Exploring Snort's Features and Modules

Matt Glass
CISSP, CEH

www.linkedin.com/in/matthewglass2

Overview

Snort's active response features

Basic modules and codecs

Inspector modules

Configuring AppId and Port Scan

Logger modules

Configuring alert_json

Snort's Active Response Features

Reject can shutdown potentially hostile connections

React can send an HTML page to a session and reset

Rewrite can replace the contents in a packet

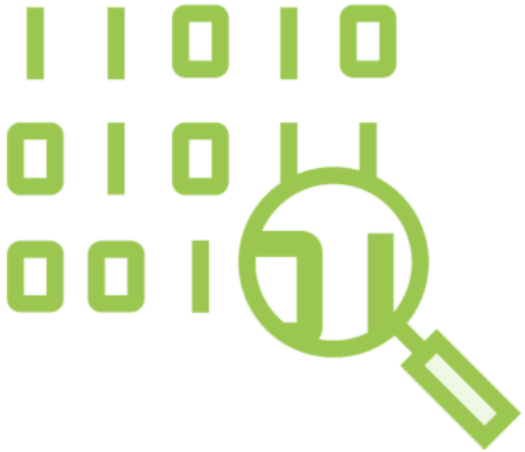
Basic Modules and Codecs

Basic modules are not plugins, but are responsible for Snort operation

Codecs are responsible for decoding and anomaly detection

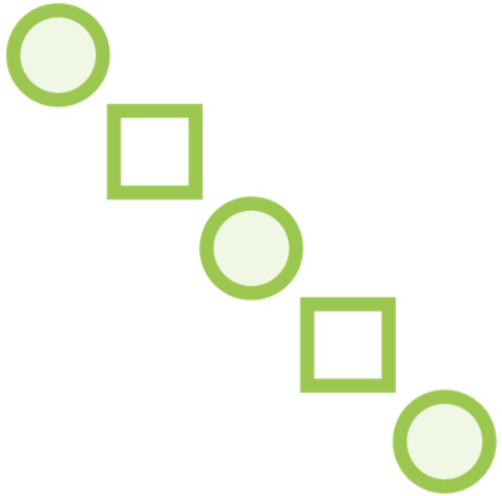
Snort 3.0 Inspector Modules

What are Inspector Modules?



Allow Snort to examine traffic in greater detail

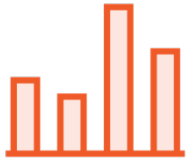
Each module performs a specific action



Enables rule writing based on patterns or applications

AppId

Provides application identifiers to Snort rules for processing



Outputs statistics for use in evaluating application usage

Contains pre-defined applications and allows for custom detectors

```
Using AppId with default settings:

appid = { }


Rule written for AppId:

alert tcp any any -> 10.0.0.0/24 any (msg:”Unauthorized inbound connection
attempt.”; appids:”ssh,telnet”; sid:10000004;)
```

Example AppId Configuration and Rule

Empty brackets tell Snort to load AppId with default settings

This alert will use AppId’s detection of SSH or Telnet use to trigger an alert

Port Scan

Detects network reconnaissance conducted through port scans

Specifically designed to detect Nmap scans

Detects TCP, UDP, and IP scans and variations of these three types

Three default levels and tuning options to refine detection

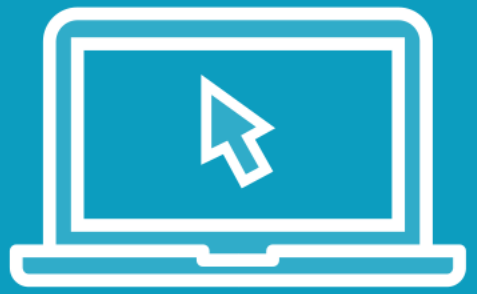
```
Default configuration in snort.lua

port_scan = default_med_port_scan


Changing port_scan to a low setting

port_scan = default_low_port_scan
```

Example Port Scan Configuration

The default level is set in snort.lua at medium

Changing to another default level is done by replace “med” with “low” or “hi”

Demo



Configure and test AppId

Satisfy the following security goals:

- Identify attempted SSH or telnet connections from external networks
- Obtain application usage statistics

Demo

Configure and test Port Scan

Satisfy the following security goal:

- Detect an attempted Nmap port scan

Snort 3.0 Logger Modules

Logger Modules

Logger modules are responsible for all event output

Multiple file types and output styles are available

Includes the Fast output primarily used in this course

```
alert_json =

{

    fields = ‘proto src_add src_port dst_addr dst_port service rule priority class
action b64_data’

}
```

Alert_JSON Example Configuration

Alert_JSON outputs Snort alerts in JSON format

Output fields are defined within the brackets

Demo

Satisfy the following security goal:

- Configure Snort to output in JSON format for processing by external applications

Summary

Summary

Extend Snort functionality

Configured modules to enable capability

Prepared alerting for external log storage