Snort Intrusion Detection System Cheatsheet

A comprehensive cheat sheet for Snort, covering installation, configuration, rule writing, and usage for network intrusion detection and prevention.

Installation and Basic Configuration

Installation

Ubuntu/Debian:

```
sudo apt-get update
sudo apt-get install snort
```

CentOS/RHEL:

```
sudo yum install snort
```

Download from Snort.org:

Download the latest version from the official Snort website and follow the installation instructions provided.

Basic Configuration File

The main configuration file is `snort.conf`. It is located in `/etc/snort/`.

Key configurations include defining network variables, setting up preprocessors, and specifying rule files.

Important variables to configure:

- `var HOME_NET`: The internal network(s) to protect.
- `var EXTERNAL_NET`: The external network(s), typically `!HOME_NET`.

Running Snort

Basic command	`sudo snort -dev -i eth0 -c /etc/snort/snort.conf` `-dev`: Display application layer data. `-i eth0`: Listen on interface eth0. `-c`: Specify the configuration file.
Test Configuration	`sudo snort -T -c /etc/snort/snort.conf` `-T`: Test the configuration file for errors.
Run in NIDS mode	`sudo snort -D -q -u snort -g snort -c /etc/snort/snort.conf -i eth0` `-D`: Run as a daemon. `-q`: Quiet mode (no console output). `-u` and `-g`: Specify user and group.

Snort Rule Structure

Rule Header

The rule header defines the action, protocol, source, and destination information.

Syntax:

```
action protocol src_ip src_port -> dst_ip dst_port (options)
```

Example:

```
alert tcp any any -> 192.168.1.0/24 80 (content:"GET"; msg:"HTTP GET detected";)
```

Rule Actions

`alert`	Generates an alert using the selected method.
`log`	Logs the packet.
`pass`	Ignores the packet.
`drop`	Drops the packet and logs it (inline mode only).
`reject`	Drops the packet and sends a TCP reset (for TCP) or ICMP port unreachable (for UDP) (inline mode only).
`sdrop`	Drops the packet but does not log it (inline mode only).

Rule Options

Rule options provide detailed inspection and action parameters within the rule. They are enclosed in parentheses `()`.

Key options include `msg`, `content`, `flow`, `depth`, `offset`, `distance`, `within`, `flags`, `ttl`, and `classtype`.

Common Rule Options

Content Matching

`content:"string";`	Matches the specified string in the packet payload. **Example:** `content:"/etc/passwd";`
`nocase`	Makes the content match case-insensitive. **Example:** `content:"GET"; nocase;`
`depth:value;`	Specifies the maximum number of bytes to search within the payload. **Example:** `content:"<script>"; depth:20;`
`offset:value;`	Specifies the starting byte to begin the search. **Example:** `content:"password"; offset:10;`
`distance:value;`	Specifies the minimum distance from the previous content match. **Example:** `content:"user"; distance:5; content:"pass";`
`within:value;`	Specifies the number of bytes that the content must be within after a previous match. **Example:** `content:"user"; within:10; content:"pass";`

Flow Control

`flow:established,to_server;`	Checks for established connections from client to server.
`flow:stateless;`	Ignores the flow state.

Metadata and Classifications

`msg:"message";`	Specifies the message to display when the rule is triggered.
`classtype:trojan-activity;`	Categorizes the type of attack or activity.
`sid:1000001;`	Specifies the Snort ID of the rule. Should be unique.
`rev:1;`	Specifies the revision number of the rule.

Advanced Rule Examples

Detecting Shellcode

```
alert tcp any any -> $HOME_NET 80
(content:"|90 90 90 90|"; msg:"Possible
shellcode detected"; sid:1000002; rev:1;)
```

This rule detects the presence of No Operation (NOP) sleds, which are commonly used in shellcode.

Detecting SQL Injection

```
alert tcp any any -> $HOME_NET 80
(content:"select "; nocase; msg:"Possible SQL
Injection"; sid:1000003; rev:1;)
```

This rule detects SQL injection attempts by looking for common SQL keywords in HTTP traffic.

Detecting Specific User-Agent

```
alert tcp any any -> $HOME_NET 80 (http_uri;
content:"User-Agent: BadBot"; msg:"BadBot
User-Agent Detected"; sid:1000004; rev:1;)
```

This rule detects a specific user agent string in HTTP requests.

File Integrity Monitoring

Snort can be configured with tools like `ossec` for enhanced file integrity monitoring and log analysis.

This typically involves integrating Snort alerts with OSSEC to provide real-time monitoring and alerting of file changes.