Лабораторная работа №1

Тема: Введение в IDS/IPS и архитектуру Suricata.

Теория

- **IDS (Intrusion Detection System):** система обнаружения вторжений, только детектирует атаки (логирует события).
- **IPS (Intrusion Prevention System):** система предотвращения вторжений — не только детектирует, но и блокирует опасный трафик, работая “inline” в потоке пакетов.
- **Suricata:** высокопроизводительный анализатор трафика и IPS/IDS со своим языком правил, может работать как в режиме мониторинга (IDS), так и в режиме блокировки (IPS, через NFQUEUE).

Шаги установки Suricata как IPS на Ubuntu 24.04

1. **Обновление системы**

```
apt update && apt upgrade -y
```

2. **Установка зависимостей**
   Необходимые утилиты и библиотеки для работы Suricata и iptables.

```
apt install -y software-properties-common curl jq iptables-persistent
libnetfilter-queue1 libnfnetlink0
```

3. **Настройка ядра для фильтрации мостов Docker**
   Включаем фильтрацию iptables для bridge-сетей Docker.

```
modprobe br_netfilter
echo "br_netfilter" > /etc/modules-load.d/br_netfilter.conf

cat > /etc/sysctl.d/99-bridge-nf.conf <<EOF
net.bridge.bridge-nf-call-iptables=1
net.bridge.bridge-nf-call-ip6tables=1
```

```
EOF
sysctl -p /etc/sysctl.d/99-bridge-nf.conf
```

4. **Установка Suricata из официального репозитория**

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

5. **Создание и добавление базовых правил**
   Пример: блокировка всего ICMP (IPS) и логирование HTTP (IDS).

```
mkdir -p /etc/suricata/rules

cat > /etc/suricata/rules/local.rules <<'EOF'
# Блокируем все ICMP пакеты — базовый IPS-тест
drop icmp any any -> any any (msg:"[IPS] BLOCK ICMP"; sid:1000007; rev:1;)

# Логируем HTTP-запросы — базовый IDS-тест
alert http any any -> any any (msg:"[IDS] HTTP Request Detected";
sid:100002; rev:1; flow:to_server; classtype:policy-violation;)
EOF
```

6. **Минимальная настройка suricata.yaml**
   Предварительно сделайте бэкап оригинального suricata.yaml. Стартовый пример конфигурационного файла:

```
cat > /etc/suricata/suricata.yaml <<'EOF'
%YAML 1.1
---
runmodes:
  - runmode: workers

# NFQUEUE для IPS
nfqueue:
  mode: accept
  repeat-mark: 1
  repeat-mask: 1
  bypass-mark: 2
  bypass-mask: 2
  queue-balance:
    - 0-3
```

```
# Правила
rule-files:
  - /etc/suricata/rules/local.rules

# EVE — основной лог
outputs:
  - eve-log:
      enabled: yes
      filetype: regular
      filename: eve.json
      types:
        - alert
        - drop
        - http
        - dns

# Отключаем лишнее для простоты
classification-file: /etc/suricata/classification.config
reference-config-file: /etc/suricata/reference.config

app-layer:
  protocols:
    http: enabled
    dns: enabled
    tls: enabled

# Логирование
logging:
  outputs:
    - console:
        enabled: no
    - file:
        enabled: yes
        level: info
        filename: suricata.log
EOF
```

7. **Настройка iptables для перехвата Docker-трафика**
   Направьте трафик в очередь NFQUEUE (Suricata может его анализировать и при необходимости блокировать):

```
iptables -D DOCKER-USER -j NFQUEUE --queue-num 1 --queue-bypass 2>/dev/null
|| true
iptables -I DOCKER-USER -j NFQUEUE --queue-num 1 --queue-bypass
netfilter-persistent save
```

8. **Получение необходимых прав и настройка systemd**
   Дайте Suricata права на работу с сетевым движком:

```
setcap cap_net_admin,cap_net_raw+ep /usr/bin/suricata

mkdir -p /etc/systemd/system/suricata.service.d

cat > /etc/systemd/system/suricata.service.d/override.conf <<'EOF'
[Service]
ExecStart=
ExecStart=/usr/bin/suricata -c /etc/suricata/suricata.yaml -q 1 --pidfile
/run/suricata.pid
EOF

systemctl daemon-reload
systemctl enable suricata
systemctl restart suricata
systemctl status suricata --no-pager
```

9. **Логи и правила**
   Основные логи Suricata:
   - `/var/log/suricata/eve.json` (основные события и алерты)
   - `/var/log/suricata/suricata.log` (служебный лог)

Задание

1. **Разработайте docker-compose для стенда**
   - Контейнеры:
     - **attacker:** образ kali-linux (kalilinux/kali-rolling:latest)
     - **victim:** образ alpine:latest
   - Обе машины в одной docker-сети.
   - Откройте shell в обеих для тестирования трафика.
   - Не забудьте установить нужные утилиты (`ping` , curl и пр.) внутри kali/alpine, а так же запустить простой http-сервер на жертве.
   - ICMP на victim должен быть блокирован Suricata (пакеты не проходят).
2. **Проверьте работу Suricata**
   - От имени attacker проверьте:

```
ping victim
curl http://victim
tail -f /var/log/suricata/eve.json | jq '.'
```

```
- В логе Suricata (`eve.json`) должны появиться события drop для ICMP и
отдельные записи для HTTP-запросов.
- В victim пакеты ICMP не должны проходить (нет ответа на ping).
```

3. **Задание со звездочкой** - запустите suricata в единой сети docker в режиме IDS. Рассмотрите вопрос, какие ограничения существуют на запуск Suricata в режиме IPS в среде docker.
