Linux Security HOWTO

Kevin Fenzi, kevin@scrye.com & Dave Wreski, DAVE@nic.com, перевод Konstantin Shepelevich, sh_ki@hotmail.com

v0.9.11, 1 May 1998

*Этот документ является общим обзором проблем безопасности, которые встают перед администратором Linux систем. Он изображает общую философию безопасности и некоторые специфические примеры лучшей защиты вашей Linux системы от злоумышленников. Также включены касающиеся безопасности материалы и программы. Примечание: Здесь представлена beta версия данного документа. Улучшения, конструктивная критика, добавления и исправления принимаются с радостью. Пожалуйста шлите Ваши пожелания обоим авторам. Чтобы избежать spam фильтров и обратить внимание авторов обязательно включите слова "Linux", "security" или "HOWTO" в subject строку Вашего письма. Прим.перевод.: Как видно из даты документ немного устарел в плане версий тех или иных программ и, соответственно, их возможностей, но приводится в полном согласии с оригиналом, поскольку принципы они всегда **принципы**.*

1. Введение

Этот документ покрывает некоторые из главных вопросов безопасности, которые касаются Linux. Обсуждаются также общая философия и порождаемых сетью ресурсов.

Существуют и другие документы HOWTO касающиеся темы безопасности, о них будет указано, когда это будет уместно.

Этот документ не подразумевает выдачу рецептов по каждодневным проблемам. Постоянно появляется большое количество новых проблем. Этот документ расскажет вам, где искать информацию, а также некоторые общие рекомендации как избежать этих проблем.

1.1 Новые версии этого документа

Новые версии этого документа периодически публикуются в *comp.os.linux.answers*. Также они добавляются на различные анонимные FTP узлы, которые собирают такую информацию, включая:

ftp://sunsite.unc.edu/pub/Linux/docs/HOWTO

К тому же, вы в общем то можете найти этот документ на странице Linux Worldwide Web через:

http://sunsite.unc.edu/mdw/linux.html

И, наконец, наиболее последняя версия этого документа должна быть доступна в различных форматах на:

http://scrye.com/~kevin/lsh/

1.2 Обратная связь

Все комментарии, сообщения об ошибках, дополнительная информация и критика всех видов должны быть направлены на

kevin@scrye.com

и

dave@nic.com

Примечание: Пожалуйста, посылайте ваши сообщения **обоим** авторам. Также не забудьте добавить слова "Linux", "security" и "HOWTO" в ваш subject во избежание попадания под spam фильтр.

1.3 Отречение

По содержанию этого документа автор не несет никаких обязательств. Вы можете использовать концепции, примеры и другую информацию этого документа на свой страх и риск. К тому же это ранняя версия с большой вероятностью содержащая неточности и ошибки.

Некоторое количество примеров и описаний основывается на дистрибутиве RedHat(tm) и его установках. Ваша система может отличаться.

Насколько мы полагаем, будут описываться только те программы, которые наверняка можно использовать для персональных потребностей. Большинство программ будут доступны полностью с исходными кодами под GNU лицензией.

1.4 Авторские права

- Если у вас есть вопросы, свяжитесь с Tim Bynum - Linux HOWTO координатором по адресу

linux-howto@sunsite.unc.edu

2. Обзор

Этот документ пытается объяснить некоторые процедуры и используемое программное обеспечение, призванные помочь сделать Linux систему более безопасной. Очень важно сначала обсудить некоторые базовые концепции и создать некий фундамент безопасности.

2.1 Зачем нам нужна безопасность?

В постоянно изменяющемся мире коммуникаций глобальных данных, недорогих Internet соединений и быстрой разработки программных продуктов, безопасность становится все более и более насущной. Безопасность сейчас является базовым требованием, поскольку глобальная компьютеризация в своей сути небезопасна. Например, когда ваши данные перемещаются из точки А в точку Б в Интернете, они на своем пути могут проходить через многие узлы, давая другим пользователям возможность перехвата и даже подмены ваших данных. Даже другие пользователи в вашей системе могут злонамеренно преобразовать ваши данные во что-то, чего вы не хотите. Неавторизованный доступ к вашей системе могут получить взломщики, также известны как "кракеры", которые затем могут использовать свои (продвинутые) знания для получения информации о вас, воровства ваших данных или, даже, запрещения вам доступа к вашим же ресурсам. Если вы все еще хотите узнать разницу между "хакером" и "кракером", посмотрите документ Eric Raymond's, "Как стать Хакером", доступным по адресу http://sagan.earthspace.net/~esr/faqs/hacker-howto.html.

2.2 Насколько безопасна безопасность?

Перво-наперво запомните, что не существует компьютерной системы, которая была бы "полностью безопасна". Все, что вы можете сделать, это существенно затруднить кое-кому нанести вред вашей системе. От среднего домашнего пользователя Linux немного требуется, чтобы сдержать случайного кракера. Для широко-профильных пользователей Linux (банки, телекоммуникационные компании и т.п.) потребуется намного больше работы.

Другим фактором, который нужно принять во внимание, является то, что чем более безопасна ваша система, тем более навязчивой становиться ваша система безопасности. Вы должны решить, где находится баланс между удобством использования системы и необходимым уровнем безопасности в вашей работе. Например, вы могли бы требовать от всех удаленных пользователей вашей системы использовать модемы с запросом на дозвон (call back modem), чтобы ваша система дозванивалась к ним на их домашний телефон. Это более безопасно, но если кто-нибудь захочет войти в вашу систему не из дому, то ему будет довольно трудно зарегистрироваться. Вы также можете установить вашу Linux систему без сети или связи с Интернет, но это повлечет за собой невозможность Web серфинга.

Если у вас система средних или больших размеров, вам нужно установить "Политику Безопасности", которая определит, насколько сильной должна быть у вас система безопасности и какой должен быть аудит для ее проверки. Вы можете найти хорошо-известный пример "политики безопасности" по адресу http://ds.internic.net/rfc/rfc2196.txt. Узел был недавно обновлен и содержит великолепный план установления политики безопасности в вашей компании.

2.3 Что вы пытаетесь защитить?

До того как вы начнете настраивать безопасность вашей системы, вы должны определить, какому уровню угрозы вы должны противостоять, какой уровень риска вы должны или не должны принимать, и насколько уязвима после этого будет ваша система. Вы должны проанализировать вашу систему, чтобы знать, что вы защищаете, почему вы это защищаете, какую это имеет ценность, и кто несет ответственность за ваши данные и другие ценности.

- Риск - это вероятность того, что взломщик может одержать победу в в своих попытках получить доступ к вашему компьютеру. Сможет ли взломщик читать или писать файлы, или выполнять программы, которые могут нанести ущерб? Может ли взломщик удалить критические данные? Помешать вам или вашей компании завершить очень важную работу? Не забудьте, что кто-то, получив доступ к вашему счету (account), или вашей системе, может притвориться вами.

  Кроме того, возникновение одного небезопасного счета в вашей системе может подвергнуть риску всю вашу сеть. Имея единичного пользователя, которому позволено регистрироваться в системе используя rhosts файл, или разрешено использовать небезопасный сервис, такой как tftp, вы рискуете, поскольку взломщик может использовать это, чтобы "открыть ногой вашу дверь". Как только взломщик заимел счет пользователя в вашей системе, или еще какой-либо системе, он может его использовать для получения доступа к другим системам и другим счетам.

- Опасность обычно исходит от кого-нибудь, имеющего желание получить неразрешенный доступ к вашей сети, или компьютеру. Вы должны решить, кому вы доверяете доступ к вашей системе, и какую угрозу они могут представлять.

  Существует несколько типов взломщиков, поэтому полезно помнить отличающие их характеристики во время создания системы безопасности.

  - **Любопытный** - Этому типу взломщика в основном интересно выведать, какого типа у вас система и что за данные вы используете.

  - **Злобный** - Этот тип взломщика стремиться либо "вырубить" вашу систему, либо обезобразить ваш Web узел, либо сделать другую пакость, отбирающую у вас время и деньги на восстановление.

  - **Высококлассный взломщик** - этот тип взломщика пытается использовать вашу систему для получения популярности. Он может использовать взлом вашей хорошо защищенной системы для рекламы своих способностей.

  - **Конкурент** - Этот тип взломщика интересуется данными, которые вы имеете в вашей системе. Это может быть кто-то, кто думает, что вы имеете что-то, что принесет ему денежную либо какую-нибудь другую прибыль.

- Понятие **уязвимость** описывает, насколько хорошо защищен ваш компьютер от других в сети, а также возможность получения кем-либо неразрешенного доступа к вашей системе.

Что будет, если кто-то вломиться в вашу систему? Конечно, важность домашнего пользователя с динамическим PPP соединением отличаться от того, кто имеет выход в интернет или другую большую сеть через сеть своей компании.

Сколько времени необходимо на восстановление/воссоздание данных, которые были утеряны? Изначально потраченное время сейчас может сэкономить в десять раз больше времени потом, когда вам будет нужно воссоздать утерянные данные. Вы уже проверили вашу стратегию резервирования (backup strategy), а позже проверяли данные на целосность?

2.4 Разработка политики безопасности

Создайте простую, общую политику для вашей системы, чтобы ваши пользователи могли быстро ее понять и следовать ей. Это должно сберечь данные, которые вы охраняете, а также конфиденциальность пользователей. Есть несколько вещей для дополнительного рассмотрения: кто должен иметь доступ к системе (Может ли мой друг использовать мой счет?), кому разрешено инсталировать программное обеспечение в системе, кто владеет данными, проводит восстановление, и соответственно использует систему.

Общепринятая политика безопасности начинается с фразы:

"То, что не разрешено, - запрещено"

Это значит, что до тех пор, пока вы не разрешите доступ пользователю к определенному сервису, этот пользователь не сможет использовать этот сервис. Убедитесь, что политика работает, зарегистрировавшись обычным пользователем, поскольку реплики типа "Ах, как я не люблю эти ограничения прав доступа, я просто сделаю все как администратор (root)" может привести к образованию очень очевидных "дыр" в системе безопасности, и даже таких, с которыми еще не известно как бороться.

2.5 Способ защиты вашего узла (site)

Этот документ будет обсуждать различные способы, с помощью которых вы можете обезопасить активы, которые вы тяжело нарабатывали: ваш локальный компьютер, данные, пользователей, сеть, и даже ваша репутация. Что случиться с вашей репутацией, если взломщик удалит данные некоторых ваших пользователей? Или обезобразит ваш web узел? Или обнародует проект корпоративного плана вашей компании на следующий квартал? Если вы планируете структуру сети, существует очень много факторов, которые вы должны принять во внимание, прежде чем добавить какой-либо новый компьютер к вашей сети.

Даже если вы имеете один коммутируемый PPP счет, или просто маленький узел, это не значит что взломщик не заинтересуется вашей системой. Целью являются не только большие широкопрофильные сети, многие взломщики просто хотят исследовать как можно больше сетей, независимо от их размера. К тому же, они они могут использовать "дыры" в безопасности вашей сети для получения доступа к другим сетям или узлам, с которыми вы соеденены.

Взломщики имеют много времени для своих делишек, и могут, не раздумывая над тем как вы скрыли вашу систему, просто перепробовать все возможности в нее попасть. Существует также несколько причин, по которым взломщик может быть заинтересован в вашей системе - их мы обсудим позже.

Безопасность сервера

Вероятно область безопасности, в которой сконцентрировано максимум усилий, - это безопасность сервера. Обычно это подразумевает постоянный контроль безопасности вашей собственной системы, и надежду, что все остальные в вашей сети делают то же. Выбор хороших паролей, поддержка безопасности сервисов локальной сети вашего сервера, поддержка хороших регистрационных записей, и обновление программ, в которых обнаружены "дыры" - вот некоторые из тех вещей, за выполнение которых отвечает локальных администратор безопасности. Хотя это абсолютно необходимо, это может стать пугающей задачей, когда ваша сеть становится больше.

Безопасность вашей сети

Безопасность сети также необходима как и безопасность локального сервера. В вашей единичной системе, распределенной вычислительной сети, Интернете, существуют сотни, если не тысячи, компьютеров соединенные в одну сеть, и вы не можете быть уверены, что все они будут безопасны. Убедиться, что только авторизованным пользователям разрешено использовать ресурсы вашей сети, построение щитов (firewalls), использование надежной системы шифрования, отслеживание появления жульничающих или небезопасных машин в сети - все это часть обязанностей администратора сетевой безопасности.

Этот документ обсудит некоторые приемы, используемые для обеспечения безопасности вашей сети, и покажет вам некоторые способы, как не дать возможности взломщику получить доступ к тому, что вы пытаетесь защитить.

Безопасность через сокрытие

Одним из типов безопасности, который необходимо рассмотреть, является "безопасность через сокрытие". Это значит, что любые действия подобно изменению регистрационного имени из 'root' на 'toor', например, чтобы попытаться предотвратить вхождение кого-нибудь в вашу систему под 'root', являются лишь ложным чувством безопасности и могут привести к нежелательным последствиям. Многие удостоверились, что любой атакующий систему взломщик очень быстро и легко пройдет через такие пустые меры безопасности. Просто то, что у вас небольшая сеть или относительно узкопрофильный узел, не означает, что взломщик не захочет посмотреть, что вы имеете. Мы обсудим степень вашей защищенности в следующих разделах.

2.6 Структура этого документа

Этот документ разделен на несколько разделов. Они раскрывают некоторые общие вопросы касающиеся безопасности. Первый - "физическая безопасность" - рассматривает как вы должны физически защитить вашу машину от преступного использования. Второй описывает как защитить вашу систему от вредных намерений локальных пользователей. Третий - "безопасность файлов и файловой системы" - показывает вам как установить ваши файловые системы и права доступа к вашим файлам. Следующий - "безопасность паролей и шифрование" - обсуждает как использовать шифрование для лучшей безопасности вашей машины и сети. "Безопасность ядра" обсуждает какие опции ядра вы должны установить или знать для большей безопасности системы. "Безопасность сети" описывает как лучше обезопасить вашу Linux систему от атак с сети. "Подготовка безопасности" обсуждает как подготовить вашу машину(ны) к выходу в Интернет в on-line режиме. Следующий обсуждает, что делать, если вы обнаружили, что происходит прорыв в систему сейчас или был недавно совершен. Далее приводится список ссылок на другие ресурсы, касающиеся безопасности, и, наконец, некоторые вопросы и ответы и несколько заключительных слов.

Есть две важные вещи, которые нужно знать при чтении данного документа:

- Знайте вашу систему. Проверяйте системные журналы (logs), такие как /var/log/messages, и смотрите за вашей системой;

- Поддерживайте вашу систему в современном состоянии, вовремя инсталируя программы текущих версий и проводя обновления при каких-либо новых сообщениях о безопасности. Эти простые действия помогут сделать вашу систему заметно более безопасной.

3. Физическая безопасность

Первым "уровнем" безопасности, который вы должны принять во внимание, является физическая безопасность систем вашего компьютера. Кто имеет прямой физический доступ к вашей машине? Должен ли он/она его иметь? Можете ли вы защитить вашу машину от их возможно вредного действия? Должны ли вы это делать?

Степень физической безопасности, которая нужна в вашей системе, очень сильно зависит от вашей ситуации и/или бюджета.

Если вы домашний пользователь, вероятно вам не нужна сильная защита (хотя вам может понадобиться защитить вашу машину от вредных детей или надоедливых родственником). Если вы в лаборатории, то вам нужна уже значительно большая защита, но пользователям все еще будет нужна возможность работать на машинах. Многие из последующих разделов помогут в этом. Если вы в офисе, вам может понадобиться, а может и нет, обезопасить вашу машину на несколько часов или пока вы вышли. В некоторых компаниях оставить терминал незащищенным считается непростительным проступком.

Обычные методы физической безопасности, такие как запирание дверей, кабелей, шкафов, а также видео сопровождение, являются хорошей идеей, но выходят за рамки этого документа.

3.1 Запирание компьютера

Многие современные компьютерные корпуса содержат "замок". Обычно это гнездо на передней панели корпуса, вставив ключ в которое вы можете запереть либо отпереть компьютер. Запирание корпуса может помочь предупредить воровство вашего ПК, или вскрытие корпуса и прямое манипулирование/воровство компонентов вашего ПК. Это также может иногда предотвратить загрузку кем-либо компьютера со своей дискеты или другого оборудования.

Эти замки делают различные вещи в зависимости от установленной материнской платы и конструкции корпуса. На многих ПК они устроены так, что если замок заперт, то вы должны фактически сломать корпус, чтобы попасть внутрь. На некоторых других они сделаны так, что вы не сможете подключить новую клавиатуру и мышь. Посмотрите в инструкцию к вашей материнской плате или корпусу для более детальной информации. Это иногда может быть очень полезным качеством, даже если замки обычно очень низкого качества и легко могут быть вскрыты взломщиком с помощью отмычек.

Некоторые корпуса (по большей мере спарки/sparcs/ и маки/macs/) имеют dongle на задней стенке и если вы через него пропустите кабель, то взломщик будет вынужден либо его отрезать, либо сломать корпус, чтобы попасть внутрь. Использование с этим висячего или комбинированного замка является хорошим средством устрашения для желающих своровать ваш компьютер.

3.2 Безопасность BIOS

BIOS является самым нижним уровнем программного обеспечения, которое конфигурирует или управляет вашим x86 оборудованием. LILO и другие методы загрузки Linux обращаются к BIOS, чтобы узнать как загружать ваш компьютер. Другое оборудование, на котором можно запускать Linux, имеет подобное программное обеспечение (OpenFirmware на маках/macs/ и новых санах/suns/, sun boot prom и другие). Вы можете использовать ваш BIOS для предотвращения взломщиком перезапуска компьютера и управления вашей Linux системой.

Многие BIOSы ПК, работающих под Linux/x86, позволяют установить стартовый пароль. Это не предоставляет полной безопасности (BIOS может быть перезаписан или удален, если кто-нибудь заберется внутрь корпуса), но может быть хорошим сдерживающим фактором (например, это заберет время и оставит следы взлома).

Многие x86 BIOSы позволяют вам установить различные другие меры безопасности. Посмотрите в руководство по вашему BIOSy или загляните в него во время очередного перезапуска. Некоторыми примерами являются: запрет загрузки с дискеты, а также назначение пароля некоторым пунктам BIOSa.

На Linux/Sparc ваш SPARC EEPROM может быть установлен так, чтобы при запуске спрашивать пароль. Это должно задержать взломщика.

Примечание: если у вас сервер и вы установили загрузочный пароль, то ваша машина не сможет без вмешательства загрузиться. Помните, что вы должны войти в ввести пароль после сбоев электропитания. ;(

3.3 Безопасность стартового загрузчика (boot loader)

Различные загрузчики Linux также имеют возможность установки стартового пароля. Используя lilo обратите внимание на свойства "restricted" и "password". "password" позволит вам установить стартовый пароль. "restricted" позволит загружать систему до тех пор пока не встретиться установленная опция lilo: сообщение (подобно 'single').

Помните, что когда вы устанавливаете все эти пароли - вы должны помнить их. :) Также помните, что все эти пароли задержат определенного взломщика. Однако это может не помешать кое-кому загрузиться с дискеты и примонтировать ваш корневой каталог. Если вы скомбинируете средства безопасности вместе с возможностями стартового загрузчика, вы можете предотвратить загрузку с дискеты в вашем BIOSe, а также назначить пароль в BIOSe.

Если кто-либо имеет информацию о мерах безопасности в различных стартовых загрузчиках, мы бы с удовольствием услышали о них (grub, silo, milo, linload, и др.).

3.4 xlock и vlock

Если вы время от времени покидаете ваше рабочее место, было бы неплохо иметь возможность "запереть" вашу консоль так, чтобы никакой злоумышленник не мог подсмотреть вашу работу. Есть две программы, которые решают эту задачу: xlock and vlock.

Xlock это замок X экрана. Скорее всего он включен во все Linux дистрибутивы, которые поддерживают X. Чтобы узнать о нем и его опциях больше, посмотрите man страницы, но в общем вы можете запустить xlock с любого xterm на вашей консоли, при этом он "запрет" дисплей и запросит пароль, если вы захотите продолжить работу.

vlock простая маленькая программа, которая позволяет вам "запереть" некоторые или все виртуальные консоли вашей Linux системы. Вы можете "запереть" только ту, на которой работаете, или их все. Если вы "запрете" только одну, кто-то может войти и использовать консоль, он просто не сможет использовать вашу vty, пока вы не отопрете ее. vlock распространяется с RedHat Linux, но ваш дистрибутив может отличаться.

Конечно, запирание вашей консоли помешает кое-кому прямо нанести вред вашей работе, однако не помешает перезагрузить вашу машину или каким- либо другим образом разрушить вашу работу. Оно также не предотвратит попыток доступа к вашей машине с других машин в сети и последующих проблем.

3.5 Определение нарушений физической безопасности

Первое, что сразу замечается,- это то, что машина была перегружена. Поскольку Linux надежная и стабильная система, то перегружаться она должна только когда **ВЫ** ее выключаете для обновления ОС, манипуляций с компонентами ПК, или подобных случаях. Если ваша машина была перегружена без вас, включайте сигнал тревоги. Многие из способов, которыми ваша защита может быть сломана, требуют от взломщика перегрузки или выключения атакуемой машины.

Проверьте наличие следов взлома на корпусе и ближнем окружении. Хотя многие взломщики скрывают за собой всякие следы присутствия стирая системные журналы, все таки будет неплохо все осмотреть на предмет каких-либо нарушений.

Вот некоторые вещи, которые нужно проверить в ваших системных журналах:

- Короткие или незаконченные системные журналы.
- Системные журналы содержат странные временные метки.
- Системные журналы содержат неверные права доступа или права собственности.
- Присутствуют записи перегрузки или перезапуска сервисов.
- Отсутствуют системные журналы.
- Точка входа или регистрация su со странного места.

Мы обсудим содержание системных журналов позже.

4. Локальная безопасность

Следующим пунктом рассматриваемым в безопасности вашей системы является защита от атак со стороны локальных пользователей. Сказали ли мы `локальные` пользователи? Вы не ослышались - ДА.

Получение доступа как локальный пользователь - это один из первых шагов, которые попытается сделать взломщик на пути к получению счета администратора. При небрежной локальной безопасности он - взломщик - может затем "перерегистрировать" свой счет с рядового пользователя на администратора используя различные ошибки (bugs) и неправильно настроенные локальные сервисы. Если вы обеспечите достаточный уровень вашей локальной безопасности, то взломщик будет иметь еще один барьер для проникновения.

Локальные пользователи также могут причинить достаточно вреда в вашей системе даже (особенно) если они и являются теми, за кого себя выдают. Предоставлять счета людям, которых вы не знаете или о которых не имеете контактной информации, является очень плохой идеей.

4.1 Создание новых счетов

Вы должны быть уверены, что предоставляете пользователям счета с минимальными допусками, необходимыми для выполняемых ими задач. Если вы даете счет вашему сыну (возраста 10 лет), то вы можете позволить ему доступ к текстовому процессору или графической программе, но не к удалению данных, которые ему не принадлежат.

Существует несколько неписанных правил, которых необходимо придерживаться при предоставлении законного доступа к вашей Linux машине:

- Предоставляйте минимальное количество привилегий.
- Отслеживайте когда/откуда происходит регистрация или ведите журнал.
- Не забудьте удалить счет, если он больше не используется.

Большинство счетов локальных пользователей, которые используются для прорыва системы безопасности, являются счетами, которые не использовались месяцы, а то и годы. Поскольку никто их не использует, они являются идеальным атакующим транспортом.

4.2 Безопасность администратора

Наиболее искомым счетом на вашей машине является счет суперпользователя - администратора (root). Этот счет имеет доступ ко всем ресурсам машины, который также может включать доступ к другим машинам в сети. Помните, что вы должны использовать счет администратора только для очень ограниченного набора определенных задач, а в большинстве случаев должны регистрироваться как обычный пользователь. Работать все время как администратор является ОЧЕНЬ ОЧЕНЬ ОЧЕНЬ ПЛОХОЙ ИДЕЕЙ.

Несколько приемов, чтобы избежать последствий из-за путаницы в счетах, с которыми вы работаете:

- Когда выполняете некоторую комплексную команду, попытайтесь сначала запустить ее в неразрушающем виде ... особенно команды, содержащие заменители (wildmarks): например, вы собираетесь сделать "rm foo*.bak", а вместо этого сначала сделайте "ls foo*.bak" и убедитесь, что вы собираетесь удалить действительно то, что думаете. Также помогает использование подтверждений при выполнении таких команд.

- Некоторые находят полезным делать "touch /-i" в их системах. Это заставит команды типа "rm -rf *" спрашивать вас, действительно ли вы хотите удалить все файлы. (Это происходит таким образом, что ваш shell сначала распознает "-i", и передает ее как опцию для rm). Однако это не поможет для rm команд без * в теле. ;(

- Регистрируйтесь как администратор только для выполнения одиночных специфических задач. Если вы вдруг поймаете себя на том, что вы пытаетесь выяснить как что-то работает или как что-то сделать,- сейчас же перерегистрируйтесь как обычный пользователь и не возвращайтесь к счету администратора пока вы действительно не будете **уверены**, **что** нужно сделать администратору.

- Очень важными являются _пути_по_умолчанию_ администратора. Путь по умолчанию, или переменная окружения PATH, определяет то место, где shell ищет программы. Попытайтесь ограничить пути по умолчанию администратора насколько это возможно и никогда не используйте '.', обозначающую "текущай каталог", в ваших установках PATH. Кроме этого никогда не разрешайте запись в каталоги, прописанные в переменной PATH, поскольку это может позволить взломщику модифицировать существующие или записать новые программы в этих каталогах, разрешив им таким образом запустить эту программу администратору в тот момент, когда ему понадобиться выполнить данную программу.

- Никогда не используйте набор утилит rlogin/rsh/rexec (называемых r-утилитами) будучи администратором. Они являются предметом интереса многих типов взломщиков и являются прямой опасностью при запуске администратором. Никогда не создавайте файл .rhosts будучи администратором.

- Файл /etc/securetty содержит список терминалов, с которых может зарегистрироваться администратор. По умолчанию (в Red Hat Linux) все установлено только на локальные виртуальные консоли (vtys). Будьте очень осторожны добавляя что-либо еще в этот файл. Вы можете зарегистрироваться удаленно как обычный пользователь, а затем использовать 'su', если вам действительно это нужно (полезно через ssh или другой зашифрованный канал), таким образом нет необходимости прямо регистрироваться как администратор.

- Никогда не спешите и обдумывайте каждый шаг работая администратором. Ваши действия могут затронуть многие вещи. Думайте (!!!) прежде чем что-либо выполнить.

Если вам абсолютно положительно необходимо разрешить кому-либо (обычно очень доверенному) иметь доступ как администратор к вашей машине, существует несколько инструментов, которые могут помочь. Sudo позволяет пользователям использовать их пароли для получения доступа к ограниченному набору команд администратора. Это позволит вам, например, разрешить пользователям менять и монтировать сменные диски в вашей системе, но не даст других привилегий. Sudo также ведет журнал всех удачных и неудачных запусков, позволяя вам отслеживать кто и для чего использовал эту команду. Поэтому sudo работает хорошо даже в тех местах, где несколько человек имеют права администратора, - используя возможности sudo, вы можете отследить, какие были сделаны изменения.

Хотя sudo может использоваться для предоставления определенным пользователям определенных привилегий для специфических задач, эта утилита имеет несколько недостатков. Она должна использоваться только для ограниченного набора задач, подобных перезагрузке сервера, или добавления новых пользователей. Любая программа, которая предоставляет возможность выхода из shell, дает пользователю права администратора. Например, это свойственно многим редакторам. Также такие безобидные программы как /bin/cat могут использоваться для перезаписи файлов, которые могут позволить эксплуатировать счет администратора. Рассматривайте sudo как средство учета, и не ожидайте, что заменив им суперпользователя, вы будете в безопасности.

5. Безопасность файлов и файловой системы

Несколько минут подготовки и прогнозного планирования прежде чем открыть вашу систему Интернету может помочь защитить как ее, так имеющиеся в ней данные.

- Нет ни одной причины, по которой нужно было бы разрешать запуск SUID/SGID программ из пользовательских домашних каталогов. Для тех разделов, в которые разрешена запись не только администратору, в /etc/fstab поставьте опцию `nosuid'. Вы также можете захотеть использовать `nodev' и `noexec' для домашних каталогов, а также /var, которые запретят выполнение программ и создание символьных и блочных устройств, которые и так никогда не нужны.

- Если вы экспортируете файловые системы используя NFS, обязательно отконфигурируйте /etc/exports с максимально возможными ограничениями. Это означает не использовать символы подстановки (wildcards), не разрешать запись администратору удаленной системы, а также монтирование с правами "только чтение" где только возможно.

- Настройте umask создания файлов для ваших пользователей настолько ограничивающей насколько это возможно. Общеупотребительными являются 022, 033, и наиболее ограничивающая 077, и добавьте все это к /etc/profile.

- Установите лимит использования файловой системы вместо разрешения неограниченного использования, что установлено по умолчанию. Вы можете контролировать лимиты каждого пользователя используя специальный модуль лимитов ресурсов PAM и /etc/pam.d/limits.conf. Например, лимиты для группы `users' могут выглядеть следующим образом:

  ```
  @users     hard  core    0
  @users     hard  nproc   50
  @users     hard  rss     5000
  ```

  Это запрещает создание core файлов, ограничение количества процессов значением 50, и ограничение использования памяти 5MB на пользователя.

- Файлы /var/log/wtmp и /var/run/utmp содержат записи регистрации для всех пользователей в вашей системе. Их накопление должно поддерживаться постоянно, поскольку их можно использовать для определения когда и откуда пользователь (или потенциальный взломщик) вошел в вашу систему. Эти файлы должны иметь маску прав доступа 644, чтобы не нарушать нормальной работы системы.

- Для предотвращения случайного удаления или перезаписи файлов, которые должны быть защищены, можно использовать имунный бит. Он также предотвращает создание кем бы то ни было символьной связи на этот файл, что является одним из методов атаки с целью удаления /etc/passwd

или /etc/shadow. Смотри руководство по chattr(1) для более детальной информации об имунном (immutable) бите.

- SUID и SGID файлы в вашей системе являются потенциальными носителями риска вашей безопасности, поэтому должны быть под постоянным и тщательным наблюдением. Поскольку эти программы предоставляют специальные привилегии пользователям, которые запускают их, необходимо убедиться, что небезопасные программы не установлены. Любимым приемом кракеров является разработка программ с SUID "root", и затем оставлять их в системе как "черный ход" для получения доступа в следующий раз, даже если изначально использованная "дыра" уже и будет закрыта.

  Найдите все SUID/SGID программы в вашей системе и посмотрите, что они из себя представляют, таким образом вы будете знать, что любое изменение в них является индикатором возможного взлома. Чтобы найти все SUID/SGID программы в вашей системе используйте следующую команду:

  ```
  root#  find / -type f \( -perm -04000 -o -perm -02000 \)
  ```

  Вы можете дискриминативно убрать все SUID или SGID права для всех подозрительных программ используя chmod(1), а затем поставить обратно, если вы будете абсолютно уверены в необходимости этого.

- Файлы с разрешенными для всех правами записи, особенно системные файлы, могут быть "дырой" в безопасности, если взломщик получит доступ к вашей системе и изменит их. Кроме того опасны каталоги с разрешенными для всех правами на запись, поскольку они позволяют взломщику по желанию добавлять или удалять файлы. Для обнаружения в вашей системе всех файлов с разрешенными для всех правами записи выполните следующую команду:

  - ```
    root# find / -perm -2 -print
    ```

  и убедитесь, что вы действительно знаете, почему в эти файлы разрешена запись. В условиях нормальной работы только для некоторых файлов будет разрешена запись, включая некоторые из /dev и символьные ссылки.

- Файлы без владельца также могут быть индикатором внедрения в вашу систему взломщика. Файлы без владельца или с таковым без принадлежности к какой-либо группе можно обнаружить с помощью команды:

  - ```
    root# find / -nouser -o -nogroup -print
    ```

- Обнаружение файлов .rhosts должно быть вашей регулярной обязаностью как системного администратора, поскольку эти файлы ни в коем случае не должны быть в вашей системе. Помните, взломщику нужен только один небезопасный счет для возможного получения доступа ко всей вашей сети. Вы можете обнаружить все файлы .rhosts в вашей системе с помощью команды:

  - ```
    root# find /home -name .rhosts -print
    ```

- И наконец, перед тем как изменить права доступа каких-либо системных файлов, убедитесь, что вы понимаете, что делаете. Никогда не изменяйте права доступа файла только потому, что это является простым способом заставить что-то работать. Прежде чем изменять, всегда определяйте, почему файл имеет именно такие права доступа.

5.1 Установки umask
    ```

Команду umask можно использовать для определения режима создания файлов в вашей системе, принимаемого по умолчанию. Она представляет битовое дополнение до желаемого значения режима файла. Если файлы создаются без какого-либо специального набора прав доступа, то можно случайно разрешить чтение или запись тому, кто не должен иметь таких прав. Типично принятыми umask являются 022, 027 и 077, которые наиболее ограничивающие. Нормальным будет установить значение umask в /etc/profile, так чтобы оно применялось ко всем пользователям в системе. Например, вы можете иметь строку подобную следующей:

```
# Значение umask по умолчанию для всех пользователей
umask 033
```

Убедитесь, что значение umask для администратора составляет 077, что запрещает чтение, запись и выполнения для остальных пользователей, до тех пор, пока это не будет изменено явно командой chmod(1).

Если вы используете Red Hat и придерживаетесь их схемы создания ID пользователя и группы (собственная группа пользователя), то для значения umask необходимо использовать только 002. Это из-за того, что настройки по умолчанию определяют одного пользователя на группу.

5.2 Права доступа файла

Важно убедиться, что ваши системные файлы закрыты для случайного редактирования пользователями и группами, которые не должны выполнять таких действий.

UNIX разделяет контроль доступа к файлам и каталогам по трем принадлежностям: владелец, группа, все остальные. Существует всегда один владелец, любое количество членов группы и еще все остальные.

Быстрое объяснение прав доступа в unix:

Собственность - Какой пользователь(ли) и группа(ы) удерживает контроль установок прав вершины (node) и родителя вершины.

Права доступа - Назначаемые или переназначаемые в битовом выражении установки, которые разрешают некоторый тип доступа к собственности. Права доступа к каталогам могут иметь отличающиеся значения от оных у файлов, содержащихся в них.

Чтение:

- Возможность просмотра содержимого файла
- Возможность чтения каталога

Запись:

- Возможность добавить или изменить файл
- Возможность удалять или перемещать файлы в каталоге

Выполнение:

- Возможность запуска программы или скрипта оболочки (shell script)

- Возможность поиска в каталоге, в комбинации с правом чтения

Save Text Attribute: (Для каталогов)

sticky бит также имеет отличное значение применимо к каталогам. Если sticky бит установлен для каталога, то пользователь может удалять только те файлы, владельцем которых он является, или к которым ему явно заданы права записи, несмотря на то, что ему разрешена запись в этот каталог. Это сделано для каталогов подобных /tmp, в которые разрешена запись всем, но в которых нежелательно разрешать любому пользователю удалять файлы от нечего делать. sticky бит видно как 't' в полном режиме отображения содержимого каталога. (long listing).

SUID Attribute: (Для файлов)

Описывает set-user-id права на файл. Если права доступа set-user-id установлены во "владелец" и файл исполняемый, то процесс, который запускает его, получает доступ к системным ресурсам основываясь на правах пользователя, который создал этот процесс. Во многих случаях это является причиной возникновения 'buffer overflow'.

SGID Attribute: (Для файлов)

Если установлен в правах доступа "группы", этот бит контролирует "set group id" статус файла. При этом он работает также как и SUID, только задействована при этом группа, а не отдельный пользователь.

SGID Attribute: (Для каталогов)

Если вы установите SGID бит для каталога (командой "chmod g+s directory"), то файлы содержащиеся в этом каталоге будут иметь установки группы такие, как у каталога.

Вы - Владелец файла

Группа - Группа, к которой вы принадлежите

Остальные - Любой в системе, кто не является владельцем либо членом группы

Пример файла:

```
-rw-r--r--  1 kevin  users         114 Aug 28  1997 .zlogin
1й бит - каталог?                                  (нет)
 2й бит - чтение владельцем?                       (да, для kevin)
  3й бит - запись владельцем?                      (да, для kevin)
   4й бит - выполнение владельцем?                 (нет)
    5й бит - чтение группой?                       (да, для users)
     6й бит - запись группой?                      (нет)
      7й бит - выполнение группой?                 (нет)
       8й бит - чтение остальными?                 (да, для остальных)
        9й бит - запись остальными?                (нет)
         10й бит - выполнение остальными?          (нет)
```

Следующие строчки являются примером минимальных установок прав, которые требуются для предоставления описанного доступа. Вы можете захотеть предоставить больше прав, чем приведено, но здесь описано, что эти минимальные права на файл разрешают:

```
-r--------  Разрешают чтение файла владельцем
--w-------  Разрешают владельцу изменение или удаление файла
---x------  Владелец может выполнять эту программу, но не скрипты
            командного интерпретатора, которые еще нуждаются в правах
            на чтение
---s------  Будет выполняться с эффективным пользовательским ID = владелец
-------s--  Будет выполняться с эффективным пользовательским ID = группа
-rw------T  Не обновлять "последнего времени изменения". Обычно используется
            для swap файлов
---t------  Не действует.  (прежде sticky бит)
```

Пример каталога:

```
        drwxr-xr-x  3 kevin  users         512 Sep 19 13:47 .public_html/
        1й бит - каталог?                                   (да, содержит много
файлов)
         2й бит - чтение владельцем?                        (да, для kevin)
          3й бит - запись владельцем?                       (да, для kevin)
           4й бит - выполнение владельцем?                  (да, для kevin)
            5й бит - чтение группой?                        (да, для users)
             6й бит - запись группой?                       (нет)
              7й бит - выполнение группой?                  (да, для users)
               8й бит - чтение остальными?                  (да, для остальных)
                9й бит - запись остальными?                 (нет)
                 10й бит - выполнение остальными?           (да, для остальных)
```

Следующие строчки являются примером минимальных установок прав, которые требуются для предоставления описанного доступа. Вы можете захотеть предоставить больше прав, чем приведено, но здесь описано, что эти минимальные права на файл разрешают:

```
dr--------  Можно просмотреть содержание, но нельзя прочесть атрибуты файла
d--x------  Можно войти в каталог, а также использовать его в полной записи
            файла (т.е. с путем к нему)
dr-x------  Владелец может теперь просмотреть атрибуты файла
d-wx------  Файлы теперь можно создавать/удалять, даже если каталог
            не текущий
d------x-t  Предотвращает файлы от удаления остальными с правами записи.
            Используется для /tmp
d---s--s--  Никаких действий
```

Системные конфигурационные файлы (обычно в /etc) имеют обычно режим 640 (что означает -rw-r-----) и являются собственностью администратора. В зависимости от требований безопасности в вашей системе, вы можете изменить эти установки. Никогда не предоставляйте возможности записи в системные файлы для "группы" или "остальных". Некоторые конфигурационные файлы, включая /etc/shadow, должны разрешать чтение только администратору, а каталоги в /etc должны по крайней мере быть недоступны другим.

SUID Shell Scripts

SUID скрипты командного интерпретатора являются также риском безопасности, по этой причине ядро не обслуживает их. Независимо от вашего мнения о том насколько безопасным является скрипт, он может быть переделан для выдачи взломщику оболочки администратора.

5.3 Проверка целостности с помощью Tripwire

Другим хорошим способом обнаружить локальные (а также сетевые) атаки на вашу систему является использование тестеров целестности (integrity checkers) подобных Tripwire. Tripwire вычисляет контрольные суммы для всех важных бинарных и конфигурационных файлов в вашей системе и сравнивает их с предыдущими, хорошо известными, из базы данных. Таким образом любые изменения в файлах будут замечены.

Хорошей идеей будет записать tripwire на дискету, а затем установить на нее защиту от записи. Таким образом взломщик не сможет подделать tripwire или изменить базу данных. Как только вы установили tripwire будет неплохо включить в свои обязанности администратора безопасности проверку с помощью него на предмет каких-либо изменений.

Вы можете даже добавить в список задач crontab запуск tripwire с вашей дискеты каждую ночь и посылку результатов вам по почте утром. Что-то наподобие этого:

```
# установить получателя
MAILTO=kevin
# запустить tripwire
15 05 * * * root /usr/local/adm/tcheck/tripwire
```

будет отсылать вам по почте отчет каждое утро в 5:15.

Tripwire может быть всевышним в обнаружении взломщиков еще до того, как вы заметите их. Как только в системе появится некоторое количество измененных файлов, вы должны понимать, что имеет место деятельность взломщика, и знать, что делать вам самим.

5.4 "Троянские кони"

Термин "Троянский Конь" взят из великого творения Гомера. Идея состоит в том, что вы создаете программу, который чем-либо привлекателен, и каким-либо способом заставляете других людей скачать ее и запустить как администратор. Затем, пока они не разобрались, вы можете разрушить их систему. Пока они думают, что программа, которую они только-что вытянули, делает одну вещь (и может даже очень хорошо), она также разрушает их систему безопасности.

Вы должны быть очень внимательны при установке новых программ на вашу машину. RedHat предоставляет MD5 контрольные суммы и PGP ключи для RPM файлов, так что вы можете проверить действительно ли вы инсталируете реальные продукты. Другие дистрибутивы имеют подобные методы. Вы никогда не должны запускать из под администратора бинарники, для которых у вас нет исходников, или о которых вы ничего не слышали! Немногие взломщики имеют желание выложить на всеобщее обозрение исходный код.

Также может быть общим совет брать исходники некоторых программ с их реальных дистрибутивных серверов. Если программу нужно запускать из под администратора, проверьте исходный код сами или дайте на проверку тому, кому вы доверяете.

6. Безопасность паролей & Шифрование

Одними из наиболее важных свойств безопасности, используемых сегодня, являются пароли. Важно как вам так и вашим пользователям иметь безопасные, не очевидные пароли. Большинство из наиболее последних дистрибутивов Linux включают программу 'passwd', которая не позволит вам установить легко угадываемый пароль. Убедитесь, что ваша программа 'passwd' современна и имеет это свойство.

Глубокое обсуждение шифрования далеко за пределами целей этого документа, однако введение мы сделаем. Шифрование очень полезно, возможно даже необходимо в это время и в этом месте. Существует большое количество разных методов шифрования данных, каждый из которых имеет свой собственный набор характеристик.

Большинство Unix (и Linux не исключение) в основном используют односторонний алгоритм шифрования, называемый DES (стандарт шифрования данных /Data Encription Standard/), для шифрования ваших паролей.

Эти зашифрованные пароли затем сохраняются (обычно) в файле /etc/passwd или (реже) в /etc/shadow. Когда вы пытаетесь зарегистрироваться, все, что вы набираете, снова шифруется и сравнивается с содержимым файла, в котором хранятся ваши пароли. Если они совпадают, должно быть это одинаковые пароли, и вам разрешают доступ. Хотя DES является двухсторонним (вы можете закодировать, а затем раскодировать сообщение, давая верный ключ), большинство Unix используют односторонний вариант. Это значит, что невозможно на основании содержания файла /etc/passwd (или /etc/shadow) провести расшифровку для получения паролей.

Атаки "методом грубой силы", такие как "Взлом" или "John the Ripper" (см. ниже), могут часто угадать ваш пароль, если он не достаточно случает (рандомизирован). PAM модули (см. ниже) позволяют вам использовать различные программы шифрования для ваших паролей (такие как MD5 или подобные).

Вы можете посетить http://consult.cern.ch/writeup/security/security_3.html для получения информации о том, как лучше выбрать пароль.

6.1 PGP и криптование открытым ключом (Public Key Cryptography)

Криптование открытым ключом, подобного как для PGP, происходит таким образом, что шифрование производится одним ключом, а расшифровка - другим. Традиционно в криптографии как для шифрования так и для расшифровки используется один ключ. Этот "личный ключ" (private key) должны знать обе стороны - передающая и получающая - а также кто-то, кто передаст его от одной стороны другой.

Криптование открытым ключом снимает необходимость секретно передавать ключ, который используется для шифрования, использованием двух различных ключей, публичного ключа и личного ключа. Публичный ключ каждого человека доступен любому другому для выполнения шифрования, в тоже время каждый человек имеет его/ее личный ключ для дешифрации сообщений, зашифрованных правильным публичным ключом.

Есть преимущества как в использовании публичного (открытого) ключа, так и криптографии личного ключа. Вы можете почитать о различиях в RSA Cryptography FAQ, приведенном в конце этого раздела.

PGP (Pretty Good Privacy) довольно хорошо поддерживается в Linux. Известно, что хорошо работают версии 2.6.2 и 5.0. Чтобы увидеть хороший пример PGP и как его использовать, почитайте PGP FAQ. http://www.pgp.com/service/export/faq/55faq.cgi Убедитесь, что вы используете версию применимую в вашей стране, поскольку существуют ограничения правительства США на экспорт - сильное шифрование рассматривается как военное оружие и запрещено к распространению в электронной форме за пределами страны. **/Прим. перев. - Вроде как есть сдвиги в этом направлении. Недавно читал, что кто-то там выиграл процесс и получил разрешение на это дело - упор делался на свободу слова/.**

Существует также пошаговое руководство по настройке PGP в Linux, находится по адресу http://mercury.chem.pitt.edu/~angel/LinuxFocus/English/November1997/article7.html Оно написано для международной версии PGP, но оно легко адаптируемо для версии США. Вам также может понадобиться заплатка (patch) для некоторых из последних версий Linux, которая находится на ftp://sunsite.unc.edu/pub/Linux/apps/crypto.

Больше информации по криптографии можно найти в RSA cryptography FAQ, доступном на http://www.rsa.com/rsalabs/newfaq/. Здесь вы найдете информацию по таким терминам как "Diffie-Hellman", "public-key cryptography", "Digital Certificates", и др.

6.2 SSL, S-HTTP, HTTPS и S/MIME

Очень часто пользователи спрашивают о различиях между различными протоколами безопасности и шифрования, и как использовать их. Поскольку это документ не о шифровании, будет неплохой идеей кратко объяснить что из себя каждый из них представляет, и где найти более детальную информацию.

- **SSL:** - SSL, или Secure Sockets Layer, является методом шифрования разработанным Netscape для обеспечения безопасности в Сети. Он поддерживает несколько различных протоколов шифрования, и обеспечивает идентификацию (authentication) как на уровне клиента так и на уровне сервера. SSL работает на транспортном уровне, создает безопасный шифрованный канал данных, и, таким образом, может бесшовно шифровать данные многих типов. Наиболее часто это случается, когда вы посещаете защищенный узел для просмотра в режиме online секретного документа с помощью Communicator, который обеспечивает вас базовыми услугами безопасности связи, а также многими другими видами шифрования данных. Больше информации можно найти на http://www.consensus.com/security/ssl-talk-faq.html. Информация о других реализациях безопасности в Netscape, а также хорошая отправная точка по этим протоколам доступа по http://home.netscape.com/info/security-doc.html.

- **S-HTTP:** - S-HTTP является еще одним протоколом, который реализует в Интернете сервис безопасности. Он был разработан для предоставления конфиденциальности, опознавания, сохранности, а также non-repudiability [не спутайте с чем-либо еще], в то же время имея механизмы управления многими ключами и криптографические алгоритмы путем выборочного согласования между участниками в каждой транзакции. S-HTTP ограничен специфическим программным обеспечением, которое реализует его, и шифрует каждое сообщение индивидуально. [Из RSA Cryptography FAQ, стр. 138]

- **S/MIME:** - S/MIME, или Secure Multipurpose Internet Mail Extension, является стандартом шифрования, используемым в электронной почте, или других типах сообщений в Интернете. Это открытый стандарт, который разработан RSA, и поэтому очень вероятно, что мы скоро увидим его в Linux. Больше информации по S/MIME можно найти по адресу http://home.netscape.com/assist/security/smime/overview.html.

6.3 Реализация IPSEC в x-ядре Linux

Наряду с CIPE и другими формами шифрования данных, существует также реализация IPSEC для Linux. IPSEC создан усилиями IETF для обеспечения криптографически безопасных соединений на уровне IP сети, который также предоставляет опознавание, сохранность, контроль доступа и конфиденциальность. Информацию по IPSEC и черновикам Интернета можно найти в http://www.ietf.org/html.charters/ipsec-charter.html. Там вы также можете найти ссылки на другие протоколы использующие управление ключами, на список рассылки и архивы IPSEC.

Реализация для Linux, которая была разработана в Университете Аризоны, использует объектно-ориентированную структуру для реализации сетевого протокола называемую x-ядро. Детальнее на http://www.cs.arizona.edu/xkernel/hpcc-blue/linux.html. В двух словах, x-ядро является методом передачи сообщений на уровне ядра, что позволяет более простую реализацию.

Как и с другими формами криптографии этот метод не распространяется с ядром из-за ограничений на экспорт.

6.4 SSH (Secure Shell), stelnet

SSH и stelnet - это программы, которые позволяют вам зарегистрироваться на удаленном сервере и иметь шифрованное соединение.

SSH является набором программ используемым как более безопасный заменитель для rlogin, rsh и rcp. Он использует криптографию открытого ключа для шифрования соединения между двумя машинами, а также для опознавания пользователей. Его можно использовать для безопасной регистрации на удаленном сервере или копировании данных между двумя машинами, в то же время предотвращая атаки путем присоединения посредине (session hijacking) и обманом сервера имен (DNS spoffing). Он предоставляет компрессию данных в вашем соединение и безопасное X11 соединение между двумя машинами. Домашнюю Web страницу SSH можно найти по адресу http://www.cs.hut.fi/ssh/

Вы также можете использовать SSH с вашей рабочей станции под Windows обращаясь к вашему Linux SSH серверу. Существует несколько бесплатных реализаций Windows клиентов, включая http://guardian.htu.tuwien.ac.at/therapy/ssh/, а также коммерческую реализацию от DataFellows, на http://www.datafellows.com.

SSLeay является бесплатной реализацией протокола Secure Sockets Layer от Netscape и состоит из нескольких приложений, таких как Secure telnet, модуль для Apache, нескольких баз данных, а также нескольких алгоритмов, включая DES, IDEA и Blowfish.

Используя эту библиотеку был создан secure telnet, который выполняет шифрование через telnet соединение. В противовес SSH, stelnet использует SSL, Secure Sockets Layer протокол разработанный Netscape. Вы можете найти Secure telnet и Secure FTP начав с SSLeay FAQ, доступного на http://www.psy.uq.oz.au/~ftp/Crypto/

6.5 PAM - Pluggable Authentication Modules

Новые версии дистрибутива Red Hat распространяются с унифицированной схемой идентификации, называемой "PAM". PAM позволяет вам налету изменять ваши методы идентификации, требования, инкапсулировать все ваши локальные методы идентификации без перекомпиляции ваших программ. Описание настройки PAM выходит за рамки этого документа, поэтому за более детальной информацией сходите на web узел PAM. http://www.kernel.org/pub/linux/libs/pam/index.html

Вот несколько вещей, которые вы можете делать с PAM:

- Использовать не-DES шифрование для ваших паролей. (делая их более устойчивыми к взлому методом "грубой силы")
- Устанавливать лимиты на ресурсы для ваших пользователей, чтобы они не могли выполнить сервисную атаку (количество процессов, количество памяти, и т.п.)
- На лету активизировать теневые пароли (shadow password) (см. ниже)

- Разрешать определенным пользователям регистрироваться только в определенное время и/или с определенного места

За несколько часов установки и настройки вашей системы вы можете предотвратить много атак еще до их возниконовения. Например, используйте PAM для запрещения широкого использования в системе файлов .rhosts в домашних каталогах пользователей добавлением этих строк к /etc/pam.d/login:

```
#
# Запретить для пользователей rsh/rlogin/rexec
#
login auth required pam_rhosts_auth.so no_rhosts
```

6.6 Криптографическая IP инкапсуляция (CIPE)

Главной целью этого программного обеспечения является предоставление средств для безопасной (против подслушивания, включая анализ трафика, и подставления поддельных сообщений) связи между подсетями через небезопасные пакетные сети, такие как Интернет.

CIPE шифрует данные на сетевом уровне. Шифруются пакеты, которые передаются между компьютерами в сети. Шифрующий код помещается недалеко от драйвера, который посылает и принимает пакеты.

Это не схоже с SSH, который шифрует данные по соединениям - на гнездовом уровне. В этом случае шифруется логическое соединение между программами, запущенными на разных машинах.

CIPE можно также использовать при тунелировании (tunnelling) для создания Виртуальных Частных Сетей (Virtual Private Networks). Преимущество низкоуровневого шифрования состоит в том, что оно позволяет прозрачную работу между двумя сетями, соединенными в VPN, без каких-либо изменений в программном обеспечении.

Выдержка из документации по CIPE:

IPSEC стандарты определяют набор протоколов, которые можно использовать (среди прочих) для построения шифрованных VPN. Однако, IPSEC является скорее тяжеловесным и сложным с большим количеством опций, реализация полного набора протоколов все еще редко используется и некоторые вещи (такие как управление ключами) еще не до конца решены. CIPE использует более простой подход, в котором многие вещи, которые можно параметризовать (такие как выбор текущего алгоритма шифрования), устанавливаются единожды во время инсталляции. Это ограничивает гибкость, но позволяет более простую (и поэтому эффективную, простую в отладке) реализацию.

Дальнейшую информацию можно найти на http://www.inka.de/~bigred/devel/cipe.html

Также как и с другими формами криптографии, он не распространяется с ядром по умолчанию ввиду экспортных ограничений.

6.7 Kerberos

Kerberos является идентификационной системой, разработанной по проекту Athena в MIT. Во время регистрации пользователя, Kerberos идентифицирует его (используя пароль) и предоставляет пользователю способ доказать его идентичность другим серверам и компьютерам разбросанным в сети.

Эта идентификация затем используется программами, такими как rlogin, для разрешения пользователю регистрации на других компьютерах без пароля (в месте .rhosts файла). Идентификация также используется почтовой системой для того чтобы гарантировать, что почта доставлена правильному адресату, а также для гарантии того, что посылающий является тем за кого себя выдает.

Общий эффект использования Kerberos и других программ, которые поставляются вместе с ним, состоит в сущности в полном исключении какой-либо возможности пользователям обмануть систему по поводу своей принадлежности. К сожалению, установка Kerberos довольно трудоемкая, требующая модификации или замены большого количества стандартных программ.

Вы можете найти больше информацию по Kerberos на http://www.veritas.com/common/f/97042301.htm а сам пакет на http://nii.isi.edu/info/kerberos/

[From: Stein, Jennifer G., Clifford Neuman, and Jeffrey L. Schiller. "Kerberos: An Authentication Service for Open Network Systems." USENIX Conference Proceedings, Dallas, Texas, Winter 1998.]

6.8 Теневые пароли (Shadow passwords)

Теневые пароли означают сокрытие секретной информации о ваших шифрованных паролях от обыкновенных пользователей. Обычно эти шифрованные пароли находятся у вас в /etc/passwd и открыты всем для чтения. Таким образом на этот файл можно напустить программу-расшифровщик, чтобы попытаться определить значения паролей. Пакет shadow записывает информацию о паролях в файл /etc/shadow, который могут читать только привилегированные пользователи. Для того, чтобы активизировать теневые пароли вам необходимо убедиться, что все ваши утилиты, которым необходим доступ к паролям, скомпилированы с поддержкой теневых паролей. PAM (см. выше), кстати, позволяет вам просто подключить shadow модуль и не требует перекомпиляции программ. Вы можете также почитать Shadow-Password HOWTO для получения более детальной информации, если это вам конечно нужно. Он доступен на http://sunsite.unc.edu/LDP/HOWTO/Shadow-Password-HOWTO.html Сейчас он скорее условный и не требуется в дистрибутивах, поддерживающих PAM.

6.9 "Crack" и "John the Ripper"

Если по какой-либо причине ваша программа passwd не может отслеживать легко узнаваемые пароли, вы можете использовать взламывающую пароли программу, чтобы убедиться в безопасности паролей ваших пользователей.

Взламывающие пароли программы основаны на простой идее. Они перебирают каждое слово и его вариации из словаря. Они зашифровывают это слово и сравнивают его с вашим зашифрованным паролем. Если они совпадают, значит задача выполнена.

Существует целый ряд таких программ... наиболее заметные из них это "Crack" and "John the Ripper" http://www.false.com/security/john/index.html . Конечно, они заберут много вашего процессорного времени, но вы сможете с уверенностью сказать, сможет ли взломщик с помощью них получить ваши пароли, - сначала себе, а затем и пользователям указать слабые пароли. Заметьте, что взломщик для получения passwd должен

был бы сначала использовать другие дыры в системе, но это уже более широкий вопрос, чем вы можете подумать.

6.10 CFS - криптографическая файловая система и TCFS - прозрачнаякриптографическая файловая система

CFS - это метод шифрования всей файловой системы, который позволяет пользователям сохранять в ней зашифрованные файлы. Он использует NFS сервер, запущенный на локальной машине. RPMS доступен на http://www.replay.com/redhat/ и больше информации о том как это работает на: ftp://ftp.research.att.com/dist/mab/

TCFS является улучшенным вариантом CFS, поскольку более интегрирован с файловой системой, и, таким образом, прозрачен для всех пользователей, использующих зашифрованную файловую систему. Более детально на: http://edu-gw.dia.unisa.it/tcfs/

6.11 X11, SVGA и экранная безопасность

X11

Очень важно для вас защитить ваш графический экран, чтобы предотвратить взломщика от действий подобных: воровству вашего пароля во время набора без вашего ведома, чтению документов или информации, оставленной вами на экране, или даже использованию дыр для получения прав суперпользователя. Запуск удаленных X приложений через сеть также может быть чреват опасностями, давая возможность взломщику (зд. sniffer) перехватить ваше взаимодействие с удаленным компьютером.

X имеет целый ряд механизмов контроля доступа. Наиболее простой из них - машинозависимый (host based). Вы можете использовать xhost для определения тех машин, с которых разрешен доступ к вашему экрану. Но в общем это не очень безопасный метод. Если кто-то имеет доступ к вашей машине, он может выполнить xhost + его машина и, таким образом, легко войти. Также, если вам нужно разрешить доступ с ненадежной машины, любой может подвергнуть риску ваш дисплей.

Если для регистрации используется xdm (x display manager), вы получаете намного лучший метод доступа: MIT-MAGIC-COOKIE-1. Генерируется 128-битный cookie и сохраняется в вашем файле .Xauthorty. Если вы хотите удаленной машине разрешить доступ к вашему дисплею, то для предоставления доступа именно этому соединению вы можете использовать команду xauth и информацию из вашего файла .Xauthority. Посмотрите также Remote-X-Apps mini-howto, доступном на http://sunsite.unc.edu/LDP/HOWTO/mini/Remote-X-Apps.html.

Вы можете также использовать ssh (см. ssh выше) для разрешения безопасных X соединений. Это имеет также преимущество, поскольку прозрачно конечному пользователю, и означает то, что не зашифрованные данные не передаются по сети.

Загляните также в Xsecurity страничку man для более детального описания безопасности в X. Безопасным будет использовать xdm для регистрации на вашей консоли, а затем использовать ssh для перехода на удаленную машину, с которой вы хотите запустить X программу.

SVGA

Программы, основанные на SVGAlib, обычно являются SUID-root, для того чтобы иметь доступ ко всем видео-ресурсам вашего компьютера. Это делает их очень опасными. Если они дают сбой, то обычно вам нужно перезагрузить компьютер, чтобы опять получить доступ к консоли. Убедитесь, что все SVGA программы, которые вы запускаете, подлинны, и как минимум такие, которым вы доверяете. А лучше - не запускайте их вообще.

GGI (проект Общего Графического Интерфейса)

Проект GGI для Linux является попыткой решить несколько проблем с видео интерфейсом в Linux. GGI будет передавать небольшие куски видео-кода в ядро Linux, и таким образом контролировать доступ к видео системе. Это значит, что GGI будет способен восстановить вашу консоль в любое время к известному рабочему состоянию. Он также позволит использовать ключ безопасности (secure attention key), так что вы сможете быть уверены, что на вашей консоли нет ни одного запущенного "Троянского коня", пытающегося зарегистрироваться. http://synergy.caltech.edu/~ggi/

7. Безопасность ядра

Здесь описываются опции конфигурации ядра, которые относятся к безопасности, а также объясняется, что они делают, и как их использовать.

Поскольку ядро контролирует поведение вашего компьютера в сети, очень важно, чтобы ядро было само очень безопасно, и его нельзя было каким-либо образом взломать или подвергнуть риску. Чтобы предотвратить некоторые из последних известных методов сетевых атак, вы должны использовать последние стабильные версии ядра. Вы можете найти их на ftp://ftp.kernel.org

7.1 Опции компиляции ядра

- IP: Drop source routed frames (CONFIG_IP_NOSR)

  Эта опция должна быть включена. Пришедшие кадры (Source routed frames) содержат полный путь их назначения внутри пакета. Это значит, что маршрутизатору, через который проходят пакеты, не нужно их проверять, а просто передавать их дальше. В противном случае это могло бы привести к тому, что входящие в вашу систему данные имели бы потенциальную возможность деструктивных действий.

- IP: Firewalling (CONFIG_IP_FIREWALL)

  Эта опция необходима если вы намереваетесь сконфигурировать вашу машину как щит (firewall), настроить маскарад (masquerading), или хотите защитить вашу станцию с комутирующимися соединениями от кого-либо, желающего проникнуть через ваш PPP dial-up интерфейс.

- IP: forwarding/gatewaying (CONFIG_IP_FORWARD)

Если вы включили IP forwarding, ваша Linux станция в сущности стала маршрутизатором. Если ваша машина в сети, то вы можете теперь ретранслировать данные из одной сети в другую, и можете, таким образом, разрушить существующий щит, поставленный именно для того, чтобы этого не происходило. Обыкновенным dial-up пользователям лучше выключить это, а другим нужно сконцентрироваться на безопасной реализации этой опции. В компьютере-щите эту опцию нужно активировать и использовать вместе с программным обеспечением реализующим щит (firewall).

Вы можете включить либо выключить IP forwarding динамически, используя команду для включения:

```
root#  echo 1 > /proc/sys/net/ipv4/ip_forward
```

и для выключения

```
root#  echo 0 > /proc/sys/net/ipv4/ip_forward
```

Этот файл (и много других файлов в /proc) всегда отображаются с нулевой длиной, но на самом деле это не так. Это новое свойство ядра, так что убедитесь, что ваше ядро имеет версию 2.0.33 или выше.

- IP: firewall packet logging (CONFIG_IP_FIREWALL_VERBOSE)

Эта опция дает вам информацию о пакетах, которые приходят на ваш щит, как то отправитель, получатель, порт и т.п.

- IP: always defragment (CONFIG_IP_ALWAYS_DEFRAG)

Обычно эта опция выключена, но если вы создаете систему-щит или настраиваете маскарад (masquerading), вам желательно включить ее. Когда данные посылаются с одной системы в другую, передача не всегда происходит одним пакетом, скорее всего данные фрагментированы на несколько частей. Проблема при этот состоит в том, что информация о номере порта, сохраняется только в первом фрагменте. Это означает, что кто-то может вставить чужеродную информацию в оставшиеся фрагменты в вашем соединении, которая вообще там не предполагалась.

- IP: syn cookies (CONFIG_SYN_COOKIES)

SYN атака генерирует "отказ в предоставлении сервиса" (DoS), который поглощает все ресурсы вашей системы, что в результате приводит к перезагрузке. Поэтому мы не видим ни одной причины, по которой эту опцию не нужно было бы включать.

- Packet Signatures (CONFIG_NCPFS_PACKET_SIGNING)

Эта опция доступна в ядрах серии 2.1, которая активирует подпись NCP пакетов для большей безопасности. Обычно вы можете оставить ее выключенной, но если она вам понадобиться, то она есть.

- IP: Firewall packet netlink device (CONFIG_IP_FIREWALL_NETLINK)

Это действительно искусная опция, которая позволяет вам проанализировать первые 128 байтов в пакетах от пользовательских программ, и определить на основании их достоверности принять или отклонить пакет.

7.2 Устройства ядра

Существует несколько блочных и символьных устройств в Linux, которые также помогут вам в вопросах безопасности.

Два устройства, /dev/random и /dev/urandom, предоставляются ядром для получения в любой момент времени случайных чисел.

И /dev/random и /dev/urandom должны быть достаточно безопасны, чтобы использоваться в генераторах PGP ключей, SSH вызовах, и других приложениях, в которых используются случайные числа. Взломщик не должен иметь возможности предугадать следующее число, выданное любой начальной последовательностью чисел из этих генераторов. Было приложено огромное количество усилий для обеспечения того, чтобы числа, которые вы получаете от этих генераторов, были случайны в полном смысле слова "случайный".

Разница состоит только в том, что /dev/random оканчивается на случайном байте и это заставляет вас больше ждать до полного накопления. Заметьте, что в некоторых системах, это может на длительное время заблокировать ввод при генерации в системе записи о новом пользователе. Поэтому вы должны с осторожностью использовать /dev/random. (Возможно наилучшим будет использовать его, когда вы генерируете чувствительную к регистру (клавиши) информацию, и вы говорите пользователю постоянно стучать по клавишам, пока вы не выдадите "ОК, достаточно")

/dev/random является высококачественной энтропией, генерируемой из измерения времени внутренних прерываний, или чего-то в этом роде. Он блокируется до тех пор, пока не наберется достаточно бит случайных данных.

Работа /dev/urandom подобна, но когда памяти под энтропию становиться мало, он возвращает криптографически надежные случайные данные (hash) того что есть на момент останова. Это не настолько же безопасно, но достаточно для большинства приложений.

Вы можете читать с этих устройств используя что-то наподобие:

```
root#  head -c 6 /dev/urandom | uuencode -
```

Эта команда выдаст 6 случайных символов на консоль, - удобно для генерации пароля.

Если вы хотите узнать алгоритм, то загляните в /usr/src/linux/drivers/char/random.c.

Спасибо Theodore Y. Ts'o, Jon Lewis, и другим ядерщикам Linux за помощь мне (Dave) в этом вопросе.

8. Безопасность сети

Безопасность сети становиться все более и более важной, поскольку люди все больше и больше времени проводят в сети. Прорвать безопасность сети часто проще нежели физическую или локальную безопасность, и это является намного более обыкновенным событием.

Существует большое количество хороших инструментов для поддержки безопасности сети, и все больше и больше из них поставляются с дистрибутивами Linux.

8.1 Пакетные ищейки (Packet Sniffers)

Одним из наиболее общих методов, которые взломщики могут использовать для получения доступа к многим машинам в вашей сети, является применение пакетного ищейки с уже взломаных машины. Этот "ищейка" просто слушает Ethernet порт на предмет наличия "Password", "Login" или "su" в потоке пакетов и записывает в журнал всю информацию, идущую следом. В такой способ взломщик получает пароли систем, которые он даже и не пробовал пока взламывать. Очень уязвимы к этому виду атаки не зашифрованные пароли, которые передаются простым текстом.

Пример: на компьютере А была взломана система безопасности. Взломщик инстралировал ищейку. Ищейка записал процесс регистрации администратора с компьютера В на компьютер Б. Таким образом он получил персональный пароль системного администратора для регистрации на Б. Затем для решения своих задач администратор набирает "su". Таким образом взломщик получает администраторский пароль компьютера Б. Позже администратор разрешает кому-то запустить telnet из его счета на компьютер Г в другой сети. Теперь взломщик знает пароль/счет на компьютере Г.

В наше время для выполнения подобных операций взломщику даже не нужно взламывать какую-либо систему, он может просто принести ноутбук или ПК в здание и присоединиться к вашей сети.

Использование ssh или других методов шифрования паролей срывает подобные атаки. Для POP счетов мешают проведению таких атак пакеты подобные APOP. (Обычная pop регистрация беззащитна от подобных атак, поскольку как и все остальное пароли по сети передаются открытым текстом.)

8.2 Системные сервисы и tcp_wrappers

Как только вы подключаете вашу Linux систему к ЛЮБОЙ сети, вам сразу же нужно решить, какие сервисы предоставлять. Сервисы, которые вы не будете предоставлять, должны быть выключены, чтобы у вас было меньше вещей, о которых вам нужно беспокоиться, и взломщику будет меньше мест для выискивания дыр.

Существует много способов выключить сервисы в Linux. Вы можете посмотреть в файле /etc/inetd.conf, какие сервисы у вас предоставляются через inetd. Чтобы выключить все, что вам ненужно, просто закоментируйте соответствующие строчки, а затем пошлите вашему inetd SIGHUP (прим. перев. killall -HUP inetd).

Вы также можете удалить (или закомментрировать) соответствующие сервисы в файле /etc/services. Это означает, что локальный клиент также не сможет использовать эти сервисы (например, если вы удалите ftp, а затем попробуете сделать ftp связь с этой машины на удаленный компьютер, вы получите ошибку типа "неизвестный сервис"). Обычно не стоит удалять сервисы, если это не приносит дополнительного повышения уровня безопасности. Если локальный пользователь хочет использовать ftp в том случае, когда вы его уже закомментировали, он может создать своего собственного клиента, который будет использовать общий ftp порт и отлично работать.

Вот некоторые из сервисов, которые вам нужно оставить включенными:

- ftp
- telnet
- mail, такие как pop-3 или imap
- identd
- time

Если вы знаете, что вы не собираетесь использовать какие-то пакеты, лучше их полностью удалить. В дистрибутиве RedHat полностью удаляет пакет команда rpm -e. В Debian подобные вещи делает dpkg.

Дополнительно вам действительно лучше в файле /etc/inetd.conf выключить rsh/rlogin/rcp, включая login (используется rlogin), shell (используется rcp) и exec (используется rsh). Эти протоколы чрезвычайно небезопасны и часто были в прошлом причиной взломов.

Вы должны также проверить ваши /etc/rc.d/rcN.d, где N стартовые уровни вашей системы, на предмет наличия сервисов в этих каталогах, которые вам не нужны. Файлы в /etc/rc.d/rcN.d фактически являются символьными ссылками на файлы в каталоге /etc/rc.d/init.d. Переименование файлов в каталоге init.d выключит все символьные ссылки в /etc/rc.d/rcN.d. Если вы хотите выключить сервис только в определенном стартовом уровне, то переименуйте соответствующий файл, чтобы он начинался с маленькой буквы "s"?, а не с большой как надо (скажем S45dhcpd).

Если у вас rc файлы в стиле BSD, вам нужно проверить /etc/rc* для обнаружения ненужных программ.

Большинство дистрибутивов Linux поставляется с tcp_wrapper, которые "заворачивают" все ваши tcp сервисы. tcp_wrapper (tcpd) вызывается из inetd, а не является отдельным сервером. tcpd затем проверяет компьютер, который запрашивает сервис, и либо запускает реальный сервер либо запрещает доступ от этого компьютера. tcpd позволяет вам ограничить доступ к вашим tcp сервисам. Вы можете создать /etc/hosts.allow и добавить в него только те машины, которым нужно иметь доступ к сервисам на вашем компьютере.

Если вы являетесь домашним пользователем с коммутируемым доступом, то мы рекомендуем вам запретить доступ всем (deny ALL). tcpd также протоколирует все неудачные попытки доступа к сервисам, так что это позволят отследить возможные атаки. Если вы добавляете новые сервисы, вы обязательно должны сконфигурировать их, чтобы использовать основываясь на tcp_wrappers. Например, обыкновенный dial-up пользователь может запретить доступ к своему компьютеру извне, и в то же время иметь возможность забирать почту и путешествовать в интернете. Чтобы это сделать, вам нужно добавить к файлу /etc/hosts.allow:

ALL: 127.

И конечно же /etc/hosts.deny должен содержать:

ALL: ALL

что запретит внешние соединения к вашей машине, позволяя тем не менее вам изнутри соединяться с серверами в Интернете.

8.3 Проверьте вашу DNS информацию

Поддержка постоянно свежей DNS информации о всех компьютерах в вашей сети может помочь повысить безопасность. В том случае, когда несанкционированный компьютер подключится к вашей сети, вы можете опознать его по неудачному запросу к DNS. Большинство сервисов можно сконфигурировать таким образом, чтобы они не принимали запросы на соединение от компьютеров без правильной DNS информации.

8.4 identd

identd маленькая программка, которой обычно оканчивается ваш inetd. Она записывает информацию о том, какой пользователь запускает какой tcp сервис, а затем выдает отчет тому, кто запрашивает.

Многие люди не понимают полезность identd, поэтому выключают ее либо блокируют все внешние запросы к ней. identd не та вещь, которая поможет удаленным компьютерам. Не существует способа узнать, корректна ли информация, которую вы получили от удаленного identd. В identd запросах нет идентификации.

Тогда зачем же нужно вам ее запускать? Потому что она помогает **ВАМ** являясь еще одним инструментом отслеживания ситуации. Если ваш identd не взломан, тогда вы знаете, что он выдает удаленным компьютерам имена пользователей или uid пользователей, используя tcp сервисы. Если администратор удаленной системы придет к вам и скажет, что такой-то пользователь так-то пытался проникнуть в его систему, вы легко можете предпринять действия против такого пользователя. Если вы не включили identd, вам нужно просмотреть много протоколов, чтобы узнать, кто был в то время, и вообще потратить много времени, чтобы вычислить пользователя.

identd, который поставляется с большинством дистрибутивов, намного более настраиваем, нежели многие думают. Вы можете закрыть identd для определенных пользователей (можно создать файл .noident), вы можете протоколировать все запросы к identd (я рекомендую это), вы можете даже заставить identd возвращать uid вместо имени пользователя, или даже NO-USER.

8.5 SATAN , ISS и другие сетевые сканеры

Существует много различных программных пакетов, которые выполняют сканирование портов или сервисов в компьютерах или сетях. SATAN и ISS являются двумя наиболее известными из них. Эти программы соединяются с целевым компьютером (или всеми целевыми машинами в сети) по всем доступным портам и пытается определить, какие там запущены сервисы. Основываясь на этой информации вы можете обнаружить уязвимые к определенным методам атаки машины.

SATAN (Инструмент администратора безопасности для анализа сетей) является сканером портов с web интерфейсом. Он может быть полезен для выполнения легкой, средней или тщательной проверки машины или сети машин. Неплохо иметь SATAN и сканировать вашу систему или сеть, и сразу же устранять обнаруженные им проблемы. Убедитесь, что ваша копия SATAN из sun-site или известного FTP или Web

сервера. Были троянские копии SATAN, которые распространялись по Сети.
http://www.trouble.org/~zen/satan/satan.html

ISS (Сканер безопасности интернета) является также сканером портов. Он быстрее чем SATAN, и таким образом может быть лучше для больших сетей. Однако SATAN предоставляет больше информации.

Abacus-Sentry является коммерческим сканером портов с www.psionic.com. Для получения большей информации сходите на их домашнюю web страничку http://www.psionic.com

Обнаружение сканеров портов

Существуют некоторые инструменты, которые призваны предупредить вас об работающих SATAN, ISS и других сканирующих программах. Однако используя tcp_wrapper, регулярно проверяя ваши протоколы, вы и сами заметите такие попытки. Даже при наименьших установках SATAN оставляет следы присутствия в журналах системы, оборудованной RedHat.

8.6 Sendmail, qmail и MTA

Одним из наиболее важных сервисов, которые вы можете предоставлять, является сервер электронной почты. К сожалению он также наиболее уязвим к атакам, просто из-за огромного числа задач, которые он должен выполнять, и привилегий, которые ему обычно нужны.

Если вы используете sendmail, очень важно иметь самую последнюю версию. Sendmail имеет очень длинную историю развития безопасности. Всегда используйте только последнюю версию. http://www.sendmail.org

Если вы устали модернизировать ваш sendmail каждую неделю, вы можете решить перейти на qmail. qmail изначально разрабатывали подразумевая безопасность. Он быстрый, стабильный и безопасный.
http://www.qmail.org

8.7 "Отказ в предоставлении сервиса"

Атака "Отказ в предоставлении сервиса" состоит в том, что взломщик пытается искусственно загрузить некоторые сервисы настолько, чтобы они не могли отвечать на законные запросы или запрещали доступ к вашей машине законным пользователям.

В последние годы количество атак данного типа очень сильно возросло. Некоторые из наиболее популярных и свежих перечислены ниже. Имейте ввиду, что все время обнаруживаются новые, так что здесь приведены только примеры. Для получения последней информации читайте списки рассылки Linux security и bugtraq, а также архивы.

- **SYN Flooding** - SYN flooding является сетевой атакой "отказ в предоставлении доступа". Он использует преимущества "лазейки" (loophole) в методе создания TCP соединения. Последние версии ядер Linux (2.0.30 и выше) имеют несколько конфигурационных настроек для предотвращения SYN Flooding атак. Смотрите раздел "Безопасность ядра" для более детальной информации о настройках.

- **Ошибка "F00F" в процессорах Pentium** - Было обнаружено, что данная серия ассемблерного кода, посланная настоящему процессору Intel Pentium, перегружает машину. Это действует на все компьютеры с процессорами Pentium (не клонами, не Pentium Pro или PII), не зависимо от операционной системы на этом компьютере. Ядра Linux выше 2.0.32 содержат код, отслеживающий эту ошибку и не позволяющий перегружать вашу машину. Ядро 2.0.33 имеет улучшенный вариант решения этой ошибки, поэтому более рекомендуем нежели 2.0.32. Если у вас Pentium, лучше вам модернизироваться сейчас.

- **Ping Flooding** - Ping flooding является простой грубой реализацией атаки "отказ в предоставлении сервиса". Взломщик посылает "поток" ICMP пакетов вашему компьютеру. Если это происходит с машины с большей полосой пропускания нежели имеет ваш компьютер, то ваша машина будет лишена возможности посылать что-либо в сеть. При вариации этой атаки, называемой "smurfing", посылается на определенный сервер поток ICMP пакетов с обратным IP адресом **вашей машины**, таким образом атакующих тяжелее обнаружить. Более детальная информация о "smurf" атаках представлена на http://www.quadrunner.com/~chuegen/smurf.txt

  Если вы подверглись атаке типа ping flood, то для обнаружения машины, с которой пришли пакеты (или откуда они появляются), используйте инструмент типа tcpdump, и затем обратитесь с этой информацией к вашему провайдеру. Ping flood легко можно остановить на уровне маршрутизатора или используя щит (firewall).

- **Ping o' Death** - Атака Ping o' Death возникла в результате того, что поступающие пакеты ICMP ECHO REQUEST могут быть больше нежели может вместить структура данных ядра, которая сохраняет эту информацию. Из-за приема единичного большого (65,510 байт) "ping" пакета многие системы зависали или даже ломались, поэтому эта проблема быстро обрела название "Ping o' Death." Вообще-то эта ошибка давно уже исправлена, так что не о чём беспокоится.

- **Teardrop / New Tear** - Это одна из недавних еще атак основана на ошибке, присутствующей в коде фрагментации IP в Linux и Windwos платформах. Она исправлена в ядре 2.0.33 и не требует включения какой-либо дополнительной опции во время компиляции ядра. Так что Linux очевидно больше не подвержен атаке "newtear".

Вы можете найти большинство из исследованного кода и детальное описание найденных ошибок на http://www.rootshell.com используя их поисковую систему.

8.8 Безопасность NFS (сетевой файловой системы)

NFS является очень широко используемым протоколом совместного использования файлов. Он позволяет серверам запуская nfsd и mountd "экспортировать" целые файловые системы для других машин со встроенной в ядро поддержкой nfs (или поддержки некоторых других клиентов, если это не Linux машины). Mountd ведет журнал примонтированых файловых систем в /etc/mtab и может выдать их по команде showmount.

Многие сервера используют NFS для предоставления пользователям домашних каталогов, так что не имеет значения на какой из машин в кластере пользователи регистрируются, они сразу получают все свои файлы.

Существуют довольно небольшие возможности реализации безопасности в экспортируемых файловых системах. Вы можете с помощью nfsd приравнять администратора удаленной системы к пользователю nobody (т.е. с минимальными правами) на вашей системе, запрещая ему тем самым полный доступ к экспортируемым файлам. Однако, поскольку конкретные пользователи имеют полный доступ к их собственным файлам (или по крайней мере с одинаковым uid), то удаленный администратор может зарегистрироваться или сделать su к их счетам, и таким образом получить доступ к их файлам. Это только

небольшое препятствие для взломщика, чтобы получить доступ для монтирования вашей удаленной файловой системы.

Если вы вынуждены использовать NFS, то прежде всего убедитесь, что вы экспортируете только тем машинам, которым это действительно нужно. Никогда не экспортируйте полностью ваш root каталог, экспортируйте только те каталоги, которые необходимо.

Для более детальной информации смотрите NFS HOWTO: NFS HOWTO

8.9 NIS (сетевой информационный сервис) (бывший YP)

Сетевой информационный сервис (бывший YP - желтые страницы) заключается в распространении информации группе машин. NIS мастер (сервер) хранит информационные таблицы и конвертирует их файлы карт NIS. Затем эти карты передаются по сети, позволяя NIS клиентам (компьютерам) получать имя счета, пароль, домашний каталог и информацию shell (фактически всю информацию стандартного файла /etc/passwd). Это позволяет пользователю изменить пароль за один раз на всех машинах в NIS домене, где он имеет счет.

NIS совсем небезопасен. Он никогда и не предполагался быть таким. Он предполагался быть удобным и полезным. Любой, кто может угадать имя вашего NIS домена (где-либо в сети) может получить копию вашего файла passwd, а затем использовать "crack" и "john the ripper" для взламывания паролей ваших пользователей. Также можно обманывать NIS и проводить другие подобные трюки. Если вы вынуждены использовать NIS, помните об опасностях связанных с ним.

Существует намного более безопасный преемник NIS, называемый NIS+. Обратитесь к NIS HOWTO за более детальной информацией: http://sunsite.unc.edu/mdw/HOWTO/NIS-HOWTO.html

8.10 Щит (firewall)

Под щитом подразумевается ограничение на прохождение информации как внутрь так и за пределы вашей локальной сети. Обычно компьютер, выполняющий роль щита, соединен с интернетом и вашей локальной сетью, и доступ к интернету из вашей локальной сети выполняется только через него. Таким образом щит может контролировать, что приходит из интернета в локальную сеть, и что уходит из локальной сети в интернет.

Существует большое количество типов и методов организации щита. Linux система реализует довольно хороший щит низкой стоимости. Код, реализующий щит, может быть встроен прямо в ядро начиная с версии 2.0 и выше. Инструмент ipfwadm позволяет вам определять, какой части сетевого трафика можно уходить в интернет или приходить из него. Вы можете также протоколировать определенные типы сетевого трафика.

Щит является очень полезным и важным инструментом в обеспечении безопасности вашей сети. Важно понять, что вы не должны забывать о безопасности только из-за того, что у вас **есть** щит, и не заботиться о безопасности машин за щитом. Это будет фатальной ошибкой. Советуем заглянуть в очень хороший Firewall-HOWTO для более детального ознакомления с реализацией щита в Linux.
http://sunsite.unc.edu/mdw/HOWTO/Firewall-HOWTO.html

Информацию по этому вопросу можно также найти в IP-Masquerade mini-howto: http://sunsite.unc.edu/mdw/HOWTO/mini/IP-Masquerade.html

Детальную информацию по ipfwadm (инструменту, который позволяет вам изменять установки вашего щита), можно найти на его домашней странице : http://www.xos.nl/linux/ipfwadm/

9. Подготовка системы безопасности (до соединения с интернет)

Итак, вы всесторонне проверили вашу систему и сделали ее настолько безопасной насколько это было возможно (исходя из ваших знаний :), а значит готовы к соединению с Интернет. Существует несколько вещей, которые вы теперь должны сделать, чтобы быть подготовленным на случай взлома, и, следовательно, быстро обезвредить взломщика, восстановиться и работать дальше.

9.1 Сделайте резервную копию всей вашей системы

Обсуждение методов резервирования и хранения вне целей этого документа, но буквально несколько слов о резервировании и безопасности:

Если у вас меньше чем 650Mb данных в одном разделе, то для резервирования можно порекомендовать CD-R (поскольку его очень трудно подделать, и данные долго сохраняются). На лентах и других перезаписываемых носителях сразу после создания резервных копий необходимо поставить защиту от записи и затем периодически проверять, чтобы предотвратить подделки (или подмену). Сохраняйте ваши резервные копии в надежных недоступных местах. Хорошие резервные копии обеспечат вам возможность восстановления вашей системы в любой ситуации.

9.2 Выбор режима резервирования

Простым для поддержания считается шести ступенчатый цикл. Он включает 4 текущие ленты на неделю, одна для четных пятниц, одна для нечетных пятниц. Делайте нарастающее резервирование каждый день и полное резервирование системы в пятницу на соответствующую ленту. Если вы сделали какие-либо особенные важные изменения в системе или добавили важные данные, то уместно будет сразу сделать резервную копию.

9.3 Создайте резервную копию вашей RPM базы

В случае вторжения вы можете использовать базу RPM как спасательную нить, но только в том случае, если вы уверены в ее целостности. Желательно скопировать базу RPM на дискету и держать ее где-либо в недоступном месте. В дистрибутиве Debian вероятно имеется что-то подобное.

Скорее всего файлы /var/lib/rpm/fileindex.rpm и /var/lib/rpm/packages.rpm не поместятся на отдельную дискету, а в сжатом виде каждый поместится на отдельную дискету.

Теперь если ваша система (не приведи Господи) будет взломана, вы можете использовать команду:

```
root#  rpm -Va
```

для верификации каждого файла в системе. Прочтите man страницу по RPM, поскольку есть другие опции, которые можно включить, чтобы сделать rpm менее многословным.

Это значит, что каждый раз как вы добавляете новый RPM в систему, вам нужно обновить резервную копию. Вы ощутите все достоинства перед недостатками.

9.4 Отслеживайте данные регистрации использования системы

Очень важно, чтобы не подверглась взлому информация, которая поступает от syslog. Начать надо с того, что разрешить чтение и запись в /var/log только ограниченному контингенту пользователей.

Обязательно следите за тем, что туда заносится, особенно посредством `auth'. Большое количество неудачных регистраций, например, может указывать на попытку вторжения.

Где искать ваши log файлы, будет зависеть от вашего дистрибутива. В Linux системах, которые поддерживают "Linux Filesystem Standard", таких как Red Hat, смотрите в /var/log для проверки messages, mail.log и других протоколов.

Чтобы узнать где ваш дистрибутив ведет системные журналы, вам нужно посмотреть в файл /etc/syslog.conf. Это файл, который указывает syslogd (системному протоколирующему демону) куда записывать различные сообщения.

Вы можете также захотеть настроить ваш log-rotating скрипт или демон для того, чтобы ваши журнальные записи дольше просуществовали, т.е. чтобы вы имели время их более детально изучить. Рекомендуем вам обратить внимание на пакет 'logrotate', поставляемый в дистрибутиве Red Hat. Другие дистрибутивы вероятно имеют подобные вещи.

Если вы заметили, что в журнальных файлах кто-то возился, посмотрите сначала можете ли вы определить, когда это началось и каких вещей касалось. Большой ли период времени таким образом содержит ненадежную информацию? Лучше всего в такой ситуации восстановить журналы с резервных копий (если у вас конечно такие есть?)

Журнальные файлы обычно изменяют взломщики, для того чтобы скрыть свои действия, но их присутствие все же можно заметить по странным событиям в системе. Вы можете отследить попытки взломщика войти в систему или изменить какую-либо программу для получения счета администратора. Вы можете просмотреть журнальные файлы до того, как взломщик изменит их.

Вам необходимо отделить `auth' данные от других запротоколированных событий, включая попытки изменения счетов с помощью `su', попытки регистрации и другую информацию, касающуюся счетов пользователей.

Если возможно, настройте syslog так, чтобы от отсылал копию наиболее важных данных на безопасную систему. Это предотвратит попытки взломщика скрыть свою деятельность путем удаления информации о его login/su/ftp и др. Изучите man страницу по syslog.conf, особенно в части `@' опции.

И наконец, журнальные файлы намного менее полезны, если их никто не читает. Выделите постоянное время для просмотра журнальных файлов, тогда вы наверняка обретете чувство ситуации - нормально все или нет. Это может сильно помочь не допустить непредвиденной ситуации.

9.5 Делайте модернизацию системы

Большинство пользователей Linux инсталируются с CDROM. Из-за быстрой природы появления исправлений в безопасности, постоянно появляются новые (исправленные) программы. До того, как вы откроете свою систему в сети, сходите на ftp сервер вашего дистрибутива (например, ftp.redhat.com) и возьмите все пакеты, которые были обновлены с момента получения вами CDROM. В большинстве случаев новые пакеты будут содержать важные исправления в области безопасности, так что неплохо их будет инсталировать.

10. Что делать во время и после взлома?

Итак, вы следуете советам этого (или какого-либо еще) документа и обнаружили вторжение. Перво-наперво нужно оставаться спокойным. Поспешные действия могут принести больше вреда нежели сам взломщик.

10.1 Нарушение безопасности в процессе

Обнаружение процесса нарушения безопасности может быть напряженным предприятием. Поскольку ваши ответные действия могут иметь большие последствия.

Если нарушение, которое вы обнаружили имеет физическую природу, есть шансы, что вы обнаружите того, кто вломился в ваш дом, офис или лабораторию. Вы должны предупредить своих представителей власти. В лаборатории вы можете обнаружить кого-либо, пытающегося открыть дипломат или перезапустить машину. В зависимости от ваших полномочий и инструкций, вы можете сами приказать ему остановиться или сообщить службе безопасности.

Если вы обнаружили локального пользователя, пытающегося нарушить систему безопасности, перво-наперво сообщите ему все, что вы о нем думаете. Проверьте систему, с которой он зарегистрировался. Та ли это система, с которой он обычно регистрируется? нет? Тогда используйте уже не электронные средства общения. Например, позвоните ему по телефону или посетите его офис/дом и поговорите с ним. Если он признает, что это был он, потребуйте с него объяснений, что он делал в вашей системе, и убедите его не делать больше этого. Если это был не он и не может понять о чем идет речь, то скорее всего этот инцидент требует дальнейшего расследования. Тщательно исследуйте инцидент и прежде чем выдвигать обвинения, соберите побольше доказательств.

Если вы обнаружили вторжение в сеть, перво-наперво (если вы можете) отсоедините вашу сеть. Если вторжение произошло через модем, отсоедините модемный кабель, тогда взломщик вероятнее всего подумает о проблемах связи, а не об обнаружении.

Если вы не можете отсоединить сеть (идет интенсивная работа, или вы не имеете физического контроля над системой), то наилучшим будет использовать что-то наподобие tcp_wrapper или ipfwadm для запрещения доступа из системы взломщика.

Если вы не можете запретить доступ всем из сети взломщика, то нужно заблокировать счета пользователей. Помните, что блокирование счетов дело не легкое. Вы должны помнить о файлах .rhosts, ftp доступе и черных входах.

После того, как вы сделаете что-либо из вышеперечисленного (отсоедините сеть, запретите доступ из сети взломщика и/или заблокируете их счета), вы должны убить все его пользовательские процессы.

Некоторое время после этого вы должны очень внимательно отслеживать состояние вашей системы, поскольку взломщик может попробовать повторить вторжение. Вероятнее всего используя другой счет и/или с другого сетевого адреса.

10.2 Нарушение безопасности уже произошло

Итак, вы либо обнаружили нарушение безопасности, которое уже произошло, либо обнаружили его и заблокировали деятельность взломщика в вашей системе (короче, выдворили его). Что же теперь?

Закрыть дыру

Если вы можете определить, что использовал взломщик для внедрения в вашу систему, вы должны попытаться закрыть эту дыру. Например, возможно вы увидите несколько ftp входов, прежде чем пользователь зарегистрировался. Выключите FTP сервис и проверьте существует ли его обновленная версия или какой-либо список заплаток известных ошибок.

Проверьте все ваши журнальные файлы, а затем посетите ваши списки безопасности и web-узлы на предмет наличия каких-либо новых обнаруженных ошибок, которые вы можете исправить. Исправления для системы безопасности Caldera вы можете найти по адресу http://www.caldera.com/tech-ref/security/. Red Hat еще не имеет отдельной страницы по ошибкам безопасности от общих ошибок, но их дистрибутив errata доступен по адресу http://www.redhat.com/errata Очень вероятно, что если один поставщик выпустит обновление в области безопасности, то большинство остальных поставщиков Linux сделают тоже самое.

Если вы не локализовали и не заблокировали взломщика, вероятнее всего он вернется. Не обязательно на ваш компьютер, может на какой-то другой, но в вашей сети. Если взломщик использовал пакетные ищейки, большие шансы, что он имел доступ и к другим локальным машинам.

Оценка повреждений

Перво-наперво нужно оценить повреждения. Что было нарушено? Если вы используете тестер целостности такой как Tripwire - запустите его и он вам все расскажет. Если нет, просмотрите все - особенно важные данные.

Так как системы Linux становиться все легче и легче инсталировать, вы можете скопировать куда-то ваши конфигурационные файлы, а затем полностью очистить диск и переинсталировать систему, восстановить файлы пользователей с резервных копий и скопировать назад конфигурационные файлы. Это будет гарантировать полностью чистую систему. Если вам нужно сделать резервную копию взломанной системы, будьте особенно внимательны к выполняемым файлам, которые вы резервируете, поскольку они могут быть "троянами", оставленными взломщиком.

Резервируйте, резервируйте и еще резервируйте!

Лучшим решением в плане безопасности является регулярное резервирование. Если ваша система подверглась вторжению, вы всегда сможете восстановиться с резервных копий. Конечно, некоторые данные могут быть ценны и для взломщика, и он могут не столько удалять их, но и воровать, делая себе копии, но в конце концов вы по крайнем мере сохраните эти данные.

Прежде чем восстановить поврежденный файл вы должны проверить несколько прошлых резервных копий, а не только последнюю. Может быть, что взломщик орудовал некоторое время назад, и вы успешно сделали несколько резервных копий уже поврежденных файлов!!!

Конечно, существует также понятие безопасности резервных копий. Убедитесь, что вы храните их в надежном месте. Знаете, кто имеет туда доступ. (Если взломщик получит ваши резервные копии, он будет иметь все ваши данные, а вы даже знать об этом не будете.)

Выслеживание взломщика

Хорошо, вы заблокировали взломщика, восстановили вашу систему, но это еще не все. Поскольку маловероятно, что все взломщики будут пойманы, вы должны сообщить об атаке.

Вы должны сообщить об атаке администратору системы, с которой была атакована ваша система. Вы можете найти этого администратора с помощью "whois" или базы internic. Вы можете послать ему по электронной почте содержимое системных журналов с датой и временем событий. Если вы заметили еще что-либо отличающее вашего взломщика, вы можете также сообщить об этом. После посылки сообщения по e-mail, вы должны (если вы к этому склонны) связаться по телефону. Если обнаружиться, что система того администратора была только проходным звеном, он может отследить и связаться с администратором системы, с которой взломщик проник к нему, и т.д.

Опытные взломщики часто используют большое количество посреднических систем. Некоторые (или многие) из которых, могут даже и не знать, что они были взломаны. Отследить обратно путь взломщика аж до его домашней системы может быть очень трудно. Будьте очень вежливы с администраторами других систем при выслеживании и они вам много в чем помогут.

Вы можете сообщить также в организацию безопасности, членом которой вы являетесь (CERT или подобную).

11. Источники информации по безопасности

Существует очень много узлов в интернете, посвященных UNIX безопасности в общем и специфике Linux безопасности. Очень важно подписаться на один (или более) список рассылки по вопросам безопасности и отслеживать текущие исправления. Большинство из таких списков небольшие по объему, но очень информативны.

11.1 FTP узлы

CERT это Computer Emergency Response Team. Они часто рассылают предупреждения о последних атаках и исправлениях. cert.org

Replay имеет архивы очень многих программ безопасности. Поскольку они находятся за пределами США, им не нужно подчиняться ограничениям по распространению криптографических программ. replay.com

Matt Blaze является автором CFS и отличным советником в вопросах безопасности. Matt Blaze's stuff

tue.nl отличный ftp узел по безопасности в Голландии. ftp.win.tue.nl

11.2 Web узлы

Hacker FAQ это FAQ про хакеров: The Hacker FAQ

COAST архив содержит большое количество информации и программного обеспечения по безопасности: COAST

Rootshell.com отличный узел для изучения современных методов взлома, которые сейчас используют взломщики: rootshell.com exploits

BUGTRAQ приводит советы в области безопасности: BUGTRAQ archives

CERT, the Computer Emergency Response Team, приводит советы по общим атакам на UNIX платформах: CERT home

Dan Farmer - автор SATAN и многих других инструментов, касающихся безопасности, его домашний web узел содержит несколько интересных обзоров по безопасности, а также программный инструментарий : Dan Farmers trouble.org

Linux security WWW хороший узел по безопасности в Linux: Linux Security WWW

Reptile имеет на своем узле очень много хорошей информации по безопасности в Linux: Reptiles Linux Security Page

Infilsec рассказывает об уязвимых местах различных платформ: Infilsec vunerability engine

CIAC периодически рассылает бюлетни по обнаруженным дырам: CIAC bulitins

Хорошей отправной точкой по подключаемым модулям идентификации Linux (PAM) является узел http://www.kernel.org/pub/linux/libs/pam/.

11.3 Списки рассылки

Bugtraq: Чтобы подписаться на bugtraq, пошлите e-mail на listserv@netspace.org содержащий в теле subscribe bugtraq. (см. ссылку выше по архивам).

CIAC: Пошлите e-mail к majordomo@tholia.llnl.gov, а в теле письма (не в subject) напишите: subscribe ciac-bulletin

11.4 Книги - Печатные материалы

Существует очень много хороших книг по безопасности. В этом разделе приведены только некоторые из них. В дополнение к специализированным книгам по безопасности, вопросы безопасности раскрыты во многих других книгах по системному администрированию.

Building Internet Firewalls By D. Brent Chapman & Elizabeth D. Zwicky

1st Edition September 1995

ISBN: 1-56592-124-0

Practical UNIX & Internet Security, 2nd Edition By Simson Garfinkel & Gene Spafford

2nd Edition April 1996

ISBN: 1-56592-148-8

Computer Security Basics By Deborah Russell & G.T. Gangemi, Sr.

1st Edition July 1991

ISBN: 0-937175-71-4

Linux Network Administrator's Guide By Olaf Kirch

1st Edition January 1995

ISBN: 1-56592-087-2

PGP: Pretty Good Privacy By Simson Garfinkel

1st Edition December 1994

ISBN: 1-56592-098-8

Computer Crime A Crimefighter's Handbook By David Icove, Karl Seger & William VonStorch (Consulting Editor Eugene H. Spafford)

1st Edition August 1995

ISBN: 1-56592-086-4

12. Глоссарий

- **Host (компьютер):** Присоединенный к сети компьютер

- **Firewall (щит):** Некий компонент или набор компонентов, который ограничивает доступ между локальной сетью и Интернетом, или между разными частями локальной сети.

- **Bastion Host (бастион):** Компьютерная система, которая должна быть очень сильно защищена из-за своем подверженности атакам, обычно поскольку постоянно открыта в Интернете или является связующим звеном для пользователей внутренних сетей. Название произошло от сильно укрепленных башен на внешних стенах средневековых замков. Бастионы поддерживают критические места в защите, обычно имея мощные стены, комнаты для резервов, и в большинстве случаев емкости с кипящей нефтью или смолой для повреждения нападающих. (Как вам АНАЛОГИИ :)

- **Dual-homed Host (узел):** Компьютерная система общего назначения, которая имеет как минимум два 2 интерфейса.

- **Packet (пакет):** Фундаментальная единица связи в сети.

- **Packet Filtering (фильтрация пакетов):** Действия, которые выполняет определенное устройство для выборочного контролирования потока данных в и из сети. Пакетные фильтры разрешают прохождение пакетов или блокируют их, обычно при пересылке из одной сети в другую (наиболее часто из Интернета в локальную сеть, и наоборот). Выполняя пакетную фильтрацию вы устанавливаете набор правил, которые определяют, какие типы пакетов (идущие на или из определенного IP адреса или порта) будут пропускаться, а какие блокироваться.

- **Perimeter network (пограничная сеть):** Сеть, которая добавляется между защищенной и внешней сетью для создания дополнительного уровня безопасности. Пограничная сеть иногда называется DMZ.

- **Proxy server (сервер-посредник):** Программа, которая работает с внешними серверами на пользу внутренних клиентов. Клиенты делают запрос к посреднику, который переправляет санкционированные запросы клиентов к реальным серверам, а ответы - обратно клиентам.

- **Denial of Service (отказ в сервисе):** Атака "отказ в сервисе" классифицируется тогда, когда взломщик тратит ресурсы вашего компьютера на задачи, которые не входят в круг его обязанностей, таким образом делая невозможным нормальное использование ваших сетевых ресурсов в законных целях.

- **Buffer Overflow (переполнение буфера):** Очень частая ситуация в программировании, когда буфер заводится недостаточно большой и не выполняется проверка не его переполнение. Когда такой буфер переполняется, работающая программа (демон или set-uid программа) может делать совсем левые вещи. Обычно происходит перезапись адреса возврата функции в стеке, указывая таким образом в другое место.

- **IP Spoofing (подмена IP):** IP-Spoofing - это комплексная техническая атака, которая состоит из нескольких компонент. Это прорыв системы безопасности, когда один компьютер посредством обмана устанавливает доверительные отношения с другим выдавая себя за того, кем в действительности не является. По этому вопросу есть расширенная статья написанная daemon9, route и infinity (это псевдонимы) в журнале Phrack Magazine (#7, стр. 48).

- **Authentication (идентификация):** Способ гарантирования, что полученные данные соответствуют отправленным, и что заявленный отправитель соответствует реальному.

- **Non-repudiation (безотказность):** Способ, которым получатель может доказать, что отправитель некоторых данных действительно их посылал, даже если отправителю позже вздумается отрицать свою причастность к отправлению этих данных.

13. Часто задаваемые вопросы ЧАВО (FAQ)

1. Не будет ли более безопасно вкомпилировать поддержку драйверов непосредственно в ядро нежели представлять их модулями?

   Ответ: Некоторые полагают, что лучше не использовать возможность загрузки драйверов устройств в виде модулей, поскольку взломщик (или они сами) может загрузить троянский модуль, который повредит систему безопасности.

   Однако, для того чтобы загрузить модуль, вы должны быть администратором. Объектные файлы модулей разрешают запись тоже только администратору. Это значит, что взломщик должен иметь права администратора, чтобы загрузить модуль. Если взломщик заполучит права администратора, то есть намного более серьезные вещи, о которых нужно будет беспокоится, нежели загрузка какого-то модуля.

   Модули созданы для динамической загрузки драйверов поддержки определенных устройств, которые обычно редко используются. Для серверов или систем, выполняющих роль щита (firewall), это маловероятно. По этой причине будет разумно для таких машин вкомпилировать поддержку устройств прямо в ядро. Модули к тому же медленнее нежели код в ядре.

2. Почему всегда запрещена регистрация администратором с удаленной машины?

   Ответ: Смотрите раздел "безопасность администратора". Это сделано намеренно, чтобы предотвратить попытки пользователей зарегистрироваться на вашей машине через telnet как администратор, что очень уязвимо. Не забывайте потенциальный взломщик имеет время запустить автоматическую программу для получения вашего пароля.

3. Как мне включить теневые пароли в моем Red Hat 4.2 или 5.0 Linux?

   Ответ: Теневые пароли это механизм сохранения ваших паролей в отличном от обычного / etc/passwd файле. Это имеет несколько преимуществ. Первое это то, что теневой файл /etc/shadow доступен для чтения только администратору, в отличие от /etc/passwd, который должен быть доступен для чтения всем. Другое преимущество в том, что вы как администратор можете включать или блокировать счета и никто не будет знать статус счетов других пользователей.

   Тогда файл /etc/passwd используется только для хранения имен пользователей и групп, а также используется программами наподобие `/bin/ls' для связывания пользовательских ID с определенным именем пользователя в списке каталога.

   Тогда файл /etc/shadow содержит только имя пользователя и его/ее пароль, и возможно информацию о счете, типа времени окончания действия и т.п.

   Чтобы включить теневые пароли, запустите 'pwconv' будучи администратором - теперь /etc/shadow существует и будет использоваться приложениями. Если вы используете Red Hat 4.2 или выше, то PAM модули автоматически перестроятся на использование теневых паролей без какого-либо вмешательства с вашей стороны.

   Если вы заинтересовались безопасностью ваших паролей, вероятно вас заинтересует методы генерирования хороших паролей. Для этого вы можете использовать модуль `pam_cracklib', который является частью PAM. Он сравнивает ваши пароли с Crack библиотеками, чтобы вы знали легко ли угадываются ваши пароли программами, которые занимаются подбором паролей.

4. Как мне включить SSL расширение для Apache?

   Ответ:

   1.Скачайте SSLeay 0.8.0 или выше из ftp://ftp.psy.uq.oz.au/pub/Crypto/SSL

   2.Соберите, протестируйте и установите его!

   3.Скачайте исходники Apache 1.2.5 или выше

   4.Скачайте SSLeay расширение для Apache отсюда

5.Разархивируйте его в каталог исходников Apache и исправьте Apache как написано в README.

6.Настройте и соберите его.

Вы можете также сходить на Replay Associates, где есть много уже скомпилированных пакетов и за пределами США.

5. Как мне манипулировать счетами пользователей не нарушая при этом безопасности?

Ответ: Дистрибутив Red Hat, особенно RH5.0, содержит огромное количество инструментов, с помощью которых можно работать со счетами пользователей.

- pwconv и unpwconv можно использовать для конвертирования паролей из обыкновенных в теневые и обратно, соответственно
- pwck и grpck можно использовать для верификации правильности организации файлов passwd и group.
- программы useradd, usermod и userdel можно использовать для добавления, удаления и модификации счетов пользователей. Программы groupadd, groupmod и groupdel делают тоже самое для групп.
- групповые пароли можно создавать с помощью gpasswd.

Все эти программы совместимы с теневыми паролями - т.е. если вы установили теневые пароли, они будут использовать файл /etc/shadow.

Для более детального ознакомления смотрите соответствующие страницы man.

6. Как мне поставить пароли на определенные HTML документы используя Apache?

Я так думаю вы и не догадываетесь о http://www.apacheweek.org, не правда ли?

Информацию по идентификации пользователей вы можете найти на http://www.apacheweek.com/features/userauth, а также на web узле по вопросам безопасности http://www.apache.org/docs/misc/security_tips.html

14. Заключение

Подписавшись на списки рассылки по безопасности, и будучи таким образом в курсе текущих событий, вы очень много сделаете в направлении безопасности вашей системы. Если вы следите за журнальными файлами вашей системы и регулярно используете что-либо вроде tripwire, вы можете сделать еще больше.

Разумный уровень безопасности не трудно поддерживать и на домашнем компьютере. Больше усилий требуется в коммерческих структурах, но Linux наверняка может быть безопасной платформой. Из-за природы создания Linux, исправления в системе безопасности выходят намного быстрее нежели в коммерческих операционных системах, делая Linux идеальной платформой, когда требуется безопасность. (Автор увлеченный человек, поскольку ничто в этом мире не идеально - не помню откуда. Прим. перевод.)

15. Благодарности

Здесь собрана информация со многих источников. Спасибо всем перечисленным ниже за прямой и косвенный вклад:

Rob Riggs <rob@DevilsThumb.com>
S. Coffin <scoffin@netcom.com>
Viktor Przebinda <viktor@CRYSTAL.MATH.ou.edu>
Roelof Osinga <roelof@eboa.com>
Kyle Hasselbacher <kyle@carefree.quux.soltec.net>
"David S. Jackson" <dsj@dsj.net>
"Todd G. Ruskell" <ruskell@boulder.nist.gov>
Rogier Wolff <R.E.Wolff@BitWizard.nl>