Как защитить сервер

В этой статье рассмотрим, как защитить сервер, используя разные методы

- Настройка iptables в Linux
- Как защитить сервер: 6 практических методов

Настройка iptables в Linux

Что такое iptables

iptables — это утилита командной строки, используемая для управления встроенным брандмауэром netfilter, доступным в ядре Linux, начиная с версии 2.4. Брандмауэр — это приложение, на котором происходит фильтрация сетевого трафика на основе заданных администратором правил. Обеспечить безопасность сервера или инфраструктуры, означает обеспечить отказоустойчивость и стабильность работы ваших серверов и приложений, что крайне чувствительно для бизнеса или персональных проектов.

В глобальной сети огромное количество угроз — боты, периодически прощупывают стандартные точки входа в системы, хулиганы, любопытные, взломщики — люди, целенаправленно пытающиеся получить несанкционированный доступ к информационным системам. Задача iptables — исключить либо хотя бы минимизировать негативное воздействие со стороны разного рода правонарушителей.

Установка iptables

В образах CentOS 8 64-bit и CentOS 8 Stream 64-bit *iptables* отсутствует, поскольку разработчики отказались от него в пользу более нового пакета — *nftables*. Его поддержка на уровне ядра доступна с версии 3.13. Если существует необходимость использовать именно *iptables*, требуется выполнить следующий порядок действий:

```
yum install iptables-services
```

Включение сервиса в автозагрузку:

```
systemctl enable iptables
```

Запуск сервиса:

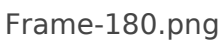
```
systemctl start iptables
```

Межсетевой экран готов к использованию.

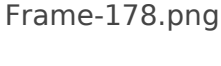
Порядок прохождения таблиц и цепочек

Любой поступивший пакет на сервер с iptables проходит через ядро, а именно — межсетевой экран netfilter. Каждый из них классифицируется в зависимости от его назначения, попадает в соответствующую ему таблицу и проходит по цепочкам, содержащим правила, установленные администратором.

На основе этих правил выполняется действие: принять пакет, отбросить, удалить или передать следующему узлу сети. Иллюстрация, представленная ниже, наглядно показывает путь прохождения пакета по системе:



Данный рисунок не отражает истинную архитектуру брандмауэра, а показывает только логику работы. Существует много распространенных заблуждений по поводу уровней вложенности таблиц, цепочек, и правил. Самым верхним уровнем представления являются таблицы, которые содержат набор свойственных им цепочек. Цепочки содержат списки правил. Схематично «матрешка» выглядит следующим образом:



Синтаксис iptables

Сетевой экран iptables очень гибок в настройке и имеет огромное количество разнообразных ключей и опций. Общий вид управляющей команды:

```
iptables таблица команда цепочка критерии действие
```

Рассмотрим каждый элемент в отдельности.

Пакет

Под пакетом понимают структурированный блок данных, содержащий в себе как пользовательскую информацию, называемую ещё полезной нагрузкой, так и служебную информацию, например об адресе отправителя, получателя, времени жизни пакета и многое

другое.

Цепочки

Существует 5 видов цепочек:

- **PREROUTING** — предназначена для первичной обработки входящих пакетов, адресованных как непосредственно серверу, так и другим узлам сети. Сюда попадает абсолютно весь входящий трафик для дальнейшего анализа.
- **INPUT** — для входящих пакетов, отправленных непосредственно этому серверу.
- **FORWARD** — для проходящих пакетов, не адресованных этому компьютеру, предназначены для передачи следующему узлу, в случае, если сервер выполняет роль маршрутизатора.
- **OUTPUT** — для пакетов, отправленных с этого сервера.
- **POSTROUTING** — здесь оказываются пакеты, предназначенные для передачи на другие узлы сети.

Также есть возможность создавать и удалять собственные цепочки, в большинстве случаев, в этом нет необходимости. Названия цепочек пишут заглавными буквами.

Таблицы

В netfilter существуют 5 типов таблиц, каждая из них имеет свое назначение.

Таблица raw

Содержит цепочки **PREROUTING** и **OUTPUT**, здесь производятся манипуляции с пакетами до задействования механизма определения состояний.

Таблица mangle

Предназначена для модификации заголовков сетевых пакетов, таких параметров как **ToS** (Type of Service), **TTL** (Time To Live), **MARK**. Содержит все существующие пять цепочек.

Таблица nat

Используется для трансляции сетевых адресов, т.е. подмены адреса получателя/отправителя, применяется, если сервер используется в качестве маршрутизатора. Содержит цепочки **PREROUTING**, **OUTPUT**, **POSTROUTING**.

Таблица filter

Основная таблица, служит для фильтрации пакетов, именно здесь происходит принятие решений о разрешении или запрете дальнейшего движения пакета в системе. Используется по умолчанию, если явно не указано имя другой таблицы. Содержит цепочки **INPUT**, **FORWARD** и **OUTPUT**.

Таблица security

Используется для взаимодействия с внешними системами безопасности, в частности с SELinux и AppArmor. Содержит цепочки **INPUT**, **OUTPUT** и **FORWARD**.

Имена таблиц принято писать строчными буквами.

Действия

Правилами задается поведение для iptables, каким образом поступить с тем или иным пакетом при попадании под заданные критерии. Решения, которые принимает брандмауэр, называют действиями, самые распространенные из них:

- **ACCEPT** — разрешить дальнейшее прохождение пакета по системе;
- **DROP** — выбросить пакет без уведомления отправителя;
- **REJECT** — отказать в прохождении пакета с уведомлением отправителя, такой способ может привести к дополнительным затратам ресурсов процессора, поэтому, в большинстве случаев рекомендуется использовать DROP;
- **LOG** — зафиксировать информацию о пакете в файле системного журнала;
- **MARK** — позволяет помечать определенные пакеты, например для маршрутизации, данная метка перестает существовать, как только пакет покинет брандмауэр;
- **CONNMARK** — то же самое, что и MARK, только для соединений;
- **QUEUE** — отправляет пакет в очередь приложению для дальнейшего взаимодействия;
- **RETURN** — прекращение движения пакета по текущей цепочке и возврат в предыдущую цепочку. Если текущая цепочка единственная — к пакету будет применено действие по умолчанию;
- **REDIRECT** — перенаправляет пакет на указанный порт, в пределах этого же узла, применяется для реализации «прозрачного» прокси;
- **DNAT** — подменяет адрес получателя в заголовке IP-пакета, основное применение — предоставление доступа к сервисам снаружи, находящимся внутри сети;
- **SNAT** — служит для преобразования сетевых адресов, применимо, когда за сервером находятся машины, которым необходимо предоставить доступ в Интернет, при этом от провайдера имеется статический IP-адрес;
- **MASQUERADE** — то же, что и SNAT, но главное отличие в том, что может использоваться, когда провайдер предоставляет динамический адрес, создаёт дополнительную нагрузку на систему по сравнению с SNAT;
- **TOS** — позволяет управлять битами в одноименном поле заголовка IP-пакета;
- **ULOG** — более продвинутый вариант записи информации, может писать как в обычный текстовый файл, так и в базу данных;
- **TTL** — используется для изменения значения поля одноименного заголовка IP-пакета, устанавливает время жизни пакета.

Команды

Для iptables команда — это инструкция к действию, при помощи них можно добавлять, удалять и сбрасывать все правила, задавать действия по умолчанию и многое другое. Команды могут подаваться как в сокращенном, так и в полном виде, более подробно в таблице:

d9fc616ce5d293bcde5d6.png

b20ad97ff7f5eeeeb0153.png

Критерии

Чтобы к пакету применить какое-либо действие, он должен попасть под определенные критерии. Одно правило может содержать несколько критериев. Они, как и команды, имеют полную и сокращенную форму. Некоторые из них поддерживают логическую **НЕ**, если перед ними поставить знак **!** — критерий будет инвертирован. Список в таблице ниже:

69a83e856dac60533d7dd.png

69797e7d64163f45d474f.png

Состояние соединений

Система отслеживания состояния соединений conntrack — важная часть сетевого стека linux, встроенная в ядро. Используется для сопоставления пакетов с конкретными соединениями. Под анализ попадают все пакеты, кроме помеченных **NOTRACK**, в таблице **raw**. Все пакеты классифицируются на:

- **NEW** — открывается новое соединение, пришел только первый пакет;
- **ESTABLISHED** — соединение установлено, пришел уже не первый пакет в рамках этого сеанса. При правильной настройке iptables — такие пакеты проходят по системе без фильтрации, поскольку она уже была выполнена для первого пакета соединения;
- **RELATED** — открывается новое соединение, связанное с другим сеансом, имеющим статус ESTABLISHED;
- **INVALID** — помечаются пакеты, которые не связаны ни с одним из существующих соединений, и не могут создать новое, их невозможно идентифицировать. В целях безопасности рекомендуется остановить движение таких пакетов по системе, используя действие DROP.

Основные команды iptables

Как посмотреть список правил iptables

```
iptables --line-numbers -L -v -n
```

Ключ *—line-numbers* нумерует строки, *-L* выводит список правил всех цепочек, *-v* отвечает за детализацию вывода, *-n* выводит IP-адреса и номера портов в числовом формате.

Как удалить правило в iptables:

В первую очередь необходимо определить номер правила, которое требуется удалить, выводим список действующих правил командой:

```
iptables --line-numbers -L -v -n
```

Предположим, требуется удалить правило 4 в цепочке **INPUT**:

```
iptables -D INPUT 4
```

Как сохранить правила iptables

Утилита iptables, как и маршрутизаторы Cisco, не сохраняет правила, если это явно не указать и после перезагрузки возвращается в предыдущее состояние. Установим пакет:

```
apt install iptables-persistent
```

В процессе установки на оба вопроса ответить Yes. Сохранить текущие правила:

```
service netfilter-persistent save
```

Система при следующей загрузке использует последние сохраненные правила.

Как восстановить правила

В процессе настройки брандмауэра, по разным причинам, возникает необходимость вернуться к заведомо рабочим, испытанным правилам. Сервис *netfilter-persistent* сохраняет их в файле */etc/iptables/rules.v4*, если не успели сохранить активные правила, значит в файле предыдущая версия, восстанавливаем:

```
iptables-restore < /etc/iptables/rules.v4
```

Примеры настройки iptables

В данном разделе рассмотрим задачи, с которыми придется столкнуться, работая с iptables.

Как заблокировать IP-адрес в iptables

Допустим, необходимо заблокировать компьютер с IP-адресом 172.10.10.1, тогда правило будет выглядеть следующим образом:

```
iptables -A INPUT -s 172.10.10.1 -j DROP
```

Как разрешить IP-адрес в iptables

Необходимо разрешить весь трафик к серверу для клиента с IP-адресом 192.168.111.1:

```
iptables -A INPUT -s 192.168.111.1 -j ACCEPT
```

Как открыть порт в iptables

Предположим, что политика по умолчанию — блокировать все, что явно не разрешено. Откроем порты веб-сервера для обеспечения работы HTTP протокола — порт 80, и поддержки HTTPS протокола совместно с SSL — порт 443. Также для доступа к серверу по SSH откроем порт 22. Эту задачу можно решить как минимум двумя способами: создать однострочное правило, либо прописать правила по каждому из портов, рассмотрим оба. В одну строку:

```
iptables -A INPUT -p tcp -m multiport --dports 22,80,443 -j ACCEPT
```

При использовании расширения multiport, всегда необходимо использовать критерий *-p tcp* или *-p udp*, таким образом, одной строкой можно указать до 15 разных портов через запятую. Важно не путать критерии *—-dport* и *—-dports*. Первый из них используется для указания одного порта, второй сразу для нескольких, аналогично с *—sport* и *—sports*.

Вариант многострочной записи — для каждого порта свое правило:

```
iptables -A INPUT -p tcp --dport 22 -j ACCEPT
iptables -A INPUT -p tcp --dport 80 -j ACCEPT
iptables -A INPUT -p tcp --dport 443 -j ACCEPT
```

Как защитить сервер: 6 практических методов

Любая IT-инфраструктура нуждается в надежной защите. Информационная безопасность — это тема, которую не охватить за пару уроков. Однако существует некоторый минимум, который поможет защититься от атак непрофессиональных хакеров и ботов. В этой статье рассмотрим, как защитить сервер, используя 6 несложных методов.

Инструменты и методы защиты

Обеспечение защиты сервера от взлома всегда включает комплекс различных мер. Условно методы можно разделить на следующие направления:

- Защита каналов связи, через которые осуществляется администрирование и использование системы.
- Организация нескольких уровней безопасности системы.
- Разграничение доступа к ресурсам инфраструктуры.
- Мониторинг и аудит систем.
- Резервное копирование.
- Своевременные обновления (или откаты) ПО.
- Антивирусная защита серверов.

Далее рассмотрим 6 практических методов, позволяющих получить уровень защиты, который не по зубам непрофессиональным взломщикам и ботам.

Разграничение привилегий

При организации доступа к ресурсам следуйте универсальному правилу — процессы и пользователи должны иметь доступ только к тем ресурсам, которые минимально необходимы для работы. Особенно это касается баз данных и операционных систем. Принцип наименьших привилегий поможет наладить защиту сервера от несанкционированного доступа извне, минимизировав ущерб, а также от внутренних угроз.

Для каждого администратора лучше всего создать отдельную учетную запись, а операции, не требующих повышенных прав, нужно выполнять с непривилегированных аккаунтов. При использовании среды Microsoft Active Directory периодически проводите проверки и конфигурирование групповых политик, так как такой механизм в руках злонамеренного администратора или хакера может привести к серьезным нарушениям безопасности.

При работе с *nix системами не следует постоянно работать под учетной записью `root`. Лучше всего ее отключить и использовать программу `sudo`. Настройки `sudo` можно изменить в файле `/etc/sudoers` или командой `visudo`. Приведем две полезные директивы `defaults`, которые помогут следить за тем, кто что делает через `sudo`.

По умолчанию лог пишется в `syslog`. Следующая настройка (файл `/etc/sudoers`) аккумулирует записи в отдельный файл для удобства:

```
Defaults log_host, log_year, logfile="/var/log/sudo.log"
```

Этот параметр заставляет sudo записывать текст сеанса работы (лог команд, сообщения `stdin`, `stderr`, `stdout` и лог с `tty/pty`) в директорию `/var/log/sudo-io`:

```
Defaults log_host, log_year, logfile="/var/log/sudo.log"
```

Мандатное управление доступом

Следующий совет касается Linux-систем и связан с предыдущим. Многие linux-админы довольствуются дискреционными механизмами разграничения доступа, которые являются основными и всегда активны. Между тем во многих дистрибутивах (AppArmor в Ubuntu, SELinux в RHEL-based системах) имеются механизмы мандатного управления. Они требуют более сложной настройки ОС и сервисов, зато позволяют детально разграничить доступ к объектам файловой системы, обеспечивая более надежную программную защиту сервера.

Удаленное администрирование ОС

При удаленном администрировании операционной системы используйте безопасные протоколы. Для Windows таким считается RDP, в Linux — SSH. Хоть эти протоколы и являются надежными, можно дополнительно усилить защиту.

Для RDP желательно заблокировать подключения учетных записей с пустым паролем. Сделать это можно через «Локальные политики безопасности» и параметр «Учетные записи:

Разрешить использование пустых паролей только при консольном входе». Если не используется VPN, RDP-сессии можно защитить безопасным транспортным протоколом TLS, речь о котором пойдет чуть позже.

По умолчанию, проверка личности пользователя в SSH происходит по паролю. Установив аутентификацию по SSH-ключам, вы повысите защиту сервера, так как длинный ключ значительно сложнее подобрать, к тому же не придется вводить пароль (ключ хранится на сервере). Настройка ключей требует всего несколько простых шагов:

Генерация пары ключей на локальной машине:

```
ssh-keygen -t rsa
```

Размещение ключей на удаленной станции:

```
ssh-copy-id логин@адрес
```

Если не хотите использовать ключи, присмотритесь к программе Fail2ban, которая ограничивает число попыток ввода пароля и блокирует IP-адреса. Также же не лишним будет поменять порты по умолчанию: 22/tcp для SSH, 3389/tcp для RDP.

Настройка фаервола

Правильная система безопасности состоит из уровней. Не стоит надеяться только на механизмы разграничения доступа. Логичней контролировать сетевые соединения до того, как они доберутся до сервисов. Для этого существуют файрволы.

Межсетевой экран (брандмауэр или файрвол) обеспечивает контроль доступа на уровне сети к участкам инфраструктуры. Руководствуясь определенным набором разрешающих правил, файрвол определяет, какой трафик пропускать через периметр. Все, что под правила не попадает, блокируется. Следует заметить, что в Linux, брандмауэр является частью ядра (netfilter), поэтому для работы в пользовательском пространстве необходимо установить фронтенд: nftables, iptables, ufw или firewalld.

Первое, что нужно сделать при настройке фаервола — закрыть неиспользуемые порты и оставить только те, к которым предполагается доступ извне. Например, для веб-сервера это порт 80 (http) и 443 (https). Ничего кардинально опасного в открытом порту нет (угроза может быть в программе, стоящем за портом), но все же лучше убрать лишнее.

Помимо обеспечения внешнего периметра безопасности, межсетевые экраны помогают разделить инфраструктуру на сегменты и контролировать трафик между ними. Если у вас имеются общедоступные сервисы, подумайте, можно ли их изолировать от внутренних ресурсов (DMZ). Также советуем присмотреться к системам обнаружения и предотвращения

вторжений (IDS/IPS). Этот вид решений работает по обратному принципу — заблокировать проблему безопасности, все остальное пропустить.

Виртуальные частные сети

До этого мы рассматривали, как защитить сервер от взлома. Теперь рассмотрим защиту нескольких серверов. Сейчас виртуальные частные сети (VPN) чаще всего применяют в качестве анонимайзера и инструмента доступа к недоступным ресурсам. Однако основное их назначение – безопасное объединение сетей филиалов организаций. В своей сути VPN представляет собой логическую сеть поверх другой сети (например, Интернет). Безопасность обеспечивается средствами криптографии, поэтому защищенность соединений не зависит от безопасности базовой сети.

Существует множество VPN-протоколов. Выбор зависит от размеров организации, сети и требуемого уровня безопасности. Для маленькой фирмы и домашней локальной сети подойдет классический PPTP: почти на любом роутере или телефоне есть возможность настроить pptp. Из недостатков можно отметить устаревшие методы шифрования. Для высокого уровня защищенности и соединений типа сеть-сеть подходящим будут протоколы IPsec, для соединений сеть-узел — OpenVPN и WireGuard Однако они требуют более тонкой настройки, в отличие от PPTP.

TLS и инфраструктура открытых ключей

Многие протоколы прикладного уровня разрабатывались во времена, когда сети не выходили за пределы институтов и военных учреждений, а web еще не изобрели. HTTP, FTP, SMTP и другие популярные протоколы передают данные в виде обычного текста. Если хотите обеспечить защиту сайта, веб-панели управления внутренним сервисом или почты, используйте TLS.

TLS — это протокол защиты транспортного уровня, предназначенный для безопасной передачи данных в небезопасной сети. Хоть вместе с TLS часто встречается название SSL (SSL-сертификат, пакет OpenSSL), учитывайте, что современной версией протокола является TLS 1.2/1.3. Ранние версии TLS и протокол-предшественник SSL считаются устаревшими.

TLS позволяет обеспечить приватность, целостность данных и аутентификацию ресурса. Последнее достигается с помощью цифровой подписи и инфраструктуры открытых ключей (PKI). PKI работает следующим образом: подлинность сервера определяется SSL-сертификатом, который подписывается удостоверяющим центром (CA). Сертификат центра

в свою очередь подписывается вышестоящим CA и так далее по цепочке. Сертификаты корневого центра являются самоподписанными, то есть доверие к ним подразумевается по умолчанию.

TLS также можно использовать вместе с VPN, например настроить авторизацию клиентов по SSL-сертификатам или TLS handshake. В таком случае необходимо внутри локальной сети самостоятельно организовать инфраструктуру открытых ключей (сервер CA, ключи и сертификаты узлов).

Чем опасны взломщики?

Степень опасности угрозы зависит от ее вида. Атаки условно делятся на несколько видов.

Первый вид связан с проникновением за периметр безопасности. В этом случае злоумышленник получает доступ к учетной записи авторизованного пользователя сервиса или системы, например базы-данных. Угрозу представляют взломы привилегированных аккаунтов, так как в руки хакера попадают средства просмотра секретной информации и изменения параметров системы. Критически опасная разновидность «проникновения» — несанкционированный доступ к учетной записи суперпользователя операционной системы. В подобной ситуации под удар попадает большая часть инфраструктуры.

Другой вид атак нацелен на вывод системы из строя. Подобные угрозы не подразумевают утечек данных, но это не делает их менее опасными. Самая показательная атака такого рода — DoS и DDoS. Их суть заключается в том, что злоумышленники нагружают сервер лавиной обращений. Рабочий сервер не справляется с нагрузкой и перестает отвечать на запросы пользователей. Иногда DoS бывает подспорьем для осуществления других атак.

Результатами атак часто становится утечки данных, финансовый и репутационный ущерб, поэтому при налаживании IT-инфраструктуры важно продумать хотя бы минимальный уровень безопасности.