**Иркутский национальный исследовательский технический университет**

**Институт ИТиАД**

**Кафедра автоматизированных систем**

**Методические указания**

к лабораторным работам по курсу   
«**Администрирование информационных систем**»

Редакция 1.2023

**Лабораторная работа №3 (2 семестр)**

**«Средства безопасности в Ubuntu Server»**

(6 часов)

**Цель работы:** Получить начальные навыки практического администрирования программных средств безопасноти в операционной системе UbuntuServer, изучить управление пользователями, Firewall, AppArmor.

**Требования к отчетным документам:** Отчетный документ представляется в печатном или электронном виде, в форматах pdf, doc(x).

Отчет должен содержать следующие главы:

1. Введение.
2. Теоретическая часть. Основные меры и средства для обеспечения защиты сервера  и сети от любого набора потенциальных угроз безопасности.
3. Практическая часть (привести использованные команды и скриншоты).

3.1. Управление пользователем. Определение суперпользователя, добавление и удаление пользователя. Безопасность профиля пользователя. Политика паролей.

3.2. Безопасность консоли.

3.3. Firewall. Настройка ufw. Интеграция приложений в ufw. Маскировка IP. Журналы.

3.4 AppArmor. Использование AppArmor. Профили.

1. Заключение (полученные результаты).

Объем отчетного документа 7 - 10 страниц или сколько получится.

**Задание:**

1. Изучить информацию по ссылке

https://help.ubuntu.ru/wiki/%D1%80%D1%83%D0%BA%D0%BE%D0%B2%D0%BE%D0%B4%D1%81%D1%82%D0%B2%D0%BE\_%D0%BF%D0%BE\_ubuntu\_server/%D0%B1%D0%B5%D0%B7%D0%BE%D0%BF%D0%B0%D1%81%D0%BD%D0%BE%D1%81%D1%82%D1%8C

<http://help.ubuntu.ru/wiki/руководство_по_ubuntu_server>/безопасность

2. Протестировать средства управления пользователями.

3. Изучить средства обеспечения безопасности консоли.

4. Настроить Firewall.

5. Настроить AppArmor.

6. Написать отчет.

## Безопасность

Безопасность всегда должна учитываться когда устанавливаются, разворачиваются и используются любые типы компьютерных систем. Хотя свежеустановленная Ubuntu сравнительно безопасна для немедленного использования в интернете, важно иметь сбалансированное понимание состояния безопасности ваших систем, исходя из их использования после развертывания.

Эта глава предоставляет обзор статей, связанных с безопасностью, применения их к Ubuntu 12.04 LTS Server Edition и схемы простейших мер, которые вы можете использовать для защиты вашего сервера и сети от любого набора потенциальных угроз безопасности.

1. [Управление пользователями](https://help.ubuntu.ru/wiki/%D1%80%D1%83%D0%BA%D0%BE%D0%B2%D0%BE%D0%B4%D1%81%D1%82%D0%B2%D0%BE_%D0%BF%D0%BE_ubuntu_server/%D0%B1%D0%B5%D0%B7%D0%BE%D0%BF%D0%B0%D1%81%D0%BD%D0%BE%D1%81%D1%82%D1%8C/user_management)
2. [Безопасность консоли](https://help.ubuntu.ru/wiki/%D1%80%D1%83%D0%BA%D0%BE%D0%B2%D0%BE%D0%B4%D1%81%D1%82%D0%B2%D0%BE_%D0%BF%D0%BE_ubuntu_server/%D0%B1%D0%B5%D0%B7%D0%BE%D0%BF%D0%B0%D1%81%D0%BD%D0%BE%D1%81%D1%82%D1%8C/console_security)
3. [Firewall](https://help.ubuntu.ru/wiki/%D1%80%D1%83%D0%BA%D0%BE%D0%B2%D0%BE%D0%B4%D1%81%D1%82%D0%B2%D0%BE_%D0%BF%D0%BE_ubuntu_server/%D0%B1%D0%B5%D0%B7%D0%BE%D0%BF%D0%B0%D1%81%D0%BD%D0%BE%D1%81%D1%82%D1%8C/firewall)
4. [AppArmor](https://help.ubuntu.ru/wiki/%D1%80%D1%83%D0%BA%D0%BE%D0%B2%D0%BE%D0%B4%D1%81%D1%82%D0%B2%D0%BE_%D0%BF%D0%BE_ubuntu_server/%D0%B1%D0%B5%D0%B7%D0%BE%D0%BF%D0%B0%D1%81%D0%BD%D0%BE%D1%81%D1%82%D1%8C/apparmor)

## 1 Управление пользователями

Управление пользователями является важной частью безопасности системы. Неэффективные пользователи и управление привилегиями часто приводят множество систем к компрометации. Поэтому важно чтобы вы понимали как защитить ваш сервер с помощью простых и эффективных методик управления пользовательскими учетными записями.

### Где суперпользователь?

Разработчики Ubuntu приняли сознательное решение заблокировать административную корневую учетную запись (root) по умолчанию во всех установках Ubuntu. Это не означает, что учетная запись root удалена или к ней нет доступа. Ей просто присвоен пароль, который не совпадает ни с одним возможным шифрованным значением, соответственно, ее невозможно использовать для входа напрямую.

Вместо этого поощряется применение пользователями инструмента с именем **sudo** для переноса административных обязанностей. **Sudo** позволяет авторизованным пользователям временно повышать их привилегии, используя их собственный пароль вместо знания пароля, присвоенного суперпользователю. Эта простая и к тому же эффективная методика обеспечивает ответственность для всех действий пользователей и дает административный раздельный контроль над тем, какие действия может выполнять пользователь с указанными привилегиями.

1. Если по какой-то причине вы хотите разрешить учетную запись суперпользователя, просто установите ей пароль:

sudo passwd

Sudo запросит ваш пароль, а затем предложит установить новый пароль для root как показано ниже:

[sudo] password for username: (вводите свой собственный пароль)

Enter new UNIX password: (вводите новый пароль суперпользователя)

Retype new UNIX password: (повторяете новый пароль суперпользователя)

passwd: password updated successfully

2. Для блокирования учетной записи root используйте следующий синтаксис passwd:

sudo passwd -l root

3. Вы можете прочитать больше по **sudo** вызвав ее man страницу:

man sudo

По умолчанию изначальный пользователь, созданный установщиком Ubuntu является членом группы "admin", которая добавлена в файл /etc/sudoers как авторизованные sudo пользователи. Если вы желаете разрешить другой учетной записи полный доступ суперпользователя через **sudo**, просто добавьте ее в группу **admin**.

### Добавление и удаление пользователей

Процесс управления локальными пользователями и группами простой и мало отличается от большинства других операционных систем GNU/Linux. Ubuntu и другие дистрибутивы на основе Debian поощряют использование пакета "adduser" для управления учетными записями.

1. Для добавления учетной записи пользователя используйте следующий синтаксис и следуйте подсказкам для указания пароля и опознавательных характеристик таких как полное имя, телефон и пр.:

sudo adduser username

2. Для удаления пользователя и его первичной группы используйте следующий синтаксис:

sudo deluser username

Удаление пользователя не удаляет связанный с ним домашний каталог. Оставлено на ваше усмотрение хотите ли вы удалить каталог вручную или оставите его в соответствии с вашими политиками хранения.

Помните, что любой пользователь, добавленный позднее с теми же UID/GID, как и предыдущий, получит доступ к этому каталогу если вы не предпримете необходимых мер предосторожности.

Вы можете захотеть изменить эти значения UID/GID каталога на что-то более подходящее, как, например, значения суперпользователя и, возможно, переместить каталог для предотвращения будущих конфликтов:

sudo chown -R root:root /home/username/

sudo mkdir /home/archived\_users/

sudo mv /home/username /home/archived\_users/

3. Для временного блокирования или разблокирования используйте следующий синтаксис:

sudo passwd -l username

sudo passwd -u username

4. Для добавления или удаления персональной группы используйте, соответственно, следующий синтаксис:

sudo addgroup groupname

sudo delgroup groupname

5. Для добавления пользователя в группу, используйте:

sudo adduser username groupname

### Безопасность профиля пользователя

Когда создается новый пользователь, утилита adduser создает, соответственно, новый именной каталог **/home/username**. Профиль по умолчанию формируется по содержимому, находящемуся в каталоге /etc/skel, который включает все основы для формирования профилей.

Если ваш сервер является домашним для множества пользователей, вы должны уделять пристальное внимание правам доступа на пользовательские домашние каталоги для поддержания конфиденциальности. По умолчанию пользовательские домашние каталоги создаются с правами чтения/выполнения для всех. Это означает, что все пользователи просматривать и получать доступ к содержимому других домашних каталогов. Это может не подходить для вашего окружения.

1. Для проверки прав доступа на домашние каталоги существующих пользователей используйте такой синтаксис:

ls -ld /home/username

Следующий вывод показывает, что каталог /home/username имеет доступ на чтение для всех:

drwxr-xr-x 2 username username 4096 2007-10-02 20:03 username

2. Вы можете удалить права чтения для всех, используя следующий синтаксис:

sudo chmod 0750 /home/username

Некоторые склоняются к тенденции использовать опцию рекурсии (-R) без разбора, которая изменяет все дочерние каталоги и файлы, хотя это необязательно и может приводить к иным нежелательным последствиям. Родительский каталог сам по себе запретит неавторизованный доступ к любому своему содержимому.

Более эффективный подход к данному вопросу будет в изменении глобальных прав доступа по умолчанию для **adduser** при создании домашних каталогов. Просто отредактируйте файл /etc/adduser.conf, изменив переменную DIR\_MODE на что-то более подходящее, после чего все новые домашние каталоги будут получать корректные права доступа.

DIR\_MODE=0750

3. После исправления прав доступа к каталогам, используя любую из ранее упоминавшихся методик, проверьте результаты используя следующую команду:

ls -ld /home/username

Результат ниже показывет, что права на чтение для всех удалены:

drwxr-x--- 2 username username 4096 2007-10-02 20:03 username

### Политика паролей

Строгая политика паролей - один из наиболее важных аспектов вашего подхода к безопасности. Много удачных прорывов безопасности использовали для атак простейший взлом (brute force) и подбор паролей по словарю против слабых паролей. Если вы намереваетесь использовать любую форму удаленного доступа с использованием вашей локальной системы паролей, убедитесь, что вы назначили адекватные минимальные требования к паролю, максимальное время жизни пароля и часто проверяете вашу систему аутентификации.

#### Минимальная длина пароля

По умолчанию Ubuntu требует минимальную длину пароля в 6 символов, также как и некоторые основные проверки на разброс значений. Эти параметры управляются файлом /etc/pam.d/common-password и приведены ниже:

password [success=2 default=ignore] pam\_unix.so obscure sha512

Если вы хотите установить минимальную длину в 8 символов, измените соответствующую переменную на min=8. Изменения приведены ниже:

password [success=2 default=ignore] pam\_unix.so obscure sha512 min=8

Базовые проверки на качество и минимальную длину пароля не применяются к администратору, использующего команды уровня sudo для настройки нового пользователя.

#### Время жизни пароля

При создании учетных записей пользователей вы должны создать политику минимального и максимального времени жизни пароля чтобы заставлять пользователей менять их пароли по истечении определенного времени.

1. Для простого просмотра текущего статуса учетной записи пользователя используйте следующий синтаксис:

sudo chage -l username

Вывод, приведенный ниже, показывает интересные факты об учетной записи пользователя, а именно что нет никаких примененных политик:

Last password change : Jan 20, 2008

Password expires : never

Password inactive : never

Account expires : never

Minimum number of days between password change : 0

Maximum number of days between password change : 99999

Number of days of warning before password expires : 7

2. Для установки этих значений просто используйте следующую команду и следуйте интерактивным подсказкам:

sudo chage username

Далее также пример того, как можно вручную изменить явную дату истечения действия пароля (-E) на 01/31/2008 (американский вариант даты - прим. пер.), минимальный срок действия пароля (-m) на 5 дней, максимальный срок действия (-M) на 90 дней, период бездействия (-I) на 5 дней после истечения срока пароля и период предупреждений (-W) на 14 дней до истечения срока пароля.

sudo chage -E 01/31/2011 -m 5 -M 90 -I 30 -W 14 username

3. Для проверки изменений используйте ту же команду, что и упоминалась выше:

sudo chage -l username

Вывод команды ниже показывает новые политики, которые применяются к учетной записи:

Last password change : Jan 20, 2008

Password expires : Apr 19, 2008

Password inactive : May 19, 2008

Account expires : Jan 31, 2008

Minimum number of days between password change : 5

Maximum number of days between password change : 90

Number of days of warning before password expires : 14

### Иные соображения безопасности

Многие приложения используют альтернативные механизмы аутентификации, которые могут быть запросто пропущены даже опытными системными администраторами. Поэтому важно понимать и контролировать как пользователи авторизуются и получают доступ к сервисам и приложениям на вашем сервере.

#### Доступ по SSH заблокированными пользователями

Обычное отключение/блокирование не исключает удаленного подключения пользователя к серверу, если ему предварительно была установлена аутентификация по открытому ключу RSA. Такие пользователи будут получать доступ к консольной оболочке (shell) на сервере без необходимости ввода какого-либо пароля. Не забывайте проверять пользовательские домашние каталоги на файлы, которые позволяют подобный тип авторизации по SSH, например, /home/username/.ssh/authorized\_keys.

Удаление или переименование каталога .ssh/ в домашнем каталоге пользователя предотвратит дальнейшую способность аутентификации по SSH.

Убедитесь что проверили любые установленные SSH соединения заблокированных пользователей, поскольку могут остаться входящие или исходящие соединения. Убейте все, которые найдете.

Ограничьте SSH доступ только для учетных записей пользователей, которым они требуются. Например, вы можете создать группу с названием "sshlogin" и добавить имя группы в качестве значения для переменной AllowGroups, находящейся в файле /etc/ssh/sshd\_config.

AllowGroups sshlogin

Затем добавьте ваших пользователей, которым разрешен SSH доступ, в группу "sshlogin" и перестартуйте SSH сервис.

sudo adduser username sshlogin

sudo service ssh restart

#### Аутентификация по внешней базе данных

Большинство корпоративных сетей требуют централизованной аутентификации и контроля доступа для всех системных ресурсов. Если вы настроили свой сервер для аутентификации пользователей по внешней базе данных, убедитесь, что вы отключаете учетные записи как внешние, так и локальные, таким образом вы будете уверены что откат на локальную аутентификацию невозможен.

## 2 Безопасность консоли

Как любой другой барьер безопасности, который вы выстраиваете для защиты вашего сервера, требуется довольно жестко защититься от невообразимого ущерба, который может возникнуть от физического доступа кого-то к вашему оборудованию, например, воровства жестких дисков, сбою по питанию, отказа в обслуживании и т.п. Поэтому безопасность консоли стоит рассматривать просто как еще один компонент вашей общей стратегии физической безопасности. Блокируемая "ширма" (screen door) может защитить от случайного криминала и очень сильно замедлить активное воздействие, поэтому очень желательно соблюдать простейшие предосторожности по отношению к безопасности консоли.

Дальнейшие инструкции помогут защитить ваш сервер от проблем, которые могут привести к очень серьезным последствиям.

### Отключение Ctrl+Alt+Delete

Для начала и в первую очередь, любой, кто получит физический доступ к клавиатуре может просто нажать комбинацию клавиш Ctrl+Alt+Delete для перезагрузки сервера без необходимости входить в систему. Несомненно, этот кто-то может просто отключить питание, но вы все равно должны заблокировать эту комбинацию клавиш на рабочем сервере. Это потребует от атакующего применить больше активных действий для перезагрузки сервера и в то же время защитит от случайных перезагрузок.

1. Для отключения перезагрузки по нажатию комбинации Ctrl+Alt+Delete, закомментируйте следующую строку в файле /etc/init/control-alt-delete.conf.

#exec shutdown -r now "Control-Alt-Delete pressed"

## 3 Firewall

### Введение

Ядро линукс включает подсистему **Netfilter** (сетевой фильтр), который используется для манипулирования или решения судьбы сетевого трафика передаваемого в или через ваш сервер. Все современные решения линукс по сетевой защите используют эту систему пакетной фильтрации.

Система пакетной фильтрации на уровне ядра была бы малоиспользуема администраторами без пользовательского интерфейса для ее управления. Для этого предназначен iptables. Когда пакет попадает на ваш сервер, он передается подсистеме Netfilter для одобрения, изменения или отказа на основе правил, которые она получает от интерфейса пользователя через iptables. Таким образом iptables - это все, что вам нужно для управления вашей сетевой защитой, если вы хорошо с ним знакомы, однако множество внешних интерфейсов доступны для упрощения этой задачи.

### ufw - простой Firewall

Инструмент для настройки сетевой защиты Ubuntu по умолчанию - это ufw. Он разработан для легкой настройки iptables и предоставляет дружественный способ создания сетевой защиты для IPv4 и IPv6.

По умолчанию изначально ufw выключен. Со страницы man руководства ufw:

«ufw не предназначен для обеспечения полной функциональности брандмауэра через свой командный интерфейс, но он предоставляет легкий способ добавления или удаления простых правил. Сейчас в большинстве случаев он используется для централизованных брандмауэров.»

Далее следуют несколько примеров использования ufw:

1. Для начала требуется разрешить ufw. Наберите в терминале:

sudo ufw enable

2. Открыть порт (в данном примере SSH):

sudo ufw allow 22

3. Правила могут быть добавлены с использованием нумерованного формата:

sudo ufw insert 1 allow 80

4. Подобным образом можно закрыть открытый порт:

sudo ufw deny 22

5. Для удаления правила используйте delete:

sudo ufw delete deny 22

6. Также можно разрешить доступ к порту с определенных компьютеров или сетей. Следующий пример разрешает на этом компьютере доступ по SSH с адреса 192.168.0.2 на любой IP адрес:

sudo ufw allow proto tcp from 192.168.0.2 to any port 22

Замените 192.168.0.2 на 192.168.0.0/24 чтобы разрешить доступ по SSH для всей подсети.

7. Добавление опции **–dry-run** команде **ufw** выведет список правил, но не применит их. Например, далее показано, что было бы применено, если открыть HTTP порт:

sudo ufw --dry-run allow http

\*filter

:ufw-user-input - [0:0]

:ufw-user-output - [0:0]

:ufw-user-forward - [0:0]

:ufw-user-limit - [0:0]

:ufw-user-limit-accept - [0:0]

### RULES ###

### tuple ### allow tcp 80 0.0.0.0/0 any 0.0.0.0/0

-A ufw-user-input -p tcp --dport 80 -j ACCEPT

### END RULES ###

-A ufw-user-input -j RETURN

-A ufw-user-output -j RETURN

-A ufw-user-forward -j RETURN

-A ufw-user-limit -m limit --limit 3/minute -j LOG --log-prefix "[UFW LIMIT]: "

-A ufw-user-limit -j REJECT

-A ufw-user-limit-accept -j ACCEPT

COMMIT

Rules updated

8. **ufw** можно выключить командой:

sudo ufw disable

9. Чтобы посмотреть статус сетевой защиты:

sudo ufw status

10. Для более полного отображения информации введите:

sudo ufw status verbose

11. Для отображения в виде формата **numbered**:

sudo ufw status numbered

Если порт, который вы хотите открыть или закрыть определен в файле /etc/services, вы можете использовать имя порта вместо его номера. В приведенных выше примерах можете заменить **22** на **ssh**.

Это лишь краткое описание использования ufw. Пожалуйста обратитесь к странице man ufw для более подробной информации.

#### Интеграция Приложений в ufw

Приложения, которые открывают порты, можно включать в профили ufw, которые детализируют какие порты необходимы этому приложению для корректной работы. Профили содержатся в /etc/ufw/applications.d, и могут быть отредактированы, если порты по умолчанию были изменены.

1. Чтобы посмотреть для каких приложений установлен профиль введите следующую команду в терминале:

sudo ufw app list

2. Аналогично, разрешить трафик по порту, используя профиль приложения, можно следующей командой:

sudo ufw allow Samba

3. Также доступен расширенный синтаксис:

ufw allow from 192.168.0.0/24 to any app Samba

Замените **Samba** и **192.168.0.0/24** на используемый вами профиль приложения и IP диапазон вашей сети.

Нет необходимости в определении протокола, поскольку эта информация детализирована в профиле. Также обратите внимание, что имя **app** заменило номер **port**.

4. Для просмотра деталей какие порты, протоколы и пр. определены для приложения введите:

sudo ufw app info Samba

Не для всех приложений, которые требуют открытие сетевого порта, поставляется профиль ufw, но если у вас есть профиль для приложения, и вы хотите чтобы этот файл был включен в пакет приложения, зарегистрируйте ошибку о пакете на сайте Launchpad.

ubuntu-bug nameofpackage

### Маскировка IP

Цель маскировки ip-адреса состоит в том, чтобы компьютеры с частными, немаршрутизируемыми ip-адресами в вашей сети могли иметь доступ к Интернету через компьютер выполняющий маскировку. Трафик из вашей частной сети попадающий в Интернет должен быть правильно выпущен для корректного ответа, отправленного именно тому компьютеру, который послал запрос. Чтобы сделать это, ядро модифицирует заголовок каждого пакета так, чтобы ответ приходил нужному компьютеру, а не частному ip-адресу, который послал запрос, что невозможно в рамках сети Интернет. Линукс использует Трассировку Соединений (conntrack) для трассировки каждого из соединений принадлежащих соответствующим компьютерам и перенаправляет каждый возвращенный пакет. Трафик, покидает вашу частную сеть «маскируясь» таким образом, будто исходит от вашего шлюза Ubuntu. Этот процесс обозначен в документации Microsoft как Internet Connection Sharing (Общий доступ к Интернет-Соединению).

#### Маскировка ufw

Маскировка ip-адреса может быть достигнута с использованием различных правил ufw. Это возможно благодаря дополнению к ufw, которым является iptables-restore с файлами правил, расположенных в /etc/ufw/\*.rules. Эти файлы являются отличным способом для добавления правил в iptables без использования ufw, а также правил, которые более гибко взаимодействуют со шлюзом или соединением типа «мост».

Эти правила разбиты на два разных файла команд, одни должны быть выполнены ufw перед списком правил, а другие выполняются после списка правил.

1. Во-первых, в ufw должно быть активировано перенаправление. Для этого нужно изменить конфигурацию двух файлов, в /etc/default/ufw измените DEFAULT\_FORWARD\_POLICY на “ACCEPT”:

DEFAULT\_FORWARD\_POLICY="ACCEPT"

Затем отредактируйте /etc/ufw/sysctl.conf следующим образом:

net/ipv4/ip\_forward=1

Аналогично для IPv6:

net/ipv6/conf/default/forwarding=1

2. Теперь добавим правила в файл /etc/ufw/before.rules. Правила по умолчанию описаны лишь в таблице filter, а для работы маскировки нам нужно отредактировать таблицу nat. Добавьте следующие строки в начало файла конфигурации сразу после заголовка с комментарием:

# nat Table rules

\*nat

:POSTROUTING ACCEPT [0:0]

# Forward traffic from eth1 through eth0.

-A POSTROUTING -s 192.168.0.0/24 -o eth0 -j MASQUERADE

# don't delete the 'COMMIT' line or these nat table rules won't be processed

COMMIT

Комментарии не обязательны, но считается хорошим тоном документировать свою конфигурацию. Также при изменении каких-либо **rules** файлов в каталоге /etc/ufw, убедитесь, что данные строки являются последними для каждой измененной таблицы:

# don't delete the 'COMMIT' line or these rules won't be processed

COMMIT

Для каждой таблицы обязательно присутствие соответствующего ей оператора **COMMIT** (применение изменений). В данных примерах показаны только таблицы nat и filter, но вы можете также добавлять правила для таблиц raw и mangle.

В примерах выше замените **eth0**, **eth1** и **192.168.0.0/24** на соответствующие интерфейсы и IP диапазон вашей сети.

3. Наконец, выключите и повторно включите ufw для применения изменений:

sudo ufw disable && sudo ufw enable

IP маскировка теперь должна быть включена. Вы также можете добавить дополнительные правила FORWARD в /etc/ufw/before.rules. Рекомендуется чтобы эти дополнительные правила были добавлены в цепочку/секцию (chain) **ufw-before-forward**.

#### Маскировка iptables

Iptables также может быть использован для маскировки соединений.

1. Подобно ufw, первым делом нам необходимо включить форвардинг для IPv4 отредактировав следующий файл /etc/sysctl.conf и убрав комментарий со строки:

net.ipv4.ip\_forward=1

Если вы хотите включить форвардинг для IPv6 также удалите комментарий:

net.ipv6.conf.default.forwarding=1

2. Далее выполните команду sysctl для применения новых настроек в файле конфигурации:

sudo sysctl -p

3. Теперь настройка маскировки ip-адреса может быть завершена с выполнением лишь одной команды, которая может немного отличатся в зависимости от вашей конфигурации сети:

sudo iptables -t nat -A POSTROUTING -s 192.168.0.0/16 -o ppp0 -j MASQUERADE

После выполнения этой команды можно считать, что ваша частная сеть имеет адрес 192.168.0.0/16, а сетевая карта с выходом в Интернет - ppp0. Синтаксис определен следующим образом:

1. -t nat – правило применяется к таблице nat

2. -A POSTROUTING – правило добавлено (-A) в цепочку POSTROUTING

3. -s 192.168.0.0/16 – правило применено к трафику, исходящему из этого адресного пространства

4. -o ppp0 – правило применено к трафику, который необходимо перенаправить через этот интерфейс

5. -j MASQUERADE – трафик подпадающий под это правило выполняет «прыжок» («jump») (-j) в цепочку MASQUERADE для дальнейшей манипуляции с ним как описаны выше

4. Также каждая цепочка таблицы filter (цепочка, где происходит фильтрация большинства пакетов) по умолчанию имеет правило ACCEPT, но если вы настраиваете ваш firewall с целью создания шлюза, вы должны выбрать либо политику DROP либо REJECT, в этом случае ваш «замаскированный» трафик должен быть пропущен через цепочку FORWARD для срабатывания предыдущих правил:

sudo iptables -A FORWARD -s 192.168.0.0/16 -o ppp0 -j ACCEPT

sudo iptables -A FORWARD -d 192.168.0.0/16 -m state \

--state ESTABLISHED,RELATED -i ppp0 -j ACCEPT

Вышеперечисленные команды разрешают форвардинг всех соединений из локальной сети в Интернет, а весь трафик относящийся к этим соединениям возвращается к компьютерам, пославшим запрос.

Если вы хотите, чтобы маскировка работала после перезагрузки ПК, отредактируйте файл /etc/rc.local и добавьте туде вышеперечисленные команды. Например, добавьте первую команду без опций фильтрации:

iptables -t nat -A POSTROUTING -s 192.168.0.0/16 -o ppp0 -j MASQUERADE

### Журналы

Журналы firewall крайне необходимы для обнаружения атак, возникающих проблем с правилами, а также обнаружения повышенной активности в вашей сети. В настройках firewall необходимо включить журналирование для правил, которое должно быть указано перед списком правил (правил, которые решают дальнейшую судьбу пакета, такие как ACCEPT, DROP или REJECT).

Если вы используете ufw, журналирование можно включить следующей командой:

sudo ufw logging on

Чтобы выключить журналирование в ufw просто замените значение on на off.

Для iptables введите:

sudo iptables -A INPUT -m state --state NEW -p tcp --dport 80 \

-j LOG --log-prefix "NEW\_HTTP\_CONN: "

В этом случае в начале поступит запрос на порт 80 от компьютера в локальной сети, затем будет сгенерирован файл журнала в dmesg который выгдлядит примерно так (одна строка разделена на три для корректного отображения на экране):

[4304885.870000] NEW\_HTTP\_CONN: IN=lo OUT= MAC=00:00:00:00:00:00:00:00:00:00:00:00:08:00

SRC=127.0.0.1 DST=127.0.0.1 LEN=60 TOS=0x00 PREC=0x00 TTL=64 ID=58288 DF PROTO=TCP

SPT=53981 DPT=80 WINDOW=32767 RES=0x00 SYN URGP=0

Этот текст журнала также появится в файле /var/log/messages, /var/log/syslog и /var/log/kern.log. Эти настройки можно изменить отредактировав файл /etc/syslog.conf аналогично или при установке и настройке ulogd, а также используя ULOG вместо LOG. Демон ulogd это пользовательский сервер который слушает инструкции ядра для дальнейшего журналирования, в частности для firewall, и может вести журналирование даже в PostgreSQL или MySQL базы данных. Для того, чтобы легко разобраться в файлах журнала можно использовать их анализаторы, такие как logwatch, fwanalog, fwlogwatch или lire.

### Другие инструменты

Существует множество утилит для помощи в построении полноценного firewall без углубленных навыков в iptables.

Для работы с интерфейсом:

1. [fwbuilder](http://www.fwbuilder.org/" \o "http://www.fwbuilder.org/) очень мощная утилита, которая настраивается подобно проприетарным продуктам вроде Checkpoint FireWall-1.

Если вы предпочитаете утилиты с поддержкой командной строки и списком файлов конфигураций:

2. [Shorewall](http://www.shorewall.net/" \o "http://www.shorewall.net/) - мощное решение, которое поможет вам создать серьезный firewall для любого типа вашей сети.

## 4 AppArmor

**AppArmor** - это реализация Модуля безопасности линукс по управлению доступом на основе имен. AppArmor ограничивает отдельные программы набором перечисленных файлов и возможностями в соответствии с правилами Posix 1003.1e.

**AppArmor** устанавливается и загружается по умолчанию. Он использует профили приложений для определения какие файлы и права доступа требуются приложению. Некоторые пакеты устанавливают свои собственные профили, а дополнительные профили можно найти в пакете **apparmor-profiles**.

Для установки пакета **apparmor-profiles** наберите в терминале:

sudo apt-get install apparmor-profiles

Профили AppArmor имеют два режима выполнения:

1. **Фиксации/Обучения**: нарушения профиля разрешаются и сохраняются в журнале. Полезно для тестирования и разработки новых профилей
2. **Предписаний/Ограничений**: принуждает следовать политике профиля, при этом также записывает нарушения в журнал.

### Использование AppArmor

Пакет apparmor-utils содержит утилиты командной строки, которые можно использовать для изменения режима выполнения AppArmor, поиска статуса профиля, создания новых профилей и т.п.

1. **apparmor\_status** используется для просмотра текущего статуса профиля AppArmor.

sudo apparmor\_status

2. **aa-complain** переводит профиль в режим обучения (complain).

sudo aa-complain /path/to/bin

3. **aa-enforce** переводит профиль в режим ограничений (enforce).

sudo aa-enforce /path/to/bin

4. Профили AppArmor расположены в каталоге /etc/apparmor.d. Его можно использовать для управления режимом всех профилей. Введите следующую команду для перевода всех профилей в режим обучения:

sudo aa-complain /etc/apparmor.d/\*

Перевод всех профилей в режим ограничений:

sudo aa-enforce /etc/apparmor.d/\*

5. Команда **apparmor\_parser** используется для загрузки профиля в ядро. Она также может использоваться для повторной загрузки загруженного профиля при использовании опции '-r'. Для загрузки введите:

cat /etc/apparmor.d/profile.name | sudo apparmor\_parser -a

Для перезагрузки:

cat /etc/apparmor.d/profile.name | sudo apparmor\_parser -r

6. /etc/init.d/apparmor служит для перезагрузки всех профилей:

sudo /etc/init.d/apparmor reload

7. Директория /etc/apparmor.d/disable может использоваться совместно с опцией **apparmor\_parser -R** для отключения профиля.

sudo ln -s /etc/apparmor.d/profile.name /etc/apparmor.d/disable/

sudo apparmor\_parser -R /etc/apparmor.d/profile.name

Для активации отключенного профиля удалите символическую ссылку на профиль в /etc/apparmor.d/disable/. Затем загрузите профиль используя опцию '-a'.

sudo rm /etc/apparmor.d/disable/profile.name

cat /etc/apparmor.d/profile.name | sudo apparmor\_parser -a

8. AppArmor можно отключить, а модуль ядра выгрузить следующей командой:

sudo /etc/init.d/apparmor stop

sudo update-rc.d -f apparmor remove

9. Для повторной активации AppArmor введите:

sudo /etc/init.d/apparmor start

sudo update-rc.d apparmor defaults

Замените **profile.name** на имя вашего профиля, которым вы хотите управлять. Также необходимо заменить /path/to/bin/ на реальный путь выполненяемого файла. Например, для команды ping используйте /bin/ping

### Профили

Профили **AppArmor** - это простые текстовые файлы, которые расположены в /etc/apparmor.d/. Файлы профиля называются соответственно полному пути до исполняемого файла, которым они управляют, с заменой символа «/» на «.». Например /etc/apparmor.d/bin.ping - это профиль AppArmor для команды /bin/ping.

Существует два основных типа правил, используемых в профиле:

1. **Записи путей** (Path entries): которые описывают к каким файлам приложение имеет доступ в файловой системе. 2. **Записи разрешений** (Capability entries): определяют какие права ограничиваемый процесс имеет право использовать.

В качестве примера посмотрим /etc/apparmor.d/bin.ping:

#include <tunables/global>

/bin/ping flags=(complain) {

#include <abstractions/base>

#include <abstractions/consoles>

#include <abstractions/nameservice>

capability net\_raw,

capability setuid,

network inet raw,

/bin/ping mixr,

/etc/modules.conf r,

}

1. **#include <tunables/global>**: включает операторы из других файлов. Это позволяет операторам, относящимся к нескольким приложениям находится в одном общем файле.
2. **/bin/ping flags=(complain)**: путь к программе, управляемой профилем, также устанавливающий режим обучения.
3. **capability net\_raw,**: разрешает приложению доступ к возможностям CAP\_NET\_RAW Posix.1e.
4. **/bin/ping mixr,**: разрешает приложению доступ на чтение и выполнение файла.

После редактирования файла профиля, он должен быть перезагружен. Для детальной информации обратитесь к разделу [Использование AppArmor](https://help.ubuntu.ru/wiki/%D1%80%D1%83%D0%BA%D0%BE%D0%B2%D0%BE%D0%B4%D1%81%D1%82%D0%B2%D0%BE_%D0%BF%D0%BE_ubuntu_server/%D0%B1%D0%B5%D0%B7%D0%BE%D0%BF%D0%B0%D1%81%D0%BD%D0%BE%D1%81%D1%82%D1%8C/apparmor#%D0%B8%D1%81%D0%BF%D0%BE%D0%BB%D1%8C%D0%B7%D0%BE%D0%B2%D0%B0%D0%BD%D0%B8%D0%B5_apparmor)

#### Создание профиля

1. **Разработка плана тестирования**: Попробуйте подумать о том как приложение будет выполняться. План тестирования стоит разделить на маленькие тестовые блоки. Каждый тестовый блок должен иметь краткое описание и перечень шагов выполнения. Некоторые стандартные тестовые блоки:

1. Запуск программы.
2. Остановка программы.
3. Перезагрузка программы.
4. Тестирование всех команд, поддерживаемых сценарием инициализации.

2. **Создание нового профиля**: Используйте **aa-genprof** для создания нового профиля. Команда в терминале:

sudo aa-genprof executable

Например:

sudo aa-genprof slapd

3. Чтобы получить ваш новый профиль в составе пакета **apparmor-profiles**, зарегистрируйте проблему в Launchpad для пакета [AppArmor](https://bugs.launchpad.net/ubuntu/+source/apparmor/+filebug" \o "https://bugs.launchpad.net/ubuntu/+source/apparmor/+filebug):

1. Включите ваш план тестирования и тестовые блоки.
2. Присоедините ваш новый профиль к зарегистрированной проблеме.

#### Обновление профилей

Когда программа ведет себя неправильно, проанализируйте сообщения отправленные в файлы журналов. Программа **aa-logprof** может быть использована для сканирования файлов журнала **AppArmor** для проверки сообщений, их рассмотрения (анализа) и обновления профилей. Команда в терминале:

sudo aa-logprof

### Ссылки

1. Смотрите [AppArmor Administration Guide](http://www.novell.com/documentation/apparmor/apparmor201_sp10_admin/index.html?page=/documentation/apparmor/apparmor201_sp10_admin/data/book_apparmor_admin.html" \o "http://www.novell.com/documentation/apparmor/apparmor201_sp10_admin/index.html?page=/documentation/apparmor/apparmor201_sp10_admin/data/book_apparmor_admin.html) для дополнительных опций настройки.
2. Для уточнения использования AppArmor с другими выпусками Ubuntu смотрите страницу [AppArmor Community Wiki](https://help.ubuntu.com/community/AppArmor" \o "https://help.ubuntu.com/community/AppArmor).
3. Страница [OpenSUSE AppArmor](http://en.opensuse.org/SDB:AppArmor_geeks" \o "http://en.opensuse.org/SDB:AppArmor_geeks) - еще одно знакомство с AppArmor.
4. Хорошее место для вопросов поддержки AppArmor и вовлечения в сообщество Ubuntu Server - IRC канал **#ubuntu-server** на [freenode](http://freenode.net/" \o "http://freenode.net/).