Snort Lab Hand-In

Zhenghong Dong, Network Security

Operation System Used: Ubuntu 1410 LTS 64bits

Snort information: Version 2.9.6.0 GRE (Build 47)
Using libpcap version 1.6.2
Using PCRE version: 8.35 2014-04-04
Using ZLIB version: 1.2.8

How to install Snort: in the terminal, run command:

sudo apt-get install snort*

then enter your user password for the operation system when asked.

Command line used for each question:

sudo snort -r /tmp/snort-ids-lab.log.out -P 5000 -c /tmp/rules -e -X -k none

The rules are located at /tmp/ . In Ubuntu normally a user can not login as root, thus sudo is required for snort installation and running command.

Task 1: Read the help file, write in your lab write-up what each of those flags should do.

Result:

The meaning of flags of snort command:

snort -r /tmp/snort-ids-lab.log -P 5000 -c /tmp/rules -e -X -v -k none

-r <path/file> read and process tcpdump file <path/file>
-P <snap> set explicit snap length of a packet (default 1514)
-c <rules> use rule file <rules>
-e display the second layer header info
-X dump the raw packet data starting at the link layer
-v be verbose(print output on the screen)
-k <mode> Checksum mode (all,noip,notcp,noudp,noicmp,none)

Task 2: Run this single rule on the packet trace.

alert icmp any any -> 192.168.10.2 any (itype:8; msg:"ping detected";)

The results will be written to /var/log/snort/alert. In your write up, state why the value 8 was used. And, include the output of that command.

Answer: itype means ICMP type, “itype: 8” means type 8 of ICMP, which is echo (request).

A list of itypes:

Value	Type of ICMP Packet
0	Echo reply
3	Destination unreachable
4	Source quench
5	Redirect
8	Echo request
11	Time exceed
12	Parameter problem
13	Timestamp request
14	Timestamp reply
15	Information request
16	Information reply

Running Snort with the rule produce result as below:

[**] [1:10000002:0] ping detected [**]
[Priority: 0]
05/04-08:53:52.830835 00:50:04:5B:64:5B -> 00:10:4B:E2:65:8E type:0x800 len:0x88
192.168.1.255 -> 192.168.10.2 ICMP TTL:254 TOS:0x0 ID:11122 IpLen:20 DgmLen:122 DF
Type:8 Code:0 ID:11122 Seq:0 ECHO

[**] [1:10000002:0] ping detected [**]
[Priority: 0]
05/04-08:53:52.701466 00:50:04:5B:64:5B -> 00:10:4B:E2:65:8E type:0x800 len:0x88
192.168.1.255 -> 192.168.10.2 ICMP TTL:254 TOS:0x0 ID:11122 IpLen:20 DgmLen:122 DF
Type:8 Code:0 ID:11122 Seq:0 ECHO

[**] [1:10000002:0] ping detected [**]
[Priority: 0]
05/04-08:53:52.965429 00:50:04:5B:64:5B -> 00:10:4B:E2:65:8E type:0x800 len:0x88
192.168.1.255 -> 192.168.10.2 ICMP TTL:254 TOS:0x0 ID:11122 IpLen:20 DgmLen:122 DF
Type:8 Code:0 ID:11122 Seq:0 ECHO

Question 1:

Rule 1: filter out ssh connection by checking port 22 on local machine:

alert tcp any any -> 192.168.10.2 22 (msg: "ssh connection";sid:10000003;)

output

[**] [1:10000003:0] ssh connection [**]
[Priority: 0]
05/04-08:53:52.800369 00:50:04:5B:64:5B -> 00:10:4B:E2:65:8E type:0x800 len:0x23B
192.168.1.5:42069 -> 192.168.10.2:22 TCP TTL:254 TOS:0x0 ID:83 IpLen:20 DgmLen:557 DF
****PR** Seq: 0x201F Ack: 0x3992 Win: 0x200 TcpLen: 20

[**] [1:10000003:0] ssh connection [**]
[Priority: 0]
05/04-08:53:52.809583 00:50:04:5B:64:5B -> 00:10:4B:E2:65:8E type:0x800 len:0x23B
192.168.1.5:42069 -> 192.168.10.2:22 TCP TTL:254 TOS:0x0 ID:84 IpLen:20 DgmLen:557 DF
****PR** Seq: 0x201F Ack: 0x3992 Win: 0x200 TcpLen: 20

[**] [1:10000003:0] ssh connection [**]
[Priority: 0]
05/04-08:53:52.819774 00:50:04:5B:64:5B -> 00:10:4B:E2:65:8E type:0x800 len:0x23B
192.168.1.5:42069 -> 192.168.10.2:22 TCP TTL:254 TOS:0x0 ID:85 IpLen:20 DgmLen:557 DF
****PR** Seq: 0x201F Ack: 0x3992 Win: 0x200 TcpLen: 20

[**] [1:10000003:0] ssh connection [**]
[Priority: 0]
05/04-08:53:52.670939 00:50:04:5B:64:5B -> 00:10:4B:E2:65:8E type:0x800 len:0x23B
192.168.1.5:42069 -> 192.168.10.2:22 TCP TTL:254 TOS:0x0 ID:87 IpLen:20 DgmLen:557 DF
****PR** Seq: 0x201F Ack: 0x3992 Win: 0x200 TcpLen: 20

[**] [1:10000003:0] ssh connection [**]
[Priority: 0]
05/04-08:53:52.680196 00:50:04:5B:64:5B -> 00:10:4B:E2:65:8E type:0x800 len:0x23B
192.168.1.5:42069 -> 192.168.10.2:22 TCP TTL:254 TOS:0x0 ID:88 IpLen:20 DgmLen:557 DF
****PR** Seq: 0x201F Ack: 0x3992 Win: 0x200 TcpLen: 20

[**] [1:10000003:0] ssh connection [**]
[Priority: 0]
05/04-08:53:52.690413 00:50:04:5B:64:5B -> 00:10:4B:E2:65:8E type:0x800 len:0x23B
192.168.1.5:42069 -> 192.168.10.2:22 TCP TTL:254 TOS:0x0 ID:89 IpLen:20 DgmLen:557 DF
****PR** Seq: 0x201F Ack: 0x3992 Win: 0x200 TcpLen: 20

[**] [1:10000003:0] ssh connection [**]
[Priority: 0]
05/04-08:53:52.934930 00:50:04:5B:64:5B -> 00:10:4B:E2:65:8E type:0x800 len:0x23B

192.168.1.5:42069 -> 192.168.10.2:22 TCP TTL:254 TOS:0x0 ID:91 IpLen:20 DgmLen:557 DF
****PR** Seq: 0x201F Ack: 0x3992 Win: 0x200 TcpLen: 20

[**] [1:10000003:0] ssh connection [**]
[Priority: 0]
05/04-08:53:52.944089 00:50:04:5B:64:5B -> 00:10:4B:E2:65:8E type:0x800 len:0x23B
192.168.1.5:42069 -> 192.168.10.2:22 TCP TTL:254 TOS:0x0 ID:92 IpLen:20 DgmLen:557 DF
****PR** Seq: 0x201F Ack: 0x3992 Win: 0x200 TcpLen: 20

[**] [1:10000003:0] ssh connection [**]
[Priority: 0]
05/04-08:53:52.953160 00:50:04:5B:64:5B -> 00:10:4B:E2:65:8E type:0x800 len:0x23B
192.168.1.5:42069 -> 192.168.10.2:22 TCP TTL:254 TOS:0x0 ID:93 IpLen:20 DgmLen:557 DF
****PR** Seq: 0x201F Ack: 0x3992 Win: 0x200 TcpLen: 20

Rule 2: filter out TCP requests to DNS port 53:

alert TCP any any -> 192.168.10.2 53 (msg: “TCP requests to DNS port”;sid:10000004;)

Output:

[**] [1:10000004:0] “TCP requests to DNS port" [**]
[Priority: 0]
05/04-08:53:52.790472 00:50:04:5B:64:5B -> 00:10:4B:E2:65:8E type:0x800 len:0x52
192.168.1.5:32569 -> 192.168.10.2:53 TCP TTL:254 TOS:0x0 ID:420 IpLen:20 DgmLen:68 DF
******** Seq: 0x1A4 Ack: 0x1A4 Win: 0x200 TcpLen: 20

[**] [1:10000004:0] “TCP requests to DNS port" [**]
[Priority: 0]
05/04-08:53:52.660053 00:50:04:5B:64:5B -> 00:10:4B:E2:65:8E type:0x800 len:0x52
192.168.1.5:32569 -> 192.168.10.2:53 TCP TTL:254 TOS:0x0 ID:420 IpLen:20 DgmLen:68 DF
******** Seq: 0x1A4 Ack: 0x1A4 Win: 0x200 TcpLen: 20

[**] [1:10000004:0] “TCP requests to DNS port" [**]
[Priority: 0]
05/04-08:53:52.923922 00:50:04:5B:64:5B -> 00:10:4B:E2:65:8E type:0x800 len:0x52
192.168.1.5:32569 -> 192.168.10.2:53 TCP TTL:254 TOS:0x0 ID:420 IpLen:20 DgmLen:68 DF
******** Seq: 0x1A4 Ack: 0x1A4 Win: 0x200 TcpLen: 20

Rule 3: filter out UDP requests to DNS port 53:

alert UDP any any -> 192.168.10.2 53 (msg: “UDP requests to DNS port”;sid:10000005;)

Output:

[**] [1:10000005:0] “UDP requests to DNS port” [**]
[Priority: 0]
05/04-08:53:52.777745 00:50:04:5B:64:5B -> 00:10:4B:E2:65:8E type:0x800 len:0x40
192.168.1.5:53 -> 192.168.10.2:53 UDP TTL:254 TOS:0x0 ID:82 IpLen:20 DgmLen:50 DF

Len: 22

[**] [1:10000005:0] “UDP requests to DNS port” [**]
[Priority: 0]
05/04-08:53:52.845315 00:50:04:5B:64:5B -> 00:10:4B:E2:65:8E type:0x800 len:0x40
192.168.1.5:53 -> 192.168.10.2:53 UDP TTL:254 TOS:0x0 ID:86 IpLen:20 DgmLen:50 DF
Len: 22

[**] [1:10000005:0] “UDP requests to DNS port” [**]
[Priority: 0]
05/04-08:53:52.715830 00:50:04:5B:64:5B -> 00:10:4B:E2:65:8E type:0x800 len:0x40
192.168.1.5:53 -> 192.168.10.2:53 UDP TTL:254 TOS:0x0 ID:90 IpLen:20 DgmLen:50 DF
Len: 22

Rule 4: filter out TCP requests to web server port 80:

alert TCP any any -> 192.168.10.2 80 (msg: “TCP requests to web server port”;sid:10000006;)

Output:

[**] [1:10000006:0] “TCP requests to web server port” [**]
[Priority: 0]
05/04-08:53:52.784584 00:50:04:5B:64:5B -> 00:10:4B:E2:65:8E type:0x800 len:0x5C
192.168.1.5:42069 -> 192.168.10.2:80 TCP TTL:254 TOS:0x0 ID:666 IpLen:20 DgmLen:78 DF
A* Seq: 0x29A Ack: 0x29A Win: 0x29A TcpLen: 20

[**] [1:10000006:0] “TCP requests to web server port” [**]
[Priority: 0]
05/04-08:53:52.850955 00:50:04:5B:64:5B -> 00:10:4B:E2:65:8E type:0x800 len:0x5C
192.168.1.5:42069 -> 192.168.10.2:80 TCP TTL:254 TOS:0x0 ID:666 IpLen:20 DgmLen:78 DF
A* Seq: 0x29A Ack: 0x29A Win: 0x29A TcpLen: 20

[**] [1:10000006:0] “TCP requests to web server port” [**]
[Priority: 0]
05/04-08:53:52.918145 00:50:04:5B:64:5B -> 00:10:4B:E2:65:8E type:0x800 len:0x5C
192.168.1.5:42069 -> 192.168.10.2:80 TCP TTL:254 TOS:0x0 ID:666 IpLen:20 DgmLen:78 DF
A* Seq: 0x29A Ack: 0x29A Win: 0x29A TcpLen: 20

Rule 5: filter out requests to DHCP port 67 from broadcasting address:

alert UDP 255.255.255.255 any -> 192.168.10.2 67 (msg: “Requests to DHCP port from broadcasting address”;sid:10000007;)

Output:

[**] [1:10000007:0] “Requests to DHCP port from broadcasting address” [**]
[Priority: 0]

05/04-08:53:52.836019 00:50:04:5B:64:5B -> 00:10:4B:E2:65:8E type:0x800 len:0x3C
255.255.255.255:68 -> 192.168.10.2:67 UDP TTL:1 TOS:0x0 ID:2513 IpLen:20 DgmLen:41 DF
Len: 13

[**] [1:10000007:0] “Requests to DHCP port from broadcasting address” [**]
[Priority: 0]
05/04-08:53:52.706591 00:50:04:5B:64:5B -> 00:10:4B:E2:65:8E type:0x800 len:0x3C
255.255.255.255:68 -> 192.168.10.2:67 UDP TTL:1 TOS:0x0 ID:2513 IpLen:20 DgmLen:41 DF
Len: 13

[**] [1:10000007:0] “Requests to DHCP port from broadcasting address” [**]
[Priority: 0]
05/04-08:53:52.970570 00:50:04:5B:64:5B -> 00:10:4B:E2:65:8E type:0x800 len:0x3C
255.255.255.255:68 -> 192.168.10.2:67 UDP TTL:1 TOS:0x0 ID:2513 IpLen:20 DgmLen:41 DF
Len: 13

Rule 6: filter out “Bad Traffic Same Src/Dst IP”:

alert ip any any -> 192.168.10.2 any (msg: “Bad Traffic Same Src/Dst IP”;sid:
10000008;sameip;)

Output:

[**] [1:10000008:0] “Bad Traffic Same Src/Dst IP” [**]
[Priority: 0]
05/04-08:53:52.824985 00:50:04:5B:64:5B -> 00:10:4B:E2:65:8E type:0x800 len:0x3C
192.168.10.2:0 -> 192.168.10.2:0 UDP TTL:254 TOS:0x0 ID:14733 IpLen:20 DgmLen:33 DF
UDP header truncated

[**] [1:10000008:0] “Bad Traffic Same Src/Dst IP” [**]
[Priority: 0]
05/04-08:53:52.695596 00:50:04:5B:64:5B -> 00:10:4B:E2:65:8E type:0x800 len:0x3C
192.168.10.2:0 -> 192.168.10.2:0 UDP TTL:254 TOS:0x0 ID:14733 IpLen:20 DgmLen:33 DF
UDP header truncated

[**] [1:10000008:0] “Bad Traffic Same Src/Dst IP” [**]
[Priority: 0]
05/04-08:53:52.958304 00:50:04:5B:64:5B -> 00:10:4B:E2:65:8E type:0x800 len:0x3C
192.168.10.2:0 -> 192.168.10.2:0 UDP TTL:254 TOS:0x0 ID:14733 IpLen:20 DgmLen:33 DF
UDP header truncated

Rule 7: filter packets marked “Destination Unreachable”:

alert icmp any any -> any any (itype:3; msg:"destination unreachable";sid:100000010;)

Output:

[**] [1:100000010:0] destination unreachable [**]
[Priority: 0]

05/04-08:53:52.772670 00:50:04:5B:64:5B -> 00:10:4B:E2:65:8E type:0x800 len:0x3E
192.168.1.5 -> 192.168.10.2 ICMP TTL:123 TOS:0x18 ID:345 IpLen:20 DgmLen:48 DF
Type:3 Code:4 DESTINATION UNREACHABLE: FRAGMENTATION NEEDED, DF SET
NEXT LINK MTU: 0
** ORIGINAL DATAGRAM DUMP:
192.168.10.2 -> 192.168.1.5 ICMP TTL:123 TOS:0x18 ID:0 IpLen:20 DgmLen:20 DF
** END OF DUMP

[**] [1:100000010:0] destination unreachable [**]
[Priority: 0]
05/04-08:53:52.840259 00:50:04:5B:64:5B -> 00:10:4B:E2:65:8E type:0x800 len:0x3E
192.168.1.5 -> 192.168.10.2 ICMP TTL:123 TOS:0x18 ID:345 IpLen:20 DgmLen:48 DF
Type:3 Code:4 DESTINATION UNREACHABLE: FRAGMENTATION NEEDED, DF SET
NEXT LINK MTU: 0
** ORIGINAL DATAGRAM DUMP:
192.168.10.2 -> 192.168.1.5 ICMP TTL:123 TOS:0x18 ID:0 IpLen:20 DgmLen:20 DF
** END OF DUMP

[**] [1:100000010:0] destination unreachable [**]
[Priority: 0]
05/04-08:53:52.710795 00:50:04:5B:64:5B -> 00:10:4B:E2:65:8E type:0x800 len:0x3E
192.168.1.5 -> 192.168.10.2 ICMP TTL:123 TOS:0x18 ID:345 IpLen:20 DgmLen:48 DF
Type:3 Code:4 DESTINATION UNREACHABLE: FRAGMENTATION NEEDED, DF SET
NEXT LINK MTU: 0
** ORIGINAL DATAGRAM DUMP:
192.168.10.2 -> 192.168.1.5 ICMP TTL:123 TOS:0x18 ID:0 IpLen:20 DgmLen:20 DF
** END OF DUMP

Question 2:

1.alert tcp $HOME_NET 23 -> $EXTERNAL_NET any (msg: “TELNET Login incorrect”; content: “Login incorrect”; flags: A+; reference: arachnids, 127;)

How it works: when the local machines send out tcp message through port 23, which is telnet port, with an ACK flag, it means that it’s responding to an telnet connection request from external network. If the response content contains “Login incorrect”, it is possibly an attempt to try weak passwords of telnet accounts.

2. alert udp $EXTERNAL_NET any -> $HOME_NET 53 (msg: “EXPLOIT BIND Tsig Overflow Attempt”; content: “|80 00 07 00 00 00 00 00 01 3F 00 01 02|/bin/sh”;)

How it works: check the DNS requests from external network with the content pattern that is related to exploiting Tsig overflow attempt. DNS works on UDP port 53.

3. alert tcp $EXTERNAL_NET any -> $HOME_NET any (msg: “SCAN FIN”; flags: F; reference: arachnids,27;)

How it works: attacker may send tcp request with FIN flag to scan opened ports on the target machine. This rule checks tcp request from external network with FIN flags indicating possible TCP FIN scan.

4. alert tcp $EXTERNAL_NET any -> $HOME_NET 23 (msg: “MISC linux rootkit attempt lrkr0x”; flags: A+; content: “ lrkr0x”;)

How it works: “ lrkr0x” is the default password used by attacker to log into rootkit account. Checking message with this content may reveal rootkit login attempt.

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

How it works: tcp requests with content showed in the rule indicate the attempt to read the password file from the local server. “nocase" option is set to make checking case insensitive.

7. alert icmp any any ->any any (msg: “ICMP Source Quench”; itype:4; icode:0;)

How it works: The ICMP Source quench message is a request to decrease the traffic rate of data messages sent to an internet destination. It’s been done by sending ICMP message with itype set to 4. Detecting ICMP with itype 4 can reveal source quench.

8. alert tcp $EXTERNAL_NET any -> $HOME_NET 53 (msg: “DNS EXPLOIT named overflow”; flags: A+; content: “thisissometempspaceforthesockinaddrinyeahyeahiknowthisislamebutanywaywhocareshorizongotitworkingsoalliscool”; reference:cve, CVE-1999-0833;)

How it works: tcp requests send to local server DNS port with unusual content pattern included in the rule is a usual way for some attack kit to exploit DNS named overflow vulnerability. Thus checking message with such pattern reveals attacks.

9. alert tcp $EXTERNAL_NET any -> $HOME_NET 139 (msg: “NETBIOS SMB ADMIN $access”; flow:to_server,established; content: “\\ADMIN$\00 41 3a 00\”;reference:arachnids, 340; classtype: attempted-admin; sid:532; rev:4;)

How it works: Port 139 is the port for NetBIOS file and print sharing service and samba on Windows. This service is often accounts, sometimes admin account, with empty or weak passwords. Attackers attempt to access ADMIN account may send commands containing “\\ADMIN$\00 41 3a 00\” to port 139 on local machine, which would not happen in ordinary file sharing. Check message with such content flow to local server can alert 139 attack attempts.

10.alert ip $EXTERNAL_NET $SHELLCODE_PORTS -> $HOME_NET any (msg: “ SHELLCODE sparc NOOP”; content: “\a61c c013 a61c c013 a61c c013 a61c c013\”; reference:arachnids, 355; classtype: shellcode-detect; sid:646; rev:4;)

How it works: Attackers may send network packets aimed at overflowing Sparc Solaris/SunOS network services and gain root access. A specific string is commonly used during the overflow

which could be detected by the bit combination indicated in this rule. Thus checking the packets sent to local server with the content pattern will reveal such overflow attack.

Question 7: How hard was this lab? Was it fair? How would you change it to improve it?
Answer: This lab is easy but enough for students who are not going to be professional security engineer. It is quite fair. I would suggest adding more practice on deployment and operations of Snort software on top of current tasks focusing on rules.