Тема № 7. Netfilter

Пакетният филтър в netfilter представлява рамка (framework), позволяващ достъп до мрежовите пакети извън стандартния socket интерфейс на unix/linux операционните системи.

iptables представлява потребителски инструмент за дефиниране на правила за филтриране на мрежвои пакети и транслиране на мрежови адреси.

На практика, често под името iptables се разбира цялата инфраструктура на netfilter & iptables, заедно със следенето на връзките и транслирането на мрежовите адреси.

iptables групира правилата за обработка на мрежовите пакети в таблици по функции (филтриране на пакети, транслиране на мрежови адреси, други модификации на пакетите), всяка от които има вериги (chains, поредици) от правила за обработване на пакетите. Правилата се състоят от условия за съвпадене (matches), които се използват за определяне на това към кое правило да бъдат насочени пакетите и цели (targets), които определят какво ще се прави с удовлетворилите условието пакети. Целта се прилага върРху всеки пакет, който удовлетворява условието на правилото.

iptables IPtables работи в мрежовия слой (layer 3) на OSI модела. За каналния слой (layer 2) се използват други технологии за филтриране – например ebtables.

Всеки мрежови пакет преминава поне през една верига и се сравнява последователно с правилата от нея. Ако се получи съвпадения, обхождането се спира и се прилага съответното правило. Ако пакетът не удовлетвори нито едно условие, тогава върху него се прилага подразбиращата се политика на веригата.

Всяко правило дефинира условие и цел. Целта се прилага върху всеки пакет, който удовлетворява условието на правилото.

Целта на някое правило може да бъде нова верига. Ако пакетът премине през новата верига без да удовлетвори някое условие от нея, се продължава с обхождането на първата верига.

Всяка верига представлява подреден списък с правила.

Веригите се групират в таблици, като всяка таблица е свързана с различен вид обработка на мрежовите пакети. Няма ограничение за влагането на веригите една в друга.

КОНЦЕПЦИЯ НА IPTABLES

точки на скачване (въздействие) – hook points

iptables дефинира пет точки на скачване (въздействие) по пътя на обработването на мрежовите пакети от ядрото на linux:

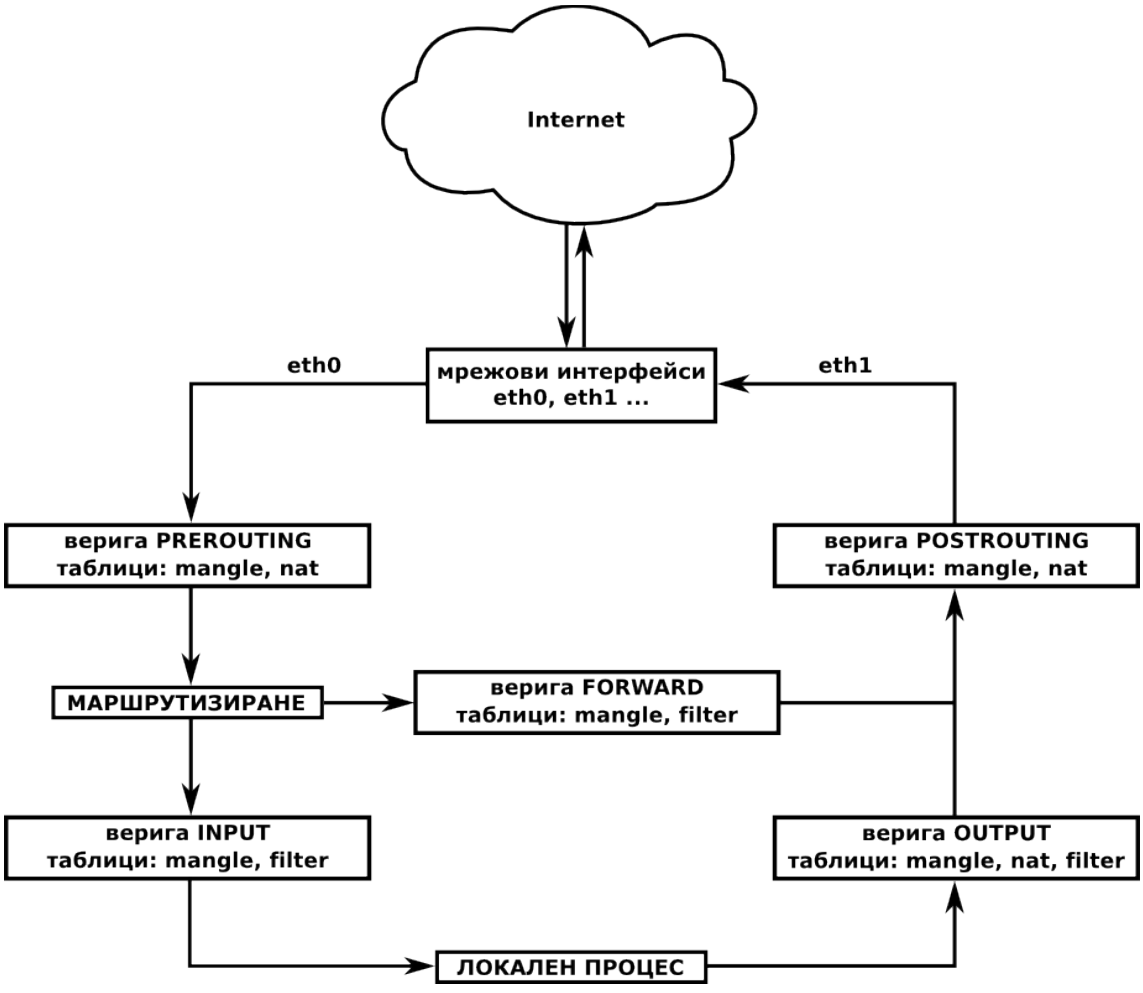
1. **PREROUTING**

2. **INPUT**
3. **FORWARD**
4. **OUTPUT**
5. **POSTROUTING**

Всяка от тези точки дава възможност за следене или оказване на въздействие върху потока от пакети. Към тези точки се прикачват вградените вериги (от правила), но към тях може да се добавят поредици и от допълнителни правила.

Обикновено се казва, че *„веригата PREROUTING от таблицата nat“*, което всъщност не е точно така. Веригите и таблиците са свързани само частично помежду си и не принадлежат една на друга. Веригите представляват точки на скачване (въздействие) в потока от данни, а таблиците представляват типа обработка, която може да се извърши в съответната точка на скачване.

Фигура 1 ни показва всички позволени комбинации и реда, по който те се преминават от мрежовите пакети, минаващи през системата.



Фигура 1.

В iptables са дефинирани следните точки на скачване (hook points):

1. **FORWARD** – обработва пакетите, преминаващи през възел, използван за шлюз (gateway), пристигащи на един мрежов интерфейс и излизащи веднага през друг;
2. **INPUT** – обработва мрежовите пакети точно преди да бъдат доставени на локалния процес, за който са предназначени;
3. **OUTPUT** – обработва мрежовите пакети веднага след генерирането им от някой локален процес;
4. **POSTROUTING** – обработва мрежовите пакети точно преди да излязат през някой мрежови интерфейс;
5. **PREROUTING** – обработва всички мрежови пакети веднага след пристигането им през някой от мрежовите интерфейси (след изхвърлянето на всякакви пакети в резултат на това, че интерфейса работи в нефилтриращ режим и след проверка на контролните суми).

Изборът на верига се основава на това, в коя част от жизнения цикъл на пакетите искаме да приложим своите правила. Филтрирането на изходящите пакети се прави във веригата OUTPUT, тъй като веригата POSTROUTING не е асоциирана с таблицата за филтриране filter.

Политиките по подразбиране на веригите са:

- iptables -P INPUT DROP
- iptables -P FORWARD DROP
- iptables -P OUTPUT ACCEPT

Повечето системи контролират входящия си трафик, а не изходящия.

НАЧИН НА ДЕЙСТВИЕ

1. получаване на мрежовия пакет – пакетът се изследва да се определи дали е предназначен за този възел;
2. ако пакетът е за този възел, той се обработва локално;
3. ако пакетът не е предназначен за този възел и е активирано ip прехвърлянето (в /etc/sysctl.conf), се извършва търсене за подходящ маршрут в маршрутизиращата таблица и пакета се прехвърля на съответния мрежов интерфейс. Ако не бъде намерен такъв маршрут, пакетът се отхвърля;
4. пакетите, генерирани от локалните процеси се изпращат за

маршрутизиране за да бъдат предадени на съответния мрежови интерфейс.

Изходящият пакет се изследва за да се определи дали съществува валиден маршрут, по който да поеме. Ако няма, той се изхвърля (игнорира се напълно) или се отхвърля (игнорира се след изпращане на ICMP съобщение, обясняващо, че няма маршрут до търсения възел;

5. пакетът се предава.

По този начин, iptables може да осъществява филтриране на пакетите, пристигащи във възела, да филтрира тези, които ще бъдат препредадени от възела, и да филтрира пакетите, които са готови за предаване.

ТАБЛИЦИ В IPTABLES

iptables има три вградени таблици: filter, nat и mangle. Всяка от тях е предварително конфигурирана с вериги, отговарящи на една или повече точки на скачване.

Таблица **filter** – използва се за задаване на политиките за типа на трафика, на който е позволено да влиза, да минава транзитно през и да излиза от възела. Ако не сте изразили изрично предпочитание към някоя от другите таблици, iptables по подразбиране ще работи върху веригите на тази таблица. Нейните вградени вериги са **FORWARD, INPUT и OUTPUT**.

Таблица **mangle** – използва се за специални промени по пакетите, като например сваляне на IP опции. Нейните вградени вериги са **FORWARD, INPUT, OUTPUT, POSTROUTING и PREROUTING**.

Таблица **nat** – използва се заедно със следенето на връзките за да пренасочва връзките за NAT, базирайки се обикновено на адресите на източника и получателя. Нейните вградени вериги са **OUTPUT, POSTRPUTING и PREROUTING**.

ВЕРИГИ

По подразбиране, всяка таблица притежава вериги (първоначално празни) за някои или за всички точки на скачване. Освен това, има възможност за добавяне на собствени вериги с цел организиране на собствени правила.

Политиката на веригите се използва за определяне на съдбата на пакетите, които стигат до нейния край без да удовлетворят някое условие и по този начин изпратени към определена цел. Само вградените цели ACCEPT и DROP могат да се използват като политики за вградените вериги, като по подразбиране е използва ACCEPT. Всички дефинирани от потребителя вериги имат заложена политика RETURN, която не може да бъде променяна.

Ако искате да имате по-сложна цел на политиката за някоя вградена верига или политика, която да е различна от RETURN за потребителска верига, на края на веригата може да добавите правило, което да съвпада с всички пакети. Това правило вече може да има произволна цел.

ПРАВИЛА

Правилата в iptables се състоят от един или повече критерии за съвпадение, определящи кои мрежови пакети ще бъдат засегнати от правилото и цел, определяща как точно ще бъдат засегнати тези пакети. За да съвпадне някое правило с даден пакет, трябва да бъдат удовлетворени всички опции за съвпадение.

Iptables поддържа броячи на пакетите и байтовете за всяко правило. Всеки път, когато даден пакет достигне до някое правило и удовлетвори неговите критерии, броячът на пакетите се увеличава с едно, а брояча на байтовете се увеличава с размера на пакета.

Критериите за съвпадение и за целта на правилата не са задължителни. Ако няма критерий за съвпадение, се приема, че всички пакети съвпадат. Ако няма цел, с пакетите не се извършват никакви действия, но пакетите се обработват все едно, че правилото не съществува, но броячите на пакетите и байтовете ще бъдат обновени. Подобно празно правило е:

```
# iptabbles -t filter -A FORWARD
```

СЪВПАДЕНИЯ (matches)

Има голямо разнобразие от съвпадения, които могат да бъдат използвани с iptables. Някои от тях са достъпни само при ядра с активирани определени модули, други са приложими за всички IP пакети (например тип протокол или адрес на източник или получател). В допълнение към общодостъпните съвпадения, iptables предлага и множество специализирани съвпадения, които са достъпни чрез динамично зареждани разширения (с опцията -m или –match). Има едно разширение, което е предназначено да работи с каналния слой (data-link layer) – mac. То търси съвпадение на базата на ethernet media access controller (MAC) адресите.

ЦЕЛИ (TARGETS)

Целите се използват за задаване на действията, които трябва да се предприемат, когато някой пакет удовлетвори критериите от някое правило. С тях се задават и политиките на веригите. В iptables има четири вградени цели, а чрез разширителните модули може да се въведат и други цели. Вградените цели са:

ACCEPT – позволи на пакета да премине към следващия етап от обработката, спри обхождането на текущата верига и премини към следващия етап от фигура 1;

DROP – прекъсни напълно обработката на пакета и не го проверявай повече с никакви други правила, таблици и вериги. Ако искате да предоставите някаква обратна връзка на подателя, тогава използвайте разширението за цели REJECT;

QUEUE – изпрати пакета в потребителското пространство (user space);

RETURN – от правило в някоя потребителска верига: прекъсни обработката на тази верига и възобнови обхождането от извикващата верига от правилото, следващо това правило, което е имало настоящата верига като цел от правило във вградена верига: прекъсни обработката на пакета и приложи нейната политика спрямо него.

За да може ядрото на linux да поддържа използването на защитни стени, то трябва да бъде конфигурирано по подходящия начин:

 # make menuconfig

в секцията Network options → [*] Network packet filtering (replaces ipchains) се конфигурира IP: Netfilter Configuration.

За да може да се използва командата iptables, трябва модула netfilter да бъде зареден в ядрото на системата, което се извършва по следния начин:

```
# modprobe ip_tables
```

ИЗПОЛЗВАНЕ НА iptables

За да се улесни конфигурирането на iptables, има две таблици с правила, наречени filter и nat. Таблицата filter се приема за дадена по подразбиране, ако не бъде предефинирана с опцията -t. Има и вградени 5 вериги – **OUTPUT, INPUT, FORWARD, PREROUTING и POSTROUTING**.

Общият синтаксис на командата iptables е следния:

```
# iptables опция правило разширение
```

Повечето от опциите на командата iptables могат да бъдат групирани в подкоманди и критерии за съвпадение с правила. Някои от опциите на iptables са описани малко по-надолу:

1. ***-c пакети байтове или --set counters*** – когато бъде комбинирана с подкомандите -A, -I или -R, задава стойността *пакети* на брояча на пакетите и стойността *байтове* на брояча на байтове за новото или променено правило;

2. ***--exact или -x*** – показва точният брой на пакетите и байтовете, а не подразбиращия се съкратен формат с метрични представки (K, M или G);

3. ***-h или –help*** – показва помощна информация за използването на

iptables;

4. -***j target [options] или --jump target [options]*** – определя какво да се прави с пакетите, удовлетворяващи критериите на това правило. Target може да бъде името на някоя потребителска верига, някоя от вградените цели или разширение на iptables (в който случай може да има още допълнителни настройки);

5. ***--line numbers*** – когато е в комбинация с -L, показва номерата на правилата във всяка верига, така че да може да се обръщаме към тях с индекси, когато вмъкваме (-I) или изтриваме (-D) правило от някоя верига. Тези номера се променят след изтриване или добавяне на правила от веригата.

6. ***-m съвпадение или --match*** – извикване на разширено съвпадение (евентуално с допълнителни опции към него);

7. ***-M cmd или --modprobe=command*** – използва се за зареждане на нов модул на iptables при промяна на правила;

8. ***-n или --numeric*** – показва адресите и портовете в числов вид, вместо да търси имената на домейните за IP адресите и имената на услугите за номерата на портовете. Тази възможност е полезна, когато DNS услугата в мрежата работи бавно или въобще я няма.

9. ***-t table или --table table*** – изпълнява дадената команда върху съответната таблица. Ако тази опция не е указана, се използва таблицата filter.

10. ***-v или --verbose*** – показва по-подробно резултатите.

Защитната стена (firewall) е решение, което първо забранява всичко, а после вече разрешават само „бели“ списъци (“добрия” трафик). Т.е. първо при нейната инсталация се забраняват всички съединения между защитаваната и отворената мрежи, а след това се добавят специфични правила, които позволяват на определен трафик да преминава през защитната стена.

Съществува и противоположен подход: разрешава се целия трафик, като с „черен“ списък се задава забранения трафик. Защитните стени са просто системи, основани на правила, които разрешават или забраняват преминаването на пакетите през тях.

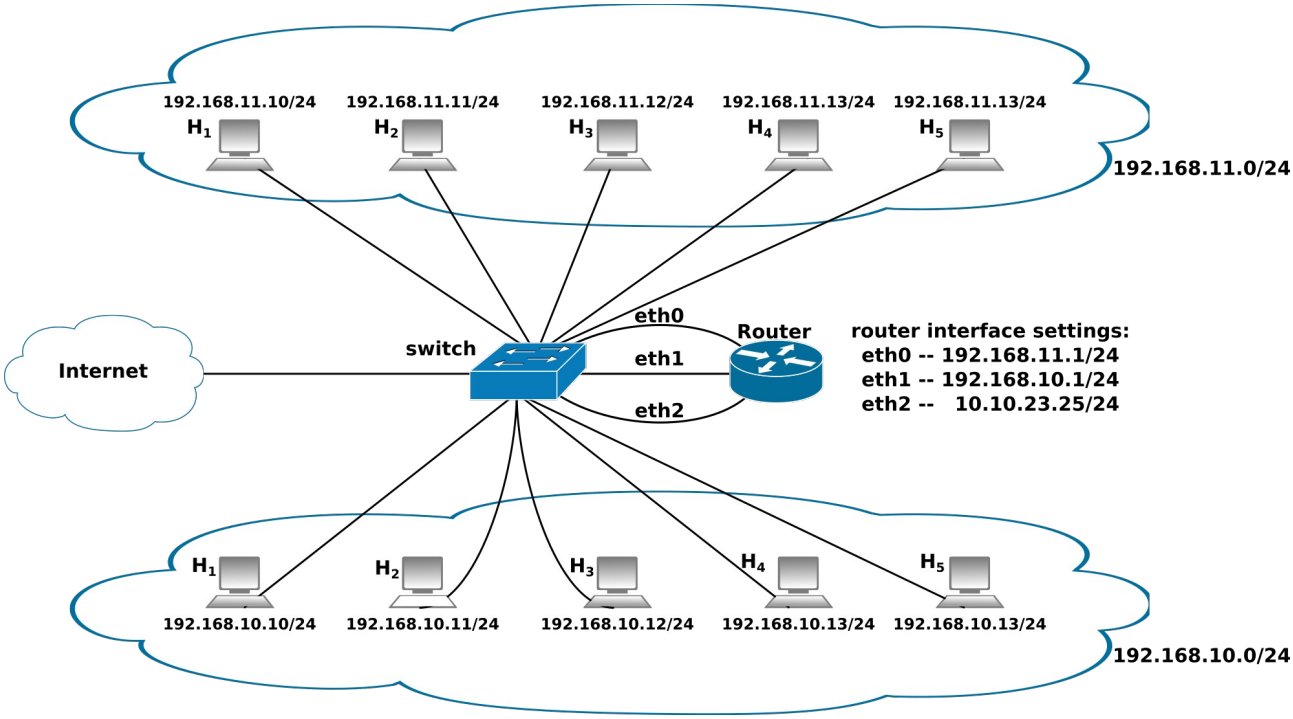
Понеже във всеки пакет се намират няколко заглавия на различни протоколи, проверяват се само тези от тях, които са важни за филтрацията на пакетите. Като правило следва да се отхвърлят пакети на всички протоколи, които не се използват в мрежата или позволяват промяна в настройките й.

Например с ICMP пакети може да се промени маршрутната таблица или да се съобщи за недостъпност на определен възел. ICMP пакетите

намират множество приложения. Те се използват за разузнаване на мрежата. Една от функциите на мрежовия филтър е в предотвратяване на получаването от външни хора на всякакви сведения за възлите в мрежата. Затова следва да се блокират съобщенията от следните типове:

- Входящи съобщения echo request и изходящи echo reply. Така ping може от вътре навън да се ползва, но отвън няма да позволи сканиране.
- Входящи съобщения redirect. Те позволяват промяна на маршрутната таблица.
- Изходящи съобщения destination unreachable и входящи service unavailable. Злосторник няма да може да определи използваните услуги и възлите ще са му недостъпни.

ПОСТАНОВКА НА ЗАДАЧАТА



Фиг. 1 Топология на примерната мрежа

Да се реализира топологията, показана на фигура 1. Компютрите в залата се разделят на две групи, които се конфигурират в две различни локални мрежи – 192.168.10.0/24 и 192.168.11.0/24. Един възел се конфигурира като маршрутизатор със следните настройки: eth0 – 192.168.10.1/24, eth1 – 192.168.11.1/24 и eth2 – 10.10.23.10/24.

конфигуриране на отделните възли:

```
ip addr add 192.168.11.xxx/24 dev eth0
```

```
ip addr add 192.168.10.xxx/24 dev eth0
```

конфигуриране на маршрутите на отделните възли:

```
ip route add 192.168.11.0/24 dev eth0
```

```
ip route add 192.168.10.0/24 dev etho
```

конфигуриране на маршрут по подразбиране:

```
ip route add default via 192.168.11.1
```

```
ip route add default via 192.168.10.1
```

конфигуриране на маршрутизатора:

```
ip addr add 192.168.11.1/24 dev eth0
```

```
ip addr add 192.168.10.1/24 dev eth1
```

```
ip addr add 10.10.23.xxx/24 dev eth0
```

```
ip route add 192.168.11.0/24 dev eth0
```

```
ip route add 192.168.10.0/24 dev eth1
```

```
ip route add 10.10.23.0/24 dev eth0 src 10.10.23.xxx
```

```
ip route add default via 10.10.23.254 dev eth2
```

Проверяваме дали параметъра (ip_forward) в ядрото за препращане на пакети между двата интерфейса е включен:

```
cat /proc/sys/net/ipv4/ip_forward
```

Ако отговорът на тази команда е 1, значи ip_forward е активен и ядрото ще препраща пакети от един мрежови интерфейс към друг. Ако отговорът е 0, значи параметъра не е активен. За да го направим активен, ще използваме следната команда:

```
echo 1 > /proc/sys/net/ipv4/ip_forward
```

след което правим повторна проверка на съдържанието на този параметър с командата cat, посочена малко по-нагоре.

Преди да се активира ip_forward (или ако е активиран, след неговото деактивиране) се проследява с командата ping достъпността до възли от собствената мрежа и до възли от другата мрежа:

```
ping -c10 192.168.11.xxx
```

```
ping -c10 192.168.10.xxx
```

При непозволен ip_forward на маршрутизатора, би трябвало да има отговор на ping само от възли от собствената мрежа. При позволен ip_forward възлите от двете логически мрежи вече би трябвало да се

виждат един друг.

Трябва да се конфигурират и правила в iptables, които да маскират пакетите, изпращани от локалните мрежи към internet:

```
# iptables -t nat -A POSTROUTING -o $EXT_IFACE -j MASQUERADE (?!)
```

ПРИМЕРИ:

Правило за филтрирне на пакети

Тази команда може да се използва за отделяне на целия не-HTTP трафик

нека eth0 е ethernet интерфейсът към локалната мрежа, а eth1 – е ethernet интерфейса към Internet.

 # iptables -t filter -P FORWARD DROP

 # iptables -t filter -A FORWARD -i eth0 -p tcp --dport 80 -j ACCEPT

 # iptables -t filter -A FORWARD -i eth1 -p tcp --sport 80 -j ACCEPT

Първата команда определя политиката по подразбиране на веригата FORWARD от таблицата filter да изхвърля (DROP) всички пакети. Втората команда разрешава всички изходящи HTTP заявки, а третата – всички входящи заявки.

Правило за превод на мрежов адрес:

 # iptables -t nat -A PREROUTING -i eth1 -p tcp -dport 80 -j DNAT –to-destination 12.34.56.78:8080

Правило, разрешаващо преминаване на трафик по 80-ти порт по TCP протокол:

 # iptables -A FORWARD –m state --state NEW –p tcp --dst $HTTP_IP --dport 80 –j ACCEPT

state – проверка на признак за състоянието от netfilter. Това са new (пакетът отваря ново съединение или принадлежи на еднопосочен поток), established (пакетът принадлежи на вече установено съединение, през което пакетите вървят в двете посоки), related (пакетът принадлежи на вече съществуващо съединение, но при това отваря ново съединение) и invalid (пакетът е свързан с неизвестен протокол или съединение и вероятно съдържа грешка в данните или заглавието).

Разрешаваме пакетите, принадлежащи към вече установено съединение/ разрешаване на всеки изходящ трафик:

iptables –A FORWARD –m state --state ESTABLISHED, RELATED –j ACCEPT

Използването на състояния позволява построяване на мощна и ефективна защита.

Признакът NEW не е еквивалентен на вдигнат бит SYN при TCP протокол.

Да се разрешат входящи уеб съединения (без използване на състояния, без функция на рутер, на самия сървър):

```
# iptables -A INPUT -p tcp --dport 80  -j ACCEPT
```

Редът на правилата е критически важен.

net.ipv4.conf.rp_filter=1 Верификация на IPsrc (защита от IP спуфинг)

Да се спре ping единствено между двойка съседи.

```
# iptables -A INPUT -p icmp --icmp-type echo-request -j DROP
```

Да се фиксира двойка съседи: ляв ssh сървър единствено за десен ssh клиент и десен е единствено ssh клиент за левия ssh сървър. (Политиката е ACCEPT, не се променя).

Правило, отхвърлящо всички изходящи мрежови връзки

Вторият ред от правилата позволява само текущите изходящи и вече създадени връзки. Това е полезно, когато ще се свързвате отдалечено със сървъра през ssh или telnet.

```
# iptables -F OUTPUT

# iptables -A OUTPUT -m state --state ESTABLISHED -j ACCEPT

# iptables -A OUTPUT -j REJECT
```

Правило, отхвърлящо всички входящи мрежови връзки

Вторият ред на правилото позволява само текущите изходящи и установени съединения. Това е много полезно, когато сте се логнал към сървър по ssh или telnet

```
# iptables -F INPUT

# iptables -A INPUT -m state --state ESTABLISHED -j ACCEPT

# iptables -A INPUT -j REJECT
```

Правило, отхвърлящо всички мрежови връзки

Това правило ще отхвърля и блокира всички мрежови връзки, както входящи, така и изходящи. От значение е, че ще се приложи и към вече създадени изходящи връзки.

```
# iptables -F

# iptables -A INPUT -j REJECT

# iptables -A OUTPUT -j REJECT
```

```
# iptables -A FORWARD -j REJECT
```

Правило за отхвърляне на входящи ping заявки

Това правило на iptables ще отхвърля (DROP) всички входящи ping заявки.

Възможно е вместо DROP да се използва (целта) REJECT. Разликата между тях е, че DROP тихомълком ще отхвърли всички входящи пакети, докато REJECT ще върне съобщение за ICMP грешка.

```
# iptables -A INPUT -p icmp --icmp-type echo-request -j DROP
```

Правило за отхвърляне на изходящи telnet връзки

Това правило ще блокира всеки изходящ трафик към всеки възел, където номера на порта е 23 (telnet).

```
# iptables -A OUTPUT -p tcp --dport telnet -j REJECT
```

Правило за отхвърляне на всички входящи telnet връзки

Това правило ще отхвърля всички входящи връзки към локален порт 23.

```
# iptables -A INPUT -p tcp --dport telnet -j REJECT
```

Правило за отхвърляне на изходящи ssh връзки

```
# iptables -A OUTPUT -p tcp --dport ssh -j REJECT
```

Правило за отхвърляне на входящи ssh връзки

отхвърля всички входящи връзки към локален порт 22 (ssh).

```
# iptables -A INPUT -p tcp --dport ssh -j REJECT
```

Правило, отхвърлящо входящия трафик с изключение на ssh и локални връзки

```
# iptables -A INPUT -i lo -j ACCEPT

# iptables -A INPUT -p tcp --dport ssh -j ACCEPT

# iptables -A INPUT -j REJECT
```

Правило, позволяващо входящи ssh връзки от точно определен IP адрес

Използвайки това правило, ние ще блокираме всички входящи връзки към порт 22 (ssh), с изключение на тези, идващи от IP адрес 77.66.55.44, което означава, че само възелът с този адрес ще може да се свърже по ssh.

```
# iptables -A INPUT -p tcp -s 77.66.55.44 --dport ssh -j
ACCEPT
```

```
# iptables -A INPUT -p tcp --dport ssh -j REJECT
```

Правило, приемащо входящи ssh връзки от точно определен MAC адрес.

Това правило блокира всички входящи връзки на порт 22 (ssh), с изключение на тези, идващи възел с MAC адрес 00:e0:4c:f1:41:6b. По този начин, всички ssh връзки ще бъдат ограничени до единичен възел с този MAC адрес.

```
# iptables -A INPUT -m mac --mac-source 00:e0:4c:f1:41:6b -p
tcp --dport ssh -j ACCEPT

# iptables -A INPUT -p tcp --dport ssh -j REJECT
```

Правило, отхвърлящо входящите връзки към точно определен TCP порт

Това правило ще отхвърля всички входящи връзки към TCP порт 3333:

```
# iptables -A INPUT -p tcp --dport 3333 -j REJECT
```

Правило, отхвърлящо всички входящи връзки към точно определен мрежови интерфейс

Това правило ще отхвърля входящия трафик върху точно определен мрежов интерфейс, идващ от подмрежа 192.168.0.0/16. То е полезно за предпазване от подменени IP адреси. Ако eth0 е външен мрежов интрефейс, на него не трябва да попада входящ трафик, произхождащ от вътрешната мрежа.

```
# iptables -A INPUT -i eth0 -s 192.168.0.0/16 -j DROP
```

Правило, извършващо просто маскиране на пакетите

То създава прост шлюз за маскиране на IP адреси, позволяващ на всички възли от една и съща подмрежа да имат достъп до Internet. Използваният в него мрежов интерфейс eth0 е външният мрежов интерфейс, свързан към Internet.

```
# echo "1" > /proc/sys/net/ipv4/ip_forward

# iptables -t nat -A POSTROUTING -o $EXT_IFACE -j MASQUERADE
```

Правило, отхвърлящо входящия telnet трафик, освен от точно определен IP адрес

Това правило ще отхвърля всичкия входящ telnet трафик, с изключение на заявките за връзка от IP адрес 222.111.111.222

```
# iptables -A INPUT -t filter ! -s 222.111.111.222 -p tcp
--dport 23 -j REJECT
```

Правило, отхвърлящо всичкия входящ трафик, освен този от точно определен адресен обхват

Това правило ще отхвърля целият ssh трафик с изключение на заявките за връзка от адресния обхват 10.1.1.90 – 10.1.1.1.100. Премахването на символа „!“ от долното правило ще доведе до отхвърлянето на целият ssh traffic, произхождащ от адресния обхват 10.1.1.90 – 10.1.1.100.

```
# iptables -A INPUT -t filter -m iprange ! --src-range
10.1.1.90-10.1.1.100  -p tcp --dport 22 -j REJECT
```

Правило, отхвърлящо изходящия трафик към точно определен отдалечен възел от мрежата

Това правило ще отхвърли всичкия изходящ трафик, адресиран към отдалечен възел от мрежата с IP адрес 222.111.111.222

```
# iptables -A OUTPUT -d 222.111.111.222 -j REJECT
```

Правило, блокиращо достъпа до определен web сайт

Това правило ще блокира всичкият входящ трафик от facebook.com, в който порта на източника е 80/www.

```
# iptables -A INPUT -s facebook.com -p tcp --sport www -j
DROP
```