Step 1: Inform Suricata about your network

vars:
more specific is better for alert accuracy and performance address-groups:
HOME_NET: "[192.168.0.0/16,10.0.0.0/8,172.16.0.0/12]"
#HOME_NET: "[192.168.0.0/16]"
#HOME_NET: "[10.0.0.0/8]"
#HOME_NET: "[172.16.0.0/12]"
#HOME_NET: "any"

EXTERNAL_NET: "!$HOME_NET"
#EXTERNAL_NET: "any"

HTTP_SERVERS: "$HOME_NET"
SMTP_SERVERS: "$HOME_NET"
SQL_SERVERS: "$HOME_NET"
DNS_SERVERS: "$HOME_NET"
TELNET_SERVERS: "$HOME_NET"
AIM_SERVERS: "$EXTERNAL_NET"
DC_SERVERS: "$HOME_NET"
DNP3_SERVER: "$HOME_NET"
DNP3_CLIENT: "$HOME_NET"
MODBUS_CLIENT: "$HOME_NET"
MODBUS_SERVER: "$HOME_NET"
ENIP_CLIENT: "$HOME_NET"
ENIP_SERVER: "$HOME_NET"

port-groups: HTTP_PORTS: "80"
SHELLCODE_PORTS: "!80"
ORACLE_PORTS: 1521
SSH_PORTS: 22
DNP3_PORTS: 20000
MODBUS_PORTS: 502
FILE_DATA_PORTS: "[$HTTP_PORTS,110,143]"
FTP_PORTS: 21
GENEVE_PORTS: 6081
VXLAN_PORTS: 4789
TEREDO_PORTS: 3544

이 파일은 Suricata라는 네트워크 침입 탐지 시스템에서 중요한 설정 파일이다.

‘vars’ 섹션으로, Suricata가 네트워크를 어떻게 인식하고 분석할지에 대한 정보를 설정하는 부분.
구체적으로는 네트워크 주소와 포트를 정의하여 Suricata의 규칙이 이들에 맞춰 적용될 수 있도록 한다.
이 파일을 통해 네트워크 안에서 어떤 IP 대역이 집(home) 네트워크인지, 외부 네트워크는 어디인지,
그리고 서버나 포트에 대한 정보를 설정할 수 있다.

1. 네트워크 주소 그룹 **(address-groups)**

vars:
more specific is better for alert accuracy and performance address-groups:
HOME_NET: "[192.168.0.0/16,10.0.0.0/8,172.16.0.0/12]"
#HOME_NET: "[192.168.0.0/16]"
#HOME_NET: "[10.0.0.0/8]"
#HOME_NET: "[172.16.0.0/12]"
#HOME_NET: "any"

이 부분은 네트워크 주소를 그룹으로 묶어서, Suricata가 어떤 네트워크를 '우리 집'으로 생각해야 하는지, 그리고 외부 네트워크는 어디인지 설정하는 부분이다.

- **HOME_NET**: 자신의 네트워크를 정의합니다. 이 네트워크를 공격받는 대상이라고 생각하면 된다. 여러 IP 대역을 쉼표로 구분하여 설정할 수 있습니다. 예를 들어, ‘[192.168.0.0/16, 10.0.0.0/8, 172.16.0.0/12]’는 세 개의 서브넷을 포함하는 설정입니다. 이 주소 그룹은 Suricata가 분석할 주요 내부 네트워크를 정의합니다. 주석처리 된 다른 예제들은 특정 서브넷만 포함하거나, ‘any’를 사용하여 모든 IP 주소를 허용 할 수 있습니다.

  이렇게 여러 개의 네트워크 범위를 HOME_NET으로 지정할 수 있다.

  - 예를 들어 192.168.0.0/16은 192.168.0.0부터 192.168.255.255까지의 IP 주소 대역.
  - 여러 네트워크 범위를 HOME_NET으로 설정해서, Suricata가 이 범위 안에 있는 IP를 '우리 집'으로 인식하도록 할 수 있음.
  - any라고 설정하면, 모든 IP를 집 네트워크로 인식하게 됨. 이건 권장하지 않는다, 정확도가 떨어지기 때문.

EXTERNAL_NET: "!$HOME_NET"
#EXTERNAL_NET: "any"

- **EXTERNAL_NET**: ‘HOME NET’과 반대되는 외부 네트워크를 정의합니다. 보통 !$HOME_NET으로 설정해서, 집 네트워크가 아닌 모든 IP를 외부 네트워크로 인식하게 한다. 여기서 !를 "제외하다"는 의미.

HTTP_SERVERS: "$HOME_NET"
SMTP_SERVERS: "$HOME_NET"
SQL_SERVERS: "$HOME_NET"
DNS_SERVERS: "$HOME_NET"
TELNET_SERVERS: "$HOME_NET"
AIM_SERVERS: "$EXTERNAL_NET"
DC_SERVERS: "$HOME_NET"
DNP3_SERVER: "$HOME_NET"
DNP3_CLIENT: "$HOME_NET"
MODBUS_CLIENT: "$HOME_NET"

MODBUS_SERVER: "$HOME_NET"
ENIP_CLIENT: "$HOME_NET"
ENIP_SERVER: "$HOME_NET"

특정 서비스나 프로토콜에 대한 서버와 클라이언트를 정의합니다.

- 서버 그룹들: `HTTP_SERVERS`, `SMTP_SERVERS`, `SQL_SERVERS` 등등은 각각 HTTP, 이메일, 데이터베이스 같은 서버들이 위치한 네트워크를 설정하는 것. 여기서는 보통 집 네트워크에 속해 있으니까 `$HOME_NET`으로 설정해놨음.
    - 예시:
        - `HTTP_SERVERS: "$HOME_NET"`: HTTP 서버들이 집 네트워크에 있다고 지정해주는 것.
- **'DNP3_SERVER', "MODBUS_SERVER'** 등은 각각 DNP3 및 MODBUS 프로토콜을 사용하는 서버의 IP 주소를 'HOME_NET'으로 설정합니다.
- 특정 서버들 **(EXTERNAL_NET)**: AIM, Telnet 같은 경우는 외부 서버로 설정할 수도 있다. 예를 들어 `AIM_SERVERS: "$EXTERNAL_NET"`는 AIM 서버들이 외부 네트워크에 있다고 지정한 것.

2. 포트 그룹 **(port-groups)**

port-groups: HTTP_PORTS: "80"
SHELLCODE_PORTS: "!80"
ORACLE_PORTS: 1521
SSH_PORTS: 22
DNP3_PORTS: 20000
MODBUS_PORTS: 502
FILE_DATA_PORTS: "[$HTTP_PORTS,110,143]"
FTP_PORTS: 21
GENEVE_PORTS: 6081
VXLAN_PORTS: 4789
TEREDO_PORTS: 3544

Suricata가 특정 포트를 모니터링할 때 사용할 포트 그룹을 정의합니다.

- **HTTP_PORTS**: 트래픽을 모니터링할 포트를 설정. 기본적으로 `80`번 포트는 웹 트래픽(HTTP)에 사용됩니다. 예를 들어, 우리가 웹사이트를 열 때 기본적으로 80번 포트를 통해 접속한다.
- **SHELLCODE_PORTS**: 쉘코드 공격을 탐지할 포트를 설정. 여기서는 80번 포트를 제외하고 다른 포트들을 지정. `!80`은 "80번 포트는 제외"라는 의미.
- 특정 서비스 포트들:
    - **ORACLE_PORTS**: Oracle 데이터베이스의 기본 포트 1521을 설정.
    - **SSH_PORTS**: SSH 프로토콜의 기본 포트 22를 설정.
    - **DNP3_PORT**: DNP3 프로토콜에 사용되는 포트 2000을 설정.
    - **MODBUS_PORTS**: MODBUS 프로토콜의 기본 포트 502를 설정.
    - **FILE_DATA_PORTS**: 파일 데이터를 주고받는 포트들을 묶어서 지정해둔 것. 여기서는 HTTP 포트, 110번 포트(POP3 이메일), 143번 포트(IMAP 이메일)을 묶어놓았음.

    - 파일 전송을 다루는 포트들을 설정. HTTP 포트 80과 추가로 110, 143 을 포함.
    - **FTP_PORTS**: FTP 프로토콜의 기본 포트 21을 설정.
    - **GENEVE_PORTS**: GENEVE 프로토콜의 포트 6081을 설정.
    - **VXLAN_PORTS**: VXLAN 프로토콜의 포트 4789를 설정.
    - **TEREDO_PORTS**: TEREDO 프로토콜의 포트 3544를 설정.
- 이 외에도 여러 가지 포트를 지정할 수 있음. 포트는 서비스를 구분하는 중요한 정보라서, 올바르게 설정해야 함.

요약

- 네트워크 주소 그룹: HOME_NET은 내 네트워크, EXTERNAL_NET은 외부 네트워크.
- 포트 그룹: 서비스마다 사용하는 포트를 지정.

이 파일을 수정하면서, Suricata에게 "이건 내 네트워크야!", "이 포트는 이 서비스가 사용해!"라고 알려주는 거지. 그렇게 해서 네트워크를 더 안전하게 지킬 수 있는 거야.

```
##
## Step 2: Select outputs to enable
##

# The default logging directory.  Any log or output file will be
# placed here if it's not specified with a full path name. This can be
# overridden with the -l command line parameter.
default-log-dir: /var/log/suricata/

# Global stats configuration
stats:
  enabled: yes
  # The interval field (in seconds) controls the interval at
  # which stats are updated in the log.
  interval: 8
  # Add decode events to stats.
  #decoder-events: true
  # Decoder event prefix in stats. Has been 'decoder' before, but that leads
  # to missing events in the eve.stats records. See issue #2225.
  #decoder-events-prefix: "decoder.event"
  # Add stream events as stats.
  #stream-events: false

# Plugins -- Experimental -- specify the filename for each plugin shared object
plugins:
#  - /path/to/plugin.so

# Configure the type of alert (and other) logging you would like.
outputs:
  # a line based alerts log similar to Snort's fast.log
  - fast:
```

```
    enabled: yes
    filename: fast.log
    append: yes
    #filetype: regular # 'regular', 'unix_stream' or 'unix_dgram'

# Extensible Event Format (nicknamed EVE) event log in JSON format
- eve-log:
    enabled: yes
    filetype: regular #regular|syslog|unix_dgram|unix_stream|redis
    filename: eve.json
    # Enable for multi-threaded eve.json output; output files are amended with
    # an identifier, e.g., eve.9.json
    #threaded: false
    #prefix: "@cee: " # prefix to prepend to each log entry
    # the following are valid when type: syslog above
    #identity: "suricata"
    #facility: local5
    #level: Info ## possible levels: Emergency, Alert, Critical,
                ## Error, Warning, Notice, Info, Debug
    #ethernet: no  # log ethernet header in events when available
    #redis:
    #  server: 127.0.0.1
    #  port: 6379
    #  async: true ## if redis replies are read asynchronously
    #  mode: list ## possible values: list|lpush (default), rpush, channel|publish
    #             ## lpush and rpush are using a Redis list. "list" is an alias for lpush
    #             ## publish is using a Redis channel. "channel" is an alias for publish
    #  key: suricata ## key or channel to use (default to suricata)
    # Redis pipelining set up. This will enable to only do a query every
    # 'batch-size' events. This should lower the latency induced by network
    # connection at the cost of some memory. There is no flushing implemented
    # so this setting should be reserved to high traffic Suricata deployments.
    #  pipelining:
    #    enabled: yes ## set enable to yes to enable query pipelining
    #    batch-size: 10 ## number of entries to keep in buffer

    # Include top level metadata. Default yes.
    #metadata: no

    # include the name of the input pcap file in pcap file processing mode
    pcap-file: false

    # Community Flow ID
    # Adds a 'community_id' field to EVE records. These are meant to give
    # records a predictable flow ID that can be used to match records to
    # output of other tools such as Zeek (Bro).
    #
    # Takes a 'seed' that needs to be same across sensors and tools
    # to make the id less predictable.

    # enable/disable the community id feature.
    community-id: false
```

```
# Seed value for the ID output. Valid values are 0-65535.
community-id-seed: 0

# HTTP X-Forwarded-For support by adding an extra field or overwriting
# the source or destination IP address (depending on flow direction)
# with the one reported in the X-Forwarded-For HTTP header. This is
# helpful when reviewing alerts for traffic that is being reverse
# or forward proxied.
xff:
  enabled: no
  # Two operation modes are available: "extra-data" and "overwrite".
  mode: extra-data
  # Two proxy deployments are supported: "reverse" and "forward". In
  # a "reverse" deployment the IP address used is the last one, in a
  # "forward" deployment the first IP address is used.
  deployment: reverse
  # Header name where the actual IP address will be reported. If more
  # than one IP address is present, the last IP address will be the
  # one taken into consideration.
  header: X-Forwarded-For

types:
  - alert:
      # payload: yes             # enable dumping payload in Base64
      # payload-buffer-size: 4kb # max size of payload buffer to output in eve-log
      # payload-printable: yes   # enable dumping payload in printable (lossy) format
      # packet: yes              # enable dumping of packet (without stream segments)
      # metadata: no             # enable inclusion of app layer metadata with alert. Default yes
      # http-body: yes           # Requires metadata; enable dumping of HTTP body in Base64
      # http-body-printable: yes # Requires metadata; enable dumping of HTTP body in printable format

      # Enable the logging of tagged packets for rules using the
      # "tag" keyword.
      tagged-packets: yes
      # Enable logging the final action taken on a packet by the engine
      # (e.g: the alert may have action 'allowed' but the verdict be
      # 'drop' due to another alert. That's the engine's verdict)
      # verdict: yes
  # app layer frames
  - frame:
      # disabled by default as this is very verbose.
      enabled: no
  - anomaly:
      # Anomaly log records describe unexpected conditions such
      # as truncated packets, packets with invalid IP/UDP/TCP
      # length values, and other events that render the packet
      # invalid for further processing or describe unexpected
      # behavior on an established stream. Networks which
      # experience high occurrences of anomalies may experience
      # packet processing degradation.
      #
      # Anomalies are reported for the following:
```

```
    # 1. Decode: Values and conditions that are detected while
    # decoding individual packets. This includes invalid or
    # unexpected values for low-level protocol lengths as well
    # as stream related events (TCP 3-way handshake issues,
    # unexpected sequence number, etc).
    # 2. Stream: This includes stream related events (TCP
    # 3-way handshake issues, unexpected sequence number,
    # etc).
    # 3. Application layer: These denote application layer
    # specific conditions that are unexpected, invalid or are
    # unexpected given the application monitoring state.
    #
    # By default, anomaly logging is enabled. When anomaly
    # logging is enabled, applayer anomaly reporting is
    # also enabled.
    enabled: yes
    #
    # Choose one or more types of anomaly logging and whether to enable
    # logging of the packet header for packet anomalies.
    types:
      # decode: no
      # stream: no
      # applayer: yes
    #packethdr: no
- http:
    extended: yes     # enable this for extended logging information
    # custom allows additional HTTP fields to be included in eve-log.
    # the example below adds three additional fields when uncommented
    #custom: [Accept-Encoding, Accept-Language, Authorization]
    # set this value to one and only one from {both, request, response}
    # to dump all HTTP headers for every HTTP request and/or response
    # dump-all-headers: none
- dns:
    # This configuration uses the new DNS logging format,
    # the old configuration is still available:
    # https://docs.suricata.io/en/latest/output/eve/eve-json-output.html#dns-v1-format

    # As of Suricata 5.0, version 2 of the eve dns output
    # format is the default.
    #version: 2

    # Enable/disable this logger. Default: enabled.
    #enabled: yes

    # Control logging of requests and responses:
    # - requests: enable logging of DNS queries
    # - responses: enable logging of DNS answers
    # By default both requests and responses are logged.
    #requests: no
    #responses: no

    # Format of answer logging:
```

```
          # - detailed: array item per answer
          # - grouped: answers aggregated by type
          # Default: all
          #formats: [detailed, grouped]

          # DNS record types to log, based on the query type.
          # Default: all.
          #types: [a, aaaa, cname, mx, ns, ptr, txt]
        - tls:
          extended: yes     # enable this for extended logging information
          # output TLS transaction where the session is resumed using a
          # session id
          #session-resumption: no
          # ja4 hashes in tls records will never be logged unless
          # the following is set to on. (Default off)
          # ja4: off
          # custom controls which TLS fields that are included in eve-log
          #custom: [subject, issuer, session_resumed, serial, fingerprint, sni, version, not_before, not_after,
certificate, chain, ja3, ja3s, ja4]
        - files:
          force-magic: no   # force logging magic on all logged files
          # force logging of checksums, available hash functions are md5,
          # sha1 and sha256
          #force-hash: [md5]
        #- drop:
        #    alerts: yes      # log alerts that caused drops
        #    flows: all       # start or all: 'start' logs only a single drop
        #                     # per flow direction. All logs each dropped pkt.
          # Enable logging the final action taken on a packet by the engine
          # (will show more information in case of a drop caused by 'reject')
          # verdict: yes
        - smtp:
          #extended: yes # enable this for extended logging information
          # this includes: bcc, message-id, subject, x_mailer, user-agent
          # custom fields logging from the list:
          #  reply-to, bcc, message-id, subject, x-mailer, user-agent, received,
          #  x-originating-ip, in-reply-to, references, importance, priority,
          #  sensitivity, organization, content-md5, date
          #custom: [received, x-mailer, x-originating-ip, relays, reply-to, bcc]
          # output md5 of fields: body, subject
          # for the body you need to set app-layer.protocols.smtp.mime.body-md5
          # to yes
          #md5: [body, subject]
      #- dnp3
        - ftp
        - rdp
        - nfs
        - smb
        - tftp
        - ike
        - dcerpc
        - krb5
```

```
      - bittorrent-dht
      - snmp
      - rfb
      - sip
      - quic:
          # ja4 hashes in quic records will never be logged unless
          # the following is set to on. (Default off)
          # ja4: off
      - dhcp:
          enabled: yes
          # When extended mode is on, all DHCP messages are logged
          # with full detail. When extended mode is off (the
          # default), just enough information to map a MAC address
          # to an IP address is logged.
          extended: no
      - ssh
      - mqtt:
          # passwords: yes           # enable output of passwords
      - http2
      - pgsql:
          enabled: no
          # passwords: yes           # enable output of passwords. Disabled by default
      - stats:
          totals: yes       # stats for all threads merged together
          threads: no       # per thread stats
          deltas: no        # include delta values
      # bi-directional flows
      - flow
      # uni-directional flows
      #- netflow

      # Metadata event type. Triggered whenever a pktvar is saved
      # and will include the pktvars, flowvars, flowbits and
      # flowints.
      #- metadata

      # EXPERIMENTAL per packet output giving TCP state tracking details
      # including internal state, flags, etc.
      # This output is experimental, meant for debugging and subject to
      # change in both config and output without any notice.
      #- stream:
      #    all: false                # log all TCP packets
      #    event-set: false          # log packets that have a decoder/stream event
      #    state-update: false       # log packets triggering a TCP state update
      #    spurious-retransmission: false  # log spurious retransmission packets

# a line based log of HTTP requests (no alerts)
- http-log:
    enabled: no
    filename: http.log
    append: yes
    #extended: yes     # enable this for extended logging information
```

```
    #custom: yes       # enable the custom logging format (defined by customformat)
    #customformat: "%{%D-%H:%M:%S}t.%z %{X-Forwarded-For}i %H %m %h %u %s %B %a:%p -> %A:%P"
    #filetype: regular # 'regular', 'unix_stream' or 'unix_dgram'

# a line based log of TLS handshake parameters (no alerts)
- tls-log:
    enabled: no  # Log TLS connections.
    filename: tls.log # File to store TLS logs.
    append: yes
    #extended: yes     # Log extended information like fingerprint
    #custom: yes       # enabled the custom logging format (defined by customformat)
    #customformat: "%{%D-%H:%M:%S}t.%z %a:%p -> %A:%P %v %n %d %D"
    #filetype: regular # 'regular', 'unix_stream' or 'unix_dgram'
    # output TLS transaction where the session is resumed using a
    # session id
    #session-resumption: no

# output module to store certificates chain to disk
- tls-store:
    enabled: no
    #certs-log-dir: certs # directory to store the certificates files

# Packet log... log packets in pcap format. 3 modes of operation: "normal"
# "multi" and "sguil".
#
# In normal mode a pcap file "filename" is created in the default-log-dir,
# or as specified by "dir".
# In multi mode, a file is created per thread. This will perform much
# better, but will create multiple files where 'normal' would create one.
# In multi mode the filename takes a few special variables:
# - %n -- thread number
# - %i -- thread id
# - %t -- timestamp (secs or secs.usecs based on 'ts-format'
# E.g. filename: pcap.%n.%t
#
# Note that it's possible to use directories, but the directories are not
# created by Suricata. E.g. filename: pcaps/%n/log.%s will log into the
# per thread directory.
#
# Also note that the limit and max-files settings are enforced per thread.
# So the size limit when using 8 threads with 1000mb files and 2000 files
# is: 8*1000*2000 ~ 16TiB.
#
# In Sguil mode "dir" indicates the base directory. In this base dir the
# pcaps are created in the directory structure Sguil expects:
#
# $sguil-base-dir/YYYY-MM-DD/$filename.<timestamp>
#
# By default all packets are logged except:
# - TCP streams beyond stream.reassembly.depth
# - encrypted streams after the key exchange
#
```

```
  - pcap-log:
      enabled: no
      filename: log.pcap

      # File size limit.  Can be specified in kb, mb, gb.  Just a number
      # is parsed as bytes.
      limit: 1000mb

      # If set to a value, ring buffer mode is enabled. Will keep maximum of
      # "max-files" of size "limit"
      max-files: 2000

      # Compression algorithm for pcap files. Possible values: none, lz4.
      # Enabling compression is incompatible with the sguil mode. Note also
      # that on Windows, enabling compression will *increase* disk I/O.
      compression: none

      # Further options for lz4 compression. The compression level can be set
      # to a value between 0 and 16, where higher values result in higher
      # compression.
      #lz4-checksum: no
      #lz4-level: 0

      mode: normal # normal, multi or sguil.

      # Directory to place pcap files. If not provided the default log
      # directory will be used. Required for "sguil" mode.
      #dir: /nsm_data/

      #ts-format: usec # sec or usec second format (default) is filename.sec usec is filename.sec.usec
      use-stream-depth: no #If set to "yes" packets seen after reaching stream inspection depth are ignored. "no"
logs all packets
      honor-pass-rules: no # If set to "yes", flows in which a pass rule matched will stop being logged.
      # Use "all" to log all packets or use "alerts" to log only alerted packets and flows or "tag"
      # to log only flow tagged via the "tag" keyword
      #conditional: all

  # a full alert log containing much information for signature writers
  # or for investigating suspected false positives.
  - alert-debug:
      enabled: no
      filename: alert-debug.log
      append: yes
      #filetype: regular # 'regular', 'unix_stream' or 'unix_dgram'

  # Stats.log contains data from various counters of the Suricata engine.
  - stats:
      enabled: yes
      filename: stats.log
      append: yes       # append to file (yes) or overwrite it (no)
      totals: yes       # stats for all threads merged together
      threads: no       # per thread stats
```

```
    #null-values: yes  # print counters that have value 0. Default: no

# a line based alerts log similar to fast.log into syslog
- syslog:
    enabled: no
    # reported identity to syslog. If omitted the program name (usually
    # suricata) will be used.
    #identity: "suricata"
    facility: local5
    #level: Info ## possible levels: Emergency, Alert, Critical,
                 ## Error, Warning, Notice, Info, Debug
# Output module for storing files on disk. Files are stored in
# directory names consisting of the first 2 characters of the
# SHA256 of the file. Each file is given its SHA256 as a filename.
#
# When a duplicate file is found, the timestamps on the existing file
# are updated.
#
# Unlike the older filestore, metadata is not written by default
# as each file should already have a "fileinfo" record in the
# eve-log. If write-fileinfo is set to yes, then each file will have
# one more associated .json files that consist of the fileinfo
# record. A fileinfo file will be written for each occurrence of the
# file seen using a filename suffix to ensure uniqueness.
#
# To prune the filestore directory see the "suricatactl filestore
# prune" command which can delete files over a certain age.
- file-store:
    version: 2
    enabled: no

    # Set the directory for the filestore. Relative pathnames
    # are contained within the "default-log-dir".
    #dir: filestore

    # Write out a fileinfo record for each occurrence of a file.
    # Disabled by default as each occurrence is already logged
    # as a fileinfo record to the main eve-log.
    #write-fileinfo: yes

    # Force storing of all files. Default: no.
    #force-filestore: yes

    # Override the global stream-depth for sessions in which we want
    # to perform file extraction. Set to 0 for unlimited; otherwise,
    # must be greater than the global stream-depth value to be used.
    #stream-depth: 0

    # Uncomment the following variable to define how many files can
    # remain open for filestore by Suricata. Default value is 0 which
    # means files get closed after each write to the file.
    #max-open-files: 1000
```

```
    # Force logging of checksums: available hash functions are md5,
    # sha1 and sha256. Note that SHA256 is automatically forced by
    # the use of this output module as it uses the SHA256 as the
    # file naming scheme.
    #force-hash: [sha1, md5]
    # NOTE: X-Forwarded configuration is ignored if write-fileinfo is disabled
    # HTTP X-Forwarded-For support by adding an extra field or overwriting
    # the source or destination IP address (depending on flow direction)
    # with the one reported in the X-Forwarded-For HTTP header. This is
    # helpful when reviewing alerts for traffic that is being reverse
    # or forward proxied.
    xff:
      enabled: no
      # Two operation modes are available, "extra-data" and "overwrite".
      mode: extra-data
      # Two proxy deployments are supported, "reverse" and "forward". In
      # a "reverse" deployment the IP address used is the last one, in a
      # "forward" deployment the first IP address is used.
      deployment: reverse
      # Header name where the actual IP address will be reported. If more
      # than one IP address is present, the last IP address will be the
      # one taken into consideration.
      header: X-Forwarded-For

# Log TCP data after stream normalization
# Two types: file or dir:
#     - file logs into a single logfile.
#     - dir creates 2 files per TCP session and stores the raw TCP
#           data into them.
# Use 'both' to enable both file and dir modes.
#
# Note: limited by "stream.reassembly.depth"
- tcp-data:
    enabled: no
    type: file
    filename: tcp-data.log

# Log HTTP body data after normalization, de-chunking and unzipping.
# Two types: file or dir.
#     - file logs into a single logfile.
#     - dir creates 2 files per HTTP session and stores the
#           normalized data into them.
# Use 'both' to enable both file and dir modes.
#
# Note: limited by the body limit settings
- http-body-data:
    enabled: no
    type: file
    filename: http-data.log

# Lua Output Support - execute lua script to generate alert and event
```

```
  # output.
  # Documented at:
  # https://docs.suricata.io/en/latest/output/lua-output.html
  - lua:
      enabled: no
      #scripts-dir: /etc/suricata/lua-output/
      scripts:
      #   - script1.lua

# Logging configuration.  This is not about logging IDS alerts/events, but
# output about what Suricata is doing, like startup messages, errors, etc.
logging:
  # The default log level: can be overridden in an output section.
  # Note that debug level logging will only be emitted if Suricata was
  # compiled with the --enable-debug configure option.
  #
  # This value is overridden by the SC_LOG_LEVEL env var.
  default-log-level: notice

  # The default output format.  Optional parameter, should default to
  # something reasonable if not provided.  Can be overridden in an
  # output section.  You can leave this out to get the default.
  #
  # This console log format value can be overridden by the SC_LOG_FORMAT env var.
  #default-log-format: "%D: %S: %M"
  #
  # For the pre-7.0 log format use:
  #default-log-format: "[%i] %t [%S] - (%f:%l) <%d> (%n) -- "

  # A regex to filter output.  Can be overridden in an output section.
  # Defaults to empty (no filter).
  #
  # This value is overridden by the SC_LOG_OP_FILTER env var.
  default-output-filter:

  # Requires libunwind to be available when Suricata is configured and built.
  # If a signal unexpectedly terminates Suricata, displays a brief diagnostic
  # message with the offending stacktrace if enabled.
  #stacktrace-on-signal: on

  # Define your logging outputs.  If none are defined, or they are all
  # disabled you will get the default: console output.
  outputs:
  - console:
      enabled: yes
      # type: json
  - file:
      enabled: yes
      level: info
      filename: suricata.log
      # format: "[%i - %m] %z %d: %S: %M"
      # type: json
```

```
  - syslog:
      enabled: no
      facility: local5
      format: "[%i] <%d> -- "
      # type: json
```

Suricata가 어떤 로그 파일을 만들고, 어떤 형식으로 데이터를 기록할지 결정하는 부분

기본 로그 디렉토리

default-log-dir: /var/log/suricata/

`default-log-dir: /var/log/suricata/`

- 설명: 로그 파일이 저장될 기본 디렉토리를 지정하는 부분. 특별한 설정이 없으면 이 경로에 로그 파일들이 저장.
- 디렉토리 변경: 기본적으로 설정된 경로를 변경하려면 이 값을 수정하면 된다. 예를 들어, 로그를 'home/user/suricata_logs/'에 저장하고 싶다면 'default-log-dir'을 '/home/user/suricata_logs/'로 설정하면 된다.
- 명령줄 파라미터: 'default-log-dir'으로 설정된 경로는 Suricata의 실행 시 명령줄의 '-l' 파라미터를 사용하여 다른 경로로 오버라이드할 수 있다. 예를 들어 Suricata를 실행할 때 '-l /path/to/other/log/dir'를 추가하면, 지정된 다른 디렉토리에 로그를 저장한다.

이 설정은 로그 파일이 적절한 위치에 저장되고, 파일 시스템의 경로를 일관되게 관리할 수 있도록 도와준다. Suricata의 로그는 네트워크 보안 모니터링 및 분석에 중요한 정보를 담고 있기 때문에, 로그 파일의 위치를 잘 관리하는 것이 중요하다.

통계 설정

```
# Global stats configuration
stats:
  enabled: yes
  # The interval field (in seconds) controls the interval at
  # which stats are updated in the log.
  interval: 8
  # Add decode events to stats.
  #decoder-events: true
  # Decoder event prefix in stats. Has been 'decoder' before, but that leads
  # to missing events in the eve.stats records. See issue #2225.
  #decoder-events-prefix: "decoder.event"
  # Add stream events as stats.
  #stream-events: false
```

`stats:`

- 설명: Suricata에서 수집한 통계 데이터를 로그에 기록할지 여부를 설정.

  `enabled: yes`로 설정하면 통계 로그가 활성화.

- **interval**: 통계를 갱신하는 주기를 초 단위로 설정. 여기서는 8초마다 통계가 업데이트돼.
- **decoder-events-prefix:** 디코더 이벤트의 접두사를 설정.
- **decoder-events**: 디코딩 중 발생한 이벤트를 통계에 추가할지 여부를 설정. 여기서는 주석 처리돼 있어 비활성 상태.
- **stream-events**: 스트림 이벤트를 통계에 포함할지 여부를 설정. 여기서는 주석 처리돼 있어 비활성 상태.

플러그인 설정

```
# Plugins -- Experimental -- specify the filename for each plugin shared object
plugins:
#   - /path/to/plugin.so
```

```
plugins:
```

- 설명: Suricata에서 사용할 플러그인 파일의 경로를 설정. 여기에 특정 플러그인 파일 경로를 추가하면 Suricata에서 해당 플러그인을 사용할 수 있다. 기본적으로 주석 처리되어 있어서 플러그인은 사용되지 않는다. 이 기능은 실험적이며 추가적인 기능을 제공할 수 있다.

로그 출력 설정

```
outputs:
```

```
# Configure the type of alert (and other) logging you would like.
outputs:
  # a line based alerts log similar to Snort's fast.log
  - fast:
      enabled: yes
      filename: fast.log
      append: yes
      #filetype: regular # 'regular', 'unix_stream' or 'unix_dgram'
```

- 설명: 이 부분에서 Suricata가 생성하는 다양한 종류의 로그 파일과 그 포맷을 설정할 수 있어.

1. **Fast** 로그 **(fast.log)**
   - `fast:`: 경량화된 경보 로그 파일로 Suricata의 경고를 라인 기반 로그로 기록, Snort의 fast.log와 유사한 형식. 알림을 빠르게 확인할 수 있는 형식으로 저장.
   - enabled: yes: Fast 로그를 활성화.
   - `filename: fast.log`: 이 로그 파일의 이름을 지정. 기본적으로 fast.log"로 저장.
   - append: yes: 기존 파일에 로그를 추가.
   - filetype: 파일 형식을 설정할 수 있다. 주석 처리된 부분에서는 'regular','unix_stream','unix_dgram' 중 하나를 선택할 수 있다.
2. **EVE** 로그 **(eve.json)**

```
 # Extensible Event Format (nicknamed EVE) event log in JSON format
  - eve-log:
      enabled: yes
```

```
filetype: regular #regular|syslog|unix_dgram|unix_stream|redis
filename: eve.json
# Enable for multi-threaded eve.json output; output files are amended with
# an identifier, e.g., eve.9.json
#threaded: false
#prefix: "@cee: " # prefix to prepend to each log entry
# the following are valid when type: syslog above
#identity: "suricata"
#facility: local5
#level: Info ## possible levels: Emergency, Alert, Critical,
            ## Error, Warning, Notice, Info, Debug
#ethernet: no  # log ethernet header in events when available
#redis:
#  server: 127.0.0.1
#  port: 6379
#  async: true ## if redis replies are read asynchronously
#  mode: list ## possible values: list|lpush (default), rpush, channel|publish
#             ## lpush and rpush are using a Redis list. "list" is an alias for lpush
#             ## publish is using a Redis channel. "channel" is an alias for publish
#  key: suricata ## key or channel to use (default to suricata)
# Redis pipelining set up. This will enable to only do a query every
# 'batch-size' events. This should lower the latency induced by network
# connection at the cost of some memory. There is no flushing implemented
# so this setting should be reserved to high traffic Suricata deployments.
#  pipelining:
#    enabled: yes ## set enable to yes to enable query pipelining
#    batch-size: 10 ## number of entries to keep in buffer

# Include top level metadata. Default yes.
#metadata: no

# include the name of the input pcap file in pcap file processing mode
pcap-file: false

# Community Flow ID
# Adds a 'community_id' field to EVE records. These are meant to give
# records a predictable flow ID that can be used to match records to
# output of other tools such as Zeek (Bro).
#
# Takes a 'seed' that needs to be same across sensors and tools
# to make the id less predictable.

# enable/disable the community id feature.
community-id: false
# Seed value for the ID output. Valid values are 0-65535.
community-id-seed: 0

# HTTP X-Forwarded-For support by adding an extra field or overwriting
# the source or destination IP address (depending on flow direction)
# with the one reported in the X-Forwarded-For HTTP header. This is
# helpful when reviewing alerts for traffic that is being reverse
# or forward proxied.
```

```
xff:
  enabled: no
  # Two operation modes are available: "extra-data" and "overwrite".
  mode: extra-data
  # Two proxy deployments are supported: "reverse" and "forward". In
  # a "reverse" deployment the IP address used is the last one, in a
  # "forward" deployment the first IP address is used.
  deployment: reverse
  # Header name where the actual IP address will be reported. If more
  # than one IP address is present, the last IP address will be the
  # one taken into consideration.
  header: X-Forwarded-For

types:
  - alert:
      # payload: yes             # enable dumping payload in Base64
      # payload-buffer-size: 4kb # max size of payload buffer to output in eve-log
      # payload-printable: yes   # enable dumping payload in printable (lossy) format
      # packet: yes              # enable dumping of packet (without stream segments)
      # metadata: no             # enable inclusion of app layer metadata with alert. Default yes
      # http-body: yes           # Requires metadata; enable dumping of HTTP body in Base64
      # http-body-printable: yes # Requires metadata; enable dumping of HTTP body in printable format

      # Enable the logging of tagged packets for rules using the
      # "tag" keyword.
      tagged-packets: yes
      # Enable logging the final action taken on a packet by the engine
      # (e.g: the alert may have action 'allowed' but the verdict be
      # 'drop' due to another alert. That's the engine's verdict)
      # verdict: yes
  # app layer frames
  - frame:
      # disabled by default as this is very verbose.
      enabled: no
  - anomaly:
      # Anomaly log records describe unexpected conditions such
      # as truncated packets, packets with invalid IP/UDP/TCP
      # length values, and other events that render the packet
      # invalid for further processing or describe unexpected
      # behavior on an established stream. Networks which
      # experience high occurrences of anomalies may experience
      # packet processing degradation.
      #
      # Anomalies are reported for the following:
      # 1. Decode: Values and conditions that are detected while
      # decoding individual packets. This includes invalid or
      # unexpected values for low-level protocol lengths as well
      # as stream related events (TCP 3-way handshake issues,
      # unexpected sequence number, etc).
      # 2. Stream: This includes stream related events (TCP
      # 3-way handshake issues, unexpected sequence number,
      # etc).
```

```
    # 3. Application layer: These denote application layer
    # specific conditions that are unexpected, invalid or are
    # unexpected given the application monitoring state.
    #
    # By default, anomaly logging is enabled. When anomaly
    # logging is enabled, applayer anomaly reporting is
    # also enabled.
    enabled: yes
    #
    # Choose one or more types of anomaly logging and whether to enable
    # logging of the packet header for packet anomalies.
    types:
      # decode: no
      # stream: no
      # applayer: yes
    #packethdr: no
- http:
    extended: yes     # enable this for extended logging information
    # custom allows additional HTTP fields to be included in eve-log.
    # the example below adds three additional fields when uncommented
    #custom: [Accept-Encoding, Accept-Language, Authorization]
    # set this value to one and only one from {both, request, response}
    # to dump all HTTP headers for every HTTP request and/or response
    # dump-all-headers: none
- dns:
    # This configuration uses the new DNS logging format,
    # the old configuration is still available:
    # https://docs.suricata.io/en/latest/output/eve/eve-json-output.html#dns-v1-format

    # As of Suricata 5.0, version 2 of the eve dns output
    # format is the default.
    #version: 2

    # Enable/disable this logger. Default: enabled.
    #enabled: yes

    # Control logging of requests and responses:
    # - requests: enable logging of DNS queries
    # - responses: enable logging of DNS answers
    # By default both requests and responses are logged.
    #requests: no
    #responses: no

    # Format of answer logging:
    # - detailed: array item per answer
    # - grouped: answers aggregated by type
    # Default: all
    #formats: [detailed, grouped]

    # DNS record types to log, based on the query type.
    # Default: all.
    #types: [a, aaaa, cname, mx, ns, ptr, txt]
```

```
    - tls:
        extended: yes     # enable this for extended logging information
        # output TLS transaction where the session is resumed using a
        # session id
        #session-resumption: no
        # ja4 hashes in tls records will never be logged unless
        # the following is set to on. (Default off)
        # ja4: off
        # custom controls which TLS fields that are included in eve-log
        #custom: [subject, issuer, session_resumed, serial, fingerprint, sni, version, not_before, not_after,
certificate, chain, ja3, ja3s, ja4]
    - files:
        force-magic: no   # force logging magic on all logged files
        # force logging of checksums, available hash functions are md5,
        # sha1 and sha256
        #force-hash: [md5]
    #- drop:
    #    alerts: yes      # log alerts that caused drops
    #    flows: all       # start or all: 'start' logs only a single drop
    #                     # per flow direction. All logs each dropped pkt.
        # Enable logging the final action taken on a packet by the engine
        # (will show more information in case of a drop caused by 'reject')
        # verdict: yes
    - smtp:
        #extended: yes # enable this for extended logging information
        # this includes: bcc, message-id, subject, x_mailer, user-agent
        # custom fields logging from the list:
        #  reply-to, bcc, message-id, subject, x-mailer, user-agent, received,
        #  x-originating-ip, in-reply-to, references, importance, priority,
        #  sensitivity, organization, content-md5, date
        #custom: [received, x-mailer, x-originating-ip, relays, reply-to, bcc]
        # output md5 of fields: body, subject
        # for the body you need to set app-layer.protocols.smtp.mime.body-md5
        # to yes
        #md5: [body, subject]
```

- `eve-log:`: 이 설정은 JSON 형식으로 다양한 이벤트를 기록하는 걸 담당. 로그 파일의 형식, 파일 이름, 기타 옵션들을 여기서 설정할 수 있다. Extensible Event Format (EVE) 이라고도 한다.
- enbaled: yes: EVE 로그를 활성화.
- `filetype:` 로그를 기록할 형식을 설정.('regular', 'syslog', 'unix_dgram', 'unix_stream', 'redis')
- `filename: eve.json`: JSON 로그 파일의 이름을 지정. 기본적으로 "eve.json"으로 저장.
- threaded: 다중 스레드 모드에서 EVE 로그를 출력할지 여부를 설정.
- prefix: 각 로그 항목에 추가할 접두사를 설정.
- identity, facility, level: syslog 관련 설정. syslog를 사용할 경우 유용함.
- ethernet: 이더넷 헤더를 로그에 포함할지 여부를 설정. 이 옵션을 yes로 설정하면 이더넷 헤더를 로그에 포함.
- redis: Redis를 사용하는 설정. Redis 서버의 주소와 포트, 비동기 처리 여부, 데이터 전송 모드 등을 설정할 수 있음.

    - pipelining: Redis에 대한 쿼리 파이프라이닝 설정.
    - metadata: EVE 로그에 메타데이터를 포함할지 여부를 설정.
    - pcap-file: pcap 파일을 처리할 때 파일 이름을 포함할지 여부를 설정.
    - **Community ID**: Community ID를 사용하여 로그 기록을 일관되게 할지 여부를 설정. 네트워크 흐름을 추적할 수 있는 고유한 ID를 기록할 수 있는 옵션. 기본적으로 비활성화돼 있음.
    - community-id-seed: Community ID의 시드를 설정.
    - **X-Forwarded-For (XFF)**: HTTP 헤더를 지원하여 IP 주소를 기록하는 방식. 프록시 서버를 통해 전달된 원래의 IP 주소를 기록할 수 있는 설정. 여기도 기본적으로 비활성화돼 있음.
3. **Alert** 로그 설정
    - `types:` 아래에 다양한 로그 종류를 설정할 수 있음. 여기서는 `alert`, `frame`, `anomaly`, `http`, `dns`, `tls` 등 여러 가지 유형의 로그가 활성화돼 있거나 설정되어 있음.
    - **Alert** 로그: 경고와 관련된 로그 설정으로, HTTP 바디를 Base64 형식으로 기록하거나 패킷 데이터를 기록하는 등의 세부 설정이 가능. 태그가 붙은 패킷을 기록하며, 패킷의 최종 행동을 로그할 수도 있음.
    - **Anomaly** 로그: 비정상적인 네트워크 동작을 기록하는 로그. 비활성화된 디코드, 스트림 등과 같은 특정 유형의 비정상 행동을 선택적으로 기록할 수 있음.
4. **HTTP** 로그
    - `http-log:`: HTTP 요청 및 응답에 대한 정보를 기록하는 로그. 이 설정에서는 비활성화돼 있음.
5. **TLS** 로그
    - `tls-log:`: TLS 핸드셰이크와 관련된 정보를 기록하는 로그. TLS 트랜잭션 정보를 기록. 역시 여기서는 비활성화 상태.
6. **PCAP** 로그
    - `pcap-log:`: 네트워크 패킷을 PCAP 형식으로 기록하는 설정. 기본적으로 이 기능도 꺼져 있음.
7. 기타 로그들
    - **Syslog**: 시스템 로그에 Suricata의 경고를 기록할 수 있는 설정이지만, 기본적으로 꺼져 있음.
    - **File-store**: 네트워크를 통해 전송된 파일을 디스크에 저장할 수 있는 설정. 여기도 비활성화돼 있음.

이 설정 파일을 통해 Suricata가 어떤 데이터를 기록하고, 어떤 형식으로 기록할지 세세하게 조정할 수 있어. 필요한 로그 종류에 따라 설정을 활성화하거나 비활성화하면 돼. 도움이 되었길 바라! 😊

#- dnp3
- ftp
- rdp
- nfs
- smb
- tftp
- ike
- dcerpc
- krb5
- bittorrent-dht
- snmp
- rfb
- sip

```
- quic:
    # ja4 hashes in quic records will never be logged unless
    # the following is set to on. (Default off)
    # ja4: off
- dhcp:
    enabled: yes
    # When extended mode is on, all DHCP messages are logged
    # with full detail. When extended mode is off (the
    # default), just enough information to map a MAC address
    # to an IP address is logged.
    extended: no
- ssh
- mqtt:
    # passwords: yes           # enable output of passwords
- http2
- pgsql:
    enabled: no
    # passwords: yes           # enable output of passwords. Disabled by default
- stats:
    totals: yes       # stats for all threads merged together
    threads: no       # per thread stats
    deltas: no        # include delta values
# bi-directional flows
- flow
# uni-directional flows
#- netflow

# Metadata event type. Triggered whenever a pktvar is saved
# and will include the pktvars, flowvars, flowbits and
# flowints.
#- metadata

# EXPERIMENTAL per packet output giving TCP state tracking details
# including internal state, flags, etc.
# This output is experimental, meant for debugging and subject to
# change in both config and output without any notice.
#- stream:
#    all: false                # log all TCP packets
#    event-set: false          # log packets that have a decoder/stream event
#    state-update: false       # log packets triggering a TCP state update
#    spurious-retransmission: false  # log spurious retransmission packets
```

네트워크 보안 시스템에서 사용하는 프로토콜들을 다루는 설정 부분. 보통 네트워크 보안 장비나 침입 탐지 시스템(IDS), 침입 방지 시스템(IPS)에서 사용하는 설정일 가능성이 높음. 각각의 항목들은 다양한 네트워크 프로토콜이나 기능들을 활성화하거나 비활성화하는 역할.

1. **DNP3** (Distributed Network Protocol 3):
    - 주로 전력 시스템에서 사용하는 통신 프로토콜. 스카다(SCADA) 시스템과 같은 산업 제어 시스템에서 많이 사용.
2. **FTP** (File Transfer Protocol):

    - 파일을 서버와 클라이언트 간에 전송하는 데 사용되는 기본적인 파일 전송 프로토콜. 하지만 보안이 취약해서 일반적으로 FTP보다는 SFTP(SSH를 사용하는 FTP)가 더 많이 사용.
3. **RDP** (Remote Desktop Protocol):
    - 원격 데스크탑 연결을 통해 다른 컴퓨터에 접속해서 그 컴퓨터를 제어할 수 있는 프로토콜. 주로 윈도우에서 사용.
4. **NFS** (Network File System):
    - 네트워크를 통해 파일을 공유하는 프로토콜. 주로 유닉스와 리눅스 시스템에서 사용.
5. **SMB** (Server Message Block):
    - 윈도우 시스템에서 네트워크 파일 공유에 사용되는 프로토콜. 윈도우 환경에서 프린터 공유나 파일 공유에 많이 사용.
6. **TFTP** (Trivial File Transfer Protocol):
    - FTP보다 더 단순한 파일 전송 프로토콜. 보안이나 인증이 필요 없는 단순한 파일 전송에 사용. 주로 네트워크 장비의 설정 파일을 전송할 때 많이 사용.
7. **IKE** (Internet Key Exchange):
    - VPN(가상 사설망)에서 사용되는 프로토콜로, 암호화 키 교환을 관리해주는 역할.
8. **DCERPC** (Distributed Computing Environment / Remote Procedure Calls):
    - 윈도우 환경에서 프로세스 간 통신을 위해 사용되는 프로토콜. 네트워크 상의 다른 장치나 프로그램과 통신하기 위해 사용.
9. **KRB5** (Kerberos version 5):
    - 네트워크 인증 프로토콜로, 특히 안전한 인증이 필요한 환경에서 많이 사용. 보안된 네트워크에서 사용자를 인증하기 위해 사용.
10. **BitTorrent-DHT**:
    - P2P(Peer-to-Peer) 네트워크에서 사용하는 프로토콜로, BitTorrent에서 사용. DHT(Distributed Hash Table)는 피어 간의 연결을 관리하고, 파일 조각을 찾는 데 사용.
11. **SNMP** (Simple Network Management Protocol):
    - 네트워크 장치들을 모니터링하고 관리하기 위해 사용되는 프로토콜. 주로 네트워크 관리자들이 장치 상태를 체크하고, 설정을 변경할 때 사용.
12. **RFB** (Remote Framebuffer Protocol):
    - VNC(Virtual Network Computing)에서 사용하는 프로토콜로, 원격 화면을 제어할 때 사용. 원격 데스크탑과 비슷한 역할.
13. **SIP** (Session Initiation Protocol):
    - VoIP(Voice over IP) 통신에서 사용되는 프로토콜로, 음성 및 영상 통화를 설정하고 종료하는 역할.
14. **QUIC** (Quick UDP Internet Connections):
    - 구글이 개발한 새로운 전송 프로토콜로, 기존 TCP보다 더 빠르고 효율적인 통신을 목표로 한다. 여기서 `ja4` 해시는 QUIC 프로토콜에 관련된 보안 해시인데, 이 기능은 기본적으로 꺼져 있음.
15. **DHCP** (Dynamic Host Configuration Protocol):
    - 네트워크에서 IP 주소를 자동으로 할당해주는 프로토콜. 이 설정에서는 `extended` 모드가 꺼져 있어서 기본적인 정보(IP와 MAC 주소 매핑)만 로그에 기록되게 되어 있음.
16. **SSH** (Secure Shell):
    - 원격 서버에 안전하게 접속할 때 사용하는 프로토콜. 터미널을 통해 서버 관리 작업을 할 때 주로 사용.
17. **MQTT** (Message Queuing Telemetry Transport):
    - 사물 인터넷(IoT) 환경에서 많이 사용되는 메시지 전송 프로토콜. 여기서는 `passwords` 설정이 주석 처리되어 있어서, 비밀번호가 로그에 기록되지 않게 돼 있음.

18. **HTTP2** (Hypertext Transfer Protocol/2):
    - 웹 브라우징에서 사용되는 최신 HTTP 프로토콜 버전. 기존 HTTP보다 더 빠르고 효율적인 데이터 전송을 가능하게 함.
19. **PGSQL** (PostgreSQL):
    - 오픈 소스 관계형 데이터베이스 시스템. 여기서는 `enabled`가 `no`로 설정되어 있어서, 이 프로토콜은 비활성화 상태.
20. **Stats**:
    - 시스템에서 통계 정보를 로그로 남기는 기능. `totals`가 `yes`로 설정되어 있어서 모든 스레드의 통계가 병합된 정보를 기록. 반면에, 각 스레드별 통계(`threads`)나 델타 값(`deltas`)은 기록되지 않음.

추가 기능들**:**

1. **Flow**:
    - 양방향 흐름을 추적하고, 이를 로그로 남기는 기능.
2. **NetFlow**:
    - 단방향 흐름을 추적하는 기능인데, 여기서는 주석 처리되어 있어서 비활성화 상태.
3. **Metadata**:
    - 패킷 변수를 저장할 때 발생하는 메타데이터 이벤트를 로그로 남기는 기능. 주석 처리된 상태로 비활성화되어 있음.
4. **Stream**:
    - TCP 상태 추적 정보를 기록하는 실험적인 기능. 주석 처리되어 있어서 사용되지 않고 있음.

이 파일에서 각각의 프로토콜이나 기능이 어떻게 설정되고, 어떤 역할을 하는지 파악하는 게 중요한데, 네트워크 보안을 관리하는 시스템에서 각 프로토콜의 로그 기록 방법이나 설정을 조정하는 것에 대해 다루고 있다. 필요한 부분을 활성화하거나 세부적으로 조정할 수 있다.

```
 # a line based log of HTTP requests (no alerts)
- http-log:
    enabled: no
    filename: http.log
    append: yes
    #extended: yes     # enable this for extended logging information
    #custom: yes       # enable the custom logging format (defined by customformat)
    #customformat: "%{%D-%H:%M:%S}t.%z %{X-Forwarded-For}i %H %m %h %u %s %B %a:%p -> %A:%P"
    #filetype: regular # 'regular', 'unix_stream' or 'unix_dgram'
```

HTTP 요청을 기록하는 로그 기능에 대한 설정. 네트워크 보안 시스템이나 웹 서버에서 HTTP 요청에 대한 로그를 남길 때 사용될 수 있음. 이 설정은 기본적으로 로그 파일을 어떻게 생성하고, 어떤 정보를 기록할지 정의하는 부분.

1. **http-log**:
    - HTTP 요청을 기록하는 로그 기능을 설정. 이 로그는 보안 경고 없이 단순히 HTTP 요청의 기록을 남기는 데 사용.
2. **enabled: no**:

    - 이 설정은 기본적으로 `no`로 설정되어 있어 HTTP 로그 기능이 비활성화되어 있다는 뜻. 만약 HTTP 요청 로그를 기록하고 싶다면, 이 값을 `yes`로 변경해야 함.
3. **filename: http.log**:
    - 로그 파일의 이름을 지정하는 부분. 여기서는 로그 파일이 `http.log`라는 이름으로 저장되도록 설정되어 있음.
4. **append: yes**:
    - 기존 로그 파일이 있을 경우, 새로운 로그를 추가로 기록할 때 파일을 덮어쓰지 않고, 기존 파일의 끝에 이어서 기록하게 됨. 이 옵션이 `yes`로 설정되어 있어서 기존 로그에 계속해서 새로운 로그를 덧붙이는 방식으로 작동.
5. **extended: yes** (주석 처리됨):
    - HTTP 요청의 확장 정보를 기록할지 여부를 설정. 이 옵션은 주석 처리되어 있어서 기본적으로 비활성화 상태. 만약 활성화하면, 기본 로그보다 더 많은 정보가 포함된 확장된 로그를 기록.
6. **custom: yes** (주석 처리됨):
    - 사용자 정의 형식으로 로그를 기록할지 여부를 설정.이 설정도 주석 처리되어 있어서 비활성화 상태.
7. **customformat: "%{%D-%H:%M:%S}t.%z %{X-Forwarded-For}i %H %m %h %u %s %B %a:%p -> %A:%P"** (주석 처리됨):
    - 사용자 맞춤형 로그 형식을 정의하는 부분. 여기 정의된 형식은 시간, HTTP 메서드, 요청한 호스트, 상태 코드, 바이트 크기 등과 같은 다양한 정보를 포함. 이 부분이 주석 처리되어 있어서, 현재는 사용되지 않고 있음.
8. **filetype: regular** (주석 처리됨):
    - 로그 파일의 타입을 정의하는 설정. 여기서 `regular`는 일반적인 파일 형식으로 로그를 저장하는 걸 의미. 다른 옵션으로는 `unix_stream`이나 `unix_dgram`이 있는데, 이는 Unix 기반 시스템에서 스트림이나 데이터그램 소켓으로 로그를 전송하는 방법을 의미. 이 부분도 주석 처리되어 있어서 비활성화 상태.

요약**:**

- 기본적으로 이 설정 파일에서는 HTTP 로그 기능이 비활성화(`enabled: no`)되어 있다.
- 로그 파일은 `http.log`라는 이름으로 저장되며, 기존 로그에 새로운 로그가 덧붙여져(`append: yes`) 기록된다.
- 확장된 로그 정보나 맞춤형 로그 형식은 주석 처리되어 있어서 현재는 사용되지 않지만, 필요에 따라 활성화할 수 있다.

이 설정을 통해 HTTP 요청에 대한 상세한 기록을 남길 수 있지만, 보안 경고는 포함되지 않아서 주로 트래픽 분석이나 문제 해결을 위한 용도로 사용될 수 있다.

```
# a line based log of TLS handshake parameters (no alerts)
 - tls-log:
    enabled: no  # Log TLS connections.
    filename: tls.log # File to store TLS logs.
    append: yes
    #extended: yes     # Log extended information like fingerprint
    #custom: yes       # enabled the custom logging format (defined by customformat)
```

```
#customformat: "%{%D-%H:%M:%S}t.%z %a:%p -> %A:%P %v %n %d %D"
#filetype: regular # 'regular', 'unix_stream' or 'unix_dgram'
# output TLS transaction where the session is resumed using a
# session id
#session-resumption: no
```

TLS(Transport Layer Security) 핸드셰이크 과정에 대한 로그를 기록하는 기능. TLS는 네트워크에서 데이터를 암호화하고 안전하게 전송하기 위해 사용하는 프로토콜. TLS 연결의 세부 정보를 로그로 남길 때 어떻게 저장할지 정의.

1. **tls-log**:
   - TLS 핸드셰이크(클라이언트와 서버 간의 보안 연결 설정 과정)에 대한 로그를 관리하는 기능을 담당. 이 로그는 보안 경고 없이, TLS 연결에 대한 정보를 단순히 기록하는 데 사용.
2. **enabled: no**:
   - 기본적으로 `no`로 설정되어 있어서, TLS 로그 기능이 비활성화되어 있음. 만약 TLS 연결에 대한 로그를 남기고 싶다면, 이 값을 `yes`로 변경.
3. **filename: tls.log**:
   - 로그 파일의 이름을 지정하는 부분. 여기서는 `tls.log`라는 이름으로 로그 파일이 생성되도록 설정되어 있음.
4. **append: yes**:
   - 기존 로그 파일이 있을 경우, 새로운 로그를 추가로 기록할 때 파일을 덮어쓰지 않고, 기존 파일의 끝에 이어서 기록하게 됨. 이 설정이 `yes`로 설정되어 있어서, 로그가 추가 기록되는 방식으로 작동.
5. **extended: yes** (주석 처리됨):
   - TLS 핸드쉐이크의 확장 정보를 기록할지 여부를 설정. 이 옵션이 활성화되면, 기본 로그보다 더 많은 정보(예: TLS 핸드셰이크에서 사용하는 인증서의 지문 정보 등)가 포함된 확장된 로그를 기록. 하지만 주석 처리되어 있어서, 현재는 사용되지 않고 있음.
6. **custom: yes** (주석 처리됨):
   - 맞춤형 로그 형식을 사용하기 위한 옵션. 활성화하면, 사용자가 정의한 형식으로 로그를 기록. 현재는 주석 처리되어 있어서 기본 로그 형식을 사용하고 있음.
7. **customformat: "%{%D-%H:%M:%S}t.%z %a:%p -> %A:%P %v %n %d %D"** (주석 처리됨):
   - 이 부분은 맞춤형 로그 형식을 정의하는 설정. 시간, 클라이언트와 서버의 IP와 포트, 서버 이름, 인증서 정보 등 다양한 정보를 포함할 수 있음. 주석 처리되어 있어서 현재는 사용되지 않고 있음.
8. **filetype: regular** (주석 처리됨):
   - 로그 파일의 타입을 정의하는 설정. `regular`는 일반적인 파일 형식으로 로그를 저장하는 걸 의미. 다른 옵션으로는 `unix_stream`이나 `unix_dgram`이 있는데, 이는 Unix 기반 시스템에서 스트림이나 데이터그램 소켓으로 로그를 전송하는 방식. 주석 처리되어 있음.
9. **session-resumption: no** (주석 처리됨):
   - TLS 연결에서는 세션 재개 기능이 있는데, 이는 이전에 설정된 TLS 세션을 재활용해 새로운 핸드셰이크 과정을 줄이는 기능. 이 옵션을 `yes`로 설정하면, 세션 재개가 이루어진 경우에도 해당 세션 정보를 로그에 기록. 현재는 `no`로 설정되어 있어서 세션 재개 로그를 기록하지 않음.

요약:

- 이 설정 파일은 TLS 핸드셰이크에 대한 로그를 기록하는 기능을 다루고 있지만, 기본적으로 비활성화(`enabled: no`) 상태이다.
- 로그 파일 이름은 `tls.log`로 설정되어 있으며, 기존 로그에 추가 기록(`append: yes`)하는 방식으로 작동.
- 확장된 정보나 맞춤형 로그 형식은 주석 처리되어 있어서 사용되지 않고 있다.
- TLS 세션 재개 정보는 현재 로그에 기록되지 않지만, 필요하면 `session-resumption` 설정을 변경해 활성화할 수 있다.

이 설정을 통해 TLS 연결에 대한 다양한 정보를 로그로 남길 수 있는데, 이를 통해 보안 관련 문제를 분석하거나 트래픽을 모니터링할 수 있다.

```
# output module to store certificates chain to disk
- tls-store:
    enabled: no
    #certs-log-dir: certs # directory to store the certificates files

# Packet log... log packets in pcap format. 3 modes of operation: "normal"
# "multi" and "sguil".
#
# In normal mode a pcap file "filename" is created in the default-log-dir,
# or as specified by "dir".
# In multi mode, a file is created per thread. This will perform much
# better, but will create multiple files where 'normal' would create one.
# In multi mode the filename takes a few special variables:
# - %n -- thread number
# - %i -- thread id
# - %t -- timestamp (secs or secs.usecs based on 'ts-format'
# E.g. filename: pcap.%n.%t
#
# Note that it's possible to use directories, but the directories are not
# created by Suricata. E.g. filename: pcaps/%n/log.%s will log into the
# per thread directory.
#
# Also note that the limit and max-files settings are enforced per thread.
# So the size limit when using 8 threads with 1000mb files and 2000 files
# is: 8*1000*2000 ~ 16TiB.
#
# In Sguil mode "dir" indicates the base directory. In this base dir the
# pcaps are created in the directory structure Sguil expects:
#
# $sguil-base-dir/YYYY-MM-DD/$filename.<timestamp>
#
# By default all packets are logged except:
# - TCP streams beyond stream.reassembly.depth
# - encrypted streams after the key exchange
#
- pcap-log:
    enabled: no
    filename: log.pcap

    # File size limit.  Can be specified in kb, mb, gb.  Just a number
    # is parsed as bytes.
```

```
    limit: 1000mb

    # If set to a value, ring buffer mode is enabled. Will keep maximum of
    # "max-files" of size "limit"
    max-files: 2000

    # Compression algorithm for pcap files. Possible values: none, lz4.
    # Enabling compression is incompatible with the sguil mode. Note also
    # that on Windows, enabling compression will *increase* disk I/O.
    compression: none

    # Further options for lz4 compression. The compression level can be set
    # to a value between 0 and 16, where higher values result in higher
    # compression.
    #lz4-checksum: no
    #lz4-level: 0

    mode: normal # normal, multi or sguil.

    # Directory to place pcap files. If not provided the default log
    # directory will be used. Required for "sguil" mode.
    #dir: /nsm_data/

    #ts-format: usec # sec or usec second format (default) is filename.sec usec is filename.sec.usec
    use-stream-depth: no #If set to "yes" packets seen after reaching stream inspection depth are ignored. "no"
logs all packets
    honor-pass-rules: no # If set to "yes", flows in which a pass rule matched will stop being logged.
    # Use "all" to log all packets or use "alerts" to log only alerted packets and flows or "tag"
    # to log only flow tagged via the "tag" keyword
    #conditional: all

  # a full alert log containing much information for signature writers
  # or for investigating suspected false positives.
  - alert-debug:
      enabled: no
      filename: alert-debug.log
      append: yes
      #filetype: regular # 'regular', 'unix_stream' or 'unix_dgram'

  # Stats.log contains data from various counters of the Suricata engine.
  - stats:
      enabled: yes
      filename: stats.log
      append: yes       # append to file (yes) or overwrite it (no)
      totals: yes       # stats for all threads merged together
      threads: no       # per thread stats
      #null-values: yes  # print counters that have value 0. Default: no

  # a line based alerts log similar to fast.log into syslog
  - syslog:
      enabled: no
      # reported identity to syslog. If omitted the program name (usually
```

```
# suricata) will be used.
#identity: "suricata"
facility: local5
#level: Info ## possible levels: Emergency, Alert, Critical,
        ## Error, Warning, Notice, Info, Debug
```

1. **tls-store**: TLS 인증서를 디스크에 저장하는 기능을 설정.
    - **enabled: no**: TLS(Transport Layer Security) 인증서 체인을 디스크에 저장하는 기능을 다루는 설정. `enabled: no`로 설정되어 있어 이 기능이 비활성화된 상태.
    - **certs-log-dir: certs** (주석 처리됨): 인증서 파일을 저장할 디렉터리를 설정. 주석 처리되어 있어서 현재는 이 디렉터리가 사용되지 않음.
2. **pcap-log**: 패킷 데이터를 PCAP(패킷 캡처 파일) 형식으로 저장하는 기능. PCAP 파일은 네트워크 트래픽 분석에 사용되는 표준 형식.
    - **enabled: no**: 패킷 로그 기능이 비활성화된 상태. 활성화하려면 `yes`로 변경.
    - **filename: log.pcap**: 패킷 로그 파일의 기본 이름을 지정하는 부분. 기본적으로 `log.pcap`라는 이름으로 저장되도록 설정되어 있음.
    - **limit: 1000mb**: 생성되는 PCAP 파일의 최대 크기를 1000MB로 제한하는 설정. 이 크기를 초과하면 새로운 파일이 생성됨. 단위는 kb, mb, gb로 지정 가능.
    - **max-files: 2000**: 최대 2000개의 파일까지만 저장되도록 설정하는 부분. 파일 개수가 이 제한을 넘으면, 가장 오래된 파일부터 삭제됨.
    - **compression: none**: pcap 파일의 압축 알고리즘을 설정. 가능한 값은 'none'(압축 없음), 'lz4'(LZ4 압축).
    - **mode: normal**: 로그 저장 모드를 설정하는 부분. `normal` 모드에서는 단일 pcap 파일에 모든 패킷을 기록, `multi` 모드에서는 각 스레드마다 별도의 파일을 생성해 성능을 향상시킬 수 있음. `sguil` 모드는 Sguil이라는 특정 침입 탐지 시스템에서 사용하는 디렉터리 구조를 생성해.
    - **dir:** Sguil 시스템에 맞는 디렉토리 구조를 사용하여 pcap 파일을 기록
    - **ts-format: usec:** 타임스탬프 형식을 설정. sec(초 단위) 또는 usec(마이크로초 단위) 중 선택할 수 있음.
    - **use-stream-depth: no**: 스트림 검사 깊이를 초과한 패킷을 무시할지 여부를 설정. `no`로 설정되면, 스트림의 최대 깊이에 도달한 후에도 모든 패킷을 계속 기록.
    - **honor-pass-rules: no**: pass 규칙이 일치하는 흐름을 로그에서 제외할지 여부를 설정. `yes`로 설정되면, 패스 규칙(특정 트래픽을 무시하는 규칙)이 적용된 흐름에 대한 로그 기록이 중단.
    - **conditional: all** (주석 처리됨): 어떤 패킷을 기록할지 설정하는 부분. `all`은 모든 패킷을 기록하고, `alerts`는 경고가 발생한 패킷 및 흐름, `tag`는 태그가 붙은 흐름.
3. **alert-debug**:
    - 서명 작성자나 오탐지(잘못된 경고)를 조사할 때 사용할 수 있는 상세한 알림 로그를 기록하는 기능.(디버깅용 경고 로그를 설정)
    - **enabled: no**: 이 기능이 비활성화되어 있는 상태. 활성화하려면 `yes`로 변경.
    - **filename: alert-debug.log**: 디버깅 경고 로그 파일의 이름을 설정.
    - **append: yes**: 로그를 덮어쓰지 않고 기존 파일 끝에 추가로 기록.
    - **filetype:** 파일 형식을 설정(**'regular', 'unix_stream', 'unix_dgram'**).
4. **stats**: Suricata 엔진의 통계 데이터를 기록하는 로그를 설정.
    - **enabled: yes**: Suricata 엔진의 다양한 카운터 데이터를 기록하는 기능 활성화.
    - **filename: stats.log**: 통계 데이터를 기록할 파일 이름을 지정.

- **append: yes**: 기존 파일에 추가로 기록하도록 설정.
- **totals: yes**: 모든 스레드의 통계 데이터를 병합하여 기록.
- **threads: no**: 각 스레드별로 개별적인 통계를 기록할지 여부를 설정.
- **null-values: yes** (주석 처리됨): 값이 0인 카운터도 기록할지 여부를 설정하는 부분. 현재는 비활성화 상태.

5. **syslog**:
   - `syslog`를 사용해 시스템 로그로 알림을 보내는 기능.
   - **enabled: no**: `syslog`를 통한 로그 전송 기능이 비활성화된 상태.
   - **identity: "suricata"** (주석 처리됨): `syslog`에서 로그를 전송할 때 사용될 프로그램 이름을 지정하는 부분. 이 부분이 주석 처리되어 있으면 기본적으로 `suricata`라는 이름을 사용.
   - **facility: local5**: syslog에 보고할 프로그램 이름을 설정. 로그의 분류를 지정. `local5`는 로컬 로그 서버로 로그를 보내는 데 사용.
   - **level: Info** (주석 처리됨): 로그 수준을 설정. 가능한 값은 'Emergency', 'Alert', 'Critical', 'Error', 'Warning', 'Notice', 'Info', 'Debug'. `syslog`에서 로그의 심각도 수준을 설정하는 부분. `Info`는 정보 수준의 로그만 전송되도록 설정.

요약**:**

이 설정 파일은 TLS 인증서 체인 저장, 패킷 로그, 알림 디버그, 통계 로그, 그리고 `syslog`를 통한 로그 전송 기능을 다루고 있음. 기본적으로 대부분의 기능이 비활성화된 상태지만, 필요한 경우 활성화해 네트워크 트래픽이나 보안 경고를 더욱 자세히 분석할 수 있음. 각 기능의 활성화 여부와 설정을 통해 로그 파일의 형식, 저장 위치, 기록 조건 등을 세밀하게 조정할 수 있음.

```
# Output module for storing files on disk. Files are stored in
 # directory names consisting of the first 2 characters of the
 # SHA256 of the file. Each file is given its SHA256 as a filename.
 #
 # When a duplicate file is found, the timestamps on the existing file
 # are updated.
 #
 # Unlike the older filestore, metadata is not written by default
 # as each file should already have a "fileinfo" record in the
 # eve-log. If write-fileinfo is set to yes, then each file will have
 # one more associated .json files that consist of the fileinfo
 # record. A fileinfo file will be written for each occurrence of the
 # file seen using a filename suffix to ensure uniqueness.
 #
 # To prune the filestore directory see the "suricatactl filestore
 # prune" command which can delete files over a certain age.
 - file-store:
     version: 2
     enabled: no
```