

Настройка Suricata в Ubuntu 22.04

Suricata - это система обнаружения вторжений (IDS) и система предотвращения вторжений (IPS) с открытым исходным кодом, предназначенная для обнаружения подозрительной активности в сетевом трафике.

Она является популярной альтернативой **Snort** и предлагает ряд преимуществ, таких как:

- **Производительность:** Suricata работает значительно быстрее, чем Snort, благодаря использованию многопоточной архитектуры и оптимизированному движку.
- **Точность:** Suricata имеет более низкий уровень ложных срабатываний, чем Snort, благодаря более совершенным методам обнаружения.
- **Гибкость:** Suricata поддерживает широкий спектр правил и подписей, а также позволяет создавать собственные правила.
- **Простота использования:** Suricata имеет простой и понятный интерфейс командной строки, а также веб-интерфейс для удобного управления.

Suricata может использоваться для:

- **Обнаружения атак:** Suricata может обнаруживать широкий спектр атак, включая сетевые атаки, атаки на приложения и атаки на основе эксплойтов.
- **Предотвращения атак:** Suricata может блокировать атаки до их завершения, защищая вашу сеть от вреда.
- **Мониторинга сетевого трафика:** Suricata может использоваться для мониторинга сетевого трафика и выявления подозрительной активности.

Suricata используется:

- **Системными администраторами:** для защиты своих сетей от атак.
- **Провайдерами услуг Интернета (ISP):** для защиты своих клиентов от атак.
- **Исследователями безопасности:** для изучения и разработки методов обнаружения вторжений.

Suricata консольная утилита и не имеет встроенного веб-интерфейса для управления или мониторинга, однако она может интегрироваться с различными веб-интерфейсами и платформами для удобства анализа и управления.

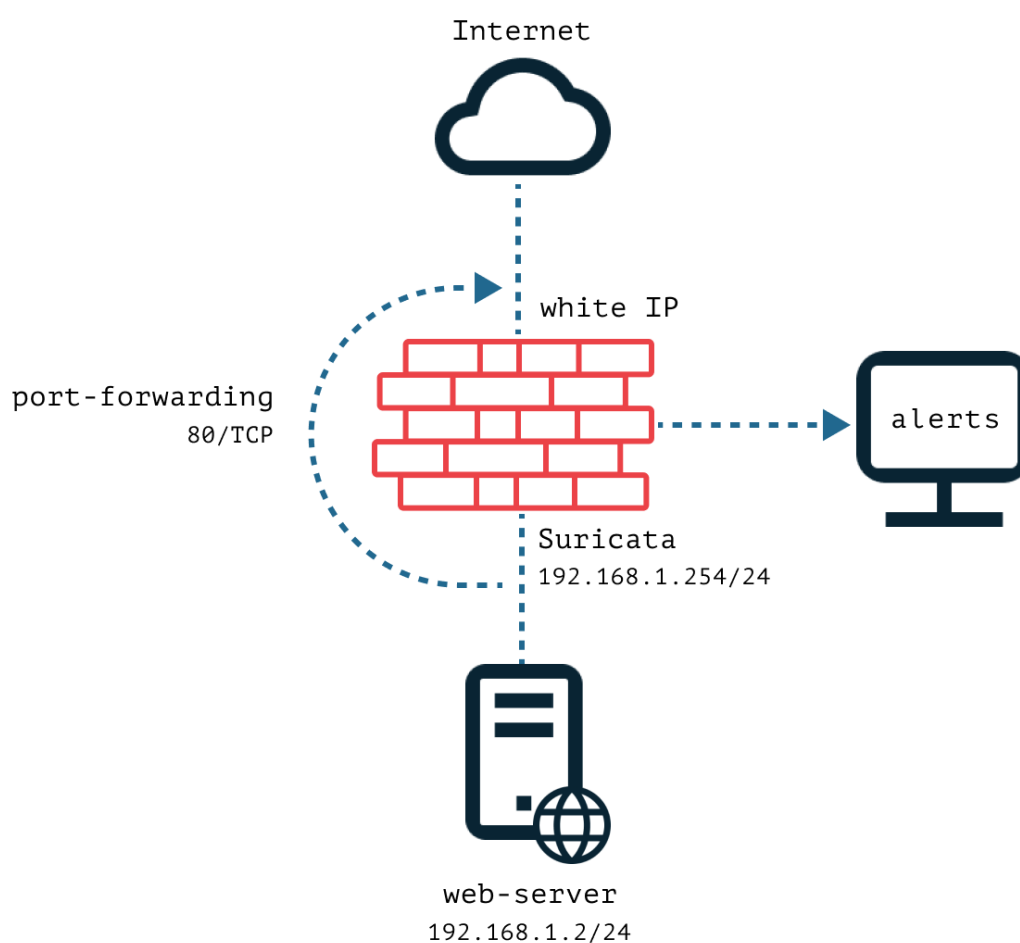
Одним из популярных вариантов для работы с Suricata через веб-интерфейс является использование **Kibana** в сочетании с **Elasticsearch** и **Logstash** (ELK стек). Эта комбинация позволяет собирать, анализировать и визуализировать логи Suricata в режиме реального времени с помощью дашбордов в Kibana.

Другие инструменты для визуализации и управления включают:

- **EveBox**: Это еще одно приложение с веб-интерфейсом, созданное для работы с событиями и алертами, сгенерированными Suricata. EveBox обеспечивает возможность отслеживания алертов, их классификации и создания отчетов.
- **Scirius**: Это еще одна платформа, которая позволяет управлять правилами Suricata, просматривать алерты и логи. Scirius может быть интегрирован с EVE выводом Suricata для анализа данных.

Сетевая схема на базе Suricata

Рассмотрим схемы включения IDPS в «разрыв».



В данной схеме между целевым веб-сервером и интернетом установлена IDPS. То есть маршрутизацию и проброс портов обеспечивает именно хост с IDPS. Таким образом, правильно настроив систему, можно блокировать трафик при срабатывании сигнатур.

Установка Suricata

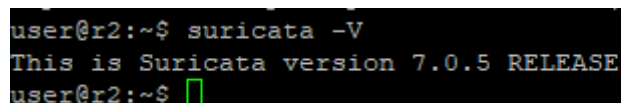
Чтобы установить последнюю стабильную версию, импортируйте репозиторий Open Information Security Foundation (OISF) с сервера Suricata. Для этого выполните следующие команды:

1-й способ.

```
sudo apt install software-properties-common  
sudo add-apt-repository ppa:oisf/suricata-stable
```

После импорта репозитория обновите APT и распакуйте программное обеспечение с помощью этой команды:

```
sudo apt update  
sudo apt install -y suricata
```



```
user@r2:~$ suricata -V  
This is Suricata version 7.0.5 RELEASE  
user@r2:~$
```

2-й способ.

Установить Suricata из Personal Package Archive, PPA. Это специальный репозиторий с open source-проектами разных компаний, в том числе разработчиков Suricata (OISF).

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

Настройка

После установки Suricata важно обновить правила (сигнатуры) и их источники:

```
sudo suricata-update
```

Далее в файле /etc/default/suricata сверим значение параметра IFACE с именем внешнего интерфейса хоста:

```

GNU nano 6.2 /etc/default/suricata
RUN_AS_USER=

# Configuration file to load
SURCONF=/etc/suricata/suricata.yaml

# Listen mode: pcap, nfqueue, custom_nfqueue or af-packet
# depending on this value, only one of the two following options
# will be used (af-packet uses neither).
# Please note that IPS mode is only available when using nfqueue
LISTENMODE=af-packet

# Interface to listen on (for pcap mode)
IFACE=eth0

# Queue number to listen on (for nfqueue mode)
NFQUEUE="-q 0"

# Queue numbers to listen on (for custom_nfqueue mode)
# Multiple queues can be specified
CUSTOM_NFQUEUE="-q 0 -q 1 -q 2 -q 3"

```

Смотрим свои интерфейсы:

```

user@r2:~$ ip a
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN gro
t qlen 1000
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
        valid_lft forever preferred_lft forever
    inet6 ::1/128 scope host
        valid_lft forever preferred_lft forever
2: enp0s3: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel st
oup default qlen 1000
    link/ether 08:00:27:a4:ad:03 brd ff:ff:ff:ff:ff:ff
    inet 10.10.10.113/24 brd 10.10.10.255 scope global dynamic noprefix
000
        valid_lft 35201sec preferred_lft 35201sec
    inet6 fe80::eb2c:3a31:3590:10dd/64 scope link noprefixroute
        valid_lft forever preferred_lft forever
3: enp0s8: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel st
oup default qlen 1000
    link/ether 08:00:27:14:8c:a0 brd ff:ff:ff:ff:ff:ff
    inet 10.1.1.2/24 brd 10.1.1.255 scope global noprefixroute enp0s8
        valid_lft forever preferred_lft forever
    inet6 fe80::a00:27ff:fe14:8ca0/64 scope link
        valid_lft forever preferred_lft forever

```

Меняем настройку:

```

GNU nano 6.2 /etc/default/suricata *
RUN_AS_USER=

# Configuration file to load
SURCONF=/etc/suricata/suricata.yaml

# Listen mode: pcap, nfqueue, custom_nfqueue or af-packet
# depending on this value, only one of the two following options
# will be used (af-packet uses neither).
# Please note that IPS mode is only available when using nfqueue
LISTENMODE=af-packet

# Interface to listen on (for pcap mode)
IFACE=enp0s3

# Queue number to listen on (for nfqueue mode)
NFQUEUE="-q 0"

# Queue numbers to listen on (for custom_nfqueue mode)
# Multiple queues can be specified
CUSTOM_NFQUEUE="-q 0 -q 1 -q 2 -q 3"

^G Справка      ^O Записать     ^W Поиск       ^K Вырезать    ^T Выполнить   ^C Позиция
^X Выход        ^R Читфайл     ^\ Замена     ^U Вставить    ^J Выровнять   ^/ К строке

```

Проверим, чтобы это же значение стояло в файле /etc/suricata/suricata.yaml в блоках pcap, pfring и af-packet.

```

GNU nano 6.2 /etc/suricata/suricata.yaml *
mtu: 1500
rss-hash-functions: auto
mempool-size: 65535
mempool-cache-size: 257
rx-descriptors: 1024
tx-descriptors: 1024
copy-mode: none
copy-iface: none

# Cross platform libpcap capture support
pcap:
- interface: enp0s3
  # On Linux, pcap will try to use mmap'ed capture and will use "buffer-size"
  # as total memory used by the ring. So set this to something bigger
  # than 1% of your bandwidth.
  #buffer-size: 16777216
  #bpf-filter: "tcp and port 25"
  # Choose checksum verification mode for the interface. At the moment
  # of the capture, some packets may have an invalid checksum due to

```

```

GNU nano 6.2 /etc/suricata/suricata.yaml *

##
## Step 3: Configure common capture settings
##
## See "Advanced Capture Options" below for more options, including Netmap
## and PF_RING.
##

# Linux high speed capture support
af-packet:
  - interface: enp0s3
    # Number of receive threads. "auto" uses the number of cores
    #threads: auto
    # Default clusterid. AF_PACKET will load balance packets based on flow.
    cluster-id: 99
    # Default AF_PACKET cluster type. AF_PACKET can load balance per flow or per
    # This is only supported for Linux kernel > 3.1
    # possible value are:
    # * cluster_flow: all packets of a given flow are sent to the same socket

```

```

GNU nano 6.2 /etc/suricata/suricata.yaml *
  #bpf-filter: port 80 or udp
#- interface: eth3
  #threads: auto
  #copy-mode: tap
  #copy-iface: eth2
  # Put default values here
- interface: default

# PF_RING configuration: for use with native PF_RING support
# for more info see http://www.ntop.org/products/pf_ring/
pfring:
  - interface: enp0s3
    # Number of receive threads. If set to 'auto' Suricata will first try
    # to use CPU (core) count and otherwise RSS queue count.
    threads: auto

    # Default clusterid. PF_RING will load balance packets based on flow.
    # All threads/processes that will participate need to have the same
    # clusterid.
    cluster-id: 99

```

Конфигурирование

Основным конфигурационным файлом suricata является `/etc/suricata/suricata.yaml` — откроем его и поправим настройки. В блоке `outputs` включим вывод данных:

```
GNU nano 6.2 /etc/suricata/suricata.yaml *
mode: extra-data
# Two proxy deployments are supported: "reverse" and "forward". In
# a "reverse" deployment the IP address used is the last one, in a
# "forward" deployment the first IP address is used.
deployment: reverse
# Header name where the actual IP address will be reported. If more
# than one IP address is present, the last IP address will be the
# one taken into consideration.
header: X-Forwarded-For

types:
- alert:
  payload: yes # enable dumping payload in Base64
  # payload-buffer-size: 4kb # max size of payload buffer to output i
  # payload-printable: yes # enable dumping payload in printable (1
  # packet: yes # enable dumping of packet (without stre
  # metadata: no # enable inclusion of app layer metadata
  http-body: yes # Requires metadata; enable dumping of HTTP
  # http-body-printable: yes # Requires metadata; enable dumping of H
```

Проверить валидность файла конфигураций можно с помощью команды

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

```
user@r2:~$ sudo suricata -T -c /etc/suricata/suricata.yaml -v
Notice: suricata: This is Suricata version 7.0.5 RELEASE running in SYSTEM mode
Info: cpu: CPUs/cores online: 1
Info: suricata: Running suricata under test mode
Info: suricata: Setting engine mode to IDS mode by default
Info: exception-policy: master exception-policy set to: auto
Info: logopenfile: fast output device (regular) initialized: fast.log
Info: logopenfile: eve-log output device (regular) initialized: eve.json
Info: logopenfile: stats output device (regular) initialized: stats.log
Info: detect: 1 rule files processed. 37238 rules successfully loaded, 0 rules failed, 0
Info: threshold-config: Threshold config parsed: 0 rule(s) found
Info: detect: 37241 signatures processed. 1128 are IP-only rules, 4888 are inspecting packet payload,
31013 inspect application layer, 108 are decoder event only
Notice: suricata: Configuration provided was successfully loaded. Exiting.
user@r2:~$
```

Если открыть лог-файлы, вы увидите обращения к веб-серверу, которые фиксирует Suricata. Это связано с тем, что в большинстве правил этой IDPS-системы указано действие alert.

Чтобы Suricata не просто логировала подозрительный трафик, но и блокировала его, нужно добавить действие drop в сигнатурах, которые находятся по адресу /var/lib/suricata/rules. В этом же каталоге можно создавать файлы со своими правилами.

Логи подозрительных обращений к веб-серверу, которые записала Suricata:

```
$ tail -f /var/log/suricata/http.log
```