Lab: SNORT Intrusion Detection System

Lab Objectives

Become familiar with SNORT as packet sniffer and intrusion detection system

Create custom rules in SNORT to detect various attacks such as ping scan and port scan Test the custom rules and analyze the snort log files

SNORT Overview

Snort's architecture consists of four basic components, the sniffer, preprocessor, detection engine and the alerts, as shown if Figure 1. The sniffer is a very useful tool, in operational mode it can be used as a stand-alone packet sniffer much like Wireshark or tcpdump; however, it's main objective is to capture packets and forward them to the preprocessor. On arrival, the preprocessor checks the data for abnormalities such as IP fragmentation and sends it on to the detection engine. The detection engine extracts the data and checks it against a set of known rules, if the data within the packet matches the rule attributes it's then transmitted to the alert processor. At this point, the alert processor either logs the alert or more importantly, notifies the administrator of the occurrence [1]. Although Snort is equipped with powerful features, its true strengths are concealed within the proficiency of custom rules.

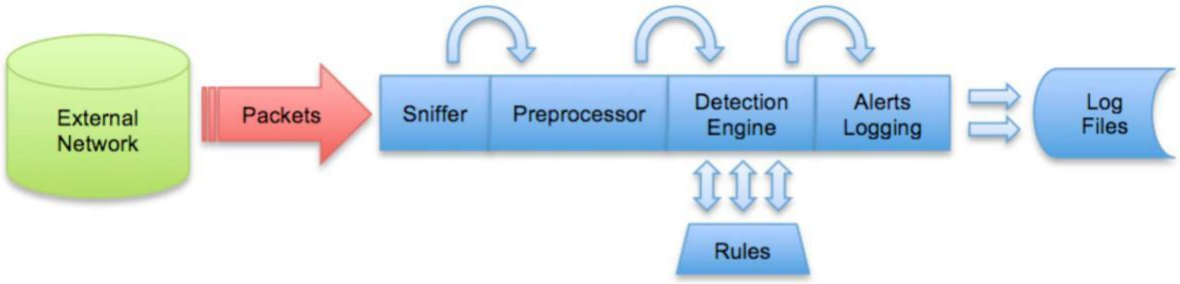
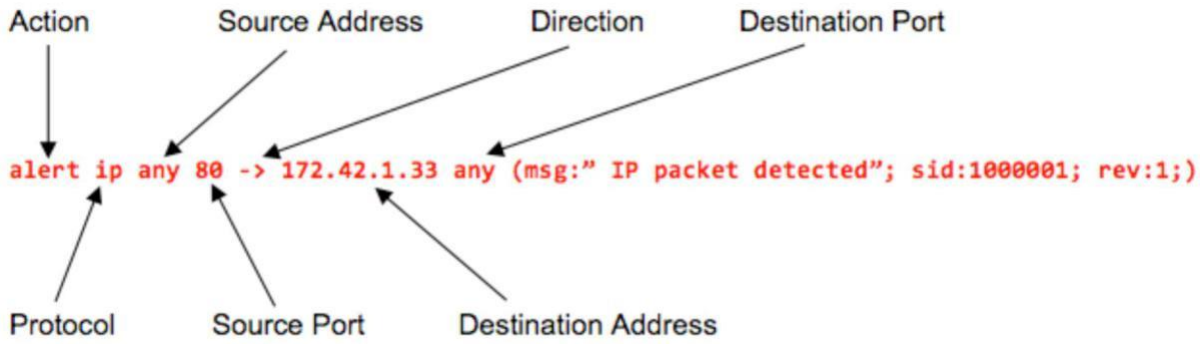


FIGURE 1. Snort's architecture.

Snort rules consist of two major elements, the rule header and rule options. The rule header is composed of seven fields and defines the main attributes of the packet. The first field in the header is the action field, in the event that all attributes have been satisfied it defines what to do next. The protocol field is used to establish what type of packet the rule applies to. It can be defined as TCP, UDP, ICMP or IP, which can be any of the preceding three. The address fields define the source and destination addresses, which can be specified as single hosts, multiple hosts or network specific. The port fields, much like the address fields define the origin as well as the destination, however they are used to describe the port in question. The final field is the direction, which describes the flow of traffic from source to destination. The address on the left of the field is always the source and the address on the right is always the destination.

Example of a very basic Snort rule:



This rule will trigger an alert when an http (port 80) packet from any IP address is sent to the destination IP address of 172.42.1.33. Displayed within the alert file will be the message "IP packet detected."

MATERIALS AND SETUP

To complete this lab, you will need an attacker and victim/defender machine
Kali linux VM can be your attacker
Metasploitable VM or Windows XP vm or your own machine can be the victim/target. Wireshark (Optional) – to observe packet flow

Feel free to create your own configurations, but describe that in your lab report.

LAB TASKS

Task 1: Use Snort as a Packet Sniffer.

Demonstrate how snort can be used as a packet sniffer.
On the command line type snort –vde and press ENTER. This command and options will run Snort as a Packet Sniffer

For this task, ping a different ip address and observe the snort output

Task 2: Run Snort as IDS to detect ping scans

Write custom rules to detect the ping scan, write configuration file, test the rule and check the logs

Task 3: Run Snort as IDS to detect port scans

Write custom rules to detect various types of port scans. Recall the nmap scans including SYN scan, FIN scan and XMAS scan

STEP 1: Use Nmap to launch a SYN scan attack, create a rule to detect the attack, test the rule and check the logs

STEP2: Use Nmap to launch a FIN scan attack, create a rule to detect the attack, test the rule and check the logs

In a FIN scan the attacker is searching for open ports using only the FIN flag. This is notifying the target that is wants to tear down a connection, even though no connection is present. Any packet not containing a SYN, RST or ACK will result in a returned RST if the port is closed and no response if the port is open.

STEP3: Use Nmap to launch a XMAS scan attack, create a rule to detect the attack, test the rule and check the logs

In an XMAS scan, the attacker is searching for open ports using the FIN, PSH and URG flags; As mentioned back in Step 2, any combination of these three flags will result in a returned RST if the port is closed and no response if the port is open.

Task 4: Write your own custom rule to detect ANY other attack on the victim machine. Demonstrate the attack and show how snort is capable of detecting this attack.

Task 5: Write a rule that will fire when you browse to facebook.com from the machine Snort is running on; it should look for any outbound TCP request to facebook.com and alert on it.

Task6: Answer the following questions

Q1: State how each of following real rules from the snort home page work:

i. **alert icmp any any -> any any (msg:"ICMP Source Quench"; itype: 4; icode: 0;)**

ii. **alert tcp $EXTERNAL_NET any -> $HTTP_SERVERS 80 (msg:"WEB-CGI view-source access";flags: A+; content:"/view source?../../../../../../../etc/passwd"; nocase;reference:cve,CVE-1999-0174;)**

Q2: Develop your own snort signature to capture DNS queries directed against the host the you choose to connect to via HTTPS. Make sure that your snort rule references the DNS data and not simply IP address of the server.