Установка Snort в Ubuntu 22.04

Snort, как система обнаружения вторжений (IDS), поддерживает несколько режимов работы, которые определяют его функциональность и способы обработки сетевого трафика. Вот основные режимы работы Snort:

1. **Sniffer Mode (Режим Прослушивания)**:
   - В этом режиме Snort работает как простой сниффер сетевого трафика.
   - Он просто прослушивает сетевой трафик на выбранных сетевых интерфейсах и выводит его на экран или записывает в файлы журналов без обнаружения атак.
   - Этот режим часто используется для анализа сетевого трафика и отладки.
2. **Packet Logging Mode (Режим Журналирования Пакетов)**:
   - В этом режиме Snort принимает сетевой трафик, анализирует его и записывает результаты в файлы журналов.
   - Snort может записывать полный пакетный поток, а также метаданные о пакетах, такие как их источник, назначение, время и другие атрибуты.
   - Этот режим полезен для регистрации сетевого трафика для последующего анализа и обнаружения атак.
3. **Inline Mode (Режим Встроенного Обнаружения)**:
   - В этом режиме Snort работает в качестве активного узла сети и может блокировать трафик на основе обнаруженных атак.
   - Snort анализирует сетевой трафик в реальном времени и может выполнять действия по блокировке, например, отбрасывать пакеты или уведомлять администратора об обнаруженных атаках.
   - Этот режим требует специальной настройки сетевых интерфейсов и может иметь большее влияние на производительность сети.
4. **Network Intrusion Detection System (NIDS) Mode (Режим Системы Обнаружения Вторжений в Сети)**:
   - Этот режим является комбинацией режимов прослушивания и журналирования пакетов.
   - Snort анализирует сетевой трафик для обнаружения атак и одновременно записывает результаты анализа в файлы журналов.

- Этот режим наиболее распространен и предпочтителен для использования в сетевых средах для обнаружения и предотвращения атак.

Режим работы "Система обнаружения вторжений в сети" (NIDS Mode) в Snort является типичным режимом работы, который обычно используется при настройке Snort для обнаружения атак в реальном времени и одновременного записывания результатов анализа в журналы. Чтобы запустить Snort в этом режиме, выполните следующие шаги:

1. **Настройте конфигурационный файл Snort**:
   - Откройте конфигурационный файл Snort (**snort.conf**) в текстовом редакторе, например:
     bashCopy code
   - Убедитесь, что конфигурация настроена на использование режима NIDS, а не в режиме прослушивания или журналирования пакетов.
   - Обычно, параметры для NIDS Mode уже настроены по умолчанию, но убедитесь, что секция **config** содержит правильные параметры для вашей среды и целей.
2. **Укажите интерфейсы для мониторинга**:
   - В конфигурационном файле укажите сетевые интерфейсы, которые вы хотите мониторить на предмет атак.
   - Обычно это делается в секции **var HOME_NET** или аналогичной секции.
3. **Запустите Snort**:
   - Запустите Snort с помощью команды **sudo snort -c /etc/snort/snort.conf -i <interface>**, где **<interface>** - это сетевой интерфейс, который вы хотите мониторить.
4. **Мониторинг событий**:
   - После запуска Snort начнет мониторить выбранные сетевые интерфейсы на предмет атак.
   - Обнаруженные атаки и другие события будут записываться в соответствующие файлы журналов, указанные в конфигурационном файле Snort.

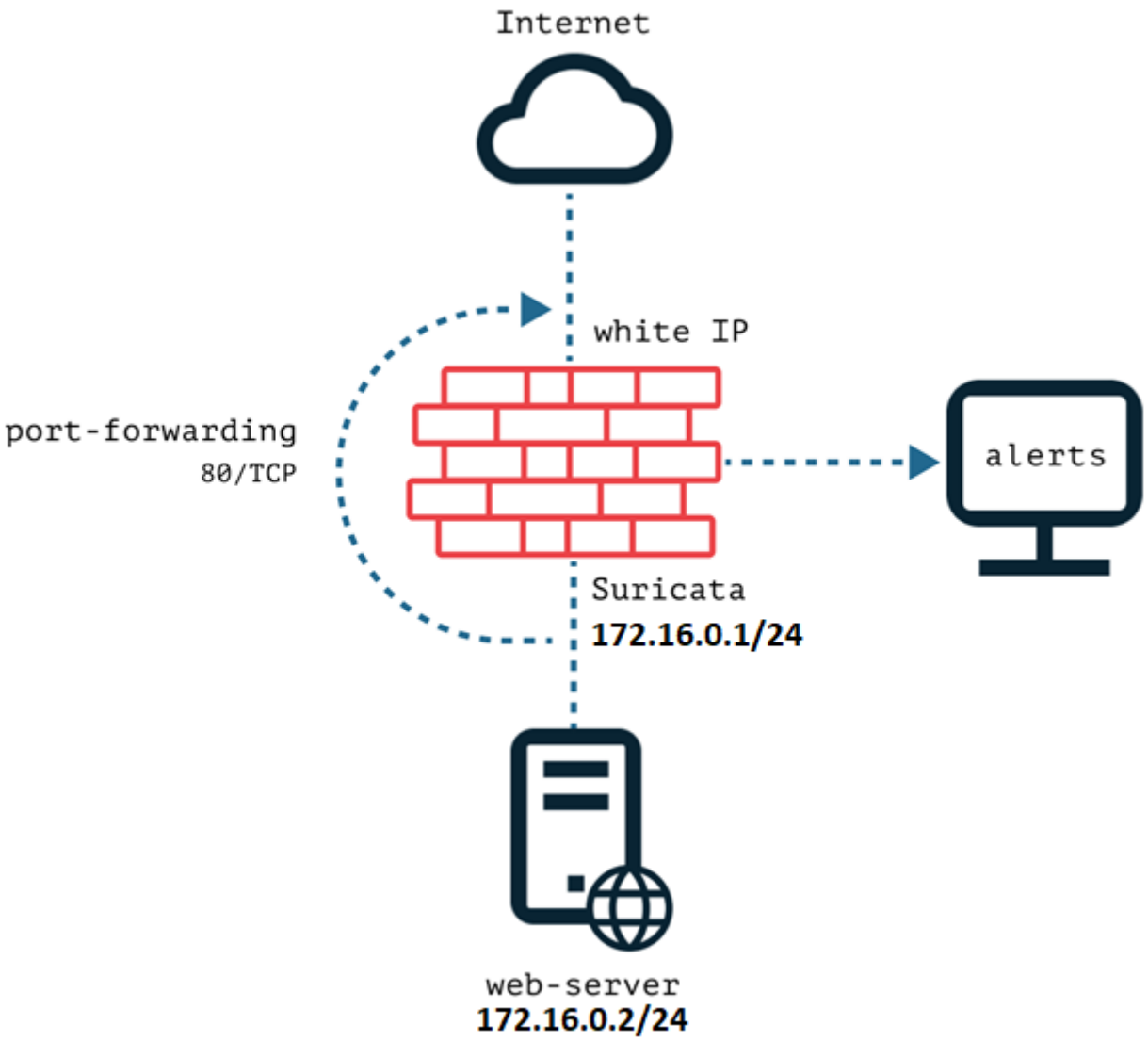
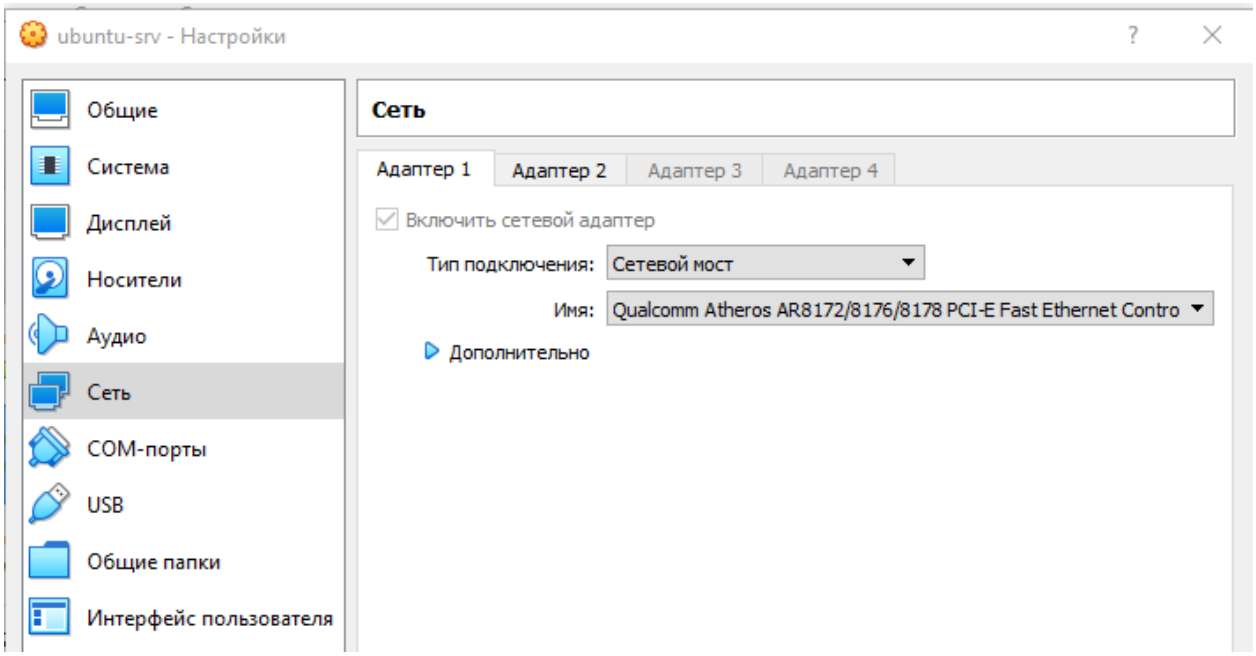
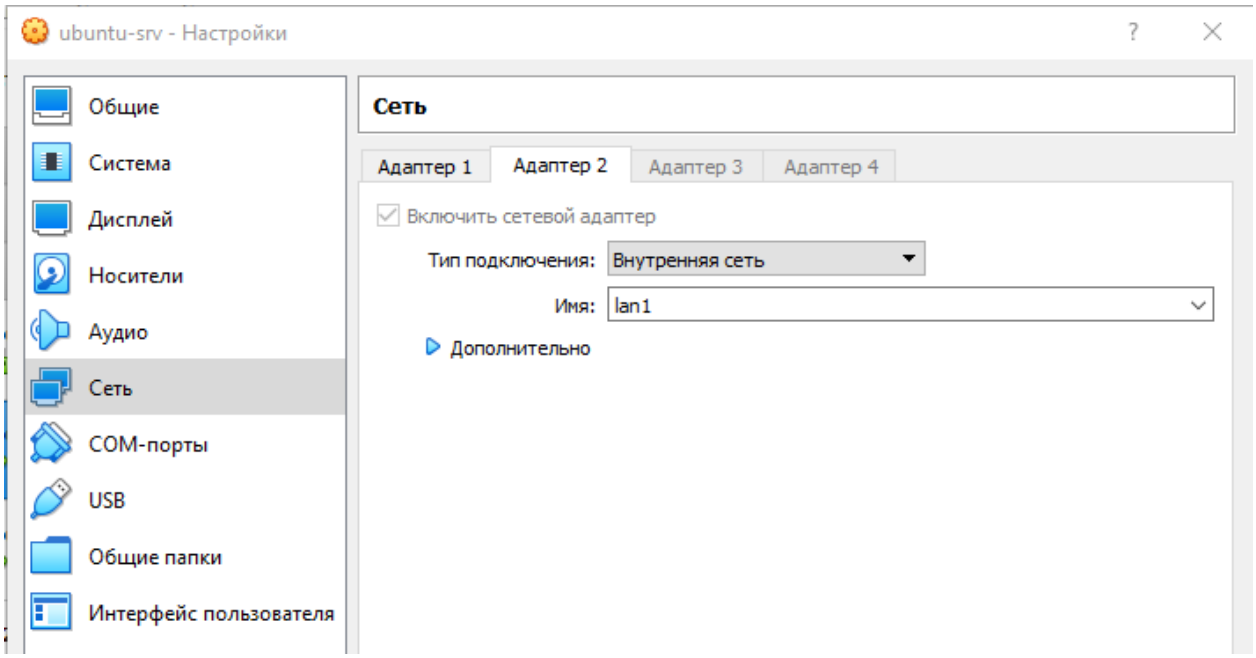


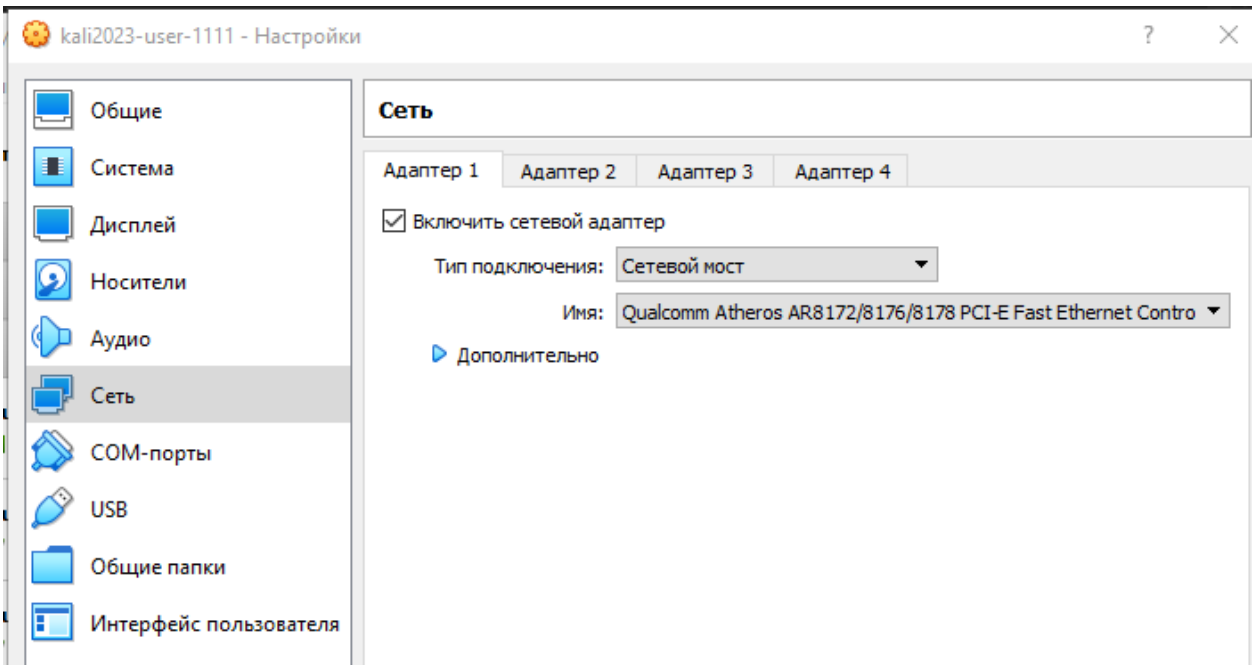
Схема включения Snort
Лабораторный стенд включает три виртуальные машины:
- сервер, через который компьютеры локальной сети выходят в Интернет. Через него же идет трафик на веб-сервер. Здесь же устанавливается система обнаружения вторжений. (Ubuntu Server 22.04)
- веб-сервер. (Ubuntu Server 22.04)
- клиентская машина, рабочее место администратора. (Ubuntu Desktop 22.04)
- компьютер «злоумышленника». (Kali Linux 2023)

Настройки сетевых интерфейсов в VirtualBox:
Сервер (веб-сервер и клиентская машина настраиваются аналогично):


ubuntu-srv - Настройки
Общие
Система
Дисплей
Носители
Аудио
Сеть
COM-порты
USB
Общие папки
Интерфейс пользователя
Сеть
Адаптер 1
Адаптер 2
Адаптер 3
Адаптер 4
Включить сетевой адаптер
Тип подключения:
Сетевой мост
Имя:
Qualcomm Atheros AR8172/8176/8178 PCI-E Fast Ethernet Contro
Дополнительно


ubuntu-srv - Настройки
Общие
Система
Дисплей
Носители
Аудио
Сеть
COM-порты
USB
Общие папки
Интерфейс пользователя
Сеть
Адаптер 1
Адаптер 2
Адаптер 3
Адаптер 4
Включить сетевой адаптер
Тип подключения:
Внутренняя сеть
Имя:
lan1
Дополнительно

Kali Linux:



Настройка сервера.

Настройка сетевых интерфейсов:

```
  GNU nano 6.2                    /etc/netplan/00-installer-config.yaml
# This is the network config written by 'subiquity'
network:
  version: 2
  ethernets:
    enp0s3:
      dhcp4: true
    enp0s8:
      addresses:
        - 172.16.0.1/24
      gateway4: 192.168.1.232
      nameservers:
        addresses: [192.168.1.232, 8.8.8.8]
```

Sudo netplan apply

```
user@serv:~$ sudo netplan apply

** (generate:2375): WARNING **: 12:25:12.292: `gateway4` has been deprecated, use default routes ins
tead.
See the 'Default routes' section of the documentation for more details.
WARNING:root:Cannot call Open vSwitch: ovsdb-server.service is not running.

** (process:2373): WARNING **: 12:25:14.007: `gateway4` has been deprecated, use default routes inst
ead.
See the 'Default routes' section of the documentation for more details.

** (process:2373): WARNING **: 12:25:14.654: `gateway4` has been deprecated, use default routes inst
ead.
See the 'Default routes' section of the documentation for more details.

** (process:2373): WARNING **: 12:25:14.660: `gateway4` has been deprecated, use default routes inst
ead.
See the 'Default routes' section of the documentation for more details.
```

```
user@serv:~$ ip a
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default 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 state UP group default qlen 1
0
    link/ether 08:00:27:97:19:37 brd ff:ff:ff:ff:ff:ff
    inet 192.168.1.236/24 metric 100 brd 192.168.1.255 scope global dynamic enp0s3
       valid_lft 86324sec preferred_lft 86324sec
    inet6 fe80::a00:27ff:fe97:1937/64 scope link
       valid_lft forever preferred_lft forever
3: enp0s8: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP group default qlen 1
0
    link/ether 08:00:27:e0:e9:b2 brd ff:ff:ff:ff:ff:ff
    inet 172.16.0.1/24 brd 172.16.0.255 scope global enp0s8
       valid_lft forever preferred_lft forever
    inet6 fe80::a00:27ff:fee0:e9b2/64 scope link
       valid_lft forever preferred_lft forever
user@serv:~$
```

Теперь настроим NAT, компьютеры из локальной сети будут выходить в Интернет через этот сервер.

Включим IP-маршрутизацию в ядре:

В файле /etc/sysctl.conf нужно раскомментировать строки:

```
  GNU nano 6.2                         /etc/sysctl.conf *
# Turn on Source Address Verification in all interfaces to
# prevent some spoofing attacks
#net.ipv4.conf.default.rp_filter=1
#net.ipv4.conf.all.rp_filter=1

# Uncomment the next line to enable TCP/IP SYN cookies
# See http://lwn.net/Articles/277146/
# Note: This may impact IPv6 TCP sessions too
#net.ipv4.tcp_syncookies=1

# Uncomment the next line to enable packet forwarding for IPv4
net.ipv4.ip_forward=1

# Uncomment the next line to enable packet forwarding for IPv6
#  Enabling this option disables Stateless Address Autoconfiguration
#  based on Router Advertisements for this host
net.ipv6.conf.all.forwarding=1
```

После этого примените изменения с помощью команды:
sudo sysctl -p

```
user@serv:~$ sudo sysctl -p
net.ipv4.ip_forward = 1
net.ipv6.conf.all.forwarding = 1
user@serv:~$
```

Настроим iptables для маршрутизации трафика: Создадим правила iptables, которые будут перенаправлять трафик от локальной сети через наш сервер.

Сетевой интерфейс смотрящий в локальную сеть (enp0s8) имеет IP-адрес 172.16.0.1, а второй интерфейс, смотрящий в интернет (enp0s3) имеет IP-адрес 192.168.1.236.

sudo iptables -A FORWARD -i enp0s8 -o enp0s3 -s 172.16.0.0/24 -j ACCEPT

sudo iptables -A FORWARD -i enp0s3 -o enp0s8 -m state --state RELATED,ESTABLISHED -j ACCEPT

```
user@serv:~$ sudo iptables -A FORWARD -i enp0s8 -o enp0s3 -s 172.16.0.0/24 -j ACCEPT
user@serv:~$ sudo iptables -A FORWARD -i enp0s3 -o enp0s8 -m state --state RELATED,ESTABLISHED -j AC
CEPT
```

Настроим NAT:

sudo iptables -t nat -A POSTROUTING -o enp0s3 -s 192.168.1.0/24 -j MASQUERADE

```
user@serv:~$ sudo iptables -t nat -A POSTROUTING -o enp0s3 -s 172.16.0.0/24  -j MASQUERADE
user@serv:~$ sudo iptables-save > /etc/iptables.rules
user@serv:~$ sudo iptables -L
Chain INPUT (policy ACCEPT)
target     prot opt source               destination

Chain FORWARD (policy ACCEPT)
target     prot opt source               destination
ACCEPT     all  --  172.16.0.0/24        anywhere
ACCEPT     all  --  anywhere             anywhere             state RELATED,ESTABLISHED
ACCEPT     all  --  172.16.0.0/24        anywhere
ACCEPT     all  --  172.16.0.0/24        anywhere
ACCEPT     all  --  anywhere             anywhere             state RELATED,ESTABLISHED

Chain OUTPUT (policy ACCEPT)
target     prot opt source               destination
user@serv:~$
```

```
user@serv:~$ sudo ip route
default via 192.168.1.1 dev enp0s3 proto dhcp src 192.168.1.236 metric 100
172.16.0.0/24 dev enp0s8 proto kernel scope link src 172.16.0.1
192.168.1.0/24 dev enp0s3 proto kernel scope link src 192.168.1.236 metric 100
192.168.1.1 dev enp0s3 proto dhcp scope link src 192.168.1.236 metric 100
user@serv:~$
```

Теперь настроим проброс порта 80 на сервере для адреса 172.16.0.2, с помощью iptables:

sudo iptables -t nat -A PREROUTING -i enp0s3 -p tcp --dport 80 -j DNAT --to-destination 172.16.0.2:80

Сохраним правила:

```
user@serv:~$ sudo iptables-save > /etc/iptables.rules
user@serv:~$ sudo iptables-restore < /etc/iptables.rules
```

Проверяем правило:

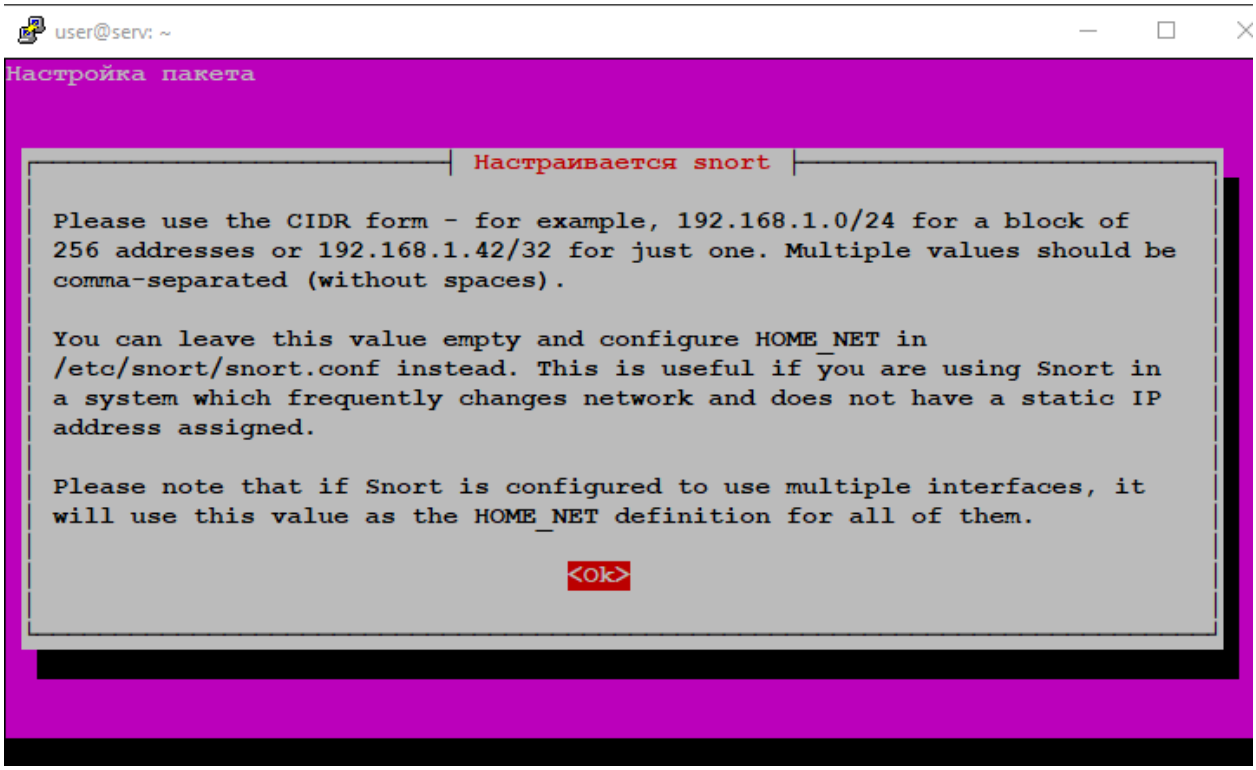
```
user@serv:~$ sudo iptables -t nat -L
Chain PREROUTING (policy ACCEPT)
target     prot opt source               destination
DNAT       tcp  --  anywhere             anywhere             tcp dpt:http to:172.16.0.2:80
DNAT       tcp  --  anywhere             anywhere             tcp dpt:http to:172.16.0.2:80

Chain INPUT (policy ACCEPT)
target     prot opt source               destination

Chain OUTPUT (policy ACCEPT)
target     prot opt source               destination

Chain POSTROUTING (policy ACCEPT)
target     prot opt source               destination
MASQUERADE  all  --  172.16.0.0/24        anywhere
user@serv:~$
```

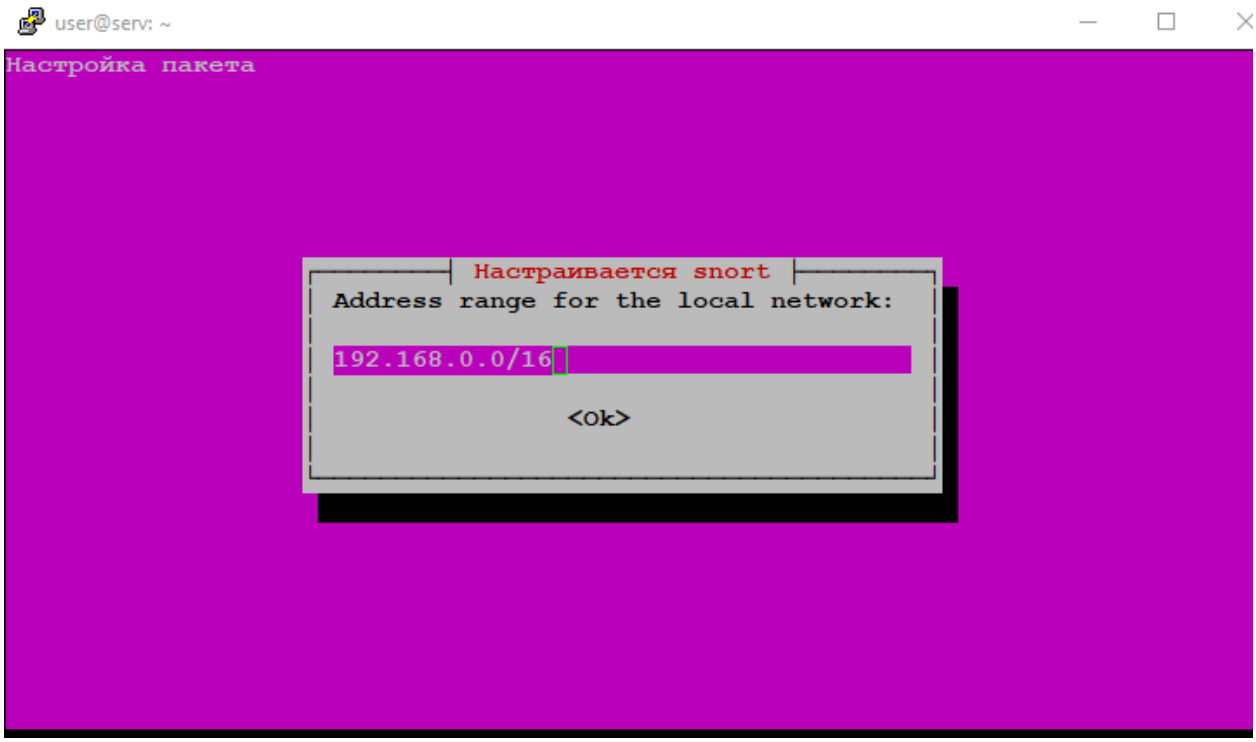
Установка Snort

wget http://archive.ubuntu.com/ubuntu/pool/universe/s/snort/snort_2.9.15.1-6build1_amd64.deb

dpkg -i snort_2.9.15.1-6build1_amd64.deb

sudo apt --fix-broken install






```
user@serv:~$ snort -V

   ,,_     -*> Snort! <*-
  o"  )~   Version 2.9.15.1 GRE (Build 15125)
   ''''    By Martin Roesch & The Snort Team: http://www.snort.org/contact#team
           Copyright (C) 2014-2019 Cisco and/or its affiliates. All rights reserved.
           Copyright (C) 1998-2013 Sourcefire, Inc., et al.
           Using libpcap version 1.10.1 (with TPACKET_V3)
           Using PCRE version: 8.39 2016-06-14
           Using ZLIB version: 1.2.11

user@serv:~$
```

```
user@serv:~$ snort --daq-list
Available DAQ modules:
pcap(v3): readback live multi unpriv
nfq(v7): live inline multi
ipfw(v3): live inline multi unpriv
dump(v3): readback live inline multi unpriv
afpacket(v5): live inline multi unpriv
user@serv:~$
```

```
  GNU nano 6.2                /etc/snort/snort.conf *
# Note to Debian users: this value is overriden when starting
# up the Snort daemon through the init.d script by the
# value of DEBIAN_SNORT_HOME_NET s defined in the
# /etc/snort/snort.debian.conf configuration file
#
ipvar HOME_NET 172.16.0.1

# Set up the external network addresses. Leave as "any" in most si>
ipvar EXTERNAL_NET any
# If HOME_NET is defined as something other than "any", alternativ>
# use this definition if you do not want to detect attacks from yo>
# IP addresses:
#ipvar EXTERNAL_NET !$HOME_NET

# List of DNS servers on your network
ipvar DNS_SERVERS $HOME_NET
```

```
  GNU nano 6.2                                  /etc/snort/snort.conf *
###################################################
# Step #6: Configure output plugins
# For more information, see Snort Manual, Configuring Snort - Output
###################################################

# unified2
# Recommended for most installs
# output unified2: filename merged.log, limit 128, nostamp, mpls_eve
output unified2: filename snort.log, limit 128, nostamp, mpls_event_

# Additional configuration for specific types of installs
# output alert_unified2: filename snort.alert, limit 128, nostamp
output alert_unified2: filename snort.alert, limit 128, nostamp
output log_unified2: filename snort.log, limit 128, nostamp
```

Протестируем конфигурацию перед запуском Snort:

```
sudo snort -T -c /etc/snort/snort.conf
```

```
           Rules Engine: SF_SNORT_DETECTION_ENGINE  Version 3.1  <Build 1>
           Preprocessor Object: SF_SDF  Version 1.1  <Build 1>
           Preprocessor Object: SF_POP  Version 1.0  <Build 1>
           Preprocessor Object: SF_FTPTELNET  Version 1.2  <Build 13>
           Preprocessor Object: appid  Version 1.1  <Build 5>
           Preprocessor Object: SF_DNS  Version 1.1  <Build 4>
           Preprocessor Object: SF_IMAP  Version 1.0  <Build 1>
           Preprocessor Object: SF_SMTP  Version 1.1  <Build 9>
           Preprocessor Object: SF_SSH  Version 1.1  <Build 3>
           Preprocessor Object: SF_DCERPC2  Version 1.0  <Build 3>
           Preprocessor Object: SF_SIP  Version 1.1  <Build 1>
           Preprocessor Object: SF_MODBUS  Version 1.1  <Build 1>
           Preprocessor Object: SF_GTP  Version 1.1  <Build 1>
           Preprocessor Object: SF_SSLPP  Version 1.1  <Build 4>
           Preprocessor Object: SF_REPUTATION  Version 1.1  <Build 1>
           Preprocessor Object: SF_DNP3  Version 1.1  <Build 1>

Snort successfully validated the configuration!
Snort exiting
user@serv:~$
```

Запуск Snort с параметрами, определенными конфигурационным файлом, чтобы он мог мониторить сетевой трафик на интерфейсе enp0s3 и обнаруживать потенциальные атаки и нарушения безопасности в соответствии с настройками, определенными в файле snort.conf:

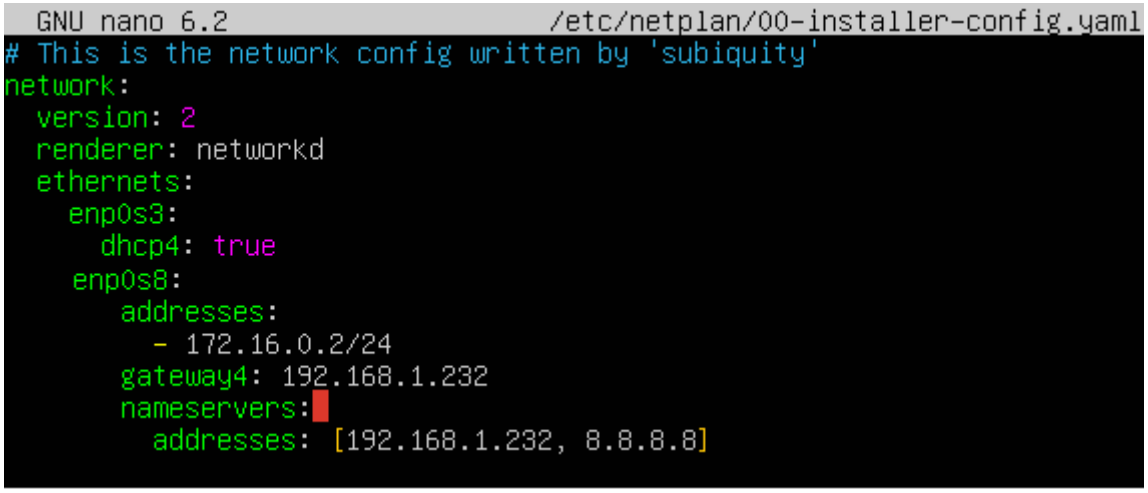
sudo snort -A console -q -u snort -g snort -c /etc/snort/snort.conf -i enp0s3

Настройка веб-сервера.

Проверяем сетевые интерфейсы:

Пропишем интерфейс в конф. файле

nano /etc/netplan/00-installer-config.yaml



Sudo netplan apply

Проверяем:

```
user@web-server:~$ ip a
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group defa
    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 state UP
0
    link/ether 08:00:27:c8:78:e1 brd ff:ff:ff:ff:ff:ff
    inet 192.168.1.250/24 brd 192.168.1.255 scope global enp0s3
       valid_lft forever preferred_lft forever
    inet6 fe80::a00:27ff:fec8:78e1/64 scope link
       valid_lft forever preferred_lft forever
3: enp0s8: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP
0
    link/ether 08:00:27:ef:17:b4 brd ff:ff:ff:ff:ff:ff
    inet 172.16.0.2/24 brd 172.16.0.255 scope global enp0s8
       valid_lft forever preferred_lft forever
    inet6 fe80::a00:27ff:feef:17b4/64 scope link
       valid_lft forever preferred_lft forever
user@web-server:~$ _
```

Теперь отключаем «внешний» интерфейс, трафик должен идти через сервер.

```
user@web-server:~$ sudo ifconfig enp0s3 down
[sudo] password for user:
user@web-server:~$ ip a
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default 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> mtu 1500 qdisc fq_codel state DOWN group default qlen 1000
    link/ether 08:00:27:c8:78:e1 brd ff:ff:ff:ff:ff:ff
3: enp0s8: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP group default qlen 1
0
    link/ether 08:00:27:ef:17:b4 brd ff:ff:ff:ff:ff:ff
    inet 172.16.0.2/24 brd 172.16.0.255 scope global enp0s8
       valid_lft forever preferred_lft forever
    inet6 fe80::a00:27ff:feef:17b4/64 scope link
       valid_lft forever preferred_lft forever
user@web-server:~$ ping 8.8.8.8
PING 8.8.8.8 (8.8.8.8) 56(84) bytes of data.
64 bytes from 8.8.8.8: icmp_seq=1 ttl=57 time=75.9 ms
64 bytes from 8.8.8.8: icmp_seq=2 ttl=57 time=76.0 ms
```

Устанавливаем веб-сервер, например Apache:

sudo apt update

sudo apt install apache2

Запускаем, проверяем:

```
user@web-server:~$ sudo systemctl start apache2
user@web-server:~$ sudo systemctl status apache2
● apache2.service - The Apache HTTP Server
     Loaded: loaded (/lib/systemd/system/apache2.service; enabled; vendor prese>
     Active: active (running) since Wed 2024-04-24 10:59:42 UTC; 9s ago
       Docs: https://httpd.apache.org/docs/2.4/
    Process: 3166 ExecStart=/usr/sbin/apachectl start (code=exited, status=0/SU>
   Main PID: 3177 (apache2)
      Tasks: 55 (limit: 1012)
     Memory: 6.7M
        CPU: 160ms
     CGroup: /system.slice/apache2.service
             ├─3177 /usr/sbin/apache2 -k start
             ├─3178 /usr/sbin/apache2 -k start
             └─3179 /usr/sbin/apache2 -k start

апр 24 10:59:41 web-server systemd[1]: Starting The Apache HTTP Server...
апр 24 10:59:42 web-server apachectl[3176]: AH00558: apache2: Could not reliabl>
апр 24 10:59:42 web-server systemd[1]: Started The Apache HTTP Server.
```

Добавим правила в файрволл:

iptables -A INPUT -i enp0s8 -p tcp --dport 80 -j ACCEPT

iptables -A INPUT -i enp0s8 -p tcp --dport 443 -j ACCEPT

```
user@web-server:~$ sudo iptables -A INPUT -i enp0s8 -p tcp --dport 80 -j ACCEPT
user@web-server:~$ sudo iptables -A INPUT -i enp0s8 -p tcp --dport 443 -j ACCEPT
user@web-server:~$ sudo iptables -L
Chain INPUT (policy ACCEPT)
target     prot opt source               destination
ACCEPT     tcp  --  anywhere             anywhere             tcp dpt:http
ACCEPT     tcp  --  anywhere             anywhere             tcp dpt:https

Chain FORWARD (policy ACCEPT)
target     prot opt source               destination

Chain OUTPUT (policy ACCEPT)
target     prot opt source               destination
user@web-server:~$
```

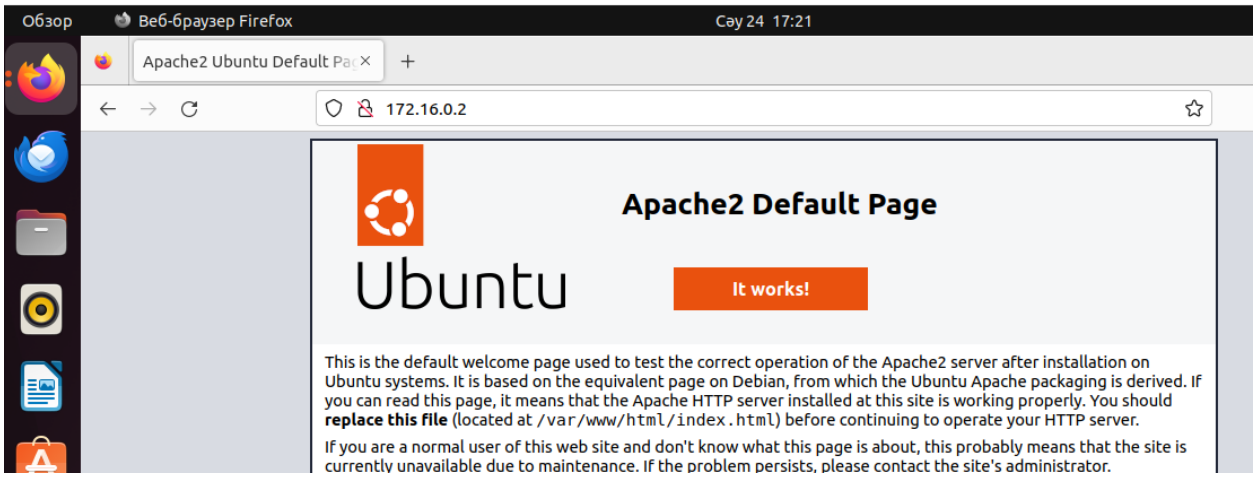
Настройка клиентской машины:

```
  GNU nano 6.2          /etc/netplan/01-network-manager-all.yaml *
# Let NetworkManager manage all devices on this system
network:
  version: 2
  renderer: NetworkManager
  ethernets:
    enp0s8:
      addresses:
        - 172.16.0.3/24
      gateway4: 172.16.0.1
      nameservers:
        addresses: [172.16.0.1, 8.8.8.8]
```

```
user@r2:~$ ip a
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group defaul
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 state UP gr
oup default qlen 1000
    link/ether 08:00:27:f7:c8:85 brd ff:ff:ff:ff:ff:ff
    inet 192.168.1.91/24 brd 192.168.1.255 scope global dynamic noprefixroute en
p0s3
       valid_lft 86395sec preferred_lft 86395sec
    inet6 fe80::c9e4:1923:8302:2904/64 scope link noprefixroute
       valid_lft forever preferred_lft forever
3: enp0s8: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP gr
oup default qlen 1000
    link/ether 08:00:27:be:47:4b brd ff:ff:ff:ff:ff:ff
    inet 172.16.0.3/24 brd 172.16.0.255 scope global noprefixroute enp0s8
       valid_lft forever preferred_lft forever
    inet6 fe80::a00:27ff:febe:474b/64 scope link
```

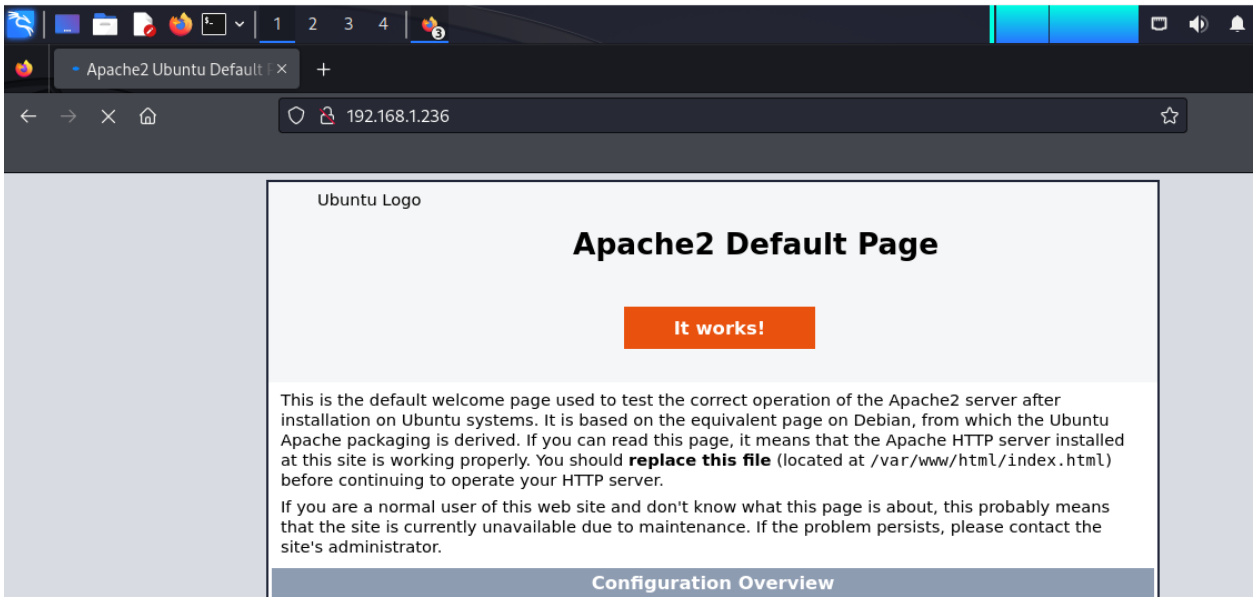
```
user@r2:~$ ping 172.16.0.2
PING 172.16.0.2 (172.16.0.2) 56(84) bytes of data.
64 bytes from 172.16.0.2: icmp_seq=1 ttl=64 time=5.57 ms
64 bytes from 172.16.0.2: icmp_seq=2 ttl=64 time=2.03 ms
64 bytes from 172.16.0.2: icmp_seq=3 ttl=64 time=2.38 ms
^C
```

Убеждаемся, что веб-сервер работает:



Теперь запустим машину «хакера»

Проверим, что веб-сервер («жертва» будущей атаки) доступен из внешней сети по внешнему адресу сервера:



Все готово для проведения атаки.

hping3 -S -p 22 192.168.1.236

Чтобы видеть предупреждения в консоли, запустите Snort с флагом -D:

sudo snort -D -c /etc/snort/snort.conf

запустить Snort в фоновом режиме и записывать вывод в журнал:

sudo snort -q -u snort -g snort -c /etc/snort/snort.conf -i enp0s3 > /var/log/snort.log 2>&1 &

Для администрирования Snort, часто используются утилиты с графическим интерфейсом, которые предоставляют удобный способ настройки и мониторинга Snort. Они позволяют администраторам просматривать события обнаружения, управлять правилами и сигнатурами, а также анализировать сетевой трафик в более удобном и интуитивно понятном интерфейсе.

Некоторые примеры утилит с графическим интерфейсом для администрирования Snort:

- **BASE (Basic Analysis and Security Engine)**: Веб-интерфейс для анализа событий обнаружения Snort.

- **Snorby**:Еще один веб-интерфейс для анализа событий Snort.
- **Security Onion**: Комплексное решение для мониторинга сетевой безопасности, включающее Snort и другие инструменты, предоставляющее удобный интерфейс для анализа и визуализации данных безопасности.