Ubuntu

- Полное руководство по работе с IPTABLES, Подробная энциклопедия с практическими примерами
- Screen инструкция
- GNU screen: авто-подстройка размеров терминала (defautoresize)
- Утилита Tmux — современный терминальный мультиплексор, позволяющий управлять несколькими терминальными окнами в одном сеансе.
- Настройка уведомлений Fail2Ban в Telegram

Полное руководство по работе с IPTABLES, Подробная энциклопедия с практическими примерами

1. Основные компоненты и принцип работы

Таблицы:

- **filter** (по умолчанию):
  - INPUT - управление входящими соединениями
  - OUTPUT - контроль исходящего трафика
  - FORWARD - обработка транзитных пакетов
- **nat**:
  - PREROUTING - преобразование перед маршрутизацией
  - POSTROUTING - преобразование после маршрутизации
- **mangle** - для специальной модификации пакетов

2. Базовые операции с IPTABLES

```
# Просмотр правил с нумерацией строк
iptables -L -n --line-numbers -v

# Очистка всех правил (осторожно!)
iptables -F && iptables -X
```

```
# Сохранение текущей конфигурации
iptables-save > /etc/iptables.conf

# Восстановление правил
iptables-restore < /etc/iptables.conf
```

3. Практические примеры использования

3.1. Базовая защита сервера

```
# Разрешаем локальный интерфейс
iptables -A INPUT -i lo -j ACCEPT

# Разрешаем установленные соединения
iptables -A INPUT -m conntrack --ctstate ESTABLISHED,RELATED -j ACCEPT

# Открываем SSH только с доверенных IP
iptables -A INPUT -p tcp -s 192.168.1.100 --dport 22 -j ACCEPT

# Защита от SYN-флуда
iptables -A INPUT -p tcp --syn -m limit --limit 1/s -j ACCEPT
```

3.2. Настройка NAT и проброса портов

```
# Включаем форвардинг
echo 1 > /proc/sys/net/ipv4/ip_forward

# Маскировка внутренней сети
iptables -t nat -A POSTROUTING -o eth0 -j MASQUERADE

# Проброс порта 80 на внутренний сервер
iptables -t nat -A PREROUTING -p tcp --dport 80 -j DNAT --to 192.168.1.10:80
iptables -A FORWARD -p tcp -d 192.168.1.10 --dport 80 -j ACCEPT
```

4. Блокировка по географическому признаку

4.1. Установка и настройка ipset

```
apt install ipset  # Для Debian/Ubuntu
yum install ipset  # Для CentOS/RHEL

# Создаем набор для хранения IP-адресов
ipset create blocked_countries hash:net

# Добавляем правило блокировки
iptables -I INPUT -m set --match-set blocked_countries src -j DROP
```

4.2. Загрузка IP-диапазонов стран

```
# Создаем скрипт /usr/local/bin/update_countries.sh:
#!/bin/bash
COUNTRIES="cn ru"  # Коды стран для блокировки
TMP_SET=temp_countries

ipset create $TMP_SET hash:net

for country in $COUNTRIES; do
  wget -O $country.zone http://www.ipdeny.com/ipblocks/data/countries/$country.zone
  for ip in $(cat $country.zone); do ipset add $TMP_SET $ip; done
  rm -f $country.zone
done

ipset swap $TMP_SET blocked_countries
ipset destroy $TMP_SET

# Делаем исполняемым и запускаем
chmod +x /usr/local/bin/update_countries.sh
/usr/local/bin/update_countries.sh
```

4.3. Автоматическое обновление

```
# Добавляем в cron еженедельное обновление
echo "0 3 * * 1 /usr/local/bin/update_countries.sh" >> /etc/crontab
```

5. Мониторинг и отладка

```
# Просмотр статистики по правилам
iptables -L -n -v

# Логирование блокируемых пакетов
iptables -A INPUT -j LOG --log-prefix "BLOCKED: " --log-level 4

# Просмотр логов в реальном времени
tail -f /var/log/syslog | grep BLOCKED
```

> **Важные рекомендации**:
>
> 1. Всегда тестируйте правила на тестовом сервере
> 2. Имейте альтернативный доступ на случай ошибок
> 3. Регулярно обновляйте списки IP-адресов
> 4. Для сложных конфигураций используйте скрипты

Screen инструкция

GNU Screen — терминальный мультиплексор для Unix-подобных систем.
Позволяет управлять несколькими терминальными окнами в рамках одной сессии, разделять экран, отсоединяться от сессии с последующим восстановлением, а также совместно использовать одну сессию несколькими пользователями.

Установка

Установка в Debian/Ubuntu:

```
sudo apt install screen -y
```

Проверка установки:

```
screen --version
```

Основные возможности

- Запуск нескольких терминальных окон в одной сессии
- Переключение между окнами
- Разделение экрана по горизонтали и вертикали
- Отсоединение от сессии с продолжением выполнения процессов
- Повторное подключение к сессии
- Совместный доступ к одной сессии
- Переименование окон и сессий
- Логирование вывода окон
- Поддержка конфигурации через `~/.screenrc`

Управление окнами

Все команды выполняются после **Ctrl + a**, затем нужная клавиша или команда.

^ — означает клавишу **Ctrl**

^a c — создать новое окно
^a n — перейти к следующему окну
^a p — перейти к предыдущему окну
^a " — список всех окон
^a 0..9 — переход к окну по номеру
^a A — переименовать текущее окно
^a K — закрыть текущее окно
^a d — отсоединиться от сессии
^a ? — справка по клавишам

Примеры:

- Создать новое окно:

```
Ctrl + a, затем c
```

- Переход к окну №2:

```
Ctrl + a, затем 2
```

- Список всех окон:

```
Ctrl + a, затем "
```

Работа с сессиями

Создать новую сессию с именем:

```
screen -S mysession
```

Посмотреть список всех сессий:

```
screen -ls
```

Присоединиться к существующей сессии:

```
screen -r mysession
```

Присоединиться к уже активной (совместно):

```
screen -x mysession
```

Отсоединиться (detach), не завершая работу программ:

```
Ctrl + a, затем d
```

Разделение экрана

Горизонтальное разделение:

```
Ctrl + a, затем S
```

Вертикальное разделение:

```
Ctrl + a, затем |
```

Переключение между областями:

```
Ctrl + a, затем Tab
```

Удалить текущую область:

```
Ctrl + a, затем X
```

Создать окно в активной области:

```
Ctrl + a, затем c
```

Логирование

Включить логирование текущего окна:

```
Ctrl + a, затем H
```

Лог сохраняется как `screenlog.0` в текущей директории.
Чтобы выключить логирование — нажать **Ctrl + a, затем H** повторно.

Пример:

```
tail -f /var/log/syslog
# Нажмите Ctrl + a, затем H для записи вывода tail в файл screenlog.0
```

Примеры использования

Пример 1. Запуск долгого процесса и отсоединение

```
screen -S backup
rsync -avz /data/ user@remote:/backup/
# Нажмите Ctrl + a, затем d
```

Позже подключиться:

```
screen -r backup
```

Пример 2. Разделение экрана и мониторинг

```
screen -S monitor
```

Внутри сессии:

- Ctrl + a, затем S — разделить горизонтально
- Ctrl + a, затем Tab — перейти вниз
- Ctrl + a, затем c — открыть окно
- В первом окне:

  ```
  htop
  ```

- Во втором окне:

  ```
  tail -f /var/log/syslog
  ```

Пример 3. Совместный доступ
Администратор запускает сессию:

```
screen -S shared
```

Другой пользователь подключается:

```
screen -x shared
```

(Понадобится одинаковый пользователь или `chmod` на `/tmp/screens/S-user`)

Конфигурация ~/.screenrc

Пример минимальной конфигурации:

```
# Изменение escape-клавиши
escape ^Zz

# Постоянный статус
caption always "%{= kG}%-w%{= kW}%n %t%{-}%+w"

# Автозапуск окон с командами
screen -t Shell 0
screen -t Logs tail -f /var/log/syslog
```

Сравнение с альтернативами

Инструмент	Особенности
screen	Классический, надёжен, прост
tmux	Более современный, скриптуемый, поддерживает плагины
byobu	Оболочка над screen/tmux с расширенными статусами

Полезные ссылки

- Официальное руководство GNU Screen
- Arch Wiki про Screen
- man screen

Заключение

GNU Screen — простой, надёжный и мощный инструмент для администраторов и разработчиков.
Особенно полезен при удалённой работе, автоматизации задач и поддержке долгих процессов.

Прекрасно сочетается с другими Unix-инструментами и обеспечивает стабильную работу даже при обрывах соединения.

GNU screen: авто-подстройка размеров терминала (defautoresize)

Что это даёт

Иногда при изменении размера окна терминала screen не подхватывает новые размеры — и «рамка» или приложения внутри выглядят обрезанными.
Решение — включить автоматическую подстройку:

```
defautoresize on
```

Шаг 1. Создайте файл ~/.screenrc

Файл `~/.screenrc` — это пользовательская конфигурация screen.
Если вы «не можете найти файл» — это нормально: по умолчанию он часто не существует.

Создать файл:

```
touch ~/.screenrc
```

Открыть для редактирования (пример с nano):

```
nano ~/.screenrc
```

Шаг 2. Добавьте минимальный конфиг

Рекомендуемый минимальный конфиг:

```
# Scrollback
defscrollback 5000

# Нормальная работа с xterm-совместимыми терминалами
termcapinfo xterm* ti@:te@
altscreen on

# Автоматическая подстройка размера
defautoresize on
```

После вставки сохраните файл и выйдите из редактора. В nano: **Ctrl+O → Enter → Ctrl+X**

Шаг 3. Перезапустите screen

screen читает `.screenrc` только при запуске, поэтому текущую сессию нужно перезапустить.

Вариант A — выйти и запустить заново:

```
exit
```

Вариант B — посмотреть сессии и завершить текущую:

```
screen -ls
screen -X quit
```

Команда **screen -X quit** завершит текущую сессию screen. Убедитесь, что все важные процессы сохранены.

Быстрая ручная подстройка

Если нужно подогнать размер вручную:

Внутри screen нажмите:
`Ctrl+a`, затем `:` и введите:

```
resize
```

или

```
fit
```

Проверка версии

Если автоматическая подстройка не работает — проверьте версию screen:

```
screen --version
```

В очень старых версиях GNU screen опция **defautoresize** может отсутствовать. В этом случае используйте ручную команду **resize**.

FAQ

Где находится глобальный конфиг?
Обычно `/etc/screenrc`, но для пользователя нужен именно `~/.screenrc`.

Работает ли по SSH?
Да, если терминал корректно передаёт изменение размеров (обычно так и есть).

Как понять, screen это или tmux?

- `tmux` — префикс `Ctrl+b`
- `screen` — префикс `Ctrl+a`

Утилита Tmux — современный терминальный мультиплексор, позволяющий управлять несколькими терминальными окнами в одном сеансе.

Tmux — современный терминальный мультиплексор, позволяющий управлять несколькими терминальными окнами в одном сеансе.
Является более гибкой альтернативой GNU Screen и активно используется администраторами, разработчиками и DevOps-специалистами.

Установка

В Debian/Ubuntu:

```
sudo apt install tmux -y
```

Проверить версию:

```
tmux -V
```

Основные понятия

- **Сессия (session)** — контейнер для окон и панелей.
- **Окно (window)** — виртуальное окно терминала, внутри сессии.
- **Панель (pane)** — часть окна, полученная путём его разделения.
- **Команда (command)** — команда tmux, выполняемая внутри сессии.

Старт работы

Запуск новой сессии:

```
tmux
```

Или с указанием имени:

```
tmux new -s mysession
```

Отсоединиться от сессии (detach):

```
Ctrl + b, затем d
```

Список активных сессий:

```
tmux ls
```

Присоединиться к сессии:

```
tmux attach -t mysession
```

Удалить завершённую сессию:

```
tmux kill-session -t mysession
```

Основные команды и сочетания клавиш

Префиксная клавиша по умолчанию — **Ctrl + b**
(после неё нажимайте нужную команду)

^ означает **Ctrl**

^b c — создать новое окно
^b n — следующее окно
^b p — предыдущее окно
^b , — переименовать окно
^b w — список окон

^b % — разделить окно вертикально
^b " — разделить окно горизонтально
^b o — переключиться между панелями
^b x — закрыть текущую панель
^b z — развернуть панель на весь экран

^b [— режим копирования (copy mode)
^b] — вставить скопированное

^b d — отсоединиться
^b ? — справка по командам

Примеры использования

Пример 1. Управление сессией

```
tmux new -s dev
# Внутри сессии...
# Создайте окно: Ctrl + b, затем c
# Переключение: Ctrl + b, затем n или p
# Отсоединение: Ctrl + b, затем d
# Подключение снова:
tmux attach -t dev
```

Пример 2. Разделение экрана

```
tmux new -s split
# Разделить вертикально: Ctrl + b, затем %
# Разделить горизонтально: Ctrl + b, затем "
# Переключиться между панелями: Ctrl + b, затем o
# Закрыть панель: Ctrl + b, затем x
```

Пример 3. Одновременный мониторинг
В одной панели:

```
htop
```

В другой панели:

```
tail -f /var/log/syslog
```

Пример 4. Сессия для фонового выполнения

```
tmux new -s longtask
./heavy_task.sh
# Отсоединиться Ctrl + b, затем d
# Позже подключиться:
tmux attach -t longtask
```

Режим копирования (Copy Mode)

Войти в режим копирования:

```
Ctrl + b, затем [
```

Управление:

- Стрелки — перемещение
- Space — начать выделение
- Enter — скопировать выделенное
- q — выйти из режима

Вставить скопированное:

```
Ctrl + b, затем ]
```

Конфигурация ~/.tmux.conf

Настройка в файле `~/.tmux.conf`
Применяется автоматически при запуске tmux.

Пример:

```
# Установить другую клавишу-префикс (Ctrl + a)
unbind C-b
set -g prefix C-a
bind C-a send-prefix

# Включить мышь для переключения окон и панелей
set -g mouse on

# Установить цветовую тему для строки статуса
set -g status-bg colour235
set -g status-fg white
set -g status-left " #S "
set -g status-right " %H:%M | %d-%b-%y "
```

Применить изменения (внутри tmux):

```
Ctrl + b, затем :source-file ~/.tmux.conf
```

Полезные команды tmux

- Список всех команд:

  ```
  Ctrl + b, затем ?
  ```

- Выполнить команду tmux внутри сессии:

  ```
  Ctrl + b, затем :<команда>
  ```

- Завершить все сессии:

  ```
  tmux kill-server
  ```

Сравнение с GNU Screen

Инструмент	Особенности
screen	Классический, стабильный, поддержка старых систем
tmux	Гибкий, поддержка скриптов, плагинов, конфигурация через файл

Плагины для tmux

Управление плагинами осуществляется через Tmux Plugin Manager (TPM)

Пример установки TPM:

```
git clone https://github.com/tmux-plugins/tpm ~/.tmux/plugins/tpm
```

В `~/.tmux.conf` добавить:

```
set -g @plugin 'tmux-plugins/tpm'
set -g @plugin 'tmux-plugins/tmux-sensible'
run '~/.tmux/plugins/tpm/tpm'
```

Внутри tmux установить плагины:

```
Ctrl + b, затем I
```

Полезные ссылки

- Официальный Wiki
- tmux man page
- Tmux Plugin Manager

Заключение

tmux — современный, гибкий мультиплексор терминала, обладающий широкими возможностями настройки и расширения.
Подходит как для повседневной работы, так и для автоматизации, работы на сервере и совместного доступа.
tmux активно развивается и имеет широкую поддержку в профессиональных сообществах.

Настройка уведомлений Fail2Ban в Telegram

Создание скрипта для уведомлений

Создаем файл скрипта:

```
mcedit /etc/fail2ban/scripts/fail2ban-telegram.sh
```

Содержимое скрипта:

```
#!/bin/bash

TOKEN="ТВОЙ_ТОКЕН_БОТА"
CHAT_ID="ТВОЙ_CHAT_ID"
TEXT="$1"

curl -s -X POST "https://api.telegram.org/bot${TOKEN}/sendMessage" \
     -d chat_id="${CHAT_ID}" \
     -d text="${TEXT}" \
     -d parse_mode="Markdown"
```

Делаем скрипт исполняемым:

```
chmod +x /etc/fail2ban/scripts/fail2ban-telegram.sh
```

Настройка Fail2Ban

Копируем стандартный action:

```
cp /etc/fail2ban/action.d/mail.conf /etc/fail2ban/action.d/telegram.conf
```

Редактируем конфигурацию:

```
mcedit /etc/fail2ban/action.d/telegram.conf
```

Содержимое файла:

```
[Definition]
actionstart =
actionstop =
actioncheck =
actionban = /etc/fail2ban/scripts/fail2ban-telegram.sh "□□[Fail2Ban] Бан: <ip> из тюрьмы
<name>"
actionunban = /etc/fail2ban/scripts/fail2ban-telegram.sh "□ [Fail2Ban] Разбан: <ip> из тюрьмы
<name>"

[Init]
```

Подключение в jail.local

```
mcedit /etc/fail2ban/jail.d/defaults-debian.conf
```

Пример конфигурации:

```
[DEFAULT]
maxretry = 2
findtime = 480
bantime = 720
action = iptables-allports
    telegram
ignoreip = 127.0.0.1/8 192.168.0.0/24
```

Применение настроек

```
sudo systemctl restart fail2ban
```

Тестирование

```
sudo fail2ban-client set sshd banip 1.2.3.4
```

Улучшенная версия с геолокацией

Установка зависимостей:

```
sudo apt install geoip-bin whois jq curl -y
```

Создание улучшенного скрипта:

```
sudo mcedit /etc/fail2ban/scripts/fail2ban-telegram-geo.sh
```

Содержимое скрипта:

```
#!/bin/bash

TOKEN="ТВОЙ_ТОКЕН_БОТА"
CHAT_ID="ТВОЙ_CHAT_ID"
IP="$1"
JAIL="$2"
REASON="$3"
DATE="$(date '+%Y-%m-%d %H:%M:%S %Z')"

# Геолокация через ipinfo.io
INFO=$(curl -s "https://ipinfo.io/${IP}/json")

COUNTRY=$(echo "$INFO" | jq -r '.country // "Unknown"')
CITY=$(echo "$INFO" | jq -r '.city // "Unknown"')
ORG=$(echo "$INFO" | jq -r '.org // "Unknown"')

# Формируем сообщение
TEXT="□□*Fail2Ban ВНИМАНИЕ*
□□Дата: ${DATE}
□□Тюрьма: ${JAIL}
□ IP: ${IP}
□□Страна: ${COUNTRY}
□□Город: ${CITY}
□□Организация: ${ORG}
□□Причина: ${REASON}"

# Отправка в Telegram
curl -s -X POST "https://api.telegram.org/bot${TOKEN}/sendMessage" \
     -d chat_id="${CHAT_ID}" \
     -d text="${TEXT}" \
     -d parse_mode="Markdown"
```

Делаем исполняемым:

```
sudo chmod +x /etc/fail2ban/scripts/fail2ban-telegram-geo.sh
```

Создаем action для геолокации:

```
sudo mcedit /etc/fail2ban/action.d/telegram-geo.conf
```

Содержимое:

```
[Definition]
actionstart =
actionstop =
actioncheck =
actionban = /etc/fail2ban/scripts/fail2ban-telegram-geo.sh "<ip>" "<name>" "Превышено число неудачных попыток входа"
actionunban = /etc/fail2ban/scripts/fail2ban-telegram-geo.sh "<ip>" "<name>" "Разбан"

[Init]
```

Обновляем конфигурацию jail:

```
mcedit /etc/fail2ban/jail.d/defaults-debian.conf
```

```
[DEFAULT]
maxretry = 2
findtime = 480
bantime = 720
action = iptables-allports
    telegram-geo
ignoreip = 127.0.0.1/8 192.168.0.0/24
```

Применяем изменения:

```
sudo systemctl restart fail2ban
sudo fail2ban-client set sshd banip 1.2.3.4
sudo fail2ban-client set sshd unbanip 1.2.3.4
```

Пример уведомления:

```
□□Fail2Ban Alert
□□Дата: 2025-08-12 14:23:45 UTC
```

□□Тюрьма: sshd

□ IP: 1.2.3.4

□□Страна: US

□□Город: Los Angeles

□□Организация: AS12345 Example ISP

□□Причина: Превышено число неудачных попыток входа