Глава 1: Введение в sudo

Контроль доступа пользователя к привилегированным программам и файлам, как правило, является головной болью. Ни одна из систем, которые создавались для отражения привилегий реального мира на цифровое пространство, не работает достаточно хорошо. Лучшие системы контроля доступа просто менее болезненны чем прочие.

Unix-подобные системы управляют доступом к программам и файлам посредством системы пользователей и групп. Каждый пользователь имеет уникальный идентификатор передаваемый ему как имя пользователя или идентификационного номера пользователя (user ID - UID). Пользователи помещаются в однозначно идентифицированных группах которым присваиваются идентификационные номера групп (group ID - GID). Отдельные пользователи и группы имеют разрешение на доступ к определённым файлам и программам.

В начале развития UNIX этой схемы было вполне достаточно. Большой университет мог иметь несколько UNIX-серверов. Сотни пользователей, зарегистрированные на каждом почтовом сервере, сервере новостей и в приложениях интенсивных вычислений. Студенты определялись в одну группу, аспиранты в другую, затем профессора и так далее. Индивидуальные классы и департаменты могли иметь свои собственные группы.

Владелец системы имел специальную учётную запись - root. Она предоставляет полный контроль над системой. В целях безопасности и стабильности, UNIX-подобные системы ограничивают определённые операции, и только root имеет право на их выполнение. Только root может выполнять конфигурирование сети, монтировать новые файловые системы и перезапускать программы присоединённые к привилегированным сетевым портам.
Это имело смысл, когда у вас было два сервера для всего кампуса - реконфигурирование сети или подключение нового диска является серьёзной задачей в этой среде. Работа по управлению системами многомиллионной стоимостью должна была оставаться в надёжных руках.

В 21-веке, UNIX-подобные системы стали дёшевы и многочисленны. Группы людей могут разделять административные задачи управления системой или один человек может иметь полный контроль над ней или может существовать нечто среднее. Ситуация может полностью изменять свои требования к безопасности относительно прошлого века.

Крупные организации часто делят обязанности по администрированию системы между несколькими квалифицированными лицами. Один человек может нести ответственность работу самой системы, в то время как другой может заниматься обслуживанием приложения работающего на сервере. Сервер поддерживает приложения, приложения существуют на сервере, но людям требуется выполнение задач для которых требуются привилегии корневого уровня. Однако передача привилегий root - это решение из категории "всё или ничего". Здесь нет никакой разницы между "доступом к изменению ядра" и "доступом для запуска привилегированного приложения". Если администратор приложения имеет root-доступ, он может произвести изменения ядра. Вы всегда можете рассчитывать на джентльменские соглашения касаться только своей ответственной части системы, но когда в вашей

организации работает команда системных администраторов и команда администраторов баз данных, поддерживающих десятки или сотни серверов, подобные джентльменские соглашения быстро приводят к кровопролитию. Такие организации нуждаются в более тонком разграничении доступа к управлению системой чем может предоставить root.

Модель "всё или ничего" становится ещё более ущербной когда у каждого имеется UNIX-подобная система. Не беря в расчёт множество телефонов и планшетов имеющих дополнительное ПО делающее их более удобными в использовании, многие используют UNIX-подобные системы на настольном ПК или ноутбуке. Каждый раз когда им требуется доступ к USB диску или настройка беспроводной сети, им потребуется привилегированный доступ к системе на уровне root. Использование прав суперпользователя нельзя назвать страшно обременительной процедурой - войдите в свою учётную запись, используйте команду su для переключения пользователя, введите пароль root, выполните необходимые команды которые требуют привилегий и выйдите из учётной записи root. Однако, когда необходимость в получение привилегий возникает постоянно, на мелких рутинных операциях это быстро начинает раздражать.

В компьютерной индустрии работает множество действительно умных людей, которые смогли расширить классическую модель управления привилегиями UNIX. Одним из способов является использование флагов setuid и setgid. Обычно, программы работают с привилегиями того пользователя который ими управляет, программы с установленными флагами setuid и setgid изменяют действующее значение UID и GID на какое-то другое. Вы можете выполнять программу с флагом setuid работающую с привилегиями root. Например изменение пароля требует редактирование защищённых файлов расположенных в директории /etc/, по этой причине команда passwd - setuid. Однако, злоумышленники очень любят программы с флагами setuid и setgid. Ошибки в таких программах могут в полной мере использоваться для доступа к root. В связи с этим, большинство операционных систем не позволяют использовать флаг setuid на скриптах оболочки, а только на программах.

Кроме этого, существует несколько разновидностей списков контроля доступа (Access control lists - ACL), значительно расширяющих модель пользователь-группа-прочие (user-group-others). Списки ACL позволяют сделать заявление вида "этот человек владеет файлом, но эти группы и пользователи могут его изменять, с некоторыми исключениями, эти группы и пользователи могут его выполнять, в то время как другие могут только читать его данные, за исключением..." и тому подобное. В этой точке системный администратор обретает дополнительную головную боль и подумывает сменить работу. И уж конечно, различные реализации ACL являются частично не совместимыми друг с другом. Очень немногие могут правильно реализовать списки ACL на одной платформе, и не факт что смогут распространить на иную платформу. Однако списки ACL имеют место в системном администрировании, и если они действительно вам необходимы - они бесценны. Печальный факт в том, что большинство из нас не нуждаются в них.

К сожалению, списки контроля доступа хороши на столько на сколько хорошими они получаются. За исключением sudo.

Что такое Sudo?

Sudo это программа управляющая доступом к выполнению команд иот имени root или другого пользователя. Владелец системы создаёт список привилегированных команд, которые может выполнять каждый пользователь. Когда пользователю требуется выполнить команду которой необходимы привилегии на уровне root, он просит sudo выполнить эту команду для него. sudo проверяет свои права по заранее установленному списку. Если пользователь имеет разрешение на выполнение этой команды, он может её выполнить. Если такового разрешения нет, sudo сообщает пользователю об этом. Запуск sudo не требует пароля root, а только собственный пароль пользователя (либо иной способ аутентификации).

Системный администратор может делегировать привилегии уровня root для конкретных пользователей выполняющих очень специфические задачи не выдавая при этом пароль root. Он может указать sudo требовать проверки подлинности для некоторых пользователей или команд и не требовать этого для других. Он может разрешить пользователям доступ на некоторых машинах и запретить доступ на других основываясь на едином, общем файле конфигурации.

Некоторые приложения, в частности, большие базы данных предприятия, работают под управлением конкретной выделенной учётной записи. Пользователям необходимо переключиться на эту учётную запись для управления программным обеспечением. Вы можете настроить sudo позволяя пользователям выполнять определённые команды от имени этой учётной записи. Возможно, что младшим администраторам баз данных требуется только выполнять резервное копирование, в то время как главный администратор базы данных должен иметь полный доступ к оболочке базы данных. Sudo позволяет решить эту проблему.

Наконец, sudo регистрирует все выполняемые действия. Можно даже воспроизвести содержимое индивидуальной sudo-сессии, чтобы разобраться кто и что нарушил.

Что же не так с sudo?

Если sudo такой прекрасный инструмент, почему он не используется повсеместно?
Sudo добавляет ещё один уровень администрирования системы. Добавления этого уровня требует затраты времени, энергии и внимания. Это требует внимательного изучения ещё одной опасной программы, в то время когда у вас и так полно работы. Если вы отвечаете ра работу системы предприятия с несколькими группами администраторов, вложения в sudo снижает нагрузку. Но вам требуется научиться использовать его правильно.

Некоторые коммерческие UNIX не включают sudo, поскольку уже имеют собственную систему управления привилегиями. Системы на базе OpenSolaris имеют pfexec и контроль доступа на основе ролей (role-based access control - RBAC). HP имеет pbrun. Если бы вы были коммерческим вендором UNIX, потратившим уйму средств на развитие системы контроля привилегий базирующейся на ACL, вы бы поощряли включение и использование вместо неё более простого инструментария? Я бы наверное мог, но я не продаю UNIX :).

Искусство SUDO:
Контроль доступа пользователей для простых людей

Author:Michael W Lucas

Многие UNIX-подобные системы с открытым кодом включают sudo в базовую систему. Некоторые из них, такие как Ubuntu и OS X, полностью отключают учётную запись root и разрешают привилегированный доступ только посредством sudo. Это крен в правильном направление, но большинство пользователей не умеют правильно использовать sudo.

Как можно неправильно использовать sudo? Во первых sudo не заменяет su. Sudo это не способ полностью избежать требования аутентификации для получения привилегированного доступа. Sudo не является инструментом который сделает за вас что-то. Правильная настройка sudo упрощает управление системой. Неправильная настройка sudo позволяет злоумышленникам и не авторизованным пользователям быстрее и проще испортить и уничтожить вашу систему.

"Правильное использование sudo" не означает использование сложных и обширных политик. Я видел системы в которых администраторы писали сложные политики sudo, а пользователи вальсировали мимо их ограничений. Иногда пользователи даже не подозревали о наличии этих ограничений. Sudo имеет свои пределы. После того, как вы поймёте эти ограничения, вы сможете принимать реалистичные решения о том как правильно развернуть sudo.

Проблемы которые я часто наблюдаю в связи с sudo не имеют ничего общего с самим программным обеспечением. Правильное развёртывание sudo в сложной организации требует сплочённой команды системного администрирования, в которой каждый знает кем является и за что отвечает. Sudo связывает обязанности и ответственность в своём файле конфигурации. Файл конфигурации является достаточно гибким и пользователи не могут превышать привилегий указанных в нём.

Каковы границы ваших обязанностей? какими разрешениями вы должны обладать чтобы выполнять свою работу и какие задачи должен решать кто-то ещё? Задумываться об этих вещах может быть неудобно и это может привести к конфликтам внутри организации. Однако после чёткого аргументирования конфликтность должна снизиться. Не должно возникать склоки о том, кто, что, когда и как сделал. Все знают, что команда администрирования базы данных не должна форматировать файловые системы, web-администраторы не должны перезапускать базу данных - журналы sudo однозначно показывают кто выполнял привилегированные операции. Наличие аудита повышает стабильность системы. Когда пользователи знают, что система регистрирует привилегированные действия и что им придётся нести ответственность за возникшие проблемы - проблемы начинают возникать реже. Странно да?

Примечание переводчика: Я всегда говорил и буду говорить, что только системное администрирование не решает проблемы. Изначально, администрировать необходимо физический мир и вырабатывать чёткую систему мер и наказаний. Только затем отражать её на цифровую реальность.

Перевод: Михайлов Алексей (www.iboxjo.ru)

Как вас защитит sudo?

Sudo защищает систему от её повреждения злоумышленниками или системными администраторами, а кроме того защищает системных администраторов от различных проблем в области управления системой.

Предоставление доступа к ограниченному набору привилегированных команд ограничивает ущерб, который могут нанести системе действия пользователя. Пользователь имеющий доступ только к управлению web-сервером или базой данных не может изменять разделы диска. Если же злоумышленник компрометирует учётную запись пользователя, он будет сдерживаться ограничениями наложенными sudo.

Кроме того, ограничение доступа защищает системного администратора при разборе полётов. Даже не рассматривая журналы sudo, пользователь с ограниченными правами администратора может сказать: "Эй, я не перенастраивал веб сервер - у меня нет к этому доступа, помните?". Ответственность - палка о двух концах. Используйте её в своих интересах.

Поддержка sudo

Sudo является программным обеспечением с открытым исходным кодом. Вы можете скачать его с основного web-сайта http://sudo.ws или с соответствующего зеркала и использовать его в своих целях бесплатно. Лицензия разрешает вам использовать sudo в качестве основы своих собственных продуктов, перепродавать его клиентам или включить его в своё ПО, а затем распространять или продавать. Вы можете использовать sudo для любых своих целей. Короче не получите вы никакой поддержки.

Sudo не создаётся коммерческой компанией. Программа разработана и поддерживается пользователями которые в ней нуждаются и в течение последних лет координируются Тоддом Миллером (Todd Miller). Вы можете внести свой вклад в sudo, предоставляя патчи или сообщения об ошибках. Так же, вы можете найти людей и компании которые могут помочь вам установить sudo или даже написать специфический код. Но жаловаться, если что-то не работает так, как вы ожидаете - ненакого. Нет бесплатного номера поддержки, нет оплаты сотрудников техподдержки.

Хочу сказать, что списки рассылки sudo являются весьма полезными для заинтересованных в реальных проблемах. Они хорошо реагируют на просьбы о помощи и отрицательно относятся к каким либо требованиям.

Кто должен прочитать эту книгу?

Каждый, кто работает с UNIX-подобной системой должен понимать sudo. Если вы системный администратор, отвечающий за поддержку сложной системы, вам, скорее всего, потребуется назначить администраторов отдельных приложений которые смогут выполнять свою работу в жёстких ограничениях. Правильная настройка sudo поможет освободить дополнительное время и защитить систему от не благобриятных воздействий.

Если же вы являетесь администратором приложения, вам необходимо выполнять свою работу. Это означает, что вам требуется доступ к выполнению привилегированных задач. Работа посредством sudo изменит процессы незначительно - но вы можете свихнуться пытаясь понять почему не работает sudo cd, пока не поймёте что происходит на самом деле. Понимание sudo позволяет необходимым образом разработать правила sudo и сообщить об этом системному администратору. Даже если системный администратор не согласен с вашими выводами, переговоры на языке политик sudo будут означать, что вы оба понимаете чего именно хотите. Вы можете конкретно обсудить границы своей ответственности. Конечно системный администратор не укажет администратору приложений что он не даст ему необходимый доступ. Вопрос в том - как лучше всего предоставить необходимый уровень доступа?

Если же разногласия между командами достаточно глубоки, необходимо установить конкретные решения и чёткие линии полномочий и разграничения ответственности. Если у вас есть капризный системный администратор, утверждающий, что предоставить вам конкретный доступ без предоставления привилегий root невозможно, эта книга поможет вам категорически его опровергнуть. А это, надо признать, доставляет некоторое удовольствие.

Если вы поддерживаете собственную систему, зачем вам озадачиваться sudo? Даже на собственном ноутбуке, некоторые команды заслуживают большего внимания и осмысливания, чем прочие. Я могу понять желание перенастроить сеть, поднастроить съёмный диск или прикончить безумный web-браузер. Вероятно вы настолько часто выполняете эти задачи, и уже считаете, что пальцы сделаю это не напрягая мозг. Однако, задачи выполняемые реже, например такие как установка ПО или форматирование дисков требуют немного больше внимания. Соответственно, имеет смысл настроить sudo для выполнения рутинных задач без ввода пароля, но, например, требовать проверки подлинности перед проведением обновления.

Требования сервера

Эта книга предполагает, что вы выполняете sudo на UNIX-подобной ОС. Sudo доступно для BSD и производных Solaris, Linux и коммерческих UNIX. Хотя я ссылаюсь на FreeBSD, sudo работает на всех этих платформах и многих других.

У меня установлена версия sudo 1.8.8. Если вы используйте более старую версию, некоторые функции могут отсутствовать. Удивительно, но многие вендоры ОС включают в системы дико устаревшие пакеты sudo. Проверьте версию sudo на вашей системе, выполнив sudo -V. Если ваша версия намного старее чем 1.8.8 - выполните соответствующее обновление. Вы всегда можете получить последнюю версию исходного кода или выбрать бинарные пакеты на сервере http://sudo.ws.

Документация sudo и эта книга предполагают, что ваша операционная система имеет традиционную схему файловой системы или достаточно близка к ней. Примеры приведённые в этой книге отражают команды в стандартных директориях, таких как /bin, /usr/bin, /sbin и

т.д. Если же ваша ОС использует иную иерархию директорий, вам необходимо учесть её при использовании примеров.

Для системного администратора

Многие важные программы требуют обширного опыта в соответствующей области ПО для их корректного использования, а sudo является самодостаточной. Вы можете освоить sudo без понимания всех программ к которым пользователи могут получать доступ посредством sudo. Sudo является инструментом управления системой, однако, чем лучше вы понимаете свою систему, тем лучше вы можете использовать sudo и тем больше будет у вас уверенности в точности конфигурации. Я предполагаю, что вы можете установить sudo из пакета ОС или из исходного кода.

Настройка sudo на UNIX-подобной системе требует доступа уровня root и знакомства с текстовым терминалом и редактором. По умолчанию используется редактор vi, но вы можете использовать emacs или любой другой редактор.

Пока это всё что необходимо. Остальные знания вы получите по ходу чтения.

Изучение sudo

Цель данной книги - позволить вам заменить доступ к привилегированным командам посредством su и пароля root с помощью команды sudo и личных учётных записей пользователя. Когда вы научитесь комфортно работать с sudo, вы сможете использовать механизм аутентификации системы, исключив получение привилегий root посредством su. Пароль root станет средством используемым только в экстренной ситуации или при работе на физической консоли. Аутентификация исключающая root позволит улучшить отчётность системного администратора в пределах организации. Однако не стоит быть торопливым при развёртывании sudo.

Конфигурирование sudo имеет свои подводные камни. Вам необходимо понимать, как sudo вписывается в рабочую среду. Ничто так не раздражает, как блокировка самого себя на собственном сервере. Не стоит спешить с отключением доступа к root посредством su, поскольку вы можете использовать этот способ для восстановления повреждённой или неверной конфигурации sudo. Конечно, sudo имеет функционал и инструменты, позволяющие убедиться, что синтаксис конфигурации политик sudo верен. Да, политика указывающая что "никто не может сделать что-то" будет иметь верный синтаксис. Но... Поэтому стоит, на время, оставить доступ посредством su, пока вы не будете абсолютно уверенны в новых механизмах sudo или полностью не освоитесь с работой в однопользовательском режиме работы для восстановления работы. Так же, для деструктивного обучения может быть весьма полезной работа с виртуальной машиной или jail-окружением.

Некоторые операционные системы (в частности Ubuntu и OS X) предоставляют root привилегии посредством sudo а не su. Если вы ещё только экспериментируете с sudo, и sudo

является вашим основным методом выполнения привилегированных команд, вы можете оказаться в рискованной ситуации. Перед началом работы с sudo, включите root доступ и установите пароль root на вашей машине. Убедитесь, что всё работает и, что вы можете получить привилегированный доступ без использования sudo. Только после этого, вы можете безопасно исследовать sudo не опасаясь блокировки привилегированного доступа. После комфортного освоения sudo, вы можете спокойно развернуть его в полной мере.

Официальная документация sudo описывает различные функции sudo в расширенной форме Бэкуса-Наура (EBNF - Extended Backus-Naur Form) - вид формальной грамматики для конфигурации программы. Хотя знакомство с EBNF является полезным навыком для любого системного администратора, я не предлагаю вам изучать формальные определения. Вместо этого, в данной книге, будут показаны наиболее важные особенности sudo - в виде частей конфигурации политик.

Так же, следует обратить внимание, что эта книга не охватывает все возможные конфигурации sudo и весь его функционал. Я рассматривал только то, что необходимо абсолютному большинству системных администраторов. Если же вы работаете со старой ОС, использующей старую версию sudo, или администрируете UNIX-подобную систему дико выпадающую из общих стандартов, вам потребуется погружение в формальную документацию, что позволит сгладить резкие грани сложившейся ситуации.

После прочтения данной книги, вы будете иметь твёрдую основу в понимании методов sudo.

Как избежать sudo

Многие системные администраторы при настройке своих систем требуют привилегий суперпользователя, хотя им следует использовать привилегии группы поддерживаемые базовой операционной системой. Часто, мы склонны рассматривать разрешения для пользователей и прочих, но мало обращаем внимания на права доступа группы. Прежде чем выполнить sudo следует определиться - возможно, вместо этого, вы можете решить вашу проблему с использованием разрешения групп. Требовать привилегии root чтобы разрешить доступ к файлам или программам может выглядеть, как потребовать использование кувалды для того, чтобы повесить картину.

Используйте разрешения группы для программ или файлов, к которым должны иметь доступ несколько пользователей. В качестве простейшего примера, предположим, что некоторые пользователи осуществляют поддержку web-сайта. Вы можете создать группу, назвав её, например, webadmins, и назначить эту группу в качестве владельца директории web-сайта и всех файлов в этой директории. Давайте посмотрим на директорий верхнего уровня нашего web-сайта.

#ls -l
total 94
drwxrwxr-x 2 mike webadmins 512 Jul 12 2013 content
-rw-rw-r-- 1 thea webadmins 16584 Oct 20 2013 logo.jpg

-rw-rw-r-- 1 pete webadmins 767 Oct 20 2013 errata.html
-rw-rw-r-- 1 mike webadmins 2736 Jul 12 2013 index.html
-rw-rw-r-- 1 pete webadmins 167 Jul 12 2011 index2.html
-rw-rw-r-- 1 thea webadmins 66959 Oct 20 2006 banner.jpg

Отдельные файлы принадлежат отдельным пользователям - mike, thea или pete. Но, кроме того, файлы могут читаться и записываться группой webadmins. Любой в этой группе может читать и редактировать эти файлы и любые другие в этом директории.

Специфика добавления групп варьируется в зависимости от операционных систем. Я бы сказал вам, что необходимо отредактировать /etc/groups, но некоторые ОС используют специальные инструменты для управления группами. Обратитесь к руководству по операционной системе.

В какие группы вхожу я?

Чтобы определить к каким группам принадлежите вы, выполните команду id(1).

#id
uid=1001(mike) gid=1001(mike) groups=1001(mike),10020(webadmins)

Как видите, моя учётная запись входит в группы mike и webadmins. Я мог бы редактировать файлы из приведённого выше примера на основе моей принадлежности к группе. Кроме того, я мог бы произвести изменение пары файлов поскольку я являюсь их владельцем и имею такие разрешения.

Программы против групп

Разрешения группы не решат все проблемы доступа к программам. Некоторые программы выполняют привилегированные функции и предоставив группе право на выполнение программы вы не передадите программе прав на выполнение этих функций. Помните, что программа работает с привилегиями пользователя запустившего программу.

Продолжим историю нашей команды по управлению web-сервером. Web-серверы работают на TCP портах 80 и/или 443/ Только root может осуществлять подключение к портам с номерами ниже 1024. Если пользователь выполняет программу web-сервера без дополнительных привилегий, программа будет запущена с учётной записью пользователя. Она не получит необходимых привилегий для подключения к требуемым сетевым портам и, соответственно web-сервер не сможет запуститься. Настройка разрешений программы таким образом чтобы пользователь мог бы её запустить не означает, что программа будет работать. Если вам требуется, чтобы группа webadmins получила привилегии суперпользователя для запуска, остановки и управления web-сервером, вам необходимо предоставить пользователям этой группы привилегии суперпользователя. Здесь то вам и подходит sudo - вы можете назначить участникам команды управления сервером и ничего другого.

Обзор книги

Sudo представляет собой комплекс взаимосвязанных программ. Вы достигнете лучшего результата в конфигурировании sudo если будете понимать, как различные части взаимодействуют друг с другом.

Традиционно, sudo включает два компонента: программу sudo и движок политик sudoers (файл политик). Глава 2 является элементарным введением в обе составляющие. Файл политик sudoers можно изменять только обладая привилегиями суперпользователя. Ошибка в файле sudoers может привести к блокировке получения привилегий пользователя с помощью sudo. Если вы отключили доступ к root с использованием иных средств, вы можете оказаться блокированными. В пакет sudo так же входит специальный инструмент, visudo, предназначенный для редактирования и проверки файла sudoers. Использование visudo снижает шансы возникновения проблем с конфигурированием. visudo рассматривается в Главе 3.

Политики sudo быстро становятся очень сложными. В Главе 4 описано, как можно снизить эту сложность с помощью использования списков и псевдонимов. Однако, вы не можете настроить все детали поведения sudo только посредством правил политик. Политики sudoers включают в себя различные настройки и параметры по умолчанию, и о том как их изменить я расскажу в Главе 5.

Некоторые программы предлагают пути обхода ограничений sudo с помощью побега из оболочки - не по причине того что они были написаны так специально, а из-за своих характерных особенностей. Глава 6 рассматривает способы предотвращения получения неограниченной оболочки root из текстовых редакторов и других подобных программ.

Большая часть книги посвящена политикам безопасности sudoers, но сама программа sudo тоже может быть настроена. В Главе 7 обсуждается файл sudo.conf.

Среда пользователя может вызывать различные виды проблем при использовании привилегированных программ. Глава 8 охватывает способы очистки среды пользователя и блокировки или разрешения переменных среды в контексте sudo.

Sudo может производить базовую проверку целостности программ перед их выполнением. Об этом мы поговорим в Главе 9.

Вместо того, чтобы поддерживать отдельные политики безопасности на каждой из десятков или сотен машин, вы можете использовать одну, централизованную политику и распространить её на все хосты. Глава 10 рассматривает распространение единой политики по сети.

Sudo может получать политику безопасности с сервера аутентификации LDAP, а не только посредством файла sudoers. Совместная работа LDAP и sudo рассматривается в Главе 11.

После того, как вы сможете проконтролировать доступ пользователей к привилегированным программам у вас возникнет следующий вопрос: "что же сделал пользователь?". sudo включает три различных системы журналирования используемых для различных потребностей. В Главе 12 мы рассмотрим все три.

Наконец, sudo может работать с учётными данными пользователя в различных формах, и если вы хотите настроить то как sudo будет управлять паролями и другими данными аутентификации, вам следует прочитать Главу 13.

Однако, прежде чем перейти к более широким знаниям, давайте начнём с основ sudo.