Setting Up Intrusion Prevention System (IPS) Using Snort

Created By Love

Intrusion Detection System (IDS) vs. Intrusion Prevention System (IPS) :

- **IDS :** Detects intrusions in a network or system but does not take action to stop them.

- **IPS :** Not only detects intrusions but also takes countermeasures to prevent them (e.g., blocking malicious traffic).

Note : *For this project, Snort will act as an IPS by actively blocking malicious traffic in real-time.*

Types of IDPS Technologies :

The IDPS categorizes into four types based on the events they monitor and their deployment:

- **Network-Based IDPS (NIDPS) :**

  Monitors network traffic for suspicious activity.

  Example: Snort operates as a NIDPS when analyzing packets in real-time.

- **Wireless-Based IDPS :**

Analyzes wireless network traffic to detect unauthorized access points or suspicious activities.

- **Network Behavior Analysis (NBA) :**

  Detects unusual traffic patterns, such as DDoS attacks or malware activity.

- **Host-Based IDPS (HIDPS) :**

  Monitors individual hosts for suspicious activities (e.g., file system changes, system calls).

  Useful for detecting attacks that bypass network-based systems.

Note : *For this project, we'll primarily focus on Network-Based IDPS since Snort is designed to analyze network traffic.*

Methodologies for Intrusion Detection :

The paper outlines several methodologies used by IDPS systems:

- **Anomaly-Based Detection :**

  Detects unknown attacks by identifying deviations from normal behavior.

  Disadvantage: High false-positive rate.

- **Signature-Based Detection :**

  Detects known attacks by comparing network traffic against a database of predefined signatures.

  Example: Snort uses signature-based detection to identify malicious packets.

- **Stateful Protocol Analysis :**

  Analyzes the state of network connections to detect anomalies.

Example: Tracks TCP handshake states to identify suspicious behavior.

- **Hybrid-Based Detection :**

Combines anomaly-based and signature-based detection to improve accuracy and reduce false positives.

Note: *we'll likely use signature-based detection because Snort excels at this methodology.*

Snort Rule Implementation :

Step 1: Finding the Snort Rules

We can see the Snort rules by navigating to ***/etc/snort/rules*** on our Kali install. Let's go to that directory and take a look.

$ cd /etc/snort/rules

Step 2: Examining a Rule

Let's take a simple rule and dissect it. Let's open the file scan.rules.

$ mousepad/etc/snort/scan.rules

Step 3: Dissecting the SF Scan Rule

Let's start by examining the first part of that rule from the beginning to the first starting parenthesis. This initial part of the rule is referred to as the header, and ours looks like this:

alert tcp $EXTERNAL_NET any -> $HOME_NET any

This rule is designed to detect scans by scanning tools such as nmap. One of those scans is called a SYN-FIN scan. This scan sends TCP packets with both the SYN and the FIN flags set to attempt to determine what ports are open on the target system. A packet with both of these

flags set should never appear--except for malicious purposes--as these two flags simultaneously open and close a TCP connection.

Therefore, it's easy to see that any packet with these flags set must be an attempt to do recon on the system and the sysadmin should be alerted.

Understand Snort Rule Syntax :

Snort rules are written in a specific format. Each rule consists of two main parts:

- **Rule Header :** Defines the action, protocol, source/destination IP addresses, and ports.

For ex : ***alert icmp any any -> any any***

- **Rule Options :** Specifies additional details like content matching, message descriptions, and metadata.

For ex : **(msg:"ICMP Ping Detected"; sid:1000001; rev:1;)**

Basic Rule Format :

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

Action : What Snort should do when the rule matches:

- **alert:** Generate an alert and log the packet.
- **log:** Log the packet without generating an alert.
- **pass:** Ignore the packet (do not process further).

drop / **reject** / **sdrop:** Block or reject the packet (only available in inline mode).

Protocol : The network protocol to match (tcp, udp, icmp, ip, etc.).

- **Source/Destination IPs and Ports :**

Use any to match any IP or port.

Use specific IPs or CIDR ranges (e.g., 192.168.1.0/24).

Use port numbers or ranges (e.g., 80, 1024:65535).

Direction (->) : Indicates the direction of traffic:

→ : Traffic flows from source to destination.

<> : Traffic flows in both directions.

Options : Additional criteria for matching packets (e.g., content, message, etc.).

Example Rule : Prevent ICMP Traffic

The Snort rules files are simple text files, so we can open and edit them with any text editor. I'll be using mousepad , but you can use vi, gedit, or any text editor you prefer.

Let's open the file local.rules. To write a new rule .

$ mousepad /etc/snort/rules/local.rules

This rule prevent all ICMP traffic (e.g., ping requests) :

drop icmp any any -> any any (msg:"ICMP Ping Detected"; sid:1000001; rev:1;)

Explanation :

drop : block and log malicious traffic.

icmp: Match ICMP traffic.

any any -> any any: Match any source/destination IP and port.

(msg:"ICMP Ping Prevented"): Add a descriptive message to the alert.

(sid:1000001): Unique rule ID (Signature ID).

(rev:1): Revision number of the rule

Applying the Snort rule :

Open the Snort configuration file (snort.lua) in a text editor:

$ sudo mousepad /etc/snort/snort.lua

Locate the ips section and ensure it includes the path to your local.rules file:

```
ips =
{
-- Enable decoder and inspector alerts if needed
-- enable_builtin_rules = true,
-- Include custom rules
include = '/etc/snort/rules/local.rules',
variables = default_variables
}
```

Save and close the file.

- **Start Snort with the Updated Configuration**

Use the following command to start Snort with your custom rule:

$ sudo snort -c /etc/snort/snort.lua -i wlan0 -A alert_fast

- -c /etc/snort/snort.lua: Specifies the configuration file.
- -i wlan0: Specifies the network interface to monitor (replace wlan0 with your actual interface).
- -A alert_fast: Outputs alerts in a simplified format for easier readability.

- **Generate ICMP Traffic to Test the Rule**

On another device or terminal, generate ICMP traffic (e.g., ping requests) to test the rule:

$ ping <IP_of_Snort_machine>

Replace <IP_of_Snort_machine> with the IP address of the machine running Snort.

- **Observe Alerts in Real-Time**

If the rule is correctly configured, Snort will block the ICMP traffic and log the event. In the terminal where Snort is running, you should see output similar to this:

[**] [1:1000001:1] ICMP Ping Prevented [**]

[Priority: 0] {ICMP} 192.168.31.104 -> 192.168.31.120