

내가 지금 수리카타를 해야하는데 나 진짜 너무 하나도 모르겠어서
네가 도와줘 ㅠㅠㅠㅠ
vi /etc/suricata/suricata.yaml 이거를 공부하라는데 알려줄수있어?

물론이죠! Suricata는 네트워크 보안 모니터링 및 침입 탐지를 위한 오픈 소스 도구입니다.
`suricata.yaml` 파일은 Suricata의 설정 파일로, 이 파일을 통해 Suricata의 동작 방식을 구성할 수 있습니다.

`vi /etc/suricata/suricata.yaml` 명령어는 `vi` 텍스트 편집기로 `suricata.yaml` 파일을 열어 편집하겠다는 뜻입니다. 이 파일을 이해하고 수정하기 위해 기본적인 내용을 살펴보겠습니다.

suricata.yaml 파일의 주요 섹션

1. 기본 설정:



```
yaml
# 기본 경로 및 로그 설정
default-rule-path: /etc/suricata/rules
```

더 스마트한 응답, 파일 및 이미지 업로드 등을 이용하세요.

- `default-rule-path`는 기본 룰 파일이 위치한 경로를 설정합니다.
- `rule-files`는 사용할 룰 파일을 지정합니다.

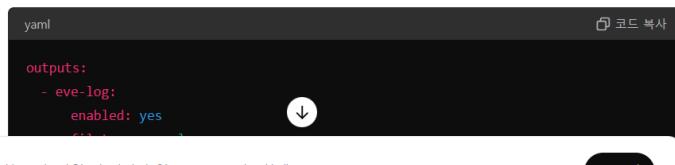
2. 네트워크 인터페이스:



```
yaml
af-packet:
- interface: eth0
```

- `af-packet` 섹션은 패킷 캡처를 위한 네트워크 인터페이스를 설정합니다. 여기서 `eth0`는 네트워크 인터페이스의 이름입니다.

3. 출력 설정:



```
yaml
outputs:
- eve-log:
  enabled: yes
```

더 스마트한 응답, 파일 및 이미지 업로드 등을 이용하세요.

- `outputs` 섹션은 Suricata의 로그 출력 형식과 위치를 설정합니다. `eve-log`는 JSON 형식으로 로그를 기록하며, `/var/log/suricata/eve.json`에 저장됩니다.

4. 네트워크 설정:



```
yaml
# IP, 포트 및 서브넷 설정
host-os:
- host: 192.168.1.1
```

- `host-os` 섹션은 IP 주소나 서브넷을 설정하여 감시할 호스트를 정의합니다.

5. 룰 설정:



```
yaml
rule-files:
- my-custom-rules.rules
```

- `rule-files` 섹션은 사용자 정의 룰 파일을 지정할 수 있습니다. 이 룰 파일들은 Suricata

더 스마트한 응답, 파일 및 이미지 업로드 등을 이용하세요.

로그인

- 파일 열기: `vi /etc/suricata/suricata.yaml`을 입력합니다.
- 편집 모드로 전환: `i`를 눌러 편집 모드로 들어갑니다.
- 편집 완료 후 저장: `Esc`를 누른 다음 `:wq`를 입력하고 `Enter`를 눌러 저장하고 종료합니다.
- 저장하지 않고 종료: `Esc`를 누른 다음 `:q!`를 입력하고 `Enter`를 눌러 변경 사항 없이 종료합니다.

이제 `suricata.yaml` 파일의 구조와 기본 설정을 이해하셨을 텐데요, 파일을 수정하면서 각 섹션의 의미를 기억하고 적절한 값을 설정하는 것이 중요합니다. 만약 구체적인 설정에 대한 질문이 있다면, 추가로 도와드릴 수 있습니다!

vi /etc/suricata/suricata.yaml 해서 나오는 내용들에 대해서도 설명해 주라

네, `vi /etc/suricata/suricata.yaml` 명령어로 파일을 열면, `suricata.yaml` 파일의 다양한 설정

더 스마트한 응답, 파일 및 이미지 업로드 등을 이용하세요.

로그인

1. 기본 정보

```
yaml
```

```
# Suricata YAML configuration file
# For more details, see https://suricata.readthedocs.io
```

코드 복사

- 파일의 맨 위에는 주석으로 이 파일이 Suricata의 설정 파일임을 알고자, 자세한 내용은 Suricata 문서를 참조하라는 안내가 있습니다.

2. 기본 설정

```
yaml
```

```
# Configure Suricata to run in a specific mode.
# By default, Suricata runs in IDS mode, which means it will only
# inspect traffic and generate alerts.

# Run mode (IDS, IPS, or INLINE)
runmode: auto
```

코드 복사

더 스마트한 응답, 파일 및 이미지 업로드 등을 이용하세요.

로그인

실행됩니다. `auto`, `ids`, `ips`, `inline` 중 하나를 선택할 수 있습니다.

3. 네트워크 인터페이스

```
yaml
```

```
af-packet:
  - interface: eth0
    threads: 4
```

코드 복사

- `af-packet` 섹션은 패킷 캡처를 위한 네트워크 인터페이스를 설정합니다.
- `interface`는 패킷을 수집할 네트워크 인터페이스를 지정합니다 (`eth0`).
- `threads`는 패킷 캡처를 처리할 스레드 수를 설정합니다.

4. 로그 및 출력 설정

```
yaml
```

```
outputs:
  eve-log:
    type: regular
    filename: /var/log/suricata/eve.json
    pcre:
      enabled: yes
```

코드 복사

더 스마트한 응답, 파일 및 이미지 업로드 등을 이용하세요.

로그인

- `outputs` 섹션은 Suricata의 로그 출력을 설정합니다.
- `eve-log`는 JSON 형식으로 로그를 저장할 수 있게 해줍니다.
- `enabled`가 `yes`로 설정되어 있으면 로그 기록이 활성화됩니다.
- `filename`은 로그 파일의 경로를 설정합니다 (`/var/log/suricata/eve.json`).
- `pcre`는 Perl Compatible Regular Expressions 사용 여부를 설정합니다.

5. 규칙 및 룰 설정

```
yaml
```

```
rule-files:
  - suricata.rules
  - /etc/suricata/rules/my-rules.rules
```

코드 복사

더 스마트한 응답, 파일 및 이미지 업로드 등을 이용하세요.

로그인

- 여러 개의 룰 파일을 추가할 수 있습니다. 각 룰 파일은 네트워크 트래픽을 분석하는 데 필요한 규칙을 포함합니다.

6. 패턴 및 정규 표현식

```
yaml
```

코드 복사

```
# PCRE (Perl Compatible Regular Expressions) configuration
pcre:
  # Enable/Disable pcre matching
  enabled: yes
  # Extra options for pcre
  default:
    # Set to yes to enable pcre
    pcre:
      enabled: yes
```

- `pcre` 섹션은 정규 표현식 관련 설정을 ↓ 의합니다.

더 스마트한 응답, 파일 및 이미지 업로드 등을 이용하세요.

로그인

```
yaml
# SSL/TLS inspection configuration
ssl:
  # Enable/disable SSL/TLS inspection
  enabled: yes
  # Path to the CA cert for SSL/TLS inspection
  cert-file: /etc/suricata/certs/suricata.pem
```

- `ssl` 섹션은 SSL/TLS 트래픽 검사 설정을 포함합니다.
- `enabled` 가 `yes` 로 설정되면 SSL/TLS 트래픽 검사가 활성화됩니다.
- `cert-file` 은 SSL/TLS 검사를 위한 인증서 파일의 경로를 지정합니다.

8. 사용자 정의 설정

```
yaml
# Custom settings can be added here ↓
```

더 스마트한 응답, 파일 및 이미지 업로드 등을 이용하세요.

로그인

- `custom-settings` 는 사용자 정의 설정을 추가할 수 있는 공간입니다.
- `custom-rule-file` 은 사용자 정의 규칙 파일의 경로를 설정합니다.

이와 같은 내용으로 `suricata.yaml` 파일은 Suricata의 다양한 동작 및 설정을 정의합니다. 파일 내 주석을 통해 각 설정의 의미를 이해하고, 필요한 설정을 조정하여 Suricata의 동작을 맞춤화할 수 있습니다.

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 ↓ ET: "[192.168.0.0/16]"
  #HOME_NFT: "[10.0.0.0/8]"
```

더 스마트한 응답, 파일 및 이미지 업로드 등을 이용하세요.

로그인

```
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_f ↓ FS: "80"
```

더 스마트한 응답, 파일 및 이미지 업로드 등을 이용하세요.

로그인

```
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  
설명해주세요
```

이 부분은 `suricata.yaml` 설정 파일의 `vars` 섹션으로, Suricata가 네트워크를 어떻게 인식하고 분석할지에 대한 정보를 설정하는 부분입니다. 구체적으로는 네트워크 주소와 포트를 정의하여 Suricata의 규칙이 이들에 맞춰 적용될 수 있도록 합니다. 각각의 설정 항목에 대해 설명해 드리겠습니다.

1. 주소 그룹 (Address Groups)

이 설정은 Suricata가 네트워크 트래픽을 분석할 때 사용할 주소 그룹을 정의합니다.

```
yaml  
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"
```

- `HOME_NET`: 자신의 네트워크를 정의합니다. 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.

주석 처리된 다른 예제들은 특정 서브넷만 포함하거나, `any`를 사용하여 모든 IP 주소를 허용할 수 있습니다.

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

• `EXTERNAL_NET`은 `HOME_NET`과 반대되는 외부 네트워크를 정의합니다. `!\$HOME_NET`는

더 스마트한 응답, 파일 및 이미지 업로드 등을 이용하세요.

```
yaml  
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"
```

- 이러한 항목들은 특정 서비스나 프로토콜에 대한 서버와 클라이언트를 정의합니다. 예를 들어:
• `HTTP_SERVERS`, `SMTP_SERVERS`, `SQL_SERVERS` 등은 `HOME_NET`에 포함된 IP 주소를 해당

더 스마트한 응답, 파일 및 이미지 업로드 등을 이용하세요.

합니다.

- `DNP3_SERVER`, `MODBUS_SERVER` 등은 각각 DNP3 및 MODBUS 프로토콜을 사용하는 서버의 IP 주소를 `HOME_NET`으로 설정합니다.

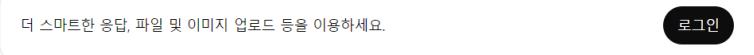
2. 포트 그룹 (Port Groups)

이 설정은 Suricata가 특정 포트를 모니터링할 때 사용할 포트 그룹을 정의합니다.

```
yaml  
port-groups:  
  HTTP_PORTS: "80"  
  SHELLCODE_PORTS: "!80"  
  ORACLE_PORTS: 1521  
  SSH_PORTS: 22  
  DNP3_PORTS: 20000
```



- `HTTP_PORTS`: HTTP 트래픽을 모니터링할 포트를 설정합니다. 기본적으로 80번 포트가 사용됩니다.
- `SHELLCODE_PORTS`: 쉘코드 공격을 탐지할 포트를 설정합니다. `!80`은 80번 포트를 제외하고 나머지 포트를 대상으로 합니다.
- `ORACLE_PORTS`: Oracle 데이터베이스의 기본 포트 1521을 설정합니다.
- `SSH_PORTS`: SSH 프로토콜의 기본 포트 22를 설정합니다.
- `DNP3_PORTS`: DNP3 프로토콜에 사용되는 포트 20000을 설정합니다.
- `MODBUS_PORTS`: MODBUS 프로토콜의 기본 포트 502를 설정합니다.
- `FILE_DATA_PORTS`: 파일 전송을 다루는 포트들을 설정합니다. HTTP 포트 80과 추가로 110, 143을 포함합니다.
- `FTP_PORTS`: FTP 프로토콜의 기본 포트 21을 설정합니다.
- `GENEVE_PORTS`: Geneve 프로토콜의 포트 6081을 설정합니다.
- `VXLAN_PORTS`: VXLAN 프로토콜의 포트 39를 설정합니다.



고를 생성하며, 다양한 네트워크 보안 이벤트를 감지할 수 있습니다. `suricata.yaml` 파일에서 이와 같은 설정을 적절히 조정하여 원하는 네트워크 환경에 맞는 보안 정책을 적용할 수 있습니다.

```
## 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
```



```
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'
```



```
- eve-log:
  enabled: yes
  filetype: regular #regular|syslog|unix_dgram|unix_stream|redis
  filename: eve.json
  # Enable for multi-threaded evejson 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,
```

더 스마트한 응답, 파일 및 이미지 업로드 등을 이용하세요.

로그인

```
#           ## 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
```

이 부분은 Suricata의 로깅 및 출력 설정에 관한 내용입니다. 이 설정을 통해 Suricata가 로그를 어떻게 기록하고 출력할지, 어떤 추가 기능을 사용할지 결정할 수 있습니다. 주요 항목을 하나씩 살펴보겠습니다.

1. 기본 로그 디렉토리 설정

yaml

코드 복사

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

더 스마트한 응답, 파일 및 이미지 업로드 등을 이용하세요.

로그인

다. 이 경로는 Suricata의 로그와 출력 파일이 저장될 기본 위치를 정의합니다. 명령줄에서 `-l` 옵션을 사용하여 다른 경로로 변경할 수도 있습니다.

2. 글로벌 통계 설정

yaml

코드 복사

```
stats:
  enabled: yes
  interval: 8
#decoder-events: true
#decoder-events-prefix: "decoder.event"
#stream-events: false
```

- `stats` 섹션은 Suricata의 통계 기록을 설정합니다.
 - `enabled: yes`: 통계를 기록하도록 설정합니다.
 - `interval: 8`: 통계 업데이트 주기를 초 단위로 설정합니다. 여기서는 8초마다 업데이트 됩니다.

더 스마트한 응답, 파일 및 이미지 업로드 등을 이용하세요.

로그인

- `decoder-events-prefix`: 디코더 이벤트의 접두사를 설정합니다. 주석 처리되어 있습니다.
- `stream-events`: 스트림 이벤트를 통계에 포함할지 여부를 설정합니다. 현재 주석 처리되어 있습니다.

3. 플러그인 설정

yaml

코드 복사

```
plugins:
#   - /path/to/plugin.so
```

- `plugins` 섹션은 Suricata에서 사용할 플러그인 파일의 경로를 설정합니다. 주석 처리되어 있으며, 플러그인을 사용하려면 해당 파일 경로를 지정해야 합니다. 이 기능은 실험적이며 추가적인 기능을 제공할 수 있습니다.

4. 출력 설정

Suricata는 다양한 형식으로 로그를 기록합니다. 다음은 주요 출력 설정 항목입니다.

더 스마트한 응답, 파일 및 이미지 업로드 등을 이용하세요.

로그인

```
- fast:
  enabled: yes
```

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

- `fast`는 Suricata의 경고를 라인 기반 로그로 기록합니다. Snort의 `fast.log`와 유사한 형식입니다.
 - `enabled: yes`: Fast 로그를 활성화합니다.
 - `filename`: 로그 파일의 이름을 설정합니다 (`fast.log`).
 - `append: yes`: 기존 파일에 로그를 추가합니다.
 - `filetype`: 파일 형식을 설정할 수 있습니다. 주석 처리된 부분에서는 `regular`, `unix_stream`, `unix_dgram` 중 하나를 선택할 수 있습니다.

EVE 로그

```
더 스마트한 응답, 파일 및 이미지 업로드 등을 이용하세요. 로그인

enabled: yes
filetype: regular #regular|syslog|unix_dgram|unix_stream|redis
filename: eve.json
#threaded: false
#prefix: "@cee: " # prefix to prepend to each log entry
#identity: "suricata"
#facility: local5
#level: Info
#ethernet: no
#redis:
#   server: 127.0.0.1
#   port: 6379
#   async: true
#   mode: list
#   key: suricata
```

- `eve-log`는 JSON 형식의 이벤트 로그를 기록합니다. Extensible Event Format (EVE)이라고도 불립니다.

• `enabled: yes` EVE 로그를 활성화합니다

더 스마트한 응답, 파일 및 이미지 업로드 등을 이용하세요. 로그인

- `filename`: 로그 파일의 이름을 설정합니다 (`eve.json`).
- `threaded`: 다중 스레드 모드에서 EVE 로그를 출력할지 여부를 설정합니다.
- `prefix`: 각 로그 항목에 추가할 접두사를 설정합니다.
- `identity`, `facility`, `level`: syslog 관련 설정입니다. syslog를 사용할 경우 유용합니다.
- `ethernet`: 이 옵션을 `yes`로 설정하면 이더넷 헤더를 로그에 포함합니다.
- `redis`: Redis를 사용하는 설정입니다. Redis 서버의 주소와 포트, 비동기 처리 여부, 데이터 전송 모드 등을 설정할 수 있습니다.

이와 같은 설정을 통해 Suricata의 로그 및 출력을 세밀하게 조정할 수 있으며, 네트워크 트래픽과 보안 이벤트를 효과적으로 모니터링하고 기록할 수 있습니다. 각 옵션을 자신의 환경에 맞게 조정하여 필요한 정보를 효율적으로 수집할 수 있습니다.

Step 2: Select controls to enable

더 스마트한 응답, 파일 및 이미지 업로드 등을 이용하세요. 로그인

```
# 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/
```



이 부분은 Suricata의 로그 및 출력 설정에서 기본 로그 디렉토리를 정의하는 설정입니다. 이 설정은 Suricata가 생성하는 모든 로그 파일과 출력 파일의 기본 저장 위치를 결정합니다. 자세한 내용은 다음과 같습니다:

1. 기본 로그 디렉토리 설정

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

코드 복사

- `default-log-dir`: Suricata가 로그 파일을 저장할 기본 디렉토리입니다. 여기서는 `/var/log/suricata/`로 설정되어 있습니다.

이 디렉토리는 Suricata가 로그를 기록할 본 위치를 정의합니다. 파일 경로를 명시하지 않고

더 스마트한 응답, 파일 및 이미지 업로드 등을 이용하세요.

로그인

추가 설명

- 디렉토리 변경:** 기본적으로 설정된 경로를 변경하려면 이 값을 수정하면 됩니다. 예를 들어, 로그를 `/home/user/suricata_logs/`에 저장하고 싶다면 `default-log-dir`을 `/home/user/suricata_logs/`로 설정하면 됩니다.
- 명령줄 파라미터:** `default-log-dir`으로 설정된 경로는 Suricata의 실행 시 명령줄에서 `-l` 파라미터를 사용하여 다른 경로로 오버라이드할 수 있습니다. 예를 들어, Suricata를 실행할 때 `-l /path/to/other/log/dir`를 추가하면, 지정된 다른 디렉토리에 로그를 저장합니다.

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

```
# 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 -l command line parameter.
default-log-dir: /var/log/suricata/
```

더 스마트한 응답, 파일 및 이미지 업로드 등을 이용하세요.

로그인

```
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 ↓ of alert (and other) logging you would like.
outputs:
```

더 스마트한 응답, 파일 및 이미지 업로드 등을 이용하세요.

로그인

```
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,
```

더 스마트한 응답, 파일 및 이미지 업로드 등을 이용하세요.

로그인

```
# 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
```

더 스마트한 응답, 파일 및 이미지 업로드 등을 이용하세요.

로그인

```
# 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
```

더 스마트한 응답, 파일 및 이미지 업로드 등을 이용하세요.

로그인

```
# 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
```

더 스마트한 응답, 파일 및 이미지 업로드 등을 이용하세요.

로그인

```
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-printable: yes # Requires metadata; enable
dumping of HTTP body in printable format
```

더 스마트한 응답, 파일 및 이미지 업로드 등을 이용하세요.

로그인

```
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
```

더 스마트한 응답, 파일 및 이미지 업로드 등을 이용하세요.

로그인

```
# 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
```

더 스마트한 응답, 파일 및 이미지 업로드 등을 이용하세요.

로그인

```
# 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,
```

더 스마트한 응답, 파일 및 이미지 업로드 등을 이용하세요.

로그인

```
# 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
#format: detailed
```

더 스마트한 응답, 파일 및 이미지 업로드 등을 이용하세요.

로그인

```

# Default.ctl.
#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

```

더 스마트한 응답, 파일 및 이미지 업로드 등을 이용하세요.

[로그인](#)

```

#- 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,

```

더 스마트한 응답, 파일 및 이미지 업로드 등을 이용하세요.

[로그인](#)

```

# 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]
설명부탁해

```

이 부분은 Suricata의 로그 및 출력 설정에 관한 것으로, 로그의 위치와 다양한 로그 형식 및 세부 정보를 정의합니다. 각 설정 항목을 자세히 설명해드릴게요.

1. 기본 로그 디렉토리 설정

yaml

[코드 복사](#)

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

- `**default-log-dir**`: Suricata가 로그 파일로 저장할 기본 디렉토리입니다. 이 경로에 모든 로그 파일이 저장되며, 명령줄에서 `-l` 옵션으로 다른 경로를 지정할 수 있습니다.

더 스마트한 응답, 파일 및 이미지 업로드 등을 이용하세요.

[로그인](#)

yaml

[코드 복사](#)

```
stats:
  enabled: yes
  interval: 8
  #decoder-events: true
  #decoder-events-prefix: "decoder.event"
  #stream-events: false
```

- `**enabled: yes**`: 통계 수집을 활성화합니다.
- `**interval: 8**`: 통계 업데이트 주기를 8초로 설정합니다.
- `**#decoder-events**`: 패킷 디코더 이벤트를 통계에 포함할지 여부를 설정합니다. 주석 처리되어 기본값이 사용됩니다.
- `**#decoder-events-prefix**`: 디코더 이벤트의 접두사를 설정합니다. 주석 처리되어 기본값이 사용됩니다.

됩니다.

- `stream-events`: 스트림 이벤트를 통계로 포함할지 여부를 설정합니다. 주석 처리되어 기본값이 사용됩니다.

더 스마트한 응답, 파일 및 이미지 업로드 등을 이용하세요.

로그인

yaml

코드 복사

```
plugins:  
# - /path/to/plugin.so
```

- `plugins`: Suricata에서 사용할 플러그인을 설정합니다. 주석 처리된 상태에서 플러그인을 사용하려면 경로를 지정해야 합니다.

4. 출력 설정

Suricata는 다양한 형식으로 로그를 기록할 수 있습니다. 주요 출력 설정 항목은 다음과 같습니다.

Fast 로그

yaml

코드 복사

```
- fast:  
  enabled: yes  
  filename: fast.log  
  append: yes
```

더 스마트한 응답, 파일 및 이미지 업로드 등을 이용하세요.

로그인

fast.log 파일은 서버에 대한 모든 정보를 기록하는 기본적인 형식입니다.

- `enabled: yes`: Fast 로그를 활성화합니다.
- `filename: fast.log`: 로그 파일의 이름을 설정합니다.
- `append: yes`: 기존 파일에 로그를 추가합니다.
- `filetype`: 파일 형식을 설정할 수 있습니다. 주석 처리된 부분에서는 `regular`, `unix_stream`, `unix_dgram` 중 하나를 선택할 수 있습니다.

EVE 로그

yaml

코드 복사

```
- eve-log:  
  enabled: yes  
  filetype: regular #regular|syslog|unix_dgram|unix_stream|redis  
  filename: eve.json  
  #threaded: false  
  #prefix: "@cee: "  
  #identity: "suricata"  
  #serializers: json
```

더 스마트한 응답, 파일 및 이미지 업로드 등을 이용하세요.

로그인

```
#redis:  
#  server: 127.0.0.1  
#  port: 6379  
#  async: true  
#  mode: list  
#  key: suricata  
#pipelining:  
#  enabled: yes  
#  batch-size: 10  
#metadata: no  
#pcap-file: false  
#community-id: false  
#community-id-seed: 0  
#xff:  
#  enabled: no  
#  mode: extra-data  
#  deployment: reverse  
#  header: X-Forwarded-For  
types:  
- alert:
```

더 스마트한 응답, 파일 및 이미지 업로드 등을 이용하세요.

로그인

```
- anomaly:  
  enabled: yes  
  #types:  
  #packethdr: no  
- http:  
  extended: yes  
  #custom: [Accept-Encoding, Accept-Language, Authorization]  
  #dump-all-headers: none  
  - dns:
```

```
#version: 2
#enabled: yes
#requests: no
#responses: no
#formats: [detailed, grouped]
#types: [a, aaaa, cname, mx, ns, ptr, txt]
- tls:
  extended: yes
  #session-resumption: no
  #ja4: off
  #custom: [subject, issuer, session_resumed, serial, fingerprint, sni, version,
```

더 스마트한 응답, 파일 및 이미지 업로드 등을 이용하세요.

로그인

```
#- drop:
#  alerts: yes
#  flows: all
#  verdict: yes
- smtp:
  #extended: yes
  #custom: [received, x-mailer, x-originating-ip, relays, reply-to, bcc]
  #md5: [body, subject]
```

- `eve-log`: JSON 형식의 EVE 로그를 설정합니다.
- `enabled: yes`: EVE 로그를 활성화합니다.
- `filetype`: 로그를 기록할 형식을 설정합니다 (`regular`, `syslog`, `unix_dgram`, `unix_stream`, `redis`).
- `filename: eve.json`: 로그 파일의 이름을 설정합니다.
- `threaded`: 다중 스레드 모드에서 ↓ 로그를 출력할지 여부를 설정합니다.

더 스마트한 응답, 파일 및 이미지 업로드 등을 이용하세요.

로그인

- `ethernet`: 이더넷 헤더를 로그에 포함할지 여부를 설정합니다.
- `redis`: Redis를 사용하는 설정입니다. 서버의 주소와 포트, 비동기 처리 여부, 데이터 전송 모드 등을 설정할 수 있습니다.
- `pipelining`: Redis에 대한 쿼리 파이프라인 설정입니다.
- `metadata`: EVE 로그에 메타데이터를 포함할지 여부를 설정합니다.
- `pcap-file`: pcap 파일을 처리할 때 파일 이름을 포함할지 여부를 설정합니다.
- `community-id`: Community ID를 사용하여 로그 기록을 일관되게 할지 여부를 설정합니다.
- `community-id-seed`: Community ID의 시드를 설정합니다.
- `xff`: X-Forwarded-For HTTP 헤더를 지원하여 IP 주소를 기록하는 방식입니다.
- `types`: 다양한 로그 유형을 설정합니다:

- `alert`: 경고 로그를 설정합니다. 태그가 붙은 패킷을 기록하며, 패킷의 최종 행동을 로그할 수도 있습니다.

- `frame`: 애플리케이션 레이어 브레임을 기록합니다. 기본적으로 비활성화되어 있습니다.

더 스마트한 응답, 파일 및 이미지 업로드 등을 이용하세요.

로그인

- `http`: HTTP 요청 및 응답에 대한 확장 정보를 기록합니다.
- `dns`: DNS 쿼리 및 응답을 기록합니다.
- `tls`: TLS 트랜잭션 정보를 기록합니다.
- `files`: 파일 관련 로그를 기록합니다.
- `drop`: 드롭된 패킷과 관련된 경고를 기록합니다.
- `smtp`: SMTP 메일 관련 로그를 기록합니다.

이 설정을 통해 Suricata의 로그를 어떻게 기록하고 출력할지, 어떤 추가 정보와 로그 형식을 사용할지 설정할 수 있습니다. 로그 설정을 적절하게 조정하여 네트워크 보안 모니터링 및 분석을 효과적으로 수행할 수 있습니다.

```
#- dnp3
- ftp
- rdp
- nfs
```

더 스마트한 응답, 파일 및 이미지 업로드 등을 이용하세요.

로그인

```
- 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
--
```

더 스마트한 응답, 파일 및 이미지 업로드 등을 이용하세요.

로그인

```
- https
- 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
```

더 스마트한 응답, 파일 및 이미지 업로드 등을 이용하세요.

로그인

```
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)
```

더 스마트한 응답, 파일 및 이미지 업로드 등을 이용하세요.

로그인

```
# 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: "%{ID-%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
```



이 설정은 Suricata의 다양한 로그 출력을 구성하는 항목들입니다. 각 항목의 역할과 설정 옵션을 살펴보세요. ↓

더 스마트한 응답, 파일 및 이미지 업로드 등을 이용하세요.

로그인

```

yaml
#- 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:

```

더 스마트한 응답, 파일 및 이미지 업로드 등을 이용하세요.

로그인

```

- mqtt:
    # passwords: yes          # enable output of passwords
- http2
- postgresql:
    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
- flow
#- netflow
#- metadata
#- 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

```

더 스마트한 응답, 파일 및 이미지 업로드 등을 이용하세요.

로그인

- 로토콜과 서비스의 로그 수집 여부와 상세 설정을 조절할 수 있습니다.
- `quic`: QUIC 프로토콜의 로그 설정을 정의합니다. JA4 해시를 로그에 포함할지 여부를 설정할 수 있습니다.
 - `dhcp`: DHCP 로그를 설정합니다. `extended` 모드가 켜지면 DHCP 메시지를 자세히 기록합니다.
 - `stats`: 통계 로그를 설정합니다. 모든 스레드의 통계, 스레드별 통계, 델타 값을 포함할지 설정할 수 있습니다.
 - `flow`: 양방향 흐름 로그를 설정합니다.
 - `metadata`: 패킷 번수와 흐름 번수를 포함한 메타데이터 로그를 설정합니다.
 - `stream`: TCP 상태 추적 세부정보를 기록하는 실험적인 로그입니다. 기본적으로 비활성화되어 있으며, TCP 패킷을 기록하지, 상태 업데이트를 기록하지 등을 설정할 수 있습니다.

2. HTTP 요청 로그

더 스마트한 응답, 파일 및 이미지 업로드 등을 이용하세요.

로그인

```

yaml
append: yes
#extended: yes
#custom: yes
#customformat: "%{D-%H:%M:%S}t.%z %{X-Forwarded-For}i %H %m %h %u %s %B %a:%p -> %A
#filetype: regular

```

- `http-log`: HTTP 요청을 기록하는 로그를 설정합니다.
- `enabled: no`: HTTP 로그 기능을 비활성화합니다.
- `filename: http.log`: HTTP 로그 파일의 이름을 설정합니다.

- `append: yes`: 기존 파일에 로그를 추가합니다.
- `extended`: HTTP 요청의 확장 정보를 기록할지 여부를 설정합니다.
- `custom`: 사용자 정의 형식으로 로그를 기록할지 여부를 설정합니다.
- `customformat`: 사용자 정의 로그 형식을 지정합니다.
- `filetype`: 파일 형식을 설정합니다. (regular, unix_stream, unix_dgram).

더 스마트한 응답, 파일 및 이미지 업로드 등을 이용하세요.

로그인

```
yaml
- tls-log:
  enabled: no
  filename: tls.log
  append: yes
  #extended: yes
  #custom: yes
  #customformat: "%{D-%H:%M:%S}t.%z %a:%p -> %A:%P %v %n %d %D"
  #filetype: regular
  #session-resumption: no
```

- `tls-log`: TLS 핸드쉐이크 정보를 기록하는 로그를 설정합니다.
- `enabled: no`: TLS 로그 기능을 비활성화합니다.
- `filename: tls.log`: TLS 로그 파일의 이름을 설정합니다.
- `append: yes`: 기존 파일에 로그를 추가합니다.
- `extended`: TLS 핸드쉐이크의 확장 정보를 기록할지 여부를 설정합니다.

더 스마트한 응답, 파일 및 이미지 업로드 등을 이용하세요.

로그인

- `filetype`: 파일 형식을 설정합니다 (regular, unix_stream, unix_dgram).
- `session-resumption`: 세션 재개 정보를 기록할지 여부를 설정합니다.

이 설정을 통해 Suricata는 다양한 네트워크 프로토콜 및 서비스에 대한 로그를 관리하고, HTTP 및 TLS 핸드쉐이크에 대한 추가 정보를 기록할 수 있습니다. 필요에 따라 로그 유형과 설정을 조정하여 원하는 정보를 기록하고 분석할 수 있습니다.

```
# 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-
```

더 스마트한 응답, 파일 및 이미지 업로드 등을 이용하세요.

로그인

```
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.
```

더 스마트한 응답, 파일 및 이미지 업로드 등을 이용하세요.

로그인

```
# 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 ↓
```

더 스마트한 응답, 파일 및 이미지 업로드 등을 이용하세요.

로그인

```
# 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 ↓ place pcap files. If not provided the default log
# directory ... be used. Required for "sguil" mode.
```

더 스마트한 응답, 파일 및 이미지 업로드 등을 이용하세요.

로그인

```
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.loa contains data from various counters of the Suricata
```

더 스마트한 응답, 파일 및 이미지 업로드 등을 이용하세요.

로그인

```
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
```

더 스마트한 응답, 파일 및 이미지 업로드 등을 이용하세요.

로그인

1. TLS 인증서 저장 설정

```
yaml
- tls-store:
  enabled: no
  #certs-log-dir: certs # directory to store the certificates files
```

- `tls-store`: TLS 인증서를 디스크에 저장하는 기능을 설정합니다.
 - `enabled: no`: TLS 인증서 저장 기능을 비활성화합니다.
 - `certs-log-dir`: 인증서 파일을 저장할 디렉토리를 지정합니다.

2. 패킷 로그

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

더 스마트한 응답, 파일 및 이미지 업로드 등을 이용하세요.

코드 복사

로그인

```
mode: normal
#dir: /nsm_data/
#ts-format: usec
use-stream-depth: no
honor-pass-rules: no
#conditional: all
```

- `pcap-log`: 패킷을 pcap 포맷으로 로그하는 기능을 설정합니다.
 - `enabled: no`: 패킷 로그 기능을 비활성화합니다.
 - `filename: log.pcap`: pcap 파일의 기본 이름을 설정합니다.
 - `limit: 1000mb`: 파일 크기 제한을 설정합니다. 단위는 kb, mb, gb로 지정할 수 있습니다.
 - `max-files: 2000`: 최대 파일 수를 설정합니다. 이 수를 초과하면 오래된 파일이 삭제됩니다.
 - `compression: none`: pcap 파일의 압축 방식을 설정합니다. 가능한 값은 `none` (압축 없음), `lz4` (LZ4 압축)입니다.

더 스마트한 응답, 파일 및 이미지 업로드 등을 이용하세요.

로그인

- `normal`: 난일 pcap 파일에 보는 패킷을 기록합니다.
- `multi`: 스레드별로 pcap 파일을 생성하여 성능을 개선합니다.
- `sguil`: Sguil 시스템에 맞는 디렉토리 구조를 사용하여 pcap 파일을 기록합니다.
- `dir`: Sguil 모드에서 pcap 파일을 저장할 기본 디렉토리를 설정합니다.
- `ts-format: usec`: 타임스탬프 형식을 설정합니다. `sec` (초 단위) 또는 `usec` (マイ크로초 단위) 중 선택할 수 있습니다.
- `use-stream-depth: no`: 스트림 검사 깊이를 초과한 패킷을 무시하지 여부를 설정합니다.
- `honor-pass-rules: no`: `pass` 규칙이 일치하는 흐름을 로그에서 제외하지 여부를 설정합니다.
- `conditional`: 조건에 따라 패킷 로그를 기록합니다. 가능한 값은 `all` (모든 패킷), `alerts` (알림이 발생한 패킷 및 흐름), `tag` (태그가 붙은 흐름)입니다.

3. 디버그 경고 로그

```
yaml
append: yes
#filetype: regular
```

더 스마트한 응답, 파일 및 이미지 업로드 등을 이용하세요.

코드 복사

로그인

- `alert-debug`: 디버깅용 경고 로그를 설정합니다.
 - `enabled: no`: 경고 디버깅 로그 기능을 비활성화합니다.
 - `filename: alert-debug.log`: 디버깅 경고 로그 파일의 이름을 설정합니다.
 - `append: yes`: 기존 파일에 로그를 추가합니다.
 - `filetype`: 파일 형식을 설정합니다 (`regular`, `unix_stream`, `unix_dgram`).

4. 통계 로그

```
yaml
#filetype: regular
```

코드 복사

```

- stats:
  enabled: yes
  filename: stats.log
  append: yes
  totals: yes
  threads: no
  null-values: yes

```

더 스마트한 응답, 파일 및 이미지 업로드 등을 이용하세요.

[로그인](#)

- `stats`: Suricata 엔진의 통계 데이터를 기록하는 로그를 설정합니다.
 - `enabled: yes`: 통계 로그 기능을 활성화합니다.
 - `filename: stats.log`: 통계 로그 파일의 이름을 설정합니다.
 - `append: yes`: 기존 파일에 로그를 추가합니다.
 - `totals: yes`: 모든 스레드의 통계를 병합하여 기록합니다.
 - `threads: no`: 스레드별 통계를 기록할지 여부를 설정합니다.
 - `null-values: yes`: 값이 0인 카운터도 출력할지 여부를 설정합니다.

5. Syslog 출력

```

yaml
- syslog:
  enabled: no
  #identity: "suricata"
  facility: local5

```

더 스마트한 응답, 파일 및 이미지 업로드 등을 이용하세요.

[로그인](#)

- `syslog`: syslog에 로그를 전송하는 기능을 설정합니다.
 - `enabled: no`: syslog 기능을 비활성화합니다.
 - `identity`: syslog에 보고할 프로그램 이름을 설정합니다. 기본값은 Suricata입니다.
 - `facility: local5`: syslog의 facility를 설정합니다. 로그의 분류를 지정합니다.
 - `level: Info`: 로그 수준을 설정합니다. 가능한 값은 `Emergency`, `Alert`, `Critical`, `Error`, `Warning`, `Notice`, `Info`, `Debug`입니다.

이 설정들을 통해 Suricata의 다양한 로그 출력을 조정하고, 필요에 맞는 로그 파일과 형식을 설정할 수 있습니다.

```

# 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 dup. file is found, the timestamps on the existing
# file will be updated to match the new file's timestamp.
# 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

```

더 스마트한 응답, 파일 및 이미지 업로드 등을 이용하세요.

[로그인](#)

 The `file-store` module configuration for Suricata deals with the storage of files on disk, particularly for file extraction and metadata management. Here's a breakdown of the configuration options provided:

```

- file-store:
  version: 2
  enabled: no

```

더 스마트한 응답, 파일 및 이미지 업로드 등을 이용하세요.

[로그인](#)

- `file-store`: This module is responsible for storing files that Suricata has identified and extracted during inspection. It manages file storage and optional metadata logging.

Configuration Details:

- `version: 2`: This specifies the version of the file-store module. Version 2 might have additional features or improvements over the previous version.
- `enabled: no`: This indicates that the file-store module is currently disabled. Setting this to `yes` would activate the file-store module, allowing Suricata to store extracted files on disk.

How It Works:

1. **File Storage:** Files are stored in a directory structure based on the SHA256 hash of the file. Specifically:

더 스마트한 응답, 파일 및 이미지 업로드 등을 이용하세요.

로그인

2. **Duplicate Handling:** If a file with the same SHA256 hash is detected again, Suricata updates the timestamps on the existing file rather than creating a new file.
3. **Metadata:** By default, the file-store does not write additional metadata. Metadata about the files is usually included in the `eve-log` under the "fileinfo" record. If you enable `write-fileinfo`, an additional `.json` file containing file metadata will be created alongside each stored file. This ensures that even files with the same hash have unique metadata files.
4. **File Pruning:** The `suricatactl filestore prune` command helps manage disk usage by deleting old files from the file-store directory based on age.

Example Configuration for Enabling File-Store:

To activate and customize the file-store module, you might adjust the configuration as follows:

```
yaml
- file-store:
    version: 2
    enabled: yes
```

더 스마트한 응답, 파일 및 이미지 업로드 등을 이용하세요.

로그인

In this example:

- `enabled: yes`: Activates the file-store module.
- `dir`: (Optional) Specifies the directory where files will be stored. If not specified, the default log directory will be used.
- `write-fileinfo: yes`: (Optional) Enables the creation of metadata files in addition to the stored files.

By configuring the file-store module, you can efficiently manage file storage, handle duplicates, and track file metadata, which can be particularly useful for in-depth forensic analysis or security investigations.

한국말로 알려줘야지

더 스마트한 응답, 파일 및 이미지 업로드 등을 이용하세요.

로그인

`file-store` 모듈 설정

```
yaml
- file-store:
    version: 2
    enabled: no
```

- `file-store`: 이 모듈은 Suricata가 검사 중에 식별하고 추출한 파일을 디스크에 저장하는 기능을 담당합니다. 파일 저장 및 메타데이터 관리가 포함됩니다.

설정 세부 사항:

- `version: 2`: 파일 저장 모듈의 버전을 지정합니다. 버전 2는 이전 버전보다 기능이 추가되었거나 개선된 사항이 있을 수 있습니다.
- `enabled: no`: 이 설정은 파일 저장 모듈이 현재 비활성화되어 있음을 나타냅니다. `yes`로 설정하면 파일 저장 모듈이 활성화되어 Suricata가 파일을 디스크에 저장할 수 있게 됩니다.

더 스마트한 응답, 파일 및 이미지 업로드 등을 이용하세요.

로그인

- **디렉토리 이름:** SHA256 해시의 첫 두 분자로 만들어집니다.
- **파일 이름:** 전체 SHA256 해시를 파일 이름으로 사용합니다.

2. **중복 처리:** 동일한 SHA256 해시를 가진 파일이 다시 감지되면, Suricata는 기존 파일의 타임스탬프를 업데이트하고 새 파일을 생성하지 않습니다.
3. **메타데이터:** 기본적으로 `file-store`는 추가 메타데이터를 기록하지 않습니다. 파일에 대한 메타데이터는 보통 `eve-log`의 "fileinfo" 레코드에 포함됩니다. `write-filinfo`를 활성화하면 각 저장된 파일에 추가 `.json` 메타데이터 파일이 생성됩니다. 이를 통해 동일한 해시를 가진 파일이라도 고유한 메타데이터 파일이 생성됩니다.
4. **파일 정리:** `suricatactl filestore prune` 명령을 사용하여 파일 저장 디렉토리의 오래된 파일을 삭제하여 디스크 공간을 관리할 수 있습니다.

파일 저장 모듈 활성화 예시 설정:

파일 저장 모듈을 활성화하고 사용자 정의 설정을 추가하려면 다음과 같이 설정을 조정할 수 있습니다:

```
yaml
# 선택적으로 파일을 저장할 디렉토리를 지정합니다.
#dir: /path/to/filestore/
# 선택적으로 메타데이터 파일 생성을 활성화합니다.
#write-filinfo: yes
```

이 예시에서는:

- `enabled: yes`: 파일 저장 모듈을 활성화합니다.
- `dir`: (선택 사항) 파일이 저장될 디렉토리를 지정합니다. 지정하지 않으면 기본 로그 디렉토리에 저장됩니다.
- `write-filinfo: yes`: (선택 사항) 메타데이터 파일 생성을 활성화합니다.

이 설정을 통해 파일 저장을 효율적으로 관리하고, 중복된 파일을 처리하며, 파일 메타데이터를 추적할 수 있습니다. 이는 깊이 있는 포렌식 분석이나 보안 조사를 수행하는 데 유용할 수 있습니다.



더 스마트한 응답, 파일 및 이미지 업로드 등을 이용하세요.

로그인

메시지 ChatGPT

ChatGPT는 실수를 할 수 있습니다. 중요한 정보를 확인하세요.