IA612: Intrusion Detection and Prevention

St. Cloud State University

LAB-04: Suricata IDS/IPS Installation and Configuration

---

1. What is Suricata

Suricata is a high-performance Network IDS, IPS and Network Security Monitoring engine. It is open source and owned by a community-run non-profit foundation, the Open Information Security Foundation (OISF). Suricata is developed by the OISF.

1.1. About the Open Information Security Foundation

The Open Information Security Foundation is a non-profit foundation organized to build community and to support open-source security technologies like Suricata, the world-class IDS/IPS engine.

1.1.1. License

The Suricata source code is licensed under version 2 of the GNU General Public License. This documentation is licensed under the Creative Commons Attribution-NonCommercial 4.0 International Public License.

2. Installation

Before Suricata can be used it has to be installed. Suricata can be installed on various distributions using binary packages: For people familiar with compiling their own software, the Source method is recommended.

2.1. Source

Installing from the source distribution files gives the most control over the Suricata installation.

Basic steps:

```
tar xzvf suricata-4.0.0.tar.gz
cd suricata-4.0.0
./configure
make
make install
```

This will install Suricata into /usr/local/bin/, use the default configuration in:
/usr/local/etc/suricata/ and will output to /usr/local/var/log/suricata

2.1.1. Common configure options

--disable-gccmarch-native

Do not optimize the binary for the hardware it is built on. Add this flag if the binary is meant to be portable or if Suricata is to be used in a VM.

--prefix=/usr/

Installs the Suricata binary into /usr/bin/. Default /usr/local/

--sysconfdir=/etc

Installs the Suricata configuration files into /etc/suricata/. Default /usr/local/etc/

--localstatedir=/var

Setups Suricata for logging into /var/log/suricata/. Default /usr/local/var/log/suricata

--enable-lua

Enables Lua support for detection and output.

--enable-geopip

Enables GeoIP support for detection.

--enable-rust

Enables experimental Rust support.

2.1.2. Dependencies

For Suricata's compilation you'll need the following libraries and their development headers installed:

libpcap, libpcre, libmagic, zlib, libyaml

The following tools are required:

make gcc (or clang) pkg-config

For full features, also add:

libjansson, libnss, libgeoip, liblua5.1, libhiredis, libevent

Rust support (experimental):

rustc, cargo

2.1.2.1. Ubuntu/Debian

Minimal:

```
apt-get install libpcre3 libpcre3-dbg libpcre3-dev build-essential libpcap-dev   \
        libyaml-0-2 libyaml-dev pkg-config zlib1g zlib1g-dev \
        make libmagic-dev
```

Recommended:

```
apt-get install libpcre3 libpcre3-dbg libpcre3-dev build-essential libpcap-dev   \
        libnet1-dev libyaml-0-2 libyaml-dev pkg-config zlib1g zlib1g-dev \
        libcap-ng-dev libcap-ng0 make libmagic-dev libjansson-dev       \
        libnss3-dev libgeoip-dev liblua5.1-dev libhiredis-dev libevent-dev
```

Extra for iptables/nftables IPS integration:

```
apt-get install libnetfilter-queue-dev libnetfilter-queue1  \
        libnetfilter-log-dev libnetfilter-log1      \
        libnfnetlink-dev libnfnetlink0
```

For Rust support (Ubuntu only):

```
apt-get install rustc cargo
```

2.2. Binary packages

2.2.1. Ubuntu

For Ubuntu, the OISF maintains a PPA `suricata-stable` that always contains the latest stable release.

To use it:

```
sudo add-apt-repository ppa:oisf/suricata-stable
sudo apt-get update
sudo apt-get install suricata
```

2.2.2. Debian

In Debian 9 (Stretch) do:

```
apt-get install suricata
```

In Debian Jessie Suricata is out of date, but an updated version is in Debian Backports.

As root do:

```
echo "deb http://http.debian.net/debian jessie-backports main" > \
    /etc/apt/sources.list.d/backports.list
apt-get update
apt-get install suricata -t jessie-backports
```

2.2.3. Fedora

```
dnf install suricata
```

2.2.4. RHEL/CentOS

For RedHat Enterprise Linux 7 and CentOS 7 the EPEL repository can be used.

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

2.3. Advanced Installation

Various installation guides for installing from GIT and for other operating systems are maintained at:

https://redmine.openinfosecfoundation.org/projects/suricata/wiki/Suricata_Installation

3. Command Line Options

Suricata's command line options are as follows:

-h

Display a brief usage overview.

-V

Displays the version of Suricata.

-c <path>

Path to configuration file.

-T

Test configuration.

-v

The -v option enables more verbosity of Suricata's output. Supply multiple times for more verbosity.

-r <path>

Run in pcap offline mode reading files from pcap file.

-i <interface>

After the -i option you can enter the interface card you would like to use to sniff packets from. This option will try to use the best capture method available.

--pcap[=<device>]

Run in PCAP mode. If no device is provided the interfaces provided in the pcap section of the configuration file will be used.

--af-packet[=<device>]

Enable capture of packet using AF_PACKET on Linux. If no device is supplied, the list of devices from the af-packet section in the yaml is used.

-q <queue id>

Run inline of the NFQUEUE queue ID provided. May be provided multiple times.

-s <filename.rules>

With the -s option you can set a file with signatures, which will be loaded together with the rules set in the yaml.

-S <filename.rules>

With the -S option you can set a file with signatures, which will be loaded exclusively, regardless of the rules set in the yaml.

-l <directory>

With the -l option you can set the default log directory. If you already have the default-log-dir set in yaml, it will not be used by Suricata if you use the -l option. It will use the log dir that is set with the -l option. If you do not set a directory with the -l option, Suricata will use the directory that is set in yaml.

-D

Normally if you run Suricata on your console, it keeps your console occupied. You can not use it for other purposes, and when you close the window, Suricata stops running. If you run Suricata as deamon (using the -D option), it runs at the background and you will be able to use the console for other tasks without disturbing the engine running.

--runmode <runmode>

With the –runmode option you can set the runmode that you would like to use. This command line option can override the yaml runmode option.

Runmodes are: workers, autofp and single.

For more information about runmodes see Runmodes in the user guide.

-F <bpf filter file>

Use BPF filter from file.

-k [all|none]

Force (all) the checksum check or disable (none) all checksum checks.

--user=<user>

Set the process user after initialization. Overrides the user provided in the run-as section of the configuration file.

--group=<group>

Set the process group to group after initialization. Overrides the group provided in the run-assection of the configuration file.

--pidfile <file>

Write the process ID to file. Overrides the pid-file option in the configuration file and forces the file to be written when not running as a daemon.

--init-errors-fatal

Exit with a failure when errors are encountered loading signatures.

--disable-detection

Disable the detection engine.

--dump-config

Dump the configuration loaded from the configuration file to the terminal and exit.

--build-info

Display the build information the Suricata was built with.

--list-app-layer-protos

List all supported application layer protocols.

--list-keywords=[all|csv|<kword>]

List all supported rule keywords.

--list-runmodes

List all supported run modes.

--set <key>=<value>

Set a configuration value. Useful for overriding basic configuration parameters in the configuration. For example, to change the default log directory:

--set default-log-dir=/var/tmp

--engine-analysis

Print reports on analysis of different sections in the engine and exit. Please have a look at the conf parameter engine-analysis on what reports can be printed

--unix-socket=<file>

Use file as the Suricata unix control socket. Overrides the filename provided in the unix-command section of the configuration file.

--pcap-buffer-size=<size>

Set the size of the PCAP buffer (0 - 2147483647).

--netmap[=<device>]

Enable capture of packet using NETMAP on FreeBSD or Linux. If no device is supplied, the list of devices from the netmap section in the yaml is used.

--pfring[=<device>]

Enable PF_RING packet capture. If no device provided, the devices in the Suricata configuration will be used.

--pfring-cluster-id <id>

Set the PF_RING cluster ID.

--pfring-cluster-type <type>

Set the PF_RING cluster type (cluster_round_robin, cluster_flow).

-d <divert-port>

Run inline using IPFW divert mode.

--dag <device>

Enable packet capture off a DAG card. If capturing off a specific stream the stream can be select using a device name like “dag0:4”. This option may be provided multiple times read off multiple devices and/or streams.

--napatech

Enable packet capture using the Napatech Streams API.

--mpipe

Enable packet capture using the TileGX mpipe interface.

--erf-in=<file>

Run in offline mode reading the specific ERF file (Endace extensible record format).

--simulate-ips

Simulate IPS mode when running in a non-IPS mode.

3.1. Unit Tests

Builtin unittests are only available if Suricata has been built with –enable-unittests.Running unittests does not take a configuration file. Use -l to supply an output directory.

`-u`

Run the unit tests and exit. Requires that Suricata be compiled with –enable-unittests.

`-U, --unittest-filter=REGEX`

With the -U option you can select which of the unit tests you want to run. This option uses REGEX. Example of use: suricata -u -U http

`--list-unittests`

List all unit tests.

`--fatal-unittests`

Enables fatal failure on a unit test error. Suricata will exit instead of continuuing more tests.

`--unittests-coverage`

Display unit test coverage report.

4. Configuring and testing Suricata

4.1 Overview

By default, Suricata doesn't log anything to disk. In this exercise we will be telling Suricata what types of logs that you would like it to create. In the following steps we will enable Suricata to log to files, turn on the TLS and DNS parsers and test that Suricata is working properly by sending test traffic through the system.

4.1.1 Configuring Suricata to log to disk

- Configure the "logging outputs" for Suricata

```
# editor /etc/suricata/suricata.yaml
```

- On line number 787 of the suricata.yaml configuration file, enable "file" logging by changing the value of the "enabled" key values set to "yes" from "no". When you have completed this step, your config file should look like the following excerpt:

```
# Define your logging outputs.  If none are defined, or they are all
# disabled you will get the default - console output.
outputs:
- console:
    enabled: yes
- file:
    enabled: yes
    filename: /var/log/suricata.log
- syslog:
    enabled: no
    facility: local5
    format: "[%i] <%d> -- "
```

4.1.2 Configuring Suricata to enable DNS and TLS logging

```
# editor /etc/suricata/suricata.yaml
```

- Around line number 40 you will see the following configuration directives. Ensure that http-log, tls-log and dns-log have the "enabled" key values sent to "yes". When you have completed this step, your config file should look like the following excerpt:

```
  - http-log:
      enabled: yes
      filename: http.log
      append: yes
      #extended: yes     # enable this for extended logging information
      #custom: yes       # enabled 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: yes  # Log TLS connections.
      filename: tls.log # File to store TLS logs.
      append: yes
      #filetype: regular # 'regular', 'unix_stream' or 'unix_dgram'
      #extended: yes # Log extended information like fingerprint
      certs-log-dir: certs # directory to store the certificates files

  # a line based log of DNS requests and/or replies (no alerts)
  - dns-log:
      enabled: yes
      filename: dns.log
      append: yes
```

4.1.3 Restart Suricata to apply changes

- When you are finished with making changes to the suricata.yaml, issue the following command to restart Suricata:

```
# service suricata restart
```

4.1.4 Verifying that Suricata is logging protocol metadata

- When Suricata is working correctly, you should be able to see the logs it generates in the directory: '/var/log/suricata'
- In order to test Suricata, lets first initate a DNS query:

```
# apt-get install dnsutils
# host nsrc.org
```

- Now check the DNS log to ensure that the query was logged:

```
# tail -f /var/log/suricata/dns.log
```

- If your query was successful, you should see the following information in the log:

```
07/16/2015-01:18:52.555394 [**] Query TX 54ab [**] nsrc.org [**] A [**] 10.0.2.15:37770 ->
10.0.2.3:53
07/16/2015-01:18:52.555394 [**] Response TX 54ab [**] Recursion Desired [**] 10.0.2.3:53 ->
10.0.2.15:37770
07/16/2015-01:18:52.555394 [**] Response TX 54ab [**] nsrc.org [**] A [**] TTL 300 [**]
128.223.157.25 [**] 10.0.2.3:53 -> 10.0.2.15:37770
07/16/2015-01:18:52.672384 [**] Query TX f870 [**] nsrc.org [**] AAAA [**] 10.0.2.15:33718 ->
10.0.2.3:53
07/16/2015-01:18:52.672384 [**] Response TX f870 [**] Recursion Desired [**] 10.0.2.3:53 ->
10.0.2.15:33718
07/16/2015-01:18:52.672384 [**] Response TX f870 [**] nsrc.org [**] AAAA [**] TTL 300 [**]
2607:8400:2880:0004:0000:0000:80df:9d1c [**] 10.0.2.3:53 -> 10.0.2.15:33718
07/16/2015-01:18:52.854448 [**] Query TX fa53 [**] nsrc.org [**] MX [**] 10.0.2.15:37918 ->
10.0.2.3:53
07/16/2015-01:18:52.854448 [**] Response TX fa53 [**] Recursion Desired [**] 10.0.2.3:53 ->
10.0.2.15:37918
07/16/2015-01:18:52.854448 [**] Response TX fa53 [**] nsrc.org [**] MX [**] TTL 10 [**] smtp.nsrc.org
[**] 10.0.2.3:53 -> 10.0.2.15:37918
```

4.1.5 Verifying that Suricata is logging signature matches

- Find a rule that you want to match against in the Suricata rules directory. Reccomendation: For testing Suricata, a rule that matches against known-bad user-agent strings is a good test.

```
# less /etc/suricata/rules/emerging-user_agents.rules
```

- Look for the following rule in 'emerging-user_agents.rules' if it exists in your ruleset then continue on to the next step, if not, ask the instructor for help.

```
alert http $HOME_NET any -> $EXTERNAL_NET any (msg:"ET USER_AGENTS Suspicious User Agent (BlackSun)";
flow:to_server,established; content:"User-Agent|3a| BlackSun"; nocase; http_header;
reference:url,www.bitdefender.com/VIRUS-1000328-en--Trojan.Pws.Wow.NCY.html;
reference:url,doc.emergingthreats.net/bin/view/Main/2008983; classtype:trojan-activity; sid:2008983;
rev:6;)
```

- What this rule says is: "Any time Suricata sees the HTTP user-agent string "BlackSun" please alert me. User-agent strings are sometimes used by malware authors as an authentication token -- the command-and-control server will not issue commands to computers that make requests of it unless the correct user-agent string is specified by the client in the HTTP session. This is one way malware authors evade malware researchers. Luckily for security professionals, these user-agent strings can be very good indicators of malware presence on a system. This is why user-agent strings are included in Suricata rules.
- Finally, issue traffic to google.com using the user-agent string "BlackSun"

```
# apt-get install curl
# curl -A "BlackSun" www.google.com
```

- If your Suricata instance is operating correctly, you should see the following line end up in your "fast.log" in /var/log/suricata:

```
07/16/2015-01:32:12.275324  [**] [1:2008983:6] ET USER_AGENTS Suspicious User Agent (BlackSun) [**]
[Classification: A Network Trojan was detected] [Priority: 1] {TCP} 10.0.2.15:49779 ->
74.125.28.99:80
```

4.2 Summary

In this exercise we learn to enable the DNS and TLS parsers for Suricata, check the logs for the protocol parsers and test out signatures from emerging threats that indicate malwares presence on a system. Suricata logs are located in: '/var/log/suricata'. The 'dns.log' and 'tls.log' are used to keep metadata extracted from network protocols, the 'fast.log' is used to keep alerts that arise from integrating and matcing against threat intelligence with Suricata.

5. Setting up IPS/inline for Linux

In this guide will be explained how to work with Suricata in layer3 inline mode and how to set iptables for that purpose.

First start with compiling Suricata with NFQ support. For instructions see Ubuntu Installation. For more information about NFQ and iptables, see NFQ.

To check if you have NFQ enabled in your Suricata, enter the following command:

- `suricata --build-info`

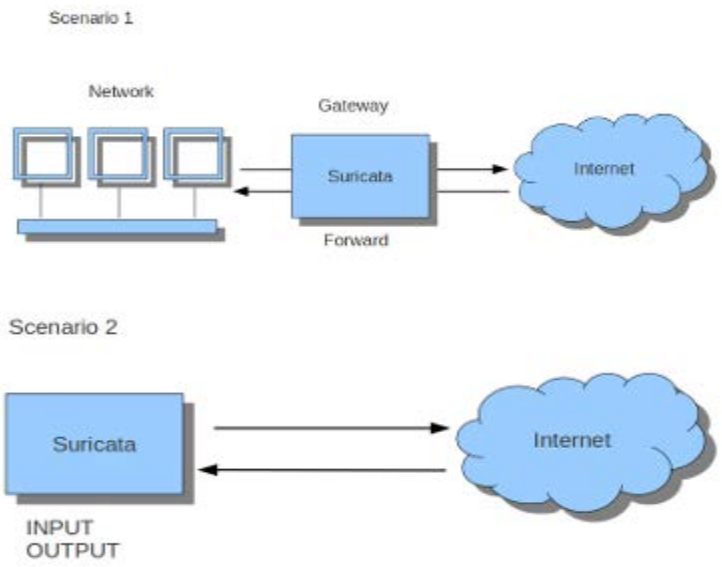
and examine if you have NFQ between the features.

To run suricata with the NFQ mode, you have to make use of the -q option. This option tells Suricata which of the queue numbers it should use.

- `sudo suricata -c /etc/suricata/suricata.yaml -q 0`

5.1. Iptables configuration

First of all it is important to know which traffic you would like to send to Suricata. Traffic that passes your computer or traffic that is generated by your computer.



If Suricata is running on a gateway and is meant to protect the computers behind that gateway you are dealing with the first scenario: *forward_ing* . If Suricata has to protect the computer it is running on, you are dealing with the second scenario: *host* (see drawing 2). These two ways of using Suricata can also be combined.

The easiest rule in case of the gateway-scenario to send traffic to Suricata is:

- `sudo iptables -I FORWARD -j NFQUEUE`

In this case, all forwarded traffic goes to Suricata.

In case of the host situation, these are the two most simple iptable rules;

- `sudo iptables -I INPUT -j NFQUEUE`
- `sudo iptables -I OUTPUT -j NFQUEUE`

It is possible to set a queue number. If you do not, the queue number will be 0 by default.

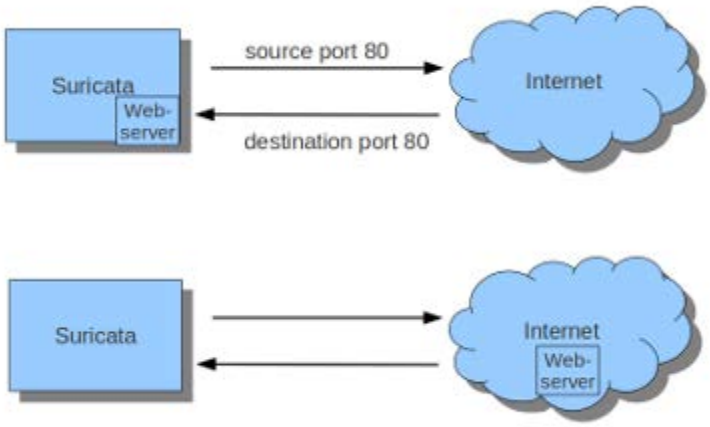
Imagine you want Suricata to check for example just TCP-traffic, or all incoming traffic on port 80, or all traffic on destination-port 80, you can do so like this:

- `sudo iptables -I INPUT -p tcp  -j NFQUEUE`
- `sudo iptables -I OUTPUT -p tcp -j NFQUEUE`

In this case, Suricata checks just TCP traffic.

- `sudo iptables -I INPUT -p tcp --sport 80  -j NFQUEUE`
- `sudo iptables -I OUTPUT -p tcp --dport 80 -j NFQUEUE`

In this example, Suricata checks all input and output on port 80.



To see if you have set your iptables rules correct make sure Suricata is running and enter:

- `sudo iptables -vnL`

In the example you can see if packets are being logged.

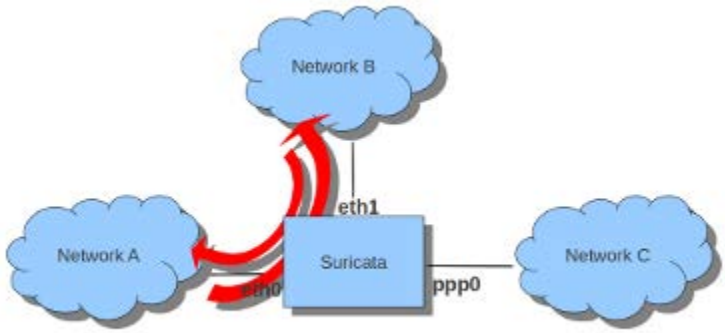
```
anne-fleur@t60:~$ sudo iptables -vnL
Chain INPUT (policy ACCEPT 258 packets, 43900 bytes)
 pkts bytes target     prot opt in     out     source               destination
 4979 5846K NFQUEUE    tcp  --  *      *       0.0.0.0/0            0.0.0.0/0           tcp spt:80 NFQUEUE num 0

Chain FORWARD (policy ACCEPT 0 packets, 0 bytes)
 pkts bytes target     prot opt in     out     source               destination

Chain OUTPUT (policy ACCEPT 278 packets, 43459 bytes)
 pkts bytes target     prot opt in     out     source               destination
 5206  388K NFQUEUE    tcp  --  *      *       0.0.0.0/0            0.0.0.0/0           tcp dpt:80 NFQUEUE num 0
anne-fleur@t60:~$
```

This description of the use of iptables is the way to use it with IPv4. To use it with IPv6 all previous mentioned commands have to start with ‘ip6tables’. It is also possible to let Suricata check both kinds of traffic.

There is also a way to use iptables with multiple networks (and interface cards). Example:



- `sudo iptables -I FORWARD -i eth0 -o eth1 -j NFQUEUE`
- `sudo iptables -I FORWARD -i eth1 -o eth0 -j NFQUEUE`

The options -i (input) -o (output) can be combined with all previous mentioned options

If you would stop Suricata and use internet, the traffic will not come through. To make internet work correctly, you have to erase all iptable rules.

To erase all iptable rules, enter:

- `sudo iptables -F`

6. Deliverables

1. Read the lab instructions above and finish all the tasks. Take screenshots where necessary (For all important steps) to add to your well organized lab report. At each step write what you did in brief in your own words.

2. Configure “Elasticsearch, Logstash, and Kibana (ELK)” data visualization tools to see the log files generated by the “Suricata” in this lab. After that add the clear and well organized steps into the lab report that how you configure the ELK to view “Suricata” log files.

7. Submission Instruction:

1. Complete all the tasks assigned in the deliverable section.
2. Take all necessary and appropriate screenshots.
3. Past those new screenshots to a MS-Word file with the step number.
4. Write a brief description bellow or above each screenshot for each step to make a good report related to this lab.
5. Submit your report to the D2L “LAB-04” drop-box by only one member of each group within one week from the assigned date.

NOTE-1: Please add all group member’s names (LAST_NAME, FIRST_NAME), the group number and page numbers of the report.

NOTE-2: Please add all group member’s contributions to complete and submit this lab as a percentage as shown below at the end of the report. (Before submitting to D2L all of the group members must know or aware their reported contribution as a percentage in the lab report)

Example:

Member-01: 100%
Member-02: 75%
Member-03: 100%
Member-04: 50%

References:

[1]. https://suricata.readthedocs.io/en/suricata-4.0.5/what-is-suricata.html
[2]. https://web.nsrc.org/workshops/2015/pacnog17-ws/raw-attachment/wiki/Track2Agenda/ex-installing-suricata.htm
[3]. https://web.nsrc.org/workshops/2015/pacnog17-ws/raw-attachment/wiki/Track2Agenda/ex-suricata-rules.htm
[4]. https://web.nsrc.org/workshops/2015/pacnog17-ws/raw-attachment/wiki/Track2Agenda/ex-suricata-config-test.htm
[5]. https://suricata.readthedocs.io/en/suricata-4.0.5/setting-up-ipsinline-for-linux.html