Chi Zhang & Yuyang Zhang

CS6740 Network Security

Lab report for Network Intrusion Detection

1. Configuration file for Snort

The version we use for this lab is Snort 2.9.9.0. Since in configuration file, there must be at most one interface configuration, we specify the interfaces as command line arguments. The testing command is

sudo snort –T –i eth0 –i eth1 –c /etc/snort/snort.conf

The file and directory need to be created before running is like the following structure:

/etc/snort

|-- \*.dtd

|-- \*.conf

|-- \*.map

|-- rules

| |--iplists

| | |-- black\_list.rules

| | |-- white\_list.rules

|--preproc\_rules

|-- so\_rules

Besides, the following directory should be created and filled with the file downloaded from snort.org

cd~/snort\_src/snort2.9.9.0/src/dynamic\_preprocessors/build/usr/local/lib/snort\_dynamicpreprocessor/

sudo cp \* /usr/local/lib/snort\_dynamicpreprocessor/

These two commands create the log used for the tests:

sudo mkdir /var/log/snort

sudo mkdir /var/log/snort/archived\_logs

After doing these jobs, we could run snort by using our configuration file:

snort.conf

#--------------------------------------------------

# VRT Rule Packages Snort.conf

#

# For more information visit us at:

# http://www.snort.org Snort Website

# http://vrt-blog.snort.org/ Sourcefire VRT Blog

#

# Mailing list Contact: snort-sigs@lists.sourceforge.net

# False Positive reports: fp@sourcefire.com

# Snort bugs: bugs@snort.org

#

# Compatible with Snort Versions:

# VERSIONS : 2.9.9.0

#

# Snort build options:

# OPTIONS : --enable-gre --enable-mpls --enable-targetbased --enable-ppm --enable-perfprofiling --enable-zlib --enable-active-response --enable-normalizer --enable-reload --enable-react --enable-flexresp3

#

# Additional information:

# This configuration file enables active response, to run snort in

# test mode -T you are required to supply an interface -i <interface>

# or test mode will fail to fully validate the configuration and

# exit with a FATAL error

#--------------------------------------------------

###################################################

# This file contains a sample snort configuration.

# You should take the following steps to create your own custom configuration:

#

# 1) Set the network variables.

# 2) Configure the decoder

# 3) Configure the base detection engine

# 4) Configure dynamic loaded libraries

# 5) Configure preprocessors

# 6) Configure output plugins

# 7) Customize your rule set

# 8) Customize preprocessor and decoder rule set

# 9) Customize shared object rule set

###################################################

###################################################

# Step #1: Set the network variables. For more information, see README.variables

###################################################

# Setup the network addresses you are protecting

ipvar HOME\_NET [10.0.100.1/24,192.168.254.15/24]

# Set up the external network addresses. Leave as "any" in most situations

ipvar EXTERNAL\_NET !$HOME\_NET

# List of DNS servers on your network

ipvar DNS\_SERVERS $HOME\_NET

# List of SMTP servers on your network

ipvar SMTP\_SERVERS $HOME\_NET

# List of web servers on your network

ipvar HTTP\_SERVERS $HOME\_NET

# List of sql servers on your network

ipvar SQL\_SERVERS $HOME\_NET

# List of telnet servers on your network

ipvar TELNET\_SERVERS $HOME\_NET

# List of ssh servers on your network

ipvar SSH\_SERVERS $HOME\_NET

# List of ftp servers on your network

ipvar FTP\_SERVERS $HOME\_NET

# List of sip servers on your network

ipvar SIP\_SERVERS $HOME\_NET

# List of ports you run web servers on

portvar HTTP\_PORTS [80,81,311,383,591,593,901,1220,1414,1741,1830,2301,2381,2809,3037,3128,3702,4343,4848,5250,6988,7000,7001,7144,7145,7510,7777,7779,8000,8008,8014,8028,8080,8085,8088,8090,8118,8123,8180,8181,8243,8280,8300,8800,8888,8899,9000,9060,9080,9090,9091,9443,9999,11371,34443,34444,41080,50002,55555]

# List of ports you want to look for SHELLCODE on.

portvar SHELLCODE\_PORTS !80

# List of ports you might see oracle attacks on

portvar ORACLE\_PORTS 1024:

# List of ports you want to look for SSH connections on:

portvar SSH\_PORTS 22

# List of ports you run ftp servers on

portvar FTP\_PORTS [21,2100,3535]

# List of ports you run SIP servers on

portvar SIP\_PORTS [5060,5061,5600]

# List of file data ports for file inspection

portvar FILE\_DATA\_PORTS [$HTTP\_PORTS,110,143]

# List of GTP ports for GTP preprocessor

portvar GTP\_PORTS [2123,2152,3386]

# other variables, these should not be modified

ipvar AIM\_SERVERS [64.12.24.0/23,64.12.28.0/23,64.12.161.0/24,64.12.163.0/24,64.12.200.0/24,205.188.3.0/24,205.188.5.0/24,205.188.7.0/24,205.188.9.0/24,205.188.153.0/24,205.188.179.0/24,205.188.248.0/24]

# Path to your rules files (this can be a relative path)

# Note for Windows users: You are advised to make this an absolute path,

# such as: c:\snort\rules

var RULE\_PATH /etc/snort/rules

var SO\_RULE\_PATH /etc/snort/so\_rules

var PREPROC\_RULE\_PATH /etc/snort/preproc\_rules

# If you are using reputation preprocessor set these

# Currently there is a bug with relative paths, they are relative to where snort is

# not relative to snort.conf like the above variables

# This is completely inconsistent with how other vars work, BUG 89986

# Set the absolute path appropriately

var WHITE\_LIST\_PATH /etc/snort/rules/iplists

var BLACK\_LIST\_PATH /etc/snort/rules/iplists

###################################################

# Step #2: Configure the decoder. For more information, see README.decode

###################################################

# Stop generic decode events:

config disable\_decode\_alerts

# Stop Alerts on experimental TCP options

config disable\_tcpopt\_experimental\_alerts

# Stop Alerts on obsolete TCP options

config disable\_tcpopt\_obsolete\_alerts

# Stop Alerts on T/TCP alerts

config disable\_tcpopt\_ttcp\_alerts

# Stop Alerts on all other TCPOption type events:

config disable\_tcpopt\_alerts

# Stop Alerts on invalid ip options

config disable\_ipopt\_alerts

# Alert if value in length field (IP, TCP, UDP) is greater th elength of the packet

# config enable\_decode\_oversized\_alerts

# Same as above, but drop packet if in Inline mode (requires enable\_decode\_oversized\_alerts)

# config enable\_decode\_oversized\_drops

# Configure IP / TCP checksum mode

config checksum\_mode: all

# Configure maximum number of flowbit references. For more information, see README.flowbits

# config flowbits\_size: 64

# Configure ports to ignore

# config ignore\_ports: tcp 21 6667:6671 1356

# config ignore\_ports: udp 1:17 53

# Configure active response for non inline operation. For more information, see REAMDE.active

# config response: eth0 attempts 2

# Configure DAQ related options for inline operation. For more information, see README.daq

#

# config daq: <type>

# config daq\_dir: <dir>

# config daq\_mode: <mode>

# config daq\_var: <var>

#

# <type> ::= pcap | afpacket | dump | nfq | ipq | ipfw

# <mode> ::= read-file | passive | inline

# <var> ::= arbitrary <name>=<value passed to DAQ

# <dir> ::= path as to where to look for DAQ module so's

# Configure specific UID and GID to run snort as after dropping privs. For more information see snort -h command line options

#

# config set\_gid:

# config set\_uid:

# Configure default snaplen. Snort defaults to MTU of in use interface. For more information see README

#

# config snaplen:

#

# Configure default bpf\_file to use for filtering what traffic reaches snort. For more information see snort -h command line options (-F)

#

# config bpf\_file:

#

# Configure default log directory for snort to log to. For more information see snort -h command line options (-l)

#

# config logdir:

###################################################

# Step #3: Configure the base detection engine. For more information, see README.decode

###################################################

# Configure PCRE match limitations

config pcre\_match\_limit: 3500

config pcre\_match\_limit\_recursion: 1500

# Configure the detection engine See the Snort Manual, Configuring Snort - Includes - Config

config detection: search-method ac-split search-optimize max-pattern-len 20

# Configure the lowmem pattern

config detection: search-method lowmem

# Configure the event queue. For more information, see README.event\_queue

config event\_queue: max\_queue 8 log 5 order\_events content\_length

###################################################

## Configure GTP if it is to be used.

## For more information, see README.GTP

####################################################

# config enable\_gtp

###################################################

# Per packet and rule latency enforcement

# For more information see README.ppm

###################################################

# Per Packet latency configuration

#config ppm: max-pkt-time 250, \

# fastpath-expensive-packets, \

# pkt-log

# Per Rule latency configuration

#config ppm: max-rule-time 200, \

# threshold 3, \

# suspend-expensive-rules, \

# suspend-timeout 20, \

# rule-log alert

###################################################

# Configure Perf Profiling for debugging

# For more information see README.PerfProfiling

###################################################

#config profile\_rules: print all, sort avg\_ticks

#config profile\_preprocs: print all, sort avg\_ticks

###################################################

# Configure protocol aware flushing

# For more information see README.stream5

###################################################

config paf\_max: 16000

###################################################

# Step #4: Configure dynamic loaded libraries.

# For more information, see Snort Manual, Configuring Snort - Dynamic Modules

###################################################

# path to dynamic preprocessor libraries

dynamicpreprocessor directory /usr/local/lib/snort\_dynamicpreprocessor/

# path to base preprocessor engine

dynamicengine /usr/local/lib/snort\_dynamicengine/libsf\_engine.so

# path to dynamic rules libraries

dynamicdetection directory /usr/local/lib/snort\_dynamicrules

###################################################

# Step #5: Configure preprocessors

# For more information, see the Snort Manual, Configuring Snort - Preprocessors

###################################################

# GTP Control Channle Preprocessor. For more information, see README.GTP

# preprocessor gtp: ports { 2123 3386 2152 }

# Inline packet normalization. For more information, see README.normalize

# Does nothing in IDS mode

preprocessor normalize\_ip4

preprocessor normalize\_tcp: ips ecn stream

preprocessor normalize\_icmp4

preprocessor normalize\_ip6

preprocessor normalize\_icmp6

# Target-based IP defragmentation. For more inforation, see README.frag3

preprocessor frag3\_global: max\_frags 65536

preprocessor frag3\_engine: policy windows detect\_anomalies overlap\_limit 10 min\_fragment\_length 100 timeout 180

# Target-Based stateful inspection/stream reassembly. For more inforation, see README.stream5

preprocessor stream5\_global: track\_tcp yes, \

track\_udp yes, \

track\_icmp no, \

max\_tcp 262144, \

max\_udp 131072, \

max\_active\_responses 2, \

min\_response\_seconds 5

preprocessor stream5\_tcp: log\_asymmetric\_traffic no, policy windows, \

detect\_anomalies, require\_3whs 180, \

overlap\_limit 10, small\_segments 3 bytes 150, timeout 180, \

ports client 21 22 23 25 42 53 79 109 110 111 113 119 135 136 137 139 143 \

161 445 513 514 587 593 691 1433 1521 1741 2100 3306 6070 6665 6666 6667 6668 6669 \

7000 8181 32770 32771 32772 32773 32774 32775 32776 32777 32778 32779, \

ports both 80 81 311 383 443 465 563 591 593 636 901 989 992 993 994 995 1220 1414 1830 2301 2381 2809 3037 3128 3702 4343 4848 5250 6988 7907 7000 7001 7144 7145 7510 7802 7777 7779 \

7801 7900 7901 7902 7903 7904 7905 7906 7908 7909 7910 7911 7912 7913 7914 7915 7916 \

7917 7918 7919 7920 8000 8008 8014 8028 8080 8085 8088 8090 8118 8123 8180 8243 8280 8300 8800 8888 8899 9000 9060 9080 9090 9091 9443 9999 11371 34443 34444 41080 50002 55555

preprocessor stream5\_udp: timeout 180

# performance statistics. For more information, see the Snort Manual, Configuring Snort - Preprocessors - Performance Monitor

# preprocessor perfmonitor: time 300 file /var/snort/snort.stats pktcnt 10000

# HTTP normalization and anomaly detection. For more information, see README.http\_inspect

preprocessor http\_inspect: global iis\_unicode\_map unicode.map 1252 compress\_depth 65535 decompress\_depth 65535

preprocessor http\_inspect\_server: server default \

http\_methods { GET POST PUT SEARCH MKCOL COPY MOVE LOCK UNLOCK NOTIFY POLL BCOPY BDELETE BMOVE LINK UNLINK OPTIONS HEAD DELETE TRACE TRACK CONNECT SOURCE SUBSCRIBE UNSUBSCRIBE PROPFIND PROPPATCH BPROPFIND BPROPPATCH RPC\_CONNECT PROXY\_SUCCESS BITS\_POST CCM\_POST SMS\_POST RPC\_IN\_DATA RPC\_OUT\_DATA RPC\_ECHO\_DATA } \

chunk\_length 500000 \

server\_flow\_depth 0 \

client\_flow\_depth 0 \

post\_depth 65495 \

oversize\_dir\_length 500 \

max\_header\_length 750 \

max\_headers 100 \

max\_spaces 200 \

small\_chunk\_length { 10 5 } \

ports { 80 81 311 383 591 593 901 1220 1414 1741 1830 2301 2381 2809 3037 3128 3702 4343 4848 5250 6988 7000 7001 7144 7145 7510 7777 7779 8000 8008 8014 8028 8080 8085 8088 8090 8118 8123 8180 8181 8243 8280 8300 8800 8888 8899 9000 9060 9080 9090 9091 9443 9999 11371 34443 34444 41080 50002 55555 } \

non\_rfc\_char { 0x00 0x01 0x02 0x03 0x04 0x05 0x06 0x07 } \

enable\_cookie \

extended\_response\_inspection \

inspect\_gzip \

normalize\_utf \

unlimited\_decompress \

normalize\_javascript \

apache\_whitespace no \

ascii no \

bare\_byte no \

directory no \

double\_decode no \

iis\_backslash no \

iis\_delimiter no \

iis\_unicode no \

multi\_slash no \

utf\_8 no \

u\_encode yes \

webroot no

# ONC-RPC normalization and anomaly detection. For more information, see the Snort Manual, Configuring Snort - Preprocessors - RPC Decode

preprocessor rpc\_decode: 111 32770 32771 32772 32773 32774 32775 32776 32777 32778 32779 no\_alert\_multiple\_requests no\_alert\_large\_fragments no\_alert\_incomplete

# Back Orifice detection.

preprocessor bo

# FTP / Telnet normalization and anomaly detection. For more information, see README.ftptelnet

preprocessor ftp\_telnet: global inspection\_type stateful encrypted\_traffic no check\_encrypted

preprocessor ftp\_telnet\_protocol: telnet \

ayt\_attack\_thresh 20 \

normalize ports { 23 } \

detect\_anomalies

preprocessor ftp\_telnet\_protocol: ftp server default \

def\_max\_param\_len 100 \

ports { 21 2100 3535 } \

telnet\_cmds yes \

ignore\_telnet\_erase\_cmds yes \

ftp\_cmds { ABOR ACCT ADAT ALLO APPE AUTH CCC CDUP } \

ftp\_cmds { CEL CLNT CMD CONF CWD DELE ENC EPRT } \

ftp\_cmds { EPSV ESTA ESTP FEAT HELP LANG LIST LPRT } \

ftp\_cmds { LPSV MACB MAIL MDTM MIC MKD MLSD MLST } \

ftp\_cmds { MODE NLST NOOP OPTS PASS PASV PBSZ PORT } \

ftp\_cmds { PROT PWD QUIT REIN REST RETR RMD RNFR } \

ftp\_cmds { RNTO SDUP SITE SIZE SMNT STAT STOR STOU } \

ftp\_cmds { STRU SYST TEST TYPE USER XCUP XCRC XCWD } \

ftp\_cmds { XMAS XMD5 XMKD XPWD XRCP XRMD XRSQ XSEM } \

ftp\_cmds { XSEN XSHA1 XSHA256 } \

alt\_max\_param\_len 0 { ABOR CCC CDUP ESTA FEAT LPSV NOOP PASV PWD QUIT REIN STOU SYST XCUP XPWD } \

alt\_max\_param\_len 200 { ALLO APPE CMD HELP NLST RETR RNFR STOR STOU XMKD } \

alt\_max\_param\_len 256 { CWD RNTO } \

alt\_max\_param\_len 400 { PORT } \

alt\_max\_param\_len 512 { SIZE } \

chk\_str\_fmt { ACCT ADAT ALLO APPE AUTH CEL CLNT CMD } \

chk\_str\_fmt { CONF CWD DELE ENC EPRT EPSV ESTP HELP } \

chk\_str\_fmt { LANG LIST LPRT MACB MAIL MDTM MIC MKD } \

chk\_str\_fmt { MLSD MLST MODE NLST OPTS PASS PBSZ PORT } \

chk\_str\_fmt { PROT REST RETR RMD RNFR RNTO SDUP SITE } \

chk\_str\_fmt { SIZE SMNT STAT STOR STRU TEST TYPE USER } \

chk\_str\_fmt { XCRC XCWD XMAS XMD5 XMKD XRCP XRMD XRSQ } \

chk\_str\_fmt { XSEM XSEN XSHA1 XSHA256 } \

cmd\_validity ALLO < int [ char R int ] > \

cmd\_validity EPSV < [ { char 12 | char A char L char L } ] > \

cmd\_validity MACB < string > \

cmd\_validity MDTM < [ date nnnnnnnnnnnnnn[.n[n[n]]] ] string > \

cmd\_validity MODE < char ASBCZ > \

cmd\_validity PORT < host\_port > \

cmd\_validity PROT < char CSEP > \

cmd\_validity STRU < char FRPO [ string ] > \

cmd\_validity TYPE < { char AE [ char NTC ] | char I | char L [ number ] } >

preprocessor ftp\_telnet\_protocol: ftp client default \

max\_resp\_len 256 \

bounce yes \

ignore\_telnet\_erase\_cmds yes \

telnet\_cmds yes

# SMTP normalization and anomaly detection. For more information, see README.SMTP

preprocessor smtp: ports { 25 465 587 691 } \

inspection\_type stateful \

b64\_decode\_depth 0 \

qp\_decode\_depth 0 \

bitenc\_decode\_depth 0 \

uu\_decode\_depth 0 \

log\_mailfrom \

log\_rcptto \

log\_filename \

log\_email\_hdrs \

normalize cmds \

normalize\_cmds { ATRN AUTH BDAT CHUNKING DATA DEBUG EHLO EMAL ESAM ESND ESOM ETRN EVFY } \

normalize\_cmds { EXPN HELO HELP IDENT MAIL NOOP ONEX QUEU QUIT RCPT RSET SAML SEND SOML } \

normalize\_cmds { STARTTLS TICK TIME TURN TURNME VERB VRFY X-ADAT X-DRCP X-ERCP X-EXCH50 } \

normalize\_cmds { X-EXPS X-LINK2STATE XADR XAUTH XCIR XEXCH50 XGEN XLICENSE XQUE XSTA XTRN XUSR } \

max\_command\_line\_len 512 \

max\_header\_line\_len 1000 \

max\_response\_line\_len 512 \

alt\_max\_command\_line\_len 260 { MAIL } \

alt\_max\_command\_line\_len 300 { RCPT } \

alt\_max\_command\_line\_len 500 { HELP HELO ETRN EHLO } \

alt\_max\_command\_line\_len 255 { EXPN VRFY ATRN SIZE BDAT DEBUG EMAL ESAM ESND ESOM EVFY IDENT NOOP RSET } \

alt\_max\_command\_line\_len 246 { SEND SAML SOML AUTH TURN ETRN DATA RSET QUIT ONEX QUEU STARTTLS TICK TIME TURNME VERB X-EXPS X-LINK2STATE XADR XAUTH XCIR XEXCH50 XGEN XLICENSE XQUE XSTA XTRN XUSR } \

valid\_cmds { ATRN AUTH BDAT CHUNKING DATA DEBUG EHLO EMAL ESAM ESND ESOM ETRN EVFY } \

valid\_cmds { EXPN HELO HELP IDENT MAIL NOOP ONEX QUEU QUIT RCPT RSET SAML SEND SOML } \

valid\_cmds { STARTTLS TICK TIME TURN TURNME VERB VRFY X-ADAT X-DRCP X-ERCP X-EXCH50 } \

valid\_cmds { X-EXPS X-LINK2STATE XADR XAUTH XCIR XEXCH50 XGEN XLICENSE XQUE XSTA XTRN XUSR } \

xlink2state { enabled }

# Portscan detection. For more information, see README.sfportscan

# preprocessor sfportscan: proto { all } memcap { 10000000 } sense\_level { low }

# ARP spoof detection. For more information, see the Snort Manual - Configuring Snort - Preprocessors - ARP Spoof Preprocessor

# preprocessor arpspoof

# preprocessor arpspoof\_detect\_host: 192.168.40.1 f0:0f:00:f0:0f:00

# SSH anomaly detection. For more information, see README.ssh

preprocessor ssh: server\_ports { 22 } \

autodetect \

max\_client\_bytes 19600 \

max\_encrypted\_packets 20 \

max\_server\_version\_len 100 \

enable\_respoverflow enable\_ssh1crc32 \

enable\_srvoverflow enable\_protomismatch

# SMB / DCE-RPC normalization and anomaly detection. For more information, see README.dcerpc2

preprocessor dcerpc2: memcap 102400, events [co ]

preprocessor dcerpc2\_server: default, policy WinXP, \

detect [smb [139,445], tcp 135, udp 135, rpc-over-http-server 593], \

autodetect [tcp 1025:, udp 1025:, rpc-over-http-server 1025:], \

smb\_max\_chain 3, smb\_invalid\_shares ["C$", "D$", "ADMIN$"]

# DNS anomaly detection. For more information, see README.dns

preprocessor dns: ports { 53 } enable\_rdata\_overflow

# SSL anomaly detection and traffic bypass. For more information, see README.ssl

preprocessor ssl: ports { 443 465 563 636 989 992 993 994 995 7801 7802 7900 7901 7902 7903 7904 7905 7906 7907 7908 7909 7910 7911 7912 7913 7914 7915 7916 7917 7918 7919 7920 }, trustservers, noinspect\_encrypted

# SDF sensitive data preprocessor. For more information see README.sensitive\_data

preprocessor sensitive\_data: alert\_threshold 25

# SIP Session Initiation Protocol preprocessor. For more information see README.sip

preprocessor sip: max\_sessions 40000, \

ports { 5060 5061 5600 }, \

methods { invite \

cancel \

ack \

bye \

register \

options \

refer \

subscribe \

update \

join \

info \

message \

notify \

benotify \

do \

qauth \

sprack \

publish \

service \

unsubscribe \

prack }, \

max\_uri\_len 512, \

max\_call\_id\_len 80, \

max\_requestName\_len 20, \

max\_from\_len 256, \

max\_to\_len 256, \

max\_via\_len 1024, \

max\_contact\_len 512, \

max\_content\_len 2048

# IMAP preprocessor. For more information see README.imap

preprocessor imap: \

ports { 143 } \

b64\_decode\_depth 0 \

qp\_decode\_depth 0 \

bitenc\_decode\_depth 0 \

uu\_decode\_depth 0

# POP preprocessor. For more information see README.pop

preprocessor pop: \

ports { 110 } \

b64\_decode\_depth 0 \

qp\_decode\_depth 0 \

bitenc\_decode\_depth 0 \

uu\_decode\_depth 0

# Modbus preprocessor. For more information see README.modbus

preprocessor modbus: ports { 502 }

# DNP3 preprocessor. For more information see README.dnp3

preprocessor dnp3: ports { 20000 } \

memcap 262144 \

check\_crc

# Reputation preprocessor. For more information see README.reputation

preprocessor reputation: \

memcap 500, \

priority whitelist, \

nested\_ip inner, \

whitelist $WHITE\_LIST\_PATH/white\_list.rules, \

blacklist $BLACK\_LIST\_PATH/black\_list.rules

###################################################

# Step #6: Configure output plugins

# For more information, see Snort Manual, Configuring Snort - Output Modules

###################################################

# unified2

# Recommended for most installs

output unified2: filename merged.log, limit 128, nostamp, mpls\_event\_types, vlan\_event\_types

# Additional configuration for specific types of installs

output alert\_unified2: filename snort.alert, limit 128, nostamp

output log\_unified2: filename snort.log, limit 128, nostamp

# syslog

output alert\_syslog: LOG\_AUTH LOG\_ALERT

# pcap

output log\_tcpdump: tcpdump.log

# metadata reference data. do not modify these lines

include classification.config

include reference.config

###################################################

# Step #7: Customize your rule set

# For more information, see Snort Manual, Writing Snort Rules

#

# NOTE: All categories are enabled in this conf file

###################################################

# site specific rules

include $RULE\_PATH/local.rules

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/deleted.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-java.rules

include $RULE\_PATH/file-multimedia.rules

include $RULE\_PATH/file-office.rules

include $RULE\_PATH/file-other.rules

include $RULE\_PATH/file-pdf.rules

include $RULE\_PATH/finger.rules

include $RULE\_PATH/ftp.rules

include $RULE\_PATH/icmp-info.rules

include $RULE\_PATH/icmp.rules

include $RULE\_PATH/imap.rules

include $RULE\_PATH/indicator-compromise.rules

include $RULE\_PATH/indicator-obfuscation.rules

include $RULE\_PATH/indicator-shellcode.rules

include $RULE\_PATH/indicator-scan.rules

include $RULE\_PATH/info.rules

include $RULE\_PATH/malware-backdoor.rules

include $RULE\_PATH/malware-cnc.rules

include $RULE\_PATH/malware-other.rules

include $RULE\_PATH/malware-tools.rules

include $RULE\_PATH/misc.rules

include $RULE\_PATH/multimedia.rules

include $RULE\_PATH/mysql.rules

include $RULE\_PATH/netbios.rules

include $RULE\_PATH/nntp.rules

include $RULE\_PATH/oracle.rules

include $RULE\_PATH/os-linux.rules

include $RULE\_PATH/os-mobile.rules

include $RULE\_PATH/os-other.rules

include $RULE\_PATH/os-solaris.rules

include $RULE\_PATH/os-windows.rules

include $RULE\_PATH/other-ids.rules

include $RULE\_PATH/p2p.rules

include $RULE\_PATH/phishing-spam.rules

include $RULE\_PATH/policy.rules

include $RULE\_PATH/policy-multimedia.rules

include $RULE\_PATH/policy-other.rules

include $RULE\_PATH/policy.rules

include $RULE\_PATH/policy-social.rules

include $RULE\_PATH/policy-spam.rules

include $RULE\_PATH/pop2.rules

include $RULE\_PATH/pop3.rules

include $RULE\_PATH/protocol-dns.rules

include $RULE\_PATH/protocol-finger.rules

include $RULE\_PATH/protocol-ftp.rules

include $RULE\_PATH/protocol-icmp.rules

include $RULE\_PATH/protocol-imap.rules

include $RULE\_PATH/protocol-nntp.rules

include $RULE\_PATH/protocol-other.rules

include $RULE\_PATH/protocol-pop.rules

include $RULE\_PATH/protocol-rpc.rules

include $RULE\_PATH/protocol-scada.rules

include $RULE\_PATH/protocol-services.rules

include $RULE\_PATH/protocol-snmp.rules

include $RULE\_PATH/protocol-telnet.rules

include $RULE\_PATH/protocol-tftp.rules

include $RULE\_PATH/protocol-voip.rules

include $RULE\_PATH/pua-adware.rules

include $RULE\_PATH/pua-other.rules

include $RULE\_PATH/pua-p2p.rules

include $RULE\_PATH/pua-toolbars.rules

include $RULE\_PATH/rpc.rules

include $RULE\_PATH/rservices.rules

include $RULE\_PATH/scada.rules

include $RULE\_PATH/scan.rules

include $RULE\_PATH/server-apache.rules

include $RULE\_PATH/server-iis.rules

include $RULE\_PATH/server-mail.rules

include $RULE\_PATH/server-mssql.rules

include $RULE\_PATH/server-mysql.rules

include $RULE\_PATH/server-oracle.rules

include $RULE\_PATH/server-other.rules

include $RULE\_PATH/server-samba.rules

include $RULE\_PATH/server-webapp.rules

include $RULE\_PATH/shellcode.rules

include $RULE\_PATH/smtp.rules

include $RULE\_PATH/snmp.rules

include $RULE\_PATH/specific-threats.rules

include $RULE\_PATH/spyware-put.rules

include $RULE\_PATH/sql.rules

include $RULE\_PATH/telnet.rules

include $RULE\_PATH/tftp.rules

include $RULE\_PATH/virus.rules

include $RULE\_PATH/voip.rules

include $RULE\_PATH/web-activex.rules

include $RULE\_PATH/web-attacks.rules

include $RULE\_PATH/web-cgi.rules

include $RULE\_PATH/web-client.rules

include $RULE\_PATH/web-coldfusion.rules

include $RULE\_PATH/web-frontpage.rules

include $RULE\_PATH/web-iis.rules

include $RULE\_PATH/web-misc.rules

include $RULE\_PATH/web-php.rules

include $RULE\_PATH/x11.rules

###################################################

# Step #8: Customize your preprocessor and decoder alerts

# For more information, see README.decoder\_preproc\_rules

###################################################

# decoder and preprocessor event rules

include $PREPROC\_RULE\_PATH/preprocessor.rules

include $PREPROC\_RULE\_PATH/decoder.rules

include $PREPROC\_RULE\_PATH/sensitive-data.rules

###################################################

# Step #9: Customize your Shared Object Snort Rules

# For more information, see http://vrt-blog.snort.org/2009/01/using-vrt-certified-shared-object-rules.html

###################################################

# dynamic library rules

#include $SO\_RULE\_PATH/browser-ie.rules

#include $SO\_RULE\_PATH/browser-other.rules

#include $SO\_RULE\_PATH/exploit-kit.rules

#include $SO\_RULE\_PATH/file-executable.rules

#include $SO\_RULE\_PATH/file-flash.rules

#include $SO\_RULE\_PATH/file-image.rules

#include $SO\_RULE\_PATH/file-java.rules

#include $SO\_RULE\_PATH/file-multimedia.rules

#include $SO\_RULE\_PATH/file-office.rules

#include $SO\_RULE\_PATH/file-other.rules

#include $SO\_RULE\_PATH/file-pdf.rules

#include $SO\_RULE\_PATH/indicator-shellcode.rules

#include $SO\_RULE\_PATH/malware-cnc.rules

#include $SO\_RULE\_PATH/malware-other.rules

#include $SO\_RULE\_PATH/netbios.rules

#include $SO\_RULE\_PATH/os-linux.rules

#include $SO\_RULE\_PATH/os-other.rules

#include $SO\_RULE\_PATH/os-windows.rules

#include $SO\_RULE\_PATH/policy-other.rules

#include $SO\_RULE\_PATH/policy-social.rules

#include $SO\_RULE\_PATH/protocol-dns.rules

#include $SO\_RULE\_PATH/protocol-nntp.rules

#include $SO\_RULE\_PATH/protocol-other.rules

#include $SO\_RULE\_PATH/protocol-scada.rules

#include $SO\_RULE\_PATH/protocol-snmp.rules

#include $SO\_RULE\_PATH/protocol-tftp.rules

#include $SO\_RULE\_PATH/protocol-voip.rules

#include $SO\_RULE\_PATH/pua-p2p.rules

#include $SO\_RULE\_PATH/server-apache.rules

#include $SO\_RULE\_PATH/server-iis.rules

#include $SO\_RULE\_PATH/server-mail.rules

#include $SO\_RULE\_PATH/server-mysql.rules

#include $SO\_RULE\_PATH/server-oracle.rules

#include $SO\_RULE\_PATH/server-other.rules

#include $SO\_RULE\_PATH/server-webapp.rules

# Event thresholding or suppression commands. See threshold.conf

include threshold.conf

After running the snort, we got the successful validation:

2. Testing the snort

We test the Snort rule by using the following signature:

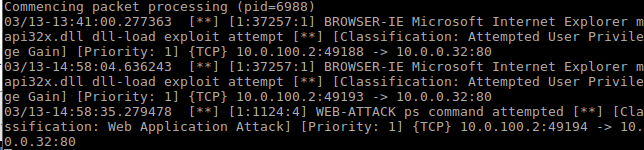
The signature goes as follows:

alert tcp $HOME\_NET any -> $EXTERNAL\_NET $HTTP\_PORTS ( sid: 1124; rev: 4; msg: "WEB-ATTACK ps command attempted"; flow: to\_server,established; uricontent:"/bin/ps"; nocase; classtype: web-application-attack;)

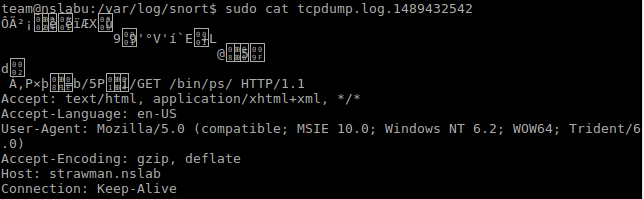
Our request URL in windows VM is:

<http://strawman.nslab/bin/ps>

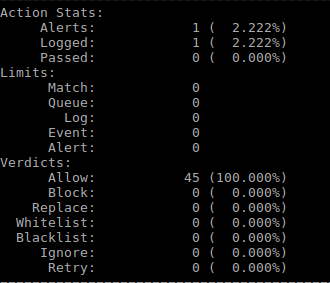
The snapshot after the web-application-attack is shown as follow:



The alert for attack is marked by red line.

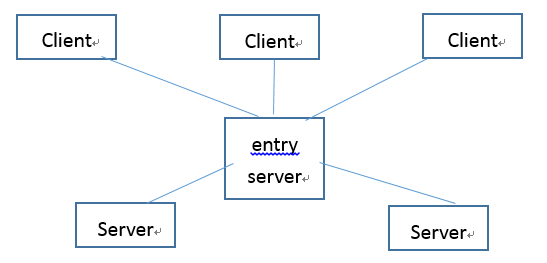
This is also recorded in the TCP dump (some of the character is not in ASCII)

Following is the aggregate for this snort session:

We could see that although the attack raised an alert, the NIDS still allowed.

3. We could set an entry server between the servers and the clients.

The diagram goes as follows:

 By this entry server, we could route the SSL traffic to the SSL servers, which would improve the performance of this network. Further, the NIDS works on this entry server to prevent the attacks without installing snort on the SSL Server.