КАК ОТКРЫТЬ ПОРТ UBUNTU

Правильная настройка брандмауэра имеет очень важное значение для безопасности вашего сервера или даже домашнего компьютера, подключенного к сети интернет.

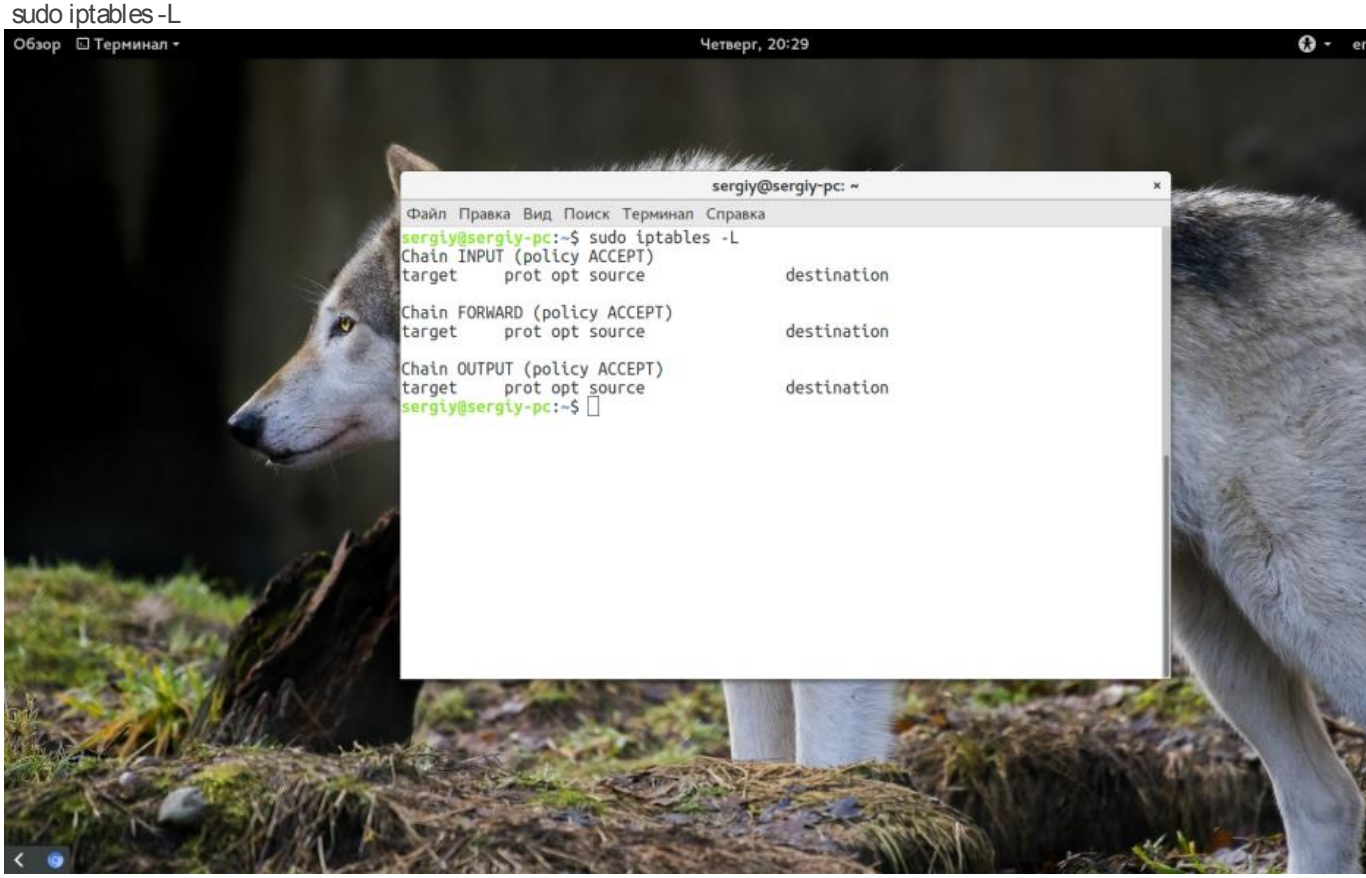
На промышленных серверах брандмауэр запрещает подключение к большинству из них, оставляя только необходимые. В этой статье мы рассмотрим как открыть порт iptables и закрыть все остальные. Хотя в большинстве дистрибутивов существуют специальные утилиты для настройки брандмауэра,мы будем использовать iptables, чтобы вы смогли понять процесс на самом низком уровне.

Содержание статьи:

ПРОСМОТР ПРАВИЛ IPTABLES

Прежде чем что-либо менять, нужно понять каким образом система работает сейчас. Возможно, для лучшего понимания материала вам сначала стоит ознакомиться со статьей iptables для начинающих. Для просмотра текущих правил iptables выполните такую команду:

```
sudo iptables -L
```



Здесь мы видим три цепочки OUTPUT, INPUT и FORWARD, за открытые порты отвечает цепочка INPUT, именно через нее проходят все входящие пакеты. Сейчас политика по умолчанию - ACCEPT, это значит, что подключение ко всем портам разрешено. Здесь нам нужно настроить все самим и это будет проще если бы какая-либо программа уже создала свои настройки, но этот вариант мы тоже рассмотрим ниже.

КАК ОТКРЫТЬ ПОРТ IPTABLES С НУЛЯ

Если в iptables уже есть какие-либо правила и вы хотите их удалить просто выполните:

```
sudo iptables -F
```

Теперь нам нужно добавить правила, которые разрешат обмен данными между любыми портами на локальном интерфейсе lo, это нужно чтобы не вызвать системных ошибок:

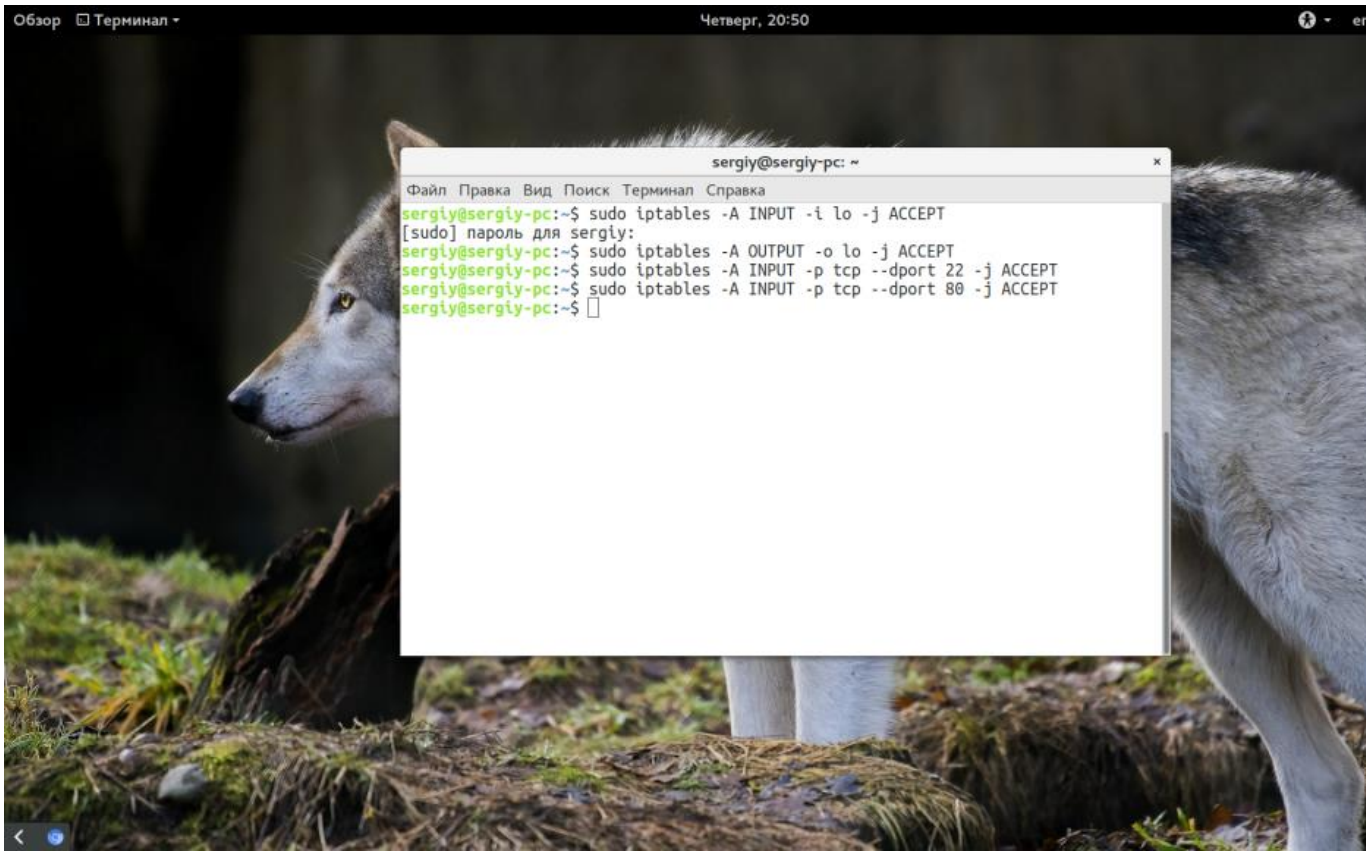
```
sudo iptables -A INPUT -i lo -j ACCEPT
$ sudo iptables -A OUTPUT -o lo -j ACCEPT
```

Если кратко, то здесь добавляется два правила в цепочки INPUT и OUTPUT, разрешающие отправку и прием данных из интерфейса lo. Еще одно интересное и важное правило, которое многие упускают. Нужно запрещать только новые соединения, а пакеты для уже открытых нужно разрешать. Иначе получится, что мы отправляем серверу запрос (цепочка OUTPUT открыта), соединение открывается, но сервер не может нам ответить, потому что все пакеты отбрасываются в INPUT. Поэтому нужно разрешить все пакеты с состоянием ESTABLISHED и RELATED. Для этого есть модуль state:

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

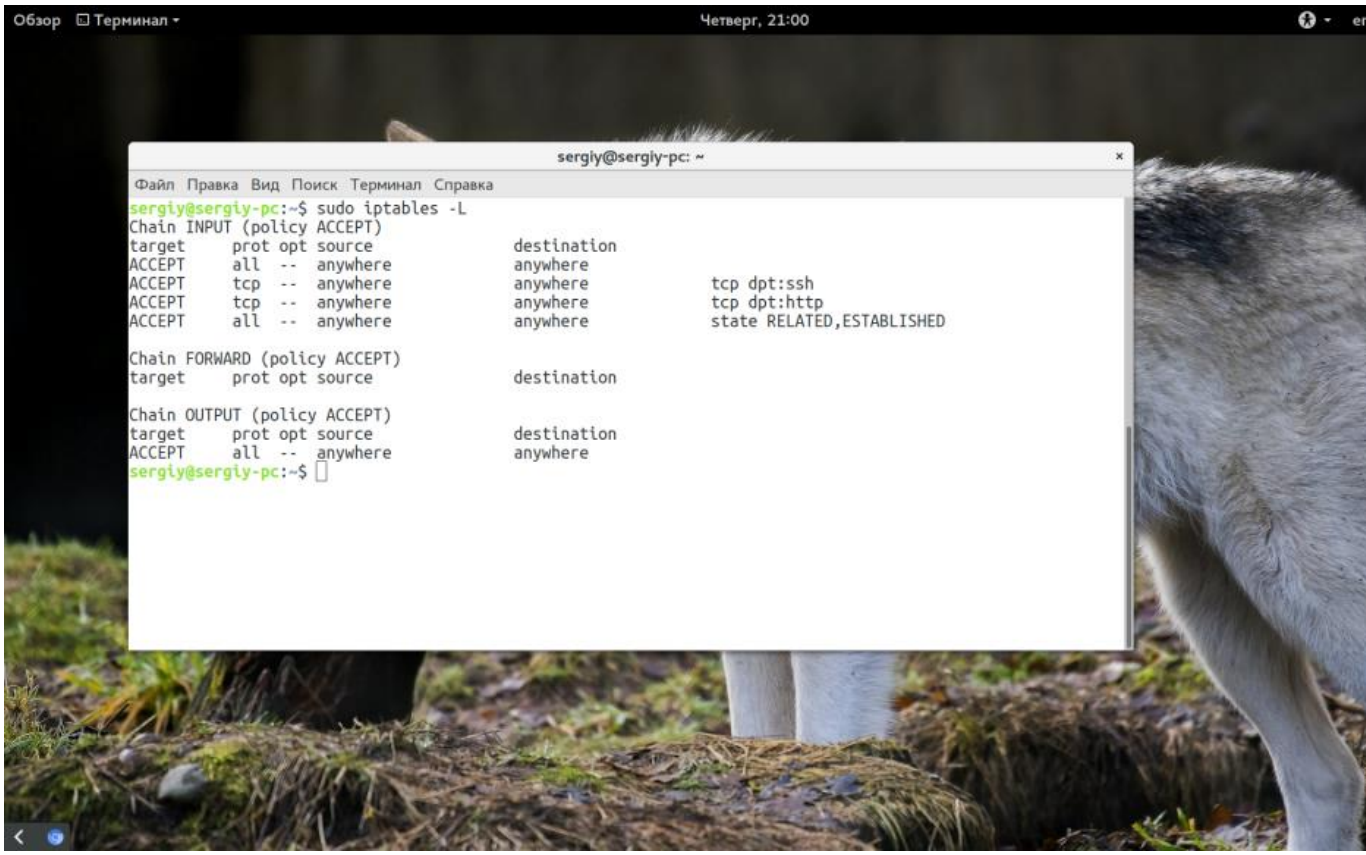
Теперь самое интересное, рассмотрим как открыть порт 22 и 80 для протокола TCP:

```
sudo iptables -A INPUT -p tcp --dport 22 -j ACCEPT
$ sudo iptables -A INPUT -p tcp --dport 80 -j ACCEPT
```



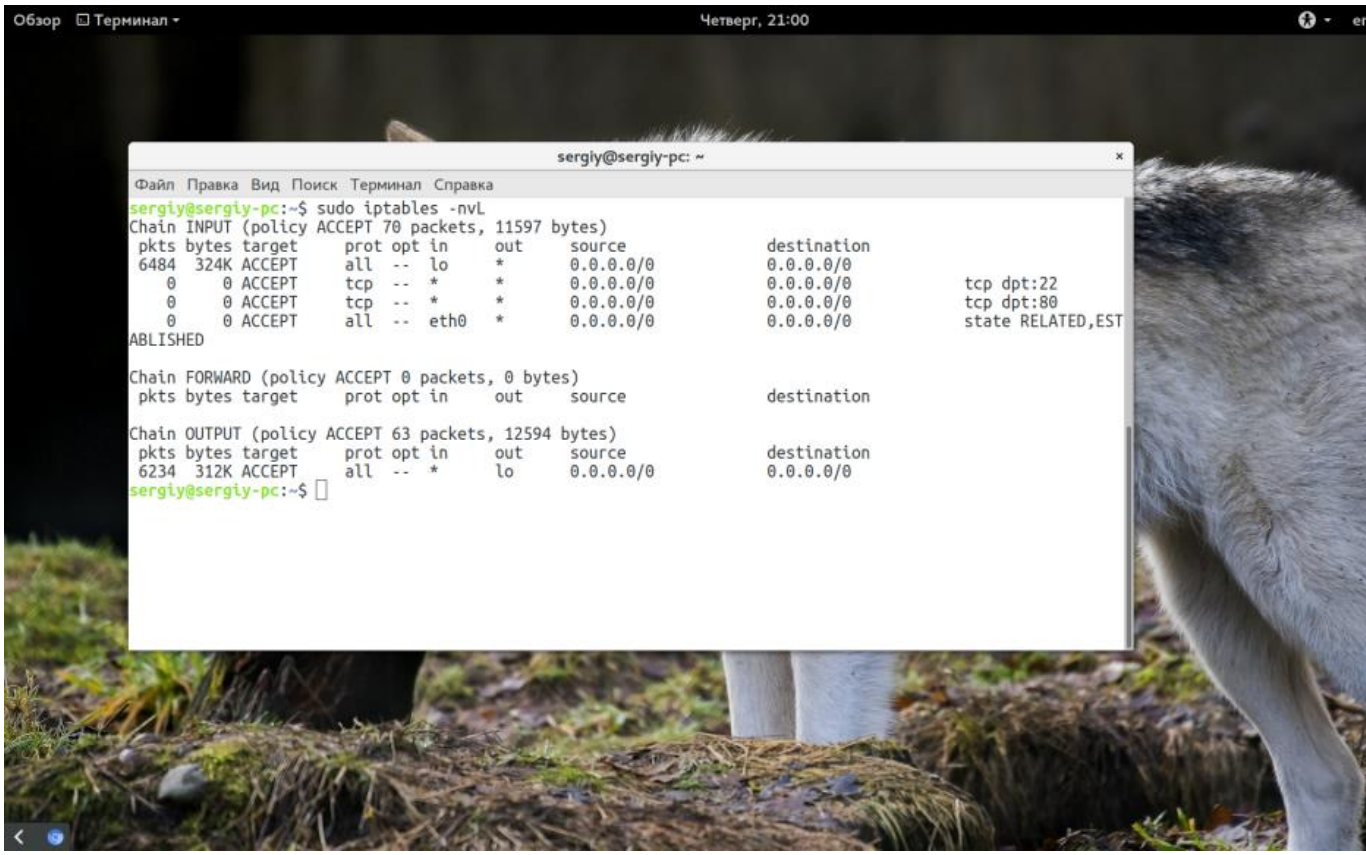
Опция -A сообщает, что нужно добавить пакет в конец цепочки, -p - указывает протокол, а --dport или Destination Port (порт назначения) указывает из какого порта пакеты нужно принимать. Теперь вы можете снова посмотреть список правил:

```
sudo iptables -L
```



Вывод очень упрощен и понять здесь что-то сложно, например, может показаться что у нас два одинаковых правила, хотя это не так. Чтобы отобразить более подробную информацию используйте:

```
sudo iptables -nvL
```



Чтобы все это в действительности заработало, осталось поменять политику по умолчанию на DROP:

```
sudo iptables -P INPUT DROP
```

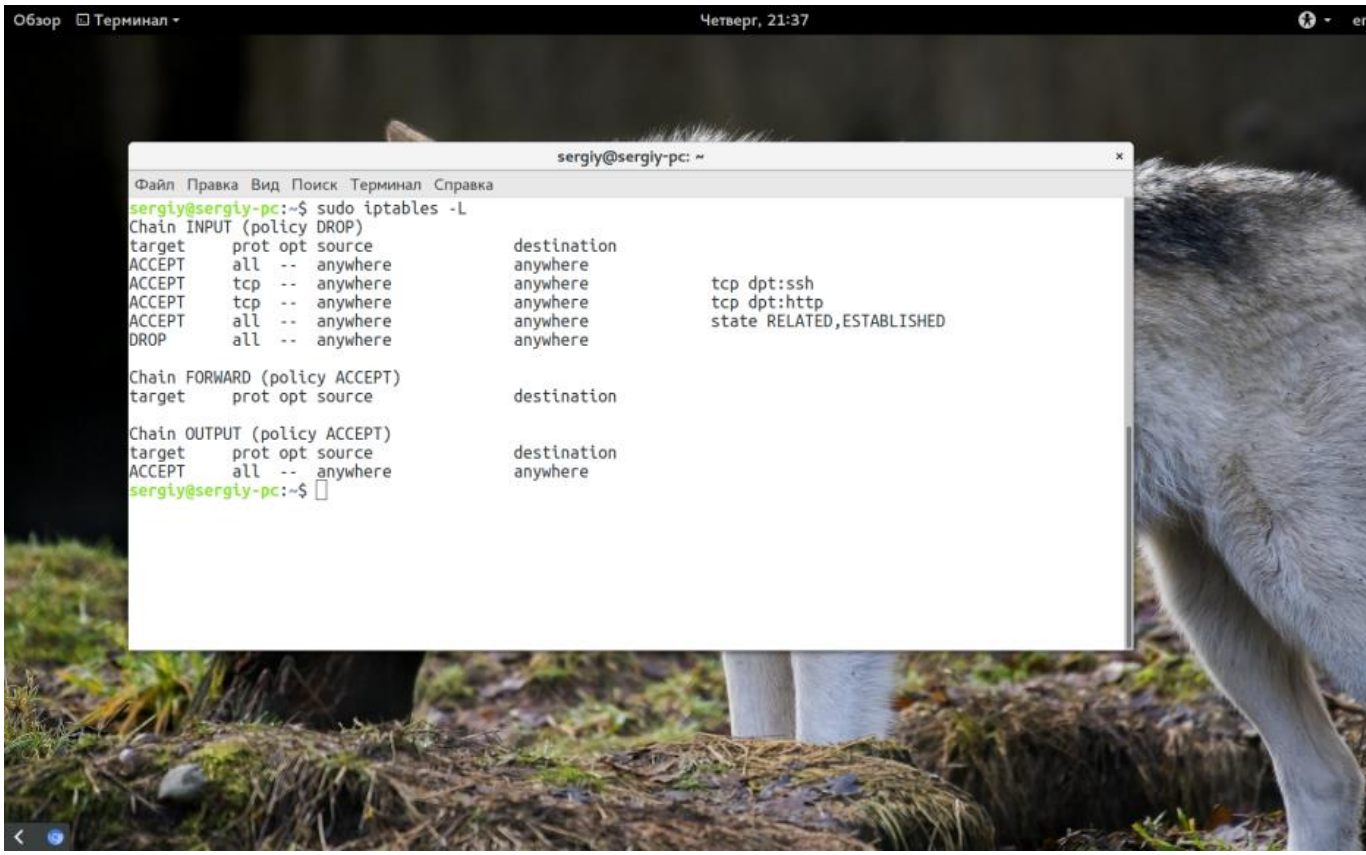
Все, можете проверять. Все пользователи смогут получить доступ к портам 22 и 80, а к остальным доступа не будет.

КАК ОТКРЫТЬ ПОРТ, ЕСЛИ УЖЕ ЕСТЬ ПРАВИЛА

Довольно часто возникает ситуация, когда вам нужно открыть порт Linux, а iptables уже содержит набор правил, запрещающих доступ к портам. Иногда вы добавляете правило, все как нужно, с помощью описанной выше команды, но не замечаете никакого эффекта. Рассмотрим почему так происходит.

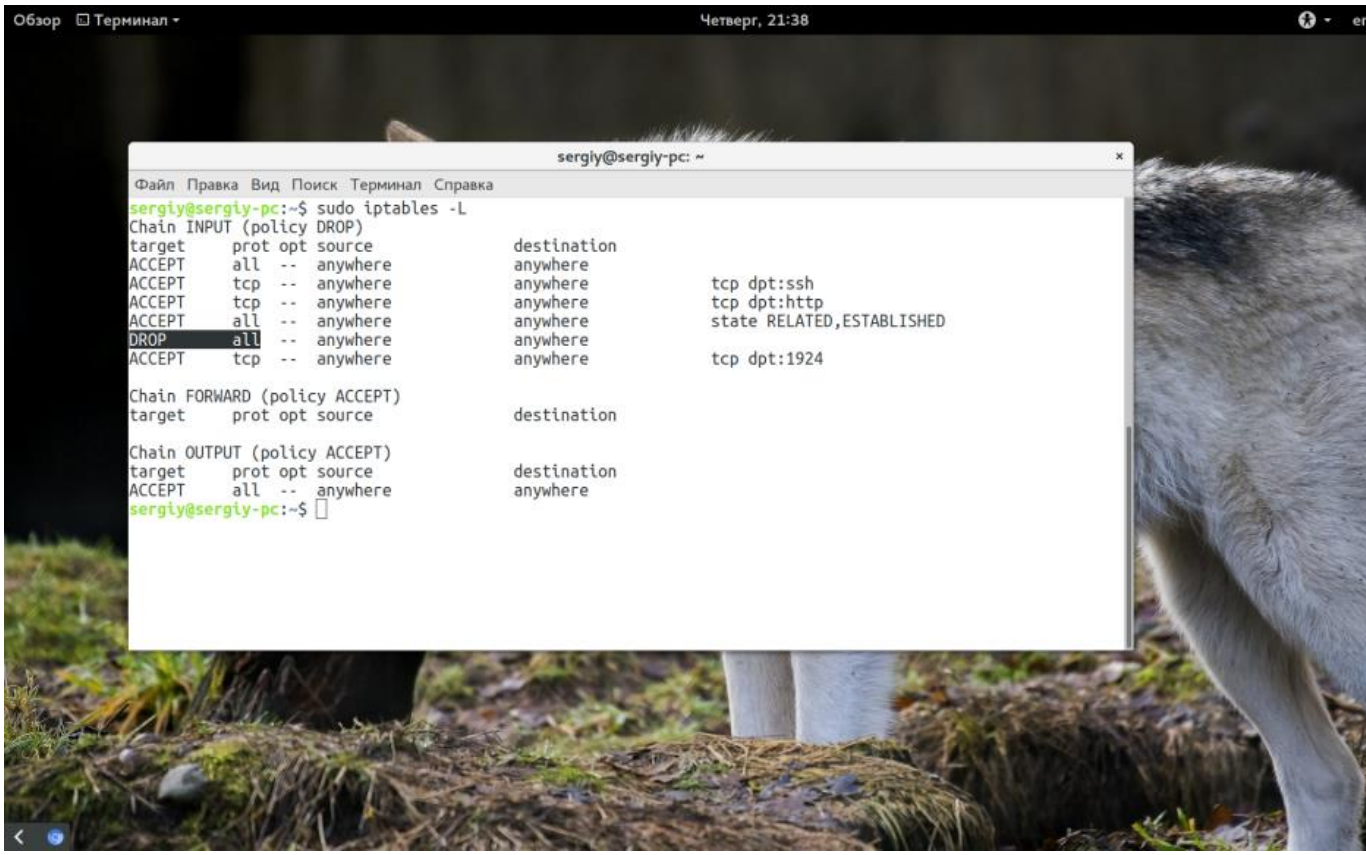
Допустим, программа или предыдущий администратор для надежности добавили в конец цепочки правило такого вида:

```
sudo iptables -A INPUT -j DROP
```



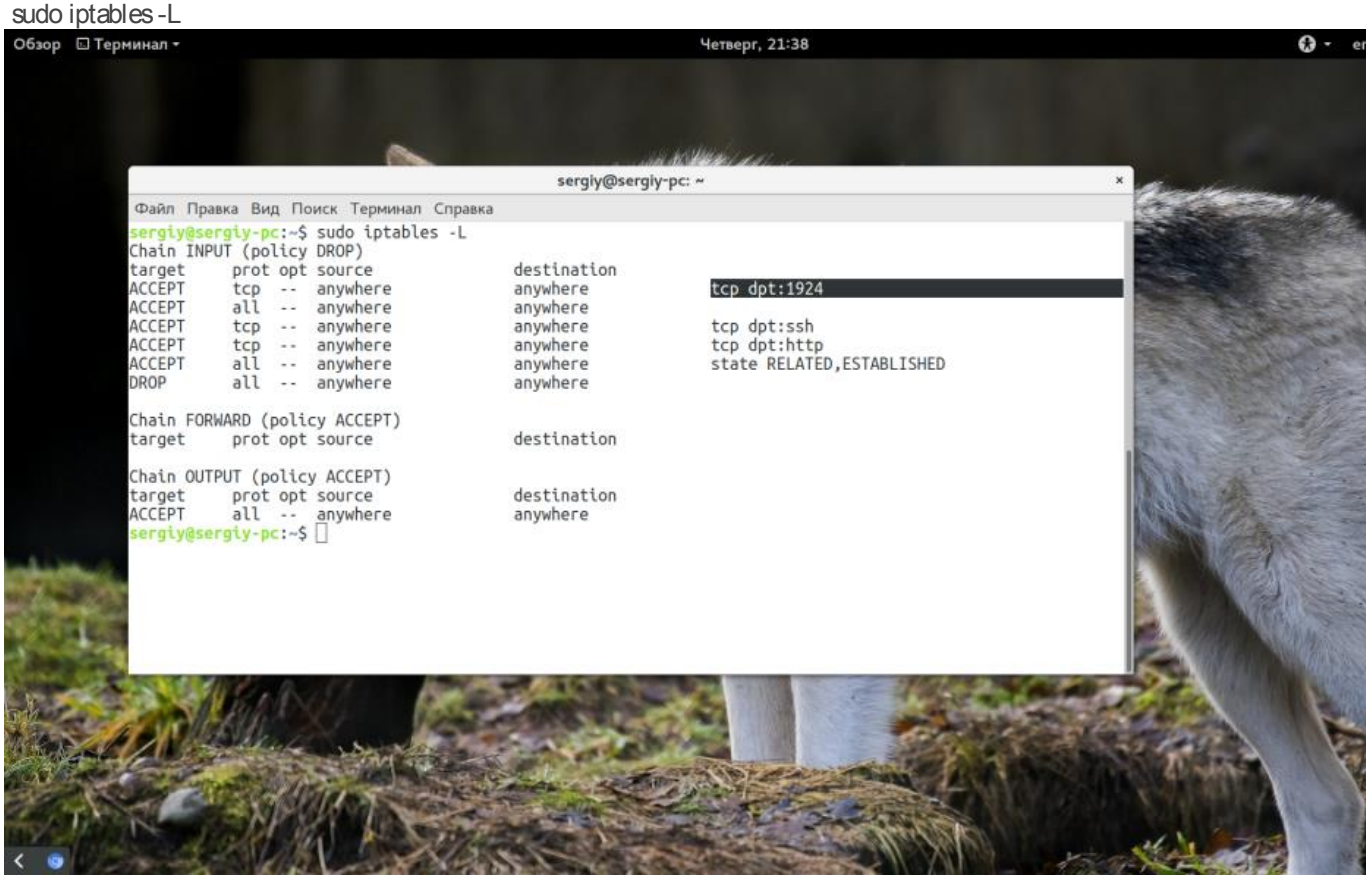
Как вы понимаете, это значит, что все пакеты, которые до него доходят, будут отброшены. Ваше правило добавляется в конец цепочки, уже после этого. Естественно, что к нему уже никакие пакеты не дойдут, потому что они были отброшены ранее. Чтобы обойти эту проблему нужно использовать опцию -I (INSERT) вместо -A (ADD), она добавляет правило в начало цепочки и все будет работать. Осталось открыть порты Linux:

```
sudo iptables -I INPUT -p tcp --dport 1924 -j ACCEPT
```



Теперь смотрим список правил и проверяем:

sudo iptables -L



ВЫВОДЫ

В этой статье мы рассмотрели как открыть порт Ubuntu 16.04 или в любом другом Linux дистрибутиве, а также закрыть ненужные порты. Это повысит безопасность вашей системы. Только на первый взгляд кажется, что с iptables сложно работать. Если разобраться, то все будет достаточно просто. Надеюсь, эта информация была полезной для вас.

ПРОБРОС ПОРТОВ IPTABLES В LINUX

С увеличением количества компьютеров, необходимое количество IP адресов увеличивалось и диапазона IPv4 начало не хватать. Тогда была разработана технология NAT, которая позволяет нескольким компьютерам объединяться в локальную сеть и быть доступными из внешней сети по IP адресу маршрутизатора. Когда пакет приходит на маршрутизатор, выполняется выяснение какому устройству он был адресован и замена ip адресата на нужный. Кроме переопределения IP получателя и отправителя, NAT может изменять порты. Это называется проброс портов и может быть полезно если вы создали частную сеть, но все же хотите пропускать некоторые виды трафика. Всем этим можно управлять с помощью iptables. В этой статье мы рассмотрим как выполняется проброс портов iptables в Linux.

Содержание статьи:

КАК РАБОТАЕТ NAT?

Чтобы иметь возможность общаться с другими компьютерами в сети компьютер должен иметь уникальный ip адрес. Но поскольку количество адресов уменьшалось нужно было придумать технологию, которая позволяла бы давать один адрес нескольким машинам. И была придумана технология NAT или Network Address Translation.

Все работает очень просто. Компьютер имеет свой адрес в локальной сети, он не виден из интернета. Когда ему нужно отправить пакет, он отправляет его роутеру, затем роутер подменяет адрес отправителя на свой и передает пакет дальше к цели. Параллельно роутер запоминает с какого локального компьютера был отправлен пакет на этот адрес. Дальше ответный пакет приходит роутеру, он подменяет адрес назначения на адрес нужного компьютера и отдает пакет в локальную сеть.

Недостаток в том, что инициировать подключение извне нельзя, потому что маршрутизатор просто еще не знает к кому обращаются. Тут на помощь приходит проброс портов. Мы можем сказать роутеру: при поступлении пакетов на порт 80 перенаправлять их на порт 80 компьютера 192.168.1.2. Теперь адрес отправителя и порт будет заменяться на указанный нами и

пакет будет передан туда, куда нужно. Если на маршрутизаторе установлен Linux, то все это можно настроить с помощью iptables.

ПРОБРОС ПОРТОВ В IPTABLES

Первое что нужно сделать, это включить переадресацию трафика на уровне ядра, если это еще не сделано. Для этого выполните:

```
echo 1 | sudo tee /proc/sys/net/ipv4/ip_forward
```

Чтобы настройка сохранялась после перезагрузки используйте такую команду:

```
sudo sysctl -w net.ipv4.ip_forward=1
```

Мы не будем здесь подробно рассматривать правила iptables и значение каждой опции, поэтому перед тем, как читать дальше вам лучше ознакомиться со статьей iptables для начинающих. Дальше рассмотрим проброс портов iptables nat.

НАСТРОЙКА ПРОХОЖДЕНИЯ ПАКЕТОВ

Сначала мы рассмотрим как разрешить прохождение пакетов через маршрутизатор. Для этого в брандмауэре есть цепочка FORWARD. По умолчанию для всех пакетов применяется правило DROP, которое означает что все нужно отбросить. Сначала разрешим инициализацию новых соединений, проходящих от eth0 до eth1. Они имеют тип contrack и представлены пакетом SYN:

```
sudo iptables -A FORWARD -i eth0 -o eth1 -p tcp --syn --dport 80 -m conntrack --ctstate NEW -j ACCEPT
```

Действие ACCEPT означает, что мы разрешаем это соединение. Но это правило разрешает только первый пакет, а нам нужно пропускать любой следующий трафик в обоих направлениях для этого порта (80). поэтому добавим правила для ESTABLIHED и RLEATED:

```
sudo iptables -A FORWARD -i eth0 -o eth1 -m conntrack --ctstate ESTABLISHED,RELATED -j ACCEPT
$ sudo iptables -A FORWARD -i eth1 -o eth0 -m conntrack --ctstate ESTABLISHED,RELATED -j ACCEPT
```

Дальше явно установим что наша политика по умолчанию - DROP:

```
sudo iptables -P FORWARD DROP
```

Это еще не проброс портов, мы только разрешили определенному трафику, а именно на порт 80, проходить через маршрутизатор на другие машины локальной сети. Теперь настроим правила, которые будут отвечать за перенаправление трафика.

МОДИФИКАЦИЯ ПАКЕТОВ В IPTABLES

Далее мы настроим правила, которые будут указывать как и куда нужно перенаправить пакеты, приходящие на порт 80. Сейчас маршрутизатор может

их пропускать в сеть, но он еще не знает куда. Для этого нам нужно будет настроить две вещи - модификацию адреса назначения (Destination) DNAT и модификацию адреса отправителя (Source) SNAT.

Правила DNAT настраиваются в цепочке PREROUTING, в таблице NAT. Эта операция изменяет адрес назначения пакета чтобы он достиг нужной нам цели, когда проходит между сетями. Клиенты будут отправлять пакеты нашему маршрутизатору, и им не нужно знать топологию внутренней сети. Пакет автоматически будет приходить нашему веб-серверу (192.168.1.2).

С помощью этого правила мы перенаправляем все пакеты, пришедшие на порт 80, к 192.168.1.2 опять же на порт 80:

```
sudo iptables -t nat -A PREROUTING -i eth0 -p tcp --dport 80 -j DNAT --to-destination 192.168.1.2
```

Но это только половина работы. Пакет будет иметь исходный адрес клиента, а значит будет пытаться отправить ответ ему. Так как клиент ожидает получить ответ от маршрутизатора, то нормального TCP соединения не получиться. Чтобы решить эту проблему нужно модифицировать адрес источника и заменить его на адрес маршрутизатора 192.168.1.1. Тогда ответ придет маршрутизатору, а тот уже от своего имени передаст его в сеть.

```
sudo iptables -t nat -A POSTROUTING -o eth1 -p tcp --dport 80 -d 192.168.1.2 -j SNAT --to-source 192.168.1.1
```

Если вы хотите перенаправить трафик на порт 8080, то нужно указать его после ip адреса:

```
sudo iptables -t nat -A POSTROUTING -o eth1 -p tcp --dport 80 -d 192.168.1.2 -j SNAT --to-source 192.168.1.1:8080
```

Также может понадобиться выполнить проброс диапазона портов iptables, для этого просто укажите диапазон, например, 1000:2000:

```
sudo iptables -t nat -A POSTROUTING -o eth1 -p tcp --dport 1000:2000 -d 192.168.1.2 -j SNAT --to-source 192.168.1.1
```

После добавления этого правила можете проверять работу перенаправление портов iptables будет выполняться и все будет отправляться так, как нужно.

СОХРАНЕНИЕ НАСТРОЕК IPTABLES

Теперь, когда все настроено, нужно сохранить этот набор правил, чтобы он загружался автоматически при каждом старте. Для этого выполните:

```
sudo service iptables-persistent save
```

Готово. Теперь проброс портов iptables ubuntu будет работать так, как нужно.

ВЫВОДЫ

В этой статье мы рассмотрели как выполняется перенаправление портов iptables. Процесс включает в себя разрешение на проходящий трафик на уровне ядра, разрешение определенного типа трафика, а также настройку адресов источника и назначения для правильного прохождения пакетов.

На завершение, видео о том, что такое NAT:

https://www.youtube.com/watch?v=F7kCOa6PybQ