**Цель**

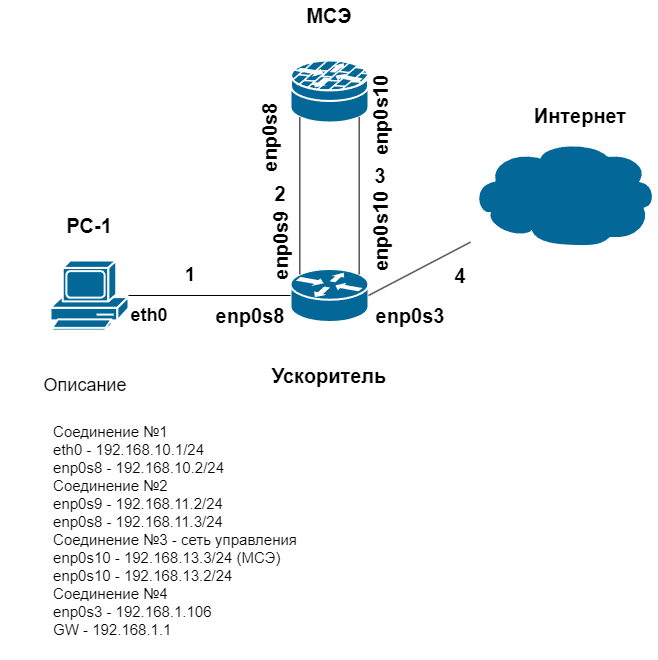
Проверка гипотезы о возможности реализации подсистемы ускорения обработки сетевых пакетов в виде отдельного устройства.

**Гипотеза**

Обработка сетевых пакетов МСЭ может быть ускорена, если пакеты сессий, для которых МСЭ принял решение (ACCEPT/DENY) обрабатывать при помощи дополнительного устройства (Далее Ускоритель). Ускоритель планируется к реализации как отдельное устройство на ПЛИС или отдельном

процессоре. Реализация отдельного системотехнического устройства — долгий и затратный проект, поэтому предварительно необходимо проверить адекватность гипотезы на стенде.

**Схема стенда**

****

**Описание для инженера.**

Для проверки работоспособности гипотезы был собран стенд из трёх виртуальных машин на VirtualBox: клиент, МСЭ и ускоритель. На клиенте и МСЭ установлена ОС Debian bookworm 12, а на ускорителе – ОС Ubuntu 22.04. Соединения 1, 2, 3 – внутренняя сеть, соединение 4 – сетевой мост.

На МСЭ и ускорители установлены следующие пакеты, которые необходимы для работы стенда:

* iptables
* build-essential python3-dev libnetfilter-queue-dev

Также необходимы библиотеки для python:

* scapy
* NetfilterQueue

На ускорители необходимо создать правила в соответствующем порядке с помощью утилиты iptables. На стенде версия iptables - v1.8.9

1. iptables –t mangle –F PREROUTING (очистка таблицы mangle цепочки PREROUTING)
2. iptables –t mangle –A PREROUTING –i enp0s8 –m conntrack --ctstate NEW –j TEE --gateway 192.168.11.3
3. iptables –t mangle –A PREROUTING –i enp0s8 –m conntrack --ctstate ESTABLISHED, RELATED –j ACCEPT
4. iptables –t mangle –A PREROUTING –i enp0s10 –j NFQUEUE --queue-num 1
5. iptables –t mangle –A PREROUTING –i enp0s8 –m conntrack --ctstate NEW –j NFQUEUE --queue-num 1

На ускорители нужно выполнить следующую команду от суперпользователя:

* echo 1 >> /proc/sys/net/ipv4/ip\_forward

Также на ускорители необходимо запустить соответствующий скрипт на python. Чтобы запустить скрипт необходимо выполнить следующие команды:

* chmod +x nfqsniff.py
* sudo ./nfqsniff.py

Листинг nfqsniff.py

#!/usr/bin/python3

**from** **netfilterqueue** **import** NetfilterQueue

**from** **scapy.all** **import** \*

pkt\_list=[]

**def** **print\_and\_accept**(pkt):

scapy\_pkt=IP(pkt.get\_payload())

**if** scapy\_pkt.src != "192.168.13.3" **and** scapy\_pkt.dst != "192.168.13.2":

pkt\_list.append(pkt)

**return**

**if** scapy\_pkt.haslayer(ICMP) == True **and** scapy\_pkt.src=="192.168.13.3" **and** scapy\_pkt.dst=="192.168.13.2":

**if** scapy\_pkt.getlayer(ICMP).load == b'DROP':

pkt.drop()

**for** i **in** range (len(pkt\_list)):

pkt\_list[i].drop()

pkt\_list.clear()

**return**

**if** scapy\_pkt.getlayer(ICMP).load == b'ACCEPT':

pkt.accept()

**for** i **in** range(len(pkt\_list)):

pkt\_list[i].accept()

pkt\_list.clear()

**return**

nfqueue = NetfilterQueue()

nfqueue.bind(**1**, print\_and\_accept)

**try**:

nfqueue.run()

**except** **KeyboardInterrupt**:

**print**('')

nfqueue.unbind()

На МСЭ необходимо выполнить следующие команды с помощью утилиты iptables для установки правил.

1. iptables –t mangle –F PREROUTING (очистка таблицы mangle цепочки PREROUTING)
2. iptables –t mangle –N DROP\_YANDEX (создание цепочки DROP\_YANDEX)
3. iptables –t mangle –A PREROUTING –i enp0s8 –d 77.88.44.242 --jump DROP\_YANDEX
4. iptables –t mangle –A PREROUTING –i enp0s8 –j NFQUEUE --queue-num 1

Также на МСЭ необходимо запустить соответствующий скрипт на python. Чтобы запустить скрипт необходимо выполнить следующие команды:

* chmod +x sniff.py
* sudo ./sniff.py

Листинг sniff.py

#!/usr/bin/python3

**from** **netfilterqueue** **import** NetfilterQueue

**from** **scapy.all** **import** \*

**def** **print\_and\_accept**(pkt):

packet = IP(pkt.get\_payload())

**if** packet.haslayer(DNS) == True:

**if** packet.getlayer(UDP).qd.qname == b'ya.ru.':

send(IP(src="192.168.13.3", dst="192.168.13.2")/ICMP()/"DROP")

pkt.drop()

**else**:

send(IP(src="192.168.13.3", dst="192.168.13.2")/ICMP()/"ACCEPT")

pkt.drop()

**elif** packet.haslayer(ICMP) == True:

send(IP(src="192.168.13.3", dst="192.168.13.2")/ICMP()/"DROP")

pkt.drop()

**else**:

send(IP(src="192.168.13.3", dst="192.168.13.2")/ICMP()/"ACCEPT")

pkt.drop()

nfqueue = NetfilterQueue()

nfqueue.bind(**1**, print\_and\_accept)

**try**:

nfqueue.run()

**except** **KeyboardInterrupt**:

**print**('')

nfqueue.unbind()

Оба скрипта будут лежать на GitHub: https://github.com/vas-vas777/udv-fw

Также надо будет добавить маршрут на Роутере (на схеме облако - интернет). Например, такой командой:

* ip r add 192.168.10.0/24 via 192.168.1.106

После настройки правил и запуска скриптов проверить работоспособность стенда.

Пример работы стенда.

МСЭ запрещает проход icmp от клиента, разрешает доступ к google, запрещает к ya.ru.

1. Запустить команду “ping –c 1 192.168.1.1” с клиента
2. Запустить команду curl –I https://ya.ru с клиента
3. Запустить команду curl –I https://google.com с клиента
4. Проверить таблицу правил межсетевого экрана на ускорители с помощью команды iptables –t mangle –L –v
5. Проверить таблицу правил на МСЭ с помощью команды iptables –t mangle –L –v

**Описание для программиста**

На клиенте и МСЭ установлена ОС Debian 12 bookworm, а на ускорителе – ОС Ubuntu 22.04 LTS.

На МСЭ и ускорителе должна быть установлена утилита iptables. На стенде версия iptables - v1.8.9.

Для осуществления поставленной «цели» необходимо разработать модули для МСЭ и ускорителя, которые будут получать пакеты из kernelspace в userspace. Для этого необходимо создать соответствующие правила в ускорители и МСЭ, с помощью утилиты iptables с действием NFQUEUE. Это действие позволяет передавать пакеты из kernelspace в userspace и складывать их в очередь. Модуль будет брать эти пакеты из очереди в userspace, обрабатывать и принимать решение о пропуске их или нет. Данные модули будут работать по «сети управления».

Модуль для ускорителя будет получать пакет от клиента (копия этого пакета будет уходить на МСЭ). Исходный пакет от клиента (оригинал) будет попадать в модуль и сохраняться до тех пор, пока не придет подтверждение о его пропуске (или нет). После обработки копии МСЭ отправляет ускорителю пакет о принятом решении (ICMP пакет с данными “ACCEPT” или “DROP”). Если решение положительно (ICMP “ACCEPT”), то модуль ускорителя принимает решение о пропуске оригинального пакета клиента, иначе оригинал пакета от клиента уничтожится. Остальные пакеты сессии уходят через ускоритель минуя модуль. Чтобы пакеты сессии проходили через ускоритель, необходимо включить маршрутизацию транзитных IP-пакетов (ip\_forward).

Модуль для МСЭ будет получать копию пакета клиента от ускорителя. Далее в модуле реализуется некоторый набор условий, на основе которого принимается решение об отправки пакета ICMP с данными «ACCEPT» или «DROP» через интерфейс управления. Пакет ICMP с данными «ACCEPT» означает, что копия пакета удовлетворяет правилам МСЭ, иначе посылается пакет ICMP с данными «DROP».

Как идут пакеты.

1. Пакеты обрабатываются в цепочке PREROUTING таблице mangle.
2. На ускорителе есть первое по списку правило «iptables –t mangle –A PREROUTING –i enp0s8 –m conntrack --ctstate NEW –j TEE --gateway 192.168.11.3». Оно отправляет копию пакета (благодоря опции TEE) на МСЭ, а сам пакет (оригинал) идет дальше по цепочке. Данное правило срабатывает только для новых соединений.
3. Далее пакет из п.2 (не копия) идет по цепочке ко второму правилу, так как он новый он в неё не попадает. Он попадет в это правило «iptables –t mangle –A PREROUTING –i enp0s8 –m conntrack --ctstate NEW –j NFQUEUE --queue-num 1». В данной цепочке пакет из kernelspace попадает в userspace (в модуль для ускорителя), в очередь №1.
4. После принятия решения о пропуске пакетов от клиента, МСЭ отправляет icmp пакет на ускоритель через интерфейс управления. Он попадает под воздействие правила «iptables –t mangle –A PREROUTING –i enp0s10 –j NFQUEUE --queue-num 1». В данной цепочке пакет из kernelspace попадает в userspace (в модуль для ускорителя), в очередь №1.
5. Если всё хорошо, то пакеты от клиента, связанные с данным соединением, будут попадать под правило «iptables –t mangle –A PREROUTING –i enp0s8 –m conntrack --ctstate ESTABLISHED, RELATED –j ACCEPT» и уходить к цели, иначе первоначальный пакет (NEW) уничтожится ещё в модуле ускорителя.
6. Копии пакета клиента, попадающие в МСЭ от ускорителя, сразу попадают в модуль. После их обработки МСЭ отправляет ICMP с данными «ACCEPT» или «DROP» через интерфейс управления.
7. В модуле для МСЭ копии пакета клиента будут исследоваться, но не изменяться.