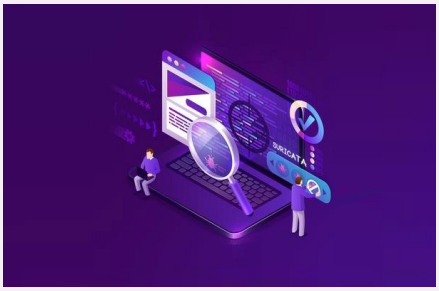
Suricata NSM: More than an IDS Cheat Sheet

by mN3m0N1c (g33k247) via cheatography.com/163410/cs/34239/

Network Security Monitoring with Suricata



Getting Started

- Easy to deploy in just a few steps
- Purpose built to do 1 thing well
- Installs easily on your favorite platform
- Configurable to your network architecture
- Write/Tune/Update/Delete your rules
- Export/forward data to a SIEM (e.g., Splunk)

Building Suricata from Source

```
$ tar xzvf suricata-6.0.0.tar.gz
$ cd suricata-6.0.0
$ ./configure
$ make && sudo make install
```

Installing Suricata from a Repository

```
$ sudo add-apt-repository ppa:oisf/suricata-stable
$ sudo apt update
$ sudo apt install suricata jq
$ sudo suricata --build-info
$ sudo systemctl status suricata
$ sudo vim /etc/suricata/suricata.yaml
$ sudo systemctl restart suricata
```

Checking the Suricata Configuration

```
$ suricata --build-info|grep -A 3 '\-\-prefix'
$ suricata --dump-config
$ suricata --list-app-layer-protos
```

Running Suricata in IDS or IPS Mode

```
$ sudo vim /etc/suricata/suricata.yaml
#LISTENMODE=af-packet #IDS[1]
#LISTENMODE=nfqueue #IPS[1]
```

[1] Remove # to activate mode (default: IDS).

Suricata File Locations

```
/etc/suricata/classification.config #class-types
/etc/suricata/rules #files end in .rules
/etc/suricata/rules/custom.rules #local rules
/etc/suricata/suricata.yaml #config file
/etc/suricata/threshold.config #limit firing
/etc/suricata/variables.config #local defined
```

Suricata Monitoring (/var/log/suricata)

```
$ tail -f /var/log/suricata/fast.log (alerts)
$ tail -f /var/log/suricata/http.log (http requests)
$ tail -f /var/log/suricata/suricata.log (changes)
$ tail -f /var/log/suricata/stats.log (counters)
```

Suricata Monitoring (JSON Log)

```
$ jq 'select(.event_type=="alert")' /var/log/-suricata/eve.json #watch all alerts fire
$ jq 'select(.alert .signature_id==2100498)' /var/log/suricata/eve.json #specific alert fire
$ jq 'select(.event_type=="stats")' /var/log/-suricata/eve.json #monitor statistics
$ jq 'select(.event_type=="stats")|.stats.c-apture.kernel_packets' /var/log/suricata/-eve.json #kernel stats
```

Troubleshooting Suricata

```
$ sudo suricata -c /etc/suricata/suricat-a.yaml -T -vvv #check config/look for <Notice> - Configuration provided was successfully loaded. Exiting. in the output
$ sudo systemctl restart suricata #restart
```

Troubleshooting Suricata (cont)

```
$ sudo systemctl status -l suricata #status
$ grep -Ril <SID#> #get flagged SID in rules
```

[ERRCODE: SC_ERR_CONF_YAML_ER-ROR(242)] - App-Layer protocol sip enable status not set (Enable in suricata.yaml app-layer stanza)

Writing Suricata Rules

- Target the vulnerability, not the exploit
- Target activity outside normal hours
- Target to eliminate traffic of no interest
- Target IP ranges based on your network
- Target unusual conns, ports/protocols
- Test, tune, and validate!

Rules Protocols (Basic)

icmp	ip[1]
tcp	udp

[1] ip stands for 'all' or 'any'

Rules Protocols (Application Layer)

dcerpc	dhcp	dnp3[1]	dns
enip[1]	ftp	http	http2
ikev2	imap	krb5	modbus*
nfs	ntp	rfb	rdp
sip	smb	smtp	snmp
ssh	tftp	tls (incl ssl)	

[1] disabled by default

Suricata Rules: The 3 Elements of a Rule

- **action** (what happens on a rule match)
- **header** (protocol, address, port, direction)
- **options** (specifics of the rule)

By **mN3m0N1c** (g33k247)
cheatography.com/g33k247/

Published 20th October, 2022.
Last updated 13th January, 2023.

Suricata NSM: More than an IDS Cheat Sheet

by mN3m0N1c (g33k247) via cheatography.com/163410/cs/34239/

Element #1: Rule Actions

- **Alert** generates an alert for later analysis
- **Pass** stops scanning, allows packet to pass, no alert
- **Drop** (IPS) stops processing/creates alert
- **Reject** (IPS) rst sent, matching packet dropped
- **Rejectsrc** same as just reject
- **Rejectdst** send RST/ICMP error packet to receiver of the matching packet
- **Rejectboth** send RST/ICMP error packets to both sides of the conversation

Element #2: Rule Headers

<proto><src_ip><port> -> <dst_ip><port>

- ip any any -> any any
- tcp $EXTERNAL_NET any -> 10.200.0.0/24 80
- ssh any any -> 203.0.113.0/24 !2
- tcp $EXTERNAL_NET any -> $HOME_NET 80
- source -> destination
- source <> destination (both directions)

Element #3: Rule Options

- arguments contain options/keyword modifiers
- match in packet, classify rule, log custom messages
- separated by ";" (may use *key: value* format)

Rule Classtypes

- categorizes traffic: **config classification:-shortname,short description,priority**
- 3 fields (machine readable name, description, priority): **config classification: bad-unknown,Potentially Bad Traffic, 2**

Rule Priority

- implicit priority assigned by **classtype** in /etc/suricata/classification.config
- to override classtype default priority add the **priority:n** option to a signature (where n is 1 to 255)

Rule Reference Keyword

- find more info/links (Do NOT put http:// into reference string, assumed with url)
- use to tag CVE **reference:cve,2014-0160;** or MITRE T-Codes **reference:tcode,1194;**

Rule Numbering (SID Allocation)

- 1000000-1999999 Custom
- 2000000-2099999 Emerging Threats (ET)
- 2100000-2103999 Forked Snort GPL
- 2200000-2200999 Decoder events
- 2210000-2210999 Stream events
- 2220000-2299999 Reserved
- 2800000-2899999 ET Pro (subscrip. only)
- 2400000-2528999 Dynamically updated

Note: Signature ID (SID) provided as last keyword (or second-to-last if a rev # included) in the rule

Example Rule for ICMP Ping

alert icmp any any -> any any (msg:"PING detected"; sid:2; rev:1;)

Example Rule for HTTP GET Request

alert http $HOME_NET any -> $EXTERNAL_NET any (msg:"HTTP GET Request Containing Rule in URI"; flow:established,to_server; http.method; content:"GET"; http.uri; content:"rule"; fast_pattern; classtype:bad-unknown; sid:123; rev:1;)

Reloading Rules

$ sudo systemctl restart suricata

$ sudo kill -USR1 $(pidof suricata)[1]

$ sudo suricatasc -c reload-rules[2]

[1] When enabled in suricata.yaml (preferred)
[2] When using the Unix socket feature

Triggering Rules (Testing)

$ curl http://testmynids.org/uid/index.html
$ grep 2100498 /var/log/suricata/fast.log

Thresholding Rules (Criteria)

- Limit to 1 alert every 60 seconds for sid #2404000: **threshold gen_id 1, sig_id 2404000, type threshold, track by_dst, count 1, seconds 60**
- Limit to 10 alerts every 60 seconds for each *source* host: **threshold gen_id 0, sig_id 0, type threshold, track by_src, count 10, seconds 60**

Suppressing Rules (Traffic)

- suppress to ensure no alerts are generated (suppression only considered post-matching): **suppress gen_id 0, sig_id 0, track by_src, ip 1.2.3.4**

Offline Processing

- Read pcap files just like network traffic

$ sudo suricata -c /etc/suricata/suricata.yaml -r <pcap_location> -l <where to log results>^1

$ tail -f /var/log/suricata/fast.log

[1] *Same directory as pcap file if -l not used*

Learn More

ET Rulesets	Suricata Docs
ET Rule Changes	Suricata User Guide
ET Rule SID Lookup	User Agent Strings
Oinkmaster	Talos VRT Rules
Snorpy Generator	YouTube Playlist
Suricata Config File	

By **mN3m0N1c** (g33k247)
cheatography.com/g33k247/

Published 20th October, 2022.
Last updated 13th January, 2023.