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
[80,81,311,383,591,593,901,1220,1414,1741,1830,2301,2381,2809,3037,3128,3702,4343,4848,5
250,6988,7000,7001,7144,7145,7510,7777,7779,8000,8008,8014,8028,8080,8085,8088,8090,81
18,8123,8180,8181,8243,8280,8300,8800,8888,8899,9000,9060,9080,9090,9091,9443,9999,113
71,34443,34444,41080,50002,55555]

# 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
[64.12.24.0/23,64.12.28.0/23,64.12.161.0/24,64.12.163.0/24,64.12.200.0/24,205.188.3.0/24,20
5.188.5.0/24,205.188.7.0/24,205.188.9.0/24,205.188.153.0/24,205.188.179.0/24,205.188.248.0/
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
```

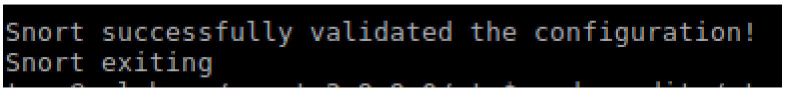
```
# performance statistics.  For more information, see the Snort Manual, Configuring Snort -
Preprocessors - Performance Monitor
# preprocessor perfmonitor: time 300 file /var/snort/snort.stats pktcnt 10000

# 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:



```
Snort successfully validated the configuration!
Snort exiting
```

2. Testing the snort

We test the Snort rule by using the following signature:

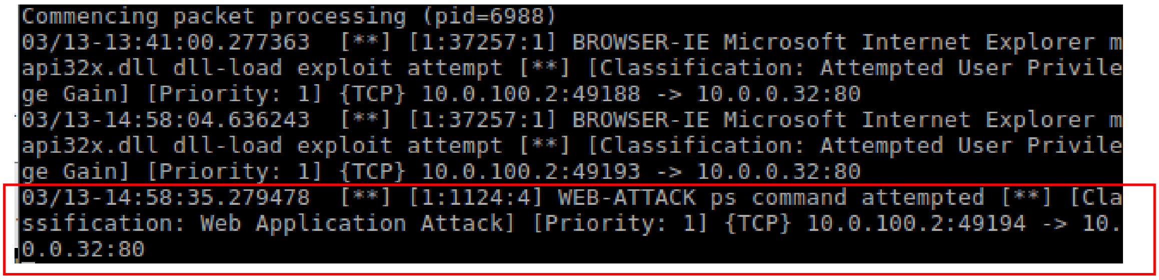
The signature goes as follows:

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

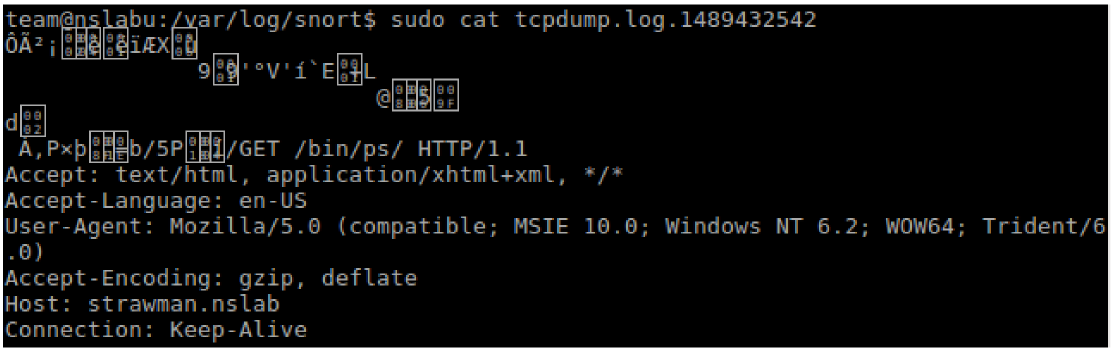
The snapshot after the web-application-attack is shown as follow:



```
Commencing packet processing (pid=6988)
03/13-13:41:00.277363  [**] [1:37257:1] BROWSER-IE Microsoft Internet Explorer m
api32x.dll dll-load exploit attempt [**] [Classification: Attempted User Privile
ge Gain] [Priority: 1] {TCP} 10.0.100.2:49188 -> 10.0.0.32:80
03/13-14:58:04.636243  [**] [1:37257:1] BROWSER-IE Microsoft Internet Explorer m
api32x.dll dll-load exploit attempt [**] [Classification: Attempted User Privile
ge Gain] [Priority: 1] {TCP} 10.0.100.2:49193 -> 10.0.0.32:80
03/13-14:58:35.279478  [**] [1:1124:4] WEB-ATTACK ps command attempted [**] [Cla
ssification: Web Application Attack] [Priority: 1] {TCP} 10.0.100.2:49194 -> 10.
0.0.32:80
```

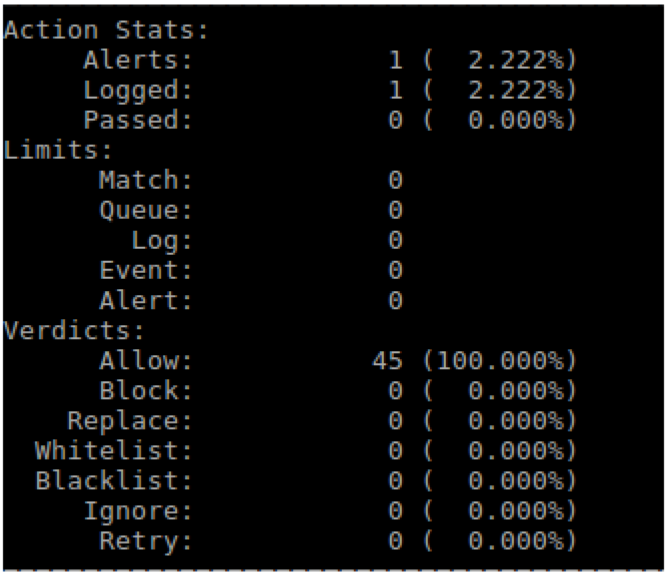
The alert for attack is marked by red line.

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



```
team@nslabu:/var/log/snort$ sudo cat tcpdump.log.1489432542
ÔÃ²¡□□□ïÆX□
                 9□'°V'í`E□L
                                 @□□□
d□
 À,Px þ□b/5P□/GET /bin/ps/ HTTP/1.1
Accept: text/html, application/xhtml+xml, */*
Accept-Language: en-US
User-Agent: Mozilla/5.0 (compatible; MSIE 10.0; Windows NT 6.2; WOW64; Trident/6
.0)
Accept-Encoding: gzip, deflate
Host: strawman.nslab
Connection: Keep-Alive
```

Following is the aggregate for this snort session:

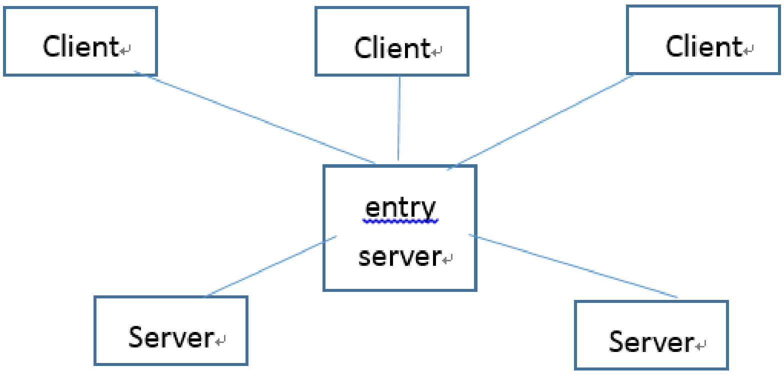


```
Action Stats:
     Alerts:            1 (  2.222%)
     Logged:            1 (  2.222%)
     Passed:            0 (  0.000%)
Limits:
      Match:            0
      Queue:            0
        Log:            0
      Event:            0
      Alert:            0
Verdicts:
      Allow:           45 (100.000%)
      Block:            0 (  0.000%)
    Replace:            0 (  0.000%)
  Whitelist:            0 (  0.000%)
  Blacklist:            0 (  0.000%)
     Ignore:            0 (  0.000%)
      Retry:            0 (  0.000%)
```

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.