Install Suricata on CentOS 8 from Source

Run system update

Update your system package by running the command below

```
dnf update
```

Install Required Build tools and Dependencies

There are a number of package dependencies and build tools required for a successful build and install of Suricata on CentOS 8 from the source.

```
dnf config-manager --set-enabled PowerTools
```

```
dnf install diffutils file-devel gcc jansson-devel make nss-devel libyaml-devel libcap-ng-devel libpcap-devel pcre-devel python3 python3-pyyaml rust-toolset zlib-devel curl wget tar lua lua-devel lz4-devel
```

Download Suricata Source Code

Download the latest stable release Suricata source code from Suricata downloads page. As of this writing, Suricata 5.0.3 is the latest stable release version.

```
wget https://www.openinfosecfoundation.org/download/suricata-5.0.3.tar.gz -P /tmp
```

Extract Suricata Source Code

Once the download is complete, extract the source code;

```
cd /tmp
```

```
tar xzf suricata-5.0.3.tar.gz
```

Build and Install Suricata on CentOS 8

Navigate to the source directory and build and install Suricata on CentOS 8.

```
cd suricata-5.0.3
```

Run the configure script to adapt Suricata to the system and verify that all required dependencies are in place.

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

The command installs Suricata into `/usr/bin/suricata`, have the config in `/etc/suricata` and use `/var/log/suricata` as log directory.

For more build options, refer to `./configure --help`.

Compile and install Suricata rules and configurations.

```
make

make install-full


...
22/7/2020 -- 21:14:44 - <Info> -- Backing up current rules.
22/7/2020 -- 21:14:44 - <Info> -- Writing rules to /var/lib/suricata/rules/suricata.rules: total: 27530; enabled: 20677; added:
27530; removed 0; modified: 0
22/7/2020 -- 21:14:44 - <Info> -- Skipping test, disabled by configuration.
22/7/2020 -- 21:14:44 - <Info> -- Done.

You can now start suricata by running as root something like:
  /usr/bin/suricata -c /etc/suricata/suricata.yaml -i eth0

If a library like libhtp.so is not found, you can run suricata with:
  LD_LIBRARY_PATH=/usr/lib /usr/bin/suricata -c /etc/suricata/suricata.yaml -i eth0

The Emerging Threats Open rules are now installed. Rules can be
updated and managed with the suricata-update tool.

For more information please see:
  https://suricata.readthedocs.io/en/latest/rule-management/index.html

make[1]: Leaving directory '/tmp/suricata-5.0.3'
```

Suricata is now installed from sources on CentOS 8.

Install Suricata on CentOS 8 from EPEL Repos

For a seamless installation of Suricata on CentOS 8, using EPEL repos to install it is a sure bet.

Install EPEL Repos on CentOS 8

```
dnf install epel-release
```

Install Suricata from EPEL Repos CentOS 8

```
dnf info suricata

Available Packages
Name         : suricata
Version      : 5.0.3
Release      : 1.el8
Architecture : x86_64
```

```
Size         : 2.3 M
Source       : suricata-5.0.3-1.el8.src.rpm
Repository   : epel
Summary      : Intrusion Detection System
URL          : https://suricata-ids.org/
License      : GPLv2
...
```

As you can see, the EPEL repos provides the latest stable release version of Suricata.

You can then install it by executing the command;

```
dnf install suricata
```

Suricata Rules

Suricata utilizes various rule sets/signatures to detect and alert on matching threats. Rules are also known as Signatures. Emerging Threats, Emerging Threats Pro and source fire's VRT are the most commonly used rules.

In most cases, you can find the rules files under `/etc/suricata/rules/`. This is when you install Suricata from repos.

```
ls /etc/suricata/rules/

app-layer-events.rules  dhcp-events.rules  dns-events.rules  http-events.rules   kerberos-events.rules  nfs-events.rules  smb-events.rules   stream-events.rules
decoder-events.rules    dnp3-events.rules  files.rules       ipsec-events.rules  modbus-events.rules    ntp-events.rules  smtp-events.rules  tls-events.rules
```

Emergency Threat rules are usually stored as **`/var/lib/suricata/rules/suricata.rules`**. The **suricata.rules** file usually contains all the rules defined on the rules file located under the **`/etc/suricata/rules/`**.

To install and update Emergency Threat rules, use the **`suricata-update`** command.

```
suricata-update
```

This downloads and installs `suricata.rules`.

A rule/signature consists of the following sections:

- The **action**, that determines what happens when the signature matches.
- The **header**, defining the protocol, IP addresses, ports and direction of the rule.
- The **rule options**, defining the specifics of the rule.

```
alert ip any any -> any any (msg:"SURICATA Applayer Mismatch protocol both directions"; flow:established; app-layer-
event:applayer_mismatch_protocol_both_directions; flowint:applayer.anomaly.count,+,1; classtype:protocol-command-decode;
sid:2260000; rev:1;)
```

Read more on introduction to Suricata rules.

Suricata Basic Setup

`/etc/suricata/suricata.yaml` is the default Suricata configuration file.

The configuration file contains a lot of configurable options. However, for our basic setup, we will only focus on the network interface on which Suricata is listening on and the IP address attached to that interface.

To find the interface and the IP address, run the command below;

```
ip --brief add

lo               UNKNOWN        127.0.0.1/8 ::1/128
enp0s3           UP             10.0.2.15/24
enp0s8           UP             192.168.56.133/24 fe80::12c8:9a8a:6d1:deaf/64
```

In our case, our interface is **enp0s8** and the IP address is **`192.168.56.133`**.

Open and edit the Suricata config file.

```
vim /etc/suricata/suricata.yaml
```

Under the **vars** section, you need to configure Suricata to differentiate between your internal network to be protected and external network. This can be done by defining the correct values for the `HOME_NET` and `EXTERNAL_NET` variables respectively under the address groups.

The **`HOME_NET`** variable should include the IP address of the interface on which Suricata is listening on and all the local networks to protect.

The `EXTERNAL_NET` variables should define any IP or network that is not local.

```
...
vars:
  # more specific is better for alert accuracy and performance
  address-groups:
    #HOME_NET: "[192.168.0.0/16,10.0.0.0/8,172.16.0.0/12]"
    HOME_NET: "[192.168.56.133]"
    #HOME_NET: "[192.168.0.0/16]"
    #HOME_NET: "[10.0.0.0/8]"
    #HOME_NET: "[172.16.0.0/12]"
    #HOME_NET: "any"
```

```
    EXTERNAL_NET: "!$HOME_NET"
...
```

Under the **`af-packet`** section, set the value of the interface to your interface name.

```
...
# Linux high speed capture support
af-packet:
  - interface: enp0s8
...
```

Save and exit the configuration file.

Specify Suricata Rules

Define the Suricata rules-files to use. We are using the default ET rules in this demo;

```
...
default-rule-path: /var/lib/suricata/rules

rule-files:
  - suricata.rules
...
```

Disable Packet Offloading

Disable Suricata packet offloading by disabling interface Large Receive Offload (LRO)/Generic Receive Offload (GRO);

```
ethtool -K <interface> gro off lro off
```

Replace `<interface>` with your interface.

First check if these features are enabled;

```
ethtool -k enp0s8 | grep -iE "generic|large"

        tx-checksum-ip-generic: on
generic-segmentation-offload: on
generic-receive-offload: off
large-receive-offload: off [fixed]
```

If enabled, disable by running the command below;

```
ethtool -K enp0s8 gro off lro off
```

Running Suricata

Suricata can be managed by a systemd service.

Before you can run it, you need to specify the interface on which it is listening in **`/etc/sysconfig/suricata`** config file.

```
vim /etc/sysconfig/suricata

...

# Add options to be passed to the daemon
#OPTIONS="-i eth0 --user suricata "
OPTIONS="-i enp0s8 --user suricata "
```

Save and exit the file,

Start and enable Suricata to run on boot on CentOS 8.

```
systemctl enable --now suricata
```

You can check the status;

```
systemctl status suricata

● suricata.service - Suricata Intrusion Detection Service
   Loaded: loaded (/usr/lib/systemd/system/suricata.service; enabled; vendor preset: disabled)
   Active: active (running) since Thu 2020-07-23 16:50:34 EAT; 29s ago
     Docs: man:suricata(1)
  Process: 19153 ExecStartPre=/bin/rm -f /var/run/suricata.pid (code=exited, status=0/SUCCESS)
 Main PID: 19154 (Suricata-Main)
    Tasks: 7 (limit: 5027)
   Memory: 387.6M
   CGroup: /system.slice/suricata.service
           └─19154 /sbin/suricata -c /etc/suricata/suricata.yaml --pidfile /var/run/suricata.pid -i enp0s8 --user suricata

Jul 23 16:50:34 ceph-admin.kifarunix-demo.com systemd[1]: Starting Suricata Intrusion Detection Service...
Jul 23 16:50:34 ceph-admin.kifarunix-demo.com systemd[1]: Started Suricata Intrusion Detection Service.
Jul 23 16:50:34 ceph-admin.kifarunix-demo.com suricata[19154]: 23/7/2020 -- 16:50:34 - <Notice> - This is Suricata version 5.0.3
RELEASE running in SYSTEM mode
Jul 23 16:50:42 ceph-admin.kifarunix-demo.com suricata[19154]: 23/7/2020 -- 16:50:42 - <Notice> - all 1 packet processing threads,
4 management threads initialized,
```

Note that instead of using s systemd service above, you can run Suricata with a simple command;

```
suricata -D -c /etc/suricata/suricata.yaml -i enp0s8
```

Suricata logging;

To check if Suricata is running check the Suricata log:

```
tail /var/log/suricata/suricata.log
```

You should see such a line;

```
...
23/7/2020 -- 16:50:42 - - all 1 packet processing threads, 4 management threads initialized, engine started.
```

To check Suricata statistics;

```
tail -f /var/log/suricata/stats.log
```

To check Suricata alert logs;

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

Suricata can also write logs in EVE Json output. The default log file is;

```
tail -f /var/log/suricata/eve.json
```

Testing Suricata Rules

In this demo, we are using the default ET Suricata rules. If you have created you own custom rules, be sure to test the Suricata rules for syntax errors;

```
suricata -c /etc/suricata/suricata.yaml -T -v

23/7/2020 -- 17:44:10 - - Running suricata under test mode
23/7/2020 -- 17:44:10 - - This is Suricata version 5.0.3 RELEASE running in SYSTEM mode
23/7/2020 -- 17:44:10 - - CPUs/cores online: 1
23/7/2020 -- 17:44:10 - - fast output device (regular) initialized: fast.log
23/7/2020 -- 17:44:10 - - eve-log output device (regular) initialized: eve.json
23/7/2020 -- 17:44:10 - - stats output device (regular) initialized: stats.log
23/7/2020 -- 17:44:12 - - 1 rule files processed. 20676 rules successfully loaded, 0 rules failed
23/7/2020 -- 17:44:12 - - Threshold config parsed: 0 rule(s) found
23/7/2020 -- 17:44:12 - - 20679 signatures processed. 1138 are IP-only rules, 3987 are inspecting packet payload, 15324 inspect
application layer, 103 are decoder event only
23/7/2020 -- 17:44:25 - - Configuration provided was successfully loaded. Exiting.
23/7/2020 -- 17:44:25 - - cleaning up signature grouping structure… complete
```

Then restart Suricata;

```
systemctl restart suricata
```

Perform SSH DDoS Test Attack

On another system, install hping3 tool and perform an SSH DDoS test attack.

```
dnf install hping3
```

Then attack SSH on the server running Suricata.

```
hping3 -S -p 22 --flood --rand-source 192.168.56.133
```

Refer to `man hping3`.

While hping is running, tail the alert logs on Suricata server;

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

You should see such log lines;

```
...
07/24/2020-21:43:02.613445 [] [1:2400000:2768] ET DROP Spamhaus DROP Listed Traffic Inbound group 1 [] [Classification: Misc
Attack] [Priority: 2] {TCP} 42.163.214.132:4391 -> 192.168.56.133:22
07/24/2020-21:43:02.751133 [] [1:2400007:2768] ET DROP Spamhaus DROP Listed Traffic Inbound group 8 [] [Classification: Misc
Attack] [Priority: 2] {TCP} 122.8.52.209:11845 -> 192.168.56.133:22
07/24/2020-21:43:02.800769 [] [1:2400012:2768] ET DROP Spamhaus DROP Listed Traffic Inbound group 13 [] [Classification: Misc
Attack] [Priority: 2] {TCP} 160.184.221.156:15315 -> 192.168.56.133:22
07/24/2020-21:43:02.801827 [] [1:2400009:2768] ET DROP Spamhaus DROP Listed Traffic Inbound group 10 [] [Classification: Misc
Attack] [Priority: 2] {TCP} 139.81.59.221:15607 -> 192.168.56.133:22
07/24/2020-21:43:02.802528 [] [1:2400013:2768] ET DROP Spamhaus DROP Listed Traffic Inbound group 14 [] [Classification: Misc
Attack] [Priority: 2] {TCP} 163.198.206.175:15818 -> 192.168.56.133:22
07/24/2020-21:43:02.803033 [] [1:2400021:2768] ET DROP Spamhaus DROP Listed Traffic Inbound group 22 [] [Classification: Misc
Attack] [Priority: 2] {TCP} 196.194.135.87:15970 -> 192.168.56.133:22
07/24/2020-21:43:02.803268 [] [1:2400006:2768] ET DROP Spamhaus DROP Listed Traffic Inbound group 7 [] [Classification: Misc
Attack] [Priority: 2] {TCP} 110.41.189.155:16042 -> 192.168.56.133:22
07/24/2020-21:43:02.803548 [] [1:2400009:2768] ET DROP Spamhaus DROP Listed Traffic Inbound group 10 [] [Classification: Misc
Attack] [Priority: 2] {TCP} 143.135.26.50:16131 -> 192.168.56.133:22
07/24/2020-21:43:02.870288 [] [1:2400021:2768] ET DROP Spamhaus DROP Listed Traffic Inbound group 22 [] [Classification: Misc
Attack] [Priority: 2] {TCP} 196.194.107.110:19140 -> 192.168.56.133:22
07/24/2020-21:43:02.871212 [] [1:2400003:2768] ET DROP Spamhaus DROP Listed Traffic Inbound group 4 [] [Classification: Misc
Attack] [Priority: 2] {TCP} 101.194.46.143:19453 -> 192.168.56.133:22
07/24/2020-21:43:02.871608 [] [1:2400021:2768] ET DROP Spamhaus DROP Listed Traffic Inbound group 22 [] [Classification: Misc
Attack] [Priority: 2] {TCP} 196.16.182.33:19588 -> 192.168.56.133:22
```

. . .

With that simple test, we can see that Suricata is setup and running well using the default Emergency Threat rules.