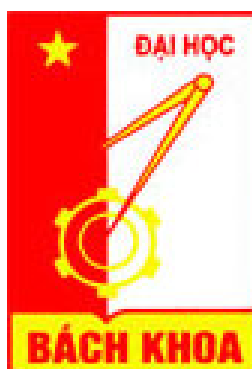


TRƯỜNG ĐẠI HỌC BÁCH KHOA HÀ NỘI
VIỆN CÔNG NGHỆ THÔNG TIN VÀ TRUYỀN THÔNG



Báo cáo Project III

ĐỀ TÀI: HỆ PHÁT HIỆN XÂM NHẬP

Giảng viên hướng dẫn : TS. Đỗ Quốc Huy

Sinh viên thực hiện : Trần Thị Thúy - 20173394

Hà Nội, 12 – 2020

MỤC LỤC

I.	Tổng quan về IDS/IPS	3
II.	Giới thiệu về SNORT	3
III.	Kiến trúc SNORT	3
	3.1. Packet decoder	4
	3.2. Preprocessors	4
	3.3. Detection Engine	4
	3.4. Logging and Alerting System.....	4
	3.5. Output module	4
IV.	Cài đặt.....	4
V.	Sử dụng Snort	6
VI.	Cách lập luật.....	9
	6.1. Rule header	10
	6.2. Rule option	10
	6.2.1. General rule options: cung cấp thông tin về luật nhưng ko gây ảnh hưởng gì đến quá trình phát hiện	10
	6.2.2. Payload rule options: tìm kiếm thông tin trong phần payload của gói tin	10
	6.2.3. Non-payload detection rule options.....	11
	6.2.4. Post-detection rule options	11
VII.	Thử nghiệm tấn công XSS, SQLinjection.....	11
VIII.	Sử dụng snort phát hiện tấn công DDOS.....	12
IX.	Thử nghiệm bắt gói tin trên app chat	14
X.	Viết công cụ tự động sinh luật.....	15
	10.1. Xây dựng database:	15
	10.2. Chức năng đăng nhập	16
	10.3. Chức năng tạo và ghi rules chi tiết	17
	10.4. Chức năng thêm luật mới với content mới	18
	10.5. Chức năng xem nhật ký	19
	Phụ lục: inline mode	20
	Danh mục tài liệu tham khảo.....	23

I. Tổng quan về IDS/IPS

IDS (Intrusion Detection System) hay hệ phát hiện xâm nhập là thiết bị hoặc ứng dụng phần mềm có khả năng giám sát mạng hoặc hệ thống để tìm ra các hành động độc hại hoặc vi phạm chính sách. Bất kỳ hoạt động xâm nhập nào thường sẽ được báo cáo cho quản trị viên hoặc được thu thập tập trung bằng cách sử dụng hệ thống giám sát an ninh mạng (SIEM).

IPS (Intrusion Prevention Systems) là hệ thống theo dõi, ngăn ngừa kịp thời các hoạt động xâm nhập không mong muốn, nó kết hợp với firewall để dừng ngay các hoạt động này sau đó đưa ra cảnh báo chi tiết về các hoạt động đó.

Phân loại IDS: thường được chia thành 2 loại :

- Network-based IDS (NIDS): NIDS thường được cài đặt cho việc kiểm tra giám sát, phát hiện xâm nhập cho một mạng
- Host-based IDS (HIDS): thường được sử dụng để kiểm tra dữ liệu cho một máy trạm đơn, một thiết bị trong mạng.

II. Giới thiệu về SNORT

Snort là hệ thống ngăn chặn xâm nhập mã nguồn mở hàng đầu thế giới. nó sử dụng tập luật để xác định các hoạt động mạng độc hại và sử dụng các luật này để tìm các gói tin phù hợp và đưa ra các cảnh báo cho người dùng. Snort có thể được triển khai nội tuyến để dừng các gói tin này.

Snort có thể được cấu hình để sử dụng trong các chế độ:

- Sniffer mode: chỉ đơn giản là snort sẽ đọc các gói tin đi qua mạng và hiển thị nó theo 1 dòng liên tục trên màn hình console.
- Packet Logger mode: ở chế độ này, Snort sẽ ghi log các gói tin đi qua mạng vào đĩa.
- Network Intrusion Detection System (NIDS) mode: phát hiện và phân tích traffic mạng. Đây là chế độ phức tạp nhất và có thể cấu hình được.
- Inline mode: một plug-in của snort, kết hợp với iptables để xử lý các gói tin vi phạm.

III. Kiến trúc SNORT

Snort bao gồm nhiều thành phần với các chức năng riêng:

- Packet decoder – bộ giải mã gói tin
- Preprocessors – bộ tiền xử lý

- Detection Engine – bộ phận phát hiện
- Logging and Alerting System – bộ phận ghi nhật kí và cảnh báo
- Output module – đầu ra

3.1. Packet decoder

Giải mã là tiến trình đầu tiên mà 1 gói tin khi đi qua Snort. Bộ phận giải mã có nhiệm vụ xác định các giao thức nào được sử dụng trong gói tin (như Ethernet, IP, TCP, ..) và lưu lại các dữ liệu này cùng với vị trí của payload trong gói và kích thước payload phục vụ cho bộ tiền xử lý và bộ phận phát hiện.

Khi bộ giải mã duyệt qua tiêu đề gói tin, nó tìm kiếm lỗi hoặc các bất thường bên trong các trường của tiêu đề; nếu được cấu hình trong file snort.conf, có thể được cảnh báo và thậm chí loại bỏ khi Snort đang chạy ở chế độ inline mode.

Sau khi dữ liệu được giải mã sẽ được chuyển đến bộ phận tiền xử lý.

3.2. Preprocessors

Bộ tiền xử lý được viết như là các plug-in tạo ra cho snort khả năng mở rộng linh hoạt, có thể cấu hình trên từng máy. Nó mang đến khả năng xử lý dữ liệu trải dài trên nhiều gói dữ liệu. snort sử dụng tiền xử lý để chuẩn hóa dữ liệu trong từng loại giao thức, và để phát hiện các dấu hiệu dị thường bằng cách tìm trong phần tiêu đề của gói tin và tạo ra các cảnh báo. Bộ tiền xử lý rất quan trọng với bất kỳ hệ thống IDS nào để chuẩn bị dữ liệu cần thiết về gói tin để bộ phận phát hiện làm việc. Bộ tiền xử lý còn dùng để tái hợp gói tin cho các gói tin có kích thước lớn.

3.3. Detection Engine

Đây là module quan trọng nhất trong kiến trúc của Snort. Nhiệm vụ của nó là phát hiện các dấu hiệu tấn công bằng cách sử dụng các rules để đối chiếu với nội dung gói tin. Nếu gói tin trùng với dấu hiệu trong rule thì snort sẽ đưa ra những hành động thích hợp.

3.4. Logging and Alerting System

Ghi lại những gói tin được phát hiện bất thường khi đi qua bộ phận detection engine, mặc định lưu trong thư mục /var/log/snort/ và đưa ra cảnh báo tương ứng.

3.5. Output module

Module này có chức năng ghi lại log theo định dạng mà người dùng muốn

IV. Cài đặt

Môi trường: máy ảo Ubuntu 18.04

Snort có trong Repository trên Ubuntu nên ta sẽ cài đặt thông qua Repository:

- Cài đặt thư viện DAQ

```
apt install libdaq-dev libdaq2
```

- Cài đặt snort:

```
apt install snort
```

- Cấu hình file snort.conf:

```
nano /etc/snort/snort.conf
```

sửa ipvar HOME_NET any thành ipvar HOME_NET 192.168.94.132 (ip máy cài snort)

```
root@ubuntu1804:~# ifconfig
ens33: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
    inet 192.168.94.132 netmask 255.255.255.0 broadcast 192.168.94.255
    inet6 fe80::dd36:bf07:fbd9:90c4 prefixlen 64 scopeid 0x20<link>
    ether 00:0c:29:14:e1:d5 txqueuelen 1000 (Ethernet)
    RX packets 1902 bytes 1912744 (1.9 MB)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 744 bytes 71801 (71.8 KB)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0
```

Comment các luật sẵn có trong file snort.conf, trừ file local.rules

```
# site specific rules
include $RULE_PATH/local.rules

# The include files commented below have been disabled
# because they are not available in the stock Debian
# rules. If you install the Sourcefire VRT please make
# sure you re-enable them again:

#include $RULE_PATH/app-detect.rules
#include $RULE_PATH/attack-responses.rules
#include $RULE_PATH/backdoor.rules
#include $RULE_PATH/bad-traffic.rules
#include $RULE_PATH/blacklist.rules
#include $RULE_PATH/botnet-cnc.rules
#include $RULE_PATH/browser-chrome.rules
#include $RULE_PATH/browser-firefox.rules
#include $RULE_PATH/browser-ie.rules
#include $RULE_PATH/browser-other.rules
#include $RULE_PATH/browser-plugins.rules
#include $RULE_PATH/browser-webkit.rules
#include $RULE_PATH/chat.rules
#include $RULE_PATH/content-replace.rules
#include $RULE_PATH/ddos.rules
#include $RULE_PATH/dns.rules
#include $RULE_PATH/dos.rules
#include $RULE_PATH/experimental.rules
#include $RULE_PATH/exploit-kit.rules
#include $RULE_PATH/exploit.rules
#include $RULE_PATH/file-executable.rules
#include $RULE_PATH/file-flash.rules
#include $RULE_PATH/file-identify.rules
#include $RULE_PATH/file-image.rules
#include $RULE_PATH/file-multimedia.rules
#include $RULE_PATH/file-office.rules
```

V. Sử dụng Snort

- Sniffer mode:

Snort lắng nghe trên mạng và hiển thị các gói tin lên màn hình, khi dùng ở chế độ này ta không cần đến file snort.conf.

Hiển thị tiêu đề gói tin IP, TCP/UDP/ICMP lên màn hình: `snort -v`

```

root@ubuntu1804:~# snort -v
Running in packet dump mode

--== Initializing Snort ==--
Initializing Output Plugins!
pcap DAQ configured to passive.
Acquiring network traffic from "ens33".
Decoding Ethernet

--== Initialization Complete ==--

o"_)~
'''  -*> Snort! <*-
      Version 2.9.7.0 GRE (Build 149)
      By Martin Roesch & The Snort Team: http://www.snort.org/contact#team
      Copyright (C) 2014 Cisco and/or its affiliates. All rights reserved.
      Copyright (C) 1998-2013 Sourcefire, Inc., et al.
      Using libpcap version 1.8.1
      Using PCRE version: 8.39 2016-06-14
      Using ZLIB version: 1.2.11

Commencing packet processing (pid=4705)

```

[illegible]

Sử dụng thêm tùy chọn `-d`, Snort sẽ hiển thị cả tiêu đề và nội dung của gói tin: `snort -vd`

```
root@ubuntu1804:~# snort -vd
Running in packet dump mode

--== Initializing Snort ==--
Initializing Output Plugins!
pcap DAQ configured to passive.
Acquiring network traffic from "ens33".
Decoding Ethernet

--== Initialization Complete ==--

,,-  -*> Snort! <*-
o"_)~ Version 2.9.7.0 GRE (Build 149)
' ' By Martin Roesch & The Snort Team: http://www.snort.org/contact#team
    Copyright (C) 2014 Cisco and/or its affiliates. All rights reserved.
    Copyright (C) 1998-2013 Sourcefire, Inc., et al.
    Using libpcap version 1.8.1
    Using PCRE version: 8.39 2016-06-14
    Using ZLIB version: 1.2.11

Commencing packet processing (pid=4715)

=====
02/20-04:56:31.653865 192.168.94.132:50982 -> 35.222.85.5:80
TCP TTL:64 TOS:0x0 ID:29544 IpLen:20 DgmLen:40 DF
***A*** Seq: 0x16429F0F Ack: 0x70817C85 Win: 0xFAF0 TcpLen: 20

=====
02/20-04:56:31.654237 192.168.94.132:50982 -> 35.222.85.5:80
TCP TTL:64 TOS:0x0 ID:29545 IpLen:20 DgmLen:127 DF
***AP*** Seq: 0x16429F0F Ack: 0x70817C85 Win: 0xFAF0 TcpLen: 20
47 45 54 20 2F 20 48 54 54 50 2F 31 2E 31 0D 0A GET / HTTP/1.1..
48 6F 73 74 3A 20 63 6F 6E 6E 65 63 74 69 76 69 Host: connectivi
74 79 2D 63 68 65 63 6B 2E 75 62 75 6E 74 75 2E ty-check.ubuntu.
63 6F 6D 0D 0A 41 63 63 65 70 74 3A 20 2A 2F 2A com..Accept: /*
0D 0A 43 6F 6E 6E 65 63 74 69 6F 6E 3A 20 63 6C ..Connection: cl
6F 73 65 0D 0A 0D 0A ose....

=====
```


snort -vde : hiển thị nội dung gói tin và tiêu đề gói tin tầng IP, datalink

```
=====
WARNING: No preprocessors configured for policy 0.
02/20-04:57:35.028195 00:50:56:EC:AB:A1 -> 00:0C:29:14:E1:D5 type:0x800 len:0x11
2
34.107.221.82:80 -> 192.168.94.132:50586 TCP TTL:128 TOS:0x0 ID:14658 IpLen:20 D
gmLen:260
***AP*** Seq: 0x1FCD653B Ack: 0x2A405D7E Win: 0xFAF0 TcpLen: 20
48 54 54 50 2F 31 2E 31 20 32 30 30 20 4F 4B 0D HTTP/1.1 200 OK.
0A 53 65 72 76 65 72 3A 20 6E 67 69 6E 78 0D 0A .Server: nginx..
44 61 74 65 3A 20 54 75 65 2C 20 32 34 20 4E 6F Date: Tue, 24 No
76 20 32 30 32 30 20 31 35 3A 31 34 3A 33 34 20 v 2020 15:14:34
47 4D 54 0D 0A 43 6F 6E 74 65 6E 74 2D 54 79 70 GMT..Content-Typ
65 3A 20 74 65 78 74 2F 70 6C 61 69 6E 0D 0A 43 e: text/plain..C
6F 6E 74 65 6E 74 2D 4C 65 6E 67 74 68 3A 20 38 ontent-Length: 8
0D 0A 56 69 61 3A 20 31 2E 31 20 67 6F 6F 67 6C ..Via: 1.1 googl
65 0D 0A 41 67 65 3A 20 36 37 34 30 31 0D 0A 43 e..Age: 67401..C
61 63 68 65 2D 43 6F 6E 74 72 6F 6C 3A 20 70 75 ache-Control: pu
62 6C 69 63 2C 20 6D 75 73 74 2D 72 65 76 61 6C blic, must-reval
69 64 61 74 65 2C 20 6D 61 78 2D 61 67 65 3D 30 idate, max-age=0
2C 20 73 2D 6D 61 78 61 67 65 3D 38 36 34 30 30 , s-maxage=86400
0D 0A 0D 0A 73 75 63 63 65 73 73 0A ....success.
=====
```

- Packet logger mode

Ở chế độ này, Snort lưu lại toàn bộ các gói tin mà nó thấy trong thư mục được chỉ định

snort -vd -l ./log -b

lựa chọn -b thường được sử dụng để ghi log dạng nhị phân nhằm tăng tốc cho quá trình ghi log

- IDS mode

Snort sử dụng ở chế độ IDS sử dụng đến file snort.conf (đã được đề cập bên trên). Trong file này, như được cấu hình bên trên cần các luật nằm trong file nằm trong thư mục /etc/snort/rules/. Trong demo này, snort.conf chỉ khai báo file local.rules.

Thử chạy với luật sau để kiểm tra có máy đang ping đến máy snort:

```
GNU nano 2.9.3 /etc/snort/rules/local.rules
$Id: local.rules,v 1.11 2004/07/23 20:15:44 bmc Exp $
# -----
# LOCAL RULES
# -----
# This file intentionally does not come with signatures. Put your local
# additions here.
alert icmp any any -> 192.168.94.132 any (msg:"someone ping"; sid:1;)
```

Trên máy Snort: Chạy snort với câu lệnh:

```
snort -A console -c /etc/snort/snort.conf
```

Trên máy kali cùng mạng, ping sang máy Snort:

```
root@kali:~/Desktop# ping 192.168.94.132
PING 192.168.94.132 (192.168.94.132) 56(84) bytes of data.
64 bytes from 192.168.94.132: icmp_seq=1 ttl=64 time=2.54 ms
64 bytes from 192.168.94.132: icmp_seq=2 ttl=64 time=1.04 ms
64 bytes from 192.168.94.132: icmp_seq=3 ttl=64 time=0.869 ms
64 bytes from 192.168.94.132: icmp_seq=4 ttl=64 time=0.838 ms
64 bytes from 192.168.94.132: icmp_seq=5 ttl=64 time=0.748 ms
64 bytes from 192.168.94.132: icmp_seq=6 ttl=64 time=0.813 ms
64 bytes from 192.168.94.132: icmp_seq=7 ttl=64 time=0.922 ms
64 bytes from 192.168.94.132: icmp_seq=8 ttl=64 time=0.837 ms
64 bytes from 192.168.94.132: icmp_seq=9 ttl=64 time=0.452 ms
64 bytes from 192.168.94.132: icmp_seq=10 ttl=64 time=0.827 ms
64 bytes from 192.168.94.132: icmp_seq=11 ttl=64 time=0.864 ms
64 bytes from 192.168.94.132: icmp_seq=12 ttl=64 time=0.567 ms
64 bytes from 192.168.94.132: icmp_seq=13 ttl=64 time=0.824 ms
```

```
Commencing packet processing (pid=17474)
02/20-04:59:34.386296  [**] [1:1:0] someone ping [**] [Priority: 0] {ICMP} 192.1
68.94.131 -> 192.168.94.132
02/20-04:59:35.389371  [**] [1:1:0] someone ping [**] [Priority: 0] {ICMP} 192.1
68.94.131 -> 192.168.94.132
02/20-04:59:36.391509  [**] [1:1:0] someone ping [**] [Priority: 0] {ICMP} 192.1
68.94.131 -> 192.168.94.132
02/20-04:59:37.392651  [**] [1:1:0] someone ping [**] [Priority: 0] {ICMP} 192.1
68.94.131 -> 192.168.94.132
02/20-04:59:38.409191  [**] [1:1:0] someone ping [**] [Priority: 0] {ICMP} 192.1
68.94.131 -> 192.168.94.132
02/20-04:59:39.433930  [**] [1:1:0] someone ping [**] [Priority: 0] {ICMP} 192.1
68.94.131 -> 192.168.94.132
02/20-04:59:40.457324  [**] [1:1:0] someone ping [**] [Priority: 0] {ICMP} 192.1
68.94.131 -> 192.168.94.132
02/20-04:59:41.459293  [**] [1:1:0] someone ping [**] [Priority: 0] {ICMP} 192.1
68.94.131 -> 192.168.94.132
```

Màn hình sẽ hiện ra các cảnh báo nếu có gói tin nào phù hợp với các luật trong local.rules, đồng thời các gói tin đó được ghi log trong thư mục mặc định là /var/log/snort/ , tên file được run tự động theo cấu trúc snort.log.{một dãy số}

```
root@ubuntu1804:/var/log/snort# ls
snort.log  snort.log.1606290921
```

- Inline mode: được trình bày trong phần phụ lục.

VI. Cách lập luật

Luật của Snort thường được viết trên một dòng, gồm 2 phần: rule header và rule options

Rule header bao gồm các nội dung: hành động, giao thức sử dụng, địa chỉ IP và port của nguồn và đích.

Rule options gồm nội dung cảnh báo và thông tin chứa trong các phần của gói tin có thể xác định liệu hành động của luật đó có được diễn ra hay không.

6.1. Rule header

- Hành động: nói cho Snort biết phải làm gì khi một gói tin khớp với luật, có 6 hành động:
 - alert: đưa ra cảnh báo và log gói tin
 - log: ghi log gói tin
 - pass: bỏ qua gói tin
 - drop: chặn và log gói tin
 - reject : chặn gói tin ghi log đồng thời gửi TCP reset hoặc thông báo “unreachable” tùy theo gói tin đó có giao thức gì
 - sdrop: chặn gói tin và không ghi log.
- Giao thức: IP, TCP, UDP, ICMP
- Địa chỉ IP và cổng

6.2. Rule option

Bốn phân loại chính cho rule options là:

- General
- Payload
- Non-payload
- post-detection

6.2.1. General rule options: cung cấp thông tin về luật nhưng ko gây ảnh hưởng gì đến quá trình phát hiện

- msg:”<message text>”;
- reference:<id system>,<id> : tài liệu tham khảo về dấu hiệu tấn công được sử dụng
sid : mỗi luật có một sid, nếu sid bị trùng, snort chỉ lấy luật sau cùng
- rev : số phiên bản chỉnh sửa
- classtype: các phân loại của các dạng tấn công kèm theo mức độ nghiêm trọng của tấn công đó (high, medium, low, very low)
- priority: mức độ nghiêm trọng của dạng tấn công được phát hiện.
- metadata: thông tin thêm về luật

6.2.2. Payload rule options: tìm kiếm thông tin trong phần payload của gói tin

Một số từ khóa:

- content: nội dung có thể tìm kiếm được trong gói tin, mà nếu được tìm thấy, hành động sẽ được thực hiện
- rawbytes: cho phép kiểm tra nội dung gói tin thô, chưa được giải mã bởi preprocessors
- offset: thường đi với content biểu thị vị trí bắt đầu tìm kiếm nội dung
- nocase: kiểm tra phần nội dung nhưng không phân biệt chữ hoa chữ thường.

6.2.3. Non-payload detection rule options

Một số từ khóa:

- ttl: kiểm tra giá trị time to live của gói tin ip
- dsize: kiểm tra kích thước payload
- flags: kiểm tra trường flags bit của gói tin
- seq: kiểm tra giá trị SEQ của gói tin TCP
- ack: kiểm tra giá trị ACK của gói tin tcp

6.2.4. Post-detection rule options

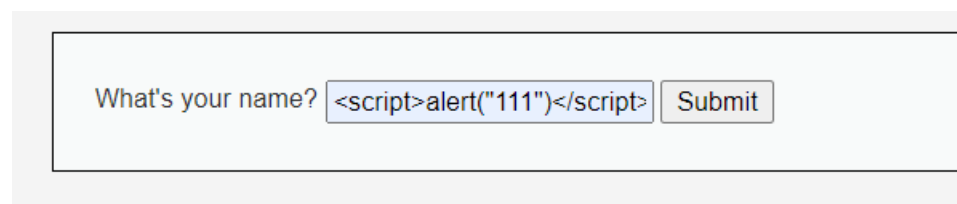
Là hành động đi kèm nếu một rules được kích hoạt: gồm các từ khóa như logto, session, tag, replace, resp, detection_filter, ...

VII. Thử nghiệm tấn công XSS, SQLinjection

Sử dụng DVWA để thử nghiệm phát hiện tấn công XSS, SQLinjection:

Cài đặt DVWA trên máy cài snort; truy cập ip 192.168.94.132 bằng máy thật window10; đặt lại DVWA Security về low. Giả sử đơn giản nhất ta có luật:

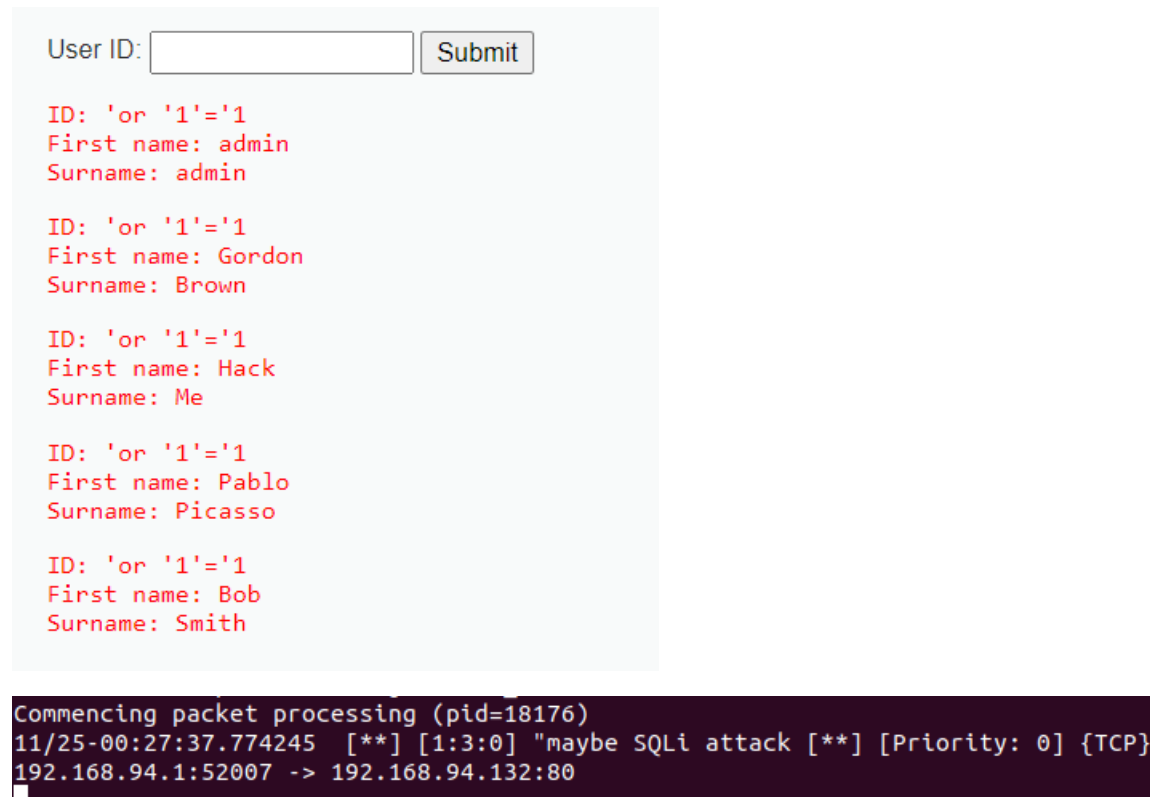
alert tcp any any -> 192.168.94.132 any (msg:"maybe XSS attack"; content:"script"; content:"alert"; nocase; sid:2;) dùng phát hiện tấn công XSS



```
Commencing packet processing (pid=17553)
02/20-05:09:43.191827  [**] [1:2:0] maybe XSS attack [**] [Priority: 0] {TCP} 19
2.168.94.1:51825 -> 192.168.94.132:80
02/20-05:09:53.213536  [**] [1:2:0] maybe XSS attack [**] [Priority: 0] {TCP} 19
2.168.94.1:51827 -> 192.168.94.132:80
```

Ví dụ luật phát hiện tấn công SQLInject:

alert tcp any any -> 192.168.94.132 any (msg:"co the la tan cong sqli"; content:"or"; nocase; sid:3;)



User ID:

ID: 'or '1'='1
First name: admin
Surname: admin

ID: 'or '1'='1
First name: Gordon
Surname: Brown

ID: 'or '1'='1
First name: Hack
Surname: Me

ID: 'or '1'='1
First name: Pablo
Surname: Picasso

ID: 'or '1'='1
First name: Bob
Surname: Smith

```
Commencing packet processing (pid=18176)
11/25-00:27:37.774245  [**] [1:3:0] "maybe SQLi attack [**] [Priority: 0] {TCP}
192.168.94.1:52007 -> 192.168.94.132:80
```

VIII. Sử dụng snort phát hiện tấn công DDOS

Trên máy kali, sử dụng metasploit tấn công SYN Flood đến máy ubuntu cài snort:

Mở metasploit với lệnh: **msfconsole**

Để bắt đầu, phải khai báo: **msf>use auxiliary/dos/tcp/synflood**

Sau đó thiết lập các giá trị RHOST, RPORT là địa chỉ IP và cổng tương ứng của máy nạn nhân:

set RHOST 192.168.94.133

set RPORT 22

Để tấn công, sử dụng lệnh: **exploit**

```

msf5 > use auxiliary/dos/tcp/synflood
msf5 auxiliary(dos/tcp/synflood) > set RHOST 192.168.94.133
RHOST => 192.168.94.133
msf5 auxiliary(dos/tcp/synflood) > set RPORT 22
RPORT => 22
msf5 auxiliary(dos/tcp/synflood) > exploit
[*] Running module against 192.168.94.133

[*] SYN flooding 192.168.94.133:22 ...

```

Có thể theo dõi traffic bằng wireshark:

No.	Time	Source	Destination	Protocol	Length	Info
41151	8.951077616	12.162.207.27	192.168.94.133	TCP	60	37819 → 22 [RST] Seq=1 Wi
41152	8.951670613	12.162.207.27	192.168.94.133	TCP	60	40023 → 22 [SYN] Seq=0 Wi
41153	8.951681641	192.168.94.133	12.162.207.27	TCP	58	22 → 40023 [SYN, ACK] Seq=
41154	8.951798677	12.162.207.27	192.168.94.133	TCP	60	40023 → 22 [RST] Seq=1 Wi
41155	8.952409945	12.162.207.27	192.168.94.133	TCP	60	27156 → 22 [SYN] Seq=0 Wi
41156	8.952421872	192.168.94.133	12.162.207.27	TCP	58	22 → 27156 [SYN, ACK] Seq=
41157	8.952514219	12.162.207.27	192.168.94.133	TCP	60	27156 → 22 [RST] Seq=1 Wi
41158	8.952974389	12.162.207.27	192.168.94.133	TCP	60	[TCP Port numbers reused]
41159	8.952985179	192.168.94.133	12.162.207.27	TCP	58	22 → 14065 [SYN, ACK] Seq=
41160	8.953061937	12.162.207.27	192.168.94.133	TCP	60	14065 → 22 [RST] Seq=1 Wi
41161	8.953647664	12.162.207.27	192.168.94.133	TCP	60	7076 → 22 [SYN] Seq=0 Win=
41162	8.953659081	192.168.94.133	12.162.207.27	TCP	58	22 → 7076 [SYN, ACK] Seq=
41163	8.953754856	12.162.207.27	192.168.94.133	TCP	60	7076 → 22 [RST] Seq=1 Win=
41164	8.954263672	12.162.207.27	192.168.94.133	TCP	60	7301 → 22 [SYN] Seq=0 Win=
41165	8.954274819	192.168.94.133	12.162.207.27	TCP	58	22 → 7301 [SYN, ACK] Seq=
41166	8.954375531	12.162.207.27	192.168.94.133	TCP	60	7301 → 22 [RST] Seq=1 Win=
41167	8.954833589	12.162.207.27	192.168.94.133	TCP	60	20509 → 22 [SYN] Seq=0 Wi
41168	8.954844474	192.168.94.133	12.162.207.27	TCP	58	22 → 20509 [SYN, ACK] Seq=
41169	8.954909192	12.162.207.27	192.168.94.133	TCP	60	20509 → 22 [RST] Seq=1 Wi
41170	8.955448958	12.162.207.27	192.168.94.133	TCP	60	3589 → 22 [SYN] Seq=0 Win=
41171	8.955459851	192.168.94.133	12.162.207.27	TCP	58	22 → 3589 [SYN, ACK] Seq=
41172	8.955541858	12.162.207.27	192.168.94.133	TCP	60	3589 → 22 [RST] Seq=1 Win=

Frame 1: 60 bytes on wire (480 bits), 60 bytes captured (480 bits) on interface 0			
Ethernet II, Src: Vmware_a1:8d:74 (00:0c:29:a1:8d:74), Dst: Vmware_14:e1:d5 (00:0c:29:14:e1:d5)			
Internet Protocol Version 4, Src: 12.162.207.27, Dst: 192.168.94.133			
Transmission Control Protocol, Src Port: 1904, Dst Port: 22, Seq: 0, Len: 0			

0000	00 0c 29 14 e1 d5 00 0c	29 a1 8d 74 08 00 45 00	..).).t..E.
0010	00 28 cf 2b 00 00 c1 06	2f b9 0c a2 cf 1b c0 a8	.(+.... /.....
0020	5e 85 07 70 00 16 c5 ed	db 99 00 00 00 00 50 02	^..p....P.
0030	07 75 04 75 00 00 00 00	00 00 00 00	..u..u....

Tại máy cài snort, viết luật phát hiện tấn công Synflood:

Luật:

```

alert tcp any any -> 192.168.94.133 22 (msg: "co the la tan con ddos syn flood", flags: S;
threshold: type threshold, track by_dst, count 900, seconds 1; sid: 1000005;)

```

Ý nghĩa của luật trên: khi các gói tin hướng đến đích là 192.168.94.133:22, nếu các gói tin có cờ là SYN, trong 1s có ≥ 900 gói tin thì sẽ alert 1 lần với message là “co the la tan con ddos syn flood”.

Khởi chạy snort: `snort -A console -c /etc/snort/snort.conf`

Kết quả:

```
Preprocessor Object: snort version 1.1 (built 11/29/09)
Commencing packet processing (pid=1989)
11/29-09:07:30.633512  ** [1:1000004:0] co the la tan cong ddos syn flood **
[Priority: 0] {TCP} 172.174.165.167:21320 -> 192.168.94.133:22
11/29-09:07:31.517436  ** [1:1000004:0] co the la tan cong ddos syn flood **
[Priority: 0] {TCP} 172.174.165.167:22007 -> 192.168.94.133:22
11/29-09:07:32.539769  ** [1:1000004:0] co the la tan cong ddos syn flood **
[Priority: 0] {TCP} 172.174.165.167:2760 -> 192.168.94.133:22
11/29-09:07:33.525109  ** [1:1000004:0] co the la tan cong ddos syn flood **
[Priority: 0] {TCP} 172.174.165.167:28774 -> 192.168.94.133:22
11/29-09:07:34.537225  ** [1:1000004:0] co the la tan cong ddos syn flood **
[Priority: 0] {TCP} 172.174.165.167:61401 -> 192.168.94.133:22
```

IX. Thử nghiệm bắt gói tin trên app chat

Bằng việc xây dựng một app chat đơn giản để demo bắt gói tin với snort, đứng trên 2 máy gồm 1 máy thật window 10 địa chỉ IP: 192.168.94.1 và máy ảo ubuntu có cài snort IP: 192.168.94.133, truy cập app chat và nhắn tin qua lại, ta xây dựng rule phát hiện:

```
alert tcp 192.168.94.1 8080 <> 192.168.94.133 any (msg:"chatting", sid=1000006;)
```

Kết quả:


```

Commencing packet processing (pid=3014)
11/23-22:15:38.528500 00:0C:29:14:E1:D5 -> 00:50:56:C0:00:08 type:0x800 len:0x106
192.168.94.133:48582 -> 192.168.94.1:8080 TCP TTL:64 TOS:0x0 ID:17808 IpLen:20 DgmLen:456 DF
***AP*** Seq: 0x780D11C1 Ack: 0x8CE47A16 Win: 0x975 TcpLen: 20
47 45 54 20 2F 6D 65 73 73 65 6E 67 65 72 2F 6D GET /messenger/m
73 67 6C 6F 67 2E 70 68 70 3F 5F 3D 31 36 30 36 sglog.php?_=1606
37 32 30 33 33 33 30 36 31 20 48 54 54 50 2F 31 720333061 HTTP/1
2E 31 0D 0A 48 6F 73 74 3A 20 31 39 32 2E 31 36 .1..Host: 192.16
38 2E 39 34 2E 31 3A 38 30 38 30 0D 0A 55 73 65 8.94.1:8080..Use
72 2D 41 67 65 6E 74 3A 20 4D 6F 7A 69 6C 6C 61 r-Agent: Mozilla
2F 35 2E 30 20 28 58 31 31 3B 20 55 62 75 6E 74 /5.0 (X11; Ubunt
75 3B 20 4C 69 6E 75 78 20 78 38 36 5F 36 34 3B u; Linux x86_64;
20 72 76 3A 38 33 2E 30 29 20 47 65 63 6B 6F 2F rv:83.0) Gecko/
32 30 31 30 30 31 30 31 20 46 69 72 65 66 6F 78 20100101 Firefox
2F 38 33 2E 30 0D 0A 41 63 63 65 70 74 3A 20 74 /83.0..Accept: t
65 78 74 2F 68 74 6D 6C 2C 20 2A 2F 2A 3B 20 71 ext/html, */*; q
3D 30 2E 30 31 0D 0A 41 63 63 65 70 74 2D 4C 61 =0.01..Accept-La
6E 67 75 61 67 65 3A 20 65 6E 2D 55 53 2C 65 6E nguage: en-US,en
3B 71 3D 30 2E 35 0D 0A 41 63 63 65 70 74 2D 45 ;q=0.5..Accept-E
6E 63 6F 64 69 6E 67 3A 20 67 7A 69 70 2C 20 64 ncoding: gzip, d
65 66 6C 61 74 65 0D 0A 58 2D 52 65 71 75 65 73 eflate..X-Reques
74 65 64 2D 57 69 74 68 3A 20 58 4D 4C 48 74 74 ted-With: XMLHtt
70 52 65 71 75 65 73 74 0D 0A 43 6F 6E 6E 65 63 pRequest..Connec
74 69 6F 6E 3A 20 6B 65 65 70 2D 61 6C 69 76 65 tion: keep-alive
0D 0A 52 65 66 65 72 65 72 3A 20 68 74 74 70 3A ..Referer: http:
2F 2F 31 39 32 2E 31 36 38 2E 39 34 2E 31 3A 38 //192.168.94.1:8
30 38 30 2F 6D 65 73 73 65 6E 67 65 72 2F 0D 0A 080/messenger/.
43 6F 6F 6B 69 65 3A 20 50 48 50 53 45 53 53 49 Cookie: PHPSESSI
44 3D 73 6F 67 72 62 31 64 70 62 6B 31 6F 6F 36 D=sogrb1dpgbk1oo6
67 75 31 66 69 6C 62 35 38 6C 6A 74 0D 0A 0D 0A guifilb58ljt....

+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+
(snort_decoder) WARNING: IP dgm len > captured len
11/23-22:15:38.546201 00:0C:29:14:E1:D5 -> 00:50:56:C0:00:08 type:0x800 len:0x36
192.168.94.133:48582 -> 192.168.94.1:8080 TCP TTL:64 TOS:0x0 ID:17809 IpLen:20 DamLen:40 DF

```

X. Viết công cụ tự động sinh luật

Xây dựng một chương trình trên nền web, với chức năng chính là thêm các rule vào files local.rules mà snort ids sử dụng.

Môi trường: máy ảo ubuntu 18.04 có cài snort.

Cài đặt môi trường cần thiết cho lập trình web với php:

- cài đặt php7.2
- cài đặt apache
- cài đặt mysql-server

10.1. Xây dựng database:

Việc xây dựng database với mục đích là lưu trữ lại các rules đã tạo trước đó đồng thời là cơ sở để xem rule sắp được tạo ra đã được khai báo trước đó hay chưa.

Truy cập mysql tao csdl với câu lệnh: `mysql -u root`

Tạo mới database: `create database rules;`

Tạo user và cấp quyền truy cập rules:

`grant all privileges on rules.* to app@localhost identified by '123456';`

`flush privileges;`

tạo bảng lưu trữ rule: `use rules;`

`create table rule(`

`sid int primary key,`

`prerule text not null);`

Kết nối database:

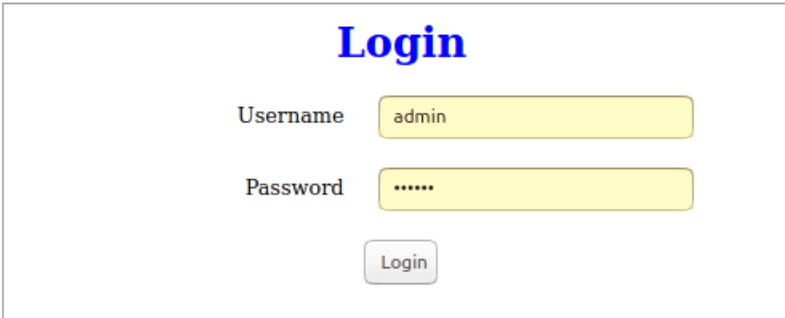
```
<?php
$namehost = 'localhost';
$userhost = 'app';
$passhost = '123456';
$database = 'rules';

// Lệnh kết nối tới database
$cn = mysqli_connect($namehost, $userhost, $passhost, $database);

// Nếu kết nối database thất bại sẽ báo lỗi
if (!$cn) {
    die (mysqli_connect_error());
}
?>
```

10.2. Chức năng đăng nhập

Người dùng sẽ sử dụng một tài khoản admin được cung cấp trước để đăng nhập vào hệ thống. Nếu người dùng đăng nhập sai, hệ thống báo lỗi và yêu cầu đăng nhập lại.



The image shows a login form titled "Login" in blue text. It contains two input fields: "Username" with the value "admin" and "Password" with masked characters "*****". Below the password field is a "Login" button.

10.3. Chức năng tạo và ghi rules chi tiết

Chức năng này được sử dụng khi người dùng muốn viết một luật chi tiết về các giá trị như IP, port, và một số kiểm tra nội dung của gói tin khi trao đổi.

Về mặt giao diện:

Action	<input type="text" value="alert"/>	Protocol	<input type="text" value="icmp"/>
IP1	<input type="text"/>		
Port 1	<input type="text"/>		
Direction	<input type="text" value="->"/>		
IP2	<input type="text"/>		
Port 2	<input type="text"/>		
Message	<input type="text"/>		
Content 1	<input type="text"/>		
Content 2	<input type="text"/>		
NOCASE	<input type="text"/>		
Rawbytes	<input type="text"/>		
Offset	<input type="text"/>		
Flags	<input type="text"/>		
<input type="button" value="ADD"/>			

Thông thường mỗi rule sẽ có 1 sid duy nhất, nếu như có 2 hoặc nhiều rule trùng sid, snort sẽ bỏ đi tất cả các rule này trừ rule cuối cùng.

Nếu ta sử dụng các rule sẵn có của snort, ta có thể thấy được ánh xạ các sid trong file `/etc/snort/gen-msg.map` và thường các rule người dùng tự định nghĩa nên có sid ≥ 1000000 .

Với công cụ này, sau khi người dùng ấn nút ‘ADD’ để tạo rule mới, các nội dung trong form sẽ được ghép lại, lấy những phần quan trọng nhất sau đó bỏ đi các kí tự đặc biệt tạo thành prerule. Kiểm tra xem prerule này đã có trong cơ sở dữ liệu chưa, nếu có rồi, thông

báo cho người, nếu chưa có, thêm prerule vào csdl, sid là giá trị sid lớn nhất trong bảng +1, sau đó ghi thêm rule vào file hệ thống: /etc/snort/rules/local.rules.

Lấy sid:

```
$result= $cn->query("select * from rule order by sid desc");
if ($result->num_rows>0){
$row=$result->fetch_assoc();
$sid= $row['sid']+1;
}
else $sid=1000001;
```

Lưu ý: file /etc/snort/rules/local.rules phân quyền cho “other” chỉ có quyền đọc, nên muốn ghi thêm vào file này, ta phải phân lại quyền cho nó:

```
sudo chmod o+w /etc/snorts/rules/local.rules
```

Code kiểm tra và ghi file:

```
$_prerule=$action." ".$protocol." ".$ip1." ".$port1." ".$direction." ".$ip2." ".$port2." (".$content1.$content2.$nocase.$flags.$rawbyte.$offset.");
$prerule=str_replace("'", "", $_prerule);
$prerule=str_replace("\", "", $prerule);

$check=$cn->query("select * from rule where prerule='$prerule'");
if ($check->num_rows>0){
    echo "rule nay da ton tai";
}else{
    $cn->query("insert into rule values ('$sid', '$prerule')");
    $rule=$action." ".$protocol." ".$ip1." ".$port1." ".$direction." ".$ip2." ".$port2." (.$msg.$content1.$content2.$nocase.$flags.$rawbyte.$offset."sid: ".$sid.";)".$n;
    $f=@fopen('/etc/snort/rules/local.rules', "a+");
    if (!$f){
        echo "ko ghi duoc";
    }else{
        fwrite($f, $rule);
    }
    echo "Da them rule moi la: ".$rule;
}
```

10.4. Chức năng thêm luật mới với content mới

Chức năng này sử dụng khi người dùng muốn kiểm tra phần nội dung của các gói tin trao đổi trên mạng với lại các máy tính được bảo vệ.

Thực hiện thêm rule mới với content là:

Action

drop ▼

content

message

ADD

[Tro lại](#)

Đối với chức năng này, nguyên tắc tạo luật và tạo sid cũng giống như ở chức năng 10.3.

10.5. Chức năng xem nhật ký

Mỗi khi khởi chạy Snort ở chế độ IDS, Snort sẽ tự động lưu lại gói tin mà nó phát hiện được vào thư mục `/var/log/snort`. Chức năng xem nhật ký cho phép người dùng có thể đọc được các lịch sử hoạt động của Snort.

Do Snort ghi nhật ký các gói tin dưới dạng các file pcap nên em sẽ sử dụng tcpdump để đọc nội dung của các file này.

Thư mục `/var/log/snort` là một thư mục được phân quyền khá chặt, các file bên trong nó chỉ có user root có quyền được đọc ghi, vì vậy, cần thiết lập cho user “www-data” được phép sử dụng quyền sudo mà không cần xác thực mật khẩu.

Danh sách file log:

snort.log.1608711778 snort.log

file bạn muốn đọc là:

Hệ thống đưa ra cho người dùng các danh sách file nằm trong thư mục `/var/log/snort`. Người dùng muốn đọc file nào thì chủ động nhập tên file, hệ thống sẽ hiển thị ra nội dung file.

Danh sách file log:

snort.log.1608711778 snort.log

file bạn muốn đọc là:

23:23:43.224130 IP 192.168.94.1 > 192.168.94.133: ICMP echo request, id 1, seq 1, length 40 23:23:44.227732 IP 192.168.94.1 > 192.168.94.133: ICMP echo request, id 1, seq 2, length 40 23:23:45.238594 IP 192.168.94.1 > 192.168.94.133: ICMP echo request, id 1, seq 3, length 40 23:23:46.252767 IP 192.168.94.1 > 192.168.94.133: ICMP echo request, id 1, seq 4, length 40 23:24:35.932206 IP 192.168.94.1 > 192.168.94.133: ICMP echo request, id 1, seq 5, length 40 23:24:35.932206 IP 192.168.94.1 > 192.168.94.133: ICMP echo request, id 1, seq 5, length 40

Phụ lục: Inline mode

1. Cài đặt

Hệ thống thử nghiệm snort IPS gồm 3 máy ảo ubuntu 18.04

Sơ đồ mạng cần cài đặt:

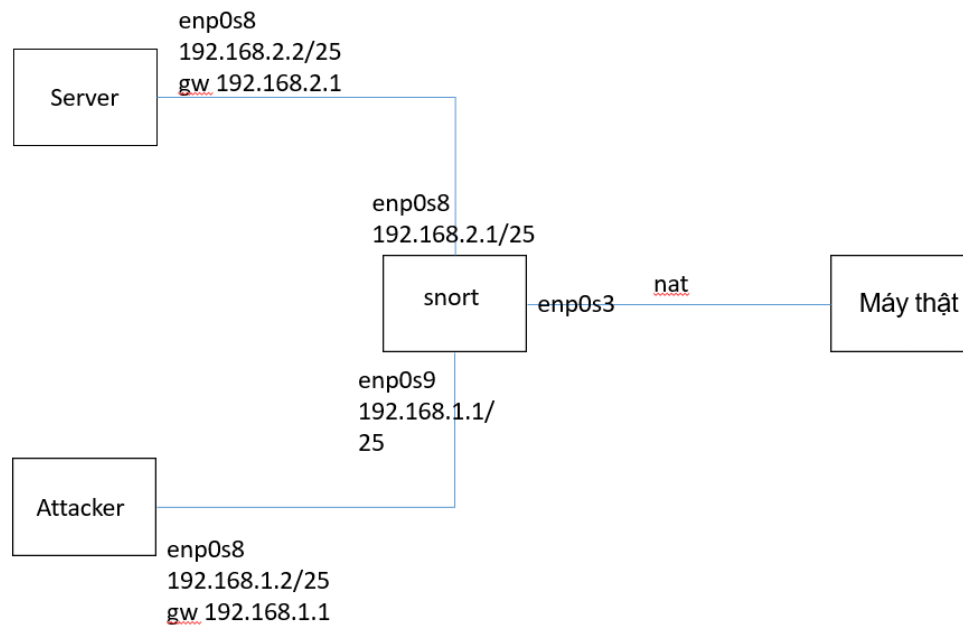


Figure 1 : sơ đồ cài đặt snort inline

Trên máy snort:

- Bật ip forwarding cho phép server và attacker thấy nhau
- Cài đặt thư viện: apt install libdaq-dev libdaq2
- Cài đặt snort: apt install snort

Tại máy server: ip route add 192.168.1.0/25 via 192.168.2.1

Tại máy attacker: ip route add 192.168.2.0/25 via 192.168.1.1

Máy server: ping 192.168.1.2

```
root@ubuntu1804:~# ping 192.168.1.2
PING 192.168.1.2 (192.168.1.2) 56(84) bytes of data.
64 bytes from 192.168.1.2: icmp_seq=1 ttl=63 time=1.57 ms
64 bytes from 192.168.1.2: icmp_seq=2 ttl=63 time=1.75 ms
64 bytes from 192.168.1.2: icmp_seq=3 ttl=63 time=1.06 ms
64 bytes from 192.168.1.2: icmp_seq=4 ttl=63 time=0.889 ms
^C
--- 192.168.1.2 ping statistics ---
4 packets transmitted, 4 received, 0% packet loss, time 3004ms
rtt min/avg/max/mdev = 0.889/1.319/1.752/0.356 ms
```

2. Cấu hình

Sửa lại file snort.conf:

config policy_mode:inline

config daq: afpacket

config daq_mode: inline

config daq_var: buffer_size_mb=1024

comment các dòng khai báo rule trừ file local.rules

viết luật : drop icmp any any -> 192.168.2.2 any (msg:"chan ping server "; sid:1000002;)

chạy lệnh: snort -A console -c /etc/snort/snort.conf -I enp0s9:enp0s8 -Q

trên máy attacker tiến hành ping đến server:

```
root@ubuntu1804:~# ping 192.168.2.2
PING 192.168.2.2 (192.168.2.2) 56(84) bytes of data.
64 bytes from 192.168.2.2: icmp_seq=1 ttl=63 time=0.948 ms
From 192.168.2.2 icmp_seq=1 Destination Port Unreachable
64 bytes from 192.168.2.2: icmp_seq=2 ttl=63 time=1.39 ms
From 192.168.2.2 icmp_seq=2 Destination Port Unreachable
64 bytes from 192.168.2.2: icmp_seq=3 ttl=63 time=1.58 ms
From 192.168.2.2 icmp_seq=3 Destination Port Unreachable
64 bytes from 192.168.2.2: icmp_seq=4 ttl=63 time=2.17 ms
64 bytes from 192.168.2.2: icmp_seq=5 ttl=63 time=1.57 ms
From 192.168.2.2 icmp_seq=5 Destination Port Unreachable
64 bytes from 192.168.2.2: icmp_seq=6 ttl=63 time=1.44 ms
From 192.168.2.2 icmp_seq=6 Destination Port Unreachable
64 bytes from 192.168.2.2: icmp_seq=7 ttl=63 time=1.45 ms
From 192.168.2.2 icmp_seq=7 Destination Port Unreachable
64 bytes from 192.168.2.2: icmp_seq=8 ttl=63 time=1.49 ms
From 192.168.2.2 icmp_seq=8 Destination Port Unreachable
64 bytes from 192.168.2.2: icmp_seq=9 ttl=63 time=1.82 ms
64 bytes from 192.168.2.2: icmp_seq=10 ttl=63 time=1.42 ms
From 192.168.2.2 icmp_seq=10 Destination Port Unreachable
^C
--- 192.168.2.2 ping statistics ---
10 packets transmitted, 10 received, +10 errors, 0% packet loss, time 9024ms
rtt min/avg/max/mdev = 0.948/1.532/2.172/0.301 ms
root@ubuntu1804:~#
```

Trên máy snort ta được kết quả sau:

```
Commencing packet processing (pid=2174)
Decoding Ethernet
11/30-17:06:54.647740 [Drop] [**] [1:1000002:0] chan ping [**] [Priority: 0] {I
CMP} 192.168.1.2 -> 192.168.2.2
11/30-17:06:54.647724 [Drop] [**] [1:1000002:0] chan ping [**] [Priority: 0] {I
CMP} 192.168.1.2 -> 192.168.2.2
11/30-17:06:54.648276 [Drop] [**] [1:1000002:0] chan ping [**] [Priority: 0] {I
CMP} 192.168.2.2 -> 192.168.1.2
11/30-17:06:54.648282 [Drop] [**] [1:1000002:0] chan ping [**] [Priority: 0] {I
CMP} 192.168.2.2 -> 192.168.1.2
11/30-17:06:55.649331 [Drop] [**] [1:1000002:0] chan ping [**] [Priority: 0] {I
CMP} 192.168.1.2 -> 192.168.2.2
11/30-17:06:55.649301 [Drop] [**] [1:1000002:0] chan ping [**] [Priority: 0] {I
CMP} 192.168.1.2 -> 192.168.2.2
11/30-17:06:55.649970 [Drop] [**] [1:1000002:0] chan ping [**] [Priority: 0] {I
CMP} 192.168.2.2 -> 192.168.1.2
11/30-17:06:55.649978 [Drop] [**] [1:1000002:0] chan ping [**] [Priority: 0] {I
CMP} 192.168.2.2 -> 192.168.1.2
11/30-17:06:56.650716 [Drop] [**] [1:1000002:0] chan ping [**] [Priority: 0] {I
CMP} 192.168.1.2 -> 192.168.2.2
```

Danh mục tài liệu tham khảo

1. SNORT Users Manual , The Snort Project, july 10, 2020
2. www.snort.org
3. Snort IPS using DAQ AFPacket
4. <http://vrt-blog.snort.org/2010/08/snort-29-essentials-daq.html>