Firewall

Исходя из общей теории брандмауэров, можно сделать вывод: для enterprise решений каждый уровень модели TCP/IP должен быть защищен. Однако статичные правила работают неэффективно. Должны быть межуровневые взаимодействия, позволяющие передавать например информацию о подозрительной активности на уровне приложений, на уровень IP фильтрации и блокировать данную активность на уровне IP. Т е enterprise решение должно объединять в себе данные всех уровней в одну картину. Кроме этого, желательно объединение аппаратных и программных решений, т е данные о необходимости блокировки передаются между физическими устройствами.

Политики блокировки

Адреса - источники, которые нужно блокировать на внешнем интерфейсе

- Ваш IP адрес, lo
- IP адреса вашей локальной сети
- Частные адреса класса A, B и C
- Мультикастовые класса D и зарезервированные класса E
- Также можно блокировать Carrier-grade NAT сети (100.64.0.0-100.127.255.255) и TEST-NET (192.0.2.0-192.0.2.255)
- Есть стратегия блокировки ненадежных пулов (наверное, где-то есть список)
- Можно добавить блокировку бродкаста, но это редко используется

Возможно ограничение количества пакетов от источника, порты удаленного источника (>1023), блокировка по флагу статуса TCP соединения (флаг должен быть SYN, не ACK).

Нужно контролировать попытки сканирования портов (могут быть разные источники, но скоординированное сканирование).

netfilter - встроенный firewall. Консольные утилиты для управления netfilter:

- iptables - для v4 и v6 различные
- ebtables - mac briging правила
- arptables - правил трансляции arp

ufw - фронтенд для iptables

nftables - иная реализация

firewalld - реализация в RedHat

Какой firewall сейчас активен

Могут быть установлен и iptables, и nftables. Если вывод следующей команды не пустой, значит убедимся, что модуль nft активен:

```
sudo nft list ruleset
```

Проверяем модуль ядра:

```
lsmod | grep nf_tables
```

Если вывод типа

```
nf_tables             376832  114 nft_compat,nft_chain_nat
```

значит nftables загружен.

Модуль iptables еще присутствует в ядре, но как говорят - для вида.

```
sudo lsmod | grep ip_tables
ip_tables              32768  0
x_tables               65536  8
xt_conntrack,nft_compat,xt_tcpudp,xt_addrtype,xt_nat,xt_set,ip_tables,xt_MASQUERADE
```

Есть промежуточный пакет iptables-nft, который транслирует правила из iptables, Но работает именно nft. Однако редактировать nft таблицы с комментариями

```
# Warning: table ip filter is managed by iptables-nft, do not touch!
```

не стоит. Проблема вот в чем: если не удален/отключен iptables-nft, то после каждого использования команды iptables данные таблицы будут перезаписаны. Поэтому либо не использовать эту команду, включить сервис nft и все, либо создать другие таблицы.

В Alt Linux, nft по умолчанию не установлен.

Iptables

Существующие сессии ssh не блокируются.

Состоит из 4 таблиц:

- Filter table Базовая защита, обычно используется
- NAT table
- Mangle table Изменение сетевых пакетов при их прохождении через брандмауэр
- Security table Используется в системах с установленным SELinux

Каждая таблица состоит из цепочек правил. Filter table состоит из цепочек INPUT, FORWARD, OUTPUT

Просмотр существующих таблиц

```
sudo iptables -L <имя цепочки или ничего> --line-numbers -v
```

Сохранение правил после перезагрузки сервера

```
sudo apt install iptables-persistent
```

Затем сохраняем

```
sudo netfilter-persistent save
```

После этого правила сохраняются в /etc/iptables*

Структура команды

```
iptables <option> <chain> <matching criteria> <target>
```

Пример:

```
sudo iptables -A INPUT -m conntrack --ctstate ESTABLISHED,RELATED -j ACCEPT
```

Аргумент	Описание
Option параметры	
-A, -I	Добавить в конец или в определенное место цепочки. -A INPUT -I INPUT 1
-R	заменить цепочку
-P	Задает дефолтную политику для цепочки. `iptables -P FORWARD DROP`
-D цепочка номер	Удалить запись `sudo iptables -D DOCKER-USER 1`
-N	Создать пользовательскую цепочку правил
-F, -X	Удалить цепочки / пользовательские цепочки
-E	Переименовать пользовательскую цепочку
Matching criteria параметры	

Аргумент	Описание
-m	Модуль.
--ctstate	Состояние пакета. Перечисляется через запятую без пробелов.
-p	Тип соединения, tcp/udp/icmp
--dport	порт назначения --dport ssh с опцией -m можно перечислять через запятую без пробелов `-m multiport --dport 2049,1080,3128` -m должна идти строго после -p ... через : диапазон портов, без multiport
--port	Определяет и входящий, и исходящий порты
-i -o	Интерфейс (входящий/исходящий) -i lo
-s -d	IP источника / приемника
	Кроме этого, еще есть фильтры по: • Текущее состояние соединения • Список портов (поддерживаемых многопортовым модулем) • MAC-адрес источника аппаратного Ethernet или физического устройства • Тип адреса, тип пакета канального уровня или диапазон IP-адресов • Различные части пакетов IPSec или политика IPSec • Тип протокола ICMP • Длина пакета • Время получения пакета • Каждый n-й пакет или случайные пакеты • Идентификатор пользователя, группы, процесса или группы процессов отправителя пакета • Поле типа обслуживания (TOS) заголовка IP (возможно, заданное в таблице управления) • Раздел TTL заголовка IP • Поле iptables mark (устанавливается в таблице управления) • Соответствие пакетов с ограниченной скоростью
Target параметры	

Аргумент	Описание
-j	Действие при соблюдении условия ACCEPT разрешить пакет DROP тихо откинуть пакет REJECT вернуть ответ о блокировании запрашивающей стороне RETURN вернуть в родительскую цепочку. Используется в случае родительских / дочерних цепочек. Еще есть очередь QUEUE Может быть переход к пользовательской цепочке, с возвратом обратно
-g	переход к другой цепочке без возврата обратно
-c	Инициализация счетчиков

В OUTPUT цепочке есть возможность фильтровать по пользователю, группе, процессу, заголовкам безопасности - полный цикл управления.

Также возможна настройка блокировки по времени. Также можно фильтровать каждый N пакет например для логирования. И это все на уровне ядра!

Разрешение доступа для ssh

```
sudo iptables -A INPUT -p tcp --dport ssh -j ACCEPT
```

Здесь для определения порта по названию использовался файл /etc/services То есть возможно прописать свое название службы в данный файл и использовать его в правилах.

```
cat /etc/services | grep http
# Updated from https://www.iana.org/assignments/service-names-port-numbers/service-names-port-
numbers.xhtml .
http            80/tcp          www             # WorldWideWeb HTTP
https           443/tcp                         # http protocol over TLS/SSL
http-alt        8080/tcp        webcache        # WWW caching service
http-serg       8081/tcp                        # test server
```

Затем правилом разрешаем доступ по имени службы. Однако не понятно, почему нужно определять тип соединения. Похоже, что из файла берется только номер порта.

```
sudo iptables -I INPUT 1 -p tcp --dport http-serg -j ACCEPT
```

Разрешение доступа к lo

```
sudo iptables -I INPUT 1 -i lo -j ACCEPT
```

Логирование

Лог всех пакетов (и входящие, и исходящие), на порту 8090.

```
sudo iptables -A INPUT -p tcp --dport 8090 -j LOG --log-prefix "IPTABLES-8090-IN: "
sudo iptables -A INPUT -p udp --dport 8090 -j LOG --log-prefix "IPTABLES-8090-IN: "
sudo iptables -A OUTPUT -p tcp --sport 8090 -j LOG --log-prefix "IPTABLES-8090-OUT: "
sudo iptables -A OUTPUT -p udp --sport 8090 -j LOG --log-prefix "IPTABLES-8090-OUT: "
```

Лог только новых соединений:

```
sudo iptables -A INPUT -p tcp --dport 8090 -m conntrack --ctstate NEW -j LOG --log-prefix
"NEW-8090-CONN: "
```

Просмотр лога в реальном времени:

```
sudo tail -f /var/log/syslog | grep "IPTABLES-8090"
# или
sudo tail -f /var/log/kern.log | grep "IPTABLES-8090"
```

Блокировка по географическому признаку

Работает на уровне ядра. Проверка установлен ли модуль.

```
lsmod | grep xt_geoip

sudo apt update
sudo apt install xtables-addons-common xtables-addons-source build-essential dkms
sudo modprobe xt_geoip
```

Загрузка геобазы.

Вариант 1 (не доделал)

Страница создания аккаунта Из России пока что нельзя создать новую учетку. Надеюсь скоро все изменится. Создали учетку, Управление аккаунтом - Управление ключами лицензирования. Создаем новый ключ, скачиваем конфиг на всякий случай.

Создаем директорию и скачиваем текущую базу

```
mkdir -p /usr/share/xt_geoip
cd /usr/share/xt_geoip

wget "https://download.maxmind.com/app/geoip_download?edition_id=GeoLite2-Country-CSV&license_key=ВАШ_КЛЮЧ&suffix=zip" -O GeoLite2-Country-CSV.zip

unzip GeoLite2-Country-CSV.zip -d geolite
```

Теперь нужно преобразовать базу в требуемый формат. Но перед этим в ubuntu возникло несколько проблем. Одна с размещением скрипта.

```
dpkg -L xtables-addons-common | grep xt_geoip
/usr/bin/xt_geoip_query
/usr/lib/x86_64-linux-gnu/xtables/libxt_geoip.so
/usr/libexec/xtables-addons/xt_geoip_build
/usr/libexec/xtables-addons/xt_geoip_build_maxmind
/usr/libexec/xtables-addons/xt_geoip_dl
/usr/libexec/xtables-addons/xt_geoip_dl_maxmind
/usr/share/man/man1/xt_geoip_build.1.gz
/usr/share/man/man1/xt_geoip_build_maxmind.1.gz
/usr/share/man/man1/xt_geoip_dl.1.gz
/usr/share/man/man1/xt_geoip_dl_maxmind.1.gz
/usr/share/man/man1/xt_geoip_query.1.gz
```

Далее нужно было преобразовать в формат при помощи python скрипта, поскольку Ubuntu изменил формат для базы. Или -

Вариант 2

На [странице сервиса] находим ссылку на текущую версию, скачиваем

```
wget https://download.db-ip.com/free/dbip-country-lite-2025-08.csv.gz -O dbip-country-lite.csv.gz
gunzip dbip-country-lite.csv.gz
```

Преобразовываем базу данных (поиск размещения скрипта в Варианте 1)

```
sudo /usr/libexec/xtables-addons/xt_geoip_build -D /usr/share/xt_geoip dbip-country-lite.csv
```

Пример разрешения / блокировки по геопризнаку

```
sudo iptables -A INPUT -m geoip --src-cc RU -j ACCEPT
sudo iptables -A INPUT -m geoip --src-cc CN,IR,KP -j DROP
sudo iptables -A INPUT -p tcp --dport https -m geoip --src-cc RU -j ACCEPT
```

Примеры настройки

Пример настройки iptables

Политика:

- Разрешить доступ к службам http, https на докере для всех
- Разрешить все исходящие соединения
- Разрешить доступ для ip1
- Docker должен функционировать корректно
- Все остальное запретить

```
sudo iptables -A INPUT -m conntrack --ctstate ESTABLISHED,RELATED -j ACCEPT
sudo iptables -A INPUT -i lo -j ACCEPT
sudo iptables -A INPUT -s 1.1.1.1 -j ACCEPT
sudo iptables -I INPUT 1 -p icmp --icmp-type echo-request -j ACCEPT

sudo iptables -A INPUT -i docker0 -j ACCEPT
sudo iptables -P INPUT DROP

sudo iptables -A DOCKER-USER -m conntrack --ctstate ESTABLISHED,RELATED -j ACCEPT
sudo iptables -A DOCKER-USER -s 1.1.1.1 -p tcp --dport 5432 -j ACCEPT
sudo iptables -A DOCKER-USER -p tcp --dport 5432 -j DROP
```

Revision #34
Created 16 August 2025 14:09:06 by Admin
Updated 17 September 2025 18:03:50 by Admin