Утилита ufw. Утилита для упрощённой настройки брандмауэра.

Category: firewall,GNU/Linux,Безопасность,утилиты
2025-07-14

Во всех дистрибутивах **GNU\Linux** для обеспечения сетевой безопасности и изоляции внутренних процессов от внешней среды используется **iptables**. Его настройка может показаться очень сложной для новых пользователей, поэтому многие дистрибутивы создают собственные оболочки, которые упрощают процесс настройки.

В **Debian** используется оболочка под названием **UFW** (полное название *Uncomplicated FireWall*). Это упрощённый интерфейс к **iptables** — системе фильтрации сетевого трафика в **GNU\Linux**.

Синтаксис.

Для управления возможностями брандмауэра используется одноимённая команда — ufw.

Синтаксис команды такой:

```
$ ufw опции действие параметры
```

Опции определяют общие настройки поведения утилиты, действие указывает, что нужно сделать, а параметры — дополнительные сведения для действия, например, **IP-адрес** или номер порта.

Например:

--version — вывести версию брандмауэра;
--dry-run — тестовый запуск, никакие реальные действия не выполняются.

Вывод списка возможных команд **UFW**:

```
$ sudo ufw show
```

Команды.

Для выполнения действий с утилитой доступны такие команды:

enable — включить фаерволл и добавить его в автозагрузку;

disable — отключить фаерволл и удалить его из автозагрузки;

reload — перезагрузить файервол;

default — задать политику по умолчанию, доступно allow, deny и reject, а также три вида трафика — incoming, outgoing или routed;

logging — включить журналирование или изменить уровень подробности;

reset — сбросить все настройки до состояния по умолчанию;

status — посмотреть состояние фаервола;

show — посмотреть один из отчётов о работе;

allow — добавить разрешающее правило;

deny — добавить запрещающее правило;

reject — добавить отбрасывающее правило;

limit — добавить лимитирующее правило;

delete — удалить правило;

insert — вставить правило.

Настройка.

Как включить.

В серверных версиях **UFW** по умолчанию включён, а в версии для **Desktop** он отключён.

Внимание! Обратите внимание, что если вы работаете по **SSH**, то перед тем, как включать брандмауэр, нужно добавить правило (или убедиться, что оно есть), разрешающее работу по **SSH**, иначе у вас не будет доступа к серверу.

Сначала смотрим состояние фаервола:

```
$ sudo ufw status
```

```
$ sudo ufw allow OpenSSH
```

Если он не включён, то его необходимо включить:

```
$ sudo ufw enable
```

Перезапустить с применением правил:

sudo ufw reload

Затем вы можете снова посмотреть состояние:

$ sudo ufw status

Политика по умолчанию.

Перед тем, как мы перейдём к добавлению правил, необходимо указать политику по умолчанию. Какие действия будут применяться к пакетам, если они не подпадают под созданные правила **UFW**?

Все входящие пакеты будем отклонять:

$ sudo ufw default deny incoming

А все исходящие разрешим:

$ sudo ufw default allow outgoing

Имена сервисов.

Например, когда вы используете команду ufw allow OpenSSH, **UFW** (**Uncomplicated Firewall**) берет имена сервисов из файла /etc/services. Этот файл содержит список известных сетевых сервисов и соответствующих им портов и протоколов.

Если вы хотите увидеть полный список доступных сервисов, вы можете открыть файл /etc/services в текстовом редакторе или использовать команду cat /etc/services в терминале.

$ sudo cat /etc/services

Когда вы указываете имя сервиса (например, OpenSSH), **UFW** ищет это имя в файле /etc/services, чтобы определить, какой порт и протокол следует использовать. Например, для OpenSSH **UFW** найдет запись, которая указывает на порт 22 и протокол **TCP**.

Пример записи в /etc/services для **OpenSSH**:

$ sudo grep -i ssh /etc/services

Ответ:

```
ssh             22/tcp                          # SSH Remote Login Protocol
```

Таким образом, это будет работать так:

```
$ sudo ufw allow OpenSSH
```

Таким образом, команда ufw allow OpenSSH фактически разрешает входящие соединения на порт 22 по протоколу **TCP**.

Добавление правил.

Чтобы создать разрешающее правило, используется команда allow.

Вместо allow могут использоваться и запрещающие правила **UFW** — deny и reject.

Правило deny, используется просто блокировать трафик без уведомления клиента, а правило reject, используется если нужно явно сообщить клиенту, что его запрос был отклонен.

Правило allow.

Для добавления правил можно использовать простой синтаксис:

```
$ ufw allow имя_службы
$ ufw allow порт
$ ufw allow порт/протокол
```

Например, чтобы открыть порт **UFW** для **SSH**, можно добавить одно из этих правил:

```
$ sudo ufw allow OpenSSH
$ sudo ufw allow 22
$ sudo ufw allow 22/tcp
```

Первое и второе правила разрешают входящие и исходящие подключения к порту 22 для любого протокола, третье правило разрешает входящие и исходящие подключения для порта 22 только по протоколу **TCP**.

Посмотреть доступные имена приложений можно с помощью команды:

```
$ sudo ufw app list
```

Можно также указать направление следования трафика с помощью слов out для исходящего и in для входящего.

```
$ ufw allow направление порт
```

Правило deny.

Действие: Просто блокирует трафик без отправки какого-либо ответа.

Поведение: Когда правило deny применяется, пакеты, соответствующие этому правилу, просто отбрасываются. Клиент, отправляющий запрос, не

получает никакого уведомления о том, что его запрос был заблокирован.

Пример использования: Это может быть полезно, если вы хотите скрыть факт существования сервиса или просто молча блокировать нежелательный трафик.

Примеры команды:

```
$ sudo ufw deny from 192.168.1.100
```

Эта команда блокирует весь трафик с **IP-адреса** 192.168.1.100 без отправки какого-либо ответа.

Правило reject.

Действие: Блокирует трафик, но отправляет обратно сообщение об ошибке.

Поведение: Когда правило reject применяется, пакеты, соответствующие этому правилу, отбрасываются, и клиенту отправляется сообщение об ошибке (например, ***ICMP unreachable*** для **TCP/UDP** или ***ICMP port unreachable*** для других протоколов). Это сообщение информирует клиента о том, что его запрос был отклонен.

Пример использования: Это может быть полезно, если вы хотите явно сообщить клиенту, что его запрос был заблокирован, что может помочь в диагностике проблем или предотвращении повторных попыток подключения.

Примеры команды:

```
$ sudo ufw reject from 192.168.1.100
```

Эта команда блокирует весь трафик с **IP-адреса** 192.168.1.100, но отправляет клиенту сообщение об ошибке.

Некоторые примеры.

Например, разрешим только исходящий трафик на порт 80, а входящий запретим:

```
$ sudo ufw allow out 80/tcp

$ sudo ufw deny in 80/tcp
```

Также можно использовать более полный синтаксис добавления правил:

```
ufw allow proto протокол from ip_источника to ip_назначения port
порт_назначения
```

В качестве ip_источника может использоваться также и адрес подсети.

Например, разрешим доступ со всех **IP-адресов** для интерфейса eth0 по протоколу **TCP** к нашему **IP-адресу** и порту 3318:

```
$ sudo ufw allow proto tcp from 0.0.0.0/24 to 192.168.1.5 port 3318
```

Правило limit.

С помощью правил limit можно ограничить количество подключений к определённому порту с одного **IP-адреса**, это может быть полезно для защиты от атак перебора паролей. По умолчанию подключения блокируются, если пользователь пытается создать шесть и больше подключений за 30 секунд:

```
$ sudo ufw limit ssh/tcp
```

К сожалению, настроить время и количество запросов можно только через **iptables**.

Просмотр состояния.

Посмотреть состояние и действующие на данный момент правила можно командой status:

```
$ sudo ufw status
```

Чтобы получить более подробную информацию, используйте параметр verbose:

```
$ sudo ufw status verbose
```

Ответ:

```
Status: active
Logging: on (low)
Default: deny (incoming), allow (outgoing), disabled (routed)
New profiles: skip

To                         Action      From
--                         ------      ----
[redacted]/tcp (OpenSSH)   ALLOW IN    Anywhere
[redacted]/tcp             ALLOW IN    Anywhere
[redacted]/tcp             ALLOW IN    Anywhere
[redacted]/tcp             ALLOW IN    Anywhere
[redacted]/tcp (OpenSSH (v6))  ALLOW IN    Anywhere (v6)
[redacted]/tcp (v6)        ALLOW IN    Anywhere (v6)
[redacted]/tcp (v6)        ALLOW IN    Anywhere (v6)
[redacted]/tcp (v6)        ALLOW IN    Anywhere (v6)
```

С помощью команды show можно посмотреть разные отчеты:

- **raw** — все активные правила в формате **iptables**;
- **builtins** — правила, добавленные по умолчанию;
- **before-rules** — правила, которые выполняются перед принятием пакета;
- **user-rules** — правила, добавленные пользователем;
- **after-rules** — правила, которые выполняются после принятия пакета;
- **logging-rules** — правила журналирования пакетов;
- **listening** — отображает все прослушиваемые порты и правила для них;
- **added** — недавно добавленные правила.

Например, посмотрим список всех правил **iptables**:

```
$ sudo ufw show raw
```

Посмотрим все прослушиваемые порты:

```
$ sudo ufw show listening
```

Ответ:

```
tcp:
  10050 [redacted] (zabbix_agent2)
  [redacted] * (sshd)
   [ 1] allow OpenSSH

  [redacted] * (rsync)
   [ 2] allow [redacted]/tcp

tcp6:
  [redacted] * (sshd)
   [ 5] allow OpenSSH

  [redacted] * (rsync)
   [ 6] allow [redacted]/tcp

  [redacted] * (prometheus)
   [ 7] allow [redacted]/tcp

  [redacted] * (node_exporter)
   [ 8] allow [redacted]/tcp

udp:
  [redacted] (connmand)
  [redacted] (connmand)
  [redacted] * (avahi-daemon)
  [redacted] * (avahi-daemon)
  [redacted] * (connmand)
  [redacted] * (dhclient)
udp6:
  [redacted] * (avahi-daemon)
  [redacted] * (avahi-daemon)
```

Или недавно добавленные правила:

```
$ sudo ufw show added
```

Ответ:

```
Added user rules (see 'ufw status' for running firewall):
ufw allow OpenSSH
ufw allow [redacted]/tcp
ufw allow [redacted]/tcp
ufw allow [redacted]/tcp
```

Удаление правил.

Чтобы удалить правило **UFW**, используется команда delete.

Например, удалим ранее созданные правила для порта 80:

```
$ sudo ufw delete allow out 80/tcp
$ sudo ufw delete deny in 80/tcp
```

Журналирование в UFW.

Чтобы отлаживать работу **UFW**, могут понадобится журналы работы брандмауэра.

Для включения журналирования используется команда logging:

```
$ sudo ufw logging on
$ sudo ufw logging medium
```

Также этой командой можно изменить уровень журналирования:

low – минимальный, только заблокированные пакеты;

medium – средний, заблокированные и разрешённые пакеты;

high – высокий.

Журнал сохраняется в каталоге /var/log/ufw.

Каждая строчка журнала имеет такой синтаксис:

> [UFW действие] IN=интерфейс OUT=интерфейс SRC=ip_источника DST=ip_назначения LEN=размер_пакета TOS=0x10 PREC=0x00 TTL=64 ID=728 DF PROTO=протокол SPT=порт_источника DPT=порт назначения LEN=размер_пакета

В качестве действия приводится то, что **UFW** сделал с пакетом, например ALLOW, BLOCK или AUDIT. Благодаря анализу журнала настройка **UFW** станет гораздо проще.

Сброс настроек.

Также, если вы что-то испортили в настройках и не знаете как исправить, можно использовать команду reset для сброса настроек до состояния по умолчанию:

```
$ sudo ufw reset
```

Отключение.

Если вы хотите полностью отключить **UFW**, для этого достаточно использовать команду disable:

```
$ sudo ufw disable
```

Порты.

Настройка **UFW** для публичный портам по умолчанию:

```
$ sudo ufw status
$ sudo ufw allow OpenSSH    --// ssh
$ sudo ufw allow Samba      --// samba
$ sudo ufw allow 80/tcp     --// http
$ sudo ufw allow 443/tcp    --// https
$ sudo ufw allow 873/tcp    --// rsync
$ sudo ufw allow 3000/tcp   --// Grafana
$ sudo ufw allow 5432/tcp   --// PostgreSQL
$ sudo ufw allow 8007/tcp   --// Proxmox Backup Server
$ sudo ufw allow 8112/tcp   --// Deluge-web
$ sudo ufw allow 9090/tcp   --// Prometheus
$ sudo ufw allow 9100/tcp   --// Node Exporter
$ sudo ufw allow 10050/tcp  --// Zabbix Agent passive
$ sudo ufw allow 10051/tcp  --// Zabbix Agent active
$ sudo ufw reload
$ sudo ufw status
```

Чтобы посмотреть базовые настройки межсетевого экрана **UFW** можно использовать команду:

```
$ sudo ufw status
```

Оригиналы источников информации.

1. losst.pro «Настройка UFW Ubuntu.»