# Linux Week 2: Network Security

Cybersecurity
Linux 2 Day 1

## Today’s Objectives

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

- ☐ Interpret existing IDS rules
- ☐ Write custom Snort rules
- ☐ Generate traffic logs and alerts with Snort

## Activity: Iptables Rules Warm-Up

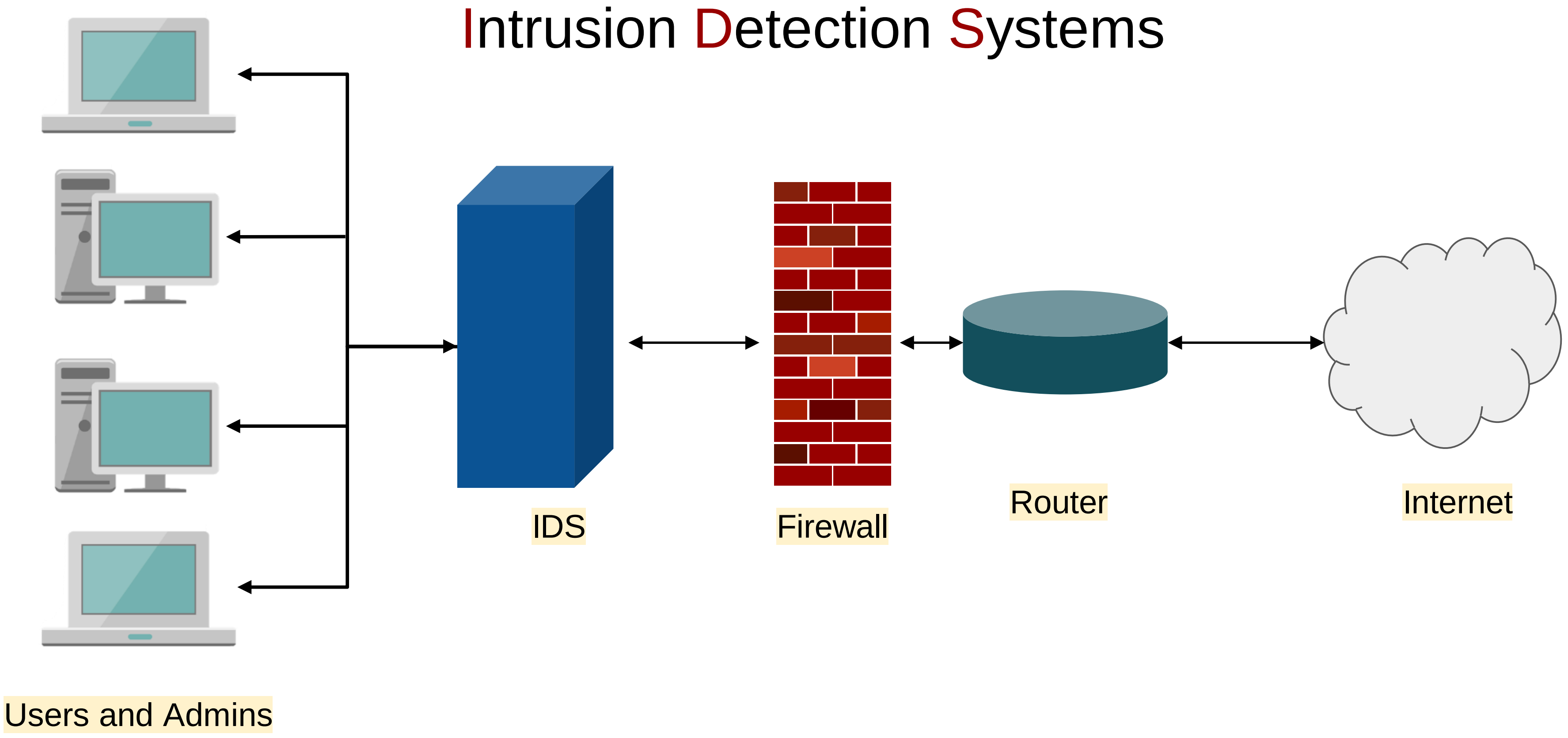
In this activity, you will interpret a handful of iptables rules to prepare for later exercises on Snort attack signatures.

Activities/01_Stu_iptables_Warm_Up/Unsolved/README

**Suggested Time:**
10 Minutes

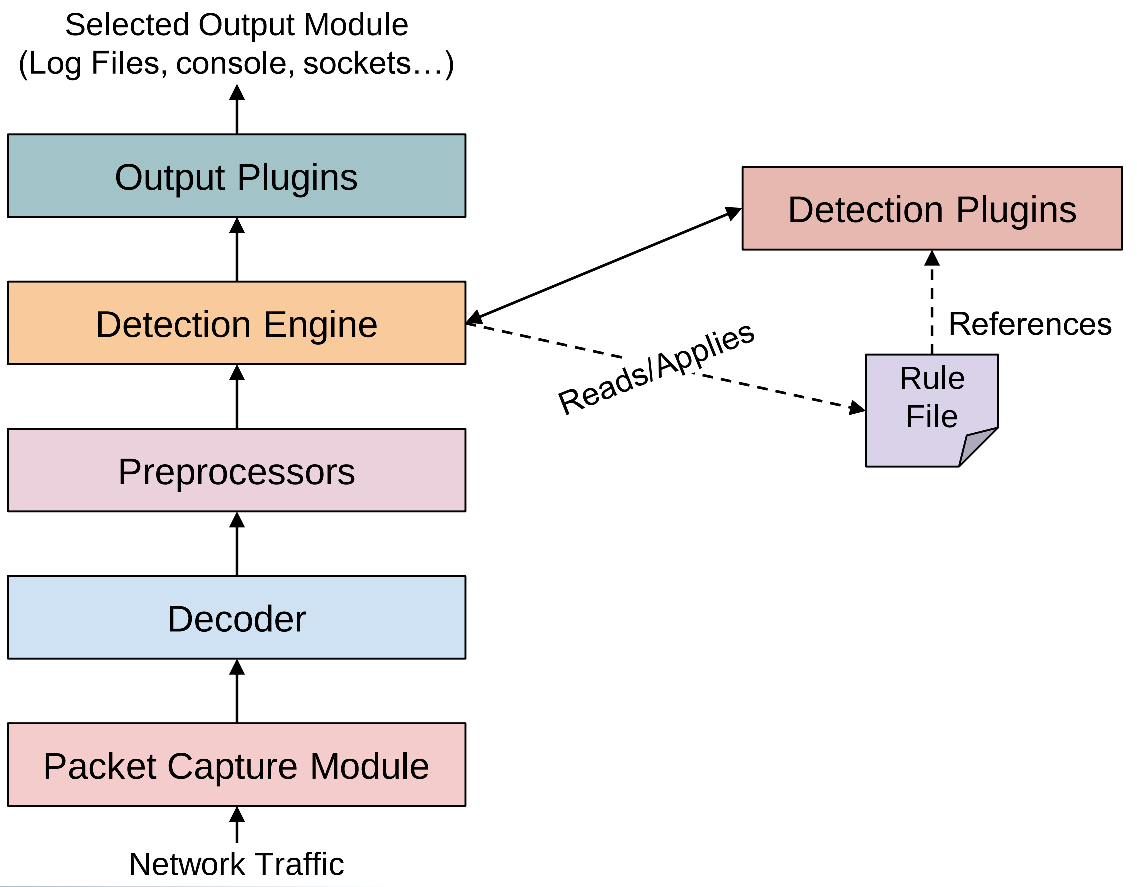
## Intrusion Detection System

# Intrusion Detection Systems



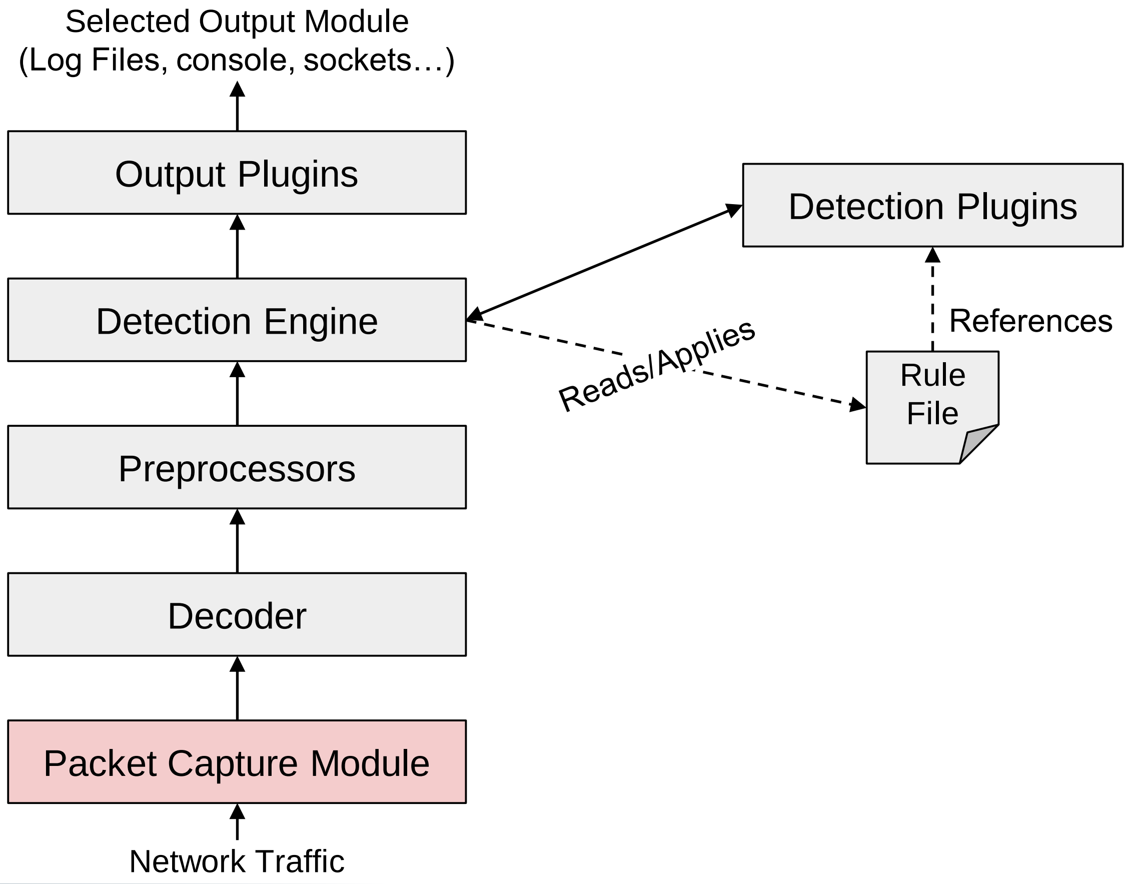
## Introducing Snort

# Introduction to Snort



Snort analyzes traffic through a series of analyzers

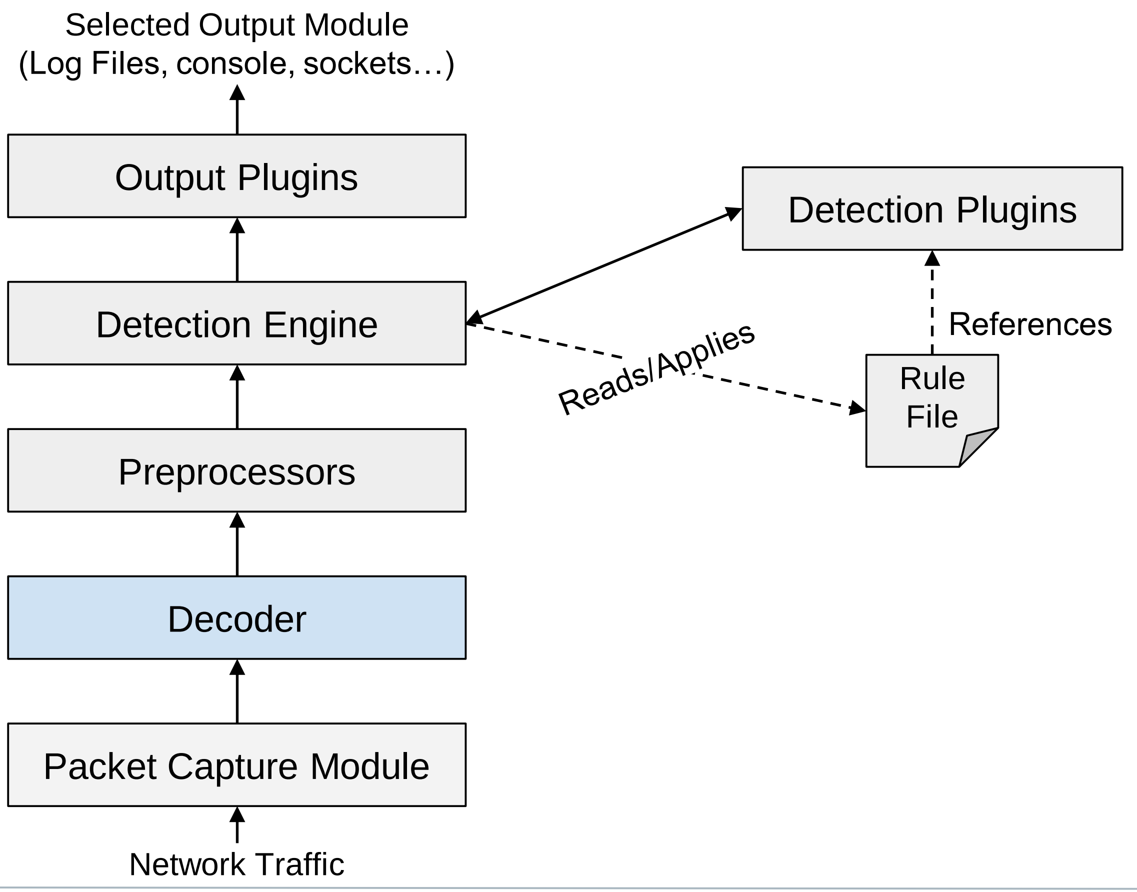
## Packet Capture Module



Captures packets from the NIC.

Based on popular programming library libpcap.

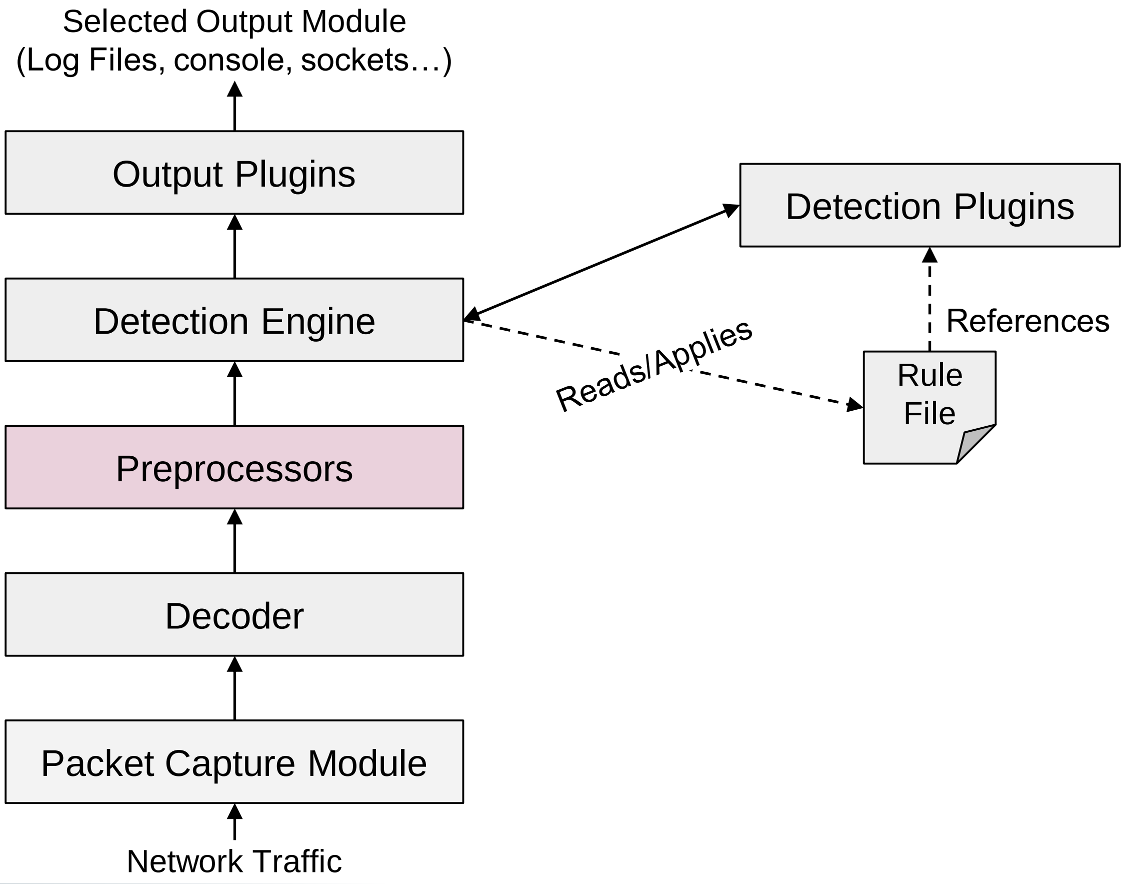
# Decoder



Fits the captured packet into data structures easily understood by Snort.

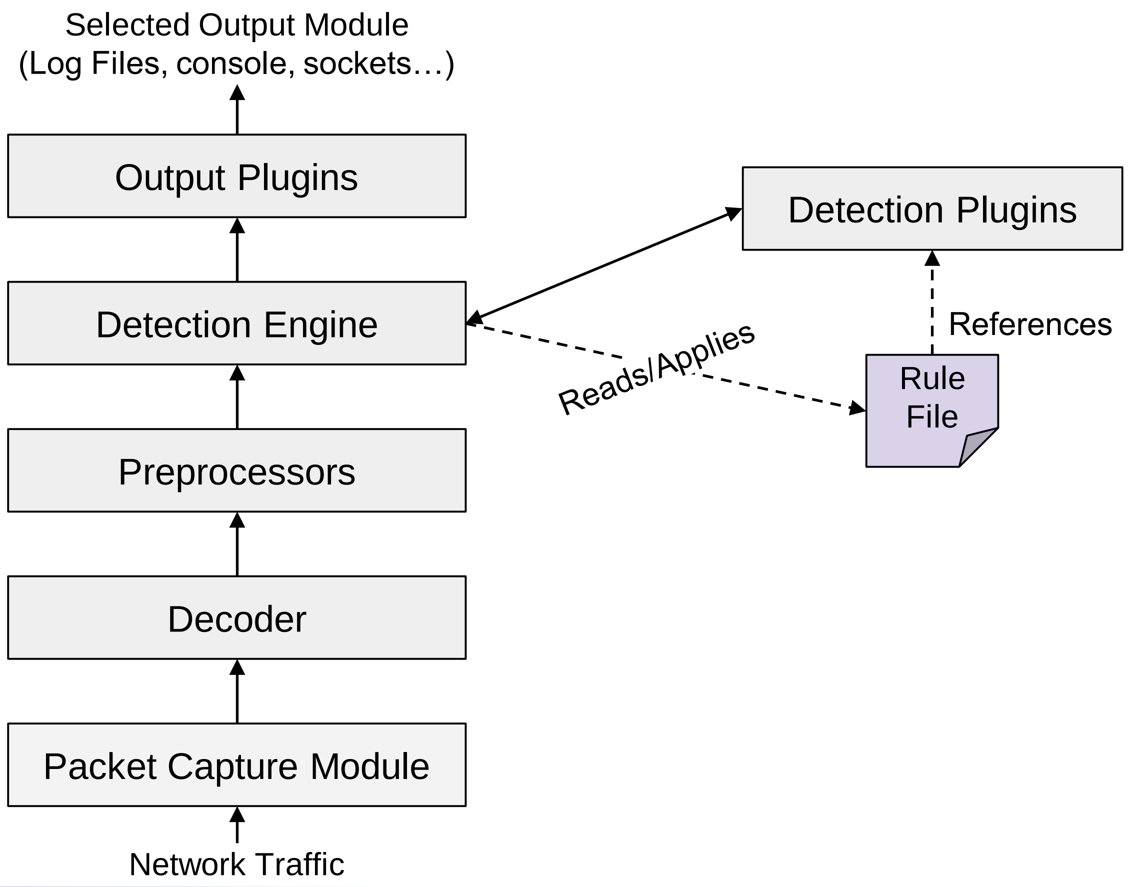
Identifies link-layer protocols.

# Preprocessors



Filters that identify packets that should be flagged for later inspection.

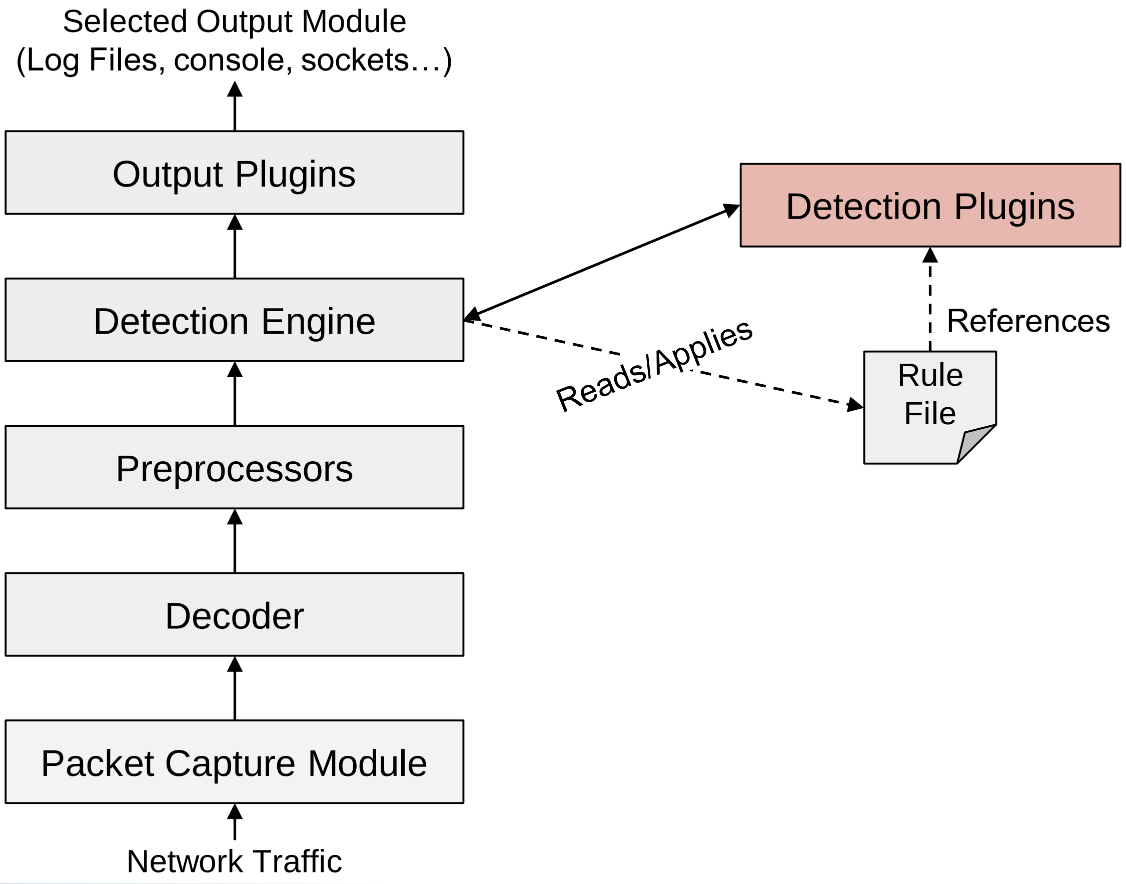
## Rules Files



Plain-text files which contain a list of rules in Snort syntax.

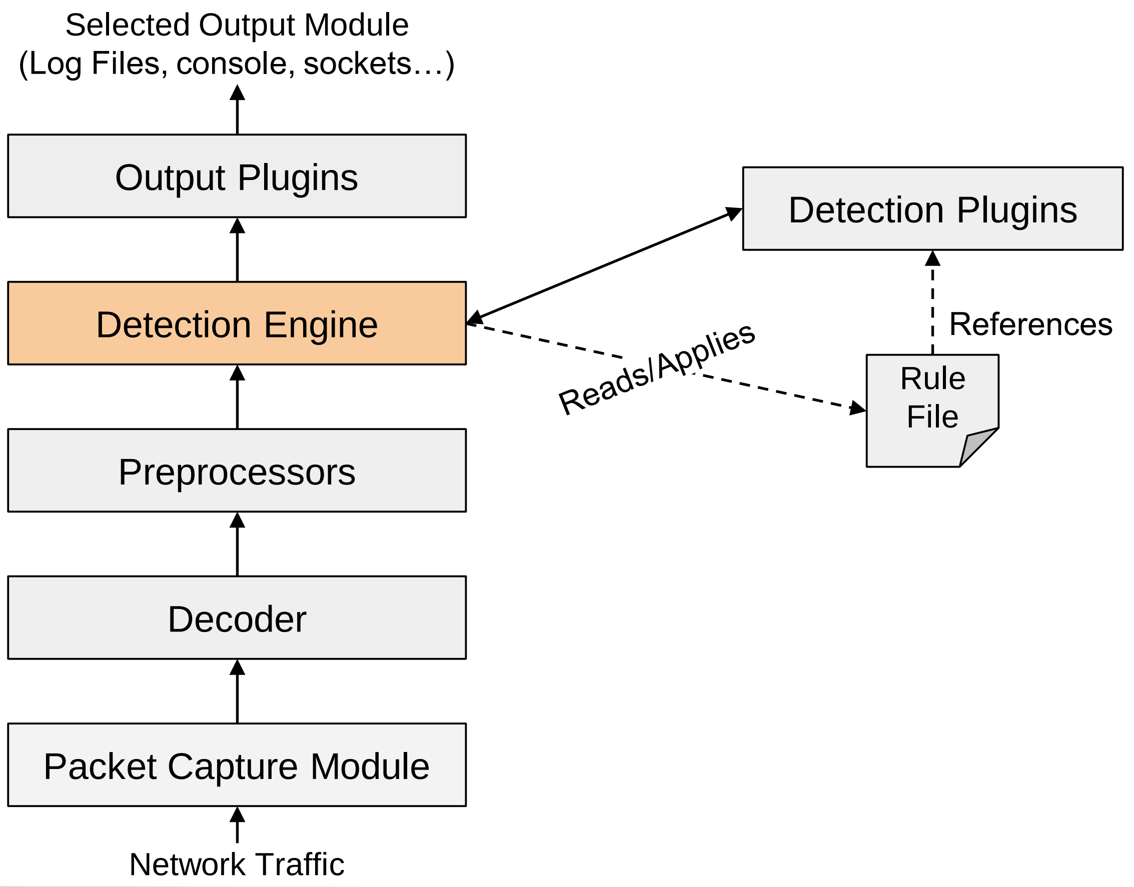
Syntax specifies the protocols, addresses, and byte sequences to monitor traffic.

# Detection Plug-Ins



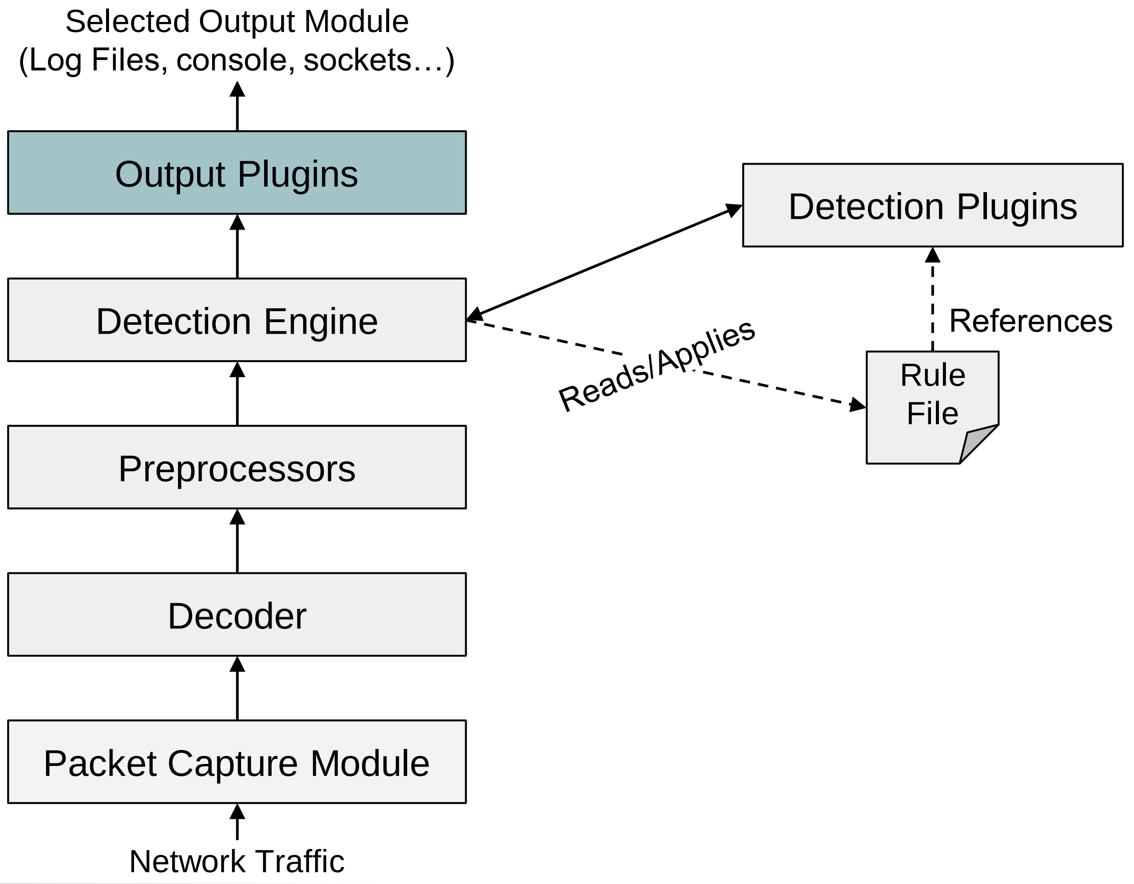
Modules used to efficiently identify patterns whenever a rule is evaluated.

# Detection Engine



Reads the Rules files, then uses Detection Plug-ins to match packets against the rules.

# Output Plugins



Modules which allow custom formatting of notifications, such as alerts and logs.

## The Rules

Snort works by:

01 Reading a configuration file, specifying where to find rules files, preprocessors, etc.

02 Loading these rules and plug-ins.

03 Capturing packets and monitoring traffic for patterns specified in the loaded rules.

04 When traffic matches a rule pattern, Snort generates an alert and /or logs the matching packet for later inspection.

## Example Snort Rules

alert ip any any -> any any {msg “IP Packet Detected”;}

alert: the action taken

## Example Snort Rules

alert ip any any -> any any {msg “IP Packet Detected”;}

alert: the action taken

ip: “Apply this rule to all IP packets...”

## Example Snort Rules

alert ip any any -> any any {msg “IP Packet Detected”;}

alert: the action taken

ip: “Apply this rule to all IP packets...”

any any: “Which comes from any source IP Address and any source port.”

# Example Snort Rules

alert ip any any -> any any {msg “IP Packet Detected”;}

alert: the action taken

ip: “Apply this rule to all IP packets...”

any any: “Which comes from any source IP Address and any source port.”

-> any any: “And is bound for any IP address and any destination port.”

## Example Snort Rules

```
alert ip any any -> any any {msg “IP Packet Detected”;}
```

- alert: the action taken
- ip: “Apply this rule to all IP packets...”
- any any: “Which comes from any source IP Address and any source port.”
- -> any any: “And is bound for any IP address and any destination port.”
- {msg “IP Packet Detected”;}: the message to print with the alert

## Activity: Exploring Snort

In this exercise, you'll explore a Snort installation.

Activities/02_Stu_Exploring_Snort/Unsolved/README

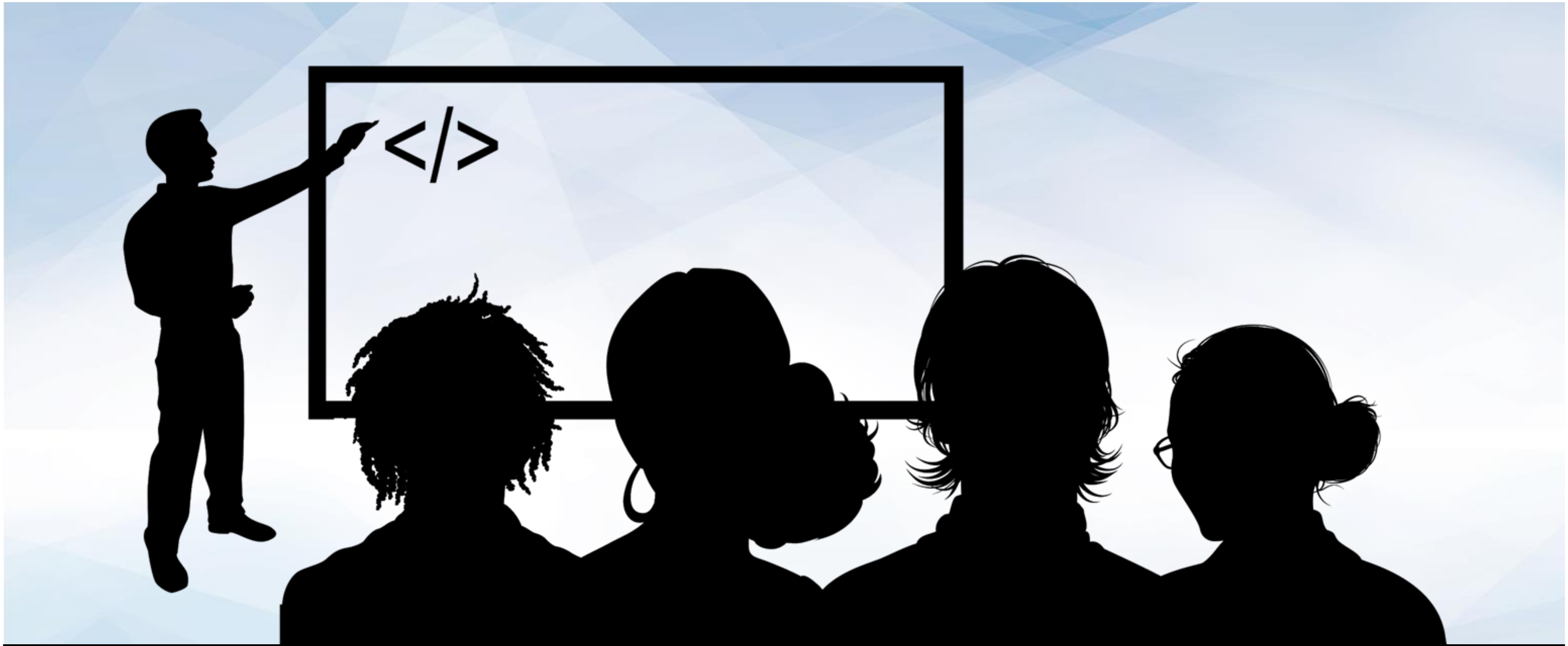
**Suggested Time:**
20 Minutes

# Times Up! Let’s Review.

Exploring Snort

You will need to know how to run Snort from the command line for two reasons:

- ☐ Using Snort to identify suspicious activity in packet captures
- ☐ Quickly testing different configurations



## Instructor Demonstration

Running Snort

## Activity: Running Snort

In this activity, you will practice running Snort from the command line.

Activities/03_Stu_Running_Snort/Unsolved/README

**Suggested Time:**
20 Minutes

## Times Up! Let’s Review.

Running Snort

## Take a Break!

## Snort Rules in Depth

Snort rules can appear pretty complicated:

```
alert icmp $EXTERNAL_NET any -> $HOME_NET any (msg:"PROTOCOL-ICMP PING Unix"; itype:8; content:"|10 11 12 13 14 15 16 17 18 19 1A 1B 1C 1D 1E 1F|"; depth:32; metadata:ruleset community; classtype:misc-activity; sid:366; rev:11;)
```

## Snort Rules in Depth

But they can be more easily understood when broken down into sections :

```
alert icmp $EXTERNAL_NET any -> $HOME_NET any (
msg:"PROTOCOL-ICMP PING Unix";
itype:8;
content:"|10 11 12 13 14 15 16 17 18 19 1A 1B 1C 1D 1E 1F|";
depth:32;
metadata:ruleset community;
classtype:misc-activity;
sid:366;
rev:11;)
```

## Snort Rules in Depth

But they can be more easily understood when simplified :

alert icmp $EXTERNAL_NET any -> $HOME_NET any (

- Action This rule tells Snort to alert and log packets that trigger this rule.

- Protocol: icmp tells Snort to apply this rule only to ICMP traffic.

- Source/Destination Addresses: means any packet from outside the local subnet into the local network that matches the rule will fire an alert.

- Rule Options: The complicated contents at the end of the rule are the options. They modify what the rule looks for, how to print its output, and other properties.

## Snort Rules in Depth

Rules options are the most complicated aspect, so we’ll look into each of them.

```
alert icmp $EXTERNAL_NET any -> $HOME_NET any (
msg:"PROTOCOL-ICMP PING Unix";
itype:8;
content:"|10 11 12 13 14 15 16 17 18 19 1A 1B 1C 1D 1E 1F|";
depth:32;
metadata:ruleset community;
classtype:misc-activity;
sid:366;
rev:11;)
```

## msg

The string to log when this rule is triggered.

```
alert icmp $EXTERNAL_NET any -> $HOME_NET any (
msg:"PROTOCOL-ICMP PING Unix";
itype:8;
content:"|10 11 12 13 14 15 16 17 18 19 1A 1B 1C 1D 1E 1F|";
depth:32;
metadata:ruleset community;
classtype:misc-activity;
sid:366;
rev:11;)
```

## itype

Check for a specific type of ICMP packet. (Type 8 is an ECHO packet.)

```
alert icmp $EXTERNAL_NET any -> $HOME_NET any (
msg:"PROTOCOL-ICMP PING Unix";
itype:8;
content:"|10 11 12 13 14 15 16 17 18 19 1A 1B 1C 1D 1E 1F|";
depth:32;
metadata:ruleset community;
classtype:misc-activity;
sid:366;
rev:11;)
```

## content

Data to look for *inside* the packet. (This sequence is 1-10 in hexadecimal)

```
alert icmp $EXTERNAL_NET any -> $HOME_NET any (
msg:"PROTOCOL-ICMP PING Unix";
itype:8;
content:"|10 11 12 13 14 15 16 17 18 19 1A 1B 1C 1D 1E 1F|";
depth:32;
metadata:ruleset community;
classtype:misc-activity;
sid:366;
rev:11;)
```

## depth

Specifies how many bytes into the packet Snort should look for content.

```
alert icmp $EXTERNAL_NET any -> $HOME_NET any (
msg:"PROTOCOL-ICMP PING Unix";
itype:8;
content:"|10 11 12 13 14 15 16 17 18 19 1A 1B 1C 1D 1E 1F|";
depth:32;
metadata:ruleset community;
classtype:misc-activity;
sid:366;
rev:11;)
```

## metadata

Contains administrative information about the rule. In this case, the metadata tells us this is a community rule.

```
alert icmp $EXTERNAL_NET any -> $HOME_NET any (
msg:"PROTOCOL-ICMP PING Unix";
itype:8;
content:"|10 11 12 13 14 15 16 17 18 19 1A 1B 1C 1D 1E 1F|";
depth:32;
metadata:ruleset community;
classtype:misc-activity;
sid:366;
rev:11;)
```

## classtype

Specifies what kind of network activity this is.

```
alert icmp $EXTERNAL_NET any -> $HOME_NET any (
msg:"PROTOCOL-ICMP PING Unix";
itype:8;
content:"|10 11 12 13 14 15 16 17 18 19 1A 1B 1C 1D 1E 1F|";
depth:32;
metadata:ruleset community;
classtype:misc-activity;
sid:366;
rev:11;)
```

## sid

The ID of the rule

```
alert icmp $EXTERNAL_NET any -> $HOME_NET any (
msg:"PROTOCOL-ICMP PING Unix";
itype:8;
content:"|10 11 12 13 14 15 16 17 18 19 1A 1B 1C 1D 1E 1F|";
depth:32;
metadata:ruleset community;
classtype:misc-activity;
sid:366;
rev:11;)
```

## rev

The revision ID. This is essentially a version of the rule

```
alert icmp $EXTERNAL_NET any -> $HOME_NET any (
msg:"PROTOCOL-ICMP PING Unix";
itype:8;
content:"|10 11 12 13 14 15 16 17 18 19 1A 1B 1C 1D 1E 1F|";
depth:32;
metadata:ruleset community;
classtype:misc-activity;
sid:366;
rev:11;)
```

## Activity: Interpreting Snort Rules

In this activity, you will interpret rules that fired in the previous exercise and interpret new rules.

**Activities/04_Stu_Interpreting_Snort_Rules/Unsolved/README**

**Suggested Time:**
15 Minutes

## Times Up! Let’s Review.

Interpreting Snort Rules

## Activity: Creating Snort Rules

In this activity, you will write Snort rules and test them against live traffic.

Activities/05_Stu_Creating_Rules/Unsolved/README

**Suggested Time:**
15 Minutes

# Times Up! Let’s Review.

Running Snort

## Today’s Objectives

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

- 🟩 Interpreting existing IDS rules
- 🟩 Write custom Snort rules
- 🟩 Generate traffic logs and alerts with Snort