Suricata 사용

메뉴얼

작성자 오석빈

목차

Suricata 사용 매뉴얼

1. 개요

IDS인 Suricata의 기능과 특징에 대해 설명을 한다. 그리고 설치와 사용법을 숙지할 수 있고, Suricata의 설정파일을 살펴봄으로써, 자신에게 필요한 항목을 설정할 수 있다.

2. Suricata 기능 및 특징

오픈 소스 기반 네트워크 IDS(Intrusion Detection Service), IPS(Intrusion-Prevention System) 모니터링 엔진으로써 기존에 사용되었던 Snort 기능 지원 할 뿐만 아니라 snort의 룰을 완벽하게 호환해준다. 실시간으로 패킷 분석 및 packet logging을 수행할 수 있으며, 직접 자신이 룰을 추가하여 원하는 패킷을 필터링 할 수 있다.

3. 설치 방법

소스 컴파일 방법(권장)

https://suricata-ids.org/download/ 에서 suricata.tar 다운 후 압축 해제

```
tar xzvf suricata-4.0.0.tar.gz
cd suricata-4.0.0
```

패키지 설치

```
yum -y install gcc libpcap-devel pcre-devel libyaml-devel file-devel zlib-devel jansson-devel
nss-devel libcap-ncg-devel libnet-devel tar make libnetfilter_queue-devel lua-devel
```

configure 설정

```
./configure --prefix=/usr --sysconfdir=/etc --localstatedir=/var --enable-nfqueue --enable-lua
```

make, make install 실행

```
make
make install
```

YUM 설치 방법

```
yum install epel-release
yum install suricata
```

4. Suricata 사용방법

명령 옵션.

옵션	설명
-h	help와 같은 명령어
-V	Suricata의 버전을 보여준다
-c <path>	설정파일의 경로를 지정한다
-T	테스트 모드로 suricata실행
-v	Suricata의 결과를 더 자세하게 보여준다
-i <interface>	패킷이 들어올 인터페이스를 지정
-s <filename.rules>	yaml에 지정되어있지 않은 rule파일을 로드
-S <filename.rules>	yaml에 지정되어있는 rule파일과는 상관없이 파일을 로드
-l	log dir을 설정할 수 있다. yaml에 default로 지정되어있는 경로가 있다면 무시하고 옵션에 적은 경로를 log dir로 인식
-D	suricata를 데몬으로 실행시켜 백그라운드에서 실행
--user <user이름>	user이름에 써져있는 사용자로 suricata를 실행
--group <group이름>	group이름에 써져있는 그룹으로 suricata 실행
--disable-detection	탐지모드 즉 ids 모드 비화성화
--init-error-fata	파일을 로드하는중 오류가 발생하면 실패로 종료

보편적으로 실행 시 다음과 같이 입력함

suricata –c /etc/suricata/suricata.yaml –i enp0s3

```
[root@localhost rules]# suricata -c /etc/suricata/suricata.yaml -i enp0s3
28/6/2018 -- 11:04:52 - <Notice> - This is Suricata version 4.0.4 RELEASE
28/6/2018 -- 11:04:55 - <Notice> - all 1 packet processing threads, 4 management
 threads initialized, engine started.
```

인터페이스 확인은 ifconfig를 사용하면 알 수 있다.

```
[root@localhost ~]# ifconfig
enp0s3: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
        inet 10.0.2.15  netmask 255.255.255.0  broadcast 10.0.2.255
        inet6 fe80::7e1d:519d:da44:bb1c  prefixlen 64  scopeid 0x20<link>
        ether 08:00:27:8c:83:6c  txqueuelen 1000  (Ethernet)
        RX packets 3063  bytes 299665 (292.6 KiB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 2206  bytes 705581 (689.0 KiB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0
```

5. Suricata Rule 설정

Rule은 말 그대로 규칙으로써, 미리 지정되어있는 룰에 부합되는 패턴이 들어오면 미리 지정해둔 행동을 할 수 있게 지정해놓은 파일들이다. Suricata가 설치되면서 각 프로토콜과 관련된 룰 파일들이 설치되며, 사용자가 직접 룰 파일을 만들어서 특정한 패턴에 맞는 룰을 설정 할 수도 있다.

Rule Format

Rule은 3가지 요소로 나누어져 있음

```
drop tcp $HOME_NET any -> $EXTERNAL_NET any (msg:"ET TROJAN Likely Bot Nick in IRC (USA +..)";
flow:established,to_server; flowbits:isset,is_proto_irc; content:"NICK "; pcre:"/NICK .*USA.*[0-
9]{3,}/i"; reference:url,doc.emergingthreats.net/2008124; classtype:trojan-activity;
sid:2008124; rev:2;)
```

- **Action(액션)** 어떤 일을 수행할지 결정한다. 4가지의 액션으로 구분된다. (pass, drop, reject, alert)
- **Header(헤더)** 프로토콜, IP주소, 포트 그리고 Rule에 대한 방향을 결정한다.
- **Rule options(규칙 옵션)** Rule에 대한 세부 설정을 기술한다.

Rule 의 구성요소

- **액션**

```
drop tcp $HOME_NET any -> $EXTERNAL_NET any (msg:"ET TROJAN Likely Bot Nick in IRC (USA +..)";
flow:established,to_server; flowbits:isset,is_proto_irc; content:"NICK ";
pcre:"/NICK .*USA.*[0-9]{3,}/i"; reference:url,doc.emergingthreats.net/2008124;
classtype:trojan-activity; sid:2008124; rev:2;)
```

Pass: 패킷이 Rule에 부합 시, 패킷을 통과시킨다.
Drop: 패킷이 Rule에 부합 시, 패킷을 버린다. (단 IPS/inlime 모드에서만 작동)
Reject: 패킷이 Rule에 부합 시, 송/수신자 패킷을 거절한다.
Alert: 패킷이 Rule에 부합 시, 사용자에게 경고 메시지를 보낸다.

- **프로토콜**

```
drop tcp $HOME_NET any -> $EXTERNAL_NET any (msg:"ET TROJAN Likely Bot Nick in IRC (USA +..)";
flow:established,to_server; flowbits:isset,is_proto_irc; content:"NICK ";
pcre:"/NICK .*USA.*[0-9]{3,}/i"; reference:url,doc.emergingthreats.net/2008124;
classtype:trojan-activity; sid:2008124; rev:2;)
```

기본적으로 제공되는 프로토콜은 다음 4개와 같다

```
tcp,udp,icmp,ip(ip경우 all 혹은 any로 작성 가능)
```

그 외에 제공하는 프로토콜의 목록이다.

```
http, ftp, tls (ssl 포함), smb, dns, dcerp, ssh, smtp, imap , msn, modbus, dnp3, enip, nfs,
ikev2, krb5, ntp, dhcp
```

- **시작/도착 주소**

```
drop tcp $HOME_NET any -> $EXTERNAL_NET any (msg:”ET TROJAN Likely Bot Nick in IRC (USA +..)”;
flow:established,to_server; flowbits:isset,is_proto_irc; content:”NICK “;
pcre:”/NICK .*USA.*[0-9]{3,}/i”; reference:url,doc.emergingthreats.net/2008124;
classtype:trojan-activity; sid:2008124; rev:2;)
```

첫 번째 강조된 부분은 시작주소를 나타내며 두 번째 강조된 부분은 도착주소를 나타냄.

- **시작/도착 포트**

```
drop tcp $HOME_NET any -> $EXTERNAL_NET any (msg:”ET TROJAN Likely Bot Nick in IRC (USA
+..)”; flow:established,to_server; flowbits:isset,is_proto_irc; content:”NICK “;
pcre:”/NICK .*USA.*[0-9]{3,}/i”; reference:url,doc.emergingthreats.net/2008124;
classtype:trojan-activity; sid:2008124; rev:2;)
```

첫 번째 강조된 부분은 시작점의 포트번호, 두 번째 강조된 부분은 도작점의 포트번호를 나타낸다.

- **방향**

```
drop tcp $HOME_NET any -> $EXTERNAL_NET any (msg:”ET TROJAN Likely Bot Nick in IRC (USA
+..)”; flow:established,to_server; flowbits:isset,is_proto_irc; content:”NICK “;
pcre:”/NICK .*USA.*[0-9]{3,}/i”; reference:url,doc.emergingthreats.net/2008124;
classtype:trojan-activity; sid:2008124; rev:2;)
```

Rule이 적용되는 방향을 결정지으며 단방향 -> , 양방향 <> 설정할 수 있음. 하지만 역방향(<-)은 존재하지 않는다.

- **Rule Option**

```
drop tcp $HOME_NET any -> $EXTERNAL_NET any (msg:”ET TROJAN Likely Bot Nick in IRC (USA
+..)”; flow:established,to_server; flowbits:isset,is_proto_irc; content:”NICK “;
pcre:”/NICK .*USA.*[0-9]{3,}/i”; reference:url,doc.emergingthreats.net/2008124;
classtype:trojan-activity; sid:2008124; rev:2;)
```

위를 제외한 나머지들은 rule 옵션. 괄호 안에 작성하며 쉼표(,)로 구분 된다. keyword와 setting으로 이루어져있지만 keyword만 있는 옵션도 있다.

Keyword 의 종류

A. Meta Keyword

msg(message)
alert 액션 발동 시 미리 설정한 문자열을 출력한다.
형식: msg: "some description";
ex) msg:”ATTACK-RESPONSES 403 Forbidden”

B. sid(signature ID)

각 규칙에 sid번호를 지정해준다.
형식: sid: 10000001;

C. rev(revision)

Rev는 버전을 표시한다.sid키워드와 rev는 거의 같이 동반된다.
형식: rev: 123;

D. gid(group ID)

그룹 id번호를 규칙에 지정해준다.
fast.log에서 gid를 확인 할 수 있다.

```
10/15/09-03:30:10.219671 [**] [1:2008124:2] ET TROJAN Likely Bot Nick in IRC (USA +..) [**]
[Classification: A Network Trojan was Detected] [Priority: 3] {TCP} 192.168.1.42:1028 ->
72.184.196.31:6667
```

[1:2008124:2] 에서 1이 gid이며 2008124는 sid, 2는 rev이다.

E. reference

rule에 대한 근거나 출처를 적을 수 있다.
형식: reference: type, reference
ex)reference: url, www.info.com 혹은 reference: cve, CVE-2014-1234

F. priority

우선순위를 책정할 수 있다. 255번까지 가능은 하지만 1~4가 가장 많이 사용된다.
형식: priority:1

G. metadata

데이터의 데이터를 작성하는 메타데이터. 자유롭게 작성할 수 있는 키워드이지만 키 값을 입력하는 방식으로 사용 하는 것이 권장된다.
형식: metadata: key value;

IP Keywords

A. ttl

TTL은 IP의 time-to-live를 나타낸다. ttl이 0이 되면 그 패킷은 소멸된다.
형식: ttl:<number>

B. ipopts

IP에 설정된 옵션을 확인할 수 있다. 설정된 옵션과 맞는 IP가 들어오면 설정된 액션을 하게 된다.
형식: ipopts: <name>

<name>에 들어가는 옵션은 다음과 같다.

IP Option	설명
rr	record Route
eol	End of List
nop	No Op
ts	time Stamp
sec	IP Security
esec	IP Extended Security
lsrr	Loose Source Routing
ssrr	Strict source Routing
satid	Sream Identifier
any	any IP

C. id

id 키워드가 있으면 IP의 특정 ID값에 따라 액션을 실행하게 된다.
형식: id:<number>;

D. fragbits (IP 단편화)

단편화된 패킷의 IP헤더를 확인 후 단편화 옵션에 따라 액션을 실행하게 된다.
형식: fragbits:[*+!]<[MDR]>

M	단편화 되어있음
D	단편화 하지 않음
R	예약된 Bit

E. fragoffset

단편화된 IP를 특정 십진수 값과 비교가 가능. 단편화의 순서를 명확하게 결정하는 키워드
형식: fragoffset:[!|<|>]<number>;
< >는 숫자와 크기 비교, !는 같지 않다.

TCP 키워드

A. seq

특정 TCP의 시퀀스 번호를 확인하기 위해 seq키워드를 사용. seq번호를 통해 데이터 스트림의 위치를 추적하는데 도움이 된다.
형식: seq:0;

B. ack

수신자가 SYNK를 받았을 때 ACK신호를 보내는데, 여기에 수신과 함께 증가하는 ack번호가 있다. 이 번호를 확인해서 특정 TCP의 승인번호를 확인할 수 있다.
형식: ack:1;

기타 키워드

A. content

문자열이 포함된 값을 비교해서 그 값이 있으면 필터링을 한다.
형식: content:"……";
"…"내용에 문자열을 입력해도 되지만 16진수의 ASCII 코드를 넣어도 인식이 된다. 형식은 content:"|hex,hex….|" 식으로 넣으면 된다.
ex) content:"|61 62 63|"; 이는 abc를 의미한다.

B. dsize

패킷의 크기를 설정할 수 있다. <,>와 같이 사용하여 패킷의 크기보다 크거나 작은 경우의 액션을 설정할 수 있다.
형식: dsize:<숫자>;

Custom Rule 만들기

자신이 원하는 Rule을 별도의 파일로 작성해서 Suricata에 적용 할 수 있다. Ping을 받았을 때 기록이 남게 하는 Rule을 추가해보자

1. /etc/suricata/rules 밑에 local.rules이라는 파일을 생성한다.

```
vi /etc/suricata/rules/local.rules
```

2. 다음과 같은 Rule을 작성하고 저장한다. Rule 작성시 위의 형식에 유의 할 것.

```
alert icmp any any -> any any ( msg:"ICMP Ping Test"; sid:1000001;)
```

위의 룰은 ICMP 패킷이 들어올 경우 경고를 해주는 규칙을 예로 만들었다.

3. /etc/suricata/suricata.yaml에 local.rules를 추가해준다.

```
vi /etc/suricata/suricata.yalm
```

4. 다음 항목 밑에 local.rules파일 이름을 작성

```
47
48 ##
49 ## Step 2: select the rules to enable or disable
50 ##
51
52 default-rule-path: /etc/suricata/rules
53 rule-files:
54  - botcc.rules
55  # - botcc.portgrouped.rules
56  - ciarmy.rules
57  - compromised.rules
97  - emerging-worm.rules
98  - tor.rules
99  - local.rules
```

5. suricata를 실행 시 오류 메시지가 없다면 정상적으로 작동.

```
[root@localhost rules]# suricata -c /etc/suricata/suricata.yaml -i enp0s3
28/6/2018 -- 11:04:52 - <Notice> - This is Suricata version 4.0.4 RELEASE
28/6/2018 -- 11:04:55 - <Notice> - all 1 packet processing threads, 4 management
 threads initialized, engine started.
```

6. fast.log를 확인하면 자신이 만든 Rule의 기록을 확인 할 수 있다.

```
06/27/2018-15:53:41.873532  [**] [1:1000001:0] ICMP Ping Test [**] [Classificati
on: (null)] [Priority: 3] {ICMP} 10.0.2.15:8 -> 192.168.56.1:0
```

6. Suricata.yaml 분석

suricata의 설정파일인 suricata.yalm에는 다양한 설정이 존재한다. 기본적으로 IP 설정을 비롯해, 패킷의 크기, 혹은 여러 로그의 출력 설정이 가능하다.

Rule-vars

```
12 vars:
13   # more specifc is better for alert accuracy and performance
14   address-groups:
15     HOME_NET: "[192.168.0.0/16,10.0.0.0/8,172.16.0.0/12]"
16     #HOME_NET: "[192.168.0.0/16]"
17     #HOME_NET: "[10.0.0.0/8]"
18     #HOME_NET: "[172.16.0.0/12]"
19     #HOME_NET: "any"
20
21     EXTERNAL_NET: "!$HOME_NET"
22     #EXTERNAL_NET: "any"
23
24     HTTP_SERVERS: "$HOME_NET"
25     SMTP_SERVERS: "$HOME_NET"
26     SQL_SERVERS: "$HOME_NET"
27     DNS_SERVERS: "$HOME_NET"
28     TELNET_SERVERS: "$HOME_NET"
29     AIM_SERVERS: "$EXTERNAL_NET"
30     DNP3_SERVER: "$HOME_NET"
31     DNP3_CLIENT: "$HOME_NET"
32     MODBUS_CLIENT: "$HOME_NET"
33     MODBUS_SERVER: "$HOME_NET"
```

Rule을 지정하면 체크가 필요한 IP와 필요하지 않은 IP의 구분이 가능하다. HOME_NET은 자신이 속한 IP 혹은 그룹을 지정할 수 있다. EXTERNAL_NET은 외부의 IP를 지정하며 대부분은 HOME_NET을 제외한 나머지를 지정한다.

```
37   port-groups:
38     HTTP_PORTS: "80"
39     SHELLCODE_PORTS: "!80"
40     ORACLE_PORTS: 1521
41     SSH_PORTS: 22
42     DNP3_PORTS: 20000
43     MODBUS_PORTS: 502
44     FILE_DATA_PORTS: "[$HTTP_PORTS,110,143]"
45     FTP_PORTS: 21
```

위와 같이 Port그룹도 지정이 가능하다.

Max-pending-packets

```
966 # If you are using the CUDA pattern matcher (mpm-algo: ac-cuda), differe
    nt rules
967 # apply. In that case try something like 60000 or more. This is because
    the CUDA
968 # pattern matcher buffers and scans as many packets as possible in paral
    lel.
969 #max-pending-packets: 1024
```

동시에 처리할 수 있는 최대의 패킷 수를 의미한다.

Runmodes

```
971 # Runmode the engine should use. Please check --list-runmodes to get the
     available
972 # runmodes for each packet acquisition method. Defaults to "autofp" (aut
    o flow pinned
973 # load balancing).
974 #runmode: autofp
```

멀티 쓰레드를 활성화하는 runmode 관련 옵션이 존재한다.

default-packet-size

```
988 # Preallocated size for packet. Default is 1514 which is the classical
989 # size for pcap on ethernet. You should adjust this value to the highest
990 # packet size (MTU + hardware header) on your system.
991 #default-packet-size: 1514
```

미리 패킷을 할당시킬 수 있는 패킷의 크기. Max-pending-packets이 활성화 되어있을 때, 메모리 안에 Suricata가 패킷을 가질 수 있는 크기.

user-group

```
924 # Run suricata as user and group.
925 #run-as:
926 #  user: suri
927 #  group: suri
```

Suricata를 실행시킬 수 있는 유저와 그룹을 지정한다.

PID-File

```
933 # Default location of the pid file. The pid file is only used in
934 # daemon mode (start Suricata with -D). If not running in daemon mode
935 # the --pidfile command line option must be used to create a pid file.
936 #pid-file: /var/run/suricata.pid
```

Deamon Mode로 실행 시, PID file의 이름을 지정한다. 이 파일은 suricata의 PID를 저장한다.

Default logging directory

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

로그를 저장시킬 기본 디렉토리를 설정한다.

Outputs - Line based alerts log (fast.log)

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

fast.log의 설정을 지정한다.

```
 {ICMP} 10.0.2.15:8 -> 192.168.56.1:0
07/02/2018-13:35:08.089574  [**] [1:1000001:0] ICMP Ping Test [**] [Classification: (null)] [Priority: 3]
 {ICMP} 192.168.56.1:0 -> 10.0.2.15:0
07/02/2018-13:35:09.089191  [**] [1:1000001:0] ICMP Ping Test [**] [Classification: (null)] [Priority: 3]
 {ICMP} 10.0.2.15:8 -> 192.168.56.1:0
07/02/2018-13:35:09.090447  [**] [1:1000001:0] ICMP Ping Test [**] [Classification: (null)] [Priority: 3]
 {ICMP} 192.168.56.1:0 -> 10.0.2.15:0
```

기초적인 suricata의 log파일. 전체적인 suricata의 활동 내역을 볼 수 있다.

예시	07/03/2018-16:56:20.662161 [**] [1:1000001:0] ICMP Ping Test [**] [Classification: (null)] [Priority: 3] {ICMP} 10.0.2.15:13 -> 192.168.56.1:0
설명	timestamp(MM/dd/YYYY-HH:mm:ss).usec [gid:sid:rev] massage [] [classification:] [Priotiyu] {protocol} source IP :source port -> destination IP: destination Port

Eve JSON output

```
141   # Extensible Event Format (nicknamed EVE) event log in JSON format
142   - eve-log:
143       enabled: yes
144       filetype: regular #regular|syslog|unix_dgram|unix_stream|redis
145       filename: eve.json
146       #prefix: "@cee: " # prefix to prepend to each log entry
147       # the following are valid when type: syslog above
148       #identity: "suricata"
149       #facility: local5
150       #level: Info ## possible levels: Emergency, Alert, Critical,
151                    ## Error, Warning, Notice, Info, Debug
152       #redis:
153       #  server: 127.0.0.1
```

EVE-log는 Json형식의 결과물로 다양한 경고와 이벤트 설정이 가능하다.

```
393 {"timestamp":"2018-07-03T12:10:29.431068+0900","flow_id":181071142782161
    ,"in_iface":"enp0s3","event_type":"http","src_ip":"10.0.2.2","src_port":
    49627,"dest_ip":"10.0.2.15","dest_port":5601,"proto":"TCP","tx_id":1,"ht
    tp":{"hostname":"192.168.56.1","url":"\/bundles\/0c6dccf6e8d60a35330c60a
    8831b1c08.svg","http_user_agent":"Mozilla\/5.0 (Windows NT 10.0; Win64;
    x64) AppleWebKit\/537.36 (KHTML, like Gecko) Chrome\/67.0.3396.99 Safari
    \/537.36","http_content_type":"image\/svg+xml","http_refer":"http:\/\/19
    2.168.56.1:5601\/app\/kibana","http_method":"GET","protocol":"HTTP\/1.1"
    ,"status":304,"length":0}}
394 {"timestamp":"2018-07-03T12:10:29.433752+0900","flow_id":130612064610203
    1,"in_iface":"enp0s3","event_type":"http","src_ip":"10.0.2.2","src_port"
    :49628,"dest_ip":"10.0.2.15","dest_port":5601,"proto":"TCP","tx_id":1,"h
    ttp":{"hostname":"192.168.56.1","url":"\/bundles\/71676fac241b6e6e516140
    74cf72152c.svg","http_user_agent":"Mozilla\/5.0 (Windows NT 10.0; Win64;
     x64) AppleWebKit\/537.36 (KHTML, like Gecko) Chrome\/67.0.3396.99 Safar
    i\/537.36","http_content_type":"image\/svg+xml","http_refer":"http:\/\/1
    92.168.56.1:5601\/app\/kibana","http_method":"GET","protocol":"HTTP\/1.1
    ","status":304,"length":0}}
```

Type에 따라 출력되는 로그의 내용이 달라진다. 타입의 종류는 Alert, HTTP, DNS, TLS, TFTP, SMB가 존재하며 각 Type에 따라 내용이 달라진다.

예시		{"timestamp":"2018-07-03T12:10:29.433752+0900","flow_id":1306120646102031,"in_iface":"enp0s3","event_type":"http","src_ip":"10.0.2.2","src_port":49628,"dest_ip":"10.0.2.15","dest_port":5601,"proto":"TCP","tx_id":1,"http":{"hostname":"192.168.56.1","url":"\/bundles\/71676fac241b6e6e51614074cf72152c.svg","http_user_agent":"Mozilla\/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit\/537.36 (KHTML, like Gecko) Chrome\/67.0.3396.99 Safari\/537.36","http_content_type":"image\/svg+xml","http_refer":"http:\/\/192.168.56.1:5601\/app\/kibana","http_method":"GET","protocol":"HTTP\/1.1","status":304,"length":0}}
형식		{"timestamp":"yyyy-mm-24T21:27:09.534255","event_type":"TYPE", ...tuple... ,"TYPE":{ ... type specific content ... }}
type 내용 예시	Alert,	"alert": { "action": "allowed", "gid": 1, "signature_id": 1, "rev": 1, "app_proto": "http", "signature": "HTTP body talking about corruption", "severity": 3, "source": { "ip": "192.168.43.32", "port": 36292 }, "target": { "ip": "179.60.192.3" "port": 80 },
	HTTP,	"http": { "hostname": "www.digip.org", "url" :"\/jansson\/releases\/jansson-2.6.tar.gz", "http_user_agent": "<User-Agent>", "http_content_type": "application\/x-gzip"}
	DNS,	"dns": { "type": "query", "id": 16000, "rrname": "twitter.com", "rrtype":"A"}
	TLS,	"tls": { "subject": "C=US, ST=California, L=Mountain View, O=Google Inc, CN=*.google.com", "issuerdn": "C=US, O=Google Inc, CN=Google Internet Authority G2"}
	TFTP,	"tftp": { "packet": "write", "file": "rfc1350.txt", "mode": "octet" }
	SMB	"smb": { "id": 1, "dialect": "unknown", "command": "SMB2_COMMAND_CREATE", "status": "STATUS_SUCCESS", "status_code": "0x0", "session_id": 4398046511201, "tree_id": 1, "filename": "atsvc", "disposition": "FILE_OPEN", "access": "normal", "created": 0, "accessed": 0, "modified": 0, "changed": 0, "size": 0, "fuid": 000004d-0000-0000-0005-0000ffffffff"}

output for use with Barnyard2

```
254
255   # alert output for use with Barnyard2
256   - unified2-alert:
257       enabled: no
258       filename: unified2.alert
259
```

다른 IDS나 또는 Barnyard2에서 사용되기 위한 바이너리 형식의 로그 포맷.

이 설정은 Barnyard2가 있어야 사용 가능하다.

A line based log of HTTP requests (http.log)

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

HTTP 트래픽 이벤트를 전부 기록하는 로그 설정이며 HTTP 요청, Hostname, URI의 내용을 기록한다.

```
06/29/2018-17:27:44.961256 192.168.56.1[**]/[**]Mozilla/5.0 (Windows NT 10.0; Wi
n64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/67.0.3396.99 Safari/537.
36[**]10.0.2.2:52147 -> 10.0.2.15:80
06/29/2018-17:27:44.997678 192.168.56.1[**]/noindex/css/fonts/Bold/OpenSans-Bold
.woff[**]Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, li
ke Gecko) Chrome/67.0.3396.99 Safari/537.36[**]10.0.2.2:52147 -> 10.0.2.15:80
06/29/2018-17:27:44.998778 192.168.56.1[**]/noindex/css/fonts/Light/OpenSans-Lig
ht.woff[**]Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML,
like Gecko) Chrome/67.0.3396.99 Safari/537.36[**]10.0.2.2:52148 -> 10.0.2.15:80
06/29/2018-17:27:45.022060 192.168.56.1[**]/noindex/css/fonts/Bold/OpenSans-Bold
.ttf[**]Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, lik
e Gecko) Chrome/67.0.3396.99 Safari/537.36[**]10.0.2.2:52148 -> 10.0.2.15:80
06/29/2018-17:27:45.022189 192.168.56.1[**]/noindex/css/fonts/Light/OpenSans-Lig
ht.ttf[**]Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, l
ike Gecko) Chrome/67.0.3396.99 Safari/537.36[**]10.0.2.2:52147 -> 10.0.2.15:80
[root@localhost ~]#
```

http 활동을 기록하는 http.log 설정.

예시	07/03/2018-18:17:24.772585192.168.56.1[**]/wordpress/wp-admin/admin-ajax.php[**]Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/67.0.3396.99 Safari/537.36[**]10.0.2.2:59233 -> 10.0.2.15:80
형식	"%{%D-%H:%M:%S}t.%z %{X-Forwarded-For}i %H %m %h %u %s %B %a:%p -> %A:%P"
설명	(날짜-시-분-초)단위의 시간스탬프 + "." + 마이크로 초+XFF 헤더 + Request 방식 + 호스트의 HTTP 헤더 + URL Query + return 코드 + User Agent + 응답 바이트 크기+Client IP + ":" + Client 포트 +"->" + 서버 IP + ":" +서버 포트

A line based log of DNS queries and replies (dns.log)

```
325   # a line based log of DNS requests and/or replies (no alerts)
326   - dns-log:
327       enabled: no
328       filename: dns.log
329       append: yes
330       #filetype: regular # 'regular', 'unix_stream' or 'unix_dgram'
```

DNS 이벤트를 전부 기록하는 로그 설정이며 DNS가 수행되거나 요청/응답되었을 때 클라이언트, 서버, ttl의 내용을 기록한다.

Packet Log(pcap-log)

```
362   - pcap-log:
363       enabled: no
364       filename: log.pcap
365
366       # File size limit.  Can be specified in kb, mb, gb.  Just a number
367       # is parsed as bytes.
368       limit: 1000mb
```

Suricata에 등록된 모든 패킷들을 저장하는 pcap-log option이다. 이 옵션으로 언제든지 패킷을 볼 수 있다. sugil(네트워크 보안 모니터링 소프트웨어)와 같이 사용할 수 있다.

tcpdump -qns 0 -X -r /var/log/suricata/log.pcap.xxxx을 통해 pcap파일 확인.

```
17:34:45.881449 IP 10.0.2.2.51221 > 10.0.2.15.ssh: tcp 64
        0x0000:  4500 0068 ee9e 0000 4006 73e1 0a00 0202  E..h....@.s.....
        0x0010:  0a00 020f c815 0016 093d 551e ab01 1e51  .........=U....Q
        0x0020:  5018 ffff 6e88 0000 940e 3849 30de bcf6  P...n.....8I0...
        0x0030:  f6a1 7aac 9985 fbdb ddc8 b48d 8a83 d3ef  ..z.............
        0x0040:  cc98 576e 55e3 5c01 425e f4b7 2ecd 0128  ..WnU.\.B^.....(
        0x0050:  bb85 c570 e1a5 7461 b841 bbdc 09e7 8ee8  ...p..ta.A......
        0x0060:  8e31 478b 5302 3dd1                      .1G.S.=.
17:34:45.882077 IP 10.0.2.15.ssh > 10.0.2.2.51221: tcp 64
        0x0000:  4510 0068 3c85 4000 4006 e5ea 0a00 020f  E..h<.@.@.......
        0x0010:  0a00 0202 0016 c815 ab01 1e51 093d 555e  ...........Q.=U^
        0x0020:  5018 8e50 186b 0000 535c eef3 f019 5783  P..P.k..S\....W.
        0x0030:  780f 350c d6cf 5232 0f9c 9884 d317 fd14  x.5...R2........
        0x0040:  c88c 2dab ebe0 dd0f 2ab1 9b8c a36b 573a  ..-.....*....kW:
        0x0050:  ffdf dc81 f0e0 fb2b 8380 a995 b334 bbb7  .......+.....4..
        0x0060:  e18e f783 cd14 48b8                      ......H.
```

다음과 같이 들어온 패킷을 상세하게 볼 수 있다.

예시	18:26:11.277402 IP 10.0.2.15.ssh > 10.0.2.2.49508: tcp 0 0x0000: 4510 0028 338c 4000 4006 ef23 0a00 020f E..(3.@.@..#.... 0x0010: 0a00 0202 0016 c164 db2f 33d7 4b26 905e ....d./3.K&.^ 0x0020: 5010 e040 182b 0000 P..@.+..
설명	timestamp(hh:mm:ss.usec) Source IP > Destination IP 패킷 내용

Verbose Alerts Log(alert-debug.log)

```
383   # a full alerts log containing much information for signature writers
384   # or for investigating suspected false positives.
385   - alert-debug:
386       enabled: no
387       filename: alert-debug.log
388       append: yes
389       #filetype: regular # 'regular', 'unix_stream' or 'unix_dgram'
```

alert에 대해서 보충적인 정보를 제공해주는 로그이며 특히 positive가 실패한 자료를 수집하기에 적합하다.
하지만 저장해야 할 정보량이 많아지기 때문에 성능이 떨어질 수 있다.

```
+================
TIME:              07/02/2018-13:30:54.816301
PKT SRC:           wire/pcap
SRC IP:            192.168.56.1
DST IP:            10.0.2.15
PROTO:             1
FLOW:              to_server: FALSE, to_client: FALSE
PACKET LEN:        98
PACKET:
 0000  08 00 27 8C 83 6C 52 54  00 12 35 02 08 00 45 00   ..'..lRT ..5...E.
 0010  00 54 70 2A 00 00 7F 01  C6 C6 C0 A8 38 01 0A 00   .Tp*.... ....8...
 0020  02 0F 00 00 5B 98 17 1D  00 02 7E AA 39 5B 00 00   ....[... ..~.9[..
 0030  00 00 0A 70 0C 00 00 00  00 00 10 11 12 13 14 15   ...p.... ........
 0040  16 17 18 19 1A 1B 1C 1D  1E 1F 20 21 22 23 24 25   ........ .. !"#$%
 0050  26 27 28 29 2A 2B 2C 2D  2E 2F 30 31 32 33 34 35   &'()*+,- ./012345
 0060  36 37                                             67
ALERT CNT:            1
ALERT MSG [00]:       ICMP Ping Test
ALERT GID [00]:       1
ALERT SID [00]:       1000001
ALERT REV [00]:       0
ALERT CLASS [00]:     <none>
ALERT PRIO [00]:      3
ALERT FOUND IN [00]: PACKET
ALERT IN TX [00]:     N/A
PAYLOAD LEN:          56
PAYLOAD:
 0000  7E AA 39 5B 00 00 00 00  0A 70 0C 00 00 00 00 00   ~.9[.... .p......
 0010  10 11 12 13 14 15 16 17  18 19 1A 1B 1C 1D 1E 1F   ........ ........
 0020  20 21 22 23 24 25 26 27  28 29 2A 2B 2C 2D 2E 2F    !"#$%&' ()*+,-./
 0030  30 31 32 33 34 35 36 37                          01234567
```

fast.log보다 들어온 정보에 대해 자세히 알려준다

Stats

```
399   # Stats.log contains data from various counters of the suricata engine.
400   - stats:
401       enabled: yes
402       filename: stats.log
403       append: yes       # append to file (yes) or overwrite it (no)
404       totals: yes       # stats for all threads merged together
405       threads: no       # per thread stats
406       #null-values: yes  # print counters that have value 0
```

시간에 경과에 따라 여러 내용을 기록할 수 있다.

```
-------------------------------------------------------------------------------
----
Date: 6/29/2018 -- 17:34:45 (uptime: 0d, 00h 04m 30s)
-------------------------------------------------------------------------------
----
Counter                                     | TM Name                   | Value
-------------------------------------------------------------------------------
----
capture.kernel_packets                      | Total                     | 69380
capture.kernel_drops                        | Total                     | 46725
decoder.pkts                                | Total                     | 22654
decoder.bytes                               | Total                     | 2695536
1
decoder.ipv4                                | Total                     | 22652
decoder.ipv6                                | Total                     | 1
decoder.ethernet                            | Total                     | 22654
decoder.tcp                                 | Total                     | 22552
decoder.udp                                 | Total                     | 100
decoder.teredo                              | Total                     | 1
decoder.avg_pkt_size                        | Total                     | 1189
decoder.max_pkt_size                        | Total                     | 1514
flow.tcp                                    | Total                     | 47
flow.udp                                    | Total                     | 42
tcp.sessions                                | Total                     | 38
tcp.syn                                     | Total                     | 39
tcp.synack                                  | Total                     | 38
tcp.rst                                     | Total                     | 7
tcp.stream_depth_reached                    | Total                     | 2
tcp.reassembly_gap                          | Total                     | 1
detect.alert                                | Total                     | 2
app_layer.flow.http                         | Total                     | 5
app_layer.tx.http                           | Total                     | 6
app_layer.flow.tls                          | Total                     | 2
app_layer.flow.failed_tcp                   | Total                     | 1
```

suricata가 실행되는 동안 들어온 모든 signitual를 기록한다.

syslog

```
408   # a line based alerts log similar to fast.log into syslog
409   - syslog:
410       enabled: no
411       # reported identity to syslog. If ommited the program name (usually
412       # suricata) will be used.
413       #identity: "suricata"
414       facility: local5
415       #level: Info ## possible levels: Emergency, Alert, Critical,
416                    ## Error, Warning, Notice, Info, Debug
```

모든 경고와 이벤트를 syslog의 결과로 보낸다.

Drop.log, a line based information for dropped packets

```
418   # a line based information for dropped packets in IPS mode
419   - drop:
420       enabled: no
421       filename: drop.log
422       append: yes
423       #filetype: regular # 'regular', 'unix_stream' or 'unix_dgram'
```

Suricata가 IPS모드로 작동중이라면 Drop 패킷의 관한 로그를 기록한다.

참고 사이트 : https://suricata.readthedocs.io/en/latest/configuration/suricata-yaml.html

-끝-