UNIVERSITY OF
LIMERICK
OLLSCOIL LUIMNIGH

EE6042 Network and Host Security

IDS Assignment

Bhavya Gaur

22079084

Contents

Setup

My setup consists of two virtual machines (VMs) as follows:

- **Victim + IDS VM**: both victim and IDS (snort) is running on the same VM. This VM is Ubuntu 22.04 LTS. The IP address of this VM is 192.168.150.129
- **Attacker VM**: This VM is Kali linux and its IP address is 192.168.150.128

I have configured **snort.conf** as follows:

Line 71: ipvar HOME_NET 192.168.150.0/24

This config sets our own subnet where the IDS operates which on 192.168.150.0/24.

Line 557: output alert_full: bhavya-alert-full.txt

This line sets the name of the detailed logs produced by the rules that match the packets.

Line 571: output alert_fast: bhavya-alert-fast.txt

This line sets the name of the condensed logs produced by the rules that match the packets.

Line 597: include $RULE_PATH/local.rules

This line sets the place where we will write the rules.

Note:

- I a have written my custom rules in local.rules file not in the snort.conf file.
- I have disabled all rules except for mine (this is to reduce the size of bhavya-alert-fast.txt and bhavya-alert-full.txt).
- I have provided bhavya-alert-full.txt just for reference I am not providing the line numbers.

Descriptions of the Rules

Attack 1: Detecting Web Scraper or Bot activity

Line 18:

alert tcp any any -> 192.168.150.129 80 (msg:"Bot User-Agent detected"; flow:stateless; content:"User-Agent"; http_header; pcre:"/.*robot/i"; sid:1000002; rev:1;)

This snort rule generates an alert when it detects TCP traffic with a destination IP of 192.168.150.129 and destination port 80, containing an HTTP header with a "User-Agent" field that includes the case-insensitive string "robot".

Description of each component:

1) **alert**: This is the action to be taken when the rule is triggered. In this case, it will generate an alert.

2) **tcp**: The rule will monitor TCP traffic.

3) **any any**: The source IP address and port are "any" which means that the rule will apply to all source IP addresses and ports.

4) **192.168.150.129 80**: Destination IP address is 192.168.150.129 and destination port is 80.

5) **msg**: This will log the message when the rule is triggered.

6) **flow:stateless**: This option indicates that the rule will not consider the state of the TCP connection.

7) **content:"User-Agent"**: This is a content-matching option, which searches for the specified string "User-Agent" in the packet.

8) **http_header**: This option indicates that the rule will search for content within the HTTP headers.

9) **pcre:"/.*robot/i"**: This is a Perl Compatible Regular Expression (PCRE) option that searches for the case-insensitive pattern "robot" within the packet.

10) **sid:1000001**: This is the Snort ID, a unique identifier for this rule.

11) **rev:1**: This is the revision number of the rule, indicating this is the first version.

This rule is designed to detect HTTP requests containing a "User-Agent" header indicative of a bot or a robot, which might be a web scraper, a malicious bot, or an automated script.

Log entry: please open file **bhavya-alert-fast.txt** and examine **line 2**

Example log entry: 04/26-22:50:52.997798 [**] [1:1000002:1] Bot User-Agent detected [**] [Priority: 0] {TCP} 192.168.150.128:52210 -> 192.168.150.129:80

To test this rule run the following command:

CMD: curl -A "robot" "http://192.168.150.129"

Attack 2: Detecting ICMP Flooding Attack

Line 22

alert icmp any any -> any any (msg:"ICMP echo detected"; itype:8; sid:1000003; rev:1;)

The rule is designed to detect Internet Control Message Protocol (ICMP) echo requests, also known as ping requests. This rule is designed to generate an alert when an ICMP echo request (ping) is detected in the network traffic, regardless of the source or destination IP addresses and ports involved.

Description of each component:

1) **itype:8**: This is an option specific to the ICMP protocol. The itype stands for ICMP type, and the number 8 corresponds to an ICMP echo request.

Remaining options are similar to attack 1.

This rule is useful for detecting ICMP flooding attacks. If we see a huge number of ICMP echo request from one IP address it means that the attacker is trying to overload our machine.

Log entry: please open file **bhavya-alert-fast.txt** and examine **line 3**

Example log entry: 04/26-22:50:57.900162 [**] [1:1000003:1] ICMP echo detected [**] [Priority: 0] {ICMP} 192.168.150.128 -> 192.168.150.129

To test this rule run the following command:

CMD: ping -c 1 192.168.150.129

Attack 3: Detecting SSH Brute Force Attack

Line 26

alert tcp any any -> 192.168.150.129 22 (msg:"Possible SSH Brute Force Detected"; flags:S; flow:stateless; detection_filter:track by_src, count 10, seconds 30; sid:1000004; rev:1;)

This rule is designed to generate an alert when a possible SSH brute force attempt is detected, specifically targeting IP address 192.168.150.129 on port 22. The rule will trigger if at least 10 connection attempts with the SYN flag are observed from the same source IP address within a 30-second time frame.

Description of each component:

1) **flags:S**: This option checks for for the SYN flag “S” in the TCP header.

2) **detection_filter:track by_src, count 10, seconds 30**: This option adds a rate-based detection filter to the rule. The filter tracks the source IP address using “by_src” and triggers the rule only if it detects at least 10 events from the same source IP address within a 30-second window.

Remaining options are similar to attack 1.

Log entry: please open file **bhavya-alert-fast.txt** and examine **line 4** (Huge number of similar logs were made because Metasploit ssh fuzzer tests for several strings resulting in several packets)

Example log entry: 04/26-22:51:02.347758 [**] [1:1000004:1] Possible SSH Brute Force Detected [**] [Priority: 0] {TCP} 192.168.150.128:42543 -> 192.168.150.129:22

To test this rule run the following commands from the attacker machine:

CMD: msfconsole

CMD: use fuzzers/ssh/ssh_version_2

CMD: set RHOSTS 192.168.150.129

CMD: run

Testing Snort

This is the simplest rule, it was written to check if the snort is working or not. Please ignore this rule.

Line 14

```
alert tcp any any -> 192.168.150.129 80 (msg:"Admin portal login detected without https"; content:"admin"; nocase; http_uri; sid:1000001; rev:1;)
```

Log entry: please open file **bhavya-alert-fast.txt** and examine **line 1**

CMD: curl http://192.168.150.129/admin-login