-configured firewall should **drop** packets by default, only allowing **access** or **egress** if it's been explicitly enabled for a particular port and/or protocol.

Drop Incoming: `ufw default deny incoming`

Drop Outgoing: `ufw default deny outgoing` (careful!)

Firewall Traffic



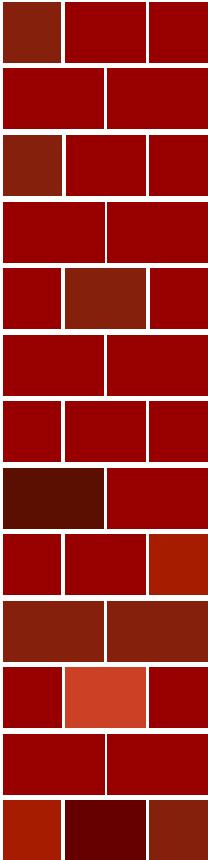
Traffic should be allowed *only* to/from ports that are required by services the machine is expected to provide. This minimizes the host's potential attack surface.

You determine which ports to allow by documenting the services the machine is expected to furnish, and implementing rules clearing only the ports they require.

Traffic should be allowed from as few hosts as possible. Ideally, this means allowing connections only from known IP addresses.

In the event you're configuring an intranet, you'll likely restrict connections to machines on the local subnet(s).

If you're configuring a public server, you can't restrict connecting IP addresses as effectively, so it's particularly important to harden the host as thoroughly as possible.



Developing Policies with UFW



Activity: Setting and Testing Firewall Rules

In this activity, you will implement a firewall policy by setting rules with ufw.

`Activities/01_Stu_Access_Control/Unsolved/README`

Suggested Time:
20 Minutes





Times Up! Let's Review.

Firewall Rules



Activity: Setting and Testing Firewall Rules

In this activity, you will implement a firewall policy by setting rules with ufw.




[Activities/02_Par_Firewall_Policies/Unsolved/README](#)

Suggested Time:
20 Minutes



Today's Objectives

By the end of class, you will be able to:

-  Develop firewall policies
-  Implement firewall rules with ufw
-  Interpret iptables rules