



# Intrusion Detection Systems




Cybersecurity  
Networks Security Day 2



# Today's Objectives

---

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

-  Install Snort from source
-  Configure a new Snort installation
-  Validate Snort rules by sending suspicious signatures from a foreign host

# Let's Review Firewalls

---

What are firewalls?

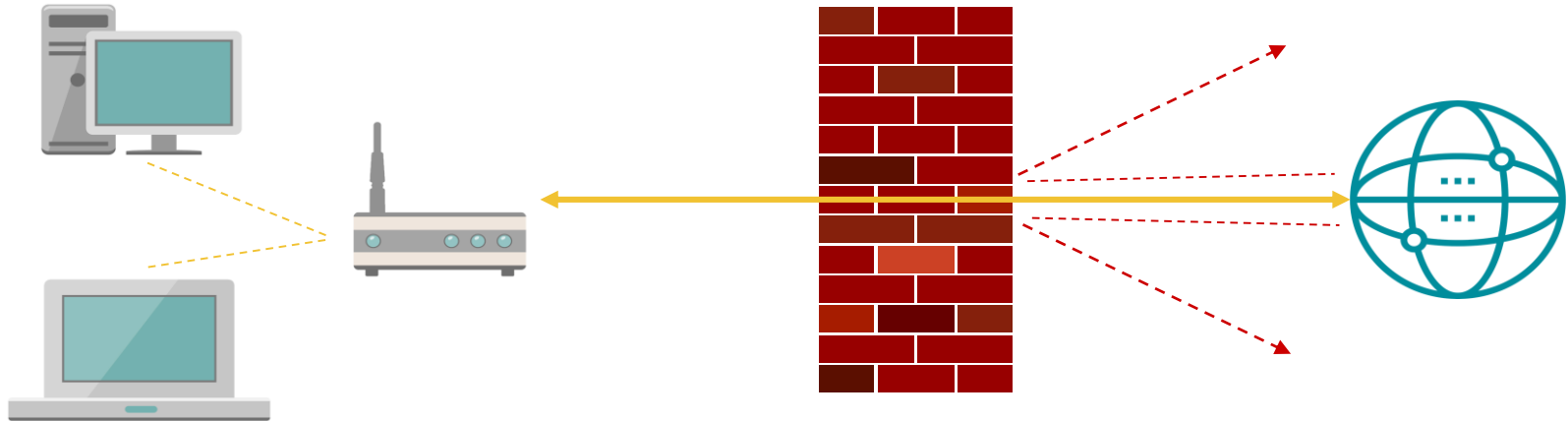
---

# Let's Review Firewalls

---

What are firewalls?

- Firewalls protect networks by deciding which packets are allowed in and out, therefore controlling who is allowed to communicate with a given host or network.



# Let's Review Firewalls

---

**True or False:** Traditional Firewalls are designed to understand the data inside the packets?

---

# Let's Review Firewalls

---

## False for Traditional Firewalls

- Firewalls can only filter traffic based on trusted or untrusted hosts or applications, *not* based on safe or malicious traffic content.
  - Therefore, a clever attacker can still send malicious data through a firewall by hijacking or impersonating trusted traffic.
-

# Intrusion Detection System

# Intrusion Detection Systems

---

IDS are tools that can both read incoming traffic *and* look for malicious signatures.

- An IDS is like a firewall that also reads the data in the packet that it inspects.
  - Static IDS look for malicious signatures, which are like fingerprints of malicious traffic.
  - Administrators configure IDS with rules telling it which signatures to block.
  - If incoming traffic triggers a rule, the IDS will fire an alert notifying the administrators of specific traffic.
-



# IDS in a Professional Context

---

Every secure network has an IDS. Familiarity with them is very valuable on the job:

**Systems/Network Administrators** commonly install and configure Snort (a popular IDS that we will zone in on soon) on Linux hosts.

- Configuring Snort requires specifying which IP addresses and subnets to protect and which traffic to inspect and what to look for.

**SOC Analyst** use traffic logs and alerts to generated by IDS in order to identify and understand attacks against their networks.

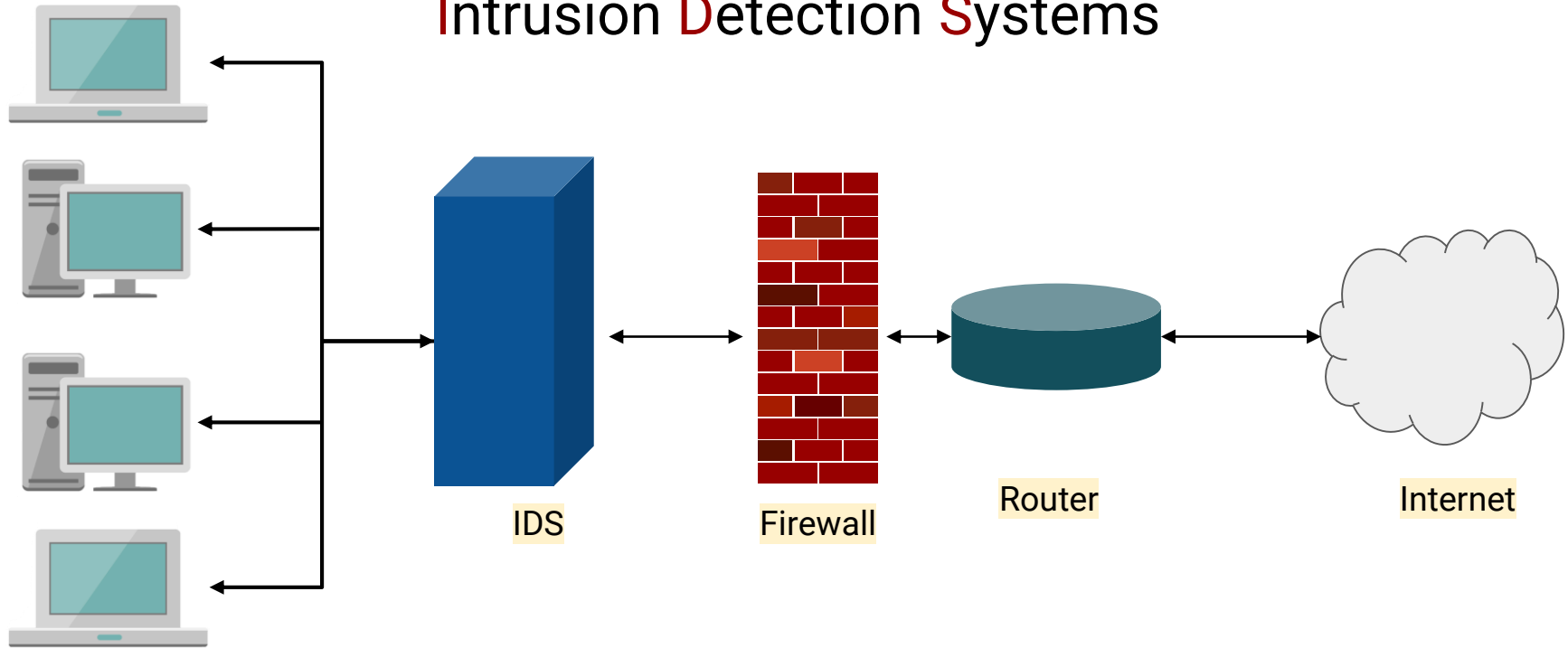
- Understanding how Snort applied rules to incoming traffic and generates alerts is required knowledge for an SOC role.

**Forensics** specialists often use intrusion detection logs to collect evidence such as the traffic an attacker was sending; where they sent it from; and when they sent it.

---

# Intrusion Detection Systems-Traditional Model

## Intrusion Detection Systems



Users and Admins

# Intrusion Detection Systems

---

IDS detect **attack signatures** by reading logs or analyzing traffic in real time and notifying administrator when they detect suspicious traffic.

- Attack signatures are patterns in network traffic that act as “fingerprints” of malicious activity.
- IDS compare all traffic that they monitor to these fingerprints to identify suspicious behavior.
- Rules are implemented to specify what actions should be taken when it recognizes attack signature. For IDS only notification can occur, IDS can not block traffic.

```
stu@ubuntu:~/sneeze$ sudo perl sneeze.pl -d 192.168.132.143 -f /etc/snort/rules/  
local.rules  
ATTACK:  
ATTACK TYPE: icmp-event  
icmp ubuntu:63073 -> 192.168.132.143:21773  
  
ATTACK:  
ATTACK TYPE: icmp-event  
tcp 192.168.132.133:17081 -> 192.168.132.143:21  
  
ATTACK:  
ATTACK TYPE: icmp-event  
tcp ubuntu:21 -> 192.168.132.143:28323  
  
ATTACK:  
ATTACK TYPE: icmp-event  
tcp ubuntu:60856 -> 192.168.132.143:54595
```

# Firewalls vs IDS

---

IDS generate more informative logs than firewalls do.

**For example, Snort (an IDS) can log all of the following information:**

- Alerts: Snort will save every alert it fires.
- PCAPs: Snort can save PCAPs of all traffic that it intercepts.
- Dropped Packets: Like a firewall, Snort can keep logs of any packet that it dropped from a fired rule.

An IDS can be placed in the same location as firewalls: on individual hosts or in front of the network as a whole. Never in front of a Firewall!

- An IDS installed on a single host is called a **Host Intrusion Detection System (HIDS)**
  - An IDS installed in front of a network is called a **Network Intrusion Detection System (NIDS)**
-

# Introducing Snort

# Introducing Snort

---

Snort is a popular, open-source IDS.

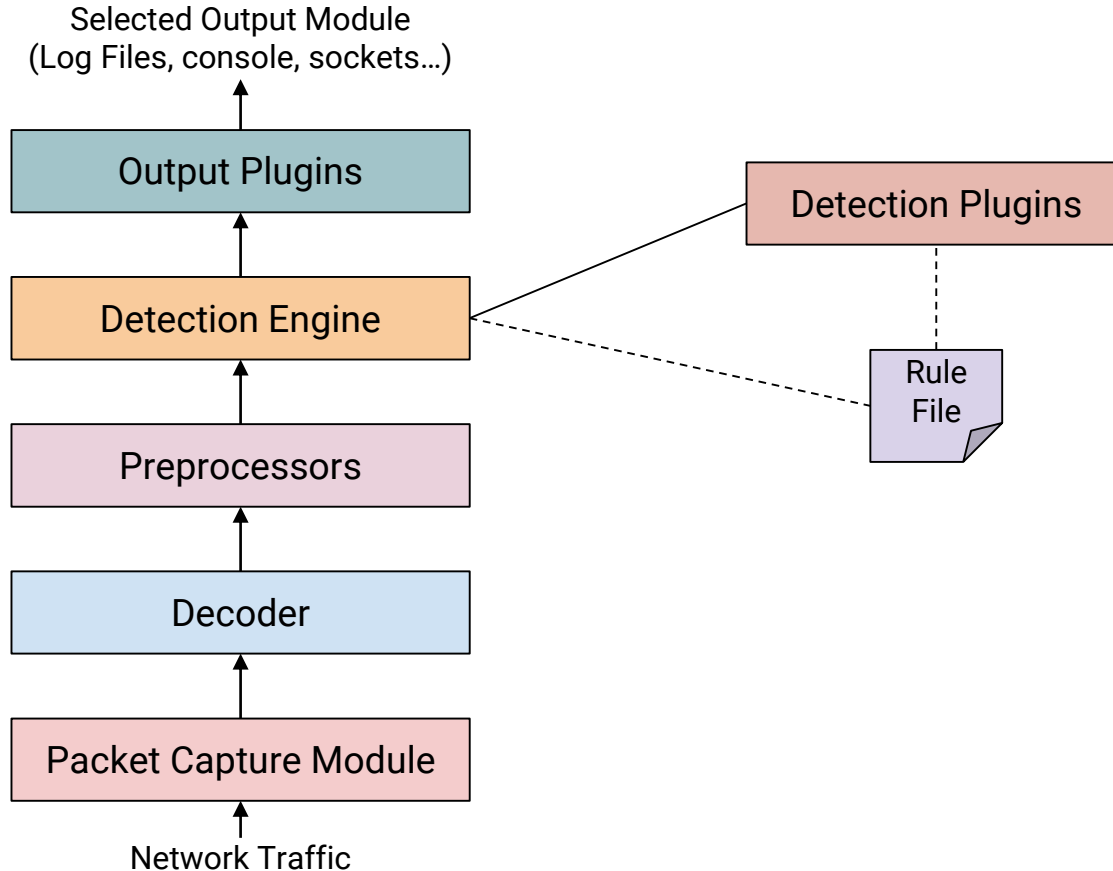
Snort works by capturing packets from the NIC, just like tcpdump.

- It then scans those packets for attack signatures in real-time.
- Snort generates logs and an alert whenever it identifies traffic matching the signature.

Snort analyzes traffic through a series of analyzers, which we will look at next.

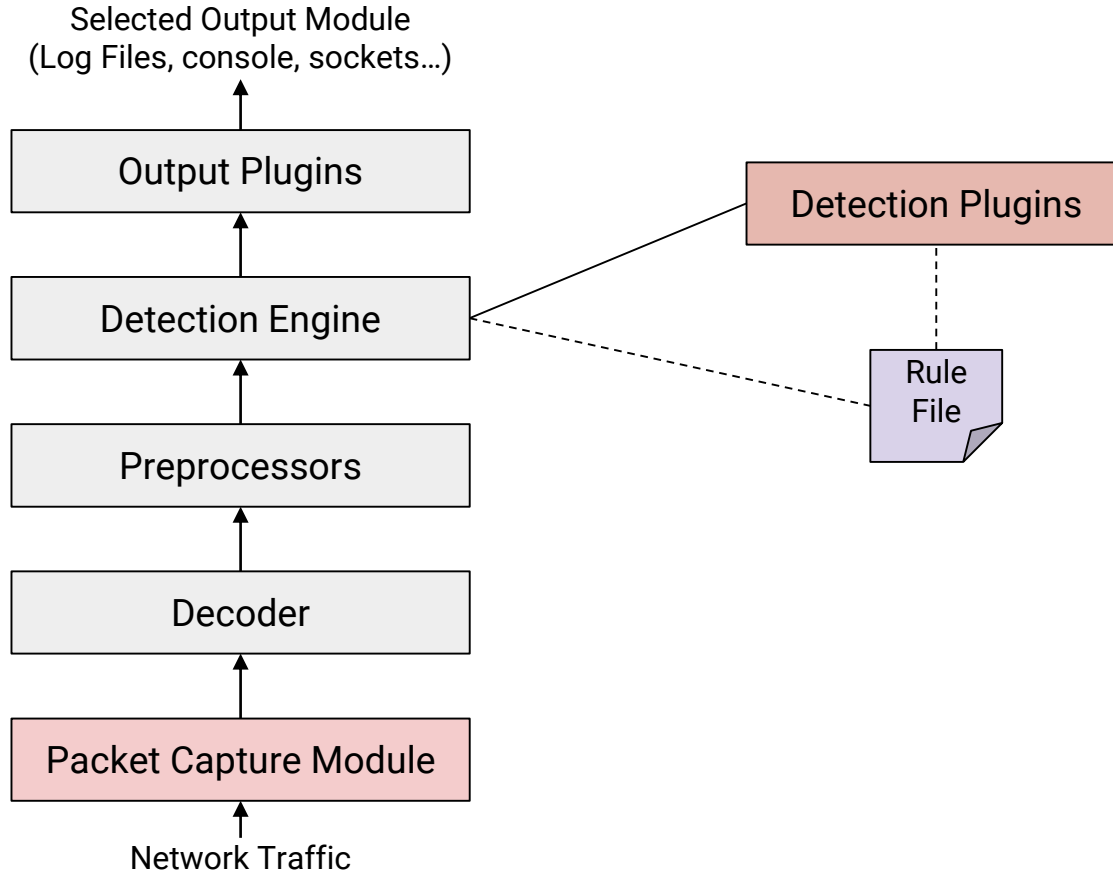
---

# Introduction to Snort



Snort pipes traffic through a series of analyzers.

# Packet Capture Module

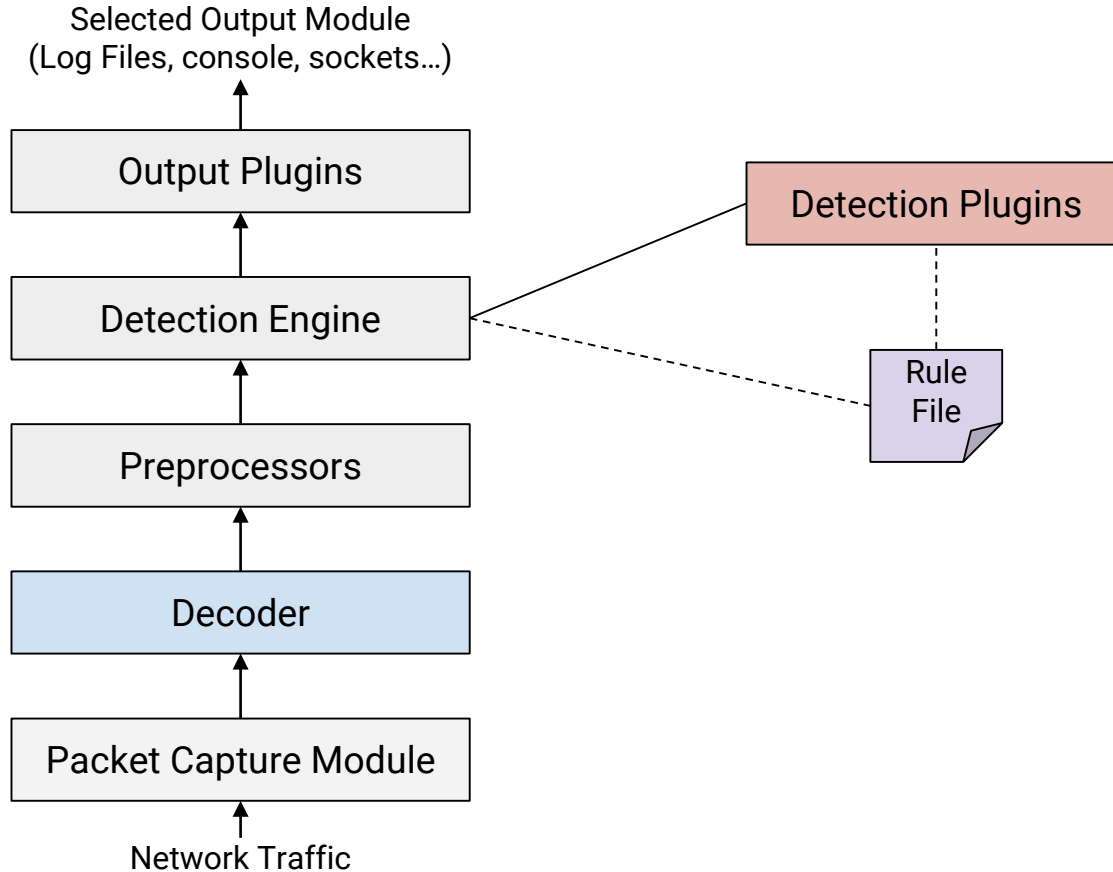


**Packet capture module** captures packets from the NIC.

Based on popular programming library `libpcap`.

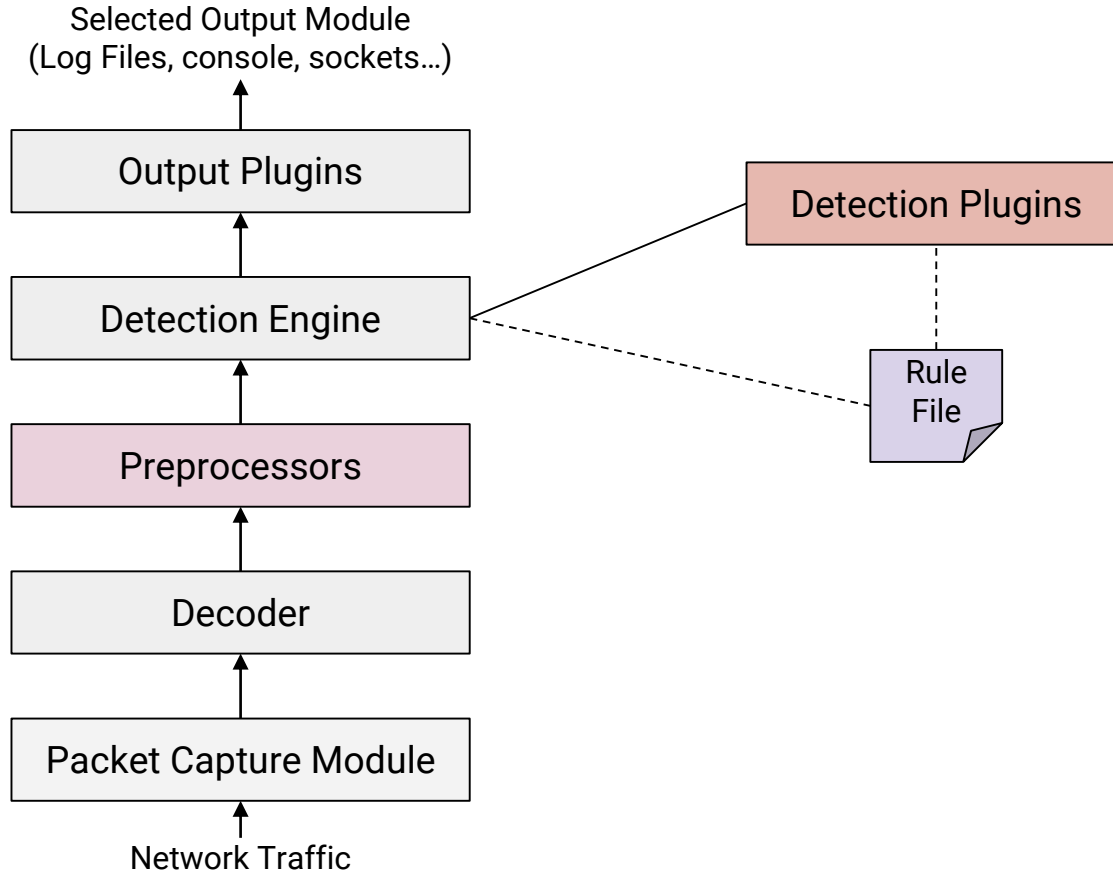


# Decoder



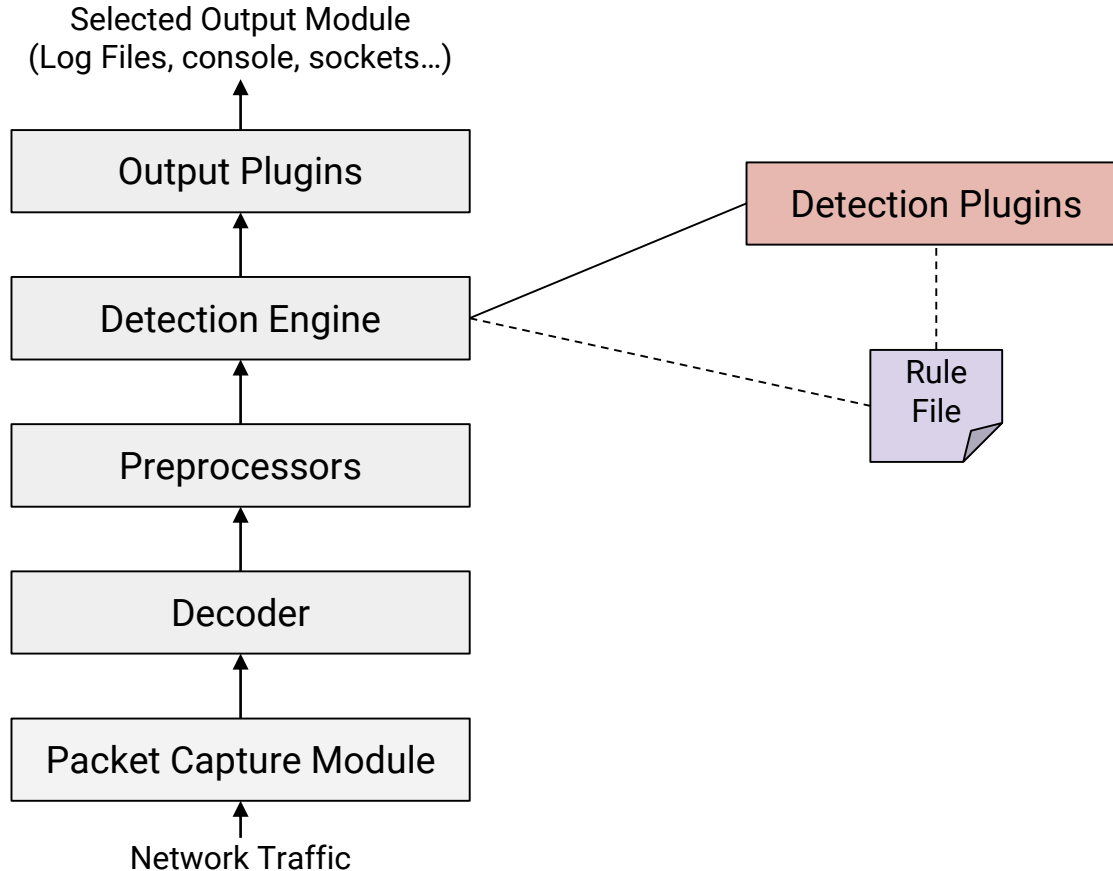
**Decoder** fits the captured packet into data structures easily understood by Snort.

# Preprocessors



**Preprocessors** are filters that identify packets that should be flagged for later inspection.

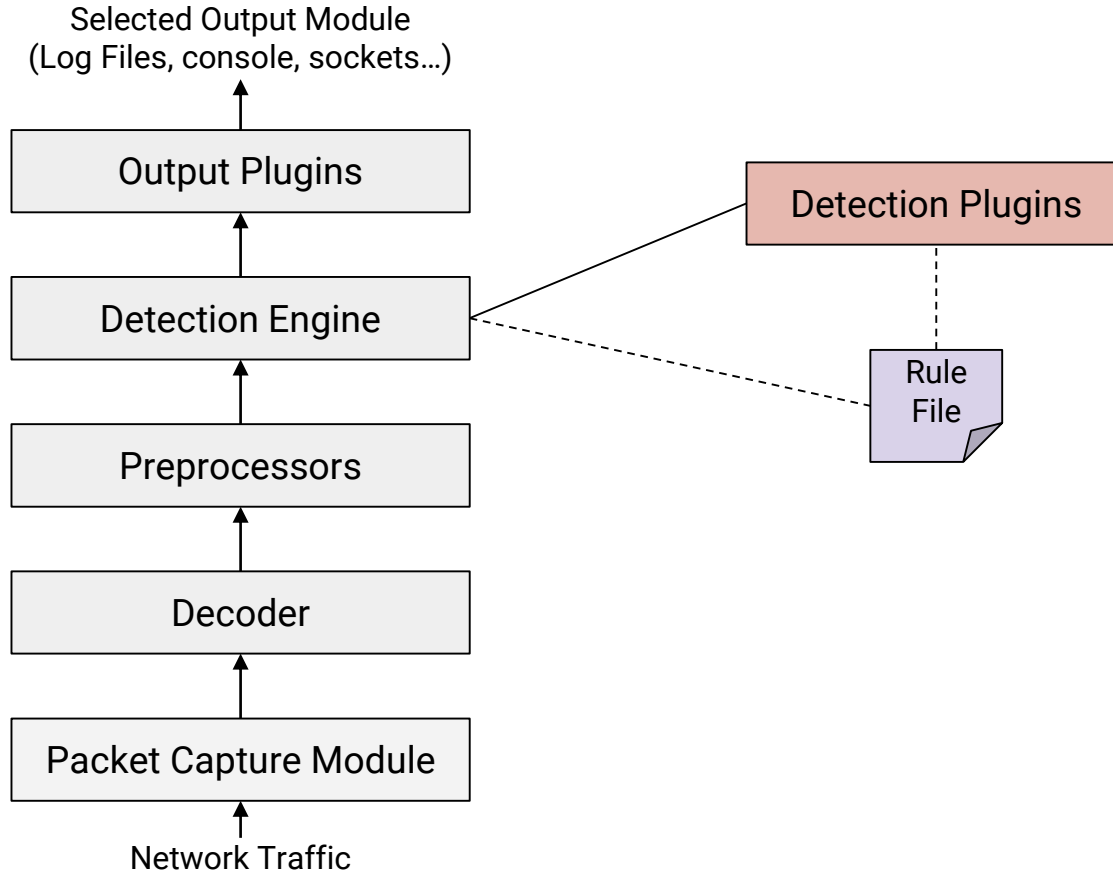
# Rules Files



**Rule files** are plain-text files which contain a list of rules in Snort syntax.

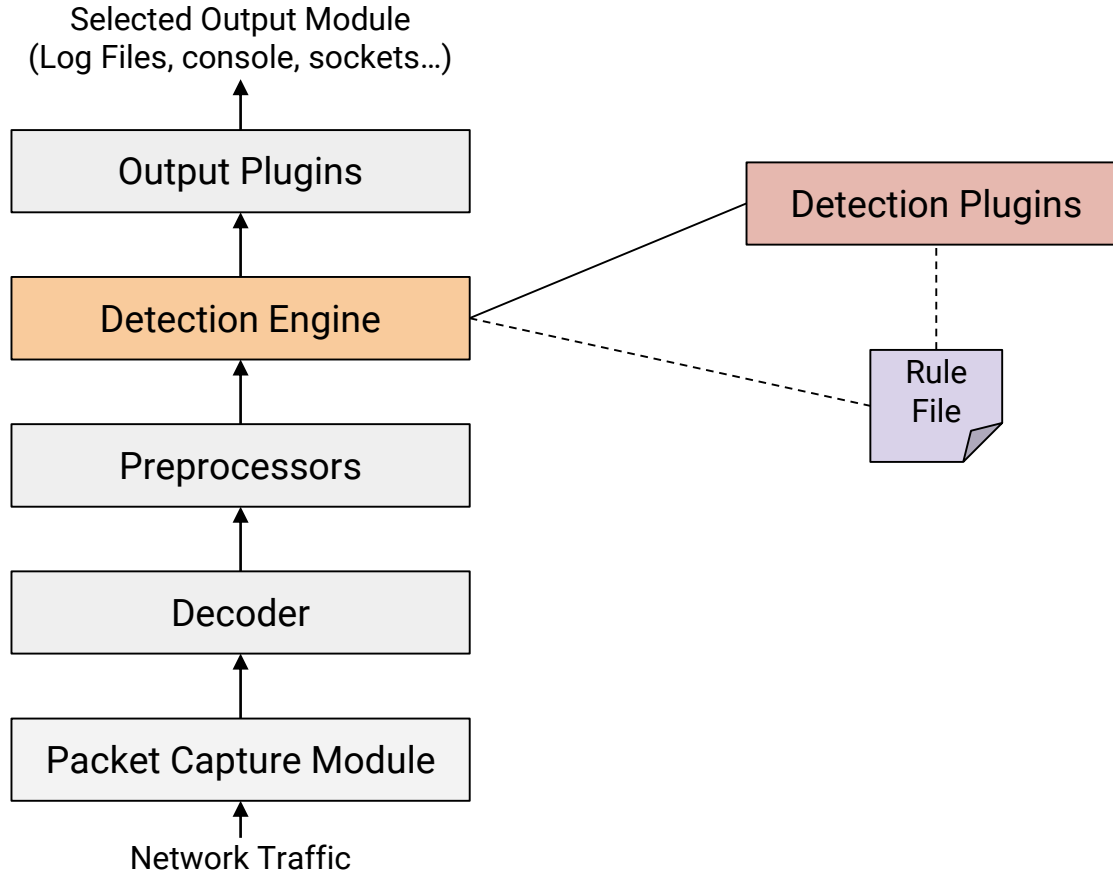
Syntax specifies the protocols, addresses, and byte sequences to monitor traffic.

# Detection Plug-Ins



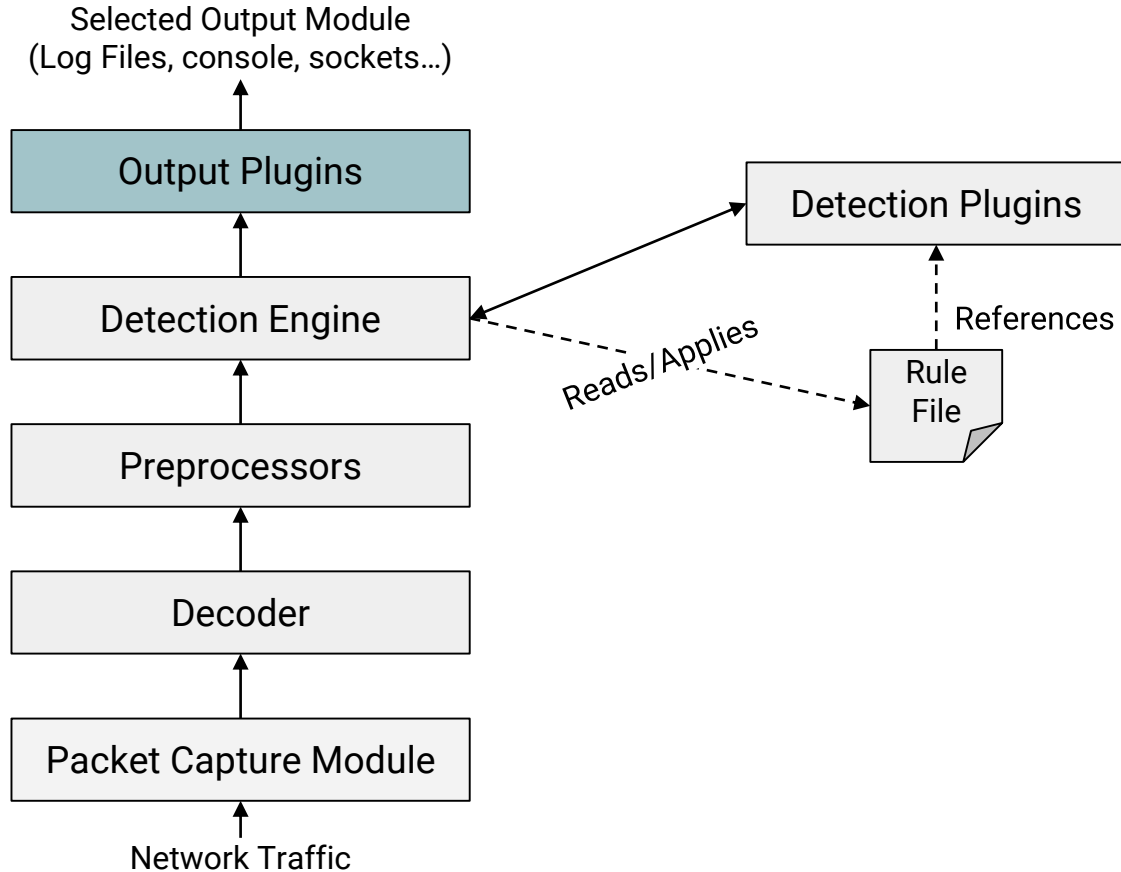
**Detection Plugins** are modules used to efficiently identify patterns whenever a rule is evaluated.

# Detection Engine



**Detection Engines** reads the Rules files, then use Detection Plugins to match packets against the rules.

# Output Plugins



**Output Plugins** are 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";}
```



# 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

---

# Installing and Configuring Snort

# Installing and Configuring Snort

---

In the next exercise, you will build and install Snort from Source and then configure it.

Building “from source” means not using apt to perform the install. Instead, you will need to download and compile the source code and manually move Snort’s configuration files to the right places.

High level overview of steps:

- **Compile:** Compiling is the process of turning source code stored in text files into a program you can run into a command. You will use a tool called [make](#) to compile Snort’s source code.
  - **Install:** After building Snort, you will move its file into the right places to install it.
  - **Set Up Snort Files:** You will then configure Snort with the necessary rules.
-

# Configuration Setup

---

After installation, administration still needs to configure Snort.

- 01 Set **Network Variables**
  - 02 Configure the **Decoder**
  - 03 Configure the **Base Detection Engine**
  - 04 Configure **Dynamically Loaded Libraries**
  - 05 Configure **Preprocessors**
  - 06 Configure **Output Plugins**
  - 07 Customize **Rulesets**
  - 08 Customize Your **Shared Object Snort Rules**
-



# Running Snort

---

In the next exercise, we'll learn more about commands. Here are some basics:



**snort** launches Snort.



**-A console** prints alerts to the terminal.



**-q** causes Snort to print only packet logs.



**-c /etc/snort/snort.conf** tells Snort to use the configuration in /etc/snort/snort.conf.



**-i ens32** tells Snort to monitor packets on the interface ens23.



## **Activity:** Installing and Configuring Snort

In this activity, install and configure Snort from source.

Lab link sent via Slack.

**Suggested Time:**  
60 Minutes



# Take a Break!

---





# Viewing Snort Logs and Writing Snort Rules

# Snort Rules

---

When a rule is triggered, Snort takes the following actions:



**Fires an Alert.** The alert can either output to a log file, or sent directly to an administrator's email, etc.



**Logs the Packet:** Snort will save details about the packet and what triggered the rule to a log file.



**Save the Packet:** Snort can be configured to capture all the traffic that passes through it.



**Drop the Packet:** In addition to firing alerts, Snort can act as a firewall and drop suspicious traffic. Making it an intrusion prevention system, as it detects and prevents activity.

---

# Reading Snort Logs

---

Reading log files would be a large amount of work for any analyst to complete by hand.

Instead, they use **SIEMs** to view and analyze them. SIEMs contain the following capabilities:

- Ability to collect logs from multiple servers
  - Monitor logs in real-time
  - Complex data analysis
-



## **Activity:** Monitoring Snort Logs and Firing Rules

In this activity, you will use Snorby to make Snort fire rules and monitor with rules were triggered.

**Do not follow directions in Cybrscore lab. Specific Instructions will be sent by Instructor.**




**Suggested Time:**  
30 Minutes



# Monitoring Snort Logs and Firing Rules

---

## Some Takeaways

-  Snort will detect Nmap's port scans, and classifies it as **Medium Severity** traffic because port scans are suggestive of reconnaissance and future malicious activity.
  -  ICMP traffic is flagged as **Low Severity** because attackers can discover machines that reply to ping requests by running ping scans.
  -  The SMB brute-force attack is flagged as **High Severity**. Unlike port or ping scans, this is an intentional *attack*, which may result in compromise of the server if successful.
-



# Today's Objectives

---

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



- ☐ Install Snort from source



- ☐ Configure a new Snort installation



- ☐ Validate Snort rules by sending suspicious signatures from a foreign host