

Building Custom IDS Sensor Suricata & Zeek

Contents

Introduction and Goal of this Document	2
Building Elasticsearch Server with TLS Communications.....	2
Building CentOS7 Sensor.....	2
Configure NIC card & Hostname.....	2
Update Sensor and add the following packages.....	2
Add Zeek Directory Path to Profile	3
Create Zeek & Suricata Log Directories	3
Extract tarball to /	4
Install and Configure Zeek	5
Configure Zeek log directories	5
Create the Following Zeek Directories.....	6
Configuring Zeek Nodes & Networks	6
Starting Zeek	7
Install and Configure Suricata	9
Configuring Suricata.....	9
Setup root Cronjob.....	9
Edit and Update suricata.yml.....	12
Adding Suricata Service Startup Script.....	13
Logging Data to Elasticsearch	15
Install Filebeat.....	15
Install Metricbeat to Monitor the Sensor	20
Install and Configure Packetbeat	21
Want to Collect Netflow Data?	25
Add to /etc/rc.local	25
References	26
Annex: Tarball Files and Directories	27

Introduction and Goal of this Document

The primary goal of this document is to provide a framework to build your own sensor(s) using CentOS 7 with Suricata and Zeek. It also has information to capture netflow data using softflowd.

The last step is to use this document to send all the logs to Elasticsearch using filebeat. It also has information to use packetbeat as a replacement or complement to netflow.

Building Elasticsearch Server with TLS Communications

This document is how to add encryption between the sensor(s) and Elasticsearch for secure communication.

[1] https://handlers.sans.edu/gbruneau/elk/TLS_elasticsearch_configuration.pdf

[2] <https://isc.sans.edu/forums/diary/Secure+Communication+using+TLS+in+Elasticsearch/26902/>

Building CentOS7 Sensor

Download and install CentOS7

Sensor needs a minimum of 2 interfaces (management and capture)

Recommend 3 drives:

- Main drive with /, swap, /home, /var/log
- Second drive with Suricata logs
- Third drive with Zeek logs

Configure NIC card & Hostname

Configure the management NIC card with a static IP.

```
$ sudo nmtui
```

Update Sensor and add the following packages:

```
$ sudo yum -y update
```

```
$ sudo yum -y install open-vm-tools ntp bind-utils net-tools wget unzip tcpdump git
```

```
$ sudo yum -y install epel-release htop
$ sudo timedatectl list-timezones
$ sudo timedatectl set-timezone UTC
$ sudo systemctl stop ntpd
$ sudo ntpdate 0.centos.pool.ntp.org
$ sudo systemctl start ntpd
$ sudo su root -
```

Add Zeek Directory Path to Profile

```
$ sudo su -
# vi /root/.bashrc
export PATH=/opt/zeek/bin:$PATH
```

Reload the root profile to include Zeek

```
# ./root/.bashrc
```

Create Zeek & Suricata Log Directories

```
# cfdisk /dev/sdb
# mkfs.xfs /dev/sdb1
# mkdir -p /nsm/suricata
```

```
# cfdisk /dev/sdc
# mkfs.xfs /dev/sdc1
# mkdir -p /nsm/zeek
```

Add these two partitions to /etc/fstab:

```
# vi /etc/fstab
```

```
/dev/sdb1    /nsm/suricata  xfs  defaults  0 0
/dev/sdc1    /nsm/zeek     xfs  defaults  0 0
```

```
# mount -a
```

```
# df -h
```

Suricata logs location: /nsm/suricata

Zeek logs location: /nsm/zeek

Extract tarball to /

I use VMware sensors for my sensors with a prebuilt VM. This tarball has all the scripts and files included in all the steps listed below. Any files or script that need to be create, update or modified to configure the sensor, they are part of this package to speed up getting the sensor built. The tarball can be downloaded at this location.

There are two tarball, the first installation.tgz is to setup all the scripts listed below to install the software and the second tarball is to preconfigure some of the sensor configuration files.

Extract this tarball as follow: `tar zxvf installation.tgz -C /`

Installation script : <https://handlers.sans.edu/gbruneau/scripts/installation.tgz>

Extract this tarball after installing all the binaries: `tar zxvf sensor.tgz -C /`

Sensor script: <https://handlers.sans.edu/gbruneau/scripts/sensor.tgz>

The list of directories and files are listed in the [Annex](#).

Install and Configure Zeek

The Zeek pre-build package is available for download at this location which will add the repo to the sensor.

<https://software.opensuse.org//download.html?project=security%3Azeek&package=zeek>

This blog is about collecting Zeek logs with Elasticsearch:

<https://www.elastic.co/blog/collecting-and-analyzing-zeek-data-with-elastic-security>

```
# cd /etc/yum.repos.d/  
# wget https://download.opensuse.org/repositories/security:zeek/CentOS_7/security:zeek.repo  
# yum -y install zeek
```

→ Zeek is installed in the /opt/zeek directory

Configure Zeek log directories

```
# vi /opt/zeek/etc/zeekctl.cfg
```

```
# Location of the log directory where log files will be archived each rotation
```

```
# interval.
```

```
LogDir = /nsm/zeek/logs
```

```
# Location of the spool directory where files and data that are currently being
```

```
# written are stored.
```

```
SpoolDir = /nsm/zeek/spool
```

```
# Location of the directory in which the databases for Broker datastore backed
```

```
# Zeek tables are stored.
```

```
BrokerDBDir = /nsm/zeek/spool/brokerstore
```

Create the Following Zeek Directories

```
# mkdir -p /nsm/zeek/logs
# mkdir -p /nsm/zeek/spool
# mkdir -p /nsm/zeek/spool/brokerstore
# chown -R root:zeek /nsm/zeek
```

Configuring Zeek Nodes & Networks

Find which network card (ifconfig) is the packet capture and configure Zeek node:

```
# vi /opt/zeek/etc/node.cfg
```

```
[zeek]
```

```
type=standalone
```

```
host=localhost
```

```
interface=ens160
```

Configure the protected network(s):

```
# vi /opt/zeek/etc/networks.cfg
```

Enter the correct network(s) here.

Edit and configure local.zeek to save the data in JSON format. Add the following at the end of the file:

```
# vi /opt/zeek/share/zeek/site/local.zeek
```

```
# Output to JSON
```

```
@load policy/tuning/json-logs.zeek
```

Add the following Zeek Service configuration file to start Zeek when the sensor boot:

```
# vi /etc/systemd/system/zeek.service
```

```
[Unit]
```

```
Description=Zeek Network Intrusion Detection System (NIDS)
```

```
After=network.target
```

```
[Service]
```

```
Type=forking
```

```
User=root
```

```
Group=zeek
```

```
Environment=HOME=/nsm/zeek/spool
```

```
ExecStart=/opt/zeek/bin/zeekctl deploy
```

```
ExecStop=/opt/zeek/bin/zeekctl stop
```

```
[Install]
```

```
WantedBy=multi-user.target
```

Starting Zeek

```
# zeekctl install
```

```
# systemctl daemon-reload
```

```
# systemctl enable zeek
```

```
# systemctl start zeek
```

```
# systemctl status zeek
```

```
[root@idstest yum.repos.d]# systemctl status zeek
● zeek.service - Zeek Network Intrusion Detection System (NIDS)
   Loaded: loaded (/etc/systemd/system/zeek.service; enabled; vendor preset: disabled)
   Active: active (running) since Thu 2021-04-08 13:19:57 GMT; 6s ago
     Process: 2101 ExecStart=/opt/zeek/bin/zeekctl deploy (code=exited, status=0/SUCCESS)
    Main PID: 2133 (bash)
   CGroup: /system.slice/zeek.service
           └─2133 bash /opt/zeek/share/zeekctl/scripts/run-zeek -l -i ens160 -U .status -p zeekctl -p zeekctl-live -p standalone -p l
             └─2139 /opt/zeek/bin/zeek -i ens160 -U .status -p zeekctl -p zeekctl-live -p standalone -p local -p zeek local.zeek zeekct

Apr 08 13:19:57 idstest zeekctl[2101]: installing site policies ...
Apr 08 13:19:57 idstest zeekctl[2101]: generating standalone-layout.zeek ...
Apr 08 13:19:57 idstest zeekctl[2101]: generating local-networks.zeek ...
Apr 08 13:19:57 idstest zeekctl[2101]: generating zeekctl-config.zeek ...
Apr 08 13:19:57 idstest zeekctl[2101]: generating zeekctl-config.sh ...
Apr 08 13:19:57 idstest zeekctl[2101]: stopping ...
Apr 08 13:19:57 idstest zeekctl[2101]: stopping zeek ...
Apr 08 13:19:57 idstest zeekctl[2101]: starting ...
Apr 08 13:19:57 idstest zeekctl[2101]: starting zeek ...
Apr 08 13:19:57 idstest systemd[1]: Started Zeek Network Intrusion Detection System (NIDS).
```

Log location: /nsm/zeek/spool/zeek

Install and Configure Suricata

Suricata pre-build packages 6.x packages information is available at the following URL:

<https://suricata.readthedocs.io/en/suricata-6.0.2/install.html#rhel-centos-8-and-7>

The following example is used to install Suricata 6.0 on CentOS. If you wish to install 5.0 instead, change the version in @oisf/suricata-6.0.

```
# yum -y install epel-release yum-plugin-copr
```

```
# yum -y copr enable @oisf/suricata-6.0
```

```
# yum -y install suricata
```

Configuring Suricata

The following steps assumes that all the events from Suricata will be stored into Elasticsearch, the log files can be removed at regular interval to keep that directory clean.

```
# chown -R suricata:suricata /nsm/suricata
```

Setup root Cronjob

Edit the root crontab and add the following configuration:

```
# crontab -e
```

```
* 0-23 * * * /usr/sbin/logrotate -f /etc/logrotate.conf > /dev/null  
2>1&
```

```
# Remove old gzip files every hours
```

```
5 0-23 * * * /root/scripts/remove_suricata.sh > /dev/null 2>1&
```

```
# Suricata rule update - /var/lib/suricata/rules
```

```
0 12 * * * /usr/bin/suricata-update update --reload-command  
"/usr/bin/systemctl kill -s USR2 suricata" > /var/log/suricata-  
update.log 2>&1
```

Suricata Rules Update: <https://suricata-update.readthedocs.io/en/latest/quickstart.html>

Suricata Update : <https://github.com/OISF/suricata-update>

```
# suricata-update update-sources
```

```
# suricata-update list-sources
```

Configuring Suricata to enable Threshold and the option to disable Signatures and Rulesets:

<https://raw.githubusercontent.com/OISF/suricata-update/master/suricata/update/configs/update.yaml>

```
# cd /etc/suricata
```

```
# wget https://raw.githubusercontent.com/OISF/suricata-  
update/master/suricata/update/configs/disable.conf
```

```
# wget https://raw.githubusercontent.com/OISF/suricata-  
update/master/suricata/update/configs/threshold.in
```

Edit *disable.conf* and disable these group at the end of the file if you are not using them. This will prevent errors when starting Suricata.

```
# vi /etc/suricata/disable.conf
```

```
group: modbus
```

```
group: dnp3
```

Update Suricata Logrotate file with the following additions:

```
# vi /etc/logrotate.d/suricata
```

```
# Sample /etc/logrotate.d/suricata configuration file.
```

```
/nsm/suricata/*.log /nsm/suricata/*.json
```

```
{
```

```
    daily
```

```
    rotate 3
```

```
size 500M
missingok
compress
delaycompress
copytruncate
create 0644 suricata suricata
shardscripts
postrotate

    /bin/kill -HUP `cat /var/run/suricata/suricata.pid
2>/dev/null` 2>/dev/null || true

    # systemctl stop suricata.service
    # systemctl stop filebeat.service
    # systemctl start suricata.service
    # systemctl start filebeat.service
endscript
}
```

Create this script to remove old Suricata files regularly:

```
# mkdir /root/scripts
# vi /root/scripts/remove_suricata.sh

#!/bin/sh
#
# Guy Bruneau, guybruneau@outlook.com
# Date: 17 March 2021
# Version: 1.0

# Remove old gzip file every hours
/usr/bin/rm -f /nsm/suricata/eve.json-*.gz
```

```
/usr/bin/rm -f /nsm/suricata/fast.log-*.gz
/usr/bin/rm -f /nsm/suricata/stats.log-*.gz
/usr/bin/rm -f /nsm/suricata/suricata.log-*.gz
```

```
# chmod 755 /root/scripts/remove_suricata.sh
```

Edit and Update suricata.yml

Update suricata.yml to match the correct network interface, monitored network(s) ranges and log directory.

```
# vi /etc/suricata/suricata.yaml
```

```
HOME_NET: "[192.168.0.0/16,10.0.0.0/8,172.16.0.0/12]"
```

```
af-packet:
```

```
# - interface: eth0
```

```
  - interface: ens160
```

```
# 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: /nsm/suricata/
```

```
##
```

```
## Configure Suricata to load Suricata-Update managed rules.
```

```
##
```

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

rule-files:

- suricata.rules

[Adding Suricata Service Startup Script](#)

Add the following Suricata Service configuration file to start Suricata when the sensor boots:

```
# vi /etc/systemd/system/suricata.service
```

```
[Unit]
```

```
Description=Suricata Intrusion Detection Service
```

```
After=syslog.target network.target
```

```
[Service]
```

```
EnvironmentFile=-/etc/sysconfig/suricata
```

```
ExecStart=/usr/sbin/suricata -c /etc/suricata/suricata.yaml  
/var/run/suricata/suricata.pid --af-packet
```

```
ExecReload=/bin/kill -HUP $MAINPID
```

```
User=suricata
```

```
Group=suricata
```

```
CapabilityBoundingSet=CAP_NET_ADMIN CAP_NET_RAW CAP_IPC_LOCK
```

```
AmbientCapabilities=CAP_NET_ADMIN CAP_NET_RAW CAP_IPC_LOCK
```

```
[Install]
```

```
WantedBy=multi-user.target
```

```
# systemctl daemon-reload
```

```
# systemctl enable suricata
```

```
# systemctl start suricata
```

```
# systemctl status suricata
```

```
[root@idstest ~]# systemctl status suricata
● suricata.service - Suricata Intrusion Detection Service
   Loaded: loaded (/etc/systemd/system/suricata.service; enabled; vendor preset: disabled)
   Active: active (running) since Fri 2021-04-09 17:51:26 GMT; 4s ago
     Main PID: 3567 (Suricata-Main)
    CGroup: /system.slice/suricata.service
            └─3567 /usr/sbin/suricata -c /etc/suricata/suricata.yaml --af-packet

Apr 09 17:51:26 idstest systemd[1]: Started Suricata Intrusion Detection Service.
Apr 09 17:51:26 idstest suricata[3567]: 9/4/2021 -- 17:51:26 - <Notice> - This is Suricata version 6.0.2 RELEASE running in SYSTEM mode
```

Logging Data to Elasticsearch

This section is to configure the sensor to send the logs collected by Suricata and Zeek (or any other applications and services) to Elasticsearch.

Install Filebeat

Install the GPG key and add the repo information.

```
# rpm --import https://artifacts.elastic.co/GPG-KEY-elasticsearch

# vi /etc/yum.repos.d/elasticsearch.repo

[elasticsearch-7.x]
name=Elasticsearch repository for 7.x packages
baseurl=https://artifacts.elastic.co/packages/7.x/yum
gpgcheck=1
gpgkey=https://artifacts.elastic.co/GPG-KEY-elasticsearch
enabled=1
autorefresh=1
type=rpm-md

# yum -y install filebeat
```

Configure Filebeat

```
# cd /etc/filebeat/modules.d
# filebeat modules -help
# filebeat modules list | head
# filebeat modules enable zeek suricata
# filebeat modules enable netflow → enable only if planning to install softflowd
# filebeat modules list | head
```

```
[root@idstest modules.d]# filebeat modules list | head
Enabled:
suricata
zeek

Disabled:
activemq
```

```
# vi /etc/filebeat/modules.d/suricata.yml
```

Add the following path :

```
var.paths: ["/nsm/suricata/eve.json"]
```

```
# Module: suricata
# Docs: https://www.elastic.co/guide/en/beats/filebeat/7.x/filebeat-module-suricata.html

- module: suricata
  # All logs
  eve:
    enabled: true

    # Set custom paths for the log files. If left empty,
    # Filebeat will choose the paths depending on your OS.
    #var.paths:
  var.paths: ["/nsm/suricata/eve.json"]
```

```
# vi /etc/filebeat/modules.d/zeek.yml
```

Set custom var.paths: for all the log files. I set *dnf3* and *modbus* to false. If you are using them, keep them as true

```
var.paths: ["/nsm/zeek/spool/zeek/*.log"]
```



```

# Module: zeek
# Docs: https://www.elastic.co/guide/en/beats/filebeat/7.x/filebeat-module-zeek.html

- module: zeek
  capture_loss:
    enabled: true
    var.paths: ["/nsm/zeek/spool/zeek/*.log"]
  connection:
    enabled: true
    var.paths: ["/nsm/zeek/spool/zeek/*.log"]
  dce_rpc:
    enabled: true
    var.paths: ["/nsm/zeek/spool/zeek/*.log"]
  dhcp:
    enabled: true
    var.paths: ["/nsm/zeek/spool/zeek/*.log"]
  dnp3:
    enabled: false
    var.paths: ["/nsm/zeek/spool/zeek/*.log"]

```

If you are going to use [softflowd](#) verify the port and the network.

```
# vi /etc/filebeat/modules.d/netflow.yml
```

Setup filebeat.yml to Elasticsearch

It is time configure filebeat to send the logs to Elasticsearch and configure the network location of Elasticsearch, some processors and enable x-pack monitoring.

```
# vi /etc/filebeat/filebeat.yml
```

→ *Goto Elasticsearch Output and Configure where Elasticsearch is located and if SSL encryption is used between the sensor and Elasticsearch*

```
# ----- Elasticsearch Output -----
```

```
output.elasticsearch:
```

```
# Array of hosts to connect to.
```

```
hosts: ["localhost:9200"]
```

```
# loadbalance: true
```

```
# pipeline: geoip-info
```

```
# Protocol - either `http` (default) or `https`.
```

```
#protocol: "https"
```

```
# Authentication credentials - either API key or username/password.
```

```
#api_key: "id:api_key"
```

```
#username: "elastic"
```

```
#password: "changeme"
```

```
queue.mem:
```

```
events: 4096
```

```
flush.min_events: 512
```

```
flush.timeout: 5s
```

<https://www.maxmind.com/en/geop2-precision-demo>

→ Goto Processors and add the JSON decode processor for Zeek and Suricata:

```
# ===== Processors =====
```

```
processors:
```

- add_host_metadata: ~
- copy_fields:
 - fields:
 - from: source.ip
 - to: source.address
 - fail_on_error: false
 - ignore_missing: true
- copy_fields:
 - fields:
 - from: destination.ip
 - to: destination.address
 - fail_on_error: false
 - ignore_missing: true

```
processors:
  - add_host_metadata: ~
  - add_fields:
      when.network.source.address: private
      fields:
        source.geo.location:
          lat: 45.3316
          lon: -75.6718
        source.geo.continent_name: North America
        source.geo.city_name: Ottawa
        source.geo.country_iso_code: CA
        source.geo.region_iso_code: CA-ON
        source.geo.region_name: Ontario
      target: ''
  - add_fields:
      when.network.destination.address: private
      fields:
        source.geo.location:
          lat: 45.3316
          lon: -75.6718
        source.geo.continent_name: North America
        source.geo.city_name: Ottawa
        source.geo.country_iso_code: CA
        source.geo.region_iso_code: CA-ON
        source.geo.region_name: Ontario
      target: ''
```

→ If you find filebeat is logging to much stuff, you can change the logging level to /var/log/messages

```
# ===== Logging =====
```

Sets log level. The default log level is info.

Available log levels are: error, warning, info, debug

#logging.level: debug

logging.level: error

===== X-Pack Monitoring =====

Set to true to enable the monitoring reporter.

monitoring.enabled: true

Testing Configuration and Enabling Filebeat

filebeat test config

filebeat test output

filebeat setup --pipelines

filebeat setup --index-management

systemctl enable filebeat

systemctl start filebeat

systemctl status filebeat

[Install Metricbeat to Monitor the Sensor](#)

Metricbeat provides statistics about the sensor.

yum -y install metricbeat

vi /etc/metricbeat/metricbeat.yml

→ *Goto Elasticsearch Output and Configure where Elasticsearch is located and if SSL encryption is used between the sensor and Elasticsearch*

----- Elasticsearch Output -----

output.elasticsearch:

Array of hosts to connect to.

hosts: ["localhost:9200"]

Protocol - either `http` (default) or `https`.

#protocol: "https"

Authentication credentials - either API key or username/password.

#api_key: "id:api_key"

#username: "elastic"

#password: "changeme"

cd /etc/metricbeat/modules.d

ls -l system.yml

system.yml is enabled by default

metricbeat modules list | head

metricbeat test config

metricbeat test output

systemctl enable metricbeat

systemctl start metricbeat

systemctl status metricbeat

Install and Configure Packetbeat

<https://www.elastic.co/beats/packetbeat>

Packetbeat is a lightweight packet analyzer that can be used to inspect certain type of traffic and provide flow data.

```
# yum -y install packetbeat
# vi /etc/packetbeat/packetbeat.yml

#===== Network device =====

# Select the network interface to sniff the data. On Linux, you can use the
# "any" keyword to sniff on all connected interfaces.

packetbeat.interfaces.device: ens160
packetbeat.interfaces.snaplen: 1514
packetbeat.interfaces.type: af_packet
packetbeat.interfaces.buffer_size_mb: 100

→ Review this section and modify as required. Suggested update for DNS, HTTP and TLS
https://www.elastic.co/guide/en/beats/packetbeat/current/configuring-howto-packetbeat.html

#===== Transaction protocols =====

- type: dns
# Configure the ports where to listen for DNS traffic. You can disable
# the DNS protocol by commenting out the list of ports.

ports: [53,5353]
include_authorities: true
include_additional: true
send_request: true
send_response: true

- type: http
# Configure the ports where to listen for HTTP traffic. You can disable
# the HTTP protocol by commenting out the list of ports.

ports: [80, 81, 5000, 7001, 7780, 8000, 8002, 8008, 8080, 8088]
decode_body: true
send_request: true
```

send_response: true

- type: tls

Configure the ports where to listen for TLS traffic. You can disable
the TLS protocol by commenting out the list of ports.

send_certificates: true

include_raw_certificates: false

include_detailed_fields: true

fingerprints: [md5, sha1, sha256]

ports:

- 443 # HTTPS

- 993 # IMAPS

- 995 # POP3S

- 4443

- 5223 # XMPP over SSL

- 8443

- 8883 # Secure MQTT

- 9243 # Elasticsearch

- 10443

→ Goto Elasticsearch Output and Configure where Elasticsearch is located and if SSL encryption is used between the sensor and Elasticsearch

----- Elasticsearch Output -----

output.elasticsearch:

Array of hosts to connect to.

hosts: ["localhost:9200"]

loadbalance: true

Note: make sure **geoip-info** has been loaded into Stack Management → *Ingest Node Pipelines*

pipeline: geoip-info

Protocol - either `http` (default) or `https`.

#protocol: "https"

Authentication credentials - either API key or username/password.

#api_key: "id:api_key"

#username: "elastic"

#password: "changeme"

#===== Processors =====

processors:

- add_host_metadata: ~

- add_fields:

 when.network.source.ip: private

 fields:

 source.geo.location:

 lat: 45.3316

 lon: -75.6718

 source.geo.continent_name: North America

 source.geo.city_name: Ottawa

 source.geo.country_iso_code: CA

 source.geo.region_iso_code: CA-ON

 source.geo.region_name: Ontario

 target: "

- add_fields:

 when.network.destination.ip: private


```
fields:
  destination.geo.location:
    lat: 45.3316
    lon: -75.6718
  destination.geo.continent_name: North America
  destination.geo.city_name: Ottawa
  destination.geo.country_iso_code: CA
  destination.geo.region_iso_code: CA-ON
  destination.geo.region_name: Ontario
target: "
```

```
#===== X-Pack Monitoring =====
# Set to true to enable the monitoring reporter.
monitoring.enabled: true
```

Want to Collect Netflow Data?

Get the tarball from: <https://github.com/irino/softflowd>

Install softflowd /usr/local/sbin

Add to /etc/rc.local

```
# Netflow data
softflowd -i ens160 -v 9 -P udp -n 127.0.0.1:2055

# chmod 755 /etc/rc.local
```

References

- [1] <https://www.elastic.co/guide/en/elasticsearch/reference/current/install-elasticsearch.html>
- [2] <https://www.elastic.co/guide/en/beats/filebeat/current/configuring-howto-filebeat.html>
- [3] <https://www.elastic.co/guide/en/beats/metricbeat/current/configuring-howto-metricbeat.html>
- [4] <https://www.elastic.co/guide/en/beats/packetbeat/current/configuring-howto-packetbeat.html>
- [5] <https://suricata-update.readthedocs.io/en/latest/quickstart.html>

Annex: Tarball Files and Directories

./etc:

filebeat

logrotate.d

packetbeat

rc.local

suricata

systemd

yum.repos.d

./etc/filebeat:

filebeat.yml

modules.d

./etc/filebeat/modules.d:

netflow.yml

suricata.yml

zeek.yml

./etc/logrotate.d:

suricata

./etc/packetbeat:

packetbeat.yml

./etc/suricata:

disable.conf

threshold.in

suricata.yaml

./etc/systemd:

system

./etc/systemd/system:

suricata.service

zeek.service

./etc/yum.repos.d:

copr@oisf-suricata-6.0.repo

elasticsearch.repo

epel.repo

epel-testing.repo

security:zeek.repo

./nsm:

suricata

zeek

./nsm/suricata:

./nsm/zeek:

logs

spool

./nsm/zeek/logs:

./nsm/zeek/spool:

brokerstore

./nsm/zeek/spool/brokerstore:

./opt:

zeek

./opt/zeek:

etc

share

./opt/zeek/etc:

node.cfg

zeekctl.cfg

./opt/zeek/share:

zeek

./opt/zeek/share/zeek:

site

./opt/zeek/share/zeek/site:

local.zeek

./root:

scripts

./root/scripts:

remove_suricata.sh

./usr:

local

./usr/local:

sbin

./usr/local/sbin:

softflowctl

softflowd

./var:

spool

./var/spool:

cron

./var/spool/cron:

root