Kali Linux Revealed

на русском

CODEBY.NET

Kali Linux Revealed

Mastering the Penetration Testing Distribution

Издатель	Offensive Security
Оригинальное название книги	Kali Linux Revealed
Перевод	Анна Давыдова
Оформление	lios
Дизайн обложки	SploitFace
Перевод выполнен по заказу	Codeby.net

Содержание

Часть 1: О Kali Linux

Содержание:

Ключевые слова главы:

- Дистрибутив Linux;
- Производный от Debian;
- Задачи, функции, принцип;

Kali Linux[1] является готовым дистрибутивом аудита безопасности Linux, который основан на Debian GNU/Linux. Целевой аудиторией Kali являются профессионалы в сфере безопасности и IT администраторы, что позволяет им проводить тестирование на проникновение, криминалистический анализ и контроль безопасности.

Что такое дистрибутив Linux?

Хотя это обычно используется как общее название для всей операционной системы**,** Linux – это просто название ядра, части программного обеспечения, которое регулирует взаимодействие между жестким диском приложениями конечного пользователя. Выражение дистрибутив Linux, с другой стороны, относится ко всей операционной системе, построенной на основе Linux, ядра, обычно включающей программу установки и много приложений, которые установлены раньше или входят в установочный пакет.

Debian GN U/Linux[1] [2] – это ведущий общий дистрибутив Linux, известный своим качеством и стабильностью. Kali Linux базируется на работе проекта Debian и добавляет свыше 300 своих специальных пакетов, относящихся к информационной безопасности, особенно, в области тестирования на проникновение. Debian – это бесплатный проект программного обеспечения, предоставляющий множественные версии своих операционных систем, и мы часто используем термин *дистрибутив для того,* чтобы отнести вас к какой-то его конкретной версии, например, дистрибутивы Debian Stable или Debian Testing. То же самое применимо к Kali Linux — с дистрибутивом Kali Rolling, например.

1.1 Немного истории

Проект Kali Linux потихоньку начинался в 2012, когда Offensive Security решила, что они хотят заменить свой старый проект BackTrack Linux, который поддерживался вручную, на что-то, что могло бы стать производной Debian [3] со всей необходимой

[1] https://www.kali.org
[2] https://www.debian.org
[3] https://wiki.debian.org/Derivatives/Census

инфраструктурой и улучшенной техникой пакетирования. Было принято решение создать Kali на основе дистрибутива Debian, потому что она известна своим качеством, стабильностью и большим выбором доступного программного обеспечения. Вот почему я (Raphael) привлекался к этому проекту в качестве консультанта Debian.

Первый выпуск (версия 1.0) произошел год спустя, в марте 2013, и был основан на Debian 7 “Wheezy”, стабильном дистрибутиве Debian. В первый год развития мы создали пакеты сотен приложений, относящихся к тестированию на проникновение, и построили полноценную инфраструктуру. Даже, если ряд приложений был действительно важен, список приложения тщательно курировался, пропуская приложения, которые больше не работали или повторяли свойства уже использовавшиеся в лучших программах.

В последующие два года после версия 1.0, Kali выпустила много пошаговых усовершенствований, расширяющих ряд доступных приложений и улучшающих поддержку различного оборудования благодаря более новым впускам ядра. При помощи инвестиций в постоянную интеграцию мы обеспечили то, что все важные пакеты хранятся в устанавливаемом состоянии, и что живой пользовательский образ (отличительная черта данного дистрибутива) всегда может быть создан.

В 2015 вышла Debian 8 “Jessie”, мы работали над пересмотром Kali Linux на его основе. Так как в Kali Linux 1.x не входит GNOME Shell (полагаясь на GNOME Fallback вместо нее), в этой версии мы решили использовать и улучшить ее: мы добавили некоторые расширения GNOME Shell для получения недостающих свойств, самым примечательным в этом является Applications menu. Результатом этой работы стала Kali Linux 2.0, опубликованная в августе 2015.

GNOME является средой рабочего стала Kali Linux по умолчанию.

Среда рабочего стола – это собрание графических приложений, которые разделяют общий графический инструментарий**,** и которые должны использоваться вместе на рабочей платформе

пользователя. Среда рабочего стола, как правило, не используется на серверах. Они обычно предоставляют лаунчер приложения, файловый менеджер, веб-браузер, email клиент, офисное помещение и т.д.
GNOME[4] – самая популярная среда рабочего стола (вместе с KDE[5], Xfce[6], LXDE[7], MATE[8]) и они являются уже установленными на основные образы ISO, предоставленные Kali Linux. Если вам не нравится GNOME, то очень легко создать пользовательский ISO образ со средой рабочего стола по вашему выбору. Инструкции, как это делать, освещены дальше в этой книге, в главе 9 "Продвинутое использование" [стр. 222].

Параллельно мы увеличили наши усилия, чтобы гарантировать, что у Kali Linux всегда присутствует последняя версия всех приложений, предназначенных для проведения тестирования на проникновение. К сожалению, цель была немного не связана с использованием Debian Stable в качестве основы для дистрибутива, потому что она требует от нас обеспечивать ретро поддержку многим пакетам. Это происходит вследствие того факта, что Debian Stable отдает приоритет стабильности программного обеспечения, часто вызывая долгую задержку между выходом более свежего обновления и интеграцией этого обновления в дистрибутив.

Учитывая наши инвестиции в непрерывную интеграцию, было вполне естественным шагом перебалансировать Kali Linux поверх Debian Testing таким образом, чтобы мы могли извлечь пользу из последней версии всех пакетов Debian, сразу же, как только они становились доступными. У Debian Testing более агрессивный цикл обновления, который сам по себе является более совместимым с философией Kali Linux.

В сущности, это концепция Kali Rolling. В то время как дистрибутив rolling был доступен совсем небольшое время, Kali 2016.1 была впервые официально выпущена для того, чтобы полностью охватить природу rolling дистрибутива: когда вы устанавливаете последнюю версию Kali ваша система, в действительности,

[4]https://www.gnome.org
[5]https://www.kde.org
[6]http://www.xfce.org
[7]http://lxde.org
[8]http://mate-desktop.org

отслеживает дистрибутив Kali Rolling, *и каждый день у вас будет новое обновление*. В прошлом, выпуски Kali были снэпшотами, лежащими в основе дистрибутива Debian со встроенными в него особенными Kali пакетами.

Rolling дистрибутив имеет много преимуществ, но у него также и много проблем, которые касаются как и тех, кто создает дистрибутивы, так пользователей, которым приходится справляться с бесконечным потоком обновлений, а иногда и полностью с несовместимыми изменениями. Эта книга создана, чтобы дать вам знания, необходимые для преодоления всяких неожиданностей, которые могут возникнуть во время установки и использования Kali Linux.

1.2 Взаимосвязь с Debian

Дистрибутив Kali Linux основан на Debian Testing[9]. Именно поэтому большинство пакетов, доступных в Kali Linux, пришли прямо из репозитория Debian.

Несмотря на то, что в основном Kali Linux полагается на Debian, он является полностью независимым в смысле того, что у нас есть своя собственная инфраструктура, и в связи с этим мы сохраняем полную свободу производить любые изменения, какие мы захотим.

1.2.1 Движение пакетов

Со стороны Debian, специалисты ежедневно работают над обновлением пакетов и загрузкой их в дистрибутив Debian Unstable. Как только самые проблемные ошибки будут устранены, пакеты переносятся в дистрибутив Debian Testing. Процесс переноса также гарантирует, что ни одна из зависимостей не будет нарушена в Debian Testing. Основная задача подобного рода действий заключается в том, что Testing всегда готов к использованию (или даже к новому выпуску!)

[9]https://www.debian.org/releases/testing/

Цели, преследуемые Debian Testing, полностью совпадают с задачами, поставленными перед Kali Linux, и именно поэтому мы взяли его за основу. Чтобы добавить в дистрибутив специальные Kali пакеты, мы следуем процессу, состоящему из двух этапов.

Сначала, мы берем Debian Testing и принудительно внедряем наши собственные пакеты Kali (расположенные в нашем репозитории *kali-dev-only*) для создания репозитория *kali-dev*. Этот репозиторий время от времени будет давать сбои: например, наши специальные Kali пакеты, могут не устанавливаться, пока они не будут перекомпилированы в отношении более новых библиотек. В других ситуациях раздвоенные пакеты, также могут быть обновлены, чтобы снова стать инсталлируемыми или для того, чтобы исправить инсталлируемость другого пакета, который зависит от более новой версии раздвоенного пакета. В любом случае, *kali-dev* не предназначен для конечных пользователей.

Kali-rolling - является дистрибутивом, информацию о котором пользователи Kali Linux скорее всего будут отслеживать. Также он создан из *kali-dev* таким же образом, как и Debian Testing создан из Debian Unstable. Пакеты переносятся только тогда, когда все зависимости могут быть удовлетворены в целевом дистрибутиве.

1.2.2 Управление различиями с Debian

В качестве конструктивного решения мы стараемся как можно больше минимизировать количество раздвоенных пакетов. Как бы то ни было, чтобы реализовать некоторые из уникальных особенностей Kali, необходимо внести ряд некоторых изменений. Чтобы ограничить влияние этих изменений, мы стремимся отправить их «выше по течению», интегрируя особенность напрямую или путем добавления необходимых методов таким образом, чтобы было легко непосредственно включить нужные функции без дальнейшей модификации самих вышестоящих пакетов.

The Kali Package Tracker[10] помогает нам продолжать отслеживать наши изменения с Debian. В любое время, мы можем проверить,

[10]http://pkg.kali.org/derivative/kali- dev/

какой пакет был раздвоен, синхронизирован ли он с Debian и требуется ли обновление. Все наши пакеты сохраняются в репозиториях[11] Git, где рядом находятся ветка Debian и ветка Kali. Благодаря этому, обновление раздвоенных пакетов является простым процессом, состоящим из двух несложных шагов: обновление ветки Debian и затем слияние с веткой Kali.

В то время, как количество раздвоенных пакетов в Kali является относительно небольшим, число дополнительных пакетов довольно таки велико: на апрель 2017 года их количество приближалось к 400. Большинство этих пакетов являются бесплатным программным обеспечением, соответствующим Debian Free Software Guidelines[11] [12], и наша конечная цель заключается в том, чтобы всегда поддерживать эти пакеты в Debian. К сожалению, есть также несколько исключений, когда практически невозможно было создать правильное пакетирование. В результате нехватки времени в Debian было отправлено несколько пакетов.

1.3 Задачи и варианты использования

Если, к примеру, основные задачи Kali могут быть вкратце изложены в нескольких словах: «тестирование на проникновение и аудит безопасности», то, несмотря на это, по-прежнему существует большое количество различных задач, которые являются неотъемлемой частью этих процессов. Kali Linux создан как *фреймворк*, потому что он включает в себя множество различных инструментов, предназначенных для разных задач (хотя это не отменяет того факта, что они могут использоваться в комбинации во время тестирования на проникновение).

Например, Kali Linux может быть использован на различного рода компьютерах: очевидно, что в первую очередь речь идет о ноутбуках пентестеров, но также Kali Linux может быть использован на серверах системных администраторов, желающих отслеживать состояние своих сетей, на рабочих платформах криминалистических аналитиков и, что самое неожиданное, на

[11] http://git.kali.org
[12] https://www.debian.org/social_contract

скрытых встроенных устройствах, как правило, с процессорами ARM, которые могут быть присоединены к диапазону беспроводной сети или подключены к компьютеру целевых пользователей. Многие ARM устройства являются прекрасными машинами для проведения атаки благодаря своему небольшому размеру и маленькому количеству потребляемой энергии. Kali Linux также может работать в облаке для создания конгломерата машин, занимающихся взломом паролей, и на мобильных телефонах и планшетах для проведения портативного тестирования на проникновение.

Но и это еще не все; пентестеры также нуждаются в серверах для использования программных средств внутри команде пентестеров, для настройки веб сервера и его последующего использования в фишинговых кампаниях, для запуска устройств по сканированию уязвимостей и других подобных инструментов.

После того, как вы загрузите Kali, вы быстро обнаружите, что главное меню Kali Linux организовано по темам, соответствующим различным задачам и действиям, которые хорошо подойдут для пентестеров и других специалистов по информационной безопасности, как показано на рисунке 1.1 «Меню приложений Kali Linux "[Стр. 6].

Figure 1.1 *Меню приложений Kali Linux*

Эти задачи и направления деятельности включают в себя:

- **Сбор информации**: сбор данных о целевой сети и ее структуре, идентификация компьютеров, их операционных

систем и служб, которые они запускают. Определение потенциально уязвимых частей информационной системы;

- **Анализ уязвимостей**: быстрая проверка покажет, подвержена ли локальная или удаленная машина влиянию какой-либо известной уязвимости или небезопасной конфигурации. Сканер уязвимостей использует базы данных, которые содержат тысячи различных подписей для определения потенциальных уязвимостей;
- **Анализ веб приложений**: определение неправильных конфигураций и слабых мест в безопасности веб приложений. Очень важно определить и свести к минимуму влияние подобных угроз, так как публичная доступность этих приложений делает их идеальной мишенью для злоумышленников;
- **Оценка базы данных**: от SQL инъекции до атаки с целью кражи учетных данных, атаки на базы данных являются очень распространенным вектором атаки. К этому роду деятельности относятся инструменты, для проверки вектора атаки, начиная от SQL инъекций и заканчивая извлечением и анализом данных;
- **Взлом пароля**: системы аутентификация всегда хорошо знакомы с векторами атак. Здесь находится очень много полезных инструментов, начиная с инструмента взлома пароля онлайн вплоть до автономных атак против зашифрованных или хэшированных систем;
- **Беспроводные атаки**: Довольно большая распространенность беспроводных сетей означает, что они всегда будут представлять большой интерес для злоумышленников и выступать одним из векторов атак. Благодаря широкому спектру поддержки нескольких беспроводных карт Kali является очевидным выбором для проведения атак на несколько типов беспроводных сетей;
- **Обратная разработка**: обратная разработка является направлением деятельности, который ставит перед собой множество задач. В рамках поддержки атакующих действий обратная разработка выступает основным методом для идентификации различных уязвимостей и разработки эксплойта. В целях защиты обратная разработка используется для анализа вредоносного программного обеспечения, используемого во время атаки. В подобных ситуациях основная цель заключается в том, чтобы определить возможности

осуществляемой против вас деятельности;

- **Инструменты эксплуатации**: эксплуатация, или использование (ранее идентифицируемой) уязвимости, позволяет вам получить контроль над удаленной машиной (или устройством). Данный доступ можно использовать в дальнейшем для проведения атак с целью расширения прав локально на взломанной машине или на других машинах, доступных в данной локальной сети. Данная категория содержит множество инструментов и утилит, которые намного упрощают процесс написания ваших собственных эксплойтов;
- **Sniffing и Spoofing:** получение доступа к данным во время их перемещения внутри сети всегда является очень выгодным для злоумышленника. Здесь вы можете найти spoofing инструменты, которые позволят вам выступать в роли законного пользователя, а также sniffing инструменты, которые предоставят вам возможность захватывать и анализировать данные прямо в момент их передачи. Использование двух этих инструментов вместе делает их намного более мощными, чем применение их по отдельности;
- **Пост эксплуатация**: как только вы получили доступ к системе, вы, скорее всего, захотите сохранить данный уровень доступа или расширить его, продвигаясь далее в сети. Инструменты, которые помогут вам в этом, находятся здесь;
- **Forensics**: живая криминалистическая среда загрузки Linux стала очень популярной в последние годы. Kali содержит большое количество популярных криминалистических инструментов на основе Linux, позволяющих вам делать абсолютно все, начиная от первоначальной сортировки и создания образа данных до полного анализа и управления делами;
- **Инструменты для отчета**: тестирование на проникновение считается полностью законченным, когда все его результаты отображены в отчете. Данная категория содержит инструменты для помощи в объединении данных, собранных специальными инструментами, обнаружении неочевидных взаимосвязей и приведении всей информации в различные отчеты;
- **Инструменты для социальной инженерии**: когда техническая сторона хорошо защищена, часто существует возможность использовать поведение человека в интересах атакующей стороны. Используя правильное влияние, людей часто могут побуждать к действиям, которые ставят под угрозу

безопасность среды. Содержал ли USB накопитель, который секретарь только что вставил в компьютер, вредоносный PDF файл? Или возможно это был троянский конь, успешно установивший бэкдор? Был ли банковский сайт, на котором только что авторизировался бухгалтер, тем самым сайтом или же это была его идеальная копия, которая используется в целях фишинга? Эта категория содержит инструменты, которые помогают вам в данных типах атаки;
- **Системные службы**: данная категория содержит инструменты, которые позволяют вам запускать и останавливать приложения, которые работают в фоновом режиме в качестве системных служб.

1.4 Основные особенности Kali Linux

Kali Linux является дистрибутивом Linux, который содержит свою собственную коллекцию сотен инструментов и программных средств, специально приспособленных для своих целевых пользователей – пентестеров и других специалистов сферы безопасности. Также он поставляется с инсталлятором для полной установки Kali Linux в качестве основной операционной системы на любой компьютер.

Это почти то же самое, что и на множестве других существующих дистрибутивах, но есть специальные отличительные особенности, которые выделяют Kali Linux среди других дистрибутивов. Большинство из этих особенностей предназначены для обслуживания конкретных нужд пентестеров. Давайте познакомимся поближе с некоторыми из них.

1.4.1 Живая система

В отличие от других дистрибутивов Linux, главный образ ISO, который вы скачиваете, предназначен не только для установки операционной системы; он также может быть использован как самозагружаемая живая система. Другими словами, вы можете использовать Kali Linux, не устанавливая его, просто путем

загрузки образа ISO (обычно после копирования образа на USB носитель).

Живая система содержит инструменты чаще всего используемые пентестерами, так что даже, если Kali Linux не является системой, которую вы используете изо дня в день, вы просто можете вставить компакт диск или USB накопитель и перезагрузиться для запуска Kali. Однако, помните, что настройки по умолчанию НЕ СОХРАНЯТ изменения после перезагрузки. Если вы выставили в настройках сохранение с помощью USB (смотри раздел 9.4, “Добавление сохранения с помощью USB на живой ISO”), вы можете настраивать систему по своему вкусу (изменять файлы конфигурации, сохранять отчеты, обновлять программное обеспечение и, например, устанавливать дополнительные пакеты), и все изменения будут сохранены, после перезагрузки.

1.4.2 Режим криминалистической экспертизы

В общем, проводя любую криминалистическую работу на системе, вы хотите избежать каких-либо действий, которые смогли бы изменить данные на анализируемой системе. К сожалению, современные среды рабочего стола, как правило, мешают этой цели, пытаясь автоматически монтировать любые обнаруженные им диск (и). Для того чтобы избежать подобного поведения у Kali Linux есть режим криминалистической экспертизы, который можно включить из меню загрузки. Данный режим заблокирует все подобные свойства.

Живая система, в частности, является очень полезной в криминалистических целях, потому что она позволяет перезагрузить любой компьютер в системе Kali Linux без изменения или получения доступа к жесткому диску.

1.4.3 Пользовательское ядро Linux

Kali Linux всегда предоставляет последнее настроенное ядро Linux, основанное на версии Debian Unstable. Это обеспечивает надежную аппаратную поддержку, особенно для широкого

спектра беспроводных устройств. Ядро модернизировано для поддержки беспроводной инъекции, поскольку многие средства оценки безопасности беспроводной сети полагаются на эту функцию.

Поскольку на многих аппаратных устройствах требуются обновленные файлы прошивки (может быть найдено в Aib / firmware /), Kali устанавливает их все по умолчанию, включая прошивку, доступную в закрытой секции Debian. Они не устанавливаются по умолчанию в Debian, потому что они являются закрытыми и, следовательно, не являются частью Debian.

1.4.4 Полностью настраиваемая

Kali Linux создана пентестерами для пентестеров, но мы прекрасно понимаем, что далеко не все согласятся с нашими проектными решениями или выбором инструментов для включения по умолчанию. Учитывая все вышесказанное, мы всегда уверяем, что Kali Linux является очень легко настраиваемой под ваши личные нужды и потребности операционной системой. С этой целью, мы публикуем live-build конфигурацию, которая используется для того, чтобы выстроить Kali образ так, чтобы вы могли настраивать его по вашему желанию. Таким образом, становится очень легко начать с этой опубликованной конфигурации и реализовать различные изменения, основанные на ваших потребностях, благодаря универсальности live-build.
Live-build включает в себя множество свойств для улучшения установленной системы, установки дополнительных файлов, установки дополнительных пакетов, выполнения произвольных команд и изменения значений, предварительно загруженных в debconf.

1.4.5 Надежная операционная система

Пользователи данного дистрибутива для обеспечения безопасности вполне законно хотят знать, что данной операционной системе можно доверять, и что она была разработана таким образом, чтобы каждый имел возможность

ознакомиться с исходным кодом. Kali Linux разработан небольшой командой высококлассных специалистов, работающих вполне открыто и придерживающихся лучших традиций безопасности: они загружают подписанные исходные пакеты, которые затем создаются на специализированных демонах сборки. Затем пакеты соединяются и распространяются как часть подписанного репозитория.

Работа, проделанная с пакетами, может быть полностью просмотрена через репозитории пакетирования Git [13] (которые содержит подписанные теги), которые используются для создания исходных пакетов Kali. Развитие каждого из пакетов можно отследить с помощью системы отслеживания Kali[14].

1.4.6 Используется в широком диапазоне устройств ARM

Kali Linux предоставляет двоичные пакеты для следующих ARM архитектур: armel, armhf, and arm64. Благодаря очень легко устанавливаемому образу, который был предоставлен Offensive Security, Kali Linux может работать на множестве различных устройствах, начина со смартфонов и планшетов, заканчивая Wi-Fi роутерами и компьютерами различных рамеров и форм.

1.5 Принципы работы Kali Linux

Хотя Kali Linux стремится следовать политике Debian, где это возможно, есть некоторые области, где мы приняли довольно специфические проектные решения из-за особых потребностей профессионалов в области безопасности.

1.5.1 Один root пользователь по умолчанию

Большинство дистрибутивов Linux очень разумно поощряют использование непривилегированной учетной записи при запуске системы и использовании такой утилиты, как sudo, в ситуациях,

[13] http://git.kali.org
[14] http://pkg.kali.org

когда необходимо обладать правами администратора. Это звучит вполне разумно, так как мы получаем дополнительный слой защиты между пользователем и потенциальными деструктивными или вредоносными операциями, или командами операционной системы. Это является особенно верным подходам для систем с более чем одним пользователем, где это является необходимой мерой безопасности, так как один пользователь может испортить или повредить работу многих пользователей.

В связи с тем, что большинство команд, включенных в Kali Linux, могут быть выполнены лишь в том случае, если пользователь обладает root правами, то аккаунт с данными привилегиями является аккаунтом по умолчанию. В отличие от других дистрибутивов Linux вам не будет предложено создать непривилегированного пользователя при установке Kali. Эта конкретная политика является серьезным отклонением от большинства систем Linux и, как правило, очень запутанна для менее опытных пользователей. Начинающие должны быть особенно осторожны при использовании Kali, поскольку самые разрушительные ошибки возникают при работе с привилегиями root.

1.5.2 Сетевые службы заблокированы по умолчанию

В отличие от Debian, Kali Linux блокирует любую установленную службу, которая будет прослушивать публичный сетевой интерфейс по умолчанию, такой как HTTP и SSH.

Основанием для такого решения является мотивация минимизировать возможность выявления во время тестирования на проникновения, когда любые неожиданные сетевые взаимодействия увеличивают риски быть обнаруженным.

Вы все еще можете включить любую службу по вашему выбору путем запуска systemctt enable *service.* Мы еще вернемся к этому немного позже в главе 5, “Настраиваем Kali Linux”.

1.5.3 Кураторский набор приложений

Debian стремится быть универсальной операционной системой и поэтому накладывает очень небольшие ограничения на то, что будет пакетировано, подразумевая, что у каждого пакета найдется свой пользователь.

В отличие от этого, Kali Linux не пакетирует все доступные инструменты для тестирования на проникновение. Вместо этого, мы намереваемся предоставить только самые лучшие инструменты с бесплатной лицензией, которые способны справиться практически со всеми заданиями, возникающими у пентестера.

Kali разработчики, работающие как пентестеры, руководят процессом отбора, а мы используем их опыт и квалификацию для того, чтобы сделать безупречный выбор. В некоторых случаях выбор является само собой разумеющимся фактом, но есть и другие, более сложные варианты, которые просто сводятся к личным предпочтениям.

Ниже приведены несколько основных моментов, на которые в первую очень обращается внимания во время оценки приложения:

- Польза от приложения в контексте тестирования на проникновение;
- Уникальность функциональной части приложения;
- Лицензия приложения;
- Требовательность приложения.

Поддержание постоянно обновляемого полезными инструментами для проведения тестирования на проникновение репозитория является довольно непростой задачей. Мы приветствуем предложения по инструментам в рамках выделенной категории *(New Tool Requests)* в Kali Bug Tracker[15]. Запрос на добавление нового инструмента лучшего всего оформлять, когда последний хорошо представлен, т.е. имеет хорошее описание, включая разъяснения того, почему данный инструмент является полезным,

[15] http://bugs.kali.org

как его можно сопоставить с другими похожими приложениями и т.д.

1.6 Подведем итоги

В данной главе мы представили вам Kali Linux, изложили немного информации об истории, прошлись по основным особенностям и привели примеры некоторых случаев использования. Мы также обсудили некоторые основные принципы, которые мы приняли во время разработки Kali Linux.

Главное из раздела:

- Kali Linux [16] является полностью готовым дистрибутивом контроля безопасности Linux, основанным на Debian GNU/Linux. Kali предназначен для профессионалов в сфере безопасности и IT администраторов, позволяя им проводить продвинутое тестирование на проникновение, криминалистический анализ и контроль безопасности;
- В отличие от большинства основных операционных систем Kali Linux является динамическим дистрибутивом. Это означает, что вы будете получать обновления каждый день;
- Дистрибутив Kali Linux основан на Debian Testing [17]. Именно поэтому, большинство пакетов, доступных в Kali Linux находятся напрямую в репозитории Debian;
- Несмотря на то, что основная задача Kali может быть выражена следующими словами “тестирование на проникновение и контроль безопасности”, в нем присутствуют некоторые инструменты, подходящие для следующих случаев использования: отслеживание сети для системных администраторов, криминалистический анализ, беспроводное отслеживание, установка встроенных устройств, установка на мобильные платформы и многое другое;
- Меню Kali облегчает доступ к инструментам для различных задач и действий, включая: анализ уязвимости, анализ веб приложений, оценка базы данных, взлом пароля, беспроводные

[16] https://www.kali.org
[17] https://www.debian.org/releases/testing/

атаки, обратная разработка, инструменты эксплуатации, sniffing и spoofing, пост эксплуатационные инструменты, криминалистические инструменты, инструменты для создания отчетов, инструменты для социальной инженерии, и системные службы;
- Kali Linux имеет множество дополнительных свойств и особенностей: использование в качестве живой (не установленной) системы, устойчивый и безопасный криминалистический режим, пользовательское ядро Linux, способность полностью настроить систему, надежная и безопасная операционная система, возможность ARM установки, безопасная сетевая политика по умолчанию, и кураторский набор приложений.

В следующей главе, мы погрузимся и опробуем Kali Linux благодаря его live режиму

Часть 2: Начало работы с Kali Linux

Содержание:

Ключевые слова главы:

- Скачивание ISO образ
- Живая загрузка

В отличие от других операционных систем начать работать с Kali Linux намного проще благодаря тому, что образ загрузочного диска является живым *ISO образом,* что в свою очередь означает, что вам не придётся следовать никаким предварительным инструкциям по установке. Это означает, что вы можете использовать один и тот же образ для тестирования, для использования в целях криминалистической экспертизы загрузочного образа USB или DVD, или для установки в качестве постоянной операционной системы на реальном или виртуальном оборудовании.

Из-за простоты в использовании очень легко забыть, что необходимо придерживаться определенных мер предосторожности. Пользователи Kali часто становятся жертвами людей с неблагими намерениями, будь то спонсируемые государством группы, элементы организованной преступности или отдельные хакеры. Открытый исходный код Kali Linux с легкостью позволяет злоумышленникам создавать и распространять поддельные версии, так что очень важно, чтобы вы выработали привычку скачивания информации с оригинальных источников и всегда проверяли подлинность и достоверность того, что вы скачали. Это особенно важно для специалистов в сфере безопасности, которые чаще всего имеют доступ к уязвимым сетям, и которым доверяются данные клиентов.

2.1 Загружаем ISO образ Kali

2.1.1 Где скачать

Единственный официальный ресурс, где вы беспрепятственно можете скачать ISO образ Kali Linux, это раздел Загрузки (Downloads) на веб сайте Kali. Из-за своей популярности в Интернете существуют многочисленные сайты, которые предлагают образ Kali для скачивания, но их нельзя считать надежными, т.к. они действительно могут быть заражены вредоносными программами или же смогут каким-либо иным образом нанести непоправимый урон вашей системе.

^ https://www.kati.org/downtoads/

Данный веб сайт доступен через *HTTPS,* что в свою очередь очень сильно затрудняет его подделку. Невозможно выполнить атаку «человек в середине», так как злоумышленнику также понадобится сертификат www.kali.org, подписанный центром сертификации Transport Layer Security (TLS), который в свою очередь заверен браузером жертвы. Т.к. центры сертификации существуют как раз для того, чтобы предотвращать подобного рода проблемы, они поставляют сертификаты только тем людям, чьи личности были проверены и тем, кто предоставил реальные доказательства того, что они контролируют соответствующий вебсайт.

cdimage.kali.org

Ссылки, найденные на странице загрузки, указывают на домен cdimage.kati.org, который перенаправляется к ближайшему к вам зеркалу, улучшая скорость передачи данных, уменьшая нагрузку на центральные серверы Kali.
Список доступных зеркал можно найти здесь:
^ http://cdimage.kati.org/README.mirrortist

2.1.2 Что скачивать

Официальная страница загрузки показывает короткий список образов ISO, как это можно увидеть на рисунке 2.1, “Список образов рекомендуемых для загрузки”.

Скачайте образы Kali Linux

Каждые несколько месяцев мы создаем новый образ Kali Linux, который доступен для скачивания. Эта страница предоставляет ссылки для скачивания последних выпусков Kali Linux. Чтобы просмотреть историю выпусков, ознакомьтесь с нашей страницей выпусков Kali Linux. Прошу вас учитывать, что здесь вы можете найти неофициальные и непроверенные выпуски:

http://cdimage.kali.org/kali-weekly/.

Image Name	Download	Size	Version	sha256sum
Kali 64 bit	ISO | Torrent	2.6G	2017.1	49b1c5769b909220060dc4c0e11ae09d97a270a80d259e05773101df62e11e9d
Kali 32 bit	ISO | Torrent	2.7G	2017.1	501b3747e5ac7c698217392fe49ec21dacee277404500fc49d4a0ee82625aabe
Kali 64 bit Light	ISO | Torrent	0.8G	2017.1	5c0f6300bf9842b724df92cb20e4637f4561ffc03029cdcb21af3902442ae9b0
Kali 32 bit Light	ISO | Torrent	0.8G	2017.1	6c83101ecf8702c7d93d32562e822b639d5c577314b448e3b8330995e0f07e0f
Kali 64 bit e17	ISO | Torrent	2.4G	2017.1	ae293cf679f38a4f17d090a272ccb13d7619e66d4502374154186c12891fb99c
Kali 64 bit KDE	ISO | Torrent	2.7G	2017.1	839741fec378114ff068df3ec2dbed9d8e4fae613e690d50b25ce9cc1468104b
Kali 64 bit Mate	ISO | Torrent	2.6G	2017.1	3ea748aa8c5f50d80f020acdbca5f0398ee90242bb4413c12985e1865186ca9e
Kali 64 bit Xfce	ISO | Torrent	2.5G	2017.1	8a17c2f54850585760b9d32a22e26df9a28f395b401753fa0a9b298aef4c4593
Kali 64 bit LXDE	ISO | Torrent	2.5G	2017.1	35eae65aaaabba8188dfd963e45b7b4d76e0684e7721c7d232cf18320b7cae3b
Kali armhf	Image | Torrent	0.5G	2017.1	a75199aa8a3d7b64561bc03fcd6e3ff8b94743c8769eecfaa4b719f04f7cbb63
Kali armel	Image | Torrent	0.4G	2017.1	180414422196f0797c1ea5f3c18682bc4b3ced871cb3e874e90de52dd4af877c

Рисунок 2.1 Список образов, рекомендуемых для загрузки

Все образы, помеченные как 32- или 64-разрядные, относятся к образам, подходящим для центральных процессоров (ЦП), которые используют большинство современных персональных компьютеров и ноутбуков. Если вы скачиваете образ для его последующего использования на полностью современной машине, вероятнее всего, что вам понадобится версия для 64-разрядного процессора. Если вы все еще не уверены, то теперь будьте уверены, что все 64-разрядные процессоры могут запускать 32-разрядные версии. Вы всегда можете загрузить и запустить 32-разрядный образ. Обратное утверждение, однако, остается неверным. Для более детальной информации ознакомьтесь с приведенной ниже информацией.

Если вы планируете установить Kali на встраиваемое устройство, смартфон, Chromebook, точку доступа, или на любое другое устройство, оснащенное ARM процессором, вы должны использовать образы Linux *armel* или *armhf*.

Является ли мой ЦП **32- или 64-разрядным?**

Работая в операционной системе Windows, вы сможете найти эту информацию путем запуска приложения Информации о системе (*System Information)* (находится в папке Оборудование > Системные инструменты). В окне «Информация о системе» вы сможете увидеть поле «Тип системы», которое будет содержать следующую информацию ”ПК на базе x64 ” для 64- разрядного ЦП или “ПК на базе x86 ” для 32-разрядного ЦП.

В системах OS X/macOS, нет стандартного приложения, которое отображало бы эту информацию, но вы все еще можете получить её на выводе команды uname –m введенной в терминале. На выводе вы увидите x86_64 для системы с 64-разрядным ядром (которая может работать только на 64-разрядном ЦП), а для систем с 32-разрядным ядром, вы увидите i386 или что-то похожее (i486, i586, или i686). Любое 32-битное ядро может работать на 64-битном процессоре, но поскольку Apple управляет оборудованием и программным обеспечением, то вряд ли вы столкнетесь с подобной конфигурацией.

В системе Linux, вы можете проверить данные в виртуальном файле /proc/cpuinfo. Если он содержит атрибут lm, то тогда ваш ЦП является 64-разрядным; в противном случае, он 32-разрядный. Следующая командная строка предоставит вам информацию о том, сколько разрядный процессор вы используете:

```
$ grep -qP '^flags\s*:.*\blm\b' /proc/cpuinfo && echo 64-bit
    ➥ || echo 32-bit
64-bit
```

Теперь, когда вы знаете какой именно 32-разрядный или 64-разрядный образ вам нужен, осталось проделать всего один шаг: выбрать тип образа. Оба образа по умолчанию для Kali Linux и для Kali Linux Light являются живыми ISO, которые могут быть использованы как для запуска живой системы, так и для старта процесса установки. Они отличаются только лишь набором предварительно установленных приложений. Образ по умолчанию идет с рабочим столом GNOME и огромным набором пакетов,

которые должны подойти практически для всех пентестеров, в то время как образ light идет с рабочим столом Xfce, (который, безусловно, является менее требовательным к системе), и имеет ограниченный набор пакетов, позволяющий только лишь выбирать необходимые вам приложения. Остальные образы используют альтернативные среды рабочего стола, но также идут с той же большой коллекцией пакетов, что и основной образ.

Как только вы определитесь с тем, какой образ вам нужен, вы сможете скачать его, нажав на ”ISO” в соответствующей строке. В качестве альтернативы, вы сможете скачать образ из одноранговой сети BitTorrent, нажав на ”Torrent,” в случае если у вас есть клиент BitTorrent, который предназначен для работы с расширением .torrent.

Пока выбранный вами образ ISO загружается, вы должны принять к сведению контрольную сумму, написанную в столбце sha256sum. После того, как вы загрузили ваш образ, используйте эту контрольную сумму, чтобы убедиться, что загруженное изображение соответствует тому, которую команда разработчиков Kali разместила в Интернете (см. Следующий раздел).

2.1.3 Проверка целостности и подлинности

Профессионалы в сфере безопасности должны проверять подлинность своих инструментов не только для защиты своих данных, но и для сохранения безопасности данных клиентов. Хотя страница загрузки Kali имеет TLS защиту, фактическая ссылка на скачивание указывает на незашифрованный URL-адрес, который не защищает от возможных атак типа «человек-в-середине». Тот факт, что Kali полагается на сеть внешних зеркал для распространения образа, не означает, что вы должны слепо доверять тому, что вы загружаете. Зеркало, на которое вы были направлены, возможно, было взломано или же конкретно вы являетесь жертвой атаки.

Для того чтобы облегчить этот процесс Kali всегда предоставляет контрольную сумму распространяемых образов. Но для того, чтобы сделать проверку эффективной, вы должны быть уверенны,

что контрольная сумма, полученная вами, является той же самой, что и контрольная сумма, опубликованная разработчиками Kali Linux. Есть несколько способов выяснить это.

Полагаемся на TLS-защищенные вебсайты

Когда вы извлекаете контрольную сумму с TLS защищенной страницы загрузки, её источник точно гарантируется сертификатом защиты модели X.509: содержание, которое вы видите, идет с веб сайта, который находится под контролем личности, запросившей TLS сертификат.

Теперь вы должны сгенерировать контрольную сумму вашего загруженного образа и убедиться в том, что она полностью соответствует тому, что написано на сайте Kali:

```
$ sha256sum kali-linux-2017.1-amd64.iso
49b1c5769b909220060dc4c0e11ae09d97a270a80d259e05773101df62e11e9d  kali-linux-2016.2-amd64.iso
```

Если вы сгенерировали контрольную сумму, которая полностью соответствует той, что находится на странице загрузки Kali Linux, то вы скачали нужный файл. Если контрольная сумма отличается, то это является проблемой, хотя это не означает взлом или атаку; загрузки иногда подвергаются изменениям и получают повреждения, по мере того как они перемещаются в Интернете. Попробуйте повторить свою загрузку снова, но уже с другого официального зеркала Kali, если конечно это является возможным (смотрите “cdimage.kali.org” для получения большей информации о доступных зеркалах).

Полагаемся на PGP Web of Trust

Если вы не доверяете использованию HTTPS для аутентификации, то вы в какой-то мере являетесь параноиком, но это вполне справедливо. Существует огромное количество примеров скверно управляемых центров сертификации, которые выдавали поддельные сертификаты, что в свою очередь не очень хорошо заканчивалось. Вы также можете оказаться жертвой «дружеской» атаки человека-по-середине, которая реализуется во многих корпоративных сетях, используя специализированное, доверенное встроенное в браузер хранилище, которое

представляет поддельные сертификаты для зашифрованных веб-сайтов, позволяя корпоративным аудиторам отслеживать зашифрованный трафик.

Для подобных случаев мы также предоставляем GnuPG ключ, который мы используем для того, чтобы подписать контрольную сумму образа, предоставленного нами. Идентификаторы ключа и его отличительные черты показаны ниже:

```
pub   rsa4096/0xED444FF07D8D0BF6 2012-03-05 [SC] [expires: 2018-02-02]
      Key fingerprint = 44C6 513A 8E4F B3D3 0875  F758 ED44 4FF0 7D8D 0BF6
uid                   [  full  ] Kali Linux Repository <devel@kali.org>
sub   rsa4096/0xA8373E18FC0D0DCB 2012-03-05 [E] [expires: 2018-02-02]
```

Этот ключ является частью глобальной сети доверия, потому что он был подписан как минимум мной (Рафаэлем Херцогом), а я являюсь частью этой сети доверия благодаря моему интенсивному использованию GnuPG в качестве разработчика Debian.

Модель безопасности PGP/GPG является довольно таки уникальной. Любой может сгенерировать какой-либо ключ с любой идентичностью, но вы сможете доверять этому ключу только в том случае, если он уже был подписан другим ключом, которому вы точно доверяете. Когда вы подписываете ключ, вы подтверждаете, что вы встречали владельца ключа и знаете, что соответствующая идентификация верна. И вы определяете исходный набор ключей, которым вы доверяете, что, очевидно, включает в себя ваш собственный ключ.

Эта модель имеет свои ограничения, поэтому вы можете выбрать загрузить публичный ключ Kali через HTTPS (или с сервера ключей) и просто решить, что вы доверяете ему, т.к. он совпадает с другим, который мы публиковали во многих местах, включая и пример, указанный выше, в этой книге.

```
$ wget -q -O - https://www.kali.org/archive-key.asc | gpg --import
[ or ]
$ gpg --keyserver hkp://keys.gnupg.net --recv-key ED444FF07D8D0BF6
gpg: key 0xED444FF07D8D0BF6: public key "Kali Linux Repository <devel@kali.org>" imported
gpg: Total number processed: 1
gpg:               imported: 1  (RSA: 1)
[...]
$ gpg --fingerprint 7D8D0BF6
[...]
      Key fingerprint = 44C6 513A 8E4F B3D3 0875  F758 ED44 4FF0 7D8D 0BF6
[...]
```

После того, как вы извлечете ключ, вы можете использовать его для проверки контрольной суммы предлагаемых образов. Давайте скачаем файл с контрольной суммой (SHA256SUMS) и связанным файлом подписи (SHA256SUMS.gpg) и затем проверим подпись:

```
$ wget http://cdimage.kali.org/current/SHA256SUMS
[...]
$ wget http://cdimage.kali.org/current/SHA256SUMS.gpg
[...]
$ gpg --verify SHA256SUMS.gpg SHA256SUMS
gpg: Signature made Thu 16 Mar 2017 08:55:45 AM MDT
gpg:                using RSA key ED444FF07D8D0BF6
gpg: Good signature from "Kali Linux Repository <devel@kali.org>"
```

Если вы получите сообщение, подобное тому, что вы видели выше “Good signature”, то вы можете смело доверять содержимому файла SHA256SUMS и использовать его для проверки тех файлов, что вы скачали. В противном случае возникает проблема. Вам следует проверить, скачали ли вы файл с законного (официального) зеркала Kali Linux.

Обратите внимание, что вы можете использовать следующую командную строку для того, чтобы проверить имеет ли скачанный файл такую же контрольную сумму, которая указана в SHA256SUMS, при условии, что загруженный ISO-файл находится в том же каталоге:

```
$ grep kali-linux-2017.1-amd64.iso SHA256SUMS | sha256sum -c
kali-linux-2017.1-amd64.iso: OK
```

Если в качестве ответа вы не получили OK, тогда файл, который вы скачали, является отличным от того, который был официально выпущен командой Kali. Этому файлу нельзя доверять и его не стоит использовать.

2.1.4 Копирование изображения на DVD-ROM или USB носитель

Сам по себе ISO образ имеет довольно ограниченное применение, если вы конечно не хотите использовать Kali Linux только лишь на виртуальной машине. В противном случае, вам нужно создать загрузочный диск или же загрузочный USB накопитель для того, чтобы запустить вашу машину с Kali Linux.

Здесь мы не будем рассказывать о том, как создать загрузочный диск, т.к. этот процесс очень сильно разнится в зависимости от операционной системы, платформы или среды, но в большинстве случаев, правый щелчок мышки на файле .iso вызовет контекстное меню, которое выполнит приложение прожига DVD-ROM. Давайте попробуем!

2.2 Загрузка ISO образа Kali в режиме реального времени

2.2.1 На реальном компьютере

Вам понадобится либо предварительно подготовленный USB накопитель (как это сделать было детально изложено в предыдущем разделе) или загрузочный диск с образом ISO Kali Linux.

IOS/UEFI отвечает за ранний процесс загрузки и может быть настроен через часть программного обеспечения под названием «Настройка» (Setup). В частности, он позволяет пользователям выбирать, какое загрузочное устройство является предпочтительным. В этом случае вы хотите выбрать либо привод

DVD-ROM, либо USB-накопитель, в зависимости от того, какое устройство вы создали.

Запуск программы установки обычно включает в себя очень быстрое нажатие определенной клавиши после включения компьютера. Чаще всего эта клавиша является Del или Esc, а иногда F2 или F10. В большинстве случаев эта клавиша моментально отображается на экране, когда компьютер включается, прежде чем загрузится операционная система.

Как только BIOS/UEFI правильно настроен для загрузки с вашего устройства, загрузку Kali Linux можно продолжить без особых проблем, просто вставьте загрузочный диск или ваш USB накопитель и запустите компьютер.

Отключите безопасную загрузку

Хотя образы Kali Linux могут быть загружены в режиме UEFI, они не поддерживают режим *безопасной загрузки.* Вы должны заблокировать это свойства в «Настройках» (Setup).

2.2.2 На виртуальной машине

Виртуальные машины имеют множество преимуществ для пользователей Kali Linux. Они особенно полезны для тех пользователей, которые хотят опробовать Kali Linux, но не готовы установить её на свою машину или же для тех, у кого есть довольно мощная система и они хотят использовать несколько операционных систем одновременно. Это довольно-таки распространенный выбор среди многих пентестеров и профессионалов в сфере безопасности, которым нужен доступ к широкому спектру инструментов, предоставляемых Kali Linux, но также им по-прежнему необходимо иметь полный доступ к их первичной операционной системе. Это также позволяет им архивировать и безопасно удалять виртуальную машину и любые данные клиента, которые она может содержать, без необходимости полной переустановки их операционной системы.

Функции моментального снимка для виртуализации также позволяют легко экспериментировать с потенциально опасными операциями, такими как анализ вредоносных программ, позволяя легко выйти, запустив режим восстановления предыдущего моментального снимка.

Существует множество инструментов для виртуализации доступных большинству операционных систем, включая *VirtualBox®, VMware Workstation®, Xen, KVM,* и *Hyper-V,* и это лишь немногие из них. В конечном счете, вы все равно будете использовать один из них, который идеально соответствует вашим задачам, но здесь мы рассмотрим два самых часто используемых инструмента в контексте рабочего стола: *VirtualBox®* и *VMware Workstation Pro®,* запущенный на Windows 10. Если вы не ограничены корпоративной политикой или какими-либо личными предпочтениями, мы рекомендуем вам сначала использовать VirtualBox, т.к. она является бесплатной, отлично работает (в большинстве случаев), имеет открытый исходный код и является доступной для большинства операционных систем.

В следующем разделе мы предположим, что уже установили соответствующий инструмент виртуализации и ознакомились со всеми основными операциями, необходимыми для работы.

Предварительные замечания

Чтобы полностью воспользоваться преимуществами виртуализации, у вас должен быть процессор с соответствующими функциями виртуализации, и они не должны быть отключены в BIOS/UEFI. Двойная проверка для всех “Intel® Virtualization Technology” и/или “Intel® VT-d Feature” опции в меню «Настройки» (Setup).

У вас также должна быть 64-разрядная операционная система, вроде amd64 архитектуры для дистрибутивов Linux на основе Debian, архитектуры x86_64 для Linux на основе RedHat и для Windows… 64-разрядная система для Windows.

Если у вас не хватает чего-то из необходимых требований для корректной работы, то инструмент виртуализации либо не будет

работать должным образом, либо он будет ограничен запуском 32-разрядной гостевой операционной системы.

Ввиду того, что инструменты виртуализации подключаются к основной операционной системе на низком уровне, то у вас постоянно будут возникать несовместимости между ними. Не стоит ожидать того, что эти инструменты будут одновременно работать должным образом. Кроме того, имейте ввиду, что профессиональные версии Windows идут сразу со встроенным и работающим *Hyper-V,* который в любом случае будет конфликтовать с тем инструментом виртуализации, который вы выбрали.

Чтобы отключить его, выполните «Включить или отключить функции Windows» в настройках Windows.

VirtualBox

После первоначальной установки, главный экран VirtualBox будет выглядеть так, как показано на рисунке 2.6, “Начальный экран Virtual-Box”

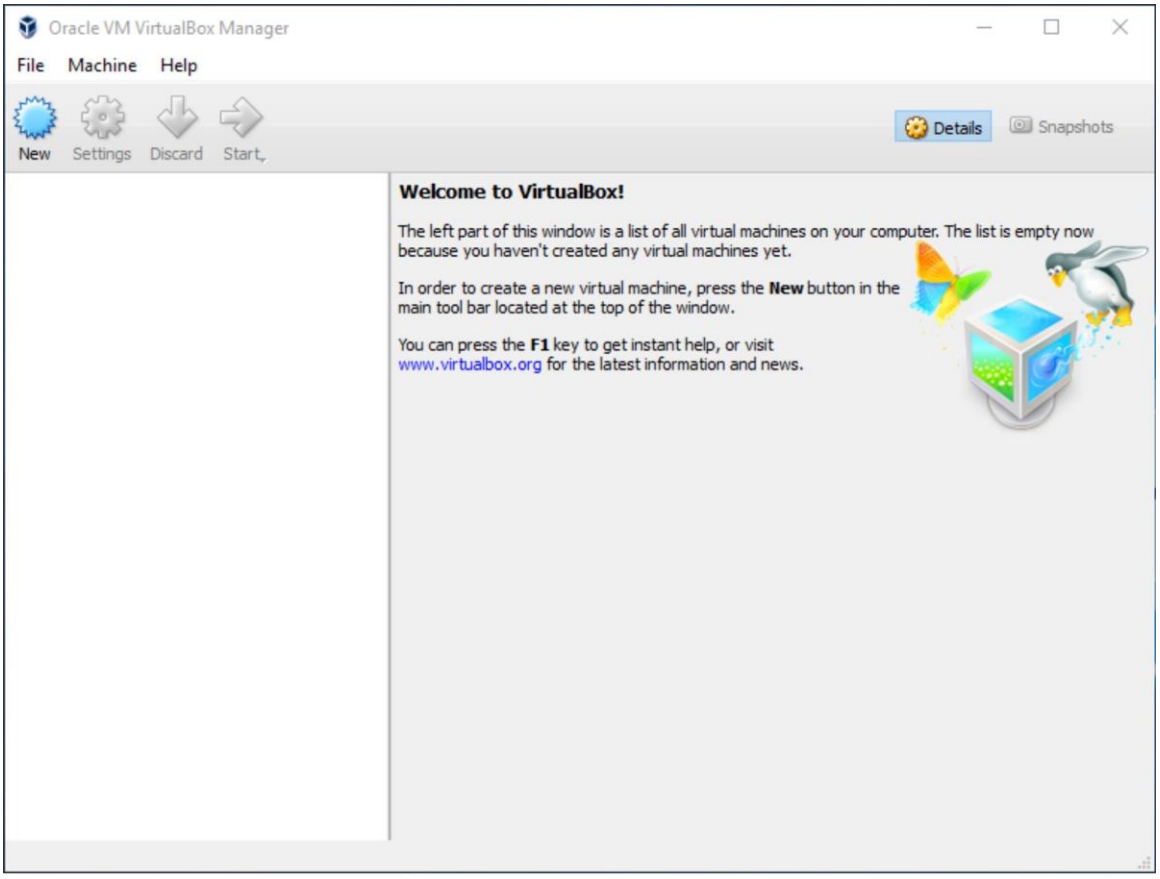

Figure 2.6 *VirtualBox's Start Screen*

Нажмите на New (Рисунок 2.7, “Название и операционная система”) чтобы запустить мастер, который проведет вас через несколько шагов, необходимых для ввода всех необходимых параметров для запуска новой виртуальной машины.

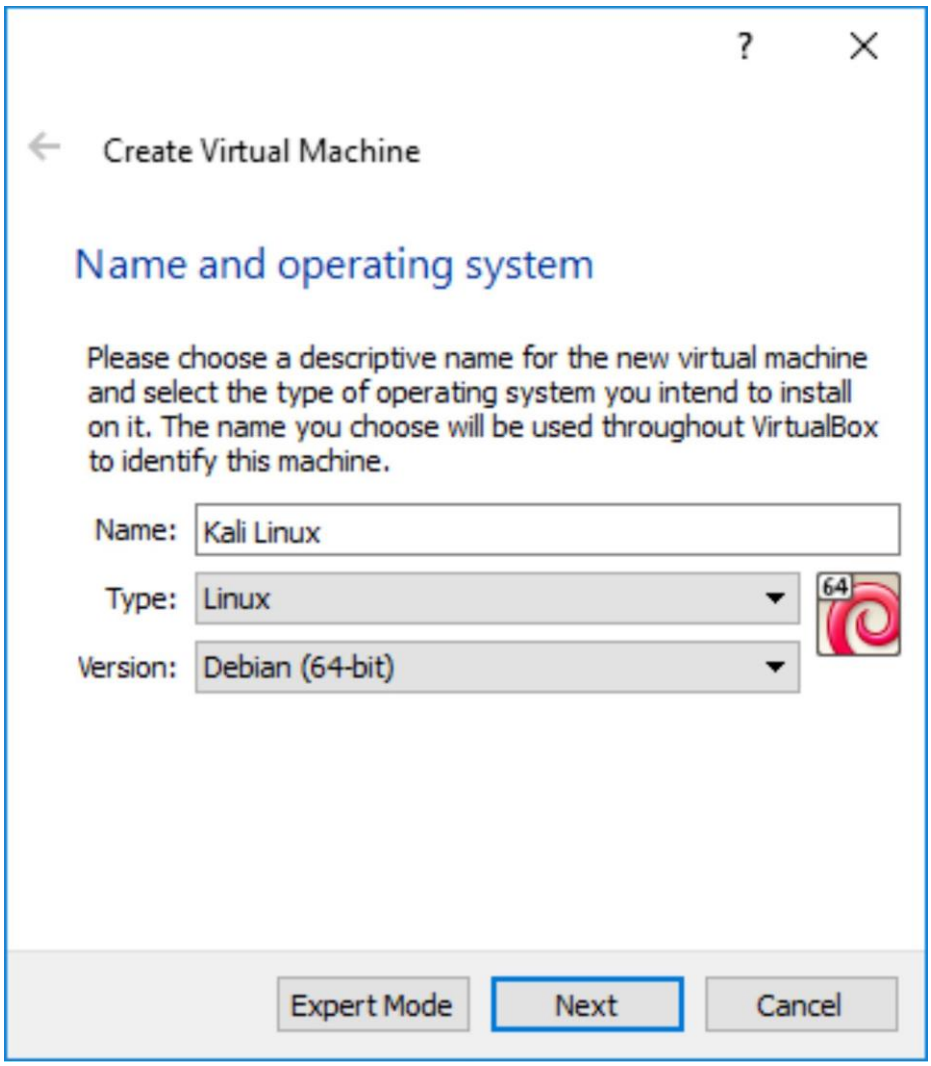

Рисунок 2.7 *Название и операционная система*

На первом шаге, показанном на Рисунке 2.7, “Название и операционная система”, вы должны назначить имя для своей новой виртуальной машины. Используйте “Kali Linux.” Вы также должны обозначить, какую именно операционную систему вы хотите использовать. Т.к. Kali Linux основан на Debian GNU/Linux, выберете тип систем Linux и версию Debian (32-разрядная) или Debian (64- разрядная). Хотя любая другая версия Linux, скорее всего, будет работать, это поможет отличать различные виртуальные машины, которые вы, возможно, будете устанавливать.

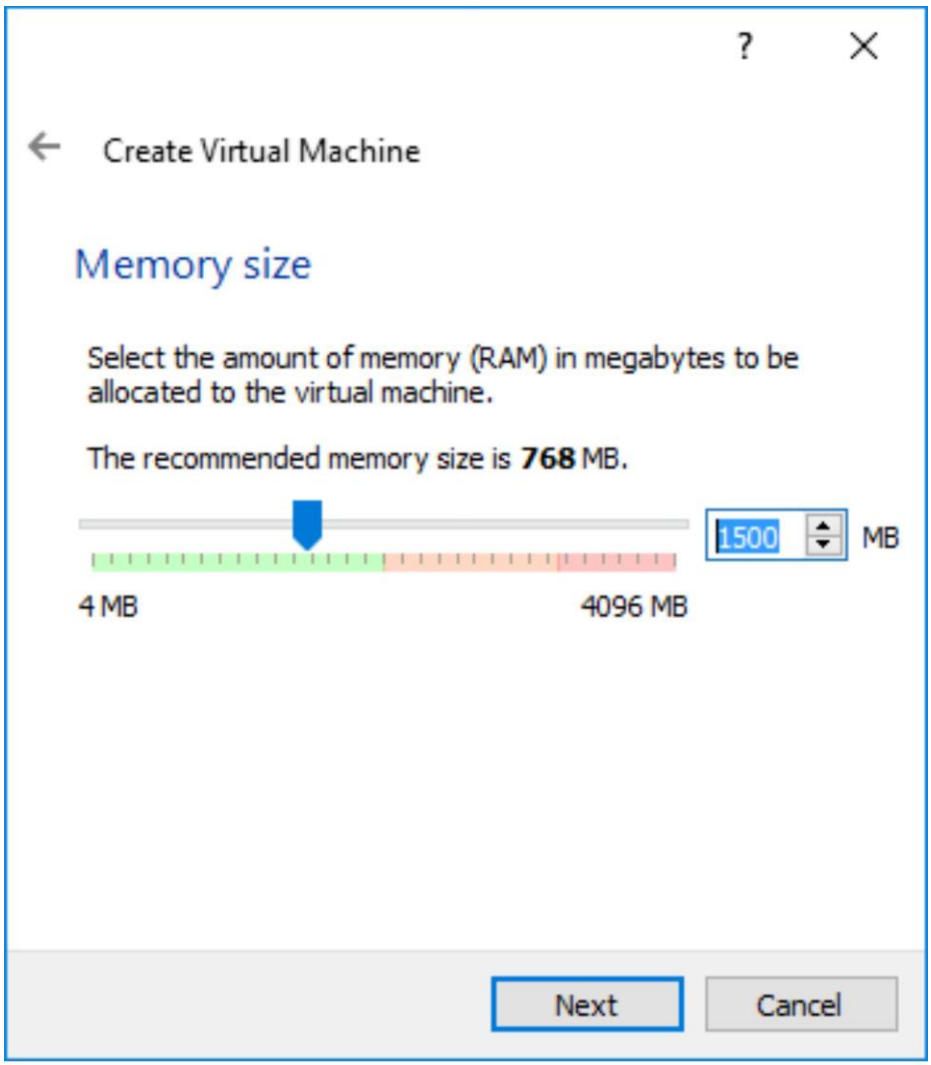

Figure 2.8 *Memory Size*

На втором этапе вы должны решить, сколько памяти будет выделено виртуальной машине. Хотя рекомендуемый размер 768 МБ является приемлемым для виртуальной машины Debian, выступающей в роли сервера, этого определенно будет недостаточно для запуска настольной системы Kali, особенно для живой системы Kali Linux, поскольку живая система использует память для хранения изменений, которые были внесены в файловую систему. Мы рекомендуем увеличить значение до 1500 МБ (рисунок 2.8, «Размер памяти» [стр. 28]) и настоятельно рекомендуем выделить не менее 2048 МБ ОЗУ.

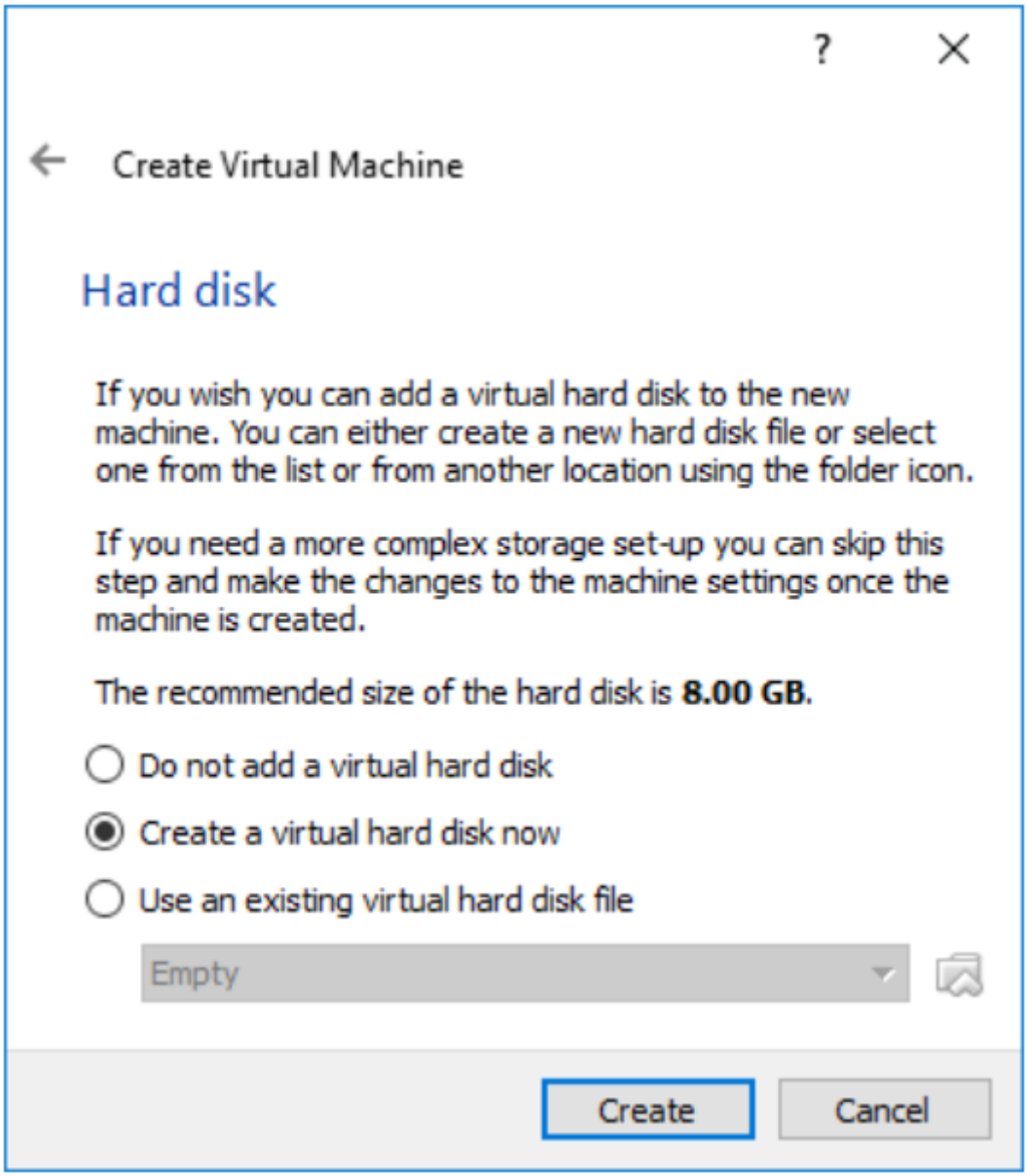

Рисунок 2.9 *Жесткий диск*

На третьем шаге (см. Рис. 2.9, «Жесткий диск») вам предлагается выбрать физический или виртуальный жесткий диск для новой виртуальной машины. Хотя жесткий диск не требуется для запуска Kali Linux в качестве живой системы, добавьте его, для процедуры установки, которую мы продемонстрируем позже, в главе 4 «Установка Kali Linux».

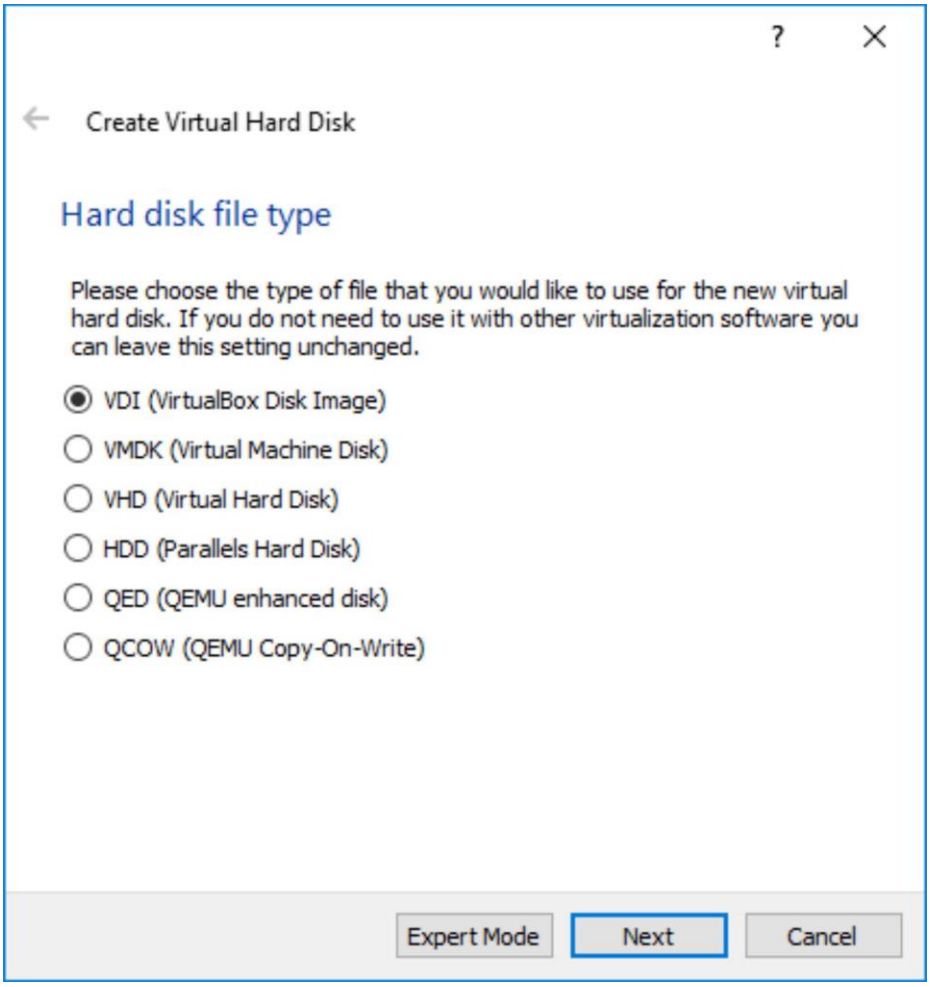

Рисунок 2.10 *Тип файла жесткого диска*

Содержимое жесткого диска виртуальной машины хранится на главной машине в виде файла. VirtuBox способен хранить содержимое жесткого диска в нескольких форматах (см. Рис. 2.10, «Тип файла жесткого диска» [стр. 30]): значение по умолчанию (VDI) соответствует собственному формату VirtualBox; VMDK - формат, используемый VMware; QCOW - формат, используемый QEMU. Сохраняйте значение по умолчанию, потому что у вас нет причин его изменять. Возможность использовать несколько форматов интересна главным образом, когда вы хотите переместить виртуальную машину из одного инструмента виртуализации в другой.

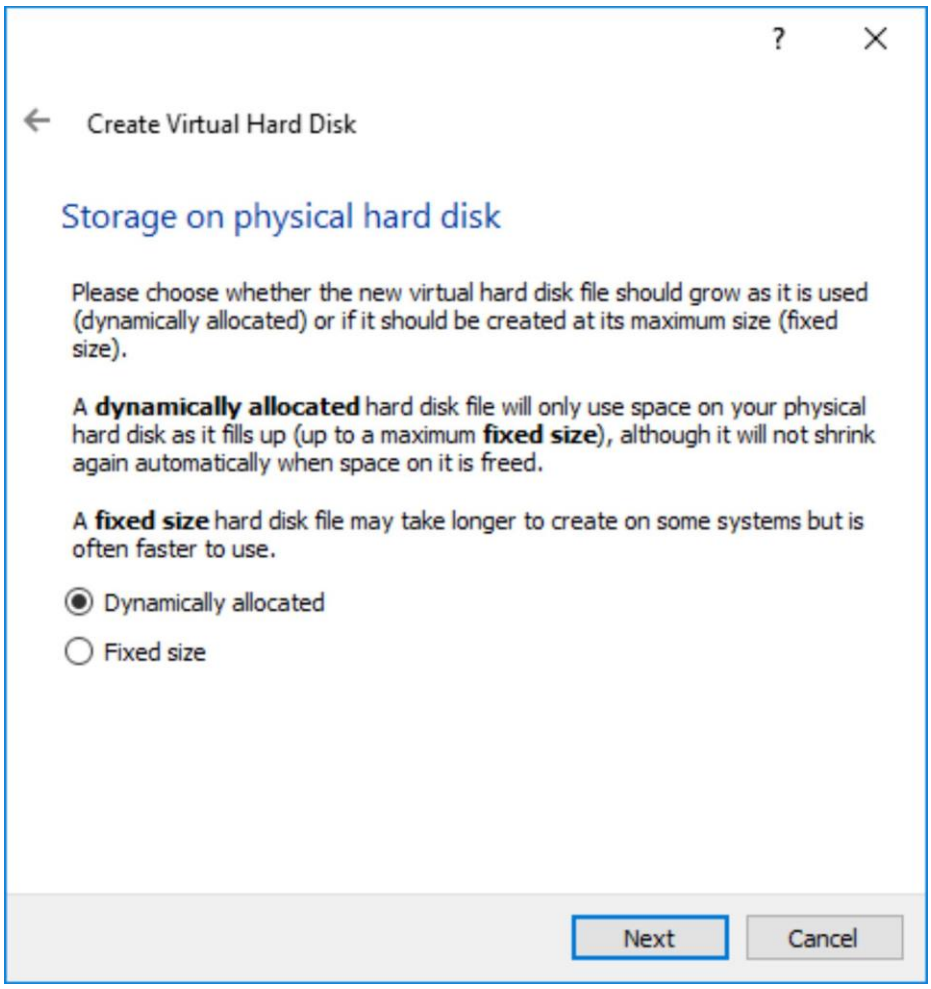

Рисунок 2.11 *Хранение на физическом жестком диске*

В пояснительном тексте на рисунке 2.11 «Хранение на физическом жестком диске» [стр. 31] четко описаны преимущества и недостатки динамического и фиксированного распределения дисков. В этом примере мы оставляем выбор по умолчанию (динамическое распределение), поскольку мы используем ноутбук с SSD-дисками. Мы не хотим тратить пространство и не нуждаемся в дополнительной доле производительности, поскольку наша машина и так является довольно быстрой.

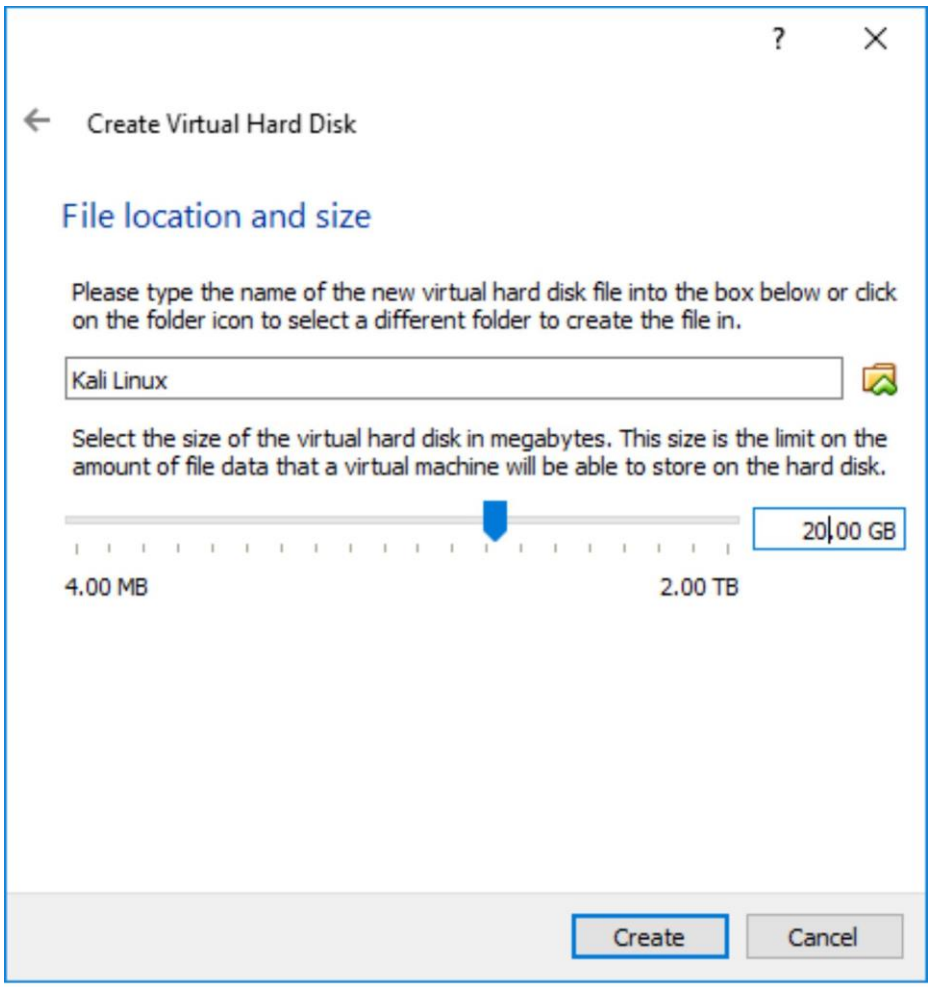

Figure 2.12 *File Location and Size*

Стандартный размер жесткого диска по умолчанию 8 ГБ, показанный на рисунке 2.12, «Местоположение и размер файла» [стр. 32] недостаточен для стандартной установки Kali Linux, поэтому увеличьте размер до 20 ГБ. Вы также можете настроить имя и расположение образа диска. Это может быть удобно, если на жестком диске недостаточно места, что позволяет сохранять образ на внешнем диске.

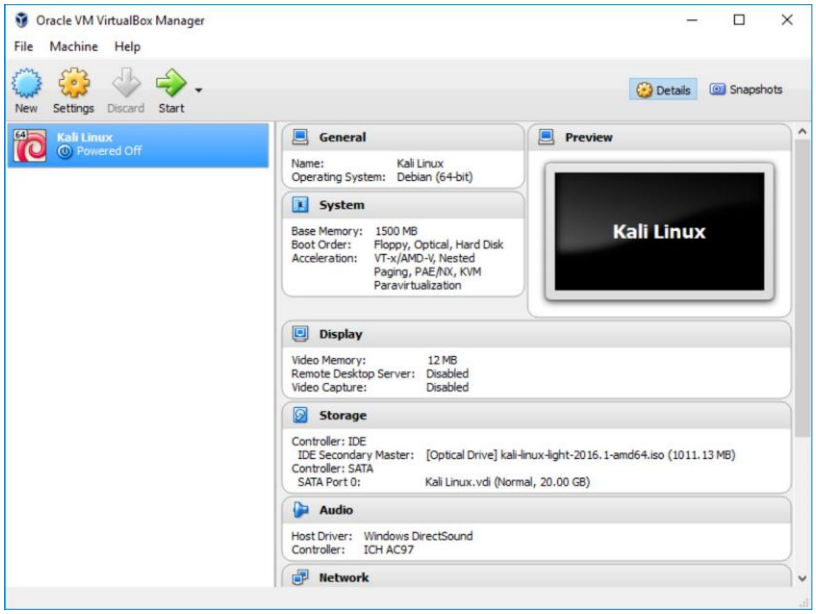

Рисунок 2.13 *Новая виртуальная машина появляется в списке*

Итак, виртуальная машина была создана, но вы все еще не можете её запустить, потому что на ней не установлена операционная система. Вы также можете выставить некоторые настройки. Нажмите «Настройки» на экране «Диспетчер виртуальной машины» и сейчас мы постараемся рассмотреть некоторые из наиболее полезных настроек.

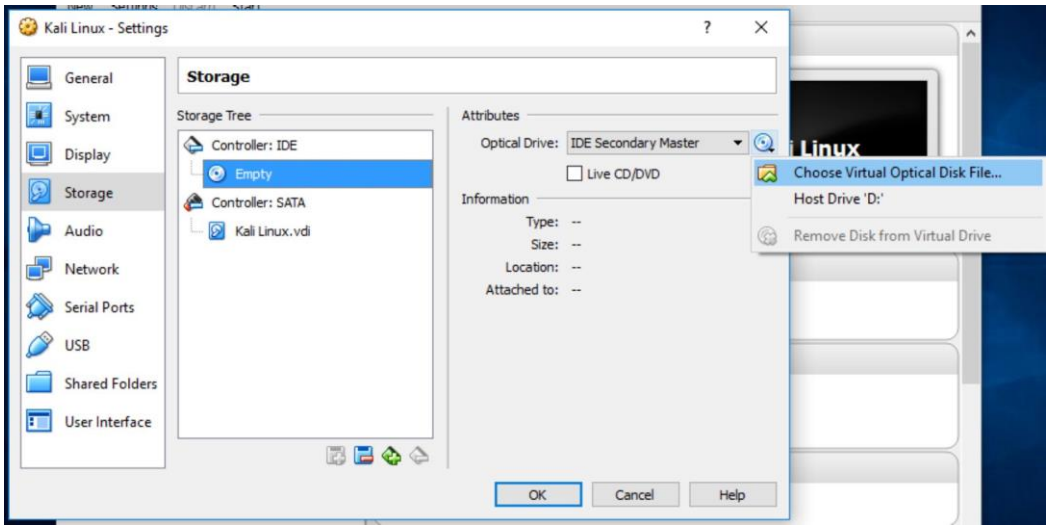

Рисунок 2.14 *Настройки хранения*

На экране Хранения (рис. 2.14 «Настройки хранения» [стр. 33]) вам следует сопоставить образ ISO Kali Linux с виртуальным устройством чтения CD/DVD-ROM. Сначала выберите привод CD-ROM в списке дерева хранения, а затем щелкните значок

маленького компакт-диска справа, чтобы отобразить контекстное меню, в котором вы можете выбрать файл виртуального оптического диска.

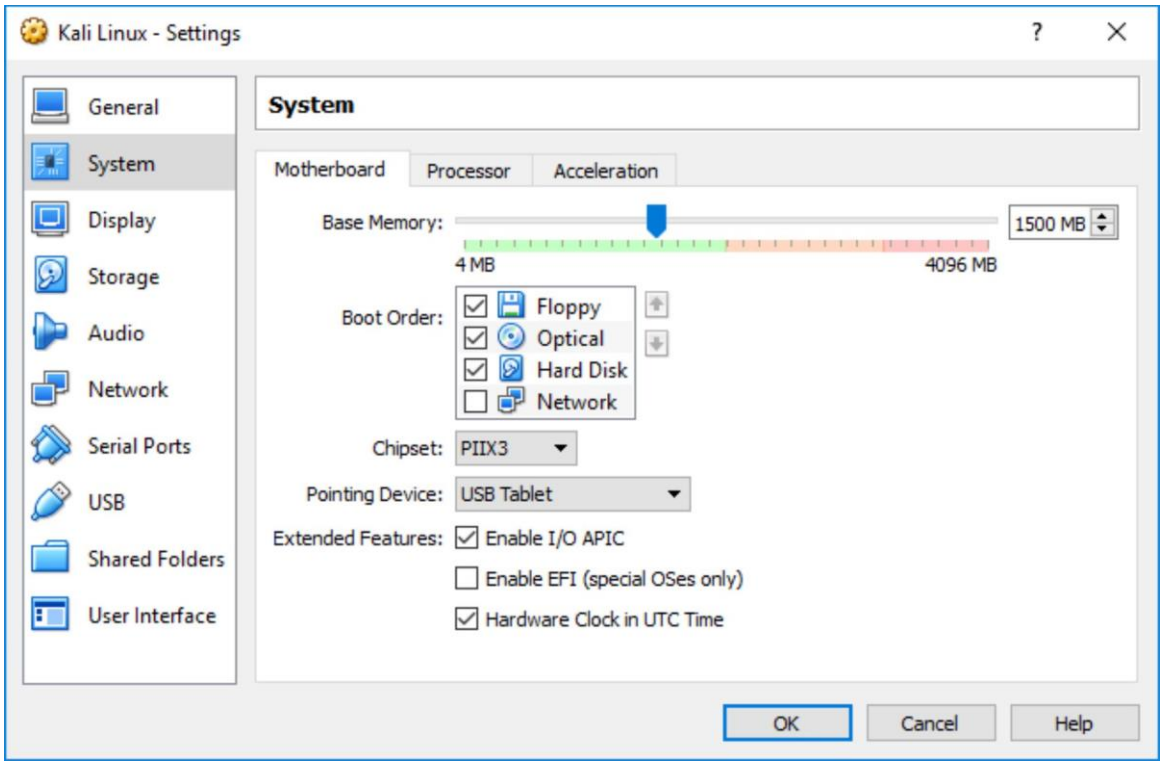

Рисунок 2.15 *Настройки системы: материнская плата*

На системном экране (Рисунок 2.15,“ Настройки системы: материнская плата” [стр. 34]), вы найдете вкладку «Материнская плата». Убедитесь, что порядок загрузки выставлен таким образом, что система сначала будет пытаться загружаться с оптических приводов, прежде чем загружаться с жесткого диска. Также вы сможете найти вкладку, в которой вы можете изменить объем памяти, выделенной виртуальной машине, в случае необходимости.

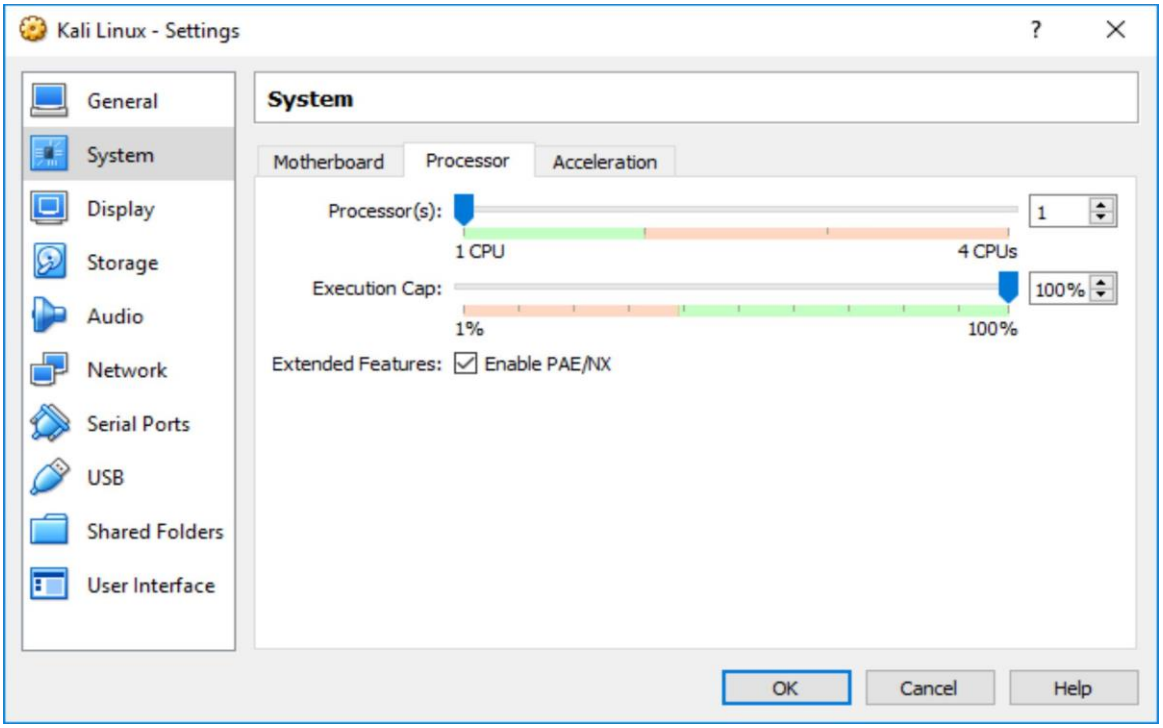

Рисунок 2.16 *Настройки системы: Процессор*

На том же экране, но на вкладке “Процессор” (Рисунок 2.16, “Настройки системы: Процессор”), вы сможете настроить количество процессоров, которое будет иметь ваша виртуальная машина. Важно запомнить, что если вы используете 32-разрядный образ, разрешите PAE/NX, иначе Kali образ не сможет загрузиться, т.к. вариант ядра по умолчанию, используемый Kali для i386 (под названием “686-pae”) скомпилирован таким образом, что требует поддержки Physical Address Extension (PAE) в вашем процессоре.

Существует также множество других параметров, которые могут быть настроены, такие как, например, настройки сети (которые определяют, каким образом будет обрабатываться трафик на сетевой карте), но тех настроек, которые были описаны выше, вполне достаточно для загрузки рабочей живой системы Kali Linux. Наконец, нажмите «Загрузка», и виртуальная машина должна загрузиться должным образом, как показано на рисунке 2.17, «Экран загрузки Kali Linux в VirtualBox» [стр. 36]. Если же этого не произошло, то внимательно просмотрите все настройки и повторите попытку.

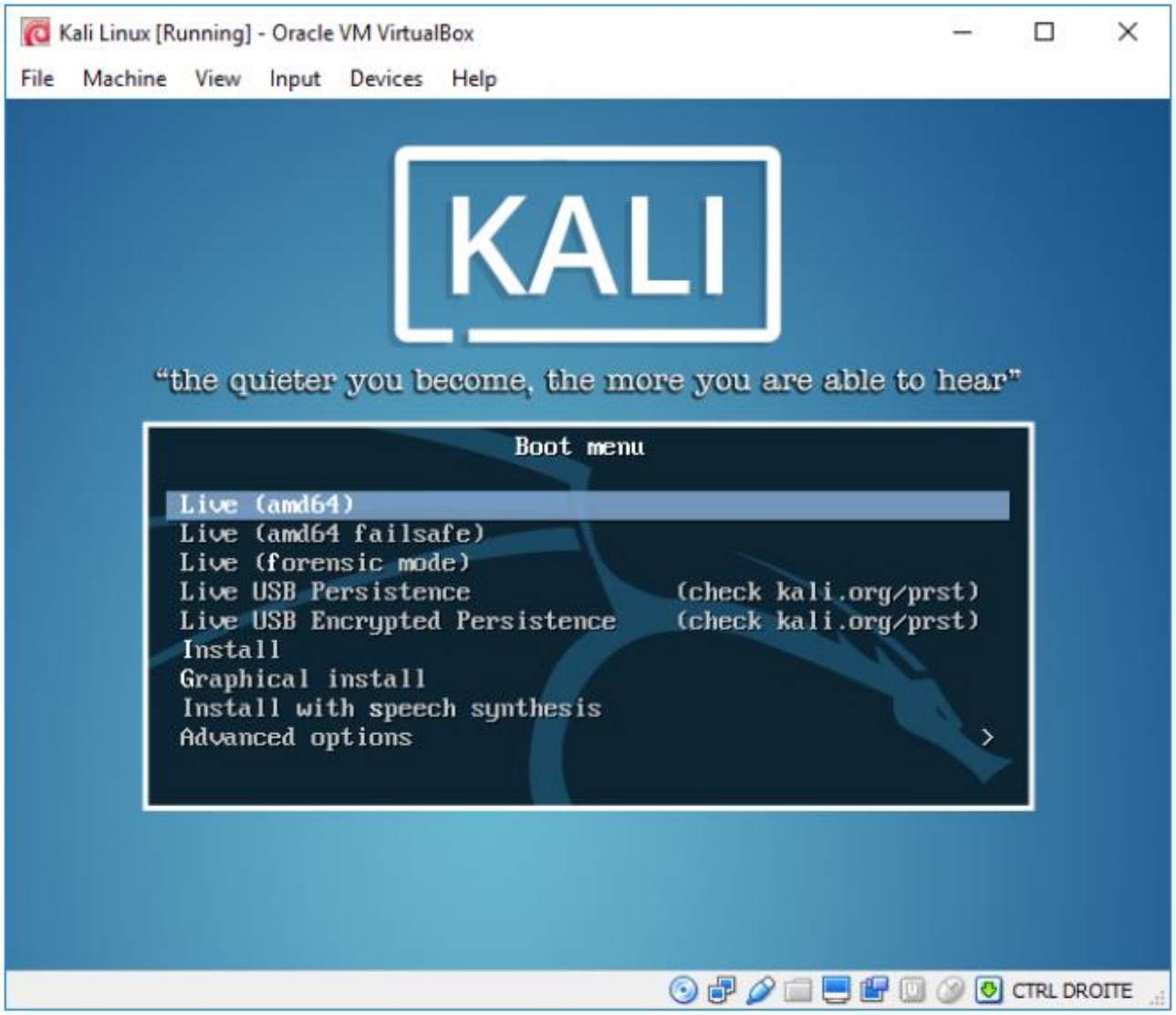

Рисунок 2.17 *Экран загрузки Kali Linux в VirtualBox*

VMware

VMware Workstation Pro очень похожа на *VirtualBox,* особенно, своим интерфейсом и свойствами, потому что они обе были разработаны для использования desktop usage, но тем не менее, процесс создания новой виртуальной машины немного отличается.

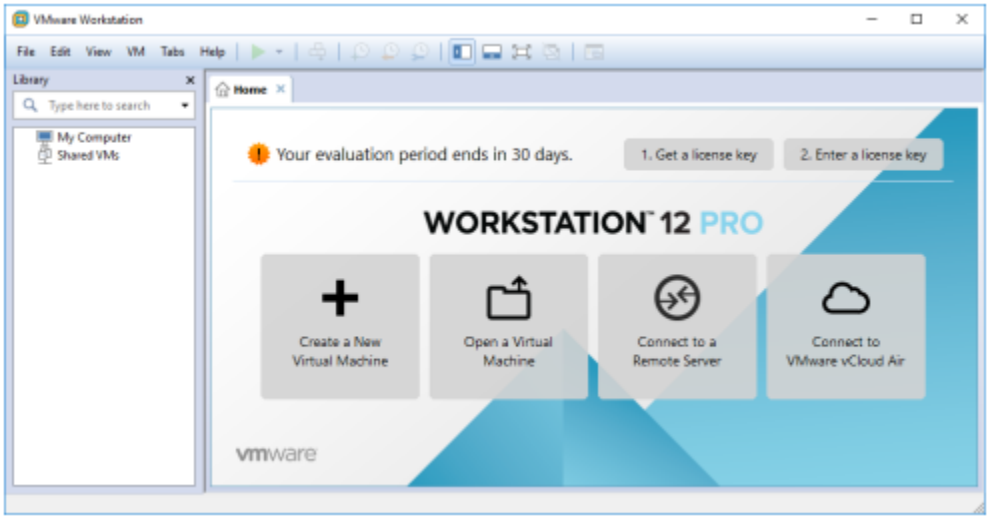

Figure 2.18 *VMware Start Screen*

На начальном экране, показанном на рисунке 2.18, «Начальный экран VMware» [стр. 37], отображается большая кнопка «Создать новую виртуальную машину», запускающая мастер, который поможет вам создать виртуальную машину.

Figure 2.19 *New virtual Machine Wizard*

На первом этапе вы должны решить, хотите ли вы получить информацию о расширенных настройках в процессе установки. В этом примере особых требований нет, поэтому выберите обычную установку, как показано на рисунке 2.19, «Мастер создания новой виртуальной машины»

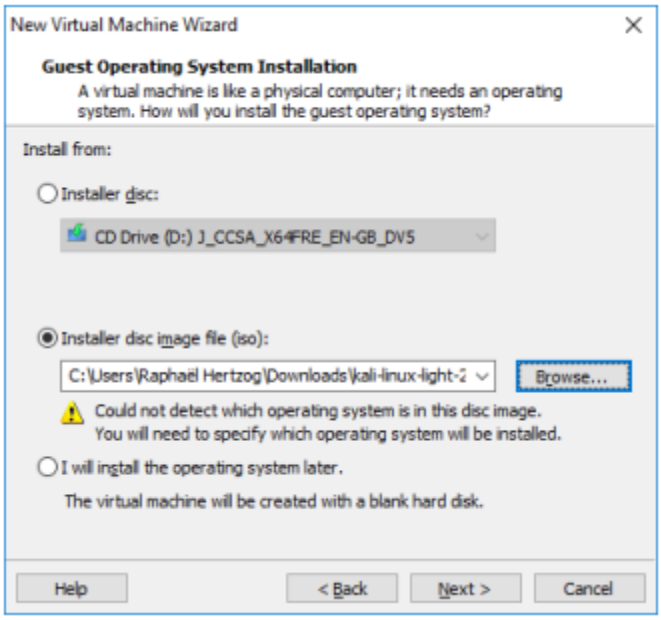

Рисунок 2.21 *Выберите гостевую операционную систему*

Когда операционная система (ОС) не будет обнаружена на выбранном ISO-образе, мастер спросит вас, какой тип гостевой ОС вы собираетесь запустить. Вы должны выбрать операционную систему «Linux» для ОС и версию «Debian 8.x», как показано на рисунке 2.21, «Выберите гостевую операционную систему» [стр. 38].

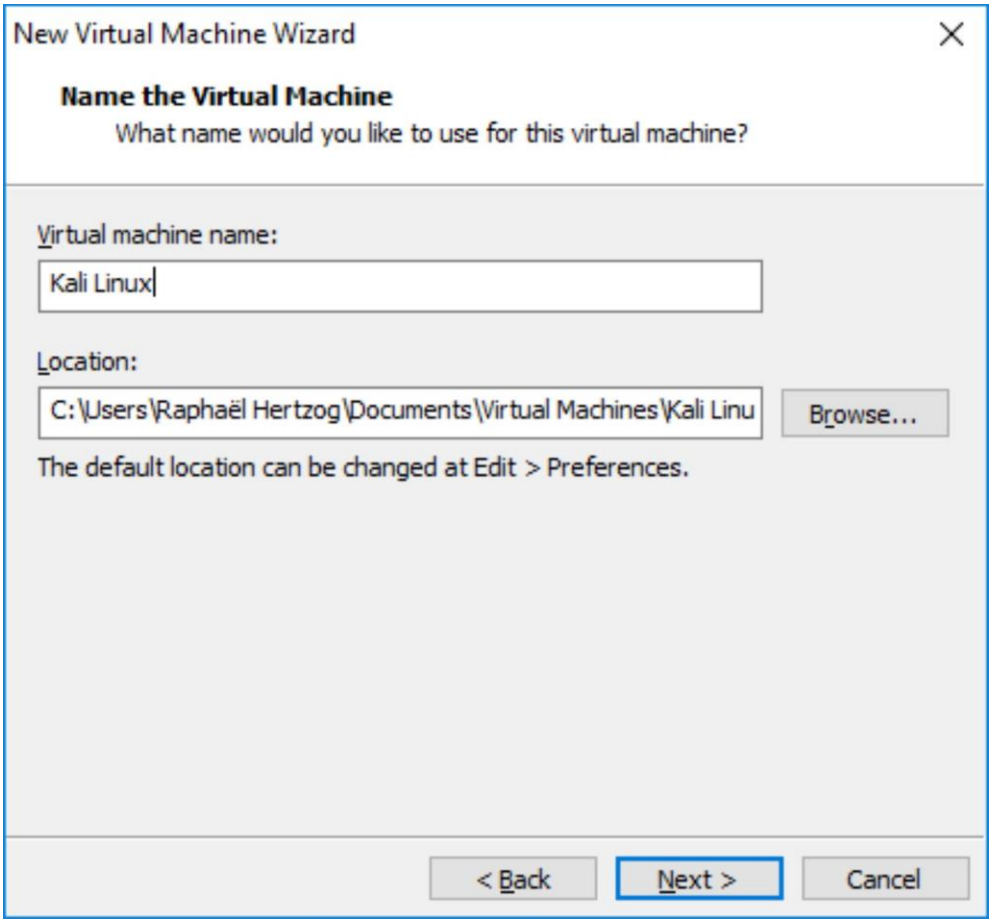

Рисунок 2.22 *Назовите виртуальную машину*

Выберите ”Kali Linux” в качестве имени для виртуальной машины. (Рисунок 2.22, “Назовите виртуальную машину”). Как и в VirtualBox, у вас также есть возможность хранить файлы виртуальной машины в альтернативном месте.

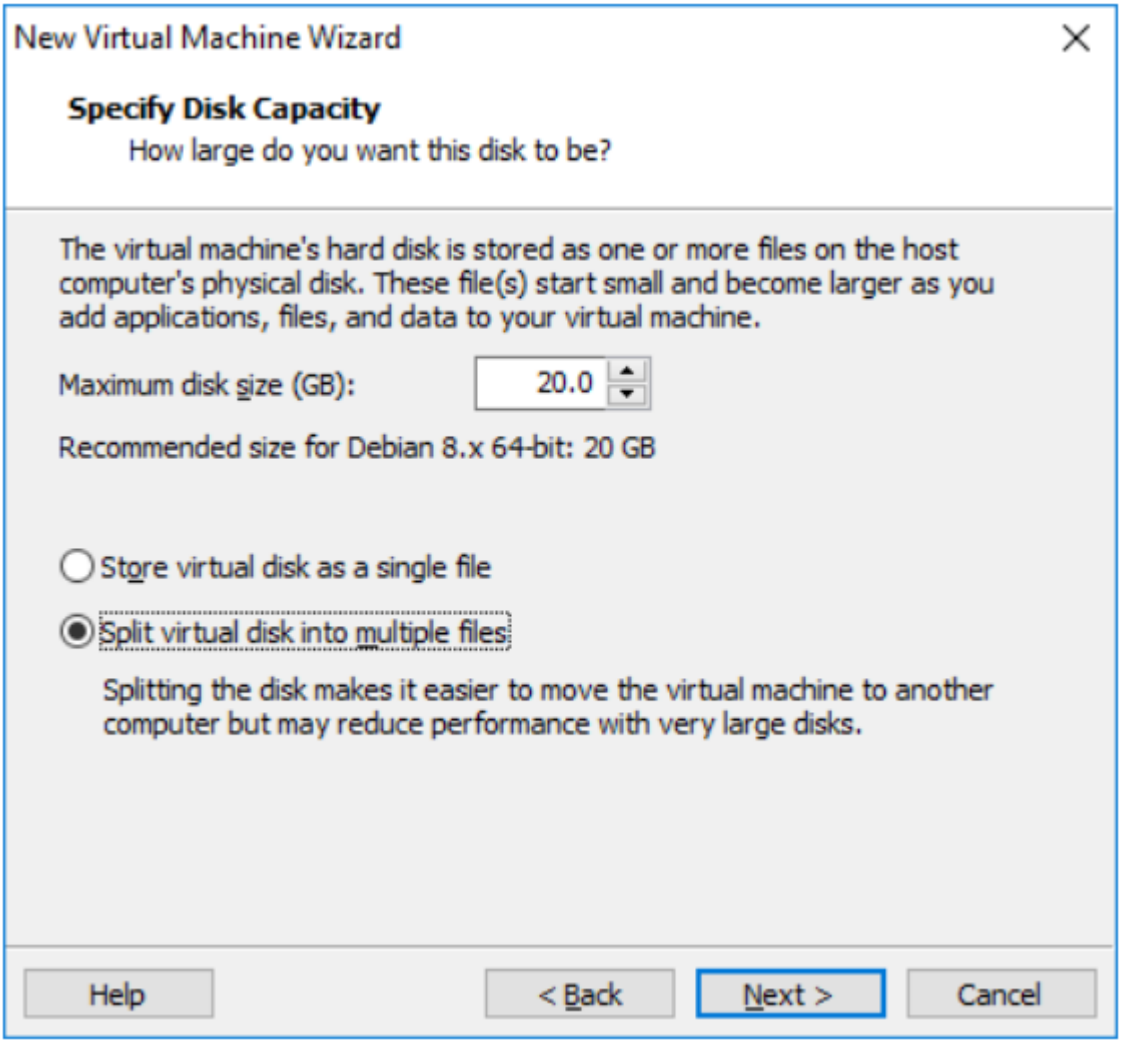

Рисунок 2.23 *Укажите емкость диска*

Размер жесткого диска по умолчанию указан 20 ГБ (рисунок 2.23, «Укажите емкость диска» [стр. 40]) и этого обычно достаточно, но вы можете настроить его здесь в зависимости от ожидаемых потребностей. В отличие от VirtualBox, который может использовать только один файл различного размера, VMware имеет возможность хранить содержимое диска в нескольких файлах. В обоих случаях целью является сохранение дискового пространства хоста.

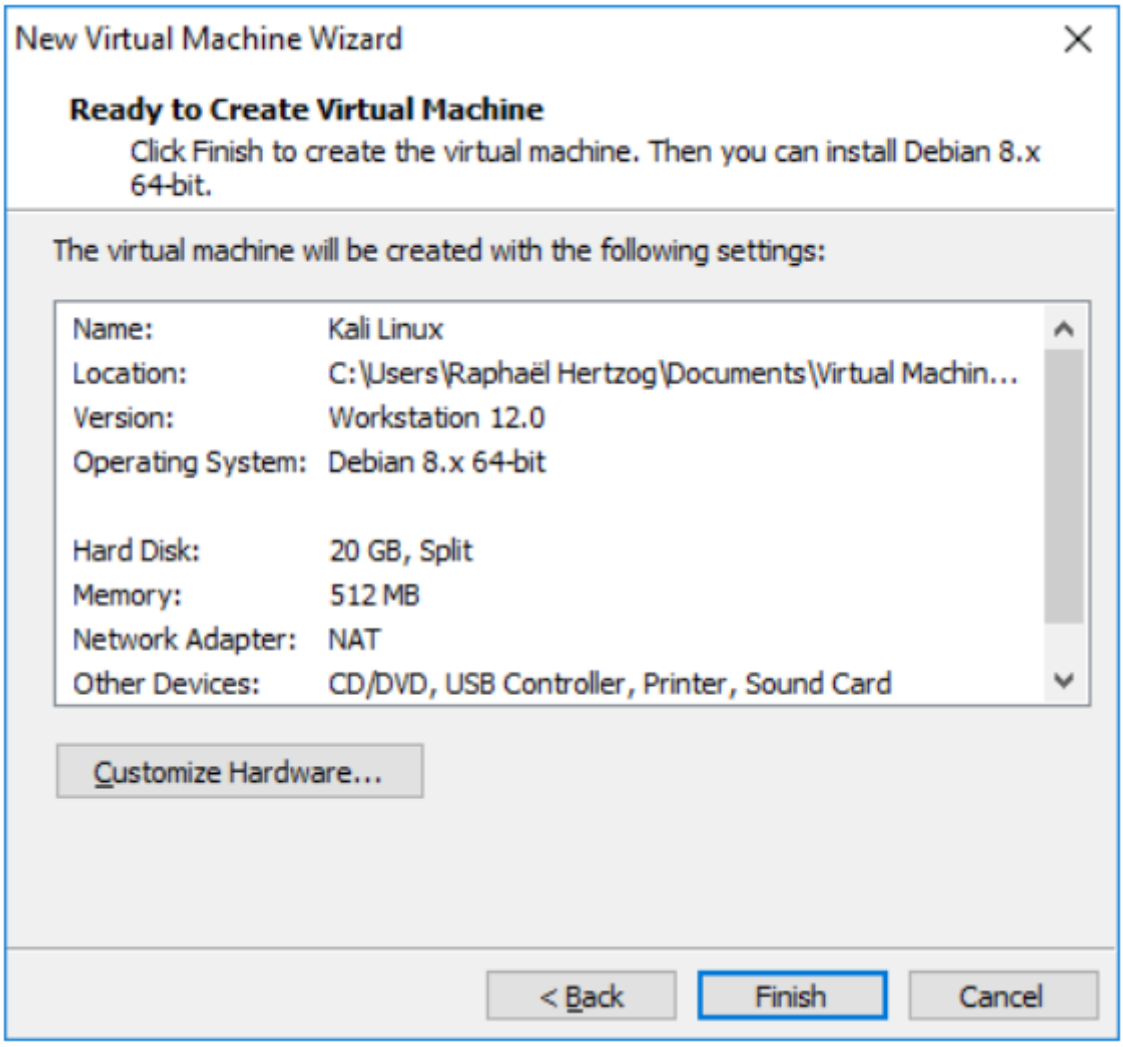

Рисунок 2.24 *Готов к созданию виртуальной машины*

VMware Workstation теперь настроена для создания новой виртуальной машины. Она отображает список всех решений, которые вы приняли в процессе настройки так, что вы можете проверить дважды ваши настройки перед созданием виртуальной машины. Обратите внимание, что мастер решил разместить на виртуальной машине только 512 МБ ОЗУ, чего недостаточно, поэтому нажмите «Настроить оборудование ...» (рисунок 2.24 «Готово к созданию виртуальной машины» [стр. 41]) и выберите настройку «Память», как показано на рисунке 2.25, «Настройка аппаратного окна» [стр. 42].

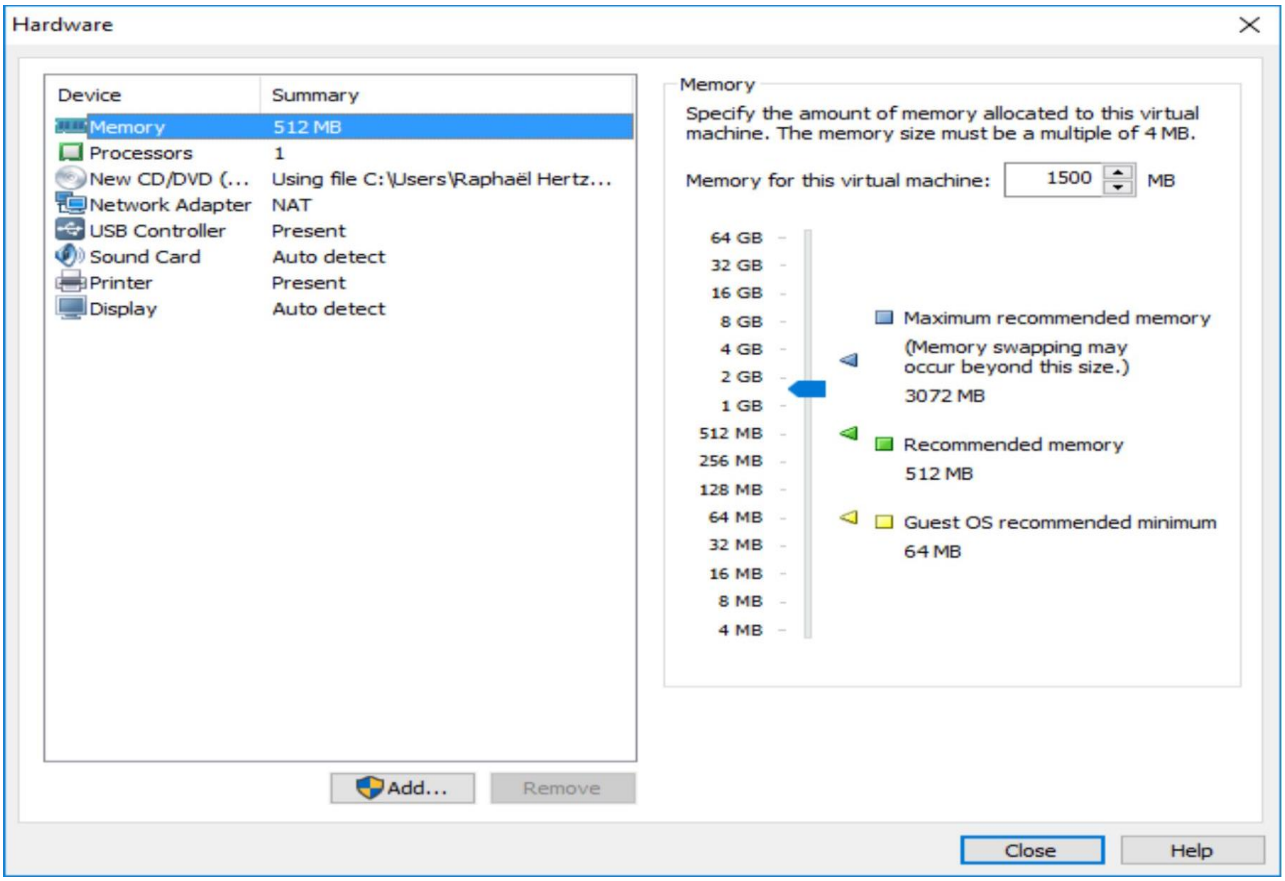

Figure 2.25 *Настройка аппаратного окна*

После последнего нажатия кнопки «Готово» (Finish) (рисунок 2.24 «Готов к созданию виртуальной машины»), виртуальная машина будет настроена и может быть запущена нажатием кнопки «Включить эту виртуальную машину» (’Power on this virtual machine), как показано на рисунке 2.26, «Готовая виртуальная машина для Kali Linux».

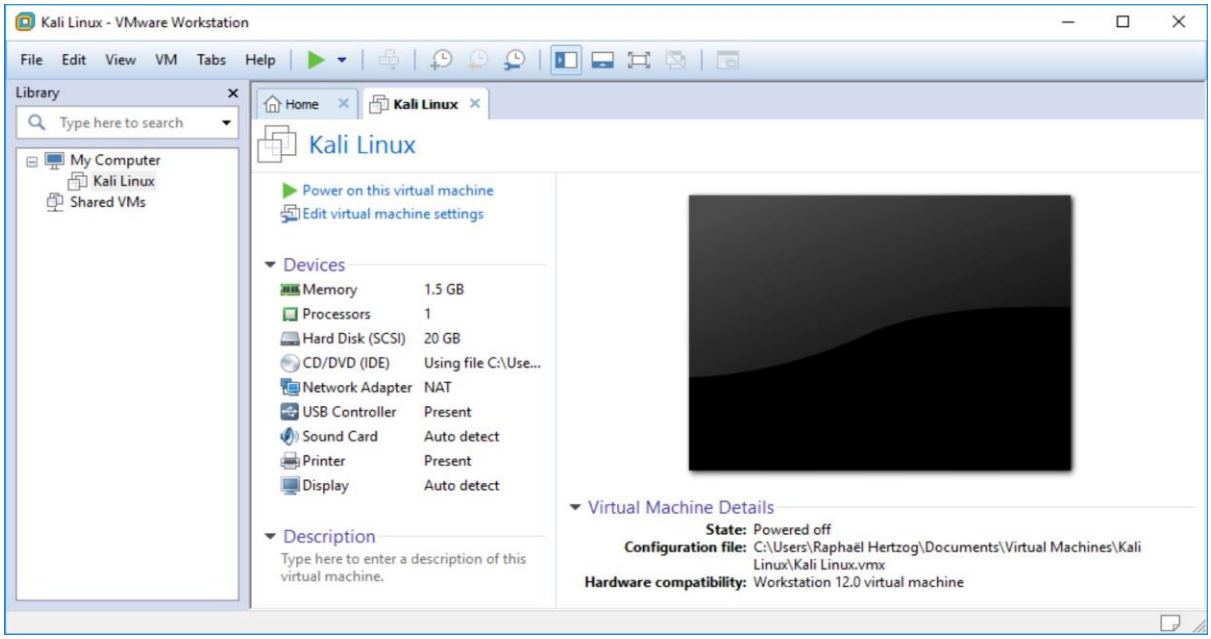

Figure 2.26 *Готовая виртуальная машина для Kali Linux*

2.3 Подведем итоги

В этой главе вы узнали о различных ISO образах Kali Linux, также вы выяснили, как проверять и скачивать их, и получили пошаговую инструкцию о том, как создавать загрузочный USB накопитель, используя данные образы, на разных операционных системах. Мы также обсудили, как загрузить USB накопитель и узнали, как настроить BIOS и выставить стартовые настройки на различных аппаратных платформах, таким образом, чтобы загрузка шла непосредственно с USB накопителя.

Главное из раздела:

www.kali.org является единственным официальным сайтом для загрузки ISO образа Kali. Не стоит скачивать образы с других ресурсов, т.к. эти файлы могут содержать вредоносное программное обеспечение.

Всегда проверяйте sha256sum своих загрузок командой sha256sum, чтобы убедиться в подлинности скачанного вами ISO файла. Если он не соответствует, попробуйте использовать для скачивания другой источник.

Вы должны записывать Kali Linux ISO образ на загружаемый

накопитель, если вы хотите использовать его на реальной машине. Используйте *Win32 Disk Imager* на Windows, утилиту Disks на Linux, или dd команду на Mac OS X/macOS. Будьте очень осторожны, записывая образ. Если вы выберите не тот диск для записи, вы сможете безвозвратно повредить данные, хранящиеся на вашей машине.

Выставьте необходимые настройки в экранах BIOS/UEFI на ПК или удерживайте клавишу Option на OS X/macOS, чтобы позволить машине загружаться с USB накопителя.

Программы для создания виртуальной машины *VirtualBox* и *VMware Workstation Pro* являются особенно полезными для людей, которые хотят опробовать Kali Linux, но не готовы установить её на свою машину или же для тех, у кого есть довольно мощная система и они хотят использовать несколько операционных систем одновременно.

Теперь, когда у вас есть рабочая, установленная версия Kali Linux, пришло самое время углубиться в изучение фундаментальных основ Linux, которые необходимы для базового и продвинутого использования Kali. Если вы являетесь пользователем Linux с умеренным или расширенным уровнем знаний, вы можете лишь бегло ознакомиться со следующей главой

Часть 3: Основы Linux

Содержание:

Ключевые слова главы:

- Ядро Linux;
- Пользовательское пространство;
- Командная строка;
- Bash;
- Иерархия файловой системы;
- Unix команды;

Прежде чем вы сможете освоить Kali Linux, вы должны свободно ориентироваться в основных понятиях системы Linux. Умение работать с Linux будет очень полезным навыком, потому что на серверах Linux работает большое количество веб-сайтов, электронной почты и других интернет-служб.

В данном разделе мы собираемся разъяснить основные понятия связанные с Linux. Выполняя поставленную перед нами задачу мы полагаем, что вы уже имеете представление о компьютерных системах в целом, включая такие компоненты как центральный процессор (ЦП), оперативная память (ОЗУ), материнская плата, жесткий диск, а также другие контроллеры и устройства, связанные с ними.

3.1 Что такое Linux и что он делает ?

Термин «Linux» часто используется для обозначения всей операционной системы, но на самом деле Linux - это ядро операционной системы, которое запускается начальным загрузчиком, запускаемым BIOS / UEFI. Ядро берет на себя роль, похожую на роль дирижера в оркестре, оно обеспечивает согласованную работу аппаратных средств и программного обеспечения. Данная роль подразумевает под собой управление оборудованием, процессами, пользователями и файловыми системами. Ядро представляет собой общую базу для других программ, работающих в данной системе, и чаще всего запускает *ring zero,* также известное, как *пространство ядра (kernel space).*

Пользовательское пространство

Мы используем термин «пользовательское пространство», чтобы объединить все, что происходит за пределами ядра.
Среди программ, работающих в пространстве пользователя, много основных утилит из проекта GNU, большинство из которых предназначено для запуска из командной строки. Вы можете использовать их в сценариях для автоматизации различных задач. Дополнительную информацию о наиболее важных командах см. в разделе 3.4 «Полезные команды» .

Давайте быстро рассмотрим различные задачи, выполняемые ядром Linux.

3.1.1 Запуск оборудования

Назначением ядра, прежде всего, является управление и контроль над основными компонентами компьютера. Оно обнаруживает и настраивает их, когда компьютер включается, а также когда устройство монтируется или извлекается (например, USB устройство). Это также делает их доступными для более высокоуровневого программного обеспечения благодаря упрощенному программному интерфейсу, поэтому приложения могут использовать преимущества устройств, не обращаясь к деталям, например к слоту расширения, в который вставлена плата. Программный интерфейс также предоставляет определенный уровень абстракции; это позволяет использовать оборудование для проведения видеоконференций, например, использовать вебкамеру независимо от её модели и производителя. Программное обеспечение может использовать интерфейс *Video for Linux* (V4L) и ядро будет переводить вызовы интерфейса в реальные аппаратные команды, необходимые для работы конкретной веб-камеры.

Ядро экспортирует данные об обнаруженном оборудовании через виртуальные системы /proc/ и /sys/. Приложения часто получают доступ к устройствам с помощью файлов, созданных в /dev/. Особые файлы, представляющие диски (например, /dev/sda), разделы (/dev/sdal), мыши (/dev/input/mouse0), клавиатуры (/dev/input/event0), звуковые карты (/dev/snd/*), последовательные порты (/dev/ttyS*) и другие компоненты.

Существует два типа файлов устройств: блочные и символьные. Первые имеют характеристики блока данных: они имеют конечный размер, и вы можете получить доступ к байтам в любой позиции блока. Последние ведут себя как поток символов. Вы можете читать и писать символы, но вы не можете искать заданную позицию и изменять произвольные байты. Чтобы узнать тип файла устройства, проверьте первую букву вывода команды

Is -1. Это может быть либо b, для блочных устройств, либо c, для символьных устройств:

```
$ ls -l /dev/sda /dev/ttyS0
brw-rw---- 1 root disk    8,  0 Mar 21 08:44 /dev/sda
crw-rw---- 1 root dialout 4, 64 Mar 30 08:59 /dev/ttyS0
```

Как вы уже возможно догадались, диски и разделы используют блочные файлы устройств, в то время как мышь, клавиатура и последовательные порты используют символьные файлы устройств. В обоих случаях программный интерфейс включает в себя специальные команды, которые могут быть активированы через системный вызов *ioctl*.

3.1.2 Объединение файловых систем

Файловые системы являются важным аспектом ядра. Системы, основанные на Unix, объединяют все хранилища файлов в одну иерархию, что позволяет пользователям и приложениям получать доступ к данным, зная их местоположение в пределах этой иерархии.

Отправная точка этого иерархического дерева называется root, представленный символом “/”. Данная директория может содержать именованные суб-директории. Например, домашняя суб-директория «/» называется /home/. Эта суб-директория, в свою очередь, может содержать другие суб-директории и т.д. Каждая директория также может содержать файлы, в которых будут храниться файлы. Таким образом, home/buxy/Desktop/hello.txt относится к файлу под названием hello.txt, который хранится в суб-директории Desktop, находящейся в buxy суб-директории домашнего каталога, который присутствует в root. Ядро компилирует между данной системой именования и местом хранения на диске.

В отличие от других систем, Linux обладает только одной такой иерархией и может интегрировать данные с нескольких дисков. Один из таких дисков становится root, а другие *монтируются* на директории в иерархии (эта команда в Linux называется mount).

Эти другие диски затем становятся доступными под точками монтирования (*mount points*) Это позволяет хранить пользовательские домашние директории (которые обычно хранятся на /home/) на отдельном жестком диске, который будет содержать директорию buxy (вместе с домашними директориями других пользователей). После того, как вы установили диск в /home/, эти каталоги становятся доступными в их обычном месте, а различные пути, такие как /home/buxy/Desktop/hello.txt, продолжают работать.

Существует множество форматов файловой системы в соответствии с множеством способов физического хранения данных на дисках. Наиболее широко известны ext2, ext3 и ext4, но существуют и другие. Например, *VFAT* является файловой системой, которая исторически использовалась DOS и операционными системами Windows. Поддержка VFAT операционной системой Linux позволяет жестким дискам быть доступными как под Kali, так и под Windows. В любом случае, вы должны подготовить файловую систему на диске, прежде чем смонтировать ее, и эта операция называется *форматированием*.

Команды, такие как mkfs.ext3 (где mkfs расшифровывается как *MaKe FileSystem)* обрабатывает форматирование. В качестве параметра эти команды требуют файл устройства, представляющий раздел, который следует отформатировать (например, /dev/sdal, первый раздел на первом диске). Эта операция уничтожает все данные и должна запускаться только один раз, если конечно вы не хотите стереть файловую систему и начать новую работу.

Есть также сетевые файловые системы, такие как NFS, которые не хранят данные на локальном диске. Вместо этого данные передаются через сеть на сервер, который хранит их и выдает по первому требованию. Благодаря абстракции файловой системы вам не нужно беспокоиться о том, как этот диск подключен, так как файлы остаются доступными по своему обычному иерархическому пути.

3.1.3 Управление процессами

Процесс является исполняемым экземпляром программы, для которой требуется хранить память, как самой программы, так и ее рабочих данных. Ядро отвечает за создание и отслеживание процессов. Когда программа запускается, ядро сначала выделяет некоторую память, загружает исполняемый код из файловой системы в эту память, а затем запускает код. Он содержит информацию об этом процессе, наиболее заметным из которых является идентификационный номер, известный как *идентификатор процесса* (*process identifier* (PID)).

Большинство современных операционных систем, а именно те, которые работают на основе Unix ядра, включая Linux, способны выполнять множество задач. Другими словами, они позволяют системе запускать множество процессов одновременно. На самом деле существует только один запущенный процесс в любой момент времени, но ядро делит время процессора на небольшие фрагменты и запускает каждый процесс по очереди. Поскольку эти временные срезы очень короткие (в миллисекундах), они создают внешний вид процессов, работающих параллельно, хотя они активны только в течение их временного интервала и бездействуют в остальное время. Основной задачей ядра является настройка механизмов планирования таким образом, чтобы сохранить этот внешний вид, одновременно увеличивая производительность системы. Если отрезок времени будет слишком длинным, может перестать отвечать должным образом. Ну а если же они будут слишком короткими, система будет терять слишком много времени на переключение между ними. Подобные решения можно регулировать с помощью приоритетов процессов, когда высокоприоритетные процессы будут выполняться в течение более длительных периодов времени и с более частыми временными срезами, чем процессы с низким приоритетом.

Мультипроцессорные Системы (и другие варианты)

Ограничения, описанные выше, о том, что одновременно может работать только один процесс, применимы не во всех ситуациях. Более верно будет сказать, что *одно ядро* может работать только с одним процессом. Многопроцессорные, многоядерные или

гиперпотоковые системы позволяют нескольким процессам работать параллельно. Тем не менее, используется одна и та же система сокращения времени для обработки ситуаций, когда есть более активные процессы, чем доступные процессорные ядра. Это не является чем-то необычным: базовая система, даже полностью бездействующая, почти всегда имеет десятки запущенных процессов.

Ядро позволяет запускать несколько независимых экземпляров одной и той же программы, но каждому разрешается доступ только к собственным временным срезам и памяти. Таким образом, их данные остаются независимыми.

3.1.4 Управление правами

Unix системы поддерживают множество пользователей и групп и позволяют контролировать права доступа. В большинстве случаев, процесс определяется пользователем, который запускает его. Данный процесс может выполнять только те действия, которые разрешены его владельцу. Например, открытие файла требует от ядра проверить процесс на наличие необходимых прав (для получения большей информации конкретно по этому примеру, см. раздел 3.4.4, “Управление правами”)

3.2 Командная строка

Под «командной строкой» мы подразумеваем текстовый интерфейс, который позволяет вводить команды, выполнять их и просматривать результаты. Вы можете запустить терминал (текстовый экран внутри графического рабочего стола или текстовую консоль вне любого графического интерфейса) и интерпретатор команд внутри него (*оболочка*).

3.2.1 Как запустить командную строку

Когда ваша система работает правильно, самым простым способом получения доступа к командной строке является запуск терминала в графическом сеансе рабочего стола.

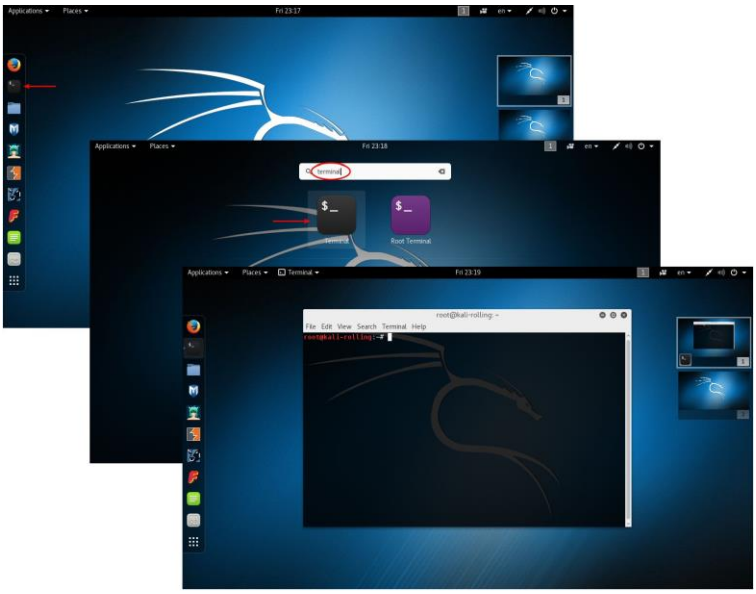

Рисунок 3.1 *Запуск терминала GNOME*

Например, в системе Kali Linux по умолчанию, GNOME терминал может быть запущен из списка избранных приложений. Также вы можете ввести «terminal» в окне Activities (окно, которое активируется, когда вы передвигаете мышь в левый верхний угол) и нажмите на необходимой вам иконке приложения, которые появятся (Рисунок 3.1, “Запуск терминала GNOME”).

В случае каких-либо нарушений или некорректной работы вашего графического интерфейса вы все равно можете запустить командную строку на виртуальных консолях (до шести из них могут быть доступны через шесть комбинаций клавиш, начиная с CTRL + ALT + F1 и заканчивая CTRL + ALT + F6 - клавишу CTRL можно не нажимать, если вы уже находитесь в текстовом режиме вне графического интерфейса Xorg или Wayland).

Вы получаете обычный экран входа, где вы вводите свой логин и пароль, перед тем как получить доступ к командной строке с её оболочкой:

```
Kali GNU/Linux Rolling kali-rolling tty3
kali-rolling login: root
Password:
Last login: Fir Mar 25 12:30:05 EDT 2016 from 192.168.122.1 on pts/2
Linux kali-rolling 4.4.0-kali1-amd4 #1 SMP Debian 4.4.6-1kali1 (2016-03-18) x86_64

The programs included with the Kali GNU/Linux system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Kali GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent
permitted by applicable law.
root@kali-rolling:~#
```

Программа, обрабатывающая введенные вами данные и выполнение ваших команд, называется *оболочкой* (*shell* или интерпретатором командной строки). По умолчанию оболочкой, предоставляемой в Kali Linux, является *Bash* (это означает *Bourne Again SHell*). Конечный символ «$» или «#» указывает, что оболочка ожидает вашего ввода. Эти символы также указывают на то, каким образом воспринимает вас Bash, как обычного пользователя (первый случай со значком долларом) или как суперпользователя (последний случай с хэшем).

3.2.2 Основы командной строки: просмотр дерева директорий и управление файлами

Данный раздел предоставляет лишь краткий обзор некоторых команд, каждая из которых имеет множество различных опций и возможностей, не описанных здесь, поэтому, пожалуйста, обратитесь к обширной документации, доступной в соответствующих страницах руководства. В тестированиях на проникновение, чаще всего вы будете получать доступ к системе через оболочку, после успешного эксплуатирования, а не через пользовательский графический интерфейс. Умение грамотно пользоваться командной строкой является необходимым для вас, если вы хотите достичь успеха как специалист в сфере безопасности.

Как только сеанс запущен, команда pwd (которая расшифровывается как *print working directory (отобразить рабочий каталог)*) выведет на экран ваше текущее

местоположение в файловой системе. Ваше текущее местоположение можно изменить с помощью команды cd *название директории* (где cd означает (сменить директорию)). В том случае, если вы не указали директорию, куда хотите перейти, вы автоматически вернетесь в вашу домашнюю директорию. Если вы введете cd -, то вы вернетесь в предыдущую рабочую директорию (в ту, в которой вы находились перед вводом последней команды cd). Родительский каталог всегда называется .. (две точки), в то время как текущий каталог обозначается . (одной точкой). Команда ls позволяет вам *перечислить* содержимое директории. Если вы не указываете дополнительных параметров команда ls, отобразит содержимое текущей директории.

```
$ pwd
/home/buxy
$ cd Desktop
$ pwd
/home/buxy/Desktop
$ cd .
$ pwd
/home/buxy/Desktop
$ cd ..
$ pwd
/home/buxy
$ ls
Desktop    Downloads  Pictures  Templates
Documents  Music      Public    Videos
```

Вы можете создать новую директорию с помощью команды mkdir *название директории,* а также удалить существующую (пустую) директорию с помощью команды rmdir *название директории.* Команда mv позволит вам *перемещать* и переименовывать файлы и директории; *удалить* файл можно с помощью rm *название файла,* а копирование файла выполняется с помощью cp *исходный-файл целевой-файл.*

```
$ mkdir test
$ ls
Desktop    Downloads  Pictures  Templates  Videos
Documents  Music      Public    test
$ mv test new
$ ls
Desktop    Downloads  new       Public     Videos
Documents  Music      Pictures  Templates
$ rmdir new
$ ls
Desktop    Downloads  Pictures  Templates  Videos
Documents  Music      Public
```

Оболочка выполняет каждую команду, запуская первую программу с данным именем, которую она находит в каталоге, указанном в переменной среде PATH. Чаще всего эти программы находятся в /bin, /sbin, /usr/bin или /usr/sbin. Например, команда ls находится в /bin/ls; Иногда команда напрямую обрабатывается оболочкой, и в этом случае она называется встроенной командой оболочки (среди них - cd и pwd); команда type позволяет запросить тип каждой команды.

```
$ echo $PATH
/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin
$ which ls
/bin/ls
```

```
$ type rm
rm is /bin/rm
$ type cd
cd is a shell builtin
```

Обратите внимание на использование команды echo, которая просто отображает строку в терминале. В данном случае, она используется для вывода на экран содержимого переменной среды, т.к. оболочка автоматически заменяет переменные с их значениями пере выполнением командной строки.

Переменные среды

Переменные среды позволяют хранить глобальные настройки для оболочки или других программ. Они являются контекстуальными, но наследуемыми. Например, каждый процесс имеет свой собственный набор переменных среды (они являются

контекстуальными). Оболочки, такие как оболочки входа, могут объявлять переменные, которые будут переданы другим исполняемым программам (они наследуются).
Эти переменные могут быть определены как для системы в/etc/profile так и для пользователя в~/ .profile, но переменные, которые не являются характерными для интерпретаторов командной строки, лучше вставлять в /etc/environment, поскольку эти переменные будут введены во все пользовательские сессии благодаря подключаемому модулю аутентификации (Pluggable Authentication Module (PAM)) - даже если никакая оболочка не выполняется.

3.3 Файловая система

3.3.1 Стандарт иерархии файловой системы

Как и другие дистрибутивы Linux, Kali Linux организован в соответствии со стандартом *Filesystem Hierarchy Standard* (FHS), что позволяет пользователям других дистрибутивов Linux с легкостью ориентироваться в Kali. FHS определяет назначение каждой директории. Директории верхнего уровня описываются следующим образом.

/bin/	основные программы
/boot/	Ядро Kali Linux и другие файлы, необходимые для его раннего процесса загрузки
/dev/	файлы устройства
/etc/	файлы конфигурации
/home/	личные файлы пользователей
/lib/	основные библиотеки

/media/	точки монтирования для съемных устройств (CD-ROM, USB накопители и т.д.)
/mnt/	временные точки монтирования
/opt/	дополнительные приложения, предоставляемые третьими лицами
/root/	личные файлы администратора (файлы root)
/run/	непостоянные файлы рабочего процесса, которые не сохраняются после перезагрузки (еще не включённые в FHS)
/sbin/	системны программы
/srv/	данные, используемые серверами, расположенными в этой системе
/tmp/	временные файлы (эта директория часто опустошается после перезагрузки)
/usr/	приложения (эта директория в дальнейшем разделяется на bin, sbin, lib согласно такой же логике, что и в директории root). Кроме того, /usr/share/ содержат данные с независимой архитектурой. Каталог / usr / local / предназначен для использования администратором для установки приложений вручную без перезаписи файлов, обрабатываемых системой пакетирования
/var/:	переменные данные, обрабатываемые демоном. Это включает в себя файлы журналов, очереди, буферы и кеши.
/proc/ и /sys/	являются характерными для ядра Linux (и не являются частью FHS). Они используются ядром для экспортирования данных в пользовательское пространство.

3.3.2 Домашняя директория пользователя

Содержимое пользовательской директории не является стандартизированным, но, тем не менее, существует несколько заслуживающих внимания условностей. Одна из них заключается в том, что пользовательский домашний каталог часто обозначается тильдой (“~”). Это очень полезно знать, потому что интерпретаторы команд автоматически заменяют тильду верной директорией (которая находится в переменной среде HOME и чье обычное значение является /home/user/).

Традиционно файлы конфигурации приложения часто хранятся непосредственно в вашем домашнем каталоге, но их имена файлов обычно начинаются с точки (например, клиент электронной почты mutt хранит конфигурацию в ~/.muttrc). Обратите внимание, что имена файлов, начинающиеся с точки, по умолчанию скрыты; команда ls перечислит их лишь, в том случае если указана опция –a, а графические файловые менеджеры должны быть явно настроены для отображения скрытых файлов.

Некоторые программы также используют несколько файлов конфигурации, организованных в одном каталоге (например, ~/.ssh/). Некоторые приложения (например, веб-браузер Firefox) также используют свой каталог для хранения кеша загруженных данных. Это означает, что эти каталоги могут в конечном итоге потреблять много дискового пространства.

Эти файлы конфигурации, которые хранятся прямо в вашей домашней директории, часто коллективно называющиеся *dotfiles,* долгое время расширяются до такой степени, что эти директории могут быть загромождены ими. К счастью, совместная работа под эгидой FreeDesktop.org привела к созданию спецификации базового каталога XDG (XDG Base Directory Specification) конвенции, целью которой является очистка этих файлов и каталогов. В этой спецификации указано, что файлы конфигурации должны храниться в~/.config, файлы кэша в -/.cache, а файлы данных приложения в -/.local (или вих суб-директориях). Эта конвенция постепенно набирает обороты.

Графический рабочий стол чаще всего использует ярлыки для отображения содержимого каталога /Desktop/ (или любого другого слова, которое является точным переводом данного, в системах, которые не используют английский язык). Наконец,

система электронной почты иногда хранит входящие письма в каталоге - /Mail/.

3.4 Полезные команды

3.4.1 Отображение и изменение текстовых файлов

Команда cat *название файла* (предназначена для соединения файлов для стандартного устройства вывода) читает файл и отображает его содержимое в терминале. Если файл является слишком большим, чтобы быть выведенным на экран, вы можете использовать пейджер для того, чтобы отображать его по страницам.

Команды редактирования запускают текстовый редактор (такой как Vi или Nano), который позволяет создавать, редактировать и читать текстовые файлы. Самые простые файлы могут быть иногда созданы прямо из интерпретатора команд благодаря перенаправлениям: команда *command >file* создаст файл с именем *file,* который будет содержать вывод данной команды. Команда *command >>file* сделает практически то же самое кроме того, что она присоединяет вывод команды вместо того, чтобы перезаписывать его.

```
$ echo "Kali rules!" > kali-rules.txt
$ cat kali-rules.txt
Kali rules!
$ echo "Kali is the best!" >> kali-rules.txt
$ cat kali-rules.txt
Kali rules!
Kali is the best!
```

3.4.2 Поиск файлов и данных внутри файлов

Команда find *критерий директории* ищет файлы в иерархии под *директорией* в соответствии с несколькими критериями. Самый часто используемый критерий это -name *имя файла,* который позволяет искать файл по его имени. Вы также можете использовать общие подстановочные знаки, такие как «*» в поиске имени файла.

```
$ find /etc -name hosts
/etc/hosts
/etc/avahi/hosts
$ find /etc -name "hosts*"
/etc/hosts
/etc/hosts.allow
/etc/hosts.deny
/etc/avahi/hosts
```

Команда grep *выражение файлов* выполняет поиск содержимого файлов и извлекает строки, соответствующие регулярному выражению. Добавление параметра -r позволяет рекурсивный поиск по всем файлам, находящимся в каталоге. Это позволяет вам искать файл, когда вы знаете только часть его содержимого.

3.4.3 Управление процессами

Команда ps aux перечисляет процессы, которые в данный момент запущены в системе и помогает идентифицировать их, показывая их PID. Как только вы узнаете *PID* процесса, команда kill *-signal pid* позволяет вам отправить сигнал (если вы являетесь владельцем процесса). Существует несколько сигналов; самыми часто используемыми являются TERM (запрос на прекращение процесса) и KILL (принудительное завершение).

Интерпретатор команд может также запускать программы в фоновом режиме, если за командой следует «&». Используя амперсанд, вы немедленно возобновляете управление оболочкой, даже если команда все еще работает (находится в срытом режиме в качестве фонового процесса). Команда jobs перечисляет процессы, запущенные в фоновом режиме; если запустить fg *%job-number* (fg означает *приоритетный (foreground)),* то задача будет восприниматься как приоритетная. Когда команда запускается как приоритетная (она могла быть запущена как в нормальном режиме, так и возвращена к приоритетным с помощью команды fg), комбинация клавиш Control+Z приостанавливает процесс и возвращает контроль над командной строкой. Процесс затем может быть перезапущен в фоновом режиме с помощью команда bg *% job-number* (bg означает фоновый (*background)).*

3.4.4 Управление правами

Linux является многопользовательской системой, так что в ней необходимо предоставить систему разрешений для управления набором разрешенных операций над файлами и каталогами, которые включают в себя все системные ресурсы и устройства (в системе Unix любое устройство представлено файлом или каталогом). Данный принцип является общим для всех Unix подобных систем.

Каждый файл или директория обладает особыми правами доступа для трех категорий пользователей:

- Его владелец (обозначается буквой u, как в слове user);
- Группа, владеющая им (обозначается буквой g, как в слове group), представляет всех членов группы;
- Другие (обозначается буквой o, как в слове other)
- Три типа прав могут быть объединены:
- Для чтения (обозначается буквой r, как в слове read);
- Для записывания (или редактирования, обозначается буквой w, как в слове write);
- Для выполнения (обозначается буквой x, как в слове eXecute).

В случае с файлом действие этих прав очень понято: доступ для чтения позволяет читать содержимое файла (включая копирование), доступ для записи позволяет его изменить, а доступ для выполнения позволяет запустить его (что будет работать только в том случае, если файл является программой).

Исполняемые файлы setuid и setgid

Два конкретных права относятся к исполняемым файлам: setuid и setgid (обозначаются буквой «s»). Обратите внимание, что мы часто говорим о битах, так как каждое из этих логических значений может быть представлено нулем или единицей. Эти два права позволяют любому пользователю выполнять программу с правами владельца или группы, соответственно. Этот механизм предоставляет доступ к функциям, требующим более высоких разрешений, чем те, которые вы обычно имели.

Поскольку корневая программа setuid систематически запускается под идентификатором суперпользователя, очень важно обеспечить ее надежность и безопасность. Любому пользователю, который сможет нарушить работу корневой программы setuid для вызова команды по своему выбору, может затем выступать в роле пользователя root и иметь все необходимые права в системе. Пентестеры регулярно ищут файлы такого типа, когда они получают доступ к системе и используют его для того, чтобы расширить свои права доступа.

Директории обрабатываются иначе, чем файлы. Доступ к чтению дает право ознакомиться со списком её содержимого (файлов и каталогов); доступ для записи позволяет создавать или удалять файлы; и доступ для выполнения позволяет переходить через директорию для получения доступа к его содержимому (например, с помощью команды cd). Возможность переходить через директорию, не имея возможности читать её, дает пользователю право доступа к записям в директориях, которые известны по имени, но не для их поиска, не зная их точного имени.

Безопасность

Директория setgid и *sticky bit*

setgid bit также применяется к директориям. Любой заново созданный объект в подобных директориях автоматически назначает группу владельца родительского каталога, а не наследует основную группу создателя, как обычно. Из-за этого вам не нужно менять основную группу (с помощью команды newgrp) при работе в дереве файлов, совместно используемом несколькими пользователями одной и той же выделенной группы.

The *sticky bit* (обозначается буквой “t”) является разрешением, которое довольно полезно в директориях. Оно особенно полезно для использования во временных директориях, где у всех есть доступ на запись (например, /tmp/): оно ограничивает удаление файлов таким образом, что только их владелец или владелец родительского каталога может их удалить. В противном случае все пользователи могли бы удалять файлы других пользователей в /tm /.

Три команды управляют разрешениями, связанными с файлом:

- chown *пользовательский файл* меняет владельца файла

> **СОВЕТ Изменение пользователя и группы**
>
> Довольно часто вы хотите изменить группу файла одновременно с изменением его владельца. Команда chown обладает особым синтаксисом для подобных задач: chown *user: group file*

- chgrp *файл группы* изменяет владельца группы
- chmod *файл прав* изменяет права доступа к файлу

Существует два способа обозначения прав. Среди них, символическое обозначение, пожалуй, является самым простым для понимания и запоминания. Оно включает в себя буквы уже упомянутые выше. Вы можете определять права для каждой из категорий пользователей (u/g/o), с помощью использования (знака =) прибавления (+) или вычитания (-). Таким образом, формула u=rwx,g+rw,o-r дает владельцу права на чтение, запись и выполнение, предоставляет владельцам групп права на чтение и запись, а также лишает прав на чтение других пользователей. Права, в которых не были внесены изменения добавлением или вычитанием с помощью подобной команды такой команде, остаются неизмененными. Буква *a* для всех охватывает все три категории пользователей, так что *a = rx* предоставляет всем трем категориям одинаковые права (чтение и выполнение, но не запись).

Восьмеричное или числовое обозначение связывает каждое право с определенной величиной: 4 для чтения, 2 для записи и 1 для выполнения. Мы связываем каждую комбинацию прав с суммой трех цифр, следовательно, определенное значение присваивается каждой категории пользователей в обычном порядке (владелец, группа, другие).

Например, команда chmod 754 *название файла* установит следующие права: чтение, запись и выполнение для владельца

(т.к. 7 = 4 + 2 + 1); чтение и выполнения для группы (т.к. 5 = 4 + 1); права только на чтение для других. Цифра 0 означает, что категория не обладает никакими правами; таким образом, chmod 754 *название файла* дает права на чтение и запись владельцу и никому больше. Самой распространенной комбинацией прав является 755 для исполняемых файлов и директорий и 644 для файлов данных.

Чтобы обозначить специальные права, вы можете приписать четвертую цифру этому номеру в соответствии с тем же принципом, где биты setuid, setgid и sticky равны 4, 2 и 1 соответственно. Команда chmod 4754 свяжет бит setuid с ранее описанными правами.

Обратите внимание, что использование восьмеричной записи позволяет вам сразу устанавливать все права на файл; вы не можете использовать его для добавления нового права, такого как доступ для чтения для владельца группы, поскольку вы должны учитывать существующие права и вычислять новое соответствующее числовое значение.

Восьмеричное обозначение также используется с командой umask, которая используется для ограничения прав на недавно созданные файлы. Когда приложение создает файл, оно назначает индикативные права доступа, зная, что система автоматически удаляет права, определенные с помощью umask. Введите umask в оболочке; вы увидите следующую маску 0022. Это просто восьмеричное обозначение прав на систематическое удаление (в этом случае права на запись для группы и других пользователей).

Если вы дадите ему новое восьмеричное значение, команда umask изменит маску. Используемый в файле начальной инициализации оболочки (например, ~/.bash_profile), он эффективно изменяет маску по умолчанию для ваших рабочих сессий.

СОВЕТ Рекурсивная операция

Иногда нам приходится менять права для всего дерева файлов. Все вышеприведенные команды имеют опцию -R для рекурсивной работы в суб-директориях.

Различие между каталогами и файлами иногда вызывает проблемы с повторными операциями. Вот почему буква «X» была введена в символическом обозначении прав. Она представляет собой право на выполнение, которое применяется только к каталогам (а не к файлам, не имеющим этого права). Таким образом, команда chmod -R a+X *название директории* будет добавлять только права на выполнение для всех категорий пользователей (a) для всех суб-директорий и файлов, в которых хотя бы одна категория пользователей (даже если они являются единоличными владельцами) уже обладает правами на выполнение.

3.4.5 Получение системной информации и журналов

Команда free отображает информацию о памяти; *disk free* (df) сообщает вам о свободном пространстве на каждом диске, который смонтирован в файловой системе. Опция данной команды -h (*читаемая для человека*) преобразует размеры в более разборчивую единицу (обычно mebibytes или gibibytes). Подобным образом команда free поддерживает -m и -g опции и отображает их данные как в mebibytes, так и в gibibytes соответственно.

```
$ free
              total        used        free      shared  buff/cache   available
Mem:        2052944      661232      621208       10520      770504     1359916
Swap:             0           0           0
$ df
Filesystem     1K-blocks     Used Available Use% Mounted on
udev             1014584        0   1014584   0% /dev
tmpfs             205296     8940    196356   5% /run
/dev/vda1       30830588 11168116  18073328  39% /
tmpfs            1026472      456   1026016   1% /dev/shm
tmpfs               5120        0      5120   0% /run/lock
tmpfs            1026472        0   1026472   0% /sys/fs/cgroup
tmpfs             205296       36    205260   1% /run/user/132
tmpfs             205296       24    205272   1% /run/user/0
```

Команда id отображает личность пользователя выполняющего сеанс, а также список групп, к которым он принадлежит. Т.к. доступ к некоторым файлам или устройствам может быть ограничен для членов группы, так что проверка доступности членства в группах может быть полезна.

```
$ id
uid=1000(buxy) gid=1000(buxy) groups=1000(buxy),27(sudo)
```

Команда uname -a возвращает одиночную строку, в которой записаны имя ядра (Linux), имя хоста, выпуск ядра, версия ядра, тип машины (строка архитектуры, такая как x86_64), и имя операционной системы (GN U / Linux). Вывод этой команды обычно должен включаться в отчеты об ошибках, так как он четко определяет используемое ядро и аппаратную платформу, на которой вы работаете.

```
$ uname -a
Linux kali 4.9.0-kali3-amd64 #1 SMP Debian 4.9.18-1kali1 (2017-04-04) x86_64 GNU/Linux
```

Все эти команды предоставляют информацию о времени исполнения, но довольно часто вам нужно обратиться к журналам, чтобы понять, что происходило на вашем компьютере. В частности, ядро отправляет сообщения, которые оно хранит в кольцевом буфере всякий раз, когда происходит что-то интересное (например, вставляемое новое USB-устройство, неудачная работа на жестком диске или первоначальное обнаружение аппаратного обеспечения при загрузке). Вы можете получить журналы ядра с помощью команды dmesg.

Журнал Systemd также хранит несколько журналов (stdout/stderr выводы демона, syslog сообщения, журналы ядра) и упрощает их запрос с помощью journalctl. Без каких-либо аргументов он просто выстраивает все доступные журналы в хронологическом порядке. С параметром -r он изменит порядок, чтобы сначала отображались новые сообщения. С параметром -f он будет непрерывно печатать новые записи журнала, поскольку они добавляются в его базу данных. Параметр -u может ограничивать сообщения теми, которые испускаются определенным модулем systemd (например: journalctl -u ssh.service).

3.4.6 Обнаружение оборудования

Ядро экспортирует множество деталей об обнаруженном оборудовании через виртуальные файловые системы /proc/ and /sys/. Несколько инструментов суммируют эти детали. Среди них lspci (в пакете *pciutils*) перечисляет PCI устройства, lsusb (в пакете *usbutils*) перечисляет USB устройства, и lspcmcia (в пакете *pcmciautils*) перечисляет PCMCIA карты. Эти инструменты являются очень полезными для определения конкретной модели устройства. Эта идентификация также позволяет проводить более конкретные поиски в Интернете, что в свою очередь, приводит к нахождению более подходящих документов. Обратите внимание, что пакеты *pciutils* и *usbutils* являются уже установленными на базовой системе Kali, в то время как *pcmciautils* должен быть установлен с помощью apt install pcmciautils. Мы выделим больше времени на рассмотрение установки пакетов и их управлению в последующей главе.

Пример 3.1 *Пример информации, предоставленной lspci и lsusb*

```
$ lspci
[...]
00:02.1 Display controller: Intel Corporation Mobile 915GM/GMS/910GML Express Graphics Controller (rev 03)
00:1c.0 PCI bridge: Intel Corporation 82801FB/FBM/FR/FW/FRW (ICH6 Family) PCI Express Port 1 (rev 03)
00:1d.0 USB Controller: Intel Corporation 82801FB/FBM/FR/FW/FRW (ICH6 Family) USB UHCI #1 (rev 03)
[...]
01:00.0 Ethernet controller: Broadcom Corporation NetXtreme BCM5751 Gigabit Ethernet PCI Express (rev 01)
02:03.0 Network controller: Intel Corporation PRO/Wireless 2200BG Network Connection (rev 05)
$ lsusb
Bus 005 Device 004: ID 413c:a005 Dell Computer Corp.
Bus 005 Device 008: ID 413c:9001 Dell Computer Corp.
Bus 005 Device 007: ID 045e:00dd Microsoft Corp.
Bus 005 Device 006: ID 046d:c03d Logitech, Inc.
[...]
Bus 002 Device 004: ID 413c:8103 Dell Computer Corp. Wireless 350 Bluetooth
```

Эти программы имеют опцию -v, которая содержит гораздо более подробную (но обычно ненужную) информацию. Наконец, команда lsdev (в пакете *procinfo*) перечисляет ресурсы связи, используемые устройствами.

Программа lshw представляет собой комбинацию указанных выше программ и отображает подробное описание аппаратного обеспечения, обнаруженного в иерархическом порядке. Вы необходимо прикладывать полный вывод данной команды к

любому отчету о проблемах с поддержкой аппаратного обеспечения.

3.5 Подведем итоги

В этом разделе мы провели беглый обзор масштабного ландшафта Linux. Мы обсудили пространство ядра и пользователя, рассмотрели многие распространенные команды оболочки Linux, обсудили процессы и способы их управления, рассмотрели концепции безопасности пользователей и групп, обсудили FHS и обсудили некоторые из наиболее распространенных директорий и файлов, найденных в Kali Linux.

Суммируем все упомянутое:

- Linux часто используется для обозначения всей операционной системы, но на самом деле Linux является ядром операционной системы, которое запускается загрузчиком, который, в свою очередь, сам запускается BIOS / UEFI.
- Пользовательское пространство относится ко всему, что происходит за пределами ядра. Среди программ, работающих в пользовательском пространстве, есть много основных утилит из проекта GNU, большинство из которых предназначено для запуска из командной строки (текстовый интерфейс, который позволяет вводить команды, выполнять их и просматривать результаты). Оболочка выполняет ваши команды в этом интерфейсе.
- Список самых часто используемых команд включает в себя: pwd (отобразить рабочую директорию (print working directory)), cd (сменить директорию (change directory)), ls (перечислить содержимое директории (list file or directory contents)), mkdir (создать директорию (make directory)), rmdir (удалить директорию (remove directory)), mv, rm, and cp (переместить (move), удалить (remove), или скопировать (copy) файл или директорию соответственно), cat (связать или показать файл), less/more (показывать файлы по одной странице за раз), editor (запустить текстовый редактор), find (показать местоположение файла или директории), free (отобразить информацию о памяти), df (показать свободное пространство на диске), id

(отобразить личность пользователя вместе со списком групп, к которым он принадлежит), dmesg (просмотреть журнал ядра), и journalctl (показать все доступные журналы).

- Вы можете проверить аппаратное обеспечение в системе Kali несколькими командами: lspci (список PCI устройств), lsusb (список USB накопителей) и lspcmcia перечисляет карты PCMCIA.
- Процесс является рабочим экземпляром программы, который требует определенный объем памяти, как для хранения самой программы, так и для её оперативных данных. Вы можете управлять процессами с помощью таких команд как: ps (показать процессы), kill (завершить процессы), bg (отправить процесс в фоновый режим), fg (вывести процесс из фонового режима на передний план), и jobs (показать все фоновые процессы).
- Системы, основанные на Unix, являются многопользовательскими. Они поддерживают множество пользователей и групп, а также позволяют получить контроль над действиями на основе прав доступа. Вы можете управлять правами файла и директории с помощью нескольких команд, включая: chmod (изменить права доступа), chown (изменить владельца), chgrp (сменить группу).
- Как и все другие профессиональные дистрибутивы Linux, Kali Linux организован таким образом, чтобы соответствовать стандарту иерархии файловой системы (FHS) *(Filesystem Hierarchy Standard* (FHS)), что в свою очередь позволяет пользователям, пришедшим из других дистрибутивов Linux, с лёгкостью начать работать с Kali.
- Традиционно, конфигурации приложений хранятся в вашей домашней директории в скрытых файлах или директориях, названия которых начинаются точки.

Теперь, когда вы ознакомились с основами Linux, давайте перейдем к установке и запуску Kali Linux.

Часть 4: Установка Kali Linux

Содержание:

Ключевые слова главы:

- Установка
- Автоматическая
- установка ARM устройств
- Устранение неполадок

В этой главе мы подробно рассмотрим процесс установки Kali Linux. Сначала, мы обсудим минимальные системные требования (раздел 4.1, «Минимальные системные требования» [стр. 66]) для того, чтобы убедиться в том, что ваша реальная или виртуальная машина соответствует необходимым требованиям и настроена должным образом для проведения выбранного вами процесса установки. Далее мы пройдемся шаг за шагом по каждому этапу процесса установки (раздел 4.2, «Пошаговая установка на жесткий диск» [стр. 66]) начиная с обычной установки и заканчивая более безопасными процессами инсталляции, которые включают в себя полностью зашифрованную файловую систему. Мы также обсудим *пресидинг,* который делает доступной автоматическую установку (раздел 4.3, "Автоматические установки" [стр. 91]) путем предоставления предопределенных ответов на вопросы при установке. Также мы покажем вам, как установить Kali Linux на различные ARM устройства (раздел 4.4,"ARM Установки" [стр. 94]), что в свою очередь расширяет возможности Kali далеко за пределы рабочего стола. И наконец, мы продемонстрируем вам, что необходимо делать в тех редких случаях сбоя установки (раздел 4.5, "Устранение неполадок во время установки" [стр. 95]), таким образом, что вы с легкостью сможете решить проблему и успешно завершить процесс инсталляции.

4.1 Минимальные системные требования

Требования к установке для Kali Linux различаются в зависимости от того, чтобы вы хотели установить. В качестве нижней границы, вы можете установить в виде севера Secure Shell (SSH) без рабочего стола, используя всего 128 MB ОЗУ (рекомендуется 512 MB ОЗУ) и 2 Гб дискового пространства. Если вам необходимо произвести установку Kali Linux с широкими функциональными возможностями, например, со средой рабочего стола по умолчанию GNOME и мета-пакетом *kali-linux-full*, то вам понадобится как минимум к 2048 MB ОЗУ и 20 Гб дискового пространства.

Помимо требований к ОЗУ и жесткому диску, ваш компьютер должен иметь процессор, поддерживаемый хотя бы одной из архитектур amd64, i386, armel, armhf или arm64.

4.2 Пошаговая установка Kali Linux на жесткий диск

В данном разделе мы исходим из того, что у вас уже есть загрузочный USB накопитель или DVD (смотри раздел 2.1.4 “Копирование образа на DVD-ROM или USB накопитель” [стр. 19] для получения детальной информации о том, как подготовить загрузочный накопитель) и что вы загружаетесь с него для начала процесса установки.

4.2.1 Обычная установка

Сначала, мы рассмотрим стандартную установку Kali с незашифрованной файловой системой.

Загрузка и запуск установщика

Как только BIOS начнет загрузку с USB накопителя или DVD-ROM, появится меню загрузки linux, как это показано на рисунке 4.1 «Экран загрузки» [стр. 67]. На этом этапе, ядро Linux еще не загружено; это меню позволяет вам выбрать необходимое ядро для загрузки и ввести дополнительные параметры, которые будут переданы ему в процессе.

Для стандартной установки вам понадобится только выбрать Install или Graphical Install (с помощью клавиш стрелочек на клавиатуре), затем нажать клавишу Enter чтобы начать оставшуюся часть процесса установки.

Каждая позиция меню скрывает определенную командную строку загрузки, которая может быть настроена в случае необходимости путем нажатия клавиши Tab до подтверждения ввода и загрузки.

Рисунок 4.1 *Экран загрузки*

После загрузки, программа установки проведет вас шаг за шагом через весь процесс. Мы рассмотрим каждый из этих шагов более детально. Мы затронем установку со стандартного DVD-ROM Kali Linux, так как установка из mini.iso может выглядеть несколько иначе. Мы также обратимся к графическому режиму установки, но единственным отличием данного режима от классического текстового режима установки является внешний вид.

В версиях задаются одинаковые вопросы и представлены одинаковые опции.

Как показано на рисунке 4.2, «Выбираем язык» [стр. 68], программа установки стартует на английском языке по умолчанию, но первый шаг позволяет вам выбрать язык, который будет использоваться до конца установки. Этот выбор языка также используется для определения языка по умолчанию на последующих этапах (в частности, раскладки клавиатуры).

Навигация с помощью клавиатуры

Некоторые шаги в процессе установки требуют ввода информации. Эти экраны имеют несколько областей, которые могут обладать фокусом ввода данных (область ввода текста, флажки, список вариантов, кнопки «ОК» и «Отмена»), а клавиша «Tab» позволяет переходить от одного фокуса к другому. В графическом режиме установки вы можете использовать мышь как обычно на установленном графическом рабочем столе.

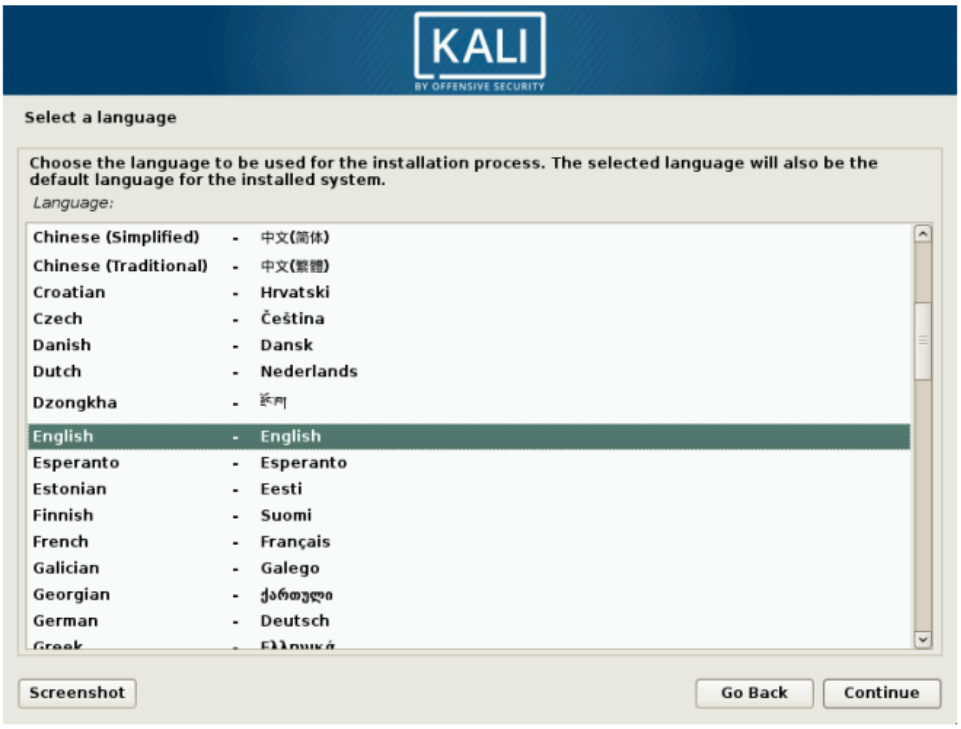

Рис. 4.2 Выбор языка

Выбор страны

Второй шаг (Рисунок 4.3, “Выбираем страну” [стр. 69]) состоит в том, чтобы выбрать вашу страну. В сочетании с языком эта информация позволяет программе установки предлагать наиболее подходящую раскладку клавиатуры. Это также повлияет на настройку часового пояса. В Соединенных Штатах предлагается стандартная клавиатура QWERTY, и установщик представляет выбор подходящих часовых поясов.

Рисунок 4.3 Выбираем страну

Выбор раскладки клавиатуры

Предложенная раскладка клавиатуры американского английского соответствует обычной раскладке QWERTY, как показано на Рисунке 4.4, “Выбор раскладки клавиатуры” [стр. 70].

Рисунок 4.4 *Выбор раскладки клавиатуры*

Обнаружение оборудования

В подавляющем большинстве случаев шаг аппаратного обнаружения полностью автоматизирован. Установщик обнаруживает ваше оборудование и пытается идентифицировать загрузочное устройство, используемое для доступа к его контенту. Он загружает модули, соответствующие различным обнаруженным аппаратным компонентам, а затем монтирует загрузочное устройство для его чтения. Предыдущие шаги полностью содержатся в загрузочном образе, включая загрузочное устройство, файл ограниченного размера, а также загружаются в память с помощью начального загрузчика в момент получения данных с загрузочного устройства.

Загрузка компонентов

Теперь, когда содержимое загрузочного устройства доступно, установщик загружает все файлы, необходимые для продолжения своей работы. Сюда входят дополнительные драйвера для оставшегося оборудования (особенно сетевой карты), а также все компоненты программы установки. На этом этапе, установщик попробует автоматически определить сетевую карту и загрузить соответствующий модуль. Если автоматическое определение не удастся, то вы сможете попробовать загрузить необходимый модуль вручную. Если же и альтернативные способы завершились неудачей, вы можете загрузить определенный модуль со съемного устройства. К последнему способу следует прибегать лишь в случае, когда нужный драйвер не включен в стандартное ядро Linux, но доступен в другом месте, как, например, на сайте производителя.

Этот шаг должен быть абсолютно успешным для сетевых установок (например, для тех, которые были загружены с mini.iso), так как пакеты Debian должны быть загружены из сети.

Настройка сети

Для того чтобы максимально автоматизировать процесс, установщик пробует автоматически настроить установки сети используя динамический протокол настройки хоста (dynamic host

configuration protocol (DHCP)) (для IPv4 и IPv6) и ICMPv6's Neighbor Discovery Protocol (для IPv6), как показано на рисунке 4.5, «Автоматическая настройка сети» [page 71].

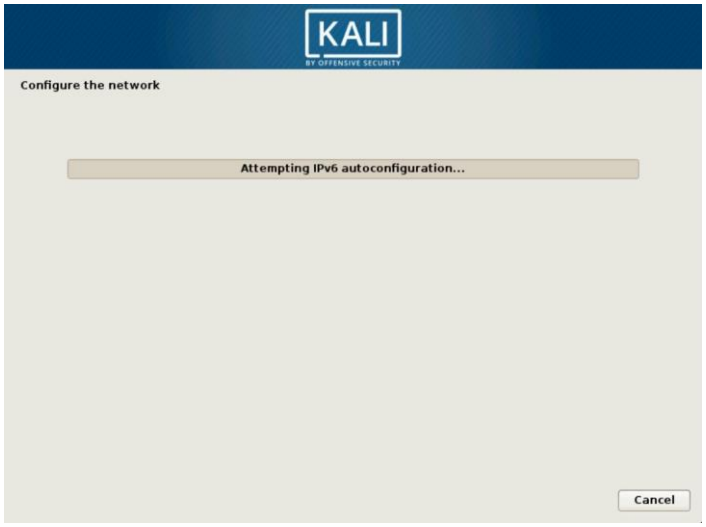

Рисунок 4.5 *Автоматическая настройка сети*

Если автоматическая настройка не удалась, программа установки предложит вам больше вариантов: например, повторить попытку с нормальной конфигурацией DHCP, попробовать настроить DHCP, указав имя машины или установить статические сетевые настройки.

Эта последняя опция требует IP-адрес, маску подсети, IP-адрес для потенциального шлюза, имя машины и имя домена.

Конфигурация без DHCP

Если локальная сеть оборудована сервером DHCP, который вы не хотите использовать, поскольку предпочитаете определять статический IP-адрес для машины во время установки, вы можете добавить параметр **netcfg / use_dhcp = false** при загрузке. Вам просто нужно отредактировать нужную позицию меню, нажав клавишу Tab и добавив желаемую опцию, прежде, чем нажимать клавишу Enter.

Корневой пароль

Установщик запрашивает пароль (рисунок 4.6, «Корневой пароль» [стр. 72]), поскольку он автоматически создает суперпользовательскую учетную запись root. Установщик также запрашивает подтверждение пароля, чтобы предотвратить любую ошибку ввода, которую впоследствии будет очень трудно устранить.

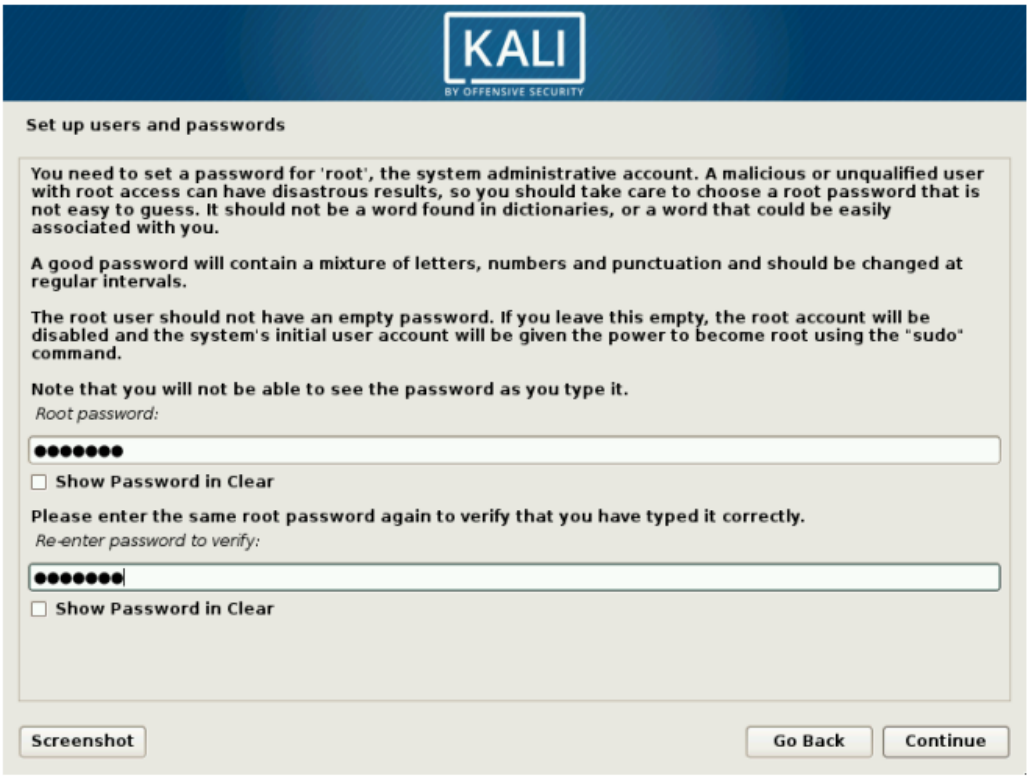

Рис. 4.6 Пароль root

Пароль администратора

Пароль пользователя root должен быть длинным (восемь символов или даже больше) и уникальным, поскольку злоумышленники, атакуя компьютеры и серверы, подключенные к Интернету, используют автоматизированные инструменты, которые пытаются войти в систему, используя самые очевидные и общепринятые пароли. Иногда злоумышленники используют атаку перебор по словарю (dictionary attack), суть которой заключается в переборе множества комбинаций слов и чисел в качестве пароля. Мы настоятельно не рекомендуем вам использовать имена детей или родителей, а также дни рождения ваших близких, потому что подобного рода пароли, всегда подбираются с особой легкостью.

Эти замечания одинаково применимы к другим паролям пользователей, но последствия взлома обычной учетной записи менее критичны для пользователей, не имеющих прав администратора.

Если же вы переживаете творческий кризис и затрудняетесь придумать уникальный пароль, то, абсолютно не задумываясь, используйте генератор паролей, такой как pwgen (его можно найти в пакете с таким же именем, который включен в базовую установку Kali).

Настройка часов

Если сеть является доступной, внутренние часы системы будут обновлены из сервера сетевого протокола времени (network time protocol (NTP) server). Это является очень полезным, потому что гарантирует, что отметки времени в журналах будут корректными с первой загрузки.

Если ваша страна охватывает несколько часовых поясов, вас попросят выбрать часовой пояс, который вы хотите использовать, как это показано на рисунке 4.7, «Выбор часового пояса» [page 73].

Рис. 4.7 Выбор часового пояса

Определение дисков и других устройств

Этот шаг автоматически определяет жесткие диски, на которых может быть установлен Kali, каждый из которых будет представлен в следующем шаге: выделение разделов.

Выделение разделов

*Выделение разделов я*вляется необходимым шагом в процессе установки, который состоит в том, чтобы разделить доступное место на жестком диске на отдельные части (*разделы*) в соответствии с предполагаемой функцией компьютера и этих разделов. Выделение разделов также включает в себя определение той файловой системы, которую будет использовать этот раздел. Сделанный вами выбор будет иметь огромное влияние на производительность, безопасность данных и управление сервером.

Шаг, связанный с выделением разделов, обычно является довольно сложным для новых пользователей. Однако файловые системы и разделы Linux, включая виртуальную память (или разделы *подкачки*), должны быть определены, поскольку они образуют фундамент системы. Эта задача может быть усложнена, если вы уже установили другую операционную систему на вашу машину, и вы хотите, чтобы эти две система совместно сосуществовали. В этом случае вы должны быть уверены, что не измените свои разделы или, если потребуется, измените их размер без нанесения какого-либо вреда их содержимому.

Для использования более общей (и простой) схемы разделения, большинство пользователей предпочтут *Управляемый (Guided)* режим, который сможет порекомендовать вам конфигурацию разделов и предоставить советы относительно каждого шага данного процесса. Более продвинутые пользователи, скорее всего, предпочтут использовать режим *Ручной настройки,* который позволяет использовать более расширенные конфигурации. Каждый режим имеет определенные возможности.

Рисунок 4.8 *Выбор режима разделения*

Управляемое разделение

Первый экран в инструменте разделения (рисунок 4.8, «Выбор режима разделения») представляет общее описание для управляемого и ручного режимов разделения. “Управляемый – использовать целый диск” (“Guided - use entire disk”) является самой простой и самой распространенной схемой разделов, которая выделяет весь диск для Kali Linux.

Следующие два варианта используют Logical Volume Manager (LVM) для настройки логических (вместо физических) выборочно зашифрованных разделов. Мы обсудим LVM и шифрование немного позже в этой главе.

И наконец, последний вариант запускает режим процесса разделения вручную, который позволяет использовать более функциональную схему разделения, как например, установить Kali Linux наряду с другими операционными системами. Мы обсудим ручной режим в следующем разделе.

В этом примере, мы будем выделять весь диск для Kali, что означает, что мы выбрали “Guided - use entire disk” для перехода к следующему шагу.

Следующий экран (показанный на рисунке 4.9 «Диск, используемый для управляемого разделения» [page 75]) позволяет вам выбрать диск, на который вы хотите установить, путем выбора соответствующего пункта (например, “Virtual disk 1 (vda) - 32.2 GB Virtio Block Device”). После выбора, управляемое разделение продолжится. На этом этапе все данные на диске будут безвозвратно стерты, так что подходите к процессу выбора с умом.

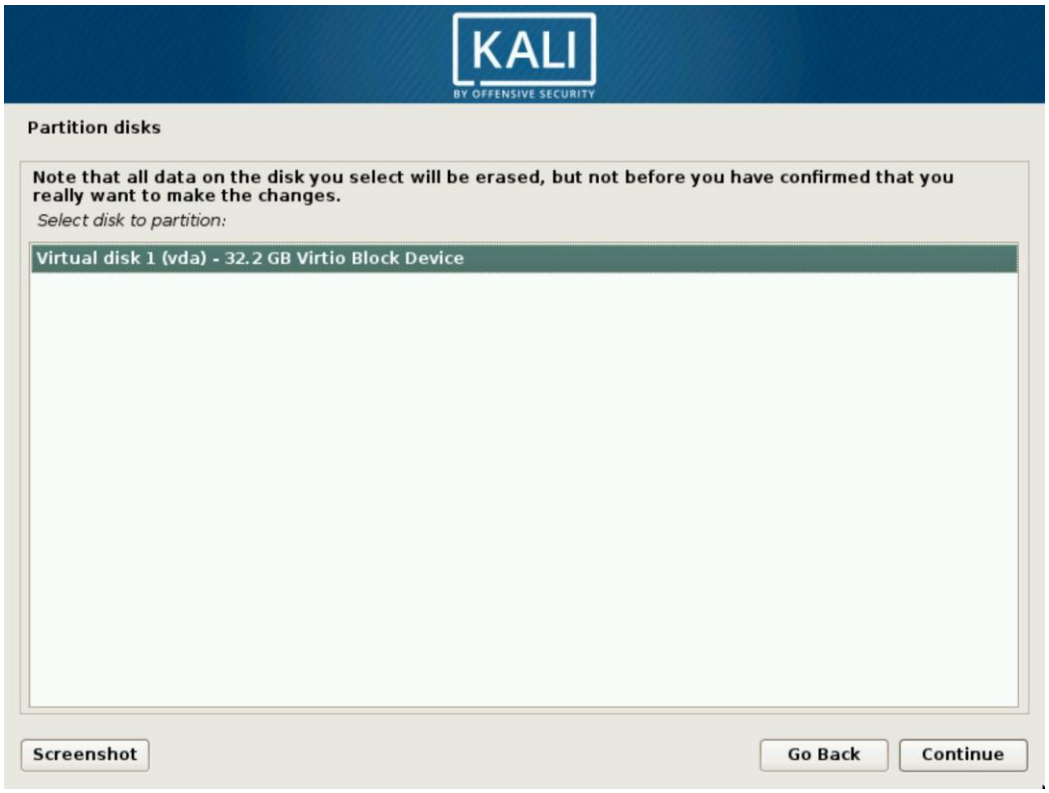

Рисунок 4.9 *Диск, используемый для управляемого разделения*

Далее, инструмент управляемого разделения предложит 3 метода разделения, которые соответствуют различным схемам разделения, как показано на рисунке 4.10, «Управляемое распределение разделов» [стр. 76].

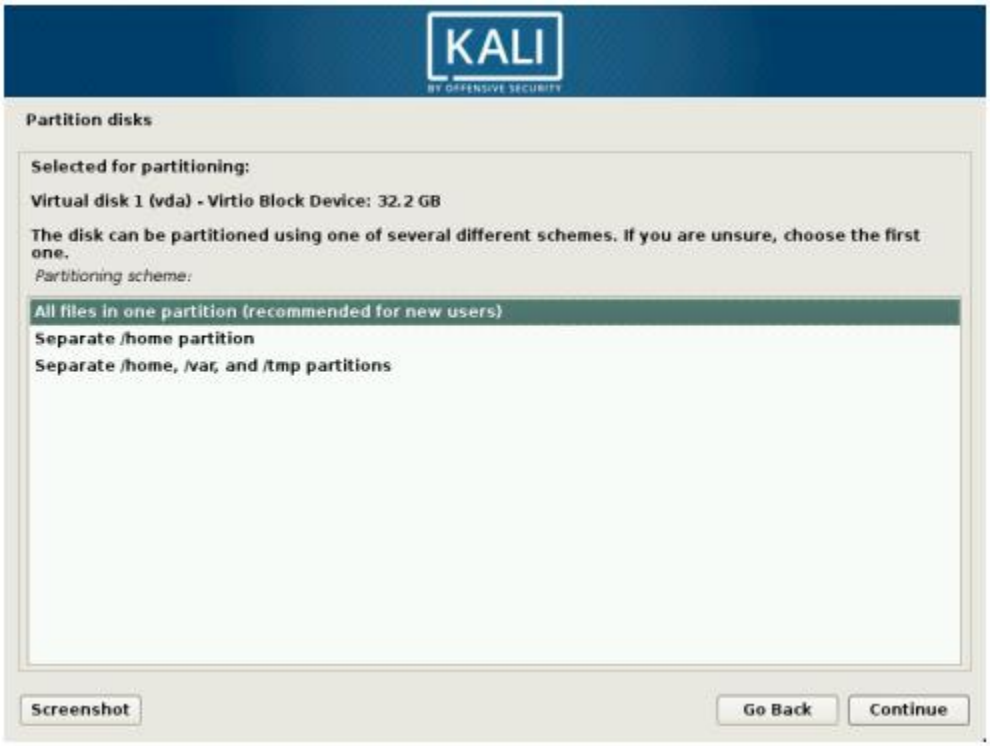

Рисунок 4.10 *Управляемое распределение разделов*

Первый метод называется «Все файлы в одном разделе» (“All files in one partition.”). Все дерево системы Linux хранится в одной файловой системе, соответствующей корневому каталогу («/»). Эта простая и надежная схема разделения отлично работает для персональных или однопользовательских систем. Несмотря на название, на самом деле будут созданы два раздела: в первом будет размещена полная система, во втором - виртуальная память (или “swap”).

Второй метод «Отдельное /home/ разделение» («Separate /home/ partition») является аналогичным, но он разделяет иерархию файлов на две части: один раздел содержит систему Linux (/), а второй содержит «домашние каталоги» (что означает данные пользователя, в файлах и подкаталогах доступны под /home/). Одно из преимуществ этого метода заключается в том, что вам будет легко сохранить данные пользователей, если вам нужно переустановить систему.

И наконец, последний метод разделения, под названием «Отдельное /home, /var, и /tmp разделение» (“Separate /home, /var, and /tmp partitions,”) очень хорошо подходит для серверов и многопользовательских систем. Он делит дерево файлов на

множество разделов: помимо разделов root (/) и учетных записей пользователей (/ home /), он также имеет разделы для данных программного обеспечения сервера (/ var /) и временных файлов (/ tmp /). Одно из преимуществ этого метода заключается в том, что конечные пользователи не могут блокировать сервер, потребляя все доступное пространство на жестком диске (они могут заполнять только / tmp / и / home /).В то же время данные демона (особенно журналы) больше не могут забивать остальную часть системы.

После выбора типа раздела установщик представит вам сводку ваших выборов на экране в виде карты разделов (рисунок 4.11, «Проверка разбиения на разделы» [стр. 77]). Вы можете отредактировать каждый раздел по отдельности путем выбором необходимого раздела. Например, вы можете сменить файловую систему, используемую разделом, если стандартная *(ext4)* вам не подходит. Однако, в большинстве предлагаемое разделение является разумным, и вы можете принять его, выбрав «Завершить разбиение на разделы и записать изменения на диск». Безусловно, можно совершить данный выбор по умолчанию, но выбирайте с умом, т.к. все данные на выбранном диске будут стерты.

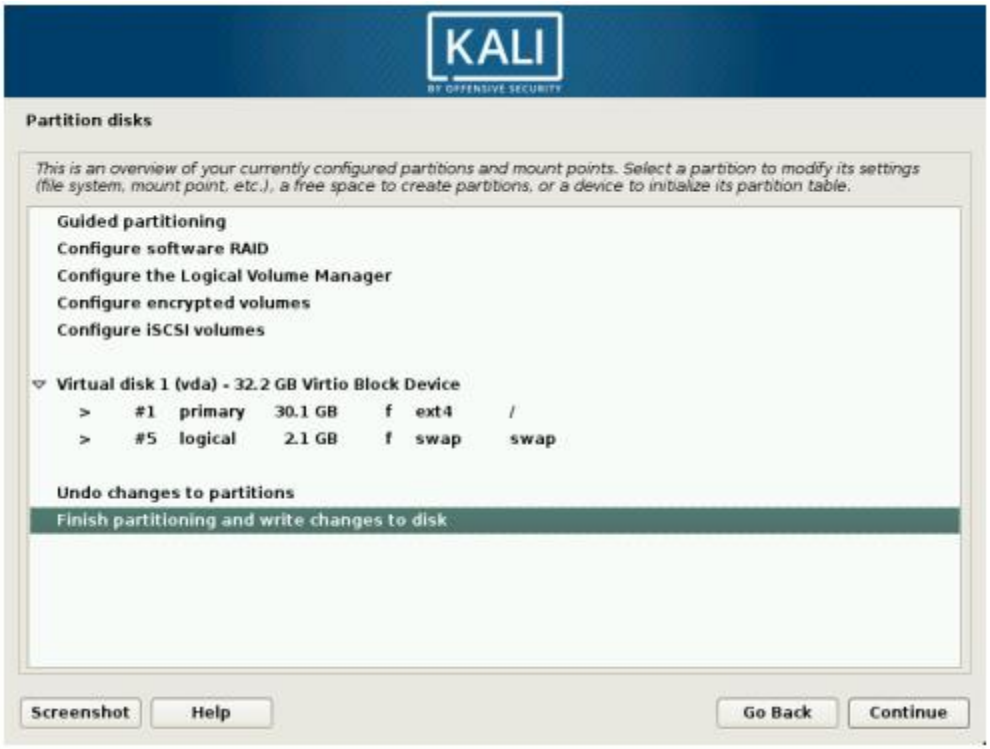

Рисунок 4.11 *Проверка разбиения на разделы*

Ручное разделение

Выбор Manual (Вручную) на главном экране “Разделение дисков” (Рисунок 4.8, “*Выбор режима разделения*” [стр.74]), предоставит вам намного большую гибкость, позволяя выбирать множество дополнительных конфигураций и конкретно выбирать назначение и размер каждого раздела. Например, этот режим позволит вам установить Kali наряду с другими операционными системами, включить резервный набор независимых дисков (redundant array of independent disks (RAID)) на основе программного обеспечения для защиты данных от сбоев в работе жесткого диска и безопасно изменять размеры существующих разделов без потери данных, помимо прочего.

Сокращение Разделов Windows

Для того, чтобы установить Kali Linux на компьютер, на котором уже имеется другая операционная система (например, Windows), вам понадобится использовать незадействованное пространство жесткого диска для разделов, предназначенных для Kali. В большинстве случаев это означает сокращение существующего раздела и повторное использование освобожденного пространства.
Если вы используете режим разделения вручную, то установщик может сократить некоторые разделы Windows довольно просто. Вам всего на всего понадобится выбрать раздел Windows и ввести его новый размер (этот способ работает как с FAT разделами, так и с NTFS разделами).

Если вы являетесь менее опытным пользователем, который работает с системой, содержащей некоторые данные, будьте очень осторожны с этим методом настройки, так как очень легко совершать ошибки, которые могут привести к безвозвратной потере данных.

Первый экран в ручном установщике является практическими таким же, как и на рисунке 4.11 «Проверка выделения разделов» [стр. 77], за исключением того, что он не содержит никаких новых разделов для создания. Это зависит от вас, захотите ли вы их добавить.

Сначала вы увидите опцию для ввода «Управляемое разделение», за которой будут следовать некоторые опции конфигурации. Затем установщик покажет доступные диски, их разделы и любое свободное пространство, которое еще не было распределено. Вы можете выбрать каждый отображаемый элемент и как обычно нажать клавишу Enter для взаимодействия с ним.

Если диск совершенно новый, вам может потребоваться создать таблицу разделов. Вы можете сделать это, выбрав диск. После этого вы должны увидеть свободное место на диске.

Чтобы использовать это свободное пространство, вы должны выбрать его, и установщик предложит вам два способа создания разделов на этом пространстве.

Рисунок 4.12 *Создание разделов на пустом дисковом пространстве*

Первый пункт создаст отдельный раздел с характеристиками (включая размер) по вашему выбору. Второй пункт будет использовать все свободное пространство и создавать в нем несколько разделов с помощью управляемого мастера разделения (см. Раздел 4.2.1.12.1 «Управляемое разделение» [стр. 75]). Этот параметр особенно интересен, если вы хотите установить Kali вместе с другой операционной системой, но если вы не хотите управлять схемой разделения. Последний пункт покажет соответственно номера цилиндра/головки/сектора (cylinder/head/sector) начала и конца свободного пространства.

Когда вы выберите «Создать новый раздел», вы попадете в сердце ручной последовательности разделения. После выбора этой опции вам будет предложено указать размер раздела. Если на диске используется таблица разделов MSDOS, вам будет предоставлена возможность создать основной или логический раздел. (Важно знать: у вас может быть всего четыре основных раздела и множество логических. Раздел, содержащий /boot, и соответственно ядро, должен быть основным. Логические разделы постоянно хранятся в расширенном разделе, который в свою очередь, использует один из четырех основных разделов.) Затем вы должны увидеть общий экран конфигурации раздела:

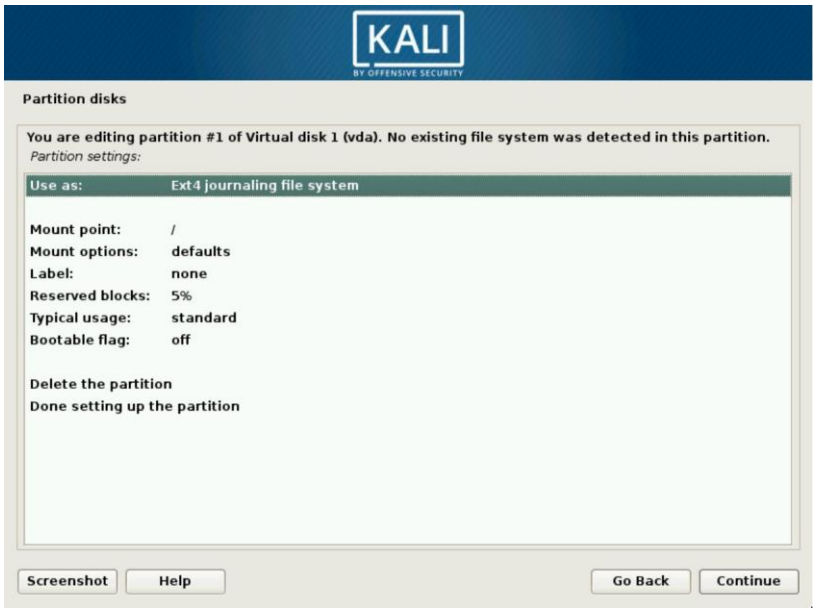

Рисунок 4.13 *Экран настройки разделения*

Чтобы обобщить этот шаг ручного разделения, давайте рассмотрим, что вы можете делать с новым разделом. Вы можете:

- Отформатировать его и включить в дерево файлов, выбрав точку монтирования. Точка монтирования - это каталог, в котором будет размещаться содержимое файловой системы на выбранном разделе. Таким образом, раздел, смонтированный в /home/, традиционно предназначен для хранения пользовательских данных, а «/» известен как *root* для дерева файлов и, следовательно, *root* раздела, на котором будет фактически установлена система Kali.
- Использовать его как *раздел подкачки (swap partition).* Когда в

ядре Linux недостаточно свободной памяти, он будет хранить неактивные части ОЗУ в специальном разделе подкачки на жестком диске. Подсистема виртуальной памяти делает его прозрачным для приложений. Для имитации дополнительной памяти Windows использует файл подкачки (swap или paging file), который содержится непосредственно в файловой системе. В то время как Linux использует раздел, предназначенный для этой цели, отсюда и возникает термин раздел подкачки (swap partition).

- Сделать его «физическим томом для LVM» (этот вопрос не рассматривается в нашей книге). Обратите внимание, что эта функция используется управляемым разделением при настройке зашифрованных разделов.
- Сделать его «физическим томом для шифрования», чтобы защитить конфиденциальность данных на определенных разделах. Этот момент автоматизирован в управляемом разделении. Для получения дополнительной информации см. Раздел 4.2.2 «Установка на полностью зашифрованную файловую систему» [стр. 85].
- Использовать его как RAID устройство (данный вопрос не рассмотрен в нашей книге).
- Не использовать раздел и оставьте его неизменным.
- По завершению вы можете либо отказаться от ручного разбиения диска, выбрав «Отменить изменения в разделах», или записать свои изменения на диск, выбрав «Завершить разбиение на разделы и записать изменения на диск» на экране ручного установщика (рисунок 4.11 «Проверка разбиения на разделы») [стр. 77]).

Копирование живого образа

Этот следующий шаг, который не требует какого-либо взаимодействия с пользователем, копирует содержимое живого образа в целевую файловую систему, как показано на рисунке 4.14 «Копирование данных из живого образа» [стр. 80].

Рисунок 4.14 *Копирование данных из живого образа*

Настройка диспетчера пакетов (apt)

Чтобы иметь возможность устанавливать дополнительное программное обеспечение, необходимо настроить APT и указать, где оно сможет найти пакеты Debian. В Kali этот шаг, в основном, является неинтерактивным, поскольку мы настаиваем на использовании зеркала http.kali. org. Вам всего на всего остается подтвердить, хотите ли вы использовать это зеркало (Рисунок 4.15, “Вы хотите использовать сетевое зеркало?” [стр. 81]). Если же вы не хотите использовать его, вы не сможете установить дополнительные пакеты с помощью apt, до тех пор, пока вы не настроите репозиторий пакетов.

Рисунок 4.15 *Вы хотите использовать сетевое зеркало?*

Если вы хотите использовать локальное зеркало вместо http.kali.org, вы можете ввести его имя в командной строке ядра (при загрузке), используя синтаксис, который будет выглядеть следующим образом: mirror/http/hostname=*my.own.mirror*.

И наконец, программа предлагает использовать *HTTP proxy,* как показано на рисунке 4.16, «Используйте HTTP Proxy» [стр. 82]. HTTP proxy является сервером, который отправляет HTTP-запросы для пользователей сети. Иногда это помогает ускорить загрузки путем сохранения копии файлов, которые были переданы через него (в таком случае мы говорим о кэшировании прокси). В некоторых случаях это единственный способ доступа к внешнему веб-серверу; в таких случаях установщик сможет загружать пакеты Debian, если вы правильно заполните это поле во время установки. Если вы не указали адрес прокси-сервера, установщик попытается подключиться непосредственно к Интернету.

Рисунок 4.16 *Используйте HTTP Proxy*

Затем файлы Packages.xz и Sources.xz будут автоматически скачаны, для того, чтобы обновить список пакетов, распознанных APT.

Установка загрузчика GRUB

Загрузчик является первой программой, запускаемой BIOS. Эта программа загружает ядро Linux в память и затем выполняет его. Загрузчик часто предлагает меню, которое позволяет вам выбрать загружаемое ядро или операционную систему.

Благодаря своему техническому превосходству GRUB является загрузчиком, который по умолчанию устанавливается Debian: он работает с большинством файловых систем и поэтому не требует обновления после каждой установки нового ядра, поскольку он считывает его конфигурацию во время загрузки и находит точное положение нового ядра.

Вы должны установить GRUB в главную загрузочную запись (Master Boot Record (MBR)), если у вас уже не установлена другая система Linux, которая знает, как загружать Kali Linux. Как отмечено на рисунке 4.17, «Установка загрузчика GRUB на

жесткий диск» [стр. 83], изменение MBR сделает нераспознанные операционные системы, которые зависят от нее, незагружаемыми, до тех пор, пока вы не исправите конфигурацию GRUB.

Рисунок 4.17 *Установка загрузчика GRUB*

В этом шаге (Рисунок 4.18, «Устройство для установки загрузчика» [стр. 84]), вам необходимо будет выбрать, на какое устройство будет установлен GRUB. Это должен быть ваш текущий загрузочный диск.

Рисунок 4.18 *Устройство для установки загрузчика*

По умолчанию в меню загрузки, предлагаемом GRUB, отображаются все установленные ядра Linux, а также любые другие операционные системы, которые были обнаружены. Именно поэтому, вам следует принять предложение установить его в главную загрузочную запись (Master Boot Record). Сохранение старых версий ядра сохраняет возможность загрузки системы, если последнее установленное ядро повреждено или плохо адаптировано к оборудованию. Поэтому мы рекомендуем сохранить несколько старых версий ядра.

Будьте внимательны: загрузчик и двойная загрузка

Эта фаза процесса установки обнаруживает операционные системы, которые уже установлены на компьютере, и автоматически добавит соответствующие записи в меню загрузки. Однако не все программы установки делают это.

В частности, если после этого вы установите (или переустановите) Windows, загрузчик будет удален. Кали будет по-прежнему находиться на жестком диске, но больше не будет доступна из меню загрузки. Затем вам нужно будет запустить установщик Kali с параметром **rescue/enable=true** в командной строке ядра, чтобы переустановить загрузчик. Эта операция подробно описана в руководстве по установке Debian.

http://www.debian.org/releases/stable/amd64/ch08s07.html

Завершение установки и перезагрузка

Теперь, когда установка завершена, программа попросит вас извлечь DVD-ROM из привода чтения (или отсоединить USB-накопитель), чтобы ваш компьютер мог загрузить вашу новую систему Kali после того, как программа установки перезапустит компьютер (рисунок 4.19, «Установка завершена» [Стр. 85]).

И наконец, установщик выполнит некоторую работу по очистке, вроде удаления пакетов предназначенных для создания живой среды.

Рисунок 4.19 *Установка завершена*

4.2.2 Установка на полностью зашифрованную файловую систему

Для того чтобы гарантировать конфиденциальность ваших данных, вы можете установить зашифрованные разделы. Это надежно защитит ваши данные в случае потери или кражи вашего

ноутбука. Инструмент разбиения на разделы как управляемый, так и ручной может, помочь вам в этом процессе.

Режим управляемого разделения будет сочетать использование двух технологий: Linux Unified Key Setup (LUKS) для шифрования разделов и управления логическими томами (Logical Volume Management (LVM)) для динамического управления хранилищем. Обе функции также можно установить и настроить в режиме разделения вручную.

Введением в LVM

Давайте сначала обсудим LVM. Используя терминологию LVM, *виртуальный раздел* представляет собой логический том, который является частью *группы томов* или объединением нескольких физических томов. Физические тома являются реальными разделами (или виртуальными разделами, экспортируемыми другими абстракциями, такими как программное устройство RAID или зашифрованный раздел).

Благодаря отсутствию явных различий между «физическими» и «логическими» разделами, LVM позволяет создавать «виртуальные» разделы, которые охватывают несколько дисков. Это имеет двойное преимущество: размер разделов больше не ограничен отдельными дисками, а их совокупным объемом, и вы можете изменить размер существующих разделов в любое время, например, после добавления дополнительного диска.

Эта техника работает очень просто: каждый том, будь то физический или логический, разделяется на блоки одинакового размера, которые LVM коррелирует. Добавление нового диска приведет к созданию нового физического тома, обеспечивающего новые блоки, которые могут быть связаны с любой группой томов. Все разделы в группе томов могут в полной мере использовать дополнительное распределяемое пространство.

Введение в LUKS

Для защиты своих данных вы можете добавить дополнительный уровень шифрования под предпочтительную вам файловую систему. Linux (и, в частности, драйвер *dm-crypt*) использует

устройство управления распределением памяти для создания виртуального раздела (содержимое которого защищено) на базе основного раздела, который будет хранить данные в зашифрованном виде (благодаря LUKS). LUKS стандартизирует хранение зашифрованных данных также, как и метаинформации, которая обозначает используемые алгоритмы шифрования.

Зашифрованный раздел подкачки

Когда используется зашифрованный раздел, ключ шифрования сохраняется в памяти (ОЗУ), а при спящем режиме ноутбук будет копировать ключ вместе с другим содержимым ОЗУ на раздел подкачки жесткого диска. Поскольку любой, у кого есть доступ к файлу подкачки (это может быть как технический специалист, так и вор), может извлечь ключ и расшифровать ваши данные. В связи с этим файл подкачки должен быть защищен с помощью шифрования. Поэтому установщик предупредит вас, если вы попытаетесь использовать зашифрованный раздел рядом с незашифрованным разделом подкачки.

Настройка зашифрованных разделов

Процесс установки для зашифрованного LVM такой же, как и стандартная установка, за исключением этапа разбиения на разделы (рис. 4.20, «Управляемое разделение с зашифрованным LVM» [стр. 87]), где вам необходимо будет выбрать Управляемый режим - использовать весь диск и настроить зашифрованный LVM (“Guided - use entire disk and set up encrypted LVM.”). Конечным результатом будет система, которую нельзя загрузить или получить доступ до тех пор, пока не будет предоставлена кодовая фраза шифрования. Это зашифрует и защитит данные на вашем диске.

Рисунок 4.20 *Управляемое разделение с зашифрованным LVM*

Установщик управляемого разделения автоматически назначит физические разделы для хранения зашифрованных данных, как это показано на Рисунке 4.21, «Подтвердить изменения таблицы разделов» [стр. 88]. На этом этапе установщик подтвердит изменения до того, как они будут записаны на диск.

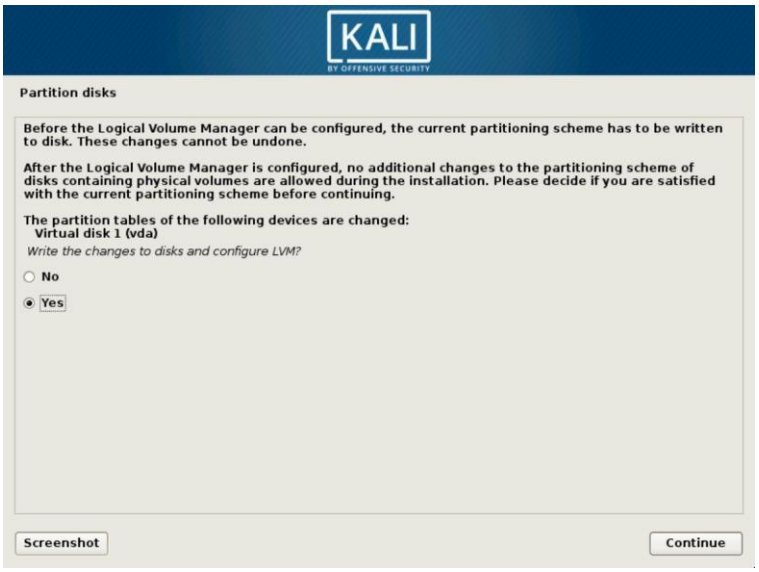

Рисунок 4.21 *Подтвердить изменения таблицы разделов*

Затем этот новый раздел инициализируется случайными данными, как показано на рисунке 4.22, «Удаление данных в зашифрованном разделе» [стр. 88]. Это делает области, которые содержат данные, неотличимыми от неиспользуемых областей, что в свою очередь затрудняет обнаружение и последующую атаку зашифрованных данных.

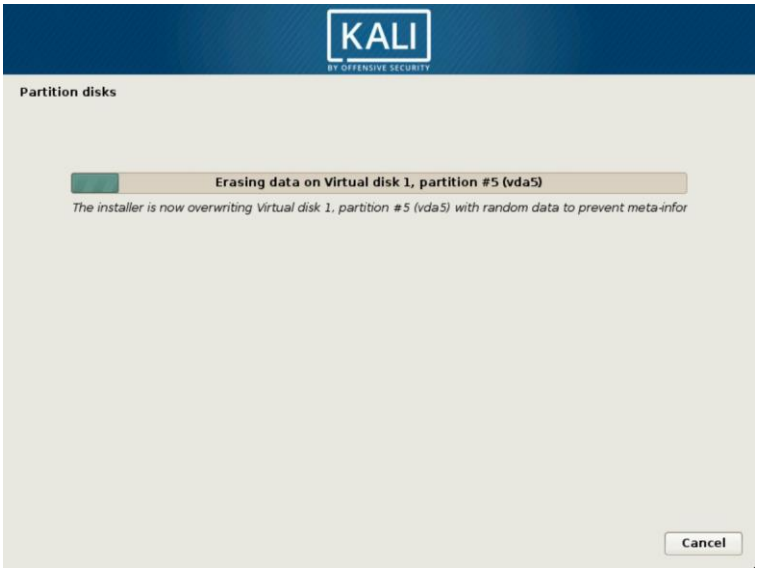

Рисунок 4.22 *Удаление данных в зашифрованном разделе*

Затем, установщик попросит вас ввести идентификационную фразу шифрования (Рисунок 4.23, «Введите идентификационную фразу шифрования» [page 89]). Для того чтобы просмотреть содержимое зашифрованного раздела, вам необходимо будет вводить эту идентификационную фразу каждый раз, как вы будете перезагружать систему. Обратите внимание на предупреждение в установщике: ваша зашифрованная система будет настолько хорошо защищена, насколько грамотно подобрана идентификационная фраза.

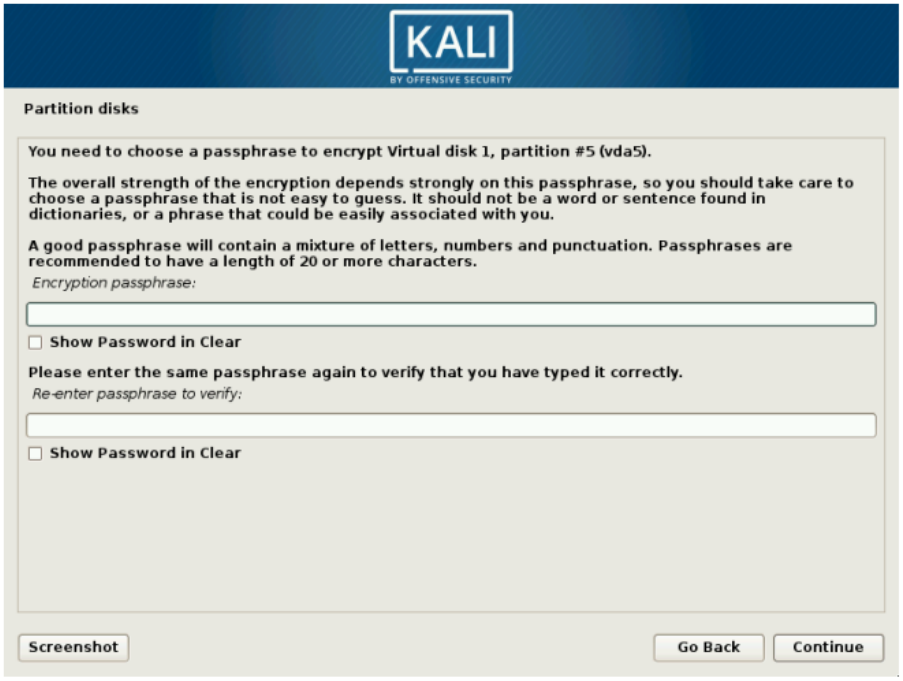

Рисунок 4.23 *Введете идентификационную фразу шифрования*

Теперь инструмент разделения обладает доступом к новому виртуальному разделу, чье содержимое хранится зашифрованным образом в основном физическом разделе. В связи с тем, что LVM использует этот новый раздел как физический том, он может защищать несколько разделов (или логических томов LVM) с помощью того же самого ключа шифрования, включая раздел подкачки (смотри вставку «Шифрование раздела подкачки» [стр. 86]). В данном случае, LVM не используется для упрощения расширения размера хранилища, он используется лишь для удобства косвенности, позволяющей разделить один зашифрованный раздел на несколько логических томов.

Завершение управляемого разделения с зашифрованным LVM
Затем на экране отобразится итоговая схема разделения (рисунок 4.24, «Проверка разделения для зашифрованной установки LVM» [стр. 90]) так, что вы можете выставить настройки необходимым вам образом.

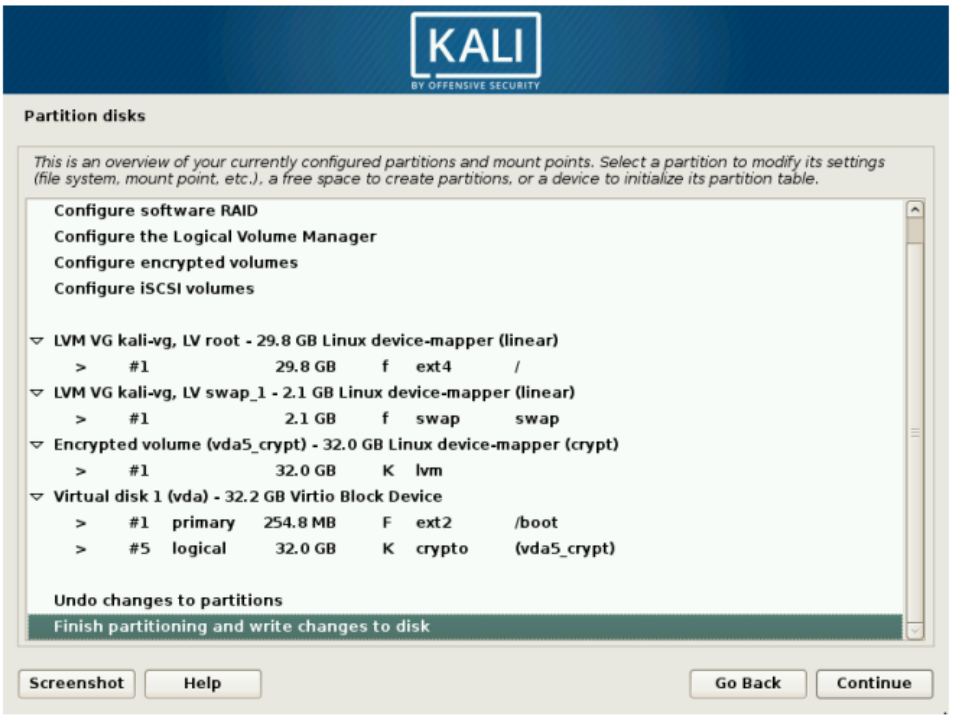

Рисунок 4.24 *Проверка разделения для зашифрованной установки LVM*

И наконец, после проверки настроек раздела, инструмент попросит вас подтвердить изменения, внесенные на диск, как это показано на рисунке 4.25, «Подтверждение форматирования разделов».

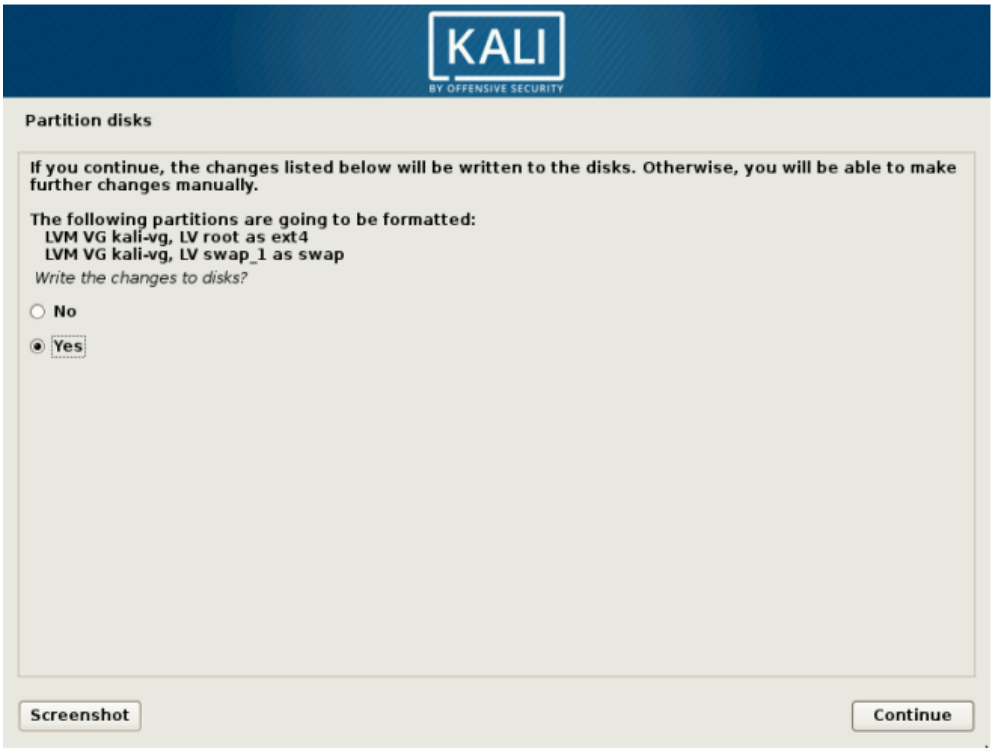

Рисунок 4.25 *Подтверждение форматирования разделов*

Далее процесс установки продолжится обычным образом, что подробно описано в разделе 4.2.1.14, «Настройка диспетчера пакетов (apt)» [page 81].

4.3 Автоматические установки

Установщики Debian и Kali являются довольно модульными: на базовом уровне, они просто выполняют множество скриптов (объединенных в крошечные пакеты под названием udeb—for µdeb или micro-deb) один за другим. Каждый из скриптов опирается на debconf (смотри «Инструмент debconf» [стр 214]), который взаимодействует с вами, пользователем и сохраняет параметры установки. В связи с этим, установщик может быть также автоматизирован через debconf preseeding, функцию, которая позволяет вам предоставлять автоматические ответы на вопросы установки.

4.3.1 Ответы пресидинга

Есть множество различных способов использовать preseed ответы для установщика. И каждый метод имеет свои преимущества и недостатки. В зависимости от того, когда выполняется пресидинг, вопросы, которые ему подвергаются, могут отличаться.

С параметрами загрузки

Вы можете выполнить пресидинг для любого вопроса установщика с параметрами загрузки, которые завершаются в ядре командной строки, доступные через /proc/cmdline. Некоторые начальные загрузчики позволят вам отредактировать эти параметры в интерактивном режиме (что практично для целей тестирования), но если вы хотите, чтобы изменения были сохранены, вам придется изменить конфигурацию загрузчика.

Вы можете напрямую использовать полный идентификатор вопросов debconf (например, debian-installer/language=en), или вы можете использовать аббревиатуры для наиболее распространенных вопросов (например, language=en или hostname=duke). Смотри полный список сокращений в руководстве по установке Debian.

Не существует ограничений на вопросы, которые вы можете подвергнуть пресидингу ввиду того, что параметры загрузки доступны с самого начала процесса установки, и они обрабатываются на очень ранних стадиях. Однако, количество параметров загрузки ограничено 32, и определенное количество из них уже используется по умолчанию. Также очень важно понимать, что изменение параметров начального загрузчика иногда может быть нетривиальным.
В разделе 9.3, «Создание живого пользовательского ISO образа Kali» [стр. 236] вы также узнаете, каким образом изменить Isolinux конфигурации, когда вы создаете ваш собственный ISO образ Kali.

С Preseed файлом в Initrd

Вы можете добавить файл под названием preseed.cfg в root установщика initrd (это тот initrd, который используется для запуска установщика). Обычно это требует восстановления исходного пакета debian-installer для генерации новых версий initrd. Тем не менее, live-build предлагает удобный способ сделать

это, что подробно описано в разделе 9.3 «Создание живого пользовательского ISO образа Kali» [стр. 236].

Этот метод также не имеет никаких ограничений на вопросы, которые могут подвергаться пресидингу, т.к. preseed файл становится доступным сразу же после загрузки. В Kali мы уже используем эту функцию для настройки поведения официального установщика Debian.

С Preseed файлом в загрузочном носителе

Вы можете добавить файл preseed на загрузочный носитель (CD или USB-накопитель); затем выполняется пресидинг, а именно после монтирования носителя, что в свою очередь означает сразу после вопросов о языке и раскладки клавиатуры. Параметр загрузки preseed/file может использоваться для указания местоположения пресидинг файла (например, /cdrom/preseed.cfg при установке с компакт-диска или /hd-media/preseed.cfg при установке с USB- накопителя).
Вы можете не подвергать пресидингу ответы на вопросы о языке и стране, т.к. пресидинг файл загружается немного позже в этом процессе, после загрузки драйверов оборудования. Также большим плюсом является то что, live-build упрощает размещение дополнительного файла в сгенерированных ISO-образах (смотри раздел 9.3, «Создание живого пользовательского ISO образа Kali» [стр. 236]).

С Preseed файлом, загруженным из сети

Вы можете получить доступ к preseed файлу в сети через веб-сервер и сказать установщику скачать preseed файл путем добавления параметра preseed/url=http://serv‘er/preseed.cfg (или с помощью альтернативного url).

Однако, используя этот метод, не забывайте, что для его реализации ваша сеть уже должна быть настроена. В свою очередь, это означает, что связанные с сетью вопросы debconf (в частности имя хоста и имя домена) и все предыдущие вопросы (например, язык и страна) не могут быть запрограммированы с помощью этого метода. Также этот метод чаще всего используется

в сочетании с параметрами загрузки, которые выполняют пресидинг тех конкретных вопросов.

Этот метод пресидинга является самым гибким, т.к. вы можете изменять настройки инсталляции без смены установочного носителя.

Задержка вопросов, связанных с языком, страной и клавиатурой

Для того, чтобы преодолеть ограничение того, что вы не можете подвергать пресидингу вопросы, связанные с языком, страной, и клавиатурой, вы можете добавить параметр загрузки **auto-install/enable=true (или auto=true).** С этой опцией вопросы будут задаваться немного позже в процессе, а именно после того, как сеть будет настроена и, таким образом, после загрузки preseed файла.

Недостатком является то, что первые шаги (в частности, конфигурация сети) всегда будут отображаться на английском языке, и если есть ошибки, пользователю придется работать через экраны, отображаемые полностью на английском языке (с клавиатурой, настроенной в QWERTY).

4.3.2 Создание Preseed файла

Preseed файл является обычным текстовым файлом, в котором каждая строка содержит ответы на один Debconf вопрос. Строка разделяется на четыре поля, разделенные пробелом (пробелом или табуляцией). Например, d-i mirror/ suite string kali-rolling:

- Первое поле обозначает владельца вопроса. Например, “d-i” используется для вопросов, относящихся к установщику. Вы также можете увидеть имя пакета для вопросов, исходящих из пакетов Debian (как в этом примере: atftpd atftpd / use_inetd boolean false).
- Второе поле является идентификатором вопроса.
- Третье поле указан тип вопроса.
- Четвертое и заключительное поле содержат значение

ожидаемого ответа. Обратите внимание, что он должен быть отделен от третьего поля одним пробелом; дополнительные пробельные символы считаются частью значения.

Самый простой способ написать preseed файл - это установить систему вручную. Затем команда **debconf-get- selections –installer** предоставит ответы, которые вы в свою очередь предоставили установщику. Вы можете получить ответы, направленные другим пакетам, с помощью debconf-get-selections. Однако, более чистым решением является запись файла preseed вручную, начиная с примера, а затем просматривая документацию. При таком подходе, лишь те вопросы, ответ по умолчанию на которые необходимо заменить, могут быть подвергнуты пресидингу. Укажите priority=critical параметр загрузки, чтобы поручить Deb-conf задавать критические вопросы и использовать ответы по умолчанию для других.

Приложение к руководству по установке

В приложении руководства по установке Debian, доступном в Интернете, содержится подробная документация по использованию файла preseed. Он также содержит подробный и прокомментированный образец файла, который может служить базой для локальных настроек.

https://www.debian.org/releases/stable/amd64/apb.html
https://www.debian.org/releases/stable/example-preseed.txt

Однако, обратите внимание, что приведенные выше ссылки документируют стабильную версию Debian и что в них Kali используется в тестовой версии, так что вы можете столкнуться с небольшими различиями. Вы также можете ознакомиться с руководством по установке, размещенным на веб-сайте проекта Debian-installer. Оно может быть более современным.
http://d-i.alioth.debian.org/manual/en.amd64/apb.html

4.4 ARM Установки

Kali Linux работает на самых разных устройствах на базе ARM (например, ноутбуки, встроенные компьютеры и платы для разработчиков, но вы не можете использовать традиционный установщик Kali на этих устройствах, поскольку они часто имеют конкретные требования в отношении ядра или конфигурации начального загрузчика. Чтобы сделать эти устройства более доступными для пользователей Kali, Offensive Security разработала сценарии для создания образов дисков[18] , которые готовы для использования с различными ARM устройствами. Offensive Security предоставили подобные образы для загрузки на своем веб-сайте:

https://www.offensive-security.com/kali-linux-arm-images/

Ввиду того, что эти образы общедоступны, ваша задача по установке Kali на ARM устройство значительно упрощена.

Ниже приведены основные шаги:

1. Загрузите образ для своего устройства ARM и убедитесь, что контрольная сумма соответствует той, которая указана на веб-сайте (смотри Раздел 2.1.3 «Проверка целостности и подлинности» [стр. 16] для объяснения того, как это сделать). Обратите внимание, что образы обычно являются xz-сжатыми, таким образом, необходимо распаковать их с помощью unxz.
2. В зависимости от слота расширения хранилища, доступного на вашем конкретном устройстве ARM, приобретите SD-карту, micro SD-карту или модуль eMMC, емкость которого не менее 8 ГБ.
3. Скопируйте загруженный образ на устройство хранения с помощью dd. Это похоже на процесс копирования образа ISO на USB-накопитель (смотри Раздел 2.1.4 «Копирование образа на DVD-ROM или USB-накопитель» [стр. 19]).
4.
```
# dd if=kali-image.img of=/dev/something bs=512k
```
5. Вставьте SD-карту/eMMC в ваше устройство ARM.
6. Загрузите устройство ARM и войдите в него (*пользователь «root», пароль «toor»*). Если у вас нет подключенного экрана, вам нужно будет определить IP-адрес, назначенный через DHCP, и подключиться к этому адресу через SSH. На некоторых

[18]https://github.com/offensive-security/kali-arm-build-scripts

серверах DHCP есть инструменты или веб-интерфейсы, чтобы показать текущие leases. Если у вас нет ничего подобного, используйте сниффер для поиска траффика lease DHCP.
7. Измените пароль root и сгенерируйте новые ключи хоста SSH, особенно, если устройство будет постоянно работать в общедоступной сети! Шаги являются довольно простыми, см. «Создание новых хост ключей SSH» [стр. 111].
8. Наслаждайтесь вашим новым устройством ARM, работающим на Kali Linux!

Специальные случаи и более подробная документация

Эти инструкции носят общий характер, и, хотя они работают для большинства устройств, всегда есть исключения. Например, для Chromebook требуется *режим разработчика (developer mode)*, а на других устройствах требуется специальное нажатие клавиши для загрузки с внешнего носителя.

Поскольку устройства ARM добавляются относительно часто, и их спецификации настолько динамичны, мы не будем описывать конкретные инструкции по установке для различных устройств ARM. Вместо этого обратитесь к выделенному разделу «Kali на ARM» документации веб-сайта Kali для получения более подробной информации о каждом устройстве ARM, поддерживаемом Offensive Security:

http://docs.kali.org/category/kali-on-arm

4.5 Устранение неполадок во время установки

Установщик достаточно надежный, но, тем не менее, вы можете столкнуться с ошибками или же с какими-либо внешними проблемами, такими как: проблемы с сетью, плохие зеркала и недостаточное место на диске. Из-за этого весьма полезно уметь устранять проблемы, возникающие в процессе установки.

Когда в программе установки произойдет сбой, она покажет вам довольно бесполезный экран, такой как тот, который показан на рисунке 4.26, «Сбой процесса установки» [стр. 96].

Рисунок 4.26 *Сбой процесса установки*

На этом этапе хорошо знать, что установщик использует несколько виртуальных консолей: главный экран, который вы видите, запускается либо на пятой консоли (для графического установщика, CTRL + Shift + F5), либо на первой консоли (для текстового установщика, CTRL + Shift + F1). В обоих случаях четвертая консоль (CTRL + Shift + F4) отображает журналы того, что происходит, и обычно вы можете увидеть там более полезное сообщение об ошибке, например, как показано на рисунке 4.27 «Экран журнала установки» [стр. 97], который показывает, что установщику не хватает дискового пространства.

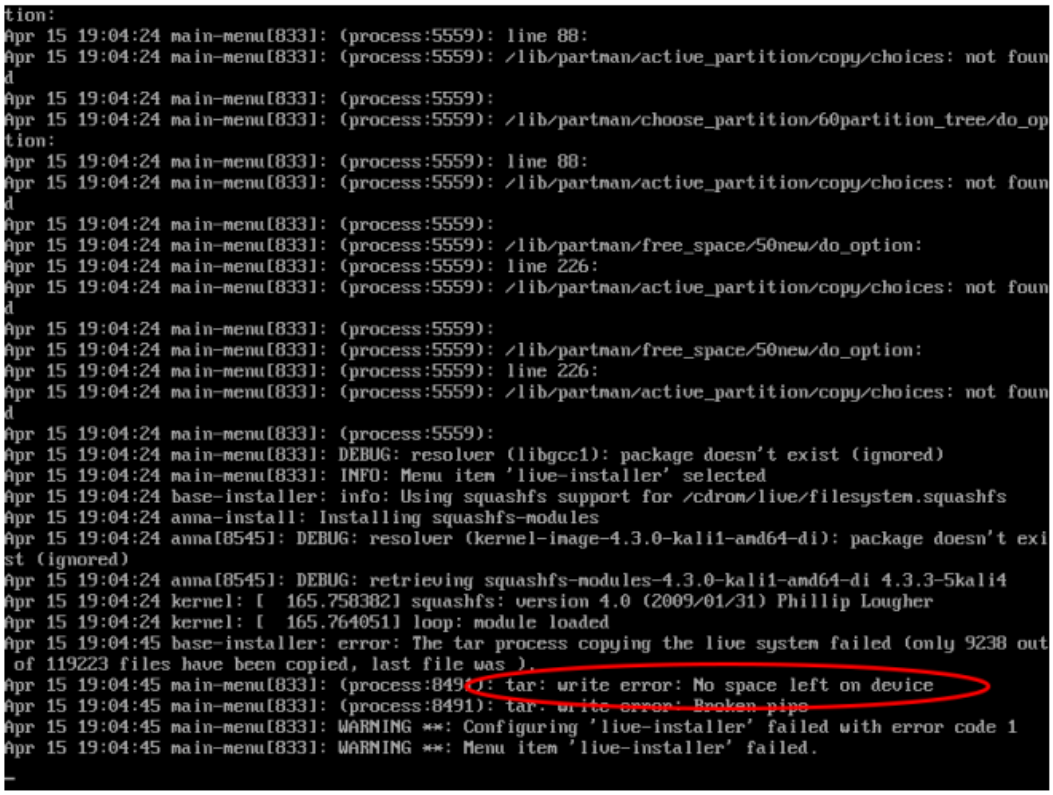

Рисунок 4.27 *Экран журнала установки*

Вторая и третья консоли (CTRL + Shift + F2 и CTRL + Shift + F3 соответственно), хранят оболочки, которые вы можете использовать для более детального изучения текущей ситуации. Большинство инструментов командной строки предоставлены в BusyBox, поэтому набор функций довольно ограничен, но, тем не менее, этого достаточно для того, чтобы разрешить большинство проблем, с которыми вы возможно столкнетесь.

Что может быть сделано с помощью оболочки установщика

Вы можете проверить и изменить базу данных debconf с помощью debconf-get и debconf-set. Эти команды особенно удобны для тестирования значений пресидинга. Вы можете проверить любой файл (например, полный журнал установки, доступный в /var /log/syslog) с помощью cat или других команд. Вы можете редактировать любой файл с помощью nano, включая все файлы, установленные в системе. Корневая файловая система будет смонтирована на /target после завершения этапа разбиение разделов процесса установки. После настройки сетевого доступа вы можете использовать wget и nc (netcat) для извлечения и экспорта данных по сети.

После того, как вы нажмете «Продолжить» на главном экране сбоя установки (Рисунок 4.26 «Сбой процесса установки» [стр. 96]), вы будете возвращены на экран, который в нормальной ситуации вы бы никогда не увидели (главное меню, показанное на рисунке 4.28, «Главное меню установщика» [стр. 98]), который в свою очередь позволяет вам запускать один шаг установки за другим. Если вам удалось устранить проблему, используя доступ к оболочке (примите наши поздравления!), вы можете повторить неудавшийся шаг.

Рисунок 4.28 *Главное меню установщика*

Если вам не удалось разрешить проблему, вы можете захотеть составить отчет об ошибке. Данный отчет должен включать в себя журналы установщика, которые вы можете получить с помощью функции главного меню «Сохранить журналы исправления ошибок» (“Save debug logs”). Она предлагает множество способов экспортировать журналы, как показано на рисунке 4.29, «Сохранить журналы исправления ошибок (1/2)» [стр. 99].

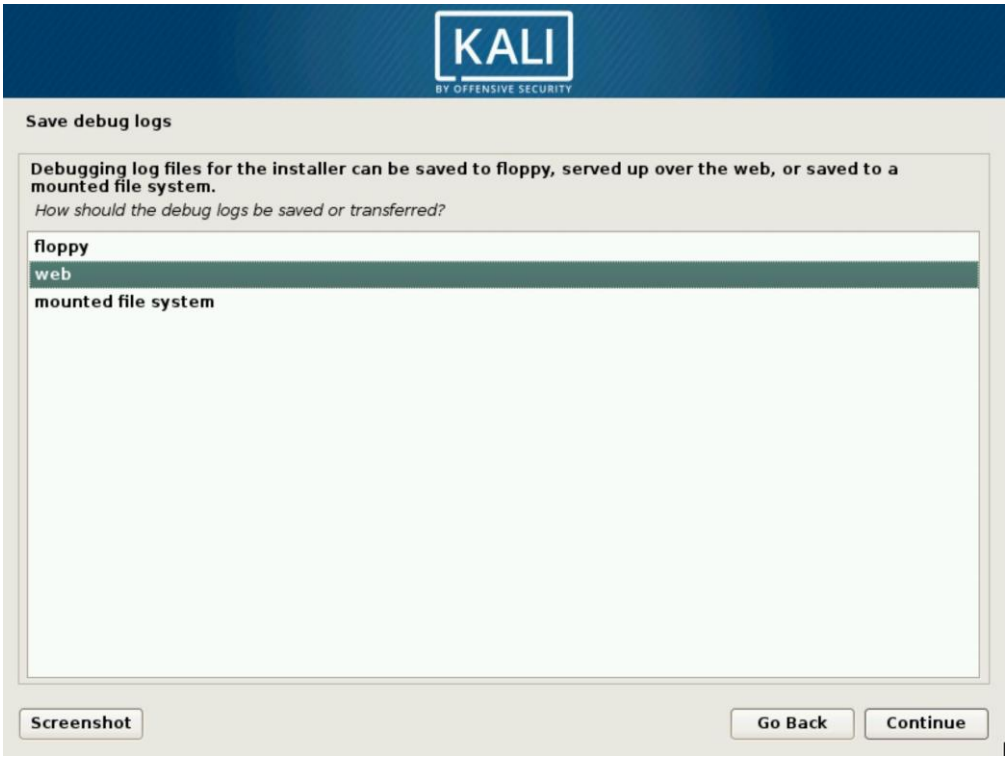

Рисунок 4.29 *Сохранить журнал исправления ошибок (1/2)*

Самый удобный способ и тот, который мы рекомендуем, - позволить установщику запустить веб-сервер, на котором размещаются файлы журнала (рисунок 4.30, «Сохранить журналы исправления ошибок (2/2)» [стр. 100]). Затем вы можете запустить браузер с другого компьютера в той же сети и загрузить все файлы журналов и снимки экрана, которые вы сделали с помощью кнопки «Скриншот», доступной на каждом экране.

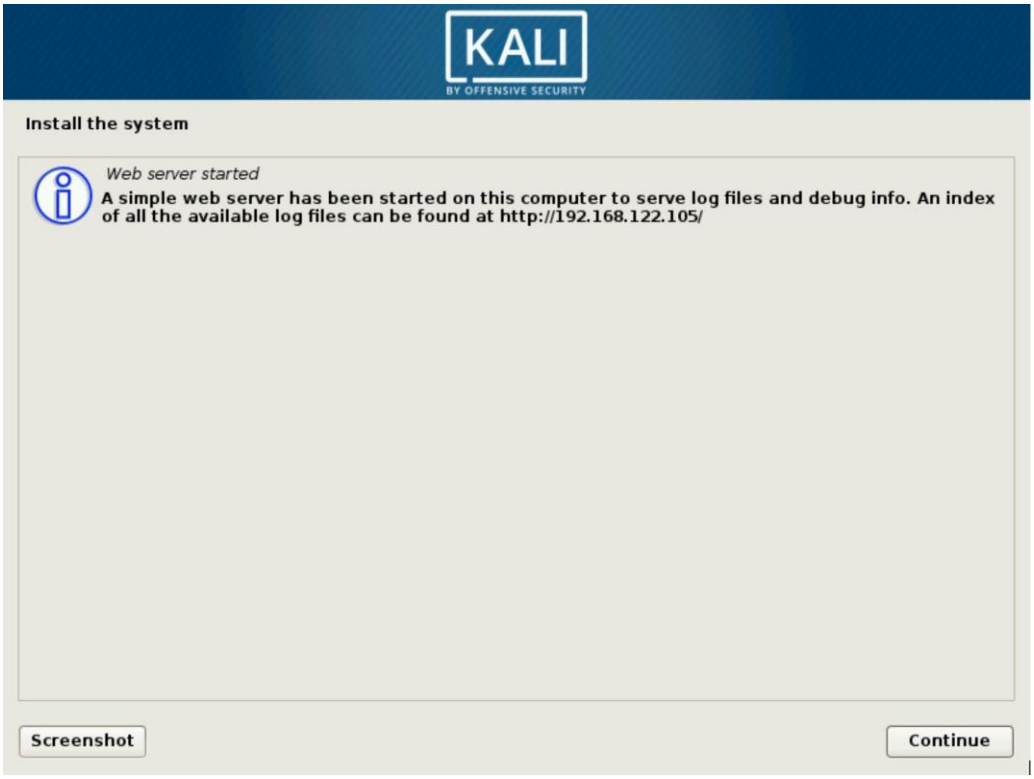

Рисунок 4.30 *Сохранить журнал исправления ошибок (2/2)*

4.6 Подведем итоги

В этой главе мы сосредоточились на процессе установки Kali Linux. Мы обсудили минимальные системные требования Kali Linux, процесс установки для стандартной и полностью зашифрованной системы, пресидинг, который делает возможной автоматическую установку, как установить Kali Linux на различных ARM устройствах и что необходимо делать в редких случаях сбоя установки.

Основные моменты:

- Системные требования Kali Linux варьируются от базового SSH сервера, который не имеет рабочего стола и которому достаточно 128 МБ ОЗУ (рекомендуется 512 МБ) и 2 ГБ дискового пространства, до очень требовательного *kali- linux-full* meta-package, который требует хотя бы 2048 МБ ОЗУ и 20 ГБ пространства на диске. Дополнительно ваша машина должна иметь центральный процессор, который поддерживал бы одну

из следующих архитектур: amd64, i386, armel, armhf, или arm64.
- Kali Linux может быть с легкостью установлен как в качестве основной операционной системы, так и наряду с другими операционными системами с помощью разбиения и изменения начального загрузчика или же в качестве виртуальной машины.
- Чтобы гарантировать конфиденциальность ваших данных, вы можете настроить зашифрованные разделы. Это защитит ваши данные, если ваш ноутбук или жесткий диск будут потеряны или украдены.
- Установщик также может быть автоматизирован с помощью debconf preseeding, функции, которая позволяет вам автоматически отвечать на вопросы установки.
- Preseed файл является обычным текстовым файлом, в котором каждая строчка содержит ответ на один вопрос Debconf. Строчка разделена на четыре поля, которые разделены между собой пробелом (пробелами или табуляцией). Вы можете подвергнуть пресидингу ответы установщику с помощью параметров загрузки, используя preseed файл в initrd, preseed файл в загрузочном устройстве или же preseed файл из сети
- Kali Linux работает на самых разных устройствах на базе ARM, таких как ноутбуки, встроенные вычислители и платы разработчиков. Установка ARM довольно проста. Загрузите нужный вам образ, запишите его на SD-карту, USB-накопитель или встроенный модуль мультимедиа-контроллера (eMMC), подключите его, загрузите устройство ARM, найдите свое устройство в сети, войдите в систему и измените Пароль SSH и ключи хоста SSH.
- Вы можете отладить сбои, возникшие в процессе установки, с помощью виртуальных консолей (доступных путем нажатия CTRL+Shift и одной из функциональных клавиш), debconf-get и debconf-set команд, прочтения /var/log/syslog файла журнала, или же с помощью составления отчета об ошибках, включающего в себя файлы журнала, которые можно извлечь, используя функцию установщика «Сохранить журналы исправления ошибок».

Теперь, когда мы обсудили основы Linux и процесс установки Kali Linux, давайте обсудим процесс настройки системы, чтобы вы могли приступить к настройке Kali в соответствии с вашими потребностями.

Часть 5: Настройка Kali Linux

Содержание:

Ключевые слова главы:

- Сеть;
- Управление пользователями и группами;
- Apache;
- PostgreSQL;
- SSH;

В этой главе мы рассмотрим различные способы настройки Kali Linux. Сначала в разделе 5.1 «Настройка сети» [стр. 104] мы покажем вам, как настроить параметры сети с помощью графической среды и командной строки. В разделе 5.2 «Управление пользователями Unix и группами Unix» [стр. 107] мы поговорим о пользователях и группах, покажем вам, как создавать и изменять учетные записи пользователей, устанавливать пароли, отключать учетные записи и управлять группами. И наконец, мы обсудим службы в разделе 5.3 «Настройка служб» [стр. 109] и объясним, как настроить и поддерживать общие службы, а также сосредоточимся на трех очень важных и конкретных службах: SSH, PostgreSQL и Apache.

5.1 Настройка сети

5.1.1 На рабочем столе с помощью NetworkManager

В стандартной инсталляции рабочего стола у вас уже есть установленный *NetworkManager,* и вы можете управлять им, настраивать его через центр управления GNOME, а также с помощью меню в правом верхнем углу, как показано на Рисунке 5.1 “Экран настройки сети”, [стр. 104].

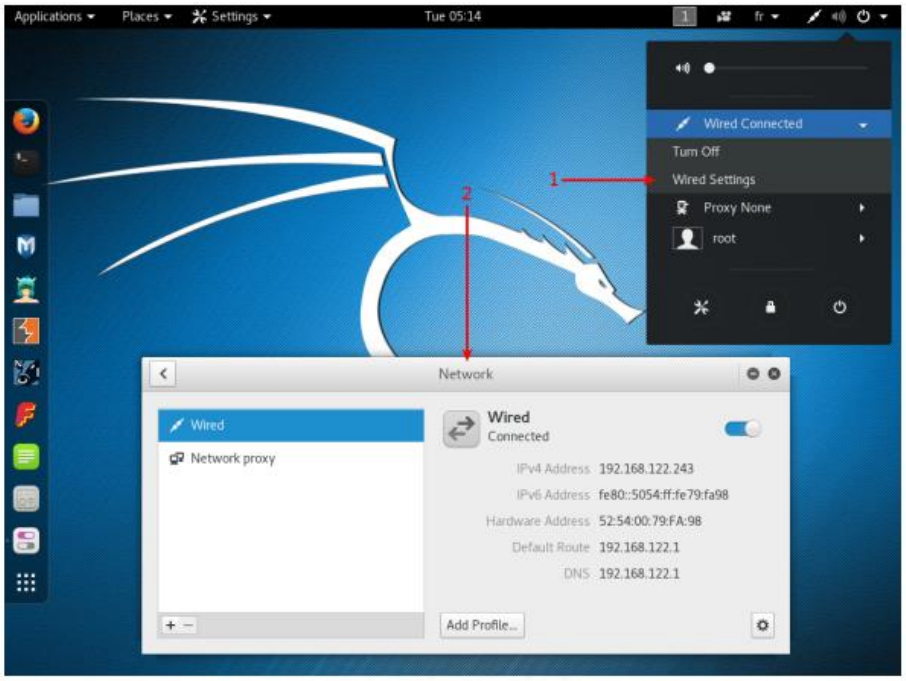

Рисунок 5.1 *Экран настройки сети*

Настройки сети по умолчанию используют DHCP для получения IP-адреса, DNS-сервера и шлюза, но вы можете использовать значок шестеренки в правом нижнем углу, чтобы изменить конфигурацию разными способами (например, установить MAC-адрес, переключиться на статические настройки, включить или отключить IPv6 и добавить дополнительные маршруты). Вы можете создать профили для сохранения нескольких проводных сетевых конфигураций и легко переключаться между ними. Для беспроводных сетей эти настройки автоматически привязываются к их публичному идентификатору (SSID).

NetworkManager также обрабатывает соединения с помощью мобильного широкополосного канала (Wireless Wide Area Network WWAN), а также модемов использующих протокол передачи от точки к точке через Ethernet (point-to-point protocol over ethernet (PPPoE)). И последнее, но не менее важное: он обеспечивает интеграцию со многими типами виртуальных частных сетей (virtual private networks (VPN)) с помощью выделенных модулей: SSH, OpenVPN, Cisco's VPNC, PPTP, Strongswan. Проверьте пакеты network-manager- *; большинство из них не установлены по умолчанию. Обратите внимание, что вам нужны пакеты с суффиксом -gnome, чтобы настроить их через графический интерфейс пользователя.

5.1.2 Через командную строку с помощью Ifupdown

В качестве альтернативы, в тех случаях, когда вы предпочитаете не использовать (или у вас просто нет доступа) графический интерфейс, вы можете настроить сеть с помощью уже установленного пакета *ifupdown,* который включает в себя инструменты ifup и ifdown. Эти инструменты считывают определения из файла конфигурации /etc/network/interfaces и лежат в основе сценария инициализации /etc/init.d/networking, который настраивает сеть во время загрузки.

Каждое сетевое устройство, управляемое ifupdown, может быть деконфигурировано в любое время с помощью *сетевого*

устройства ifdown. Затем вы можете изменить /etc/network/interfaces и вернуть сеть (с новой конфигурацией) с помощью *сетевого устройства* ifup.

Давайте поближе познакомимся с тем, что мы можем вложить в файл конфигурации ifupdown. Существуют две основные директивы: автоматическое *сетевое устройство*, которое говорит *ifupdown* автоматически настроить сетевой интерфейс, как только он становится доступным, и iface *сетевой интерфейс типа inet/inet6* для того, чтобы настраивать данный интерфейс. Например, простая конфигурация DHCP выглядит следующим образом:

```
auto lo
iface lo inet loopback

auto eth0
iface eth0 inet dhcp
```

Обратите внимание, что специальные конфигурации для устройства шлейфа, всегда должны присутствовать в этом файле. Для конфигурации фиксированного IP-адреса вам необходимо предоставить более подробную информацию, такую как IP-адрес, сеть и IP-адрес шлюза:

```
auto eth0
iface eth0 inet static
  address 192.168.0.3
  netmask 255.255.255.0
  broadcast 192.168.0.255
  network 192.168.0.0
  gateway 192.168.0.1
```

Для беспроводного интерфейса, у вас должен быть пакет wpasupplicant (включенный в Kali по умолчанию), который предоставляет множество wpa-* опций, которые в свою очередь могут быть использованы в /etc/network/interfaces. Для получения примеров и дополнительных разъяснений ознакомьтесь с /usr/share/doc/wpasupplicant/README.Debian.gz. Самыми часто используемыми опциями являются wpa-ssid (которая определяет имя беспроводной сети для присоединения) и wpa-psk (которая определяет идентификационную фразу или ключ, защищающий сеть).

```
iface wlan0 inet dhcp
  wpa-ssid MyNetWork
  wpa-psk plaintextsecret
```

5.1.3 Через командную строку с помощью systemd-networkd

Несмотря на то, что *ifupdown* является давно используемым инструментом в Debian, и то, что он до сих пор является стандартом для серверов или других минимальных установок, существует более новый инструмент, который заслуживает нашего внимания: *systemd-networkd. Его интеграция с systemd* инициализирующей системой делает его очень привлекательным. Он не является целевым для дистрибутивов на основе Debian (в отличие от ifupdown) и был разработан как очень маленький, эффективный и относительно простой в настройке, в том случае если вы хорошо понимаете синтаксис файлов systemd. Для вас это будет особенно привлекательным выбором, если вы считаете, что *NetworkManager* является чересчур раздутым и сложно настраиваемым.

Вы настраиваете systemd-networkd путем помещения сетевых файлов (.network files) в директорию /etc/systemd/network/. В качестве альтернативы вы можете использовать /lib/systemd/network/ для пакетированных файлов или /run/systemd/ network/для файлов, сгенерированных во время выполнения программы. Формат этих файлов документируется в systemd. network(5). Раздел Match указывает сетевые интерфейсы, к которым применяется конфигурация. Вы можете указать интерфейс различными способами, например, по адресу управления доступом к среде (media access control (MAC)) или по типу устройства. Раздел Network определяет конфигурацию сети.

Пример 5.1 *Конфигурация на базе DHCP в /etc/systemd/network/80-dhcp.network*

```
[Match]
Name=en*

[Network]
DHCP=yes
```

Пример 5.2 *Статическая конфигурация в /etc/systemd/network/50-static.network*

```
[Match]
Name=enp2s0

[Network]
Address=192.168.0.15/24
Gateway=192.168.0.1
DNS=8.8.8.8
```

Обратите внимание, что system-networkd отключен по умолчанию, поэтому, если вы хотите его использовать, его необходимо включить. Это также зависит от systemd-resolved для правильной интеграции DNS разрешения, что в свою очередь требует от вас замены /etc/resolv.conf символической ссылкой на /run/system/resolve/resolv.conf, которая управляется systemd-resolved.

```
# systemctl enable systemd-networkd
# systemctl enable systemd-resolved
# systemctl start systemd-networkd
# systemctl start systemd-resolved
# ln -sf /run/system/resolve/resolv.conf /etc/resolv.conf
```

Хотя systemd-networkd страдает от определенных ограничений, вроде недостатка интегрированной поддержки беспроводных сетей, вы все равно можете полагаться на ранее существовавшую внешнюю конфигурацию wpa_supplicant для поддержки беспроводной сети. Однако, она особенно полезна в контейнерах и виртуальных машинах и была первоначально разработана для сред, в которых конфигурация сети контейнера зависела от конфигурации сети хоста. В этом сценарии systemd-networkd упрощает управление обеими сторонами последовательно, поддерживая всевозможные виртуальные сетевые устройства, которые могут потребоваться в этом типе сценариев (смотри systemd.netdev(5)).

5.2 Управление Unix пользователями и Unix группами

База данных пользователей и групп Unix состоит из текстовых файлов /etc/passwd (список пользователей), /etc/shadow (зашифрованные пароли пользователей), /etc/group (список групп) и /etc/gshadow (зашифрованные пароли групп). Их форматы документируются в passwd(5), shadow(5), group(5), и gshadow(5) соответственно. Хотя эти файлы могут быть отредактированы вручную с помощью таких инструментов, как vipw и vigr, для выполнения наиболее распространенных операций существуют инструменты более высокого уровня.

5.2.1 Создание пользовательского аккаунта

Несмотря на то, что Kali чаще всего запускается, проводя аутентификации в качестве пользователя root, вам часто может понадобиться создавать непривилегированные учетные записи пользователей по разным причинам, особенно если вы используете Kali в качестве основной операционной системы. Самый простой способ добавить пользователя - это использовать команду adduser, которая требует определенный аргумент: имя пользователя (username) для нового пользователя, которого вы бы хотели создать.

Команда adduser задает несколько вопросов перед созданием учетной записи, но ее использование довольно простое. Его конфигурационный файл /etc/adduser.conf содержит множество интересных настроек. Вы можете, например, определить диапазон идентификаторов пользователей (user identifiers (UID)), который может быть использован для того, чтобы обозначить используют ли пользователи общую группу или нет, определить оболочку по умолчанию и многое другое.

Создание аккаунта запускает популяцию домашних директорий пользователей с содержимым шаблона /etc/skel/. Это предоставляет пользователю набор стандартных директорий и файлов конфигурации.

В некоторых случаях будет очень полезно добавлять пользователя в группу (отличную от основной группы по умолчанию), чтобы предоставить дополнительные разрешения. Например,

пользователь, включенный в группу *sudo*, имеет полные административные права через sudo команду. Подобного результата можно достичь с помощью использования следующей команды adduser *группа пользователя*.

Использование getent для обращения к базе данных пользователя: Команда getent (получение записей (get entries)) проверяет системные базы данных (включая базы данных пользователей и групп) используя соответствующие функции библиотек, которые в свою очередь вызывают модули системы идентификации имен (name service switch (NSS)), настроенные в файле /etc/nsswitch.conf. Данная команда использует один или два аргумента: имя базы данных, которую необходимо проверить, и возможный ключ поиска. Таким образом, команда getent passwd kaliuserl выдаст информацию из пользовательской базы данных о пользователе kaliuserl.

```
root@kali:~# getent passwd kaliuser1
kaliuser1:x:1001:1001:Kali User
    ➥ ,4444,123-867-5309,321-867-5309:/home/kaliuser1:/bin/
    ➥ bash
```

5.2.2 Изменение существующей учетной записи или пароля

Следующие команды разрешают изменения информации, хранящейся в конкретных областях пользовательской базы данных:

- passwd — позволяет обычным пользователям изменять свой пароль, который в свою очередь, обновляет файл /etc/ shadow файл;
- chfn — (Сменить полное имя (CHange Full Name)), зарезервированное для суперпользователя (root), изменяет GECOS, или область ”общая информация”;
- chsh —(Сменить оболочку (CHange SHell)) изменяет оболочку входа пользователя. Однако доступные варианты будут ограничены теми, которые перечислены в /etc/shells; администратор, с другой стороны, не связан этим ограничением и может установить оболочку в любую выбранную программу;

- chage—(CHange AGE) позволяет администратору изменять настройки срока действия пароля, указывая имя пользователя в качестве аргумента или отображая текущие настройки с помощью параметра -I *пользователь*. Кроме того, вы также можете принудительно завершить использование пароля с помощью команды passwd -e *пользователь,* которая заставляет пользователя менять свой пароль при следующем входе в систему.

5.2.3 Блокирование аккаунта

Вам может потребоваться отключить учетную запись (заблокировать пользователя) в качестве дисциплинарной меры в целях расследования или просто в случае длительного или окончательного отсутствия пользователя. Заблокированная учетная запись означает, что пользователь не может войти в систему или получить доступ к машине. Учетная запись остается неповрежденной на машине, и никакие файлы или данные не удаляются; она просто недоступна. Это достигается с помощью команды passwd -l пользователь (lock). Повторное включение учетной записи выполняется аналогичным образом с опцией -u (разблокировка).

5.2.4 Управление Unix группами

Команды addgroup и delgroup добавляют или удаляют группу, соответственно. Команда groupmod изменяет информацию о группе (ее gid или идентификатор). Команда *gpasswdgroup* изменяет пароль для группы, а команда gpasswd -r *группа* удаляет его.

Работа с несколькими группами

Каждый пользователь может быть членом многих групп. Основная группа пользователей по умолчанию создается во время начальной настройки пользователя. По умолчанию каждый файл, созданный пользователем, принадлежит пользователю, а также основной группе пользователя. Это не всегда желательно;

например, когда пользователь должен работать в директории, разделяемой группой, отличной от их основной группы. В этом случае пользователю необходимо изменить группы, используя одну из следующих команд: newgrp, которая запускает новую оболочку или sg, которая просто выполняет команду, используя предоставленную альтернативную группу. Эти команды также позволяют присоединиться к группе, к которой они в настоящее время не принадлежат. Если группа защищена паролем, им необходимо будет предоставить соответствующий пароль перед выполнением команды.

В качестве альтернативы, пользователь может установить setgid bit на директорию, что позволяет файлам, созданным в директории, автоматически принадлежать верной группе. Для получения большей информации, смотри заметку «директория setgid и sticky bit» [page 58].

Команда id отображает текущее состояние пользователя, его личный идентификатор (переменную uid), текущую основную группу (переменную gid) и список групп, к которым они принадлежат (переменная groups).

5.3 Настройка служб

В этом разделе мы рассмотрим службы (иногда называемые демонами) или программы, которые работают в фоновом режиме и выполняют различные функции для системы. Мы начнем с обсуждения конфигурационных файлов и далее затронем тему касательно того, как работают некоторые важные службы (такие как SSH, Post-greSQL и Apache) и как их можно настроить.

5.3.1 Настройка конкретной программы

Если вы хотите настроить неизвестный пакет, вы должны действовать поэтапно. Сначала, вам следует прочитать документацию. предоставленную непосредственно эксплуатационником. Debian файл /usr/share/doc/***package***/README - довольно таки хорошее место

для начала. Этот файл чаще всего будет содержать информацию о пакете, включая ссылки, которые могут перенаправить вас на дополнительную документацию. Вы всегда сэкономите себе много времени и избегнете утомительной работы, прочитав этот файл с самого начала, поскольку он часто довольно детально описывает наиболее распространенные ошибки и решения большинства распространенных проблем.

Далее вам нужно ознакомиться с официальной документацией данного программного обеспечения. Мы отсылаем вас к разделу 6.1, «Источники документации» [стр. 124] для получения дополнительных советов о том, каким образом лучше искать различные источники документации. Команда dpkg -L *пакет* предоставляет список файлов, включенных в пакет; вы можете быстро определить имеющуюся документацию (а также файлы конфигурации, расположенные в /etc/). Также, команда dpkg -s *пакет* отображает метаданные пакета и показывает любые возможные рекомендуемые или предлагаемые пакеты; там вы можете найти документацию или, возможно, утилиту, которая облегчит настройку программного обеспечения.

Наконец, файлы конфигурации часто самодокументируются многими пояснительными комментариями, подробно описывающими различные возможные значения для каждого параметра конфигурации. В некоторых случаях вы можете запустить программное обеспечение путем раскоментирования одной строки в файле конфигурации. В других случаях примеры файлов конфигурации содержатся в каталоге /usr/share/doc /package/examptes/. Они могут служить основой для вашего собственного файла конфигурации.

5.3.2 Настройка SSH для удаленного входа

SSH позволяет вам удаленно входить в систему, передавать файлы или выполнять команды. Это стандартный инструмент (ssh) и служба (sshd) для удаленного подключения к машинам.

В то время как ***openssh-server*** пакет установлен по умолчанию, служба ***SSH*** отключена по умолчанию и, таким образом, не

запускается при загрузке системы. Вы можете вручную запустить службу с помощью команды systemctt start ssh или выставить, чтобы служба запускалась при загрузке с помощью команды systemctt enable ssh.

Служба SSH имеет относительно нормальную конфигурацию по умолчанию, но, учитывая её мощные возможности и восприимчивый характер, будет очень хорошо узнать, что вы сможете сделать с её файлом конфигураций, /etc/ssh/sshd_ config. Все параметры задокументированы в sshd_config (5).

Конфигурация по умолчанию не разрешает вход для пользователя с помощью пароля, что означает, что вы сначала должны установить SSH ключ с помощью ssh-keygen. Вы можете расширить это право на всех пользователей, установив PasswordAuthentication на no, или вы можете снять это ограничение, изменив PermitRootLogin на yes (вместо стандартного запрета-пароля).

Служба SSH прослушивает по умолчанию на порту 22, но вы можете изменить это с помощью директивы Port. Чтобы применить новые настройки, вы должны запустить команду systemctt reload ssh.

Создание новых ключей SSH Host

Каждый SSH сервер имеет свои собственные криптографические ключи, которые называются “SSH host keys” и хранятся в /etc/ssh/ssh_host_*. Они должны быть приватными, если вам требуется конфиденциальность и не могут использоваться на нескольких машинах.

При установке вашей системы путем копирования полного образа диска (вместо использования debian- installer) образ может содержать предварительно сгенерированные SSH ключи хоста, которые вы должны заменить на вновь сгенерированные ключи. Возможно, образ также содержит и пароль root по умолчанию, который вы захотите сбросить одновременно. Вы можете сделать все это с помощью следующих команд:

```
# passwd
[...]
# rm /etc/ssh/ssh_host_*
# dpkg-reconfigure openssh-server
# service ssh restart
```

5.3.3 Настройка PostgreSQL баз данных

PostgreSQL является сервером базы данных. Он редко бывает полезен сам по себе, но используется многими другими службами для хранения данных. Эти службы обычно получают доступ к серверу базы данных через сеть и требуют учетные данные для аутентификация, чтобы иметь возможность подключиться. Таким образом, для настройки этих служб требуется создание баз данных PostgreSQL и учетных записей пользователей с соответствующими правами в базе данных. Чтобы это сделать, нам нужно, чтобы служба была запущена, поэтому давайте начнем с команды systemctt start postgresqt.

Поддержка множества версий PostgreSQL

Пакет PostgreSQL позволяет совместно устанавливать несколько версий сервера базы данных. Также возможно обрабатывать несколько кластеров (кластер представляет собой набор баз данных, обслуживаемых одним и тем же постмастером). Чтобы достичь этого файлы конфигурации должны храниться в файле */etc/postgresql/version/cluster-name/.*

Для того чтобы кластеры запускались бок о бок, каждому новому кластеру присваивается следующий номер доступного порта (обычно 5433 для второго кластера). Файл postgresql.service представляет собой пустую оболочку, что упрощает работу со всеми кластерами, поскольку каждый кластер имеет свой собственный блок (postgresql @ version- cluster .service).

Тип подключения и аутентификация клиента

По умолчанию PostgreSQL прослушивает входящие соединения двумя способами: на TCP-порту 5432 интерфейса локального хоста и на файловом сокете /var/run/postgresql/.s.PGSQL.5432. Это может быть сконфигурировано в postgresql.conf с различными директивами: listen_addresses для адресов для прослушивания, port для TCP-порта и unix_socket_directories для определения директории, в которой будут созданы файловые сокеты.

В зависимости от того, каким образом они подключаются, клиенты проходят аутентификацию различными способами. Файл конфигурации pg_hba.conf определяет, кому разрешено подключаться к каждому сокету и как они аутентифицируются. По умолчанию соединения в файловом сокете используют учетную запись пользователя Unix в качестве имени пользователя PosgreSQL и предполагают, что дальнейшая проверка подлинности не требуется. В соединении TCP PostgreSQL требует от пользователя аутентификации с помощью логина и пароля (хотя это не Unix имя пользователя и пароль, а скорее учетные данные, регулируемые самой PostgreSQL itself). Пользователь postgres является особенным и обладает полными административными правами, которые распространяются на все базы данных. Мы будем использовать этот идентификатор для создания новых пользователей и новых баз данных.

Создание пользователей и баз данных

Команда createuser добавляет нового пользователя, а dropuser удаляет его. Аналогично, команда createdb добавляет новую базу данных, а dropdb удаляет ее. Каждая из этих команд имеет свои собственные страницы руководства, но мы обсудим некоторые из вариантов здесь. Каждая команда действует на кластере по умолчанию (работает на порту 5432), но вы можете указать —port=*port* для того, чтобы изменить пользователей и базы данных альтернативного кластера. Эти команды должны подключаться к серверу PostgreSQL, чтобы выполнять поставленные перед ними задачи должным образом, и также они должны быть аутентифицированы в качестве пользователя с достаточными полномочиями для выполнения указанной операции. Самый простой способ добиться этого - использовать учетную запись postgres Unix и подключиться к файловому сокету:

```
# su - postgres
$ createuser -P king_phisher
Enter password for new role:
Enter it again:
$ createdb -T template0 -E UTF-8 -O king_phisher king_phisher
$ exit
```

В этом примере опция –P просит createuser запросить пароль сразу же после создания нового пользователя king_phisher. Рассматривая команду createdb, -O определяет пользователя,

владеющего новой базой данных (которая, таким образом, имеет полные права на создание таблиц и предоставление разрешений и т. д.). Мы также хотим использовать строки Unicode, поэтому мы добавляем параметр -E UTF-8 для установки кодировки, что, в свою очередь, требует от нас использовать параметр -T для выбора другого шаблона базы данных.

Теперь мы можем проверить, что мы подключились к базе данных через сокет, прослушивающий localhost (-h localhost), в качестве пользователя king_phisher (-U king_phisher):

```
# psql -h localhost -U king_phisher king_phisher
Password for user king_phisher:
psql (9.5.2)
SSL connection (protocol: TLSv1.2, cipher: ECDHE-RSA-AES256-GCM-SHA384, bits: 256,
    ➥ compression: off)
Type "help" for help.

king_phisher=>
```

Как вы можете видеть, соединение было проведено успешно.

Управление кластерами PostgreSQL

Во-первых, стоит отметить, что понятие «кластер PostgreSQL» является особенным дополнением Debian и что вы не найдете ссылки на этот термин в официальной документации PostgreSQL. С точки зрения инструментов PostgreSQL такой кластер представляет собой всего лишь экземпляр сервера базы данных, работающего на определенном порту.

Тем не менее, пакет Debian *postgresql-common* предоставляет несколько инструментов для управления такими кластерами: pg_createctuster, pg_dropctuster, pg_cttctuster, pg_upgradecluster, pg_renamectuster и pg_tsctusters. Мы не будем описывать все эти инструменты здесь, но вы можете обратиться к их соответствующим страницам руководства для получения дополнительной информации.

Вы должны знать, что при установке новой версии PostgreSQL в вашей системе, она создаст новый кластер, который будет работать на следующем порту (обычно 5433), и вы будете продолжать использовать старую версию до тех пор, пока вы не перенесите свои базы данных из старого кластера на новый.

Вы можете получить список всех кластеров и их статус с помощью pg_tsctusters. Более важно то, что вы можете автоматизировать перенос вашего кластера в последнюю версию PostgreSQL с помощью команды pg_upgradecluster *старая-версия имя-кластера* (*old-version cluster-name).* Для того чтобы эта процедура увенчалась успехом, вы должны сначала удалить (пустой) кластер, который был создан для новой версии (с помощью команды pg_dropctuster *новая-версия имя-кластера* (*new-version cluster-name*)). Старый кластер не выбрасывается процессу, но он также не будет запущен автоматически. Вы можете сбросить его, как только убедитесь в том, что обновленный кластер работает должным образом.

5.3.4 Настройка Apache

Типичная установка Kali Linux включает в себя веб-сервер Apache, предоставляемый пакетом *apache2.* Будучи сетевой службой, он по умолчанию отключен. Вы можете запустить его вручную с помощью systemctt start apache2.

Поскольку все большее количество приложений распространяются как веб-приложения, очень важно иметь базовые знания об Apache для того, чтобы размещать эти приложения, будь то для локального использования или для их доступности по сети.

Apache является модульным сервером, и многие функции реализуются внешними модулями, которые загружаются основной программой во время его инициализации. Конфигурация по умолчанию включает только самые распространенные модули, но включение новых модулей легко выполняется с помощью запуска a2enmod *модуля*. Также вы можете использовать a2dismod *модуль* для отключения модуля. Эти программы на самом деле только создают (или удаляют) символьные ссылки в /etc/apache2/mods-enabted/, указывая на фактические файлы (хранящиеся в /etc/apache2/mods-avaitabte/).

Существует много модулей, но два из них заслуживают первоначального рассмотрения: PHP и SSL. Веб-приложения,

написанные на PHP, выполняются веб-сервером Apache с помощью выделенного модуля, предоставляемого пакетом *libapache-mod-php*, и его установка автоматически включает модуль.

Apache 2.4 включает в себя модуль SSL, который необходим для обеспечения безопасного HTTP (HTTPS). Сначала его нужно включить с помощью a2enmod ssl, затем в файлы конфигурации должны быть добавлены необходимые директивы. Пример конфигурации представлен в /etc/apache2/sites-available/default-ssl.conf.

Также смотри:

http://httpd.apache.Org/docs/2.4/mod/mod_ssl.html

Полный список стандартных Apache модулей может быть найден онлайн на

http://httpd.apache.org/docs/2.4/mod/index.html

Исходя из начальных настроек по умолчанию веб-сервер прослушивает порт 80 (как настроено в / etc / apache2 / ports.conf) и по умолчанию загружает страницы из каталога / var / www / html / (как указано в/etc/apache2/sites-enabled/000-default.conf).

Настройка виртуальных хостов

Виртуальный хост является дополнительной идентификацией для веб-сервера. Один и тот же процесс Apache может обслуживать несколько веб-сайтов (например, www.kali.org и www.offensive-security.com), поскольку HTTP-запросы включают в себя как имя запрашиваемого веб-сайта, так и локальную часть URL (эта функция называется *name-based virtual hosts*).
Конфигурация по умолчанию для Apache 2 включает виртуальные хосты на основе имени. Кроме того, виртуальный хост по умолчанию определяется в файле /etc/apache2/sites-enabled/000-default.conf; этот виртуальный хост будет использоваться, если не найден хост, соответствующий запросу, отправленному клиентом.

Важно знать

Запросы относительно неизвестных виртуальных хостов всегда будут обслуживаться первым определенным виртуальным хостом, поэтому пакет отправляет 000-default. conf, файл конфигурации, который сортирует первый среди всех других файлов, которые вы могли бы создать.

Затем каждый виртуальный хост описывается файлом, хранящимся в /etc/apache2/sites-available/. Обычно файл называется именем хоста сайта, за которым следует суффикс .conf (например: www.example.com.conf). Затем вы можете включить новый виртуальный хост с помощью a2ensite www.example. ком. Ниже приведена минимальная конфигурация виртуального хоста для веб-сайта, файлы которого хранятся в /srv/www. example.com/www/ (определяется с помощью параметра DocumentRoot):

```
<VirtualHost *:80>
ServerName www.example.com
ServerAlias example.com
DocumentRoot /srv/www.example.com/www
</VirtualHost>
```

Вы также можете рассмотреть возможность добавления директив CustomLog и ErrorLog для настройки Apache для вывода журналов в файлы, которые предназначены для виртуального хоста.

Общие директивы

В этом разделе кратко рассматриваются некоторые из обычно используемых конфигурационных настроек Apache.

Основной файл конфигурации обычно включает несколько блоков Directory; они позволяют указывать разные типы поведения для сервера в зависимости от местоположения файла, который будет обслуживаться. Такой блок обычно включает в себя директивы AllowOverride и Options:

```
<Directory /var/www>
Options Includes FollowSymLinks
AllowOverride All
DirectoryIndex index.php index.html index.htm
</Directory>
```

Директива DirectoryIndex содержит список файлов, которые нужно попробовать, когда клиентский запрос соответствует директиве. Первый существующий файл в списке используется и отправляется как ответ.

Директива Options следует список параметров для включения. Значение None отключает все параметры; соответственно, All включает их все, кроме MultiViews. Доступные опции включают в себя:

1. ExecCGI — означает, что CGI скрипты могут быть выполнены;
2. FollowSymLinks — сообщает серверу, что символьные ссылки работают и, что ответ должен содержать контент подобной целевой ссылки;
3. SymLinksIfOwnerMatch — также сообщает серверу следовать символьным ссылкам, но только тогда, когда ссылка и ее цель имеют одного и того же владельца;
4. Includes — включает*Server Side Includes* (SSI). Это директивы, встроенные в HTML страницы, которые моментально выполняются для каждого запроса;
5. Indexes — сообщает серверу о списке содержимого каталога, если HTTP-запрос, отправленный клиентом, указывает на каталог без индексного файла (то есть, когда в этом каталоге не существует файлов, упомянутых директивой DirectoryIndex);
6. MultiViews — разрешает согласование контента; это может быть использовано сервером для возврата веб-страницы, соответствующей предпочитаемому языку, как это указано в браузере.

Требование аутентификации В некоторых случаях доступ к части веб-сайта должен быть изменен, поэтому доступ к содержимому предоставляется только легальным пользователям, которые предоставляют имя пользователя и пароль.

Файл .htaccess содержит директивы конфигурации Apache, которые применяются каждый раз, когда запрос обрабатывает элемент из директории, в котором хранится файл .htaccess. Эти директивы являются рекурсивными, расширяя область действия на все суб-директории.

Большинство директив, которые могут выполняться в блоке Directory, также являются легальными в файле .htaccess. В директиве AllowOverride перечислены все параметры, которые можно включить или отключить по пути .htaccess. Обычным использованием этой опции является ограничение ExecCGI, таким образом, чтобы администратор выбирал, какие пользователи могут запускать программы под идентификатором веб-сервера (пользователь www-data).

Пример 5.3 *Файл .htaccess требует аутентификацию*

```
Require valid-user
AuthName "Private directory"
AuthType Basic
AuthUserFile /etc/apache2/authfiles/htpasswd-private
```

Базовая аутентификация не предоставляет достаточную безопасность

Система аутентификации, используемая в приведенном выше примере (Basic), имеет минимальную степень безопасности поскольку пароль отправляется в доступном и открытом тексте (он только кодируется как *base64,* что является простой кодировкой, а не методом шифрования). Следует также отметить, что документы, защищенные этим механизмом, также проходят через сеть в довольно открытом виде. Если важна безопасность, весь HTTP-сеанс должен быть зашифрован с помощью Transport Layer Sequence (TLS).

Файл /etc/apache2/authfiles/htpasswd-private содержит список пользователей и паролей; его обычно обрабатывают с помощью команды htpasswd. Например, для добавления пользователя или изменения пароля используется следующая команда:

```
# htpasswd /etc/apache2/authfiles/htpasswd-private user
New password:
Re-type new password:
Adding password for user user
```

Ограничение доступа Директива Require directive управляет ограничениями доступа для директории (и её суб-директорий, рекурсивно).

Эту директиву можно использовать для ограничения доступа по многим критериям; мы остановимся на описании ограничения доступа на основе IP-адреса клиента, но его можно сделать гораздо более мощным, особенно если несколько директив Require согласованы в блоке RequireAll.
Например, вы можете ограничить доступ к локальной сети с помощью следующей директивы:

```
Require ip 192.168.0.0/16
```

5.4 Управление службами

Kali использует systemd как свою инициализирующую систему, которая не только отвечает за последовательность загрузки, но также постоянно выступает в качестве полнофункционального менеджера служб, который запускает и контролирует службы.

systemd можно запрашивать и контролировать с помощью systemctl. Без каких-либо аргументов он запускает команду systemctl list-units, которая выводит список активных *структурных элементов*. Если вы запустите systemctl status, на выходе будет показан иерархический обзор работающих служб. Сравнивая оба выхода, вы сразу видите, что существует несколько видов структурных элементов, и что службы являются лишь одним из них.

Каждая служба представлена *служебным структурным элементом*, который описывается служебным файлом, обычно отправленным в / lib / systemd / system / (или / run / systemd / system /, или / etc / systemd / system /; они перечислены путем увеличения порядок важности, где последний является самым важным).

Каждый из них может быть изменен другим файлом *имя-службы* .service.d/*.conf в том же наборе директорий. Эти файлы являются простыми текстовыми файлами, чей формат вдохновлен хорошо известными файлами “*.ini” в Microsoft Windows, с парами *key* = *value* сгруппированными между *[section]* заголовками. Здесь мы видим пример служебного файла для /lib/ systemd/system/ssh.service:

```
[Unit]
Description=OpenBSD Secure Shell server
After=network.target auditd.service
ConditionPathExists=!/etc/ssh/sshd_not_to_be_run

[Service]
EnvironmentFile=-/etc/default/ssh
ExecStart=/usr/sbin/sshd -D $SSHD_OPTS
ExecReload=/bin/kill -HUP $MAINPID
KillMode=process
Restart=on-failure
RestartPreventExitStatus=255
Type=notify

[Install]
WantedBy=multi-user.target
Alias=sshd.service
```

Целевые структурные элементы являются частью дизайна systemd. Они представляют желаемое состояние, которое вы хотите достичь исходя из активированных элементов (что означает работающую службу в отношении служебного элемента). Они существуют главным образом как способ группировки зависимостей от других элементов. Когда система запускается, она позволяет элементам, требуемым для достижения default.target (которая является символьной ссылкой на graphic.target и, в свою очередь, зависит от multi-user.target). Таким образом, все зависимости этих целей активируются во время загрузки.

Подобные зависимости выражаются с помощью директивы Wants на целевом элементе. Но вам не нужно редактировать целевой элемент для добавления новых зависимостей, вы также можете создать символьную ссылку, указывающую зависимый элемент в директории /etc/systemd/system/target-name .target.wants/. И это как раз является именно тем, что делает systemctl enable foo.service. Когда вы включаете службу, вы говорите systemd добавить зависимость от целей, указанных в записи WantedBy

раздела [Install] файла служебного элемента. И наоборот, systemctl disable foo.service сбрасывает ту же символьную ссылку и, следовательно, зависимость.

Команды включения и выключения ничего не меняют в отношении текущего состояния служб. Они влияют только на то, что произойдет при следующей загрузке. Если вы хотите немедленно запустить службу, вы должны выполнить команду systemctl start foo.service. И наоборот, вы можете остановить её с помощью команды systemctl stop foo.service. Вы также можете проверить текущий статус службы, используя команду systemctl status foo.service, которая очень кстати включает в себя последние строки сопутствующего журнала. После изменения конфигурации службы вы можете перезагрузить или перезапустить её: эти операции выполняются с помощью with systemctl reload foo.service и systemctl restart foo. service соответственно.

```
# systemctl status postgresql
● postgresql.service - PostgreSQL RDBMS
   Loaded: loaded (/lib/systemd/system/postgresql.service; disabled; vendor preset:
       ➥ disabled)
   Active: inactive (dead)
# ls -al /etc/systemd/system/multi-user.target.wants/postgresql.service
ls: cannot access '/etc/systemd/system/multi-user.target.wants/postgresql.service': No
    ➥ such file or directory
# systemctl enable postgresql
[...]
# ls -al /etc/systemd/system/multi-user.target.wants/postgresql.service
lrwxrwxrwx 1 root root 38 Apr 21 16:21 /etc/systemd/system/multi-user.target.wants/
    ➥ postgresql.service -> /lib/systemd/system/postgresql.service
# systemctl status postgresql
● postgresql.service - PostgreSQL RDBMS
   Loaded: loaded (/lib/systemd/system/postgresql.service; enabled; vendor preset:
       ➥ disabled)
   Active: inactive (dead)
# systemctl start postgresql
# systemctl status postgresql
● postgresql.service - PostgreSQL RDBMS
   Loaded: loaded (/lib/systemd/system/postgresql.service; enabled; vendor preset:
       ➥ disabled)
   Active: active (exited) since Thu 2016-04-21 16:22:29 EDT; 2s ago
  Process: 6355 ExecStart=/bin/true (code=exited, status=0/SUCCESS)
 Main PID: 6355 (code=exited, status=0/SUCCESS)

Apr 21 16:22:29 kali-rolling systemd[1]: Starting PostgreSQL RDBMS...
Apr 21 16:22:29 kali-rolling systemd[1]: Started PostgreSQL RDBMS.
```

5.5 Подведем итоги

В этой главе мы узнали, как настроить Kali Linux. Мы настроили параметры сети, поговорили о пользователях и группах и

обсудили, как создавать и изменять учетные записи пользователей, устанавливать пароли, отключать учетные записи и управлять группами. Наконец, мы затронули тему о службах и объяснили, как настраивать и поддерживать общие службы, в частности SSH, PostgreSQL и Apache.

Основные моменты:

- При обычной установке через интерфейс рабочего стола у вас уже установлен NetworkManager, и его можно контролировать и настраивать через центр управления GNOME, а также с помощью меню, находящегося в правом верхнем углу;
- Вы можете настроить сеть с помощью командной строки с помощью инструментов ifup и ifdown, которые черпают свои инструкции из /etc/network/interfaces файла конфигурации. Даже более новый инструмент *systemd-networkd* работает с инициирующей системой *systemd;*
- По умолчанию база данных пользователей и групп Unix состоит из текстовых файлов /etc/passwd (список пользователей), /etc/shadow (зашифрованные пароли пользователей), /etc/group (список групп), and /etc/gshadow (зашифрованные пароли групп);
- Вы можете использовать команду getent, чтобы ознакомиться с пользовательской базой данных и другими системными базами данных;
- Команда adduser задает несколько вопросов перед созданием учетной записи, но является прямым способом создания новой учетной записи пользователя;
- Несколько команд могут быть использованы для изменения некоторых областей в пользовательской базе данных: passwd (изменить пароль), chfn (изменить полное имя и поле GECOS, или общее информационное поле), chsh (изменить оболочку входа), chage (изменить «возраст» пароля), и passwd -e *пользователь* (заставляет пользователя менять свой пароль при следующем входе в систему);
- Каждый пользователь может быть членом одной или нескольких групп. Для изменения идентификатора группы можно использовать несколько команд: newgrp изменяет текущий идентификатор группы, sg выполняет команду, используя предоставленную альтернативную группу, setgid bit может быть помещен в директорию, заставляя файлы, созданные в этом

каталоге, автоматически принадлежать к нужной группе. Кроме того, команда id отображает текущее состояние пользователя, включая список членов своей группы;
- Вы можете вручную запустить SSH с помощью systemctl start ssh или включить его на постоянной основе с помощью systemctl enable ssh. Конфигурация по умолчанию отключает пароли для аутентификации пользователя root, что означает, что вы должны сначала настроить SSH-ключи с помощью ssh-keygen;
- PostgreSQL - это сервер базы данных. Он редко бывает полезен сам по себе, но используется многими другими службами для хранения данных;
- Обычная установка Kali Linux включает в себя веб-сервер Apache, предоставляемый пакетом apache2. Будучи сетевой службой, он по умолчанию отключен. Вы можете вручную запустить его с помощью systemctl start apache2;
- Исходя из настроек по умолчанию, Apache прослушивает порт 80 (как настроено в /etc/apache2/ports.conf) и по умолчанию использует страницы из директории /var/www/html/ (как указано в /etc/apache2/sites-enabled/000-default.conf).

Теперь, когда мы рассмотрели основы Linux и обсудили с вами установку и настройку Kali Linux, давайте коснемся вопроса, как устранить проблемы, которые могут возникнуть во время использования Kali и научим вас некоторым инструментам и трюкам для того, чтобы вы могли создавать резервные копии и беспрепятственно запускать систему в случае возникновения каких либо проблем.

Часть 6: Получение помощи

Содержание:

Ключевые слова главы:

- Документация;
- Форумы;
- Канал IRC;
- Отчет об ошибке;

Независимо от того, сколько лет опыта у вас есть, нет сомнений в том, что рано или поздно вы столкнетесь с проблемой. Решая возникшую проблему, первое, что необходимо сделать - это осознать ее суть, а затем использовать различные ресурсы для поиска решения или работы.

В этой главе мы рассмотрим различные доступные источники информации и обсудим лучшие стратегии поиска необходимой вам помощи или решения проблемы, с которой вы можете столкнуться. Мы также познакомим вас с некоторыми доступными ресурсами Сообщества Kali Linux, включая веб-форумы и систему групповых дискуссий в интернете (Internet Relay Chat (IRC)). Наконец, мы познакомим вас с таким понятием? как отчет об ошибках, и покажем вам, как использовать системы регистрации ошибок для устранения неполадок и разработки стратегий, которые помогут вам создать собственный отчет об ошибках, чтобы можно было быстро и эффективно обрабатывать недокументированные проблемы.

6.1 Источники документации

Прежде чем вы сможете понять, что происходит на самом деле, когда у вас возникает проблема, вам необходимо осознать теоретическую роль, которую играет каждая из программ, вовлеченных в эту проблему. Одним из самых лучших способов сделать это является просмотр документации программы. Давайте начнем наше обсуждение с того места, где вы точно сможете найти данную документацию, т.к. она чаще всего является разбросанной по разным местам.

Как избежать RTFM ответов

Данная аббревиатура расшифровывается как «прочти долбанную инструкцию» (“read the f***ing manual,”), однако, она также может быть расшифрована в более дружественной манере «прочитайте точное руководство» (“read the fine manual.”). Эта фраза иногда используется в (кратких) ответах на вопросы новичков. Безусловно, это является довольно резким, и вызывает

определенное раздражение из-за того, что вопрос, задается тем, кто даже не удосужился прочитать основную документацию. Некоторые говорят, что этот классический ответ лучше, чем вообще никакого ответа, поскольку это хотя бы намекает, что ответ лежит в документации.

Когда вы задаете подобные вопросы, не обязательно принимать близко к сердцу случайные ответы RTFM, но вы должны сделать все возможное, чтобы хотя бы показать, что вы потратили время на изучение темы, прежде чем публиковать вопрос; укажите источники, с которыми вы уже ознакомились, и описывайте различные шаги, которые вы лично проделали в процессе поиска информации. Это, безусловно, займет много времени, но тем не менее покажет, что вы не ленивы и действительно прикладываете определенные усилия для поиска знаний. Следуя рекомендациям Эрика Раймонда, вы сможете избежать наиболее распространенных ошибок и получить действительно полезные ответы.

http://catb.org/~esr/faqs/smart-questions.html

6.1.1 Страницы руководства

Страницы руководства, несмотря на определенную сжатость в стиле, содержат действительно необходимую и полезную информацию. Для того чтобы просмотреть страницу руководства, просто введите man *manual-page.* Страница руководства (*manual-page*) обычно совпадает с именем команды. Например, для того, чтобы узнать больше о возможных опциях команды cp, вам нужно будет ввести man cp в командной строке.

Man страницы - это не только документы программ, доступные из командной строки, но также файлы конфигурации, системные вызовы, функции библиотеки C и т. д. Иногда имена могут вступать в противоречия.

Например, команда чтения оболочки имеет то же самое название, что и команда системного вызова чтения. Именно поэтому

страницы руководства организованы в следующие пронумерованные разделы:

1. Команды, которые могут быть выполнены из командной строки;
2. Системные вызовы (функции, предоставленные ядром);
3. Функции библиотеки (предоставленные системными библиотеками);
4. Устройства (на Unix подобных системах, это обычно особенные файлы, которые чаще всего хранятся в /dev/ директории);
5. Файлы конфигурации (форматы и условные обозначения);
6. Игры;
7. Наборы макросов и стандартов;
8. Команды администрирования системы;
9. Подпрограммы ядра.

Вы можете указать раздел страницы руководства, которую вы ищете: для того, чтобы просмотреть документацию для системного вызова чтения, вам необходимо будет ввести man 2 read. Если ни один раздел не указан явно, первый раздел, который имеет страницу руководства с запрошенным именем, будет отображен. Таким образом, команда man shadow выдаст на выводе shadow(5), потому что не существует страницы руководства для *shadow* в разделах 1-4.

Безусловно, если вы не знаете название команд, руководство будет не очень полезно для вас. Введите команду apropos, которая ищет страницы руководства (или более конкретно их краткое описание) для любых ключевых слов, которые вы указали. Затем команда apropos выводит список страниц руководства, в чьих сводках упоминаются запрошенные ключевые слова наряду с однострочными сводками из страниц руководства. Если вы подобрали ключевые слова должным образом, то вы без проблем найдете имя команды, которая вас интересует.

Пример 6.1 *Поиск cp с помощью apropos*

```
$ apropos "copy file"
cp (1)               - copy files and directories
cpio (1)             - copy files to and from archives
gvfs-copy (1)        - Copy files
gvfs-move (1)        - Copy files
hcopy (1)            - copy files from or to an HFS volume
install (1)          - copy files and set attributes
ntfscp (8)           - copy file to an NTFS volume.
```

Просмотр документации по следующим ссылкам

На многих страницах руководства есть раздел «См. Также» (“See Also”), обычно рядом с документом, который ссылается на другие справочные страницы, относящиеся к аналогичным командам, или же на альтернативную внешнюю документацию. Вы можете использовать этот раздел, чтобы найти соответствующую документацию, даже если первый выбор не предоставил вам необходимой информации.

В дополнение к man, вы также можете использовать konqueror (в KDE) и yelp (в GNOME) для поиска man страниц.

6.1.2 Документация в формате info

Проект GNU подготовил руководства для большинства своих программ в информационном формате; поэтому многие страницы руководства ссылаются на соответствующую *info* документацию. Этот формат дает некоторые преимущества, но программа по умолчанию для просмотра этих документов (также называемая info) является более сложной. Мы рекомендуем вам использовать вместо нее pinfo (из пакета *pinfo*). Для её установки просто запустите apt update, а затем apt install pinfo (см. раздел 8.2.2.2, «Установка пакетов с APT» [стр. 177]).

Документация *info* имеет иерархическую структуру, и если вы вызываете pinfo без указания параметров, то она отобразит

список узлов, доступных на первом уровне. Обычно узлы носят имена соответствующих команд.

Вы можете использовать клавиши со стрелками для перемещения между узлами. Кроме того, вы также можете использовать графический браузер (который намного удобнее), например, konqueror или yelp.

Что касается языкового перевода, система *info* всегда предоставляется на английском и не подходит для какого-либо перевода, в отличие от системы страницы руководства (man page). Однако, когда вы попросите программу pinfo отобразить несуществующую *info* страницу, она вернется на страницу руководства с тем же именем (если оно существует), которое может быть переведено.

6.1.3 Пакетная документация

Каждый пакет включает в себя свою собственную документацию, и даже наименее документированные программы обычно содержат файл README, содержащий некоторую интересную и/или важную информацию. Эта документация установлена в директории /usr/share/doc/package/ (где *package* представляет имя пакета). Если документация особенно велика, она не может быть включена в основной пакет программы, но может быть разгружена в выделенный пакет, который обычно называется *package*-doc. Основной пакет обычно рекомендует пакет документации, чтобы вы могли легко найти его.

Директория /usr/share/doc/*package/* о также содержит некоторые файлы, предоставленные Debian, которые дополняют документацию путем указания особенностей пакета или его улучшений по сравнению с традиционной установкой программного обеспечения. Файл README.Debian также обозначает все возможные адаптации, которые были сделаны для того, чтобы соответствовать политике Debian. Файл changelog.Debian.gz позволяет пользователю следить за изменениями, внесенными в пакет с течением времени; очень важно попытаться понять, что изменилось между двумя

установленными версиями, которые не обладают одинаковым поведением. И наконец, иногда есть файл NEWS.Debian.gz, который документирует основные изменения в программе, которые могут непосредственно касаться администратора.

6.1.4 Вебсайты

Во многих случаях вы можете встретить сайты, которые используются для распространения бесплатных программ и объединения сообщества разработчиков и пользователей. Эти сайты, как правило, наполняются соответствующей информацией в различных формах, такой как официальная документация, часто задаваемые вопросы (FAQ) и архивы списков рассылки. В большинстве случаев, FAQ или архивы списков рассылки рассматривают проблемы, с которыми вы могли столкнуться. При поиске информации в Интернете очень важно изучить синтаксис поиска. Один полезный совет: попробуйте ограничить поиск конкретным доменом, например, посвященным программе, в которой у вас и возникли проблемы. Если поиск выдают вам слишком много страниц или если результаты не соответствуют тому, что вы искали, вы можете добавить такие ключевые слова как kali или debian для ограничения результатов соответственной целевой информации.

От проблемы к решению

Если программное обеспечение выдает очень специфическое сообщение об ошибке, введите его (это сообщение) в поисковик (между двумя кавычками, для обозначения цитаты для того, чтобы начать поиск по конкретной фразе, а не по отдельным ключевым словам). В большинстве случаев, первые ссылки будут содержать ответ, который вам нужен.

В других случаях, вы получите очень общие ошибки, вроде “Permission denied”. В такой ситуации, лучше всего проверить разрешения соответствующих элементов (файлов, пользовательских ID, групп и т.д.). Вкратце, не стоит вырабатывать привычку постоянного использования поисковиков,

для поиска решения проблемы, иначе вы рискуете забыть о том, что такое использовать здравый смысл.

Если вы не знаете адрес веб-сайта программного обеспечения, существуют различные способы его определения. Сначала найдите поле «Домашняя страница» в метаинформации пакета (apt show package). Кроме того, описание пакета может содержать ссылку на официальный сайт программы. Если URL-адрес не указан, в сопровождающем файле пакета может содержаться URL-адрес в файле /usr/share/doc/package/copyright. Наконец, вы можете использовать поисковую систему (например, Google, DuckDuckGo, Yahoo и т. Д.), чтобы найти веб-сайт программного обеспечения.

6.1.5 Kali документация на docs.kali.org

Проект Kali содержит сборник полезной документации по адресу http://docs.kali.org. Хотя эта книга охватывает значительную часть того, что вы должны знать о Kali Linux, документация на данном сайте может быть полезной, поскольку она содержит пошаговые инструкции (во многом похожие на практические руководства) касательно многих тем. http://docs.kali.org/

Давайте рассмотрим различные темы, затрагиваемые там:

- Начало работы: серия инструкций, включая инструкцию по загрузке, для тех, кто не знаком с Kali;
- Kali Linux Live: документация, описывающая использование Kali Linux в качестве живой системы (live system);
- Установка Kali Linux: различные документы, описывающие установку Kali Linux, в том числе, как установить ее наряду с другими операционными системами;
- Kali Linux на ARM: многие рецепты о запуске Kali Linux на различных устройствах на базе ARM;
- Использование Kali Linux: многочисленные практические пособия, затрагивающие множество распространённых вопросов;
- Настройка Kali Linux: инструкции для смельчаков, которые любят перестраивать Kali согласно своим собственным

требованиям и предпочтениям;
- Поддержка сообщества Kali: указатели на различные сообщества, где вы можете получить поддержку и получить подробное разъяснение относительно того, как отправлять отчеты об ошибках;
- Политика Kali Linux: объяснения того, что делает Kali Linux особенным по сравнению с другими дистрибутивами Linux;
- Kali Linux додзё: видео Black Hat и DEF CON семинаров.

6.2 Сообщества Kali Linux

Существует довольно много сообществ Kali Linux по всему миру, которые в свою очередь используют множество различных инструментов для обмена сообщениями (например, форумы и социальные сети). В этом разделе мы представим только два официальных сообщества Kali Linux.

6.2.1 Веб-форумы на forums.kali.org

Официальные форумы сообщества проекта Kali Linux расположены на forums.kali.org[1]. Как и на любом веб-форуме, вы должны создать учетную запись, чтобы начать отправлять сообщения, и система сразу запоминает, какие сообщения вы уже видели, что позволяет легко следить за общением на регулярной основе.

Перед тем как начать общение, вам необходимо ознакомиться с правилами форума:

http://docs.kali.org/community/kali-linux-community-forums

Мы не будем копировать здесь весь перечень правил, но стоит отметить, что вам не разрешается говорить о незаконных действиях, таких как проникновение в чужие сети. Вы должны уважать других членов сообщества, чтобы поддерживать приветливое общение. Реклама запрещена, и следует избегать обсуждений вне темы. Существует достаточно различного рода

категорий, чтобы охватить всё, что вы хотели бы обсудить о Kali Linux.

6.2.2 #kali-linux IRC канал на Freenode

IRC является чат-системой реального времени. Обсуждения происходят в чатах, которые называются *каналами* и обычно сосредоточены вокруг определенной темы или сообщества. Проект Kali Linux использует канал # kali- linux в сети Freenode[19] [20] (вы можете использовать chat.freenode.net в качестве IRC-сервера на порту 6667 для TLS-шифрованного соединения или порт 6666 для открытого текстового соединения).

Чтобы присоединиться к обсуждениям в IRC, вы должны использовать IRC-клиент, такой как hexchat (в графическом режиме) или irssi (в консольном режиме). Существует также веб-клиент, доступный на webchat.freenode.net[21].

Несмотря на то, что довольно таки просто присоединиться к обсуждению, вы должны знать, что каналы IRC имеют свои собственные правила и что есть операторы каналов (их никнейм имеет префикс @), которые могут наказывать пользователей за невыполнение правил: они могут принудительно отсоединить вас от канала (или даже забанить вас в том случае, если вы настойчиво и принципиально отказываетесь выполнять правила). Канал #kali-linux не является исключением. Все правила приведены здесь:

http://docs.kali.org/community/kali-linux-irc- channel

Подводя итог правилам: вы должны быть дружелюбными, терпимыми и разумными. Вы должны избегать обсуждения вне темы. В частности, запрещены дискуссии о незаконной деятельности, такой как: краденое программное обеспечение, размещаемое на хакерских сайтах/взломы/ пиратское программное обеспечение, также запрещены любые обсуждения

[19] http://forums.kali.org
[20] http://www.freenode.net
[21] http://webchat.freenode.net

о политике и религиях. Имейте в виду, что ваш IP-адрес будет доступен другим пользователям.

Если вы хотите обратиться за помощью, следуйте рекомендациям, приведенным в разделе «Как избежать ответов RTFM» [стр. 124]: сначала детально исследуйте вопрос самостоятельно и затем поделитесь результатами. Когда вас попросят предоставить дополнительную информацию, пожалуйста, предоставьте ее точно (если вы должны предоставить несколько подробных результатов, не вставляйте их непосредственно в канал, а используйте службу, например, Pastebin[22] , и отправляйте только URL-адрес Pastebin.

Не ожидайте немедленного ответа. Несмотря на то, что IRC является платформой связи в реальном времени, участники регистрируются со всего мира, поэтому часовые пояса и графики работы различаются. Для ответа на ваш вопрос может потребоваться несколько минут или часов. Однако, когда другие включают ваш ник в ответ, ваш ник будет подсвечен, и в большинстве случаев клиент IRC уведомит вас, поэтому оставьте клиент подключенным и наберитесь терпения.

6.3 Подача грамотно составленного отчета об ошибке

Если все ваши попытки разрешить проблему увенчались неудачей, то вполне вероятно, что проблема связана с ошибкой (багом) в самой программе. В данном случае проблему можно детально изложить в отчете об ошибке. Вы можете продолжить поиск отчетов об ошибке, которые идентичны вашей, но давайте все же рассмотрим саму процедуру составления отчета об ошибке и предоставления его Kali, Debian, или напрямую разработчикам, таким образом, чтобы вы окончательно поняли, как именно следует составлять ваш собственные отчет об ошибке.

Целью отчета об ошибке является предоставление достаточной информации для того, чтобы разработчики или специалисты по эксплуатации (предположительно) неисправной программы могли воспроизвести проблему, отладить ее поведение и разработать исправление ошибки. Это означает, что ваш отчет об ошибке

[22]http://pastebin.com

должен содержать соответствующую информацию и должен быть направлен к правильному человеку или команде проекта. Отчет также должен быть хорошо написанным и тщательным, что в свою очередь обеспечит более быстрый ответ.

Точная процедура для отчета об ошибке будет различаться в зависимости от того, кому именно вы будете отправлять отчет (Kali, Debian, или напрямую разработчикам), но есть некоторые общие рекомендации, применимые ко всем случаям. В этой главе мы обсудим эти рекомендации.

6.3.1 Общие рекомендации

Давайте обсудим общие рекомендации и основные принципы, которые помогут вам составить и подать отчет об ошибке, который будет понятным, всеобъемлющим и увеличит шансы того, что ошибка будет устранена разработчиками своевременно.

Как обращаться

Составляйте ваш отчет исключительно на английском.

Сообщество свободного программного обеспечения (The Free Software community) является международным, и, если вы не знаете своего собеседника, вы должны использовать простой английский. Если вы являетесь носителем английского языка, используйте простые предложения и избегайте конструкций, которые могут вызвать сложности в понимании для людей с ограниченными навыками английского языка. Несмотря на то, что большинство разработчиков очень интеллектуальны, не все из них обладают сильными навыками английского языка. Так что давайте постараемся, чтобы любой сотрудник мог с легкостью понять суть содержимого.

Относитесь с уважением к работе разработчиков

Помните, что большинство разработчиков Free Software (включая тех, кто стоит за Kali Linux) доброжелательны и тратят свое

ограниченное свободное время на работу с программным обеспечением, которое вы свободно используете. Многие делают это из альтруизма. Таким образом, когда вы отправляете отчет об ошибке, будьте почтительны (даже если проблема выглядит как очевидная ошибка разработчика) и не стоит думать, что они должны вам немедленно исправить ошибку. Будьте благодарными за их вклад.

Если вы знаете, как модифицировать и перекомпилировать программное обеспечение, предложите помочь разработчикам в тестировании любых патчей, которые они представляют вам. Это покажет им, что вы тоже готовы инвестировать свое время.

Будьте активными и готовыми предоставить дополнительную информацию

В некоторых случаях, разработчик может обратиться к вам с просьбой предоставить большее количество информации, или воссоздать проблему используя другие варианты или обновленные пакеты. Вы должны постараться отвечать на подобные запросы как можно быстрее. Чем быстрее вы ответите на подобного рода запрос, тем выше шансы на то, что разработчики смогут разрешить проблему быстро, пока первичные данные еще свежи.

Хотя вы должны стремиться быстро реагировать, вы также не должны чересчур торопиться: представленные данные должны быть правильными и должны содержать все, что запросили разработчики. Они будут очень раздосадованы, если им потребуется обращаться к вам за чем-то во второй раз.

Что необходимо указывать в отчете об ошибке

Подробные инструкции о том, каким образом можно воссоздать проблему

Чтобы иметь возможность воспроизвести проблему, разработчики должны знать, что вы используете, каким образом вы ее получили, и как вы ее установили.

Вы должны предоставить точные, пошаговые инструкции, описывающие, как воспроизвести проблему. Если вам нужно использовать некоторые данные для воспроизведения проблемы, прикрепите соответствующий файл к отчету об ошибке. Попытайтесь придумать минимальный набор инструкций, необходимых для воспроизведения ошибки.

Предоставьте необходимый контекст и укажите свои ожидания

Объясните, что вы пытались сделать и каким образом вы ожидали, что программа будет вести себя.

В некоторых случаях ошибка срабатывает только потому, что вы использовали программу для выполнения тех задач, какие она не предназначена выполнять. Объясняя, чего вы пытались достичь, вы позволяете разработчикам ясно понимать так это или нет.

В некоторых других случаях поведение, которое вы описываете как ошибку, на самом деле может быть нормальным поведением программы. Постарайтесь излагать свои мысли максимально ясно, когда будете описывать то, что вы ожидали от программы. Это прояснит ситуацию для разработчиков. Они могут либо исправить поведение программы, либо улучшить документацию, но, по крайней мере, они будут знать, что поведение их программы запутывает некоторых пользователей!

Будьте максимально конкретными

Включите номера версий программного обеспечения, которое вы используете и, по возможности, укажите номера версий их зависимостей. Когда вы ссылаетесь на что-то, что вы скачали, укажите его полный URL-адрес.

Когда вы получите сообщение об ошибке, укажите его точно в таком виде, в каком вы его получили. Если возможно, включите копию вывода на экране или снимок экрана. Включите копию любого соответствующего файла журнала, однако предварительно убедитесь, что вы удалили все конфиденциальные данные.

Упоминание возможных вариантов исправлений или обходных решений

Перед подачей отчета об ошибке вы, вероятно, попытались решить проблему. Объясните, что вы пробовали, и какие результаты вы получили. Будьте предельно ясны в том, что вы реально пытались сделать и что является всего лишь гипотезой с вашей стороны.

Если вы выполняли интернет поиск и находили какого-либо рода разъяснения касательно вашей проблемы, вы можете указать это, особенно если вы нашли похожий отчет об ошибке в Debian трекере.

Если вы нашли способ достижения желаемого результата без вызывания ошибки, пожалуйста, задокументируйте это. Это поможет другим пользователям, которые пострадали от одной и той же проблемы.

Длинные отчеты об ошибках являются вполне нормальным явлением

Отчет об ошибке с двумя строками является недостаточным; для обеспечения всей необходимой информации обычно требуется несколько абзацев (или иногда страниц) текста.

Предоставьте всю информацию, которую вы можете. Постарайтесь придерживаться того, что имеет значение, но если вы не уверены, то помните, что слишком много лучше, чем слишком мало.

Если ваш отчет об ошибках действительно длинный, найдите время, чтобы структурировать его содержимое и представить краткое резюме в начале.

Различные советы

Избегайте подачи дублированных отчетов об ошибках

В мире свободного программного обеспечения все трекеры ошибок являются общедоступными. Открытые проблемы можно просмотреть, и у них даже есть функция поиска. Таким образом, перед подачей нового отчета об ошибке попытайтесь определить, была ли ваша проблема уже сообщена кем-то другим.

Если вы обнаружите существующий отчет об ошибках, подпишитесь на него и, возможно, добавьте дополнительную информацию. Не публикуйте комментарии, такие как «Я тоже» или «+1»; они не имеют никакого смысла. Но вы можете указать, что вы готовы к дальнейшим испытаниям, если исходный заявитель этого не предлагал.

Если вы не нашли отчета о своей проблеме, то можете смело переходить к регистрации. Если вы нашли связанные с вашей проблемой свидетельства, обязательно укажите их.

Обязательно убедитесь в том, что вы используете последнюю версию

Разработчиков очень сильно запутывают отчеты об ошибках относительно проблем, которые они уже решили, или проблем, которые они не могут воспроизвести с использованием версии, которую они на данный момент используют (разработчики почти всегда используют последнюю версию своего продукта). Даже когда старые версии поддерживаются разработчиками, такая поддержка часто ограничивается исправлениями в области безопасности или более серьезными проблемами. Вы уверены, что ваша ошибка одна из них?

Вот почему перед подачей отчета об ошибке вы должны убедиться, что используете последнюю версию проблемной системы или приложения, и что вы можете воспроизвести проблему в случае необходимости.

Если Kali Linux не предлагает самую последнюю версию приложения (ни в kali-rolling ни в kalibleeding-edge, смотри раздел 8.1.3.3, “Репозиторий Kali-Bleeding-Edge Repository” [стр. 174]), у вас есть альтернативное решение: вы можете попробовать выполнить ручную установку последней версии на одноразовой виртуальной машине, или вы можете просмотреть восходящий

ChangeLog (или Git commit историю), чтобы увидеть, что не было никаких изменений, которые могли бы устранить проблему, (и затем вы можете регистрировать ошибку, не смотря на то, что вы не пробовали последнюю версию).

Не смешивайте несколько проблем в одном отчете об ошибке

Регистрируйте один отчет об ошибке для каждой отдельной ошибке. Таким образом, последующие обсуждения не становятся слишком беспорядочными, и каждая ошибка может быть исправлена в соответствии с ее собственным особенностями. Если вы этого не сделаете, то либо одна и та же ошибка должна будет видоизменяться множество раз и может быть закрыта только в том случае, когда все ошибки будут устранены, либо разработчики должны будут регистрировать дополнительные отчеты, которые необходимо было сделать вам в первую очередь.

6.3.2 Где регистрировать отчет об ошибке

Чтобы иметь возможность решить, где записать отчет об ошибке, вы должны хорошо понимать проблему, и вы должны определить, в какой части программного обеспечения заключена проблема.

Идеально, вам необходимо отследить проблему прямо до конкретного файла в вашей системе, а затем вы можете использовать dpkg для того, чтобы определить какой пакет владеет эти файлом и кем поставляется этот пакет. Давайте предположим, что вы нашли ошибку в графическом приложении. После просмотра списка запущенных процессов (вывод команды ps auxf), что приложение было запущено с исполняемым файлом /usr/bin/Sparta:

```
$ dpkg -S /usr/bin/sparta
sparta: /usr/bin/sparta
$ dpkg -s sparta | grep ^Version:
Version: 1.0.1+git20150729-0kali1
```

Вы узнаете, что /usr/bin/sparta предоставляется пакетом sparta, который находится в версии 1.0.1+git 20150729-Okalil. Тот факт,

что строка версии содержит kali, указывает на то, что пакет поставляется Kali Linux (или был изменен Kali Linux). Любой пакет, который не имеет kali в своей строке версии (или в имени пакета), поставляется Debian (Debian Testing вообще).

Двойная проверка перед тем, как в идеале подать файлы с ошибками в Debian

Если вы обнаружили ошибку в пакете, импортированном прямо из Debian, он должен сообщаться и исправляться со стороны Debian. Однако, перед этим убедитесь, что проблема воспроизводима в простой системе Debian, поскольку Kali, возможно, вызвала проблему, изменив другие пакеты или зависимости.

Самый простой способ сделать это - настроить виртуальную машину, на которой запущен Debian Testing. Вы можете найти установочный образ ISO для Debian Testing на веб-сайте Debian Installer:

https://www.debian.org/devel/debian-installer/

Если вы можете подтвердить проблему на виртуальной машине, вы можете отправить отчет об ошибке в Debian, выполнив функцию отчет об ошибке в виртуальной машине и следуя инструкциям.

Большинство отчетов об ошибках в отношении поведения приложений должны быть направлены непосредственном владельцам этих проектов, кроме случаев, когда вы сталкиваетесь с проблемой интеграции: в этом случае неполадка является ошибкой, вызванной тем, как программное обеспечение пакетируется и интегрируется в Debian или Kali.

Например, если приложение предлагает параметры времени компиляции, которые пакет не разрешает или приложение не работает из-за отсутствующей библиотеки (таким образом, в результате возникает недостающая зависимость в метаинформации пакета), вы можете столкнуться с проблемой интеграции. Когда вы не знаете, с какой проблемой вы сталкиваетесь, обычно лучше всего зарегистрировать проблему с обеих сторон и перекрестно ссылаться на нее.

Определение проекта, к которому относится ошибка и нахождение места, куда можно подать/зарегистрировать отчет об ошибке на самом деле довольно легко. Вы просто должны просмотреть необходимый сайт, на который есть ссылка в поле «Домашняя страница» (Homepage) метаданных пакетирования.

```
$ dpkg -s sparta | grep ^Homepage:
Homepage: https://github.com/SECFORCE/sparta
```

6.3.3 Как регистрировать отчет об ошибке

Создание отчета об ошибке в Kali

Kali использует веб-трекер ошибок на http://bugs.kali.org/, где вы можете анонимно проконсультироваться относительного любого рода отчетах об ошибках, но если вы хотите прокомментировать или опубликовать новый отчет об ошибке, вам нужно будет зарегистрировать учетную запись.

Регистрация учетной записи на баг трекере

Для начала просто нажмите на *Создать новый аккаунт (Signup for new account)* на веб-сайте баг трекера, как показано на рисунке 6.1, “Начальная страница Kali баг трекера” [стр. 134].

Рисунок 6.1 *Начальная страница Kali баг трекера*

Затем, укажите имя пользователя, e-mail адрес, и ответьте на запрос CAPTCHA. Далее нажмите на кнопку Signup для продолжения (Рисунок 6.2, «Страница регистрации» [стр. 134]).

Рисунок 6.2 *Страница регистрации*

Если все прошло успешно, следующая страница (Рисунок 6.3, "Страница подтверждения регистрации" [стр. 135]) сообщит вам о том, что регистрация вашего аккаунта продолжается и что система баг трекера отправила вам письмо для подтверждения создания вашей учетной записи, на предоставленный вами ранее электронный адрес. Вам нужно будет перейти по ссылке, которая указана в полученном вами письме для того, чтобы активировать ваш аккаунт.

Как только ваш аккаунт был активирован, нажмите Proceed, чтобы перейти на страницу входа в баг трекер.

Рисунок 6.3 *Страница подтверждения регистрации*

Создание отчета

Чтобы начать свой отчет, войдите в свою учетную запись и нажмите на ссылку «Сообщить о проблеме» (Report Issue) на целевой странице. Вам будет представлена форма с множеством полей для заполнения, как показано на рисунке 6.4, «Форма для создания отчета об ошибке» [стр. 136].

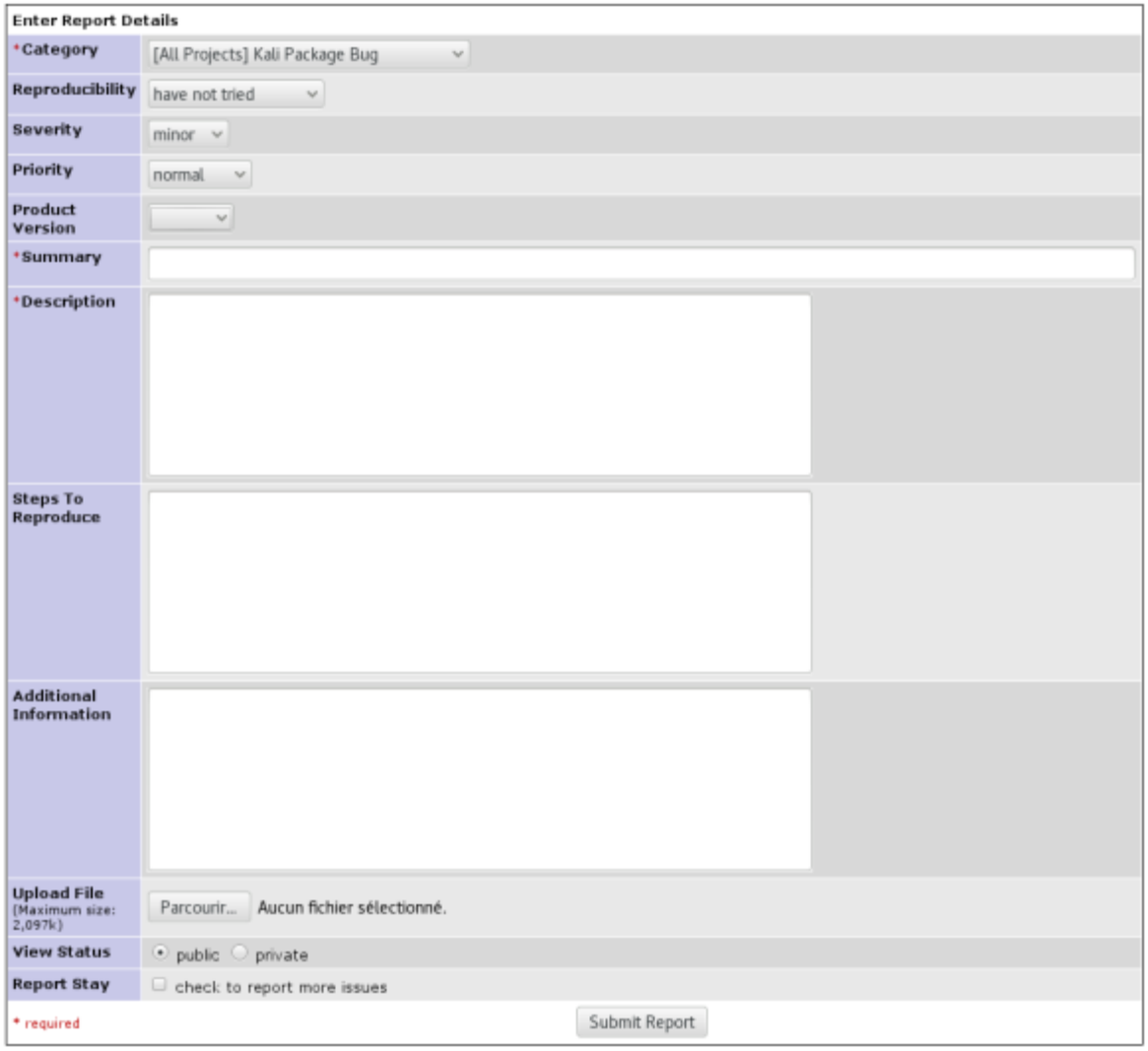

Рисунок 6.4 Форма для создания отчета об ошибке

Ниже приведено краткое описание всех полей формы:

- **Категория (обязательно для заполнения)** Это поле описывает категорию ошибки, которой посвящен отчет. Отчеты, которые относятся к конкретному пакету, должны быть зарегистрированы в категориях Kali Package Bug или Kali Package Improvement. Другие отчеты должны использовать категории General Bug или Feature Requests. Оставшиеся категории предназначены для особых случаев: категория Tool Upgrade может быть использована для того, чтобы сообщить разработчикам Kali о доступности новой версии программного обеспечения, пакетированного в Kali. Категория New Tool Requests может быть использована для предложения новых инструментов для пакетирования и интеграции в дистрибутив

Kali.

- **Воспроизводимость -** В этом поле указывается, может ли проблема воспроизводиться предсказуемым образом, или она постоянно возникает случайным образом.

- **Серьезность и приоритетность -** Эти поля лучше всего оставить без изменений, поскольку они предназначены главным образом для разработчиков. Они могут использовать их для сортировки списка проблем в соответствии с серьезностью проблемы и приоритетом, в соответствии с которым она должна обрабатываться.

- **Версия продукта -** В этом поле должно быть указано, какую версию Kali Linux вы используете (или хотя бы ту, которая наиболее близка к той, что вы используете). Подумайте дважды, прежде чем сообщать о проблеме в старой версии, которая больше не поддерживается.

- **Краткое содержание (обязательно для заполнения) -** Это, по сути, является заголовком вашего отчета об ошибке, и это первое, что люди увидят. Убедитесь, что он передает причину, по которой вы отправляете отчет. Избегайте общих описаний, таких как «X не работает», и вместо этого старайтесь следовать подобной конструкции «X с ошибкой Y при условии Z».

- **Описание (Description (обязательно для заполнения)) -** Это тело вашего отчета. Здесь вы должны ввести всю информацию, которую вы собрали о проблеме, с которой вы столкнулись. Не забывайте все рекомендации, приведенные в предыдущем разделе.

- **Действия по воспроизведению (Steps to Reproduce) -** В этом поле перечислены все подробные инструкции, объясняющие, каким образом можно вызвать данную проблему.

- **Дополнительная информация (Additional Information) -** В этом разделе вы можете предоставить любую дополнительную информацию, которая, по вашему мнению, имеет отношение к проблеме. Если у вас есть рекомендации относительно того, как исправить или обойти проблему, предоставьте их в этом разделе.

- **Загрузить файл (Upload File) -** Не все можно объяснить простым текстом. В этом поле вы можете прикреплять произвольные файлы к своим отчетам: скриншоты, чтобы показать ошибку, образцы документов, запускающие проблему, файлы журналов и т. д.

- **Просмотреть статус (View Status)** - Оставьте это поле «общедоступным» (“public”), чтобы каждый мог видеть ваш отчет об ошибке. Используйте «конфиденциально» («private») только для отчетов, связанных с безопасностью, которые содержат информацию о нераскрытых уязвимостях безопасности.

Создание отчета об ошибке в Debian

Debian использует (в основном) систему отслеживания ошибок на основе электронной почты, известную как Debbugs. Чтобы открыть новый отчет об ошибке, вы отправите электронное письмо (со специальным синтаксисом) на адрес submit@bugs.debian.org. Это присвоит номер ошибке XXXXXX и сообщит вам, что вы можете отправить дополнительную информацию, отправив *XXX XXX*@bugs.debian.org. Каждая ошибка связана с пакетом Debian. Вы можете просмотреть все ошибки данного пакета (включая ту ошибку, о которой вы хотите составить отчет) на https://bugs.debian.org/ *package.* Вы можете проверить историю данной ошибки на странице https://bugs.debian.org/XXXXXX.

Настраиваем Reportbug Хотя вы можете сообщить о новой ошибке с помощью простого электронного письма, мы рекомендуем использовать reportbug, потому что это поможет вам составить серьезный отчет об ошибке со всей необходимой информацией. В идеале вы должны запустить его из системы Debian (например, на виртуальной машине, где вы воспроизвели проблему).

Первый запуск reportbug запускает сценарий конфигурации. Сначала выберите уровень навыка. Вы должны выбрать Новичка (Novice) или Стандарт (Standard); мы используем последний, потому что он предлагает более детальный контроль. Затем выберите интерфейс и введите свои личные данные. Наконец,

выберите пользовательский интерфейс. Сценарий конфигурации позволит вам использовать локальный агент транспорта почты, SMTP-сервер или, в крайнем случае, SMTP-сервер Debian.

```
Welcome to reportbug! Since it looks like this is the first time you have
used reportbug, we are configuring its behavior. These settings will be
saved to the file "/root/.reportbugrc", which you will be free to edit
further.
Please choose the default operating mode for reportbug.

1 novice    Offer simple prompts, bypassing technical questions.

2 standard  Offer more extensive prompts, including asking about things
            that a moderately sophisticated user would be expected to
            know about Debian.

3 advanced  Like standard, but assumes you know a bit more about Debian,
    ➥ including "incoming".

4 expert    Bypass most handholding measures and preliminary triage
            routines. This mode should not be used by people unfamiliar
            with Debian's policies and operating procedures.

Select mode: [novice] standard
Please choose the default interface for reportbug.

1 text   A text-oriented console user interface

2 gtk2   A graphical (GTK+) user interface.

3 urwid  A menu-based console user interface

Select interface: text
Will reportbug often have direct Internet access? (You should answer
yes to this question unless you know what you are doing and plan to
check whether duplicate reports have been filed via some other channel.)
[Y|n|q|?]? Y
What real name should be used for sending bug reports?
[root]> Raphaël Hertzog
Which of your email addresses should be used when sending bug reports?
(Note that this address will be visible in the bug tracking system, so you
may want to use a webmail address or another address with good spam
filtering capabilities.)
[root@localhost.localdomain]> buxy@kali.org
Do you have a "mail transport agent" (MTA) like Exim, Postfix or SSMTP
configured on this computer to send mail to the Internet? [y|N|q|?]? N
Please enter the name of your SMTP host. Usually it's called something
like "mail.example.org" or "smtp.example.org". If you need to use a
different port than default, use the <host>:<port> alternative
format. Just press ENTER if you don't have one or don't know, and so a
Debian SMTP host will be used.
>
Please enter the name of your proxy server. It should only use this
parameter if you are behind a firewall. The PROXY argument should be
formatted as a valid HTTP URL, including (if necessary) a port number; for
example, http://192.168.1.1:3128/. Just press ENTER if you don't have one
or don't know.
>
Default preferences file written. To reconfigure, re-run reportbug with
the "--configure" option.
```

Использование Reportbug После завершения фазы настройки вы можете начать создание отчета об ошибке. Вам будет предложено указать имя пакета, хотя вы также можете указать имя пакета непосредственно в командной строке с помощью reportbug package.

```
Running 'reportbug' as root is probably insecure! Continue [y|N|q|?]? y
Please enter the name of the package in which you have found a problem, or
type 'other' to report a more general problem. If you don't know what
package the bug is in, please contact debian-user@lists.debian.org for
assistance.
> wireshark
```

В отличие от рекомендаций, приведенных выше, если вы не знаете, с каким пакетом следует подать ошибку, вы должны связаться с форумом поддержки Kali (см. Раздел 6.2 «Kali Linux сообщества» [стр. 128]). На следующем шаге reportbug загружает список ошибок, поданных с данным пакетом, и это в свою очередь позволяет вам узнать, можете ли вы найти в нем свою ошибку.

```
*** Welcome to reportbug.  Use ? for help at prompts. ***
Note: bug reports are publicly archived (including the email address of
the submitter).
Detected character set: UTF-8
```

```
Please change your locale if this is incorrect.

Using '"Raphaël Hertzog" <buxy@kali.org>' as your from address.
Getting status for wireshark...
Verifying package integrity...
Checking for newer versions at madison...
Will send report to Debian (per lsb_release).
Querying Debian BTS for reports on wireshark (source)...
35 bug reports found:

Bugs with severity important
   1) #478200  tshark: seems to ignore read filters when writing to…
   2) #776206  mergecap: Fails to create output file > 2GB
   3) #780089  wireshark: "On gnome wireshark has not title bar. Does…
Bugs with severity normal
   4) #151017  ethereal: "Protocol Hierarchy Statistics" give misleading…
   5) #275839  doesn't correctly dissect ESMTP pipelining
[...]
  35) #815122  wireshark: add OID 1.3.6.1.4.1.11129.2.4.2
(24-35/35) Is the bug you found listed above [y|N|b|m|r|q|s|f|e|?]? ?
y - Problem already reported; optionally add extra information.
N - (default) Problem not listed above; possibly check more.
b - Open the complete bugs list in a web browser.
m - Get more information about a bug (you can also enter a number
    without selecting "m" first).
r - Redisplay the last bugs shown.
q - I'm bored; quit please.
s - Skip remaining problems; file a new report immediately.
f - Filter bug list using a pattern.
e - Open the report using an e-mail client.
? - Display this help.
(24-35/35) Is the bug you found listed above [y|N|b|m|r|q|s|f|e|?]? n
Maintainer for wireshark is 'Balint Reczey <balint@balintreczey.hu>'.
Looking up dependencies of wireshark...
```

Если вы обнаружили, что отчет о вашей ошибке уже подан, вы сможете отправить дополнительную информацию, иначе вам будет предложено подать новый отчет об ошибке:

```
Briefly describe the problem (max. 100 characters allowed). This will be
the bug email subject, so keep the summary as concise as possible, for
example: "fails to send email" or "does not start with -q option
specified" (enter Ctrl+c to exit reportbug without reporting a bug).
> does not dissect protocol foobar
Rewriting subject to 'wireshark: does not dissect protocol foobar'
```

После предоставления однострочной сводки о вашей проблеме вы должны оценить ее серьезность в расширенном масштабе:

```
How would you rate the severity of this problem or report?

1 critical        makes unrelated software on the system (or the whole
                  system) break, or causes serious data loss, or
                  introduces a security hole on systems where you install
                  the package.
2 grave           makes the package in question unusable by most or all
                  users, or causes data loss, or introduces a security
                  hole allowing access to the accounts of users who use
                  the package.
3 serious         is a severe violation of Debian policy (that is, the
                  problem is a violation of a 'must' or 'required'
                  directive); may or may not affect the usability of the
                  package. Note that non-severe policy violations may be
                  'normal,' 'minor,' or 'wishlist' bugs. (Package
                  maintainers may also designate other bugs as 'serious' and
                  thus release-critical; however, end users should not do
                  so.). For the canonical list of issues worthing a serious
                  severity you can refer to this webpage:
                  http://release.debian.org/testing/rc_policy.txt
4 important       a bug which has a major effect on the usability of a
                  package, without rendering it completely unusable to
                  everyone.
5 does-not-build  a bug that stops the package from being built from source.
                  (This is a 'virtual severity'.)
6 normal          a bug that does not undermine the usability of the whole
                  package; for example, a problem with a particular option
                  or menu item.
7 minor           things like spelling mistakes and other minor cosmetic
                  errors that do not affect the core functionality of the
                  package.
8 wishlist        suggestions and requests for new features.

Please select a severity level: [normal]
```

Если вы не уверены, просто соблюдайте правила стандартной процедуры. Вы также можете отметить свой отчет несколькими ключевыми словами:

Применимо ли что-либо из нижеперечисленного к вашему отчету?

```
Do any of the following apply to this report?

1 d-i       This bug is relevant to the development of debian-installer.
2 ipv6      This bug affects support for Internet Protocol version 6.
3 l10n      This bug reports a localization/internationalization issue.
4 lfs       This bug affects support for large files (over 2 gigabytes).
5 newcomer  This bug has a known solution but the maintainer requests someone
            else implement it.
```

Большинство из тегов будут понятны скорее лишь посвященным, но если ваш отчет содержит исправление проблемы, вам следует выбрать тег patch.

Когда этот этап будет завершен, reportbug откроет текстовый редактор с шаблоном, который вы должны отредактировать (пример 6.2, «Шаблон, созданный reportbug» [стр. 142]). Он содержит несколько вопросов, которые вы должны сначала удалить и на их месте указать собственный ответ, а также некоторую информацию о вашей системе, которая была автоматически собрана. Обратите внимание, как структурируются первые несколько строк. Они не должны быть изменены, поскольку они будут проанализированы баг трекером, чтобы назначить отчет соответствующему пакету.

Пример 6.2 Шаблон созданный reportbug

```
Subject: wireshark: does not dissect protocol foobar

Package: wireshark
Version: 2.0.2+ga16e22e-1
Severity: normal

Dear Maintainer,

*** Reporter, please consider answering these questions, where appropriate ***

   * What led up to the situation?
   * What exactly did you do (or not do) that was effective (or
     ineffective)?
   * What was the outcome of this action?
   * What outcome did you expect instead?

*** End of the template - remove these template lines ***


-- System Information:
Debian Release: stretch/sid
  APT prefers testing
  APT policy: (500, 'testing')
Architecture: amd64 (x86_64)
Foreign Architectures: i386

Kernel: Linux 4.4.0-1-amd64 (SMP w/4 CPU cores)
Locale: LANG=fr_FR.utf8, LC_CTYPE=fr_FR.utf8 (charmap=UTF-8)
Shell: /bin/sh linked to /bin/dash
Init: systemd (via /run/systemd/system)
```

```
Versions of packages wireshark depends on:
ii  wireshark-qt  2.0.2+ga16e22e-1

wireshark recommends no packages.

wireshark suggests no packages.

-- no debconf information
```

После сохранения отчета и закрытия текстового редактора вы возвращаетесь к reportbug, который предоставляет множество других опций и предложений для отправки результативного отчета.

```
Spawning sensible-editor...
Report will be sent to "Debian Bug Tracking System" <submit@bugs.debian.org>
Submit this report on wireshark (e to edit) [Y|n|a|c|e|i|l|m|p|q|d|t|s|?]? ?
Y - (default) Submit the bug report via email.
n - Don't submit the bug report; instead, save it in a temporary file (exits reportbug).
a - Attach a file.
c - Change editor and re-edit.
e - Re-edit the bug report.
i - Include a text file.
l - Pipe the message through the pager.
m - Choose a mailer to edit the report.
p - print message to stdout.
q - Save it in a temporary file and quit.
d - Detach an attachment file.
t - Add tags.
s - Add a X-Debbugs-CC recipient (a CC but after BTS processing).
? - Display this help.
Submit this report on wireshark (e to edit) [Y|n|a|c|e|i|l|m|p|q|d|t|s|?]? Y
Saving a backup of the report at /tmp/reportbug-wireshark-backup-20160328-19073-87oJWJ
Connecting to reportbug.debian.org via SMTP...

Bug report submitted to: "Debian Bug Tracking System" <submit@bugs.debian.org>
Copies will be sent after processing to:
  buxy@kali.org

If you want to provide additional information, please wait to receive the
bug tracking number via email; you may then send any extra information to
n@bugs.debian.org (e.g. 999999@bugs.debian.org), where n is the bug
number. Normally you will receive an acknowledgement via email including
the bug report number within an hour; if you haven't received a
confirmation, then the bug reporting process failed at some point
(reportbug or MTA failure, BTS maintenance, etc.).
```

Создание отчета об ошибке для любого другого проекта свободного программного обеспечения (Free Software Project)

Существует большое разнообразие проектов свободного программного обеспечения, использующих различные рабочие процессы и инструменты. Эта разница также применима к всеми используемым баг трекерам. Хотя многие проекты размещены на GitHub и используют GitHub Issues для отслеживания их ошибок, есть также много других, на которых размещаются собственные трекеры, основанные на Bugzilla, Trac, Redmine, Flyspray и других. Большинство из них работают в Интернете и требуют, чтобы вы зарегистрировали учетную запись для отправки нового отчета.

Здесь мы не будем описывать все трекеры. Это зависит только от вас, какой трекер вы будете использовать из всех существующих трекеров для других проектов свободного программного обеспечения, но поскольку GitHub относительно популярен, мы

кратко рассмотрим его здесь. Как и в случае с другими трекерами, вы должны сначала создать учетную запись и войти в нее. Затем перейдите на вкладку «Проблемы» (Issues), как показано на рисунке 6.5, «Главная страница проекта GitHub» [стр. 144].

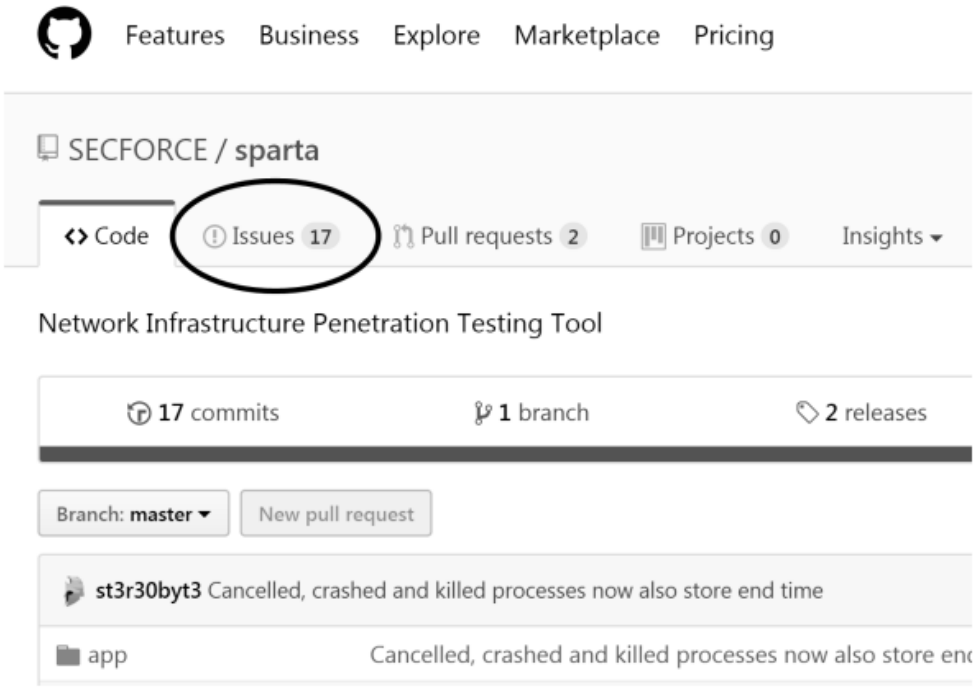

Рисунок 6.5 *Главная страница проекта GitHub*

Затем вы можете просматривать (и искать) список открытых проблем. Если вы уверены, что ваша ошибка еще не зарегистрирована, вы можете нажать кнопку «Новая проблема» (New issue) (Рисунок 6.6, страница проблем проекта GitHub »[стр. 145]).

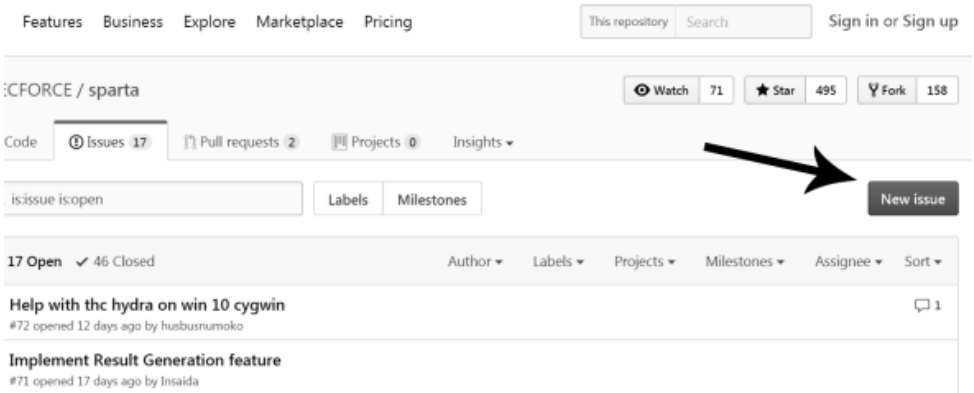

Рис. 66 Страница проблем проекта GitHub

Теперь вы находитесь на странице, где вы должны описать свою проблему (рисунок 6.7, «Форма GitHub для создания новой проблемы» [стр. 145]). Несмотря на отсутствие шаблона, подобного тому который предоставлялся в reportbug, механизм отчетности об ошибках довольно прост. Он позволяет вам прикреплять файлы, применять форматирование к тексту и многое другое. Конечно, для достижения наилучших результатов обязательно следуйте нашим рекомендациям по созданию подробного и хорошо описанного отчета.

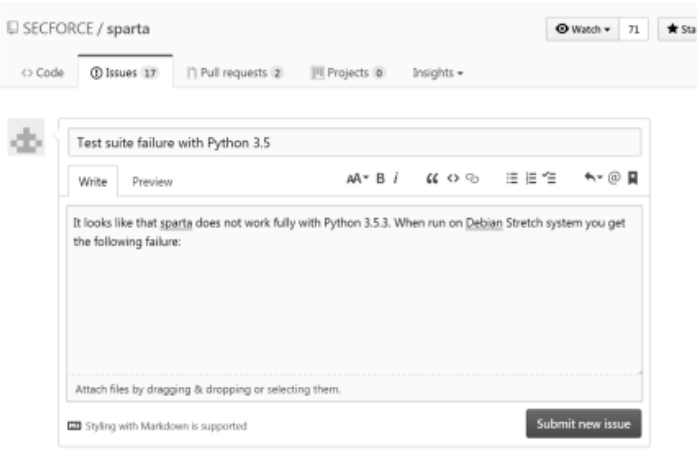

Рис. 6.7 *GitHub форма для регистрации нового отчета об ошибке*

6.4 Подведем итоги

В этом разделе мы обсудили различные методы, которые помогут вам найти документацию и информацию о программах и поиску помощи в решении проблем, с которыми вы можете столкнуться. Мы посмотрели на man и info страницы, а также познакомились с командами apropos и info. Мы обсудили отслеживание ошибок, предоставили несколько советов о том, как искать и отправлять хорошие отчеты об ошибках, а также привели несколько советов, которые помогут вам понять, кто является владельцем данной программы или проекта.

Основные моменты:

- Прежде чем вы сможете понять, что происходит, когда возникает проблема, вам нужно знать теоретическую роль каждой программы, связанной с проблемой. Один из лучших

способов сделать это - просмотреть документацию программы;

- Для того чтобы просмотреть страницу руководства просто введите *manual-page,* указав имя команды после необязательного номера раздела;
- Команда apropos выводит список страниц руководства, в сводке которых указаны запрошенные ключевые слова, а также однострочное резюме из страницы руководства;
- Проект GNU подготовил руководства для большинства своих программ в *info* формате. Вот почему многие страницы руководства ссылаются на соответствующую *info* документацию.
- Каждый пакет включает в себя собственную документацию, и даже наименее задокументированные программы обычно содержат файл README, содержащий некоторую интересную и / или важную информацию. Эта документация устанавливается в директорию /usr/share/doc/package/;
- В большинстве случаев в разделе FAQ или в архивах почтовых рассылок на официальном веб-сайте программы могут находиться информация о разрешении проблемы, с которыми вы столкнулись;
- Проект Kali содержит и поддерживает собрание очень полезной документации на http://docs.kali.org;
- Проект Kali Linux использует канал # kali-linux в сети IRC Freenode. Вы можете использовать chat.freenode.net в качестве IRC-сервера на порту 6667 для TLS-зашифрованного соединения или порта 6666 для текстового соединения. Чтобы присоединиться к обсуждениям в IRC, вы должны использовать IRC-клиент, такой как hexchat (в графическом режиме) или irssi (в консольном режиме). Существует также веб-клиент, доступный на webchat.freenode.net[23];
- Официальные форумы сообщества Kali Linux расположены на forums.kali.org[24];
- Если вы обнаружите ошибку в программе, вы можете искать отчеты об ошибках или создавать их самостоятельно. Обязательно следуйте изложенным выше рекомендациям, чтобы убедиться, что отчет четкий, всеобъемлющий и повышает вероятность того, что ошибка будет устранена разработчиками своевременно;
- Некоторые отчеты об ошибках должны быть отправлены в Kali,

[23]https://webchat.freenode.net
[24]https://forums.kali.org

а другие могут быть поданы в Debian. Команда типа **dpkg -s *package-name* | grep ^Version**: покажет номер версии и будет иметь пометку «kali», если это модифицированный Kali пакет;

- Определения соответствующего вашей ошибке проекта и нахождение места, куда подавать отчет обычно не является серьезной проблемой. Просто просмотрите соответствующий веб-сайт, который обычно отображен в поле Homepage мета-данных пакетирования;
- Kali использует баг трекер в интернете на https://bugs.kali.org, где вы можете анонимно ознакомиться со всеми отчетами об ошибках, но если вы хотите прокомментировать или подать новый отчет об ошибке, вам необходимо зарегистрировать учетную запись;
- Debian использует (в основном) систему отслеживания ошибок на основе электронной почты, известную как Debbugs. Чтобы открыть новый отчет об ошибке, вы можете отправить электронное письмо (со специальным синтаксисом) на submit@bugs.debian.org или вы можете использовать команду reportbug, которая проведет вас через этот процесс;
- Хотя многие проекты размещены на GitHub и используют GitHub Issues для отслеживания их ошибок, есть и многие другие, у которых есть свои собственные трекеры. Возможно, вам придется исследовать основы и правила других трекеров ошибок, если вам нужно будет опубликовать там отчет об ошибке.

Теперь, когда у вас есть основные инструменты для навигации по Linux, установки и настройки Kali, а также для настройки вашей системы и получения помощи, пришло время взглянуть на блокировку Kali, чтобы вы могли защитить свою инсталляцию, а также данные своего клиента.

Часть 7: Защита и мониторинг Kali Linux

Содержание:

Ключевые слова главы:

- Политика безопасности;
- Брандмауэр iptables;
- Мониторинг Логирование;

Как только вы начнете использовать Kali Linux для более конфиденциальной и высокопрофильной работы, вам, скорее всего, сразу понадобиться отнестись к безопасности вашей инсталляции намного серьезней. В этой главе мы вначале обсудим политику безопасности, выделим различные моменты, которые следует учитывать при определении подобного рода политики, и выделить некоторые угрозы для вашей системы и для вас как специалиста в сфере безопасности. Мы также обсудим меры безопасности для ноутбуков и десктопных систем, а также сосредоточимся на брандмауэрах и фильтрации пакетов. В частности, мы обсудим инструменты мониторинга и стратегии, а также покажем вам, как наилучшим образом реализовать их для обнаружения потенциальных угроз для вашей системы.

7.1 Определение политики безопасности

Нет никакого смысла обсуждать безопасность в общем, поскольку идея представляет собой широкий спектр понятий, инструментов и процедур, ни одна из которых не применяется повсеместно. Ваш выбор среди этого широкого спектра, в первую очередь, будет зависеть от поставленной вами задачи. Защита системы начинается с ответа на несколько вопросов. Стремление к внедрению произвольного набора инструментов сопряжено с риском сосредоточиться на неправильных аспектах безопасности.

Безусловно, всегда лучше определить конкретную цель. Грамотный подход, который поможет вам в этом заключается в постановки следующих вопросов:

- *Что* вы пытаетесь защитить? Политика безопасности будет отличаться в зависимости от того, хотите ли вы защитить компьютеры или данные. В последнем случае вам также необходимо знать, какие именно данные вы хотели бы защитить.
- *От чего* вы хотите защититься? Это может быть утечка информации или случайная потеря данных, возможно речь идет о потере дохода, вызванной нарушением работы служб.
- Также *от кого* вы хотите защититься? Меры безопасности будут совершенно разными, например, для защиты от

случайной опечатки постоянным пользователем системы или же для защиты от определенной группы внешних злоумышленников.

Термин «риск» обычно используется для совместного обозначения этих трех факторов: что защитить, что нужно предотвратить, и кто должен это сделать. Моделируя риски вам в первую очередь необходимо ответить на три этих вопроса. На этой модели риска может быть построена политика безопасности, и далее эта политика может быть реализована путем принятия конкретных решений и выполнения последовательности определенных действий.

Неизменный вопрос

Брюс Шнайер (Bruce Schneier), мировой эксперт по вопросам безопасности (и не только по компьютерной безопасности), пытается противостоять одному из самых важных мифов в современной безопасности, используя девиз: «Безопасность - это процесс, а не продукт». Активы, которые нужно защищать, меняются со временем, и точно также меняются угрозы и средства, доступные потенциальным злоумышленникам. Даже если политика безопасности изначально была абсолютно грамотно разработана и реализована, вы никогда не должны останавливаться на достигнутом. Компоненты риска развиваются и меняются, и ответ на этот риск должен развиваться должным образом.

Кроме того, очень важно учитывать дополнительные ограничения, поскольку они могут ограничивать диапазон доступных политик. Важно понимать, насколько далеко вы готовы зайти для защиты собственной системы? Этот вопрос имеет большое значение для реализации политики. Слишком часто ответ определяется только с точки зрения денежных издержек, но также следует учитывать другие элементы, такие как количество неудобств, наложенных на пользователей системы или ухудшение производительности.

Как только основные риски будут смоделированы, вы сможете приступить к проектированию политики безопасности.

Есть крайности, с которыми вы можете столкнуться при принятии решения о необходимом уровне безопасности. С одной стороны, чрезвычайно просто обеспечить базовую безопасность системы.

Например, если система, которая должна быть защищена, представляет собой всего лишь подержанный компьютер, единственной задачей которого является добавление нескольких чисел в конце дня, то решение не делать ничего особенного для того, чтобы защитить его, является вполне себе разумным и адекватным. Внутреннее значение системы низкое, а значение данных равно нулю, так как они не хранятся на компьютере. Потенциальный злоумышленник, проникший в эту систему, получит только калькулятор. Стоимость обеспечения такой системы, вероятно, будет больше, чем стоимость её взлома.

С другой стороны, вы, возможно, захотите защитить конфиденциальность секретных данных самым всеобъемлющим путем, каким только возможно, забыв при этом о каких-либо других вопросах. В этом случае подходящим способом реализации вашей цели будет полное уничтожение данных (безопасное стирание файлов, измельчение жестких дисков в биты, затем растворение этих битов в кислоте и т. д.). Если есть дополнительное требование, которое заключается в том, что эти данные должны сохраниться в памяти для будущего использования (хотя и не быть всегда легко доступными), и если стоимость для вас по-прежнему не является весомым фактором, то оптимальным вариантом будет хранение данных в корпусе, состоящем из иридий-платиновых пластин, хранящемся в бомбонепроницаемых бункерах под разными горами в мире, каждый из которых (конечно же) и является полностью секретным и охраняется целыми армиями.

Возможно, эти примеры могут показаться вам очень гротескными, они, тем не менее, дают абсолютно адекватный ответ на определенные риски, поскольку они являются результатом мыслительного процесса, который в свою очередь рассматривает поставленные цели и ограничения, которые вы также учитываете. Исходя из обоснованного решения, никакая политика безопасности не является более или менее достаточной, чем любая другая.

Возвращаясь к более типичному случаю, информационная система может быть сегментирована в последовательные и, в основном, независимые подсистемы. Каждая подсистема будет иметь свои собственные требования и ограничения, поэтому оценка риска и разработка политики безопасности должны проводиться отдельно для каждой подсистемы. Хороший принцип, который следует иметь в виду, состоит в том, что малую поверхность атаки намного легче защищать, чем большую. Сетевая организация также должна быть спроектирована следующим образом: уязвимые службы должны быть сконцентрированы на небольшом количестве машин, и эти машины должны быть доступны только через минимальное количество маршрутов или контрольных точек. Логика очень проста: легче защищать эти контрольные точки, чем защищать все уязвимые машины, содержащие конфиденциальную информацию, от всего внешнего мира. Именно в этот момент становится очевидной польза сетевой фильтрации (в том числе брандмауэрами). Эта фильтрация может быть реализована с помощью использования специального оборудования, но более простым и гибким решением является использование программного брандмауэра, интегрированного в ядро Linux.

7.2 Возможные меры безопасности

Как уже пояснялось в предыдущем разделе, нет единого ответа на вопрос о том, как защитить Kali Linux. Все зависит от того, каким образом вы его используете и что именно вы пытаетесь защитить.

7.2.1 На сервере

Если вы запустите Kali Linux на общедоступном сервере, вы, скорее всего, захотите защитить сетевые службы путем изменения любых паролей по умолчанию, которые могут быть настроены (см. Раздел 7.3 «Защита сетевых служб» [стр. 153]) и, возможно, также путем ограничения их доступа с помощью брандмауэра (см. раздел 7.4, «Брандмауэр или фильтрация пакетов» [стр. 153]).

Если вы передаете учетные записи пользователей либо непосредственно на сервере, либо на одной из служб, вы в любом случае захотите убедиться, что вы устанавливаете надежные пароли (они должны быть способны противостоять brute-force атакам). В то же время вы можете захотеть установить *fail2ban,* что значительно усложнит взлом паролей с помощью brute-force по сети (отфильтровывая IP-адреса, которые превышают лимит неудачных попыток входа в систему). Установите *fail2ban* с помощью apt update, за которым следует команда apt install fai! 2ban.

Если вы запускаете веб-службы, вы, вероятно, захотите разместить их через HTTPS, чтобы сетевые посредники не отслеживали ваш трафик (который может включать в себя файлы аутентификации cookie).

7.2.2 На ноутбуке

Ноутбук тестировщика на проникновение не подвержен тем же рискам, что и открытый сервер: например, менее вероятно, что вы станете объектом случайного сканирования со стороны скрипт-кидди, и даже, если это произойдет, у вас вряд ли будут доступными какие-либо сетевые службы.

Реальный риск возникает именно тогда, когда вы путешествуете от одного клиента к другому. Например, ваш ноутбук может быть украден во время подобного рода поездки или изъят на таможне. Вот почему вы, скорее всего, захотите использовать полное шифрование диска (см. Раздел 4.2.2 «Установка на полностью зашифрованной файловой системе» [стр. 85]) и, возможно, также настроите функцию «nuke» (см. «Добавление пароля Nuke для дополнительной безопасности "[стр. 245]): данные, которые вы собрали во время вашей работы, являются конфиденциальными и требуют максимальной защиты.

Вам также могут потребоваться правила брандмауэра (см. Раздел 7.4, «Брандмауэр или фильтрация пакетов» [стр. 153]), но не для той же цели, что и на сервере. Возможно, вы захотите запретить весь исходящий трафик, кроме трафика, генерируемого вашим

VPN-доступом. Это подобно безопасной сети, так что если перестает работать, вы моментально замечаете это (вместо того, чтобы возвращаться к локальному сетевому доступу). Таким образом, вы не разглашаете IP-адреса своих клиентов при просмотре веб-страниц или других сетевых действиях. Кроме того, если вы выполняете локальное внутреннее взаимодействие, лучше всего контролировать свою деятельность, чтобы уменьшить шум, создаваемый в сети, который может предупредить клиента и их системы защиты.

7.3 Защита сетевых служб

Безусловно, отключить службы, которые вы не используете, является отличной идеей. Kali упрощает это, поскольку большинство сетевых служб по умолчанию отключены.

Пока службы остаются отключенными, они не представляют угрозы безопасности. Однако вы должны быть осторожны при их включении, потому что:

- По умолчанию у них нет брандмауэра, поэтому, если они прослушивают все сетевые интерфейсы, то они являются довольно доступными для общественности.
- Некоторые службы не имеют учетных данных и позволяют устанавливать их при первом использовании; другие имеют стандартные (и, следовательно, широко известные) учетные данные. Удостоверьтесь, что вы (пере)установили пароль, который известен только вам.
- Многие службы выполняются с правами root и, соответственно, с полными правами администратора, поэтому последствия несанкционированного доступа или нарушения безопасности обычно являются катастрофическими.

Учетные данные по умолчанию

Мы не будем перечислять здесь все инструменты, которые поставляются с учетными данными по умолчанию, вместо этого вы должны проверить файл README.Debian соответствующих

пакетов, а также docs.kali.org[1] и tools.kali.org[25] [26], чтобы узнать, нуждается ли служба в специальном обслуживании для обеспечения достаточного уровня безопасности.

Если вы запуститесь в режиме реального времени, пароль учетной записи root будет «toor». Таким образом, вы не должны включать SSH перед изменением пароля учетной записи root или перед тем, как настроить свою конфигурацию для запрета входа на основе пароля.

Также обратите внимание, что, как известно, проект BeEF ((из уже установленного пакета *beef-xss*) имеет учетные данные по умолчанию: имя пользователя “beef”, и пароль “beef”), жестко закодирован в файле конфигурации.

7.4 Брандмауэр или фильтрация пакетов

Брандмауэр является частью компьютерного оборудования с аппаратным обеспечением, программным обеспечением или и тем, и другим, которое анализирует входящие или исходящие сетевые пакеты (приходящие или исходящие из локальной сети) и пропускает только те, которые соответствуют конкретным предопределенным условиям.

Фильтрующий сетевой шлюз является типом брандмауэра, который защищает всю сеть. Обычно он устанавливается на выделенный компьютер, сконфигурированный как шлюз для сети, таким образом, что он может анализировать все пакеты, которые проходят и выходят из сети. В качестве альтернативы существует локальный брандмауэр, выступающий службой программного обеспечения, которая работает на одной конкретной машине, чтобы фильтровать или ограничивать доступ к некоторым службам на этом компьютере или, возможно, предотвращать исходящие соединения от различного шпионского программного обеспечения, которое пользователь мог установить случайно или специально.

[25] https://docs.kaTi.org
[26] https://tools.kali.org

Ядро Linux встраивает брандмауэр *netfilter*. Не существует окончательного решения по настройке любого брандмауэра т.к. требования пользователя и сети довольно разнятся. Тем не менее, вы можете контролировать *netfilter* из пользовательского пространства с помощью команд iptabtes и ip6tabtes. Разница между этими двумя командами заключается в том, что первая работает для сетей IPv4, тогда как последняя работает на IPv6. Поскольку оба стека сетевых протоколов, вероятно, будут работать в течение многих лет, оба инструмента необходимо будет использовать параллельно. Вы также можете использовать отличный инструмент fwbuitder на основе графического интерфейса пользователя, который обеспечивает графическое представление правил фильтрации.

Однако, вы все же решили настроить его, *netfilter* является реализацией брандмауэра Linux, поэтому давайте подробнее рассмотрим, как он работает.

7.4.1 Поведение Netfilter

Netfilter использует четыре различные таблицы, в которых хранятся правила, регулирующие три типа операций проводимых над пакетами:

- filter касается правил фильтрации (принятия, отказа или игнорирования пакета);
- nat (Трансляция сетевых адресов (Network Address Translation)) касается перевода исходных или целевых адресов и портов пакетов;
- mangle относится к другим изменениям в IP-пакетах (включая ToS (поле типа обслуживания) *Type of Service-field* и опции));
- raw позволяет другие изменения, проводимые вручную, над пакетом пока они не достигнут системы отслеживания соединения.

Каждая таблица содержит списки правил, называемых цепями (*chains*). Брандмауэр использует стандартные цепи для обработки пакетов на основе предопределенных обстоятельств.

Администратор может создавать другие цепочки, которые будут использоваться только при передаче одной из стандартных цепочек (будь то прямо или косвенно).

Таблица filter обладает тремя стандартными цепями:

- INPUT (ВХОДЯЩИМИ): касается пакетов, назначением которых является сам брандмауэр;
- OUTPUT (ИСХОДЯЩИМИ): касается пакетов, выпускаемых брандмауэром;
- FORWARD (ПРЯМЫМИ): касается пакетов, проходящих через брандмауэр (который не является ни их источником, ни местом назначения).
- Таблица nat также обладает тремя стандартными цепями:
- PREROUTING: для изменения пакетов сразу после их поступления;
- POSTROUTING: для изменения пакетов, когда они готовы начать свой путь;
- OUTPUT (ИСХОДЯЩИМИ): для изменения пакетов, сгенерированных самим брандмуэром.

Эти цепи изображены на рисунке 7.1, “Как называются цепи *Netfilter*” [стр. 155].

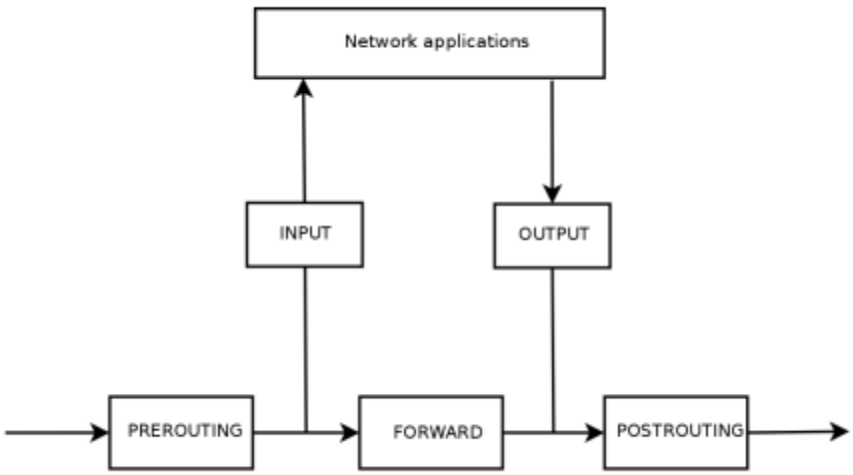

Рисунок 7.1 *Как называются цепи* **Netfilter**

Каждая цепочка представляет собой список правил; каждое правило представляет собой набор условий и действие, которые

предпринимаются при выполнении условий. При обработке пакета брандмауэр сканирует соответствующую цепочку, одно правило за другим, и когда условия для одного правила выполняются, он перескакивает (отсюда параметр -j в командах) к указанному действию для продолжения обработки. Наиболее распространенные типы поведения стандартизированы, и для них существуют специальные действия. Принятие одного из этих стандартных действий прерывает обработку цепочки, поскольку дальнейшая судьба пакетов уже предрешена (не принимая во внимание исключение, упомянутое ниже). Ниже перечислены действия *Netfilter*.

- ACCEPT (ПРИНЯТЬ): позволить пакетам двигаться далее своей дорогой.
- REJECT (ОТКЛОНИТЬ): отклонить пакет с помощью пакета ошибок протокола управляющих сообщений в интернете (Internet control message protocol (ICMP)) (опция -reject-with *type* опция iptables определяет тип отправляемой ошибки).
- DROP (СБРОСИТЬ): удалить (игнорировать) пакет.
- LOG (ЗАРЕГИСТРИРОВАТЬ): зарегистрировать (через syslogd) сообщения с описанием пакета. Обратите внимание, что это действие не прерывает обработку, а выполнение цепочки продолжается по следующему правилу, поэтому регистрация отклоненных пакетов требует как правила LOG, так и правила REJECT/DROP. Общие параметры, связанные с регистрацией, включают в себя:
- -log-level, с предупреждением по умолчанию, указывает уровень серьезности syslog.
- -log-prefix позволяет указать префикс текста для различения зарегистрированных сообщений.
- -log-tcp-sequence, -log-tcp-options, и —log-ip-options обозначают дополнительные данные, которые должны быть помещены в сообщение: соответственно, порядковый номер TCP, параметры TCP и параметры IP.
- ULOG: зарегистрировать сообщения через ulogd, который может быть лучше адаптирован и более эффективен, чем syslogd для обработки большого количества сообщений; обратите внимание, что это действие, подобно LOG, также возвращает обработку к следующему правилу в вызывающей цепочке.
- *chain_name (имя цепи):* перескочить на указанную цепь и оценить её правила.

- RETURN (ВЕРНУТЬ): прервать обработку текущей цепочки и вернуться к вызывающей цепочке; в случае, если текущая цепочка является стандартной, то вызывающей цепочки не существует, поэтому вместо нее выполняется действие по умолчанию (которое определяется с помощью параметра -p для iptables).
- SNAT (только в таблице nat): применяет *источник трансляции сетевых адресов* (*Source Network Address Translation* (SNAT)). Дополнительные опции описывают точные изменения, которые нужно применить, включая опцию -to-source ***address-port,*** которая определяет новый источник IP адреса и/или порта.
- DNAT (только в таблице nat): применяет *назначение трансляции сетевых адресов* (*Destination Network Address Translation* (DNAT)). Дополнительные опции описывают точные изменения, которые нужно применить, включая опцию -to-destination ***address-port,*** которая определяет новый источник IP адреса и/или порта.
- MASQUERADE (МАСКИРОВКА (только в таблице nat)): применяет *маскировку* (особый случай *Source NAT).*
- REDIRECT (ПЕРЕНАПРАВЛЕНИЕ (только в таблице nat)): прозрачно перенаправить пакет на данный порт самого брандмауэра; это можно использовать для настройки прозрачного веб-прокси, который работает без конфигурации на клиентской стороне, поскольку клиент считает, что он подключается к получателю, тогда как на самом деле сообщения фактически проходят через прокси-сервер. Опция -to-ports ***port(s)*** указывает порт или диапазон портов, в которых пакеты должны быть перенаправлены.

Другие действия, особенно те, которые касаются таблицы mangle, не вошли в данный раздел. Не смотря на это, на страницах руководства iptables (8) и ip6tables (8) имеется исчерпывающий объем информации.

Что такое ICMP?

Межсетевой протокол управления сообщениями (Internet Control Message Protocol (ICMP)) является протоколом, используемым для передачи вспомогательной информации по сообщениям. Он проверяет сетевое соединение с помощью команды ping, которая отправляет сообщение запроса отклика ICMP, на которое

получатель должен отвечать с соответствующим ICMP сообщением. Он сигнализирует о том, что брандмауэр отклоняет пакет, указывает на переполнение в буфере приема, предлагает лучший маршрут для следующих пакетов в соединении и т. д. Этот протокол определяется несколькими документами RFC. RFC777 и RFC792 были первыми, но многие другие расширили и/или пересмотрели протокол.

http://www.faqs.org/rfcs/rfc777.html
http://www.faqs.org/rfcs/rfc792.html

Для справки, приемный буфер представляет собой небольшую зону памяти, в которой хранятся данные между тем временем, когда они приходят из сети, и временем, когда ядро обработает их. Если эта зона заполнена, то соответственно, новые данные не могут быть получены, и ICMP будет сигнализировать о проблеме, чтобы источник мог замедлить скорость их передачи (которая в идеале должно должна быть отрегулирована через некоторое время).

Обратите внимание, что несмотря на то, что сеть IPv4 может работать без ICMP, ICMPv6 строго требуется для сети IPv6, поскольку он объединяет в себе несколько функций, которые были в мире IPv4, распространяются через ICMPv4, *Internet Croup Membership Protocol* (IGMP), и *Address Resolution Protocol* (ARP). ICMPv6 характеризуется в RFC4443.

http://www.faqs.org/rfcs/rfc4443.html

7.4.2 Синтаксис iptables и ip6tables

Команды iptables и ip6tables используются для управления таблицами, цепями и правилами. Их опция -t *table* обозначает? с какой таблицей работать (по умолчанию, filter).

Команды

Большинство опций, которые взаимодействуют с цепями перечислены ниже:

- -L *chain* перечисляет правила в цепочке. Обычно это используется с опцией -n для отключения разрешения имен (например, iptables -n -L INPUT будет отображать правила, относящиеся к входящим пакетам).
- -N *chain* создает новую цепочку. Вы можете создавать новые цепочки для множества различных целей, включая тестирование новой сетевой службы или для парирования сетевой атаки.
- -X *chain* удаляет пустую неиспользуемую цепь (например, iptables -X ddos-attack).
- -A *chain rule* добавляет правило в конце данной цепочки. Помните, что правила обрабатываются сверху вниз, поэтому не забывайте об этом при их добавлении.
- -I *chain rule_num rule* вставляет правило перед номером правила *rule_num.* Как и в опции -A, учитывайте порядок обработки при вводе новых правил в цепочку.
- -D *chain rule_num* (или -D *chain rule)* удаляет правило в цепочке; первый вариант синтаксиса определяет правило, которое должно быть удалено по его числу (iptables -L -line-numbers будет отображать эти числа), а последний вариант идентифицирует его по его содержимому.
- -F *chain* сбрасывает цепочку (удаляет все ее правила). Например, чтобы удалить все правила, связанные с исходящими пакетами, вы должны запустить iptables -F OUTPUT. Если ни одна цепочка не указана, все правила в таблице удаляются.
- -P *chain action* определяет действие по умолчанию или «политику» для данной цепочки; обратите внимание, что только стандартные цепи могут иметь такую политику. Чтобы сбросить весь входящий трафик по умолчанию, вам необходимо выполнить iptables -P INPUT DROP.

Правила

Каждое правило выражается как *conditions -j action action_options.*. Если несколько условий описаны в одном правиле, то критерием является конъюнкция (логическая функция И*)*,

которая хотя бы имеет ограничение, как и каждое отдельное условие.

Условие -p *protocol* соответствует полю протокола IP-пакета. Наиболее распространенными значениями являются tcp, udp, icmp и icmpv6. Это условие может быть дополнено условиями портов TCP, такими как -source-port port *port* и -destination-portporf.

Отрицательные условия

Добавление к условию восклицательного знака, отрицает условие. Например, отрицание условия на опции -p соответствует «любой пакет с протоколом отличным от того, который указан». Этот механизм отрицания может быть применен ко всем другим условиям.

Условия -s *address* или -s *network/mask* соответствует исходному адресу пакета. Аналогично, -d *address* or -d *network/mask* соответствует адресу назначения.

Условие -i *interface* выбирает пакеты, исходящие из заданного сетевого интерфейса. -o *interface* выбирает пакеты, выходящие на определенный интерфейс.

Условие —state *state (состояние)* condition соответствует состоянию пакета в соединении (это требует модуль ipt_ conntrack kernel для отслеживания соединения). Состояние NEW описывает пакет запускающий новое соединение, ESTABLISHED соответствует пакетам, принадлежащим к уже существующему соединению, и RELATED соответствует пакетам, инициирующим новое соединение, которое связано с существующим (что полезно для соединений ftp-данных в активном режиме протокола FTP).

Существует множество доступных опций для iptables и ip6tables, и их освоение требует глубокого изучения и длительного процесса набора опыта. Однако одна из опций, которую вы будете использовать чаще всего, - это та, которая блокирует вредоносный сетевой трафик с хоста или диапазона хостов. Например, чтобы незаметно блокировать входящий трафик с IP-адреса 10.0.1.5 и 31.13.74.0/24 класса C подсети:

```
# iptables -A INPUT -s 10.0.1.5 -j DROP
# iptables -A INPUT -s 31.13.74.0/24 -j DROP
# iptables -n -L INPUT
Chain INPUT (policy ACCEPT)
target     prot opt source               destination
DROP       all  --  10.0.1.5            0.0.0.0/0
DROP       all  --  31.13.74.0/24       0.0.0.0/0
```

Другая часто используемая команда iptables - это разрешить сетевой трафик для определенной службы или порта. Чтобы пользователи могли подключаться к SSH, HTTP и IMAP, вы можете запускать следующие команды:

```
# iptables -A INPUT -m state --state NEW -p tcp --dport 22 -j ACCEPT
# iptables -A INPUT -m state --state NEW -p tcp --dport 80 -j ACCEPT
# iptables -A INPUT -m state --state NEW -p tcp --dport 143 -j ACCEPT
# iptables -n -L INPUT
Chain INPUT (policy ACCEPT)
target     prot opt source               destination
DROP       all  --  10.0.1.5            0.0.0.0/0
DROP       all  --  31.13.74.0/24       0.0.0.0/0
ACCEPT     tcp  --  0.0.0.0/0           0.0.0.0/0           state NEW tcp dpt:22
ACCEPT     tcp  --  0.0.0.0/0           0.0.0.0/0           state NEW tcp dpt:80
ACCEPT     tcp  --  0.0.0.0/0           0.0.0.0/0           state NEW tcp dpt:143
```

Правилом хорошей компьютерной ***гигиены*** является очистка старых и ненужных правил. Самый простой способ удаления правил iptables - ссылаться на правила по номеру строки, который вы можете получить с помощью опции -line-numbers. Тем не менее, будьте очень осторожны: при сбросе правила будут перенумерованы все правила, появляющиеся дальше в цепочке.

```
# iptables -n -L INPUT --line-numbers
Chain INPUT (policy ACCEPT)
num  target     prot opt source               destination
1    DROP       all  --  10.0.1.5            0.0.0.0/0
2    DROP       all  --  31.13.74.0/24       0.0.0.0/0
3    ACCEPT     tcp  --  0.0.0.0/0           0.0.0.0/0           state NEW tcp dpt:22
4    ACCEPT     tcp  --  0.0.0.0/0           0.0.0.0/0           state NEW tcp dpt:80
5    ACCEPT     tcp  --  0.0.0.0/0           0.0.0.0/0           state NEW tcp dpt:143
# iptables -D INPUT 2
# iptables -D INPUT 1
# iptables -n -L INPUT --line-numbers
Chain INPUT (policy ACCEPT)
num  target     prot opt source               destination
1    ACCEPT     tcp  --  0.0.0.0/0           0.0.0.0/0           state NEW tcp dpt:22
2    ACCEPT     tcp  --  0.0.0.0/0           0.0.0.0/0           state NEW tcp dpt:80
3    ACCEPT     tcp  --  0.0.0.0/0           0.0.0.0/0           state NEW tcp dpt:143
```

Существуют более конкретные условия, в зависимости от общих условий, описанных выше. Для получения дополнительной информации рекомендуем ознакомиться с Iptables (8) и ip6tables (8)

7.4.3 Создание правил

Для создания каждого правила требуется один вызов iptables или ip6tables. Ввод этих команд вручную может быть очень утомительным процессом, поэтому вызовы обычно хранятся в сценарии, таким образом, что система автоматически настраивается одинаково каждый раз, когда машина загружается. Этот скрипт может быть написан вручную, но также может быть довольно интересно подготовить его с помощью инструмента высокого уровня, такого как *fwbuilder.*

```
# apt install fwbuilder
```

Принцип прост. На первом этапе опишите все элементы, которые будут задействованы в действительных правилах:

- Сам брандмауэр с сетевыми интерфейсами;
- Сети, с их соответственными диапазонами IP ranges;
- Сервера;
- Порты, принадлежащие к службам, размещенным на данном сервере.

Затем создайте правила с помощью простых действий перетаскивания объектов, как показано на рисунке 7.2, «Главное окно Fwbuilder» [стр. 160]. Несколько контекстных меню могут изменить условие (например, отрицая его). Затем нужно выбрать и настроить действие.

Что касается IPv6, вы можете либо создать два разных набора правил для IPv4 и IPv6, либо создать только одно, и позволить fwbuilder переводить правила в соответствии с адресами, назначенными объектам.

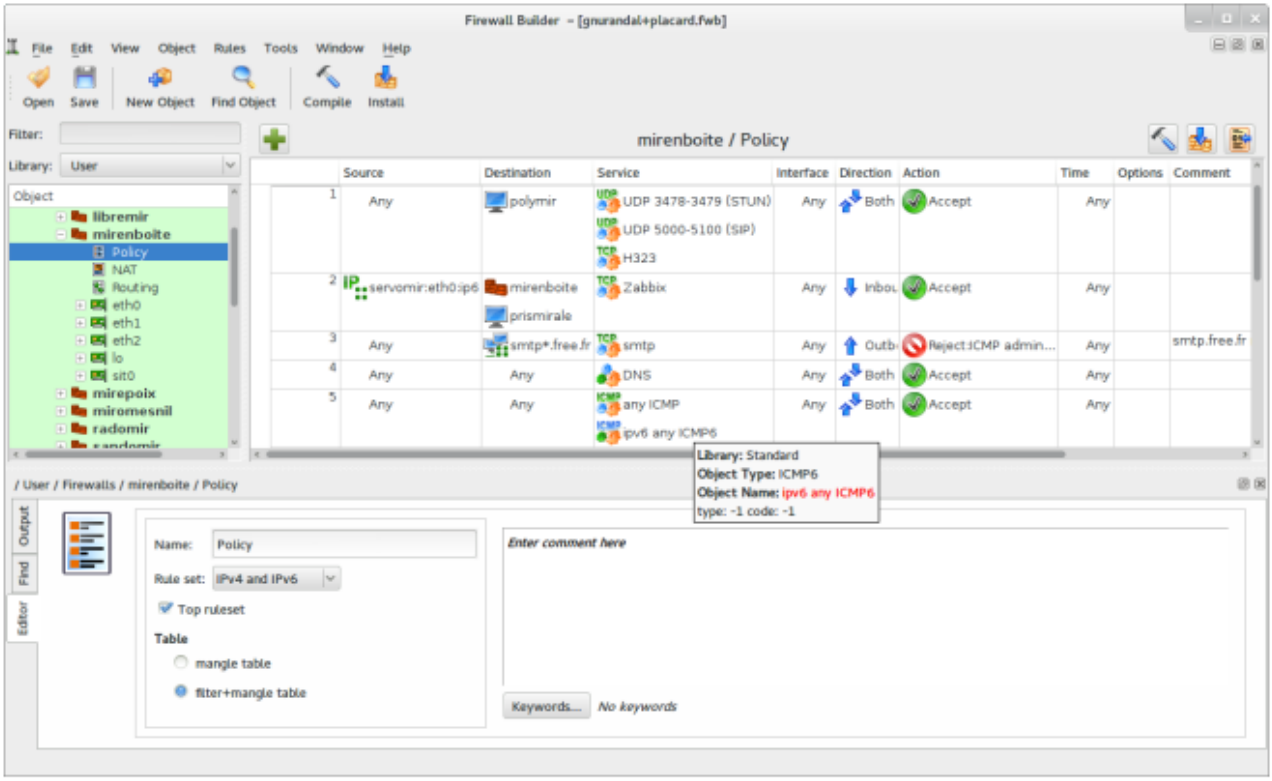

Рисунок 7.2 Главное окно Fwbuilders

fwbuilder создаст скрипт, настраивающий брандмауэр в соответствии с правилами, которые вы определили. Его модульная архитектура дает возможность генерировать сценарии, предназначенные для разных систем, включая iptables для Linux, ipf для FreeBSD и pf для OpenBSD.

7.4.4 Установка правил для каждой загрузки

Для того чтобы внедрять правила брандмауэра каждый раз, когда машина загружается, вам необходимо зарегистрировать скрипт конфигурации в соответствующей директиве файла /etc/network/interfaces file. В следующем примере скрипт хранится в /usr/tocat/etc/arrakis.fw.

```
auto eth0
iface eth0 inet static
    address 192.168.0.1
    network 192.168.0.0
    netmask 255.255.255.0
    broadcast 192.168.0.255
    up /usr/local/etc/arrakis.fw
```

Данный пример предполагает, что вы используете ***ifupdown*** для настройки сетевых интерфейсов. Если вы используете что-то другое (вроде ***NetworkManager*** или ***systemd-networkd***), тогда ознакомьтесь с их соответствующей документацией для того, чтобы определить способ, каким образом выполнить скрипт после того, как интерфейс был запущен.

7.5 Мониторинг и протоколирование (регистрация)

Конфиденциальность и защита данных являются важным аспектом безопасности, но не менее важным является обеспечение доступности услуг. В качестве администратора и специалиста в сфере безопасности вы должны следить за тем, чтобы все работало должным образом, и ваша непосредственная ответственность - своевременно выявлять аномальное поведение и ухудшение обслуживания. Программное обеспечение для мониторинга и регистрации играет ключевую роль в этом аспекте безопасности, обеспечивая понимание того, что происходит в системе и в сети.

В этом разделе мы рассмотрим некоторые инструменты, которые можно использовать для мониторинга нескольких аспектов системы Kali.

7.5.1 Мониторинг журналов с помощью logcheck

Программа logcheck отслеживает файлы журналов каждый час по умолчанию и отправляет необычное сообщения журнала в электронных письмах администратору для дальнейшего анализа.

Список отслеживаемых файлов хранится в /etc/logcheck/logcheck.logfiles. Значения по умолчанию будут работать должным образом, если файл /etc/rsyslog.conf не был полностью переработан.

Logcheck может отчитываться с различным уровнем детализации: ***paranoid***, ***server***, and ***workstation. paranoid*** ver является очень подробным и вероятно, должен быть ограничен конкретными серверами, такими как брандмауэры. ***Server*** является режимом по умолчанию и рекомендуется для большинства серверов. ***Workstation***, очевидно, предназначена для рабочих станций и является чрезвычайно сжатой, отфильтровывая больше сообщений, чем другие параметры.

Во всех трех случаях, logcheck вероятно должен быть настроен таким образом, чтобы исключить некоторые дополнительные сообщения (в зависимости от установленных служб), если вы конечно не хотите получать каждый час массивные партии длинных неинтересных электронных писем. Ввиду того, что механизм выбора сообщений является довольно сложным, соответственно/usr/share/doc/ logcheck-database/README.logcheck-database.gz будет необходимым (—if challenging—read.)

Применяемые правила можно разделить на несколько типов:

- те, которые квалифицируют сообщение как попытку взлома (хранятся в файле в директории /etc/logcheck/cracking.d/);
- игнорируют попытки взлома (/etc/logcheck/cracking.ignore.d/);
- те, кто классифицирует сообщение как предупреждение безопасности (/etc/logcheck/violations.d/);
- игнорируют предупреждения безопасности (/etc/logcheck/violations.ignore.d/);
- и наконец, которые применяются к остальным сообщениям (рассматриваются как ***системные события***).

Игнорируемые файлы используются для игнорирования (очевидно) сообщений. Например, сообщение, помеченное как попытка взлома или предупреждение безопасности (следуя правилу, хранящемуся в файле /etc/togcheck/viotations.d/myfite), может быть проигнорировано только правилом в файле

/etc/togcheck/viotations.ignore .d/myfite или /etc /logcheck/viotations.ignore.d/myfite- файл *расширения*.

Вы всегда будете оповещены о системном событии, пока правило в одной из директорий /etc/togcheck/ignore.d. {paranoid, server,workstation}/ не утвердит, что событие должно быть проигнорировано. Разумеется, единственными воспринимаемыми директориями будут являться те, чьи соответственные уровни словесного наполнения равняются или превышают выбранный режим работы.

7.5.2 Мониторинг активности в реальном времени

top является интерактивным инструментом, отображающим список текущих запущенных процессов. Сортировка по умолчанию основана на текущей загруженности процессора и может быть получена с помощью ключа P. Другие сортировки приказов включают сортировку по занимаемой памяти (ключ M), общему времени процессора (ключ T) и идентификатору процесса (ключ N). Ключ K завершает процесс, путем ввода идентификатор процесса. Клавиша r изменяет приоритет процесса.

Когда вам кажется, что система перегружена, top является отличным инструментом для просмотра и определения какие процессы конкурируют за процессорное время или потребляют слишком много памяти. В частности, всегда интересно проверить соответствуют ли процессы, потребляющие ресурсы, реальным службам, которые должны быть размещены на машине. Неизвестный процесс, работающий как ”www-data”, должен действительно выделяться и изучаться, поскольку он, вероятнее всего, является экземпляром программного обеспечения, установленного и выполняемого в системе, с помощью уязвимости в веб-приложении.

Top является очень гибким инструментом и его страница руководства предоставляет детали о том, как настроить его интерфейс и адаптировать его под ваши необходимости и привычки.

Графический инструмент gnome-system-monitor является очень похожим на top и предоставляет те же самые свойства и функции.

7.5.3 Обнаружение изменений

После установки и настройки системы большинство системных файлов должны оставаться относительно статичными до тех пор, пока система не будет обновлена. Поэтому рекомендуется следить за изменениями в системных файлах, поскольку любое непредвиденное изменение может быть причиной тревоги и должно быть исследовано. В этом разделе представлены некоторые из наиболее распространенных инструментов, используемых для мониторинга системных файлов, обнаружения изменений и, вероятно, уведомления вас как администратора системы.

Проверка пакетов с помощью dpkg –verify

dpkg --verify (или dpkg -V) - интересный инструмент, поскольку он отображает системные файлы, которые были изменены (скорее всего, злоумышленником), но этот вывод следует воспринимать с определенной долей скепсиса. Для выполнения своей работы dpkg полагается на контрольные суммы хранящиеся в своих собственных базах данных, которые в свою очередь находятся на жестком диске (могут быть найдены в /var/lib/dpkg/info/package .md5sums). Ввиду этого довольно тщательный злоумышленник будет изменять эти файлы, чтобы они содержали новые контрольные суммы для поврежденных файлов, или же более продвинутый злоумышленник может взломать пакет в вашем зеркале Debian. Для того чтобы защититься от этого класса атаки, используйте систему верификации цифровой подписи APT (см. Раздел 8.3.6 «Проверка подлинности пакета» [стр. 202]) для правильной проверки пакетов.

Что такое контрольная сумма файла?

Мы считаем нужным напомнить, что контрольная сумма является величиной, чаще всего числом (хотя и в шестнадцатеричной

системе исчисления), которая содержит что-то вроде подписи для содержимого файла. Подпись рассчитывается алгоритмом (MD5 или SHA1 являются хорошо известными примерами), который более или менее гарантирует, что даже самые незначительные изменения содержимого файла приведут к изменению контрольной суммы; Это явление известно как «эффект лавины». Простая цифровая сигнатура затем служит средством для проверки того, изменилось ли содержимое файла или нет. Эти алгоритмы являются не обратимыми; другими словами, для большинства из них, даже если вы даже знаете контрольную сумму, то это не позволит вам найти соответствующее содержимое. Недавние математические достижения, по-видимому, ослабили абсолютность этих принципов, но их использование до сих пор не ставится под сомнение, поскольку создание другого содержимого, дающего одну и ту же контрольной суммы, по-прежнему представляется довольно сложной задачей.

Запуск команды dpkg -V проверяет все установленные пакеты и выводит на экран строку для каждого файла, который не прошел проверку. Каждый символ обозначает проверку на конкретные метаданные. К сожалению, dpkg не хранит метаданные необходимые для большинства тестов и таким образом выводит вопросительные знаки вместо них. В настоящее время если проверка контрольной суммы провалилась, то на третьей позиции будет находиться цифра 5.

```
# dpkg -V
??5?????? /lib/systemd/system/ssh.service
??5?????? c /etc/libvirt/qemu/networks/default.xml
??5?????? c /etc/lvm/lvm.conf
??5?????? c /etc/salt/roster
```

В приведенном выше примере, dpkg сообщает об изменении файла службы SSH, который администратор сделал в пакетированном файле вместо того, чтобы использовать соответствующую замену /etc/systemd/system/ssh.service (которая буден храниться ниже /etc как и должны храниться любые изменения конфигурации). В нем также перечислены несколько файлов конфигурации (обозначенных буквой «c» на втором поле), которые были легально изменены.

Мониторинг файлов: AIDE

Инструмент Advanced Intrusion Detection Environment (AIDE) проверяет целостность файла и обнаруживает любые изменения, которые не соответствуют ранее записанному образу действительной системы. Образ хранится в виде базы данных (/var/lib/aide/aide.db), содержащей соответствующую информацию обо всех файлах системы (контрольные суммы, разрешения, временные метки и т. д.).

Вы можете установить AIDE путем запуска apt update, за которой должна следовать apt install aide. Сначала вы инициализируете базу данных с помощью aideinit; она будет запускаться ежедневно (через сценарий /etc/cron.daily/aide), чтобы проверить, что за это время не произошло существенных изменений. Если изменения будут обнаружены, то AIDE записывает их в файлы журнала (/var/log/aide/*.log) и отправляет свои результаты администратору по электронной почте.

Защита базы данных

Поскольку AIDE использует локальную базу данных для сравнения состояний файлов, достоверность подобных действий напрямую связана достоверностью базы данных. Если злоумышленник получает права root на взломанную систему, то он сможет заменить базу данных и скрыть следы взлома. Одним из способов предотвращения подобного рода деятельности является сохранение справочных данных на носителе, предназначенном только для чтения.

Вы можете использовать опции в /etc/default/aide для настройки пакета package. AIDE внутренние настройки программы хранятся в файлах/etc/aide/aide.conf и /etc/aide/aide.conf.d/ (на самом деле, эти файлы используются только update-aide.conf для генерации /var/lib/aide/aide.conf. autogenerated). Конфигурация указывает, какие свойства должны быть проверены. Например, содержимое файлов журнала изменяется в обычном режиме, и такие изменения можно игнорировать, если разрешения этих

файлов остаются неизменными, но как содержимое, так и разрешения исполняемых программ должны быть постоянными. Хотя все это и не очень сложно, синтаксис конфигурации не является полностью интуитивно ясным, и мы рекомендуем прочитать дополнительную справочную страницу aide.conf (5) для получения дополнительной информации.

Новая версия базы данных создается ежедневно в /var/lib/aide/aide.db.new; если все записанные изменения были законными, то смело можно выполнять замену базы данных.

Инструмент Tripwire очень похож на AIDE; даже синтаксис файла конфигурации почти одинаковый. Основное дополнение, предоставляемое *tripwire,* заключается в том, что он включает в себя механизм подписи файла конфигурации, чтобы злоумышленник не мог заставить его указывать на другую версию справочной базы данных.

Samhain также предлагает похожие свойства, а также некоторые функции, которые помогут определить руткиты (смотри вставку "Пакеты ***checksecurity*** и ***chkrootkit/rkhunter*** " [стр. 164]). Он также может быть развернут глобально во всей сети и записывать результаты своей работы на центральном сервере (с подписью).

Пакеты *checksecurity* и *chkrootkit/rkhunter*

checksecurity состоит из нескольких небольших скриптов, которые выполняют основные проверки в системе (поиск пустых паролей, новых файлов setuid и т. д.) и оповещает вас в случае обнаружении этих условий. Несмотря на свое явное имя, вы не должны полагаться исключительно на него, для того чтобы удостовериться, что система Linux безопасна.
Пакеты *chkrootkit* и *rkhunter* обнаруживают определенные *руткиты*, потенциально установленные в системе. Напомним, что это части программного обеспечения, предназначенные для скрытия взлома системы, но при этом сохраняя контроль над машиной. Тесты не на 100 процентов надежны, но обычно их результаты могут привлечь ваше внимание к потенциальным проблемам.

7.6 Подведем итоги

В этой главе мы рассмотрели концепцию политик безопасности, подчеркнув различные моменты, которые следует учитывать при определении подобно политики, и обсудили некоторые угрозы вашей системе и лично вам как специалисту в сфере безопасности. Мы также подняли вопрос о мерах безопасности для ноутбуков и десктопных систем, а также о брандмауэрах и фильтрационных пакетах. Наконец, мы рассмотрели инструменты и стратегии мониторинга и показали, как наилучшим образом реализовать их для обнаружения потенциальных угроз для вашей системы.

Основные моменты:

- Потратьте какое-то время для определения четкой и всеобъемлющей политики безопасности.
- Если вы используете Kali на общедоступном сервере, измените все пароли по умолчанию для служб, которые могут быть настроены (см. Раздел 7.3 «Защита сетевых служб» [стр. 153]) и ограничьте их доступ с помощью брандмауэра (см. Раздел 7.4, «Брандмауэр или фильтрация пакетов» [стр. 153]) перед их запуском.
- Используйте fail2ban для обнаружения и блокировки атак угадывания пароля и brute force атак.
- Если вы запускаете веб-службы, размещайте их на HTTPS, чтобы сетевые посредники не могли просматривать ваш трафик (который может содержать в себе файлы cookie для аутентификации).
- Реальные риски чаще всего возникают, когда вы путешествуете от одного клиента к другому. Например, ваш ноутбук может быть украден во время подобного рода поездки или изъят на таможне. Будьте всегда готовыми к подобным неприятным неожиданностям и используйте полное шифрование диска (смотри раздел 4.2.2., «Установка на полностью зашифрованную файловую систему» [стр. 85]), а также не забудьте рассмотреть функцию nuke (смотри “Добавление nuke пароля для дополнительной безопасности” [стр. 245]) для того, чтобы защитить данные вашего клиента.
- Необходимо внедрить правила брандмауэра (см. Раздел 7.4,

«Брандмауэр или фильтрация пакетов» [стр. 153]), чтобы запретить весь исходящий трафик, кроме трафика, генерируемого вашим VPN-доступом. Это подобно защитной сетке, поэтому, когда VPN отключается, вы сразу замечаете это (вместо того, чтобы возвращаться к локальному сетевому доступу).

- Заблокируйте службы, которые вы не используете. Kali делает эту процедуру намного проще, т.к. все внешние сетевые службы отключены по умолчанию.
- В ядро Linux встроен ***netfIter*** брандмауэр. Не существует окончательного решения вопроса настройки любого брандмауэра, т.к. требования сети и пользователя довольно разнятся. Тем не менее, вы можете контролировать ***netfIter*** из пользовательского пространства с помощью команд iptables и ip6tables.
- Программа logcheck отслеживает файлы журнала каждый час по умолчанию и отправляет электронные письма с особыми сообщениями журнала администратору для дальнейшего анализа.
- top является интерактивным инструментом, который выводит на экран список запущенных процессов на данный момент.
- dpkg --verify (или dpkg -V) отображает системные файлы, которые были изменены (скорее всего злоумышленником), но полагается на контрольные суммы, которые могут быть искажены грамотным атакующим.
- Инструмент Advanced Intrusion Detection Environment (AIDE) проверяет целостность файла и определяет любые изменения в отношении ранее записанного образа действительной системы.
- Tripwire является очень похожим на AIDE, но он использует механизм для подписи файла конфигурации, чтобы злоумышленник не мог указать на другую версию справочной базы данных.
- Рассмотрите использование rkhunter, checksecurity, и chkrootkit для получения помощи в обнаружении руткитов на вашей системе.

В следующей главе мы рассмотрим основные моменты Debian и управление пакетами. Вы быстро осознаете всю силу, лежащую в основе корней Debian Kali, и узнаете, как разработчики использовали эту мощь. Будьте осторожны, следующая глава довольно насыщенная, но крайне важно, чтобы вы понимали

основы Debian и управление пакетами, если вы собираетесь стать уверенным пользователем Kali.

Часть 8: Управление пакетами Debian

Содержание:

Ключевые слова главы:

- Dpkg;
- aptsources.list;
- Обновления;
- Пакетные репозитории;

После того, как вы ознакомились с основами Linux, пришло время изучить систему управления пакетами дистрибутива на базе Debian. В таких дистрибутивах, включая Kali, пакет Debian представляет собой канонический способ сделать программное обеспечение доступным для конечных пользователей. Понимание системы управления пакетами даст вам представление о том, каким образом структурирована Kali, позволит вам более эффективно решать проблемы и быстро находить помощь и документацию для широкого спектра инструментов и утилит, включенных в Kali Linux.

В этой главе мы представим вашему вниманию систему управления пакетами Debian и познакомим вас с dpkg и набором инструментов APT. Одной из основных преимуществ Kali Linux является гибкость системы управления пакетами, которая использует эти инструменты для обеспечения практически непрерывной инсталляции, обновления, удаления и обработки прикладного программного обеспечения и даже самой базовой операционной системы. Очень важно понять, как эта система работает, чтобы максимально использовать Kali и оптимизировать ваши усилия. Дни болезненных компиляций, провальных обновлений, отладки gcc, долгого создания и настройки различных опций давно прошли, однако, количество доступных приложений значительно выросло и сейчас вам необходимо понимать инструменты, разработанные для их использования. Этот навык также является необходимым, т.к. существует огромное количество инструментов безопасности, которые по причине лицензирования или из-за других нюансов не могут быть включены в Kali, но имеют пакеты Debian для скачивания. Очень важно, чтобы вы знали, как обрабатывать и устанавливать эти пакеты и понимать, как они влияют на систему, особенно, в тех случаях, когда все идет не так как ожидалось.

Мы начнем с базового обзора APT, опишем структуру и содержимое двоичных и исходных пакетов, посмотрим на некоторые базовые инструменты и сценарии и затем углубимся в изучение для того, чтобы помочь вам выжать максимум из этой эффективной пакетной системы и набора инструментов.

8.1 Введение в APT

Давайте начнем с некоторых базовых определений, общего обзора, и небольшой истории пакетов Debian, начиная наше повествование с dpkg и APT.

8.1.1 Взаимосвязь между APT и dpkg

Пакет Debian представляет собой сжатый архив программного приложения. А ***бинарный пакет (binary package*** (файл .deb)) содержит файлы, которые могут быть прямо использованы (такие как программы или документации), в то время как ***исходный пакет (source package)*** содержит исходный код для программного обеспечения, а также инструкции, которые необходимы для создания бинарных пакетов. Пакет Debian содержит файлы приложения также как и другие ***метаданные***, включая названия зависимостей, которые необходимы приложению и скрипты, которые разрешают выполнение команд на разных стадиях жизненного цикла пакета (установка, удаление и обновление).

Инструмент dpkg был создан для обработки и установки пакетов .deb, но если встречает зависимость, которая не может быть удовлетворена (вроде отсутствующей библиотеки), то это помешает установке пакетов. В подобных случаях dpkg просто перечислит отсутствующую зависимость, потому что у него просто нет вариантов действия или встроенной логики для обработки пакетов, которые должны удовлетворить эти зависимости. Инструмент The Advanced Package Tool (APT), включая apt и apt-get, были разработаны для устранения этих недостатков, и таким образом он может автоматически решить эти проблемы. В этой главе мы поговорим об инструментах dpkg и APT.

Базовой командой для обработки пакетов Debian в системе является dpkg, которая выполняет установку или анализ пакетов .deb и их содержимого. Тем не менее, dpkg имеет только частичное представление о вселенной Debian: он знает, что установлено в системе и что вы предоставляете в командной строке, но ничего не знает о других доступных пакетах. Таким образом, он не будет

работать, если зависимость не будет выполнена. APT устраняет данные ограничения.

APT является набором инструментов, которые помогают управлять пакетами Debian или приложениями в вашей системе Debian. Вы можете использовать APT для установки и удаления приложений, обновления пакетов и даже обновления всей системы. Вся магия APT заключается в том, что он является полноценной системой управления пакетов, которая будет не просто устанавливать или удалять пакеты, но также будет рассматривать требования и зависимости пакетированного приложения (и даже их требования и зависимости) и пытаться удовлетворить их все автоматически. APT полагается на dpkg, но не смотря на это отличается от dpkg. APT устанавливает последнюю версию пакета из онлайн источника и работает так, чтобы разрешить зависимости, в то время как dpkg устанавливает пакет, расположенный на вашей локальной системе, и не разрешает зависимости автоматически.

Если вы работаете в данной сфере достаточно долго, чтобы помнить о компиляции программ с помощью gcc (даже с помощью утилит, таких как make и configure), вы, вероятно, помните, что это был довольно болезненный процесс, особенно, если приложение имело несколько зависимостей. Дешифруя различные предупреждения и сообщения об ошибках, вы могли определить, какая часть кода была неудачной, и чаще всего эта неудача была вызвана отсутствующей библиотекой или другой зависимостью. Затем вы отслеживали эту недостающую библиотеку или зависимость, исправляли ее и повторяли попытку. Далее, если вам повезет, компиляция должна завершиться, но часто сборка снова потерпит неудачу, жалуясь на другую нарушенную зависимость.

APT был разработан для того, чтобы помочь решить эту проблему, сопоставить программные требования и зависимости, а также решить их. Эта функциональность работает по умолчанию на Kali Linux, но она не является защищённой от неумелого обращения. Важно понимать, как работает система пакетирования Debian и Kali, потому что вам нужно будет устанавливать пакеты, обновлять программное обеспечение или устранять проблемы, связанные с пакетами. Вы будете использовать APT в своей повседневной работе с Kali Linux, и в этой главе мы познакомим вас с APT и

покажем вам, как устанавливать, удалять, обновлять и управлять пакетами и даже покажем вам, как перемещать пакеты между разными дистрибутивами Linux. Мы также поговорим о графических инструментах, которые используют APT, покажем вам, как проверять подлинность пакетов, и углубимся в концепцию rolling distribution - метод, который ежедневно обновляет вашу систему Kali.

Прежде чем мы копнем глубже и покажем вам, как использовать dpkg и APT для установки и управления пакетами, очень важно, чтобы мы углубились в некоторые внутренние действия APT и обсудили некоторую терминологию, используемую в нем.

Источник пакета и исходный пакет (Package Source and Source Package)

Слово *исходный/источник* (*source)* может быть двусмысленным. Исходный пакет (source package)—является пакетом, который содержит исходный код программы – не следует путать с источником пакета (package source) – репозиторием (веб-сайтом, FTP сервером, CD-ROM, локальной директорией, и т.д.), который содержит пакет.

APT извлекает свои пакеты из репозитория, хранилища пакетов или просто, ”источника пакета”. Файл /etc/apt/sources.list перечисляет различные репозитории (или источники), которые содержат пакеты Debian.

8.1.2 Правильное понимание sources.list файла

Файл sources.list является ключевым файлом конфигурации для определения источника пакетов, и поэтому очень важно понимать, как он разбивается и как его настраивать, т.к. APT не будет работать без правильно определенного списка источника пакетов. Давайте обсудим его синтаксис. Для начала мы взглянем на различные репозитории, которые используются Kali Linux, и обсудим зеркала и зеркальные перенаправления, и только после этого вы будете готовы к использованию APT.

Каждая активная строчка файла /etc/apt/sources.list (и файлов /etc/apt/sources.list.d/*. list) содержит описание источника, сделанного из трех частей, разделенных пробелами. Комментируемые строки начинаются с символа #:

```
# deb cdrom:[Debian GNU/Linux 2016.1 _Kali-rolling_ - Official Snapshot amd64 LIVE/
    ➥ INSTALL Binary 20160830-11:29]/ kali-rolling contrib main non-free

deb http://http.kali.org/kali kali-rolling main non-free contrib
```

Давайте взглянем на синтаксис этого файла. Первое поле обозначает тип источника:

- deb для бинарных пакетов,
- deb-src для исходных пакетов.

Второе поле дает базовый URL-адрес источника: он может состоять из зеркала Debian или любого другого архива пакетов, настроенного третьей стороной. URL-адрес может начинаться с file:// для указания локального источника, установленного в иерархии файлов системы, с http: // для указания источника, доступного с веб-сервера, или с ftp:// для источника, доступного на FTP-сервере. URL-адрес также может начинаться с cdrom: для установки с CD-ROM/DVD-ROM/ Blu-ray, хотя этот способ становится реже используемым, поскольку методы установки из сети становятся более распространенными.

Информация cdrom описывает ваши устройства CD/DVD-ROM. В отличие от других записей, CD-ROM не всегда бывает доступен, так как он должен быть вставлен в привод, и, как правило, за один раз можно считывать информацию только с одного диска. По этим причинам эти источники управляются несколько иначе и должны быть добавлены программой apt-cdrom, обычно выполняемой с помощью параметра add. Затем вас попросят вставить диск в дисковод, где его содержимое будет просматриваться в поисках файла пакетов. Он будет использовать эти файлы для обновления своей базы данных пакетов (эта операция обычно выполняется командой apt update). После этого APT запросит дополнительный диск, если ему нужен пакет, хранящийся на нем.

Синтаксис последнего поля зависит от структуры репозитория. В самых простых случаях вы можете просто указать суб-директорию (с необходимым знаком слэша) желаемого источника (это чаще всего обычный знак "./", который означает отсутствие суб-директории – в таком случае пакеты находятся непосредственно по указанному URL). Но в большинстве случаев репозитории будут структурированы, как зеркало Debian, с множество дистрибутивов, каждый из которых обладает большим количеством компонентов. В этих случаях назовите выбранный дистрибутив, а затем компоненты (или разделы), которые нужно включить. Давайте выделим еще немного времени на представление этих разделов.

Debian и Kali используют три раздела для дифференцирования пакетов согласно лицензии, выбранной авторами каждой работы.

Main содержит все пакеты, которые соответствуют генеральной линии бесплатного программного обеспечения Debian (Debian Free Software Guidelines[27]).

Архивы, которые являются non-free, довольно сильно отличаются, потому что они содержат программное обеспечение, которое не (полностью) соответствует данным принципам, но которое, тем не менее, может распространяться без ограничения.

Contrib (contributions) представляет собой набор программ с открытым исходным кодом, которые не могут функционировать без некоторых non-free элементов. Эти элементы могут включать программное обеспечение из раздела non-free или non-free файлов, таких как игровые ПЗУ, BIOS консолей и т. д. Contrib также включает в себя бесплатное программное обеспечение, для компиляции которого требуются патентованные элементы, такие как VirtualBox, который в свою очередь требует non-free компилятор для создания некоторых своих файлов.

Теперь давайте посмотрим на стандартные источники пакетов Kali Linux или репозитории.

[27]https://www.debian.org/social_contract#guidelines

8.1.3 Репозитории Kali

Стандартный файл sources.list для системы, работающей на Kali Linux, относится к одному репозиторию (kali-rolling) и трем ранее упомянутым компонентам: main, contrib и non-free:

```
# Main Kali repository
deb http://http.kali.org/kali kali-rolling main contrib non-free
```

Давайте познакомимся с различными Kali репозиториями.

Kali-Rolling репозиторий

Это основной репозиторий для конечных пользователей. Он всегда должен содержать самые новые и устанавливаемые пакеты. Он управляется инструментом, который объединяет Debian Testing и особые пакеты Kali, таким образом, гарантируя, что зависимости каждого пакета могут быть удовлетворены в kali-rolling. Другими словами, исключая вероятность любой ошибки в сценариях поддержки, все пакеты должны быть установлены.

Ввиду того, что Debian Testing развивается ежедневно, точно также эволюционирует и Kali Rolling. Особые Kali пакеты также регулярно обновляются, поскольку мы контролируем своевременные выпуски самых важных пакетов.

Kali-Dev репозиторий

Данный репозиторий не предназначен для общего пользования. Это пространство, в котором разработчики Kali решают проблемы зависимостей, возникающие в результате слияния особых пакетов Kali в Debian Testing.

Это также место, куда сначала загружаются обновленные пакеты, поэтому, если вам нужно обновление, которое было выпущено

недавно, но которое еще не достигло kali-rolling, вы можете получить его из этого репозитория. Это не рекомендуется для обычных пользователей.

Репозиторий Kali-Bleeding-Edge

Этот репозиторий содержит пакеты, автоматически созданные из соответствующего репозитория Git (или Subversion). Положительным моментом является то, что вы сразу же получаете доступ к последним функциям и исправлениям ошибок менее, чем через 24 часа после того, как они были сделаны. Это идеальный способ проверить, исправлена ли ошибка, о которой вы ранее сообщали.

Недостатком является то, что эти пакеты не были протестированы или проверены: если соответственные изменения повлияли на пакетирование (добавив новую зависимость), тогда этот пакет может не работать. Из-за этого репозиторий помечен таким образом, что APT автоматически не устанавливает пакеты из него, особенно, во время обновления.

Вы можете зарегистрировать репозиторий либо путем редактирования /etc/apt/sources.list или путем создания нового файла в директории /etc/apt/sources.list.d, который имеет преимущество при выходе из исходной системы sources.list file un –altered. В данном примере мы предпочли создать отдельный файл /etc/apt/ sources.list.d/kali-bleeding-edge.list следующим образом:

```
# Kali Bleeding Edge repository
deb http://http.kali.org/kali kali-bleeding-edge main contrib non-free
```

Зеркала Kali Linux

Выдержки из sources.list, указанные выше, относятся к http.kali.org: это сервер, на котором работает MirrorBrain[2],

который перенаправляет ваши HTTP-запросы в официальное зеркало, находящееся рядом с вами. MirrorBrain контролирует каждое зеркало, чтобы гарантировать, что они работают должным образом и являются обновленными; он всегда перенаправит вас на хорошее зеркало.

Отладка перенаправления зеркал Если у вас возникает проблема с зеркалом (например, из-за неудачного обновления apt), вы можете использовать curl -sI, чтобы увидеть, куда конкретно вы перенаправляетесь:

```
$ curl -sI http://http.kali.org/README
HTTP/1.1 302 Found
Date: Mon, 11 Apr 2016 09:43:21 GMT
Server: Apache/2.4.10 (Debian)
X-MirrorBrain-Mirror: ftp.free.fr
X-MirrorBrain-Realm: country
Link: <http://http.kali.org/README.meta4>; rel=describedby;
    ➥ type="application/metalink4+xml"
Link: <http://ftp.free.fr/pub/kali/README>; rel=duplicate;
    ➥ pri=1; geo=fr
Link: <http://de-rien.fr/kali/README>; rel=duplicate; pri=2;
    ➥ geo=fr
Link: <http://ftp.halifax.rwth-aachen.de/kali/README>; rel=
    ➥ duplicate; pri=3; geo=de
Link: <http://ftp.belnet.be/kali/kali/README>; rel=duplicate;
    ➥  pri=4; geo=be
Link: <http://ftp2.nluug.nl/os/Linux/distr/kali/README>; rel=
    ➥ duplicate; pri=5; geo=nl
Location: http://ftp.free.fr/pub/kali/README
Content-Type: text/html; charset=iso-8859-1
```

Если проблема сохраняется, вы можете отредактировать /etc/apt/sources.Ust имя другого известного рабочего зеркала вместо (или до) записи http.kati.org

У нас также есть второй экземпляр MirrorBrain: где http.kali.org размещает репозитории пакетов, а cdimage.kali.org размещает выпущенные ISO-образы.
http://cdimage.kali.org

Если вы хотите запросить список официальных зеркал Kali Linux, вы можете добавить .mirrorlist в любой допустимый URL-адрес, указывающий на http.kali.org или cdimage.kali.org.

http://http.kali.org/README.mirrorlist
http://cdimage.kali.org/README.mirrorlist

Эти списки не являются исчерпывающими из-за некоторых ограничений MirrorBrain (в частности, зеркала, зависящие от некоторых стран, не отображаются в списке, если вы не находитесь в данной стране). Но они содержат лучшие зеркала: они в хорошем состоянии и имеют большую пропускную способность.

8.2 Основное взаимодействие пакетов

Теперь вооружившись базовым пониманием ландшафта APT, давайте рассмотрим некоторые базовые взаимодействия пакетов, включая инициализацию APT; установку, удаление и очистку пакетов; и модернизацию системы Kali Linux. Затем давайте займемся командной строкой для того, чтобы взглянуть на некоторые графические инструменты APT.

8.2.1 Инициализация APT

APT является обширным проектом и набором инструментов, чьи первоначальные планы включали графический интерфейс. С точки зрения клиента, он сосредоточен вокруг инструмента командной строки apt-get, а также apt, который позднее был разработан для преодоления недостатков дизайна apt-get.

Существуют графические альтернативы, разработанные третьими сторонами, в том числе synaptic и aptitude, которые мы обсудим немного позже. Мы предпочитаем использовать apt, которая будет использована во всех последующих примерах. Однако, мы также подробно изложим некоторые из основных различий синтаксиса между инструментами по мере их возникновения.

При работе с APT вы должны сначала загрузить список доступных пакетов с обновлением apt. В зависимости от скорости вашего подключения это может занять некоторое время, потому что

список различных пакетов, список источников и файлы переводов выросли в размере наряду с разработкой Debian. Конечно, сборки для установки с CD/DVD инсталлируются намного быстрее, потому что они являются локальными для вашей машины.

8.2.2 Установка пакетов

Благодаря продуманному дизайну системы пакетов Debian вы можете легко устанавливать пакеты с или без их зависимостей. Давайте посмотрим на установку пакета с помощью dpkg и apt.

Установка пакетов с помощью dpkg

---

dpkg является основным инструментом, который вы будете использовать (прямо или косвенно через APT), когда вам нужно установить пакет. Это также отличный выбор, если вы работаете в автономном режиме, поскольку для него не требуется подключение к Интернету. Помните, что dpkg не установит никаких зависимостей, которые могут потребоваться для пакета. Чтобы установить пакет с dpkg, просто укажите параметр -i или -install и путь к .deb. В данном случае подразумевается, что вы ранее загрузили (или получили каким-то другим способом) файл .deb пакета для установки.

```
# dpkg -i man-db_2.7.0.2-5_amd64.deb
(Reading database ... 86425 files and directories currently installed.)
Preparing to unpack man-db_2.7.0.2-5_amd64.deb ...
Unpacking man-db (2.7.0.2-5) over (2.7.0.2-4) ...
Setting up man-db (2.7.0.2-5) ...
Updating database of manual pages ...
Processing triggers for mime-support (3.58) ...
```

Мы можем видеть различные шаги, выполняемые dpkg, и соответственно можем видеть, в какой момент может произойти какая-либо ошибка. Параметр -i или -install выполняет два этапа автоматически: он распаковывает пакет и запускает сценарии конфигурации. Вы можете выполнить эти два шага

самостоятельно (как это обычно делает apt) с параметрами -unpack и -configure, соответственно:

```
# dpkg --unpack man-db_2.7.0.2-5_amd64.deb
(Reading database ... 86425 files and directories currently installed.)
Preparing to unpack man-db_2.7.0.2-5_amd64.deb ...
Unpacking man-db (2.7.0.2-5) over (2.7.0.2-5) ...
Processing triggers for mime-support (3.58) ...
# dpkg --configure man-db
Setting up man-db (2.7.0.2-5) ...
Updating database of manual pages ...
```

Обратите внимание, что строки «Триггеры обработки» относятся к коду, который выполняется автоматически, когда пакет добавляет, удаляет или изменяет файлы в некоторых контролируемых директориях. Например, пакет ***mime- support*** контролирует usr/lib/mime/packages и выполняет команду update-mime всякий раз, когда что-то изменяется в этой директории (например, /usr/lib/mime/packages/man-db в конкретном случае man-db).

Иногда dpkg не сможет установить пакет и выдает ошибку. Тем не менее, вы можете приказать dpkg, игнорировать это, и только выдать предупреждение с различными параметрами -force- *. Вывод команды dpkg --force- help отобразит полный список этих параметров. Например, вы можете использовать dpkg для принудительной установки zsh:

```
$ dpkg -i --force-overwrite zsh_5.2-5+b1_amd64.deb
```

Частая ошибка, с которой вы рано или поздно столкнетесь, - это конфликт файлов. Когда пакет содержит файл, который уже установлен другим пакетом, dpkg откажется его установить. Появятся следующие типы сообщений:

```
Unpacking libgdm (from .../libgdm_3.8.3-2_amd64.deb) ...
dpkg: error processing /var/cache/apt/archives/libgdm_3.8.3-2_amd64.deb (--unpack):
    ➥ trying to overwrite '/usr/bin/gdmflexiserver', which is also in package gdm3
    ➥ 3.4.1-9
```

В этом случае, если вы считаете, что замена этого файла не является значительным риском для стабильности вашей системы (как правило, это так), вы можете использовать -force-overwrite для перезаписывания файла.

Хотя существует множество доступных параметров -force- *, можно использовать только -force-overwrite. Эти варианты существуют для исключительных ситуаций, и лучше оставить их в покое как можно на дольше, чтобы соблюдать правила, установленные механизмом пакетирования. Не забывайте, что эти правила обеспечивают согласованность и стабильность вашей системы.

Установка пакетов с помощью APT

Хотя APT является намного более продвинутым, чем dpkg, и выполняет гораздо больше работы, вы обнаружите, что его взаимодействие с пакетами довольно простое. Вы можете добавить пакет в систему с помощью простой команды apt install *package***.** APT автоматически установит необходимые зависимости:

```
# apt install kali-linux-gpu
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following additional packages will be installed:
  oclgausscrack oclhashcat
The following NEW packages will be installed:
  kali-linux-gpu oclgausscrack oclhashcat
0 upgraded, 3 newly installed, 0 to remove and 416 not upgraded.
Need to get 2,494 kB of archives.
After this operation, 51.5 MB of additional disk space will be used.
Do you want to continue? [Y/n]
Get:1 http://archive-2.kali.org/kali kali-rolling/non-free amd64 oclhashcat amd64 2.01+
    ➥ git20160114-0kali2 [2,451 kB]
Get:2 http://archive-2.kali.org/kali kali-rolling/main amd64 oclgausscrack amd64 1.3-1
    ➥ kali2 [37.2 kB]
Get:3 http://archive-2.kali.org/kali kali-rolling/main amd64 kali-linux-gpu amd64
    ➥ 2016.3.2 [6,412 B]
Fetched 2,494 kB in 0s (3,060 kB/s)
Selecting previously unselected package oclhashcat.
(Reading database ... 317084 files and directories currently installed.)
Preparing to unpack .../0-oclhashcat_2.01+git20160114-0kali2_amd64.deb ...
Unpacking oclhashcat (2.01+git20160114-0kali2) ...
Selecting previously unselected package oclgausscrack.
Preparing to unpack .../1-oclgausscrack_1.3-1kali2_amd64.deb ...
Unpacking oclgausscrack (1.3-1kali2) ...
Selecting previously unselected package kali-linux-gpu.
Preparing to unpack .../2-kali-linux-gpu_2016.3.2_amd64.deb ...
Unpacking kali-linux-gpu (2016.3.2) ...
Setting up oclhashcat (2.01+git20160114-0kali2) ...
Setting up oclgausscrack (1.3-1kali2) ...
Setting up kali-linux-gpu (2016.3.2) ...
```

Вы также можете использовать apt-get install *package,* или aptitude install *package.* Для простой установки пакетов они делают практически то же самое. Как вы увидите позже, отличия будут более заметны по отношению к обновлениям или когда разрешение зависимостей не имеет идеального решения.

Если перечисляет несколько дистрибутивов, вы можете указать версию пакета с помощью apt install *package=version,* но всегда желательно указать происхождение дистрибутива (kali-rolling, kali-dev, or kali-bleeding- edge) с помощью apt install *package/distribution.*

Как и в случае с dpkg, вы можете поручить apt принудительно установить пакет и перезаписать файлы с помощью -force-

overwrite, но синтаксис в этом случае будет выглядеть немного странно, поскольку вы передаете аргумент через dpkg:

```
# apt -o Dpkg::Options::="--force-overwrite" install zsh
```

8.2.3 Обновление Kali Linux

В качестве rolling дистрибутива Kali Linux обладает впечатляющими возможностями обновления. В этом разделе мы рассмотрим, каким образом вы можете просто обновить Kali, а также обсудим стратегии планирования ваших обновлений.

Мы рекомендуем использовать регулярные обновления, так как они будут устанавливать последние обновления безопасности. Чтобы начать процесс обновления, используйте apt update, за которым следуют apt upgrade, apt-get upgrade или aptitude safe-upgrade. Эти команды ищут установленные пакеты, которые можно обновить без удаления каких-либо пакетов. Другими словами, цель состоит в том, чтобы обеспечить наименее навязчивое обновление насколько это возможно. Инструмент командной строки apt-get немного более требовательный, чем aptitude или apt, потому что он откажется устанавливать пакеты, которые не были установлены заранее.

Инструмент apt обычно выбирает самый последний номер версии (за исключением пакетов с kali¬bleeding-edge, которые по умолчанию игнорируются независимо от их номера версии).

Чтобы использовать конкретный дистрибутив при поиске обновленных пакетов, вам необходимо использовать параметр -t или -target-release, за которым следует имя нужного вам дистрибутива (например: apt -t kali-rotting upgrade). Чтобы избежать указания этой опции каждый раз, когда вы используете apt, вы можете добавить APT :: Default-Release «kali-roll», в файле /etc/apt/apt.conf.d/local.

Для более важных обновлений, таких как обновление основных версий, используйте apt full-upgrade. С помощью этой команды apt завершит обновление, даже если ему нужно удалить некоторые

устаревшие пакеты или установить новые зависимости. Это также команда, которую вы должны использовать для регулярных обновлений вашей системы Kali Rolling. Это настолько просто, что вряд ли требует каких-либо дополнительных объяснений: популярность APT как раз основана на этой замечательной функциональности.

В отличие от apt и aptitude, apt-get не знает команду full-upgrade. Вместо этого вы должны использовать apt-get dist-upgrade (обновление дистрибутива), известную команду, которую apt и aptitude также принимают для обратной совместимости.

Будьте всегда в курсе важных изменений

Чтобы предвидеть некоторые из этих проблем, вы можете установить пакет apt-listchanges, который отображает информацию о возможных проблемах в начале обновления пакета. Эта информация собирается составителями пакетов и помещается в /usr/share/doc/package/NEWS.Debian. Внимательное чтение этих файлов (возможно, через apt-listchanges) должно помочь вам избежать неприятных сюрпризов.

Ввиду того, что Kali является rolling дистрибутивом, он получает обновления несколько раз в день. Однако, это не всегда является лучшей стратегией. Итак, насколько часто вам следует обновлять Kali Linux. Безусловно, не существует каких-либо строгих правил, но есть некоторые основные принципы, которые, которые могут помочь вам. Вам следует выполнять обновления в следующих случаях:

- Когда вам известно о проблеме безопасности, исправленной в обновлении;
- Если вы подозреваете, что обновленная версия может исправить ошибку, с которой вы столкнулись;
- Прежде чем сообщать об ошибке для того, чтобы убедиться, что она все еще присутствует в доступной вам последней версии;
- Достаточно часто, чтобы получить обновления безопасности, о которых вы не слышали.
- Также существуют случаи, в которых лучше не выполнять обновление. Например, мы не рекомендуем выполнять обновления в следующих случаях:

- Если у вас не будет достаточно времени исправить возникшие неполадки (например, потому что вы будете офлайн или потому, что собираетесь подготовить презентацию на своем компьютере); лучше выполнить обновление позже, когда у вас будет достаточно времени для устранения проблемы, возникшей в процессе.
- Если в последнее время у вас произошло (или продолжается) нарушение работы, и вы опасаетесь, что все проблемы еще не разрешены. Например, при выпуске новой версии GNOME не все пакеты обновляются одновременно, и у вас, вероятно, будет сочетание пакетов старой и новой версий. В большинстве случаев это нормально, и это помогает всем выпускать эти обновления постепенно, но всегда существуют исключения, и работа некоторых приложений могут быть нарушена из-за таких несоответствий.
- Если вывод apt full-upgrade говорит вам, что пакеты, необходимые для вашей работы будут удалены. В подобных случаях вам следует рассмотреть ситуацию и попытаться понять, почему apt хочет их удалить. Возможно, пакеты в настоящее время повреждены, и, следовательно, вам может понадобиться подождать, пока будут доступны исправленные версии, или они просто-напросто устарели, и вы должны определить, чем их замены, а затем продолжить полное обновление.

В общем, мы рекомендуем вам обновлять Kali не реже одного раза в неделю. Вы можете, конечно, обновляться ежедневно, но мы считаем, что это не имеет смысла. Даже если зеркала синхронизируются четыре раза в день, обновления от Debian, обычно поступают только один раз в день.

8.2.4 Удаление и очистка пакетов

Удаление пакета еще проще, чем его установка. Давайте посмотрим, как удалить пакет с помощью dpkg и apt.

Чтобы удалить пакет с помощью dpkg, выставьте параметр -r или -remove, а затем имя пакета. Однако, на этом удаление не завершено: все файлы конфигурации, сценарии поддержки,

файлы журналов (системные журналы), данные, созданные демоном (например, содержимое каталога сервера LDAP или содержимое базы данных для SQL-сервера), и большинство других данных пользователя, обрабатываемых пакетом, остаются нетронутыми. Опция remove позволяет легко удалить программу, а затем переустановить ее с той же конфигурацией. Также помните, что зависимости не удаляются. Рассмотрим этот пример:

```
# dpkg --remove kali-linux-gpu
(Reading database ... 317681 files and directories currently installed.)
Removing kali-linux-gpu (2016.3.2) ...
```

Вы также можете удалить пакеты из системы с помощью команды apt remove *package.* APT автоматически удалит пакеты, которые зависят от уже удаленных пакетов. Как и в примере dpkg, файлы конфигурации и данные пользователя не будут удалены.

С помощью добавления суффиксов к именам пакетов вы можете использовать apt (или apt-get и aptitude) для установки определенных пакетов и удаления других в той же командной строке. Используя команду apt install добавьте "-" к имени пакетов, которые вы хотите удалить. С помощью команды apt remove вы можете добавить "+" к имени пакетов, которые вы хотите установить.

Следующий пример показывает два различных способа, установки *package1* и удаления *package2*.

```
# apt install package1 package2-
[...]
# apt remove package1+ package2
[...]
```

Этот способ также можно использовать для исключения пакетов, которые в противном случае были бы установлены, например, ввиду рекомендаций (мы обсудим данный вопрос немного позже). В общем, решатель зависимостей будет использовать эту информацию в качестве подсказки для поиска альтернативных решений.

Для удаления всех данных связанных с каким-либо пакетом, вы можете очистить пакет с помощью команд dpkg -P *package,* или apt purge *package*. Эти команды полностью удалят пакет и все данные пользователя, и в случае с apt также удалит и все зависимости.

```
# dpkg -r debian-cd
(Reading database ... 97747 files and directories currently installed.)
Removing debian-cd (3.1.17) ...
# dpkg -P debian-cd
(Reading database ... 97401 files and directories currently installed.)
Removing debian-cd (3.1.17) ...
Purging configuration files for debian-cd (3.1.17) ...
```

Предупреждение! Учитывая окончательный характер чистки, дважды подумайте, прежде чем прибегать к нему. Вы потеряете все, что связано с этим пакетом.

8.2.5 Проверка пакетов

Далее, давайте познакомимся с некоторыми инструментами, которые могут быть использованы для проверки пакетов Debian. Мы познакомимся с dpkg, apt, и apt-cache командами, которые могут использоваться для запроса и визуализации базы данных пакета.

Запрос базы данных **dpkg** и проверка **.deb** файлов

Начнем с нескольких опций dpkg, которые запрашивают внутреннюю базу данных dpkg. Эта база данных находится в файловой системе в /var/lib/dpkg и содержит несколько разделов, включая сценарии конфигурации (/var/lib/dpkg/info), список файлов, в которых установлен пакет (/var/lib/dpkg/info/ *. list) и статус каждого установленного пакета (/var/lib/dpkg/status). Вы можете использовать dpkg для взаимодействия с файлами в этой базе данных. Обратите внимание, что большинство параметров доступны как в длинной версии (одно или несколько релевантных слов, которым предшествует двойное тире), так и в короткой (одна буква, которой предшествует тире, чаще всего имеется

ввиду начальная буква из одного слова из длинной версии). Это условное обозначение настолько распространено, что оно является стандартом POSIX.

Сначала, давайте рассмотрим пакет -listfiles (или -L), в котором перечислены файлы, которые были установлены указанным пакетом:

```
$ dpkg -L base-passwd
/.
/usr
/usr/sbin
/usr/sbin/update-passwd
/usr/share
/usr/share/lintian
/usr/share/lintian/overrides
/usr/share/lintian/overrides/base-passwd
/usr/share/doc-base
/usr/share/doc-base/users-and-groups
/usr/share/base-passwd
/usr/share/base-passwd/group.master
/usr/share/base-passwd/passwd.master
/usr/share/man
/usr/share/man/pl
/usr/share/man/pl/man8
/usr/share/man/pl/man8/update-passwd.8.gz
[...]
/usr/share/doc
/usr/share/doc/base-passwd
/usr/share/doc/base-passwd/users-and-groups.txt.gz
/usr/share/doc/base-passwd/changelog.gz
/usr/share/doc/base-passwd/copyright
/usr/share/doc/base-passwd/README
/usr/share/doc/base-passwd/users-and-groups.html
```

Далее, dpkg --search file (or -S), находит любые пакеты, которые содержат файл или путь, передаваемый в аргументе. Например, для поиска пакета, который содержит /bin/date:

```
$ dpkg -S /bin/date
coreutils: /bin/date
```

Команда dpkg --status package (or -s) отображает заголовок установленного пакета. Например, для поиска заголовков пакета coreutils:

```
$ dpkg -s coreutils
Package: coreutils
Essential: yes
Status: install ok installed
Priority: required
Section: utils
Installed-Size: 13855
Maintainer: Michael Stone <mstone@debian.org>
Architecture: amd64
Multi-Arch: foreign
Version: 8.23-3
Replaces: mktemp, realpath, timeout
Pre-Depends: libacl1 (>= 2.2.51-8), libattr1 (>= 1:2.4.46-8), libc6 (>= 2.17),
    ➥ libselinux1 (>= 2.1.13)
Conflicts: timeout
Description: GNU core utilities
This package contains the basic file, shell and text manipulation
utilities which are expected to exist on every operating system.
.
Specifically, this package includes:
arch base64 basename cat chcon chgrp chmod chown chroot cksum comm cp
csplit cut date dd df dir dircolors dirname du echo env expand expr
factor false flock fmt fold groups head hostid id install join link ln
logname ls md5sum mkdir mkfifo mknod mktemp mv nice nl nohup nproc numfmt
od paste pathchk pinky pr printenv printf ptx pwd readlink realpath rm
rmdir runcon sha*sum seq shred sleep sort split stat stty sum sync tac
tail tee test timeout touch tr true truncate tsort tty uname unexpand
uniq unlink users vdir wc who whoami yes
Homepage: http://gnu.org/software/coreutils
```

Команда dpkg -list (или -I) отображает список известных в системе пакетов и их статус установки. Вы также можете использовать grep на выходе для поиска определенных полей или для создания подстановочных символов (например, b *) для поиска пакетов, которые соответствуют определенной части строки поиска. Это покажет сводку пакетов. Например, чтобы показать сводный список всех пакетов, начинающихся с 'b':

```
$ dpkg -l 'b*'
Desired=Unknown/Install/Remove/Purge/Hold
| Status=Not/Inst/Conf-files/Unpacked/halF-conf/Half-inst/trig-aWait/Trig-pend
|/ Err?=(none)/Reinst-required (Status,Err: uppercase=bad)
||/ Name           Version      Architecture Description
+++-==============-============-============-=================================
ii  b43-fwcutter   1:019-3      amd64        utility for extracting Broadcom 4
ii  backdoor-facto 3.4.2-0kali1 all          Patch win32/64 binaries with shel
un  backupninja    <none>       <none>       (no description available)
un  backuppc       <none>       <none>       (no description available)
ii  baobab         3.22.1-1     amd64        GNOME disk usage analyzer
[...]
```

Команда dpkg --contents *file.deb* (или -c) перечисляет все файлы в конкретном .deb файле:

```
$ dpkg -c /var/cache/apt/archives/gnupg_1.4.18-6_amd64.deb
drwxr-xr-x root/root         0 2014-12-04 23:03 ./
drwxr-xr-x root/root         0 2014-12-04 23:03 ./lib/
drwxr-xr-x root/root         0 2014-12-04 23:03 ./lib/udev/
drwxr-xr-x root/root         0 2014-12-04 23:03 ./lib/udev/rules.d/
-rw-r--r-- root/root      2711 2014-12-04 23:03 ./lib/udev/rules.d/60-gnupg.rules
drwxr-xr-x root/root         0 2014-12-04 23:03 ./usr/
drwxr-xr-x root/root         0 2014-12-04 23:03 ./usr/lib/
drwxr-xr-x root/root         0 2014-12-04 23:03 ./usr/lib/gnupg/
-rwxr-xr-x root/root     39328 2014-12-04 23:03 ./usr/lib/gnupg/gpgkeys_ldap
-rwxr-xr-x root/root     92872 2014-12-04 23:03 ./usr/lib/gnupg/gpgkeys_hkp
-rwxr-xr-x root/root     47576 2014-12-04 23:03 ./usr/lib/gnupg/gpgkeys_finger
-rwxr-xr-x root/root     84648 2014-12-04 23:03 ./usr/lib/gnupg/gpgkeys_curl
-rwxr-xr-x root/root      3499 2014-12-04 23:03 ./usr/lib/gnupg/gpgkeys_mailto
drwxr-xr-x root/root         0 2014-12-04 23:03 ./usr/bin/
-rwxr-xr-x root/root     60128 2014-12-04 23:03 ./usr/bin/gpgsplit
-rwxr-xr-x root/root   1012688 2014-12-04 23:03 ./usr/bin/gpg
[...]
```

Команда dpkg --info *file.deb* (или-I) выведет на экран заголовки указанного .deb файла:

```
$ dpkg -I /var/cache/apt/archives/gnupg_1.4.18-6_amd64.deb
 new debian package, version 2.0.
 size 1148362 bytes: control archive=3422 bytes.
    1264 bytes,    26 lines      control
    4521 bytes,    65 lines      md5sums
     479 bytes,    13 lines   *  postinst             #!/bin/sh
     473 bytes,    13 lines   *  preinst              #!/bin/sh
 Package: gnupg
 Version: 1.4.18-6
 Architecture: amd64
 Maintainer: Debian GnuPG-Maintainers <pkg-gnupg-maint@lists.alioth.debian.org>
 Installed-Size: 4888
 Depends: gpgv, libbz2-1.0, libc6 (>= 2.15), libreadline6 (>= 6.0), libusb-0.1-4 (>=
     ➥ 2:0.1.12), zlib1g (>= 1:1.1.4)
 Recommends: gnupg-curl, libldap-2.4-2 (>= 2.4.7)
 Suggests: gnupg-doc, libpcsclite1, parcimonie, xloadimage | imagemagick | eog
 Section: utils
 Priority: important
 Multi-Arch: foreign
 Homepage: http://www.gnupg.org
 Description: GNU privacy guard - a free PGP replacement
  GnuPG is GNU's tool for secure communication and data storage.
  It can be used to encrypt data and to create digital signatures.
  It includes an advanced key management facility and is compliant
   with the proposed OpenPGP Internet standard as described in RFC 4880.
[...]
```

Вы также можете использовать dpkg для сравнения версий пакетов с помощью параметра compare-versions, который часто вызывается внешними программами, включая скрипты конфигураций, которые выполняются самим dpkg. Эта опция требует три параметра: номер версии, оператора сравнения и второго номера версии. Существуют различные возможные операторы: It (строго меньше), le (меньше или равно), eq (равно), ne (не равно), ge (больше или равно) и gt (строго больше). Если сравнение правильное, dpkg возвращает 0 (успех); если нет, он дает ненулевое возвращаемое значение (обозначающее отказ). Рассмотрим эти сравнения:

```
$ dpkg --compare-versions 1.2-3 gt 1.1-4
$ echo $?
0
$ dpkg --compare-versions 1.2-3 lt 1.1-4
$ echo $?
1
$ dpkg --compare-versions 2.6.0pre3-1 lt 2.6.0-1
$ echo $?
1
```

Обратите внимание на неожиданный сбой последнего сравнения: для dpkg строка «pre» (обычно обозначающая предварительную версию) не имеет конкретного значения, а dpkg просто интерпретирует ее как строку, и в этом случае «2. 6.0pre3-1 "в алфавитном порядке больше, чем" 2.6.0-1 ". Когда мы хотим, чтобы номер версии пакета указывал, что это предварительный релиз, мы используем символ тильды, «~»:

```
$ dpkg --compare-versions 2.6.0~pre3-1 lt 2.6.0-1
$ echo $?
0
```

Запрос к базе данных на наличие доступных пакетов с помощью **apt-cache** и **apt**

Команда apt-cache может отображать большую часть информации, хранящейся во внутренней базе данных APT. Эта информация является своего рода кешем, поскольку она собирается из разных источников, перечисленных в файле sources.list. Это происходит во время операции обновления apt.

ТЕРИМНОЛОГИЯ кэша

Кэш является временной системой хранения, используемой для ускорения частого доступа к данным, когда обычный метод доступа является дорогостоящим (по отношению к производительности). Эта концепция может применяться во многих ситуациях и в разных масштабах - от ядра микропроцессоров до высокопроизводительных систем хранения.

В случае с APT, ссылочными файлами пакетов являются те, которые расположены на зеркалах Debian. Тем не менее, было бы очень неэффективно проводить каждый поиск через базы данных онлайн-пакетов. Вот почему APT хранит копию этих файлов (в /var/lib/apt/lists/), и поиск выполняется в этих локальных файлах. Аналогично, /var/cache/apt/archives/ содержит кэшированную копию уже загруженных пакетов, чтобы избежать их повторной загрузки, если вам нужно переустановить их.

Чтобы избежать чрезмерного использования диска при частом обновлении, вы должны регулярно сортировать каталог /var/cache/apt/archives/. Для этого можно использовать две команды: apt clean (или apt-get clean), которая полностью опустошает каталог; apt autoclean (apt-get autoclean), которая удаляет только те пакеты, которые больше не могут быть загружены, поскольку они исчезли из зеркала и поэтому являются бесполезными. Обратите внимание, что параметр конфигурации APT :: Clean-Installed может использоваться для предотвращения удаления файлов .deb, которые в настоящее время установлены. Также обратите внимание, что apt удаляет загруженные файлы после их установки, поэтому это имеет значение, главным образом, во время использования других инструментов.

Команда apt-cache может выполнять поиск пакетов по ключевым словам с помощью apt-cache search key word. Она также может отображать заголовки доступных версий пакета с помощью apt-cache show package. Эта команда предоставляет описание пакета, его зависимости и название его эксплуатационника. Эта особенность довольно полезна при определении пакетов, которые устанавливаются через метапакеты, такие как kali-linux-wireless, kali-linux-web и kali-linux-gpu. Обратите внимание, что поиск apt, apt show, aptitude search и aptitude показывают то же самое.

Альтернативный axi-cache:

Поиск apt-cache является очень элементарнымм инструментом, в основном реализующим grep в описаниях пакетов. Он часто возвращает слишком много результатов или вообще ничего, когда включено слишком много ключевых слов.

С другой стороны axi-cache search term, показывает лучшие результаты, отсортированные по степени важности. Он использует поисковую систему Xapian и является частью пакета apt-xapian-index, который индексирует всю информацию о пакете (и многое другое, например файлы .desktop из всех пакетов Debian). Он знает о тегах и предоставляет результаты через несколько миллисекунд.

```
$ axi-cache search forensics graphical
5 results found.
Results 1-5:
100% autopsy - graphical interface to SleuthKit
82% forensics-colorize - show differences between files using
    ➥  color graphics
73% dff - Powerful, efficient and modular digital forensic
    ➥ framework
53% gpart - Guess PC disk partition table, find lost
    ➥ partitions
46% testdisk - Partition scanner and disk recovery tool, and
    ➥ PhotoRec file recovery tool
More terms: colorize partitions file disklabel autopsy
    ➥ digital differences
More tags: admin::forensics security::forensics role::program
    ➥  admin::recovery interface::commandline admin::boot
    ➥ scope::utility
```

Некоторые функции используются более редко. Например, apt-cache policy отображает приоритеты как источников пакетов, так и отдельных пакетов. Другим примером является apt-cache dumpa vail, который отображает заголовки всех доступных версий всех пакетов. apt-cache pkgnames отображает список всех пакетов, которые появляются хотя бы один раз в кеше.

8.2.6 Устранение проблем

Рано или поздно у вас могут возникнуть определенные проблемы во время взаимодействия с пакетами. В данном разделе мы постараемся очертить основные шаги устранения проблем, которые вам необходимо будет предпринять, а также расскажем вам о некоторых инструментах, которые приблизят вас кразрешению той или иной проблемы.

Проблемы с обработкой после обновления

Несмотря на то, что Kali/Debian приложил все усилия для того, чтобы обновление системы проходило без проблем, оно выполняется не всегда так гладко, как мы надеемся. Новые версии программного обеспечения могут быть несовместимы с предыдущими (например, их поведение по умолчанию или их формат данных могут быть изменены), или же некоторые ошибки могут сохраняться в новой версии, несмотря на тестирование, выполняемое эксплуатационниками пакетов и пользователями Debian Unstable.

Использование отчетов об ошибке Иногда вы можете обнаружить, что новая версия программного обеспечения вообще не работает. Это обычно происходит, если приложение не особенно популярно и недостаточно проверено. Первое, что нужно сделать, - взглянуть на Kali баг трекер[28] Kali и Debian баг трекер [29] на https://bugs.debian.org/*package*, и проверить, сообщил ли кто-то уже об этой проблеме. Если по поводу данной ошибки не поступало никаких отчетов, вам следует составить его самостоятельно (смотри раздел 6.3,"Подача грамотно составленного отчета об ошибке" [стр. 129] для получения более детальной информации). Если отчет уже был подан до вас, то важно понимать, что он и все связанные с ним сообщения являются отличным источником информации относительно самой ошибки. В некоторых случаях патч, исправляющий ошибку уже существует и является доступным в самом отчете об ошибке; вы можете перекомпилировать исправленную версию неисправного пакета локально (смотри раздел 9.1, «Изменение пакетов Kali» [стр. 222]). В других случаях, пользователи могли найти некий искусный метод или обходной путь для работы с этой проблемой и поделились им в своих ответах к отчету; подобного рода инструкции помогут вам разобраться с возникшей проблемой, пока не выйдет соответствующий патч. В идеале ошибка пакета может быть уже исправлена, и вы можете найти информацию об этом в отчете об ошибке.

[28]http://bugs.kali.org
[29]https://bugs.debian.org

Переход на рабочую версию Когда проблема представляет собой четкую регрессию (где работала прежняя версия), вы можете попытаться использовать предыдущую версию пакета. В этом случае вам понадобится копия старой версии. Если у вас есть доступ к старой версии в одном из репозиториев, настроенных в APT, вы можете использовать простую однострочную команду для понижения версии (см. Раздел 8.2.2.2, «Установка пакетов с помощью APT» [стр. 177]). Но пользуясь Kali rolling вы обычно найдете только одну версию для каждого пакета.

Вы все равно можете попытаться найти старый файл .deb и установить его вручную с помощью dpkg. Старые файлы .deb можно найти в нескольких местах:

- В кэше APT в /var/cache/apt/archives/
- В директории pool на нашем обычном зеркале Kali (удаленные и устаревшие пакеты хранятся в течение трех-четырех дней, чтобы избежать проблем с пользователями, не имеющими последних индексов пакета)
- в http://snapshot.debian.org если поврежденный пакет был предоставлен Debian, а не Kali; эта служба хранит абсолютно все версии пакетов Debian

Работа с поврежденными сценариями поддержки Иногда обновление прерывается, потому что один из сценариев поддержки пакета не работает (обычно это postinst). В этих случаях вы можете попытаться диагностировать проблему и, возможно, обойти ее, отредактировав проблемный скрипт.

Здесь мы берем во внимание факт, что сценарий поддержки хранится в /var/lib/dpkg/info/, и что мы можем просмотреть и изменить его.

Ввиду того, что сценарий поддержки является обычно простым сценарием оболочки, мы можем выставить строчку –x сразу после строчки shebang и сделать так, чтобы они запустились повторно (с помощью dpkg --configure -a для postinst) для того, чтобы четко увидеть, что происходит и в чем заключается ошибка. Этот вывод

также может прекрасно дополнять любой отчет об ошибке, который вы можете подавать.

С этими вновь полученными знаниями, вы можете как исправлять основную проблему, так и трансформировать неисправную команду в работающую (например, путем добавления || true в конец строчки).

Обратите внимание, что данная методика не работает в случае сбоя preinst, поскольку этот скрипт выполняется еще до того, как пакет будет установлен, поэтому он еще не находится в конечном месте своего назначения. Он работает для postrm и prerm, хотя вам нужно будет выполнить удаление пакета (и, соответственно, обновление), чтобы запустить их.

Файл журнала dpkg

Инструмент dpkg хранит журнал всех своих действий в /var/log/dpkg.log. Этот журнал является чрезвычайно подробным, поскольку он описывает все этапы каждого пакета. В дополнение к тому, что он позволяет отслеживать поведение dpkg, он помогает сохранить историю развития системы: вы можете найти точный момент, когда каждый пакет был установлен или обновлен, и эта информация может быть чрезвычайно полезна для понимания недавнего изменение поведения. Кроме того, при записи всех версий легко перекрестно проверить информацию с помощью changelog.Debian.gz для пакетов, которые вызывают вопросы или даже с помощью онлайн-отчетами об ошибках.

```
# tail /var/log/dpkg.log
2016-12-22 09:04:05 status installed kali-linux-gpu:amd64 2016.3.2
2016-12-22 09:20:07 startup packages remove
2016-12-22 09:20:07 status installed kali-linux-gpu:amd64 2016.3.2
2016-12-22 09:20:07 remove kali-linux-gpu:amd64 2016.3.2 <none>
2016-12-22 09:20:07 status half-configured kali-linux-gpu:amd64 2016.3.2
2016-12-22 09:20:07 status half-installed kali-linux-gpu:amd64 2016.3.2
2016-12-22 09:20:07 status config-files kali-linux-gpu:amd64 2016.3.2
2016-12-22 09:20:07 status config-files kali-linux-gpu:amd64 2016.3.2
2016-12-22 09:20:07 status config-files kali-linux-gpu:amd64 2016.3.2
2016-12-22 09:20:07 status not-installed kali-linux-gpu:amd64 <none>
```

Переустановка пакетов с помощью apt —reinstall и aptitude reinstall

Когда вы ошибочно наносите ущерб своей системе, удаляя или изменяя определенные файлы, самым простым способом их восстановления является переустановка поврежденного пакета. К сожалению, система пакетирования обнаруживает, что пакет уже установлен и вежливо отказывается переустанавливать его. Чтобы этого избежать, используйте параметр -reinstall команд apt и apt-get. Следующая команда переустанавливает ***postfix***, даже если он уже присутствует:

```
# apt --reinstall install postfix
```

Команда aptitude немного отличается, но, тем не менее, она приводит к тому же самому результату, если использовать aptitude reinstall postfix Команда dpkg не предотвращает переустановку, но ее редко вызывают напрямую.

Не используйте apt --reinstall для того, чтобы восстановиться после атаки

Использование apt --reinstall для восстановления пакетов, измененных во время атаки, не приведет к восстановлению системы, в том виде, в котором она была до.

После атаки вы не можете положиться абсолютно ни на что: dpkg и apt, возможно, были заменены вредоносными программами, и они не будут заниматься переустановкой файлов, как вы, скорее всего, ожидаете. Злоумышленник также мог изменять или создавать файлы вне контроля dpkg.

Также следует помнить, что вы можете указать конкретный дистрибутив с помощью apt, что означает, что вы можете откатиться к более старой версии пакета (если, к примеру, вы уверены, что она будет работать должным образом), при условии, что она по-прежнему доступна в одном из источников, на которые ссылается файл sources.list:

```
# apt install w3af/kali-rolling
```

Использование -force- * для восстановления поврежденных зависимостей

Если вы были не достаточно осторожным, использование опции -force-* или каких-либо других неисправностей может привести к тому, что APT команды будут отказываться выполнять свою привычную функцию в системе. По сути, некоторые из этих параметров позволяют устанавливать пакет, когда зависимость не выполняется, или когда присутствует конфликт. Результатом является непоследовательная система с точки зрения зависимостей, и команды APT откажутся выполнять любое действие, кроме тех, которые вернут систему в согласованное состояние (достижение этой цели часто состоит в установке отсутствующей зависимости или удалении проблемного пакета). Обычно это приводит к сообщению, подобному этому, полученному после установки новой версии *rdesktop*, игнорируя при этом ее зависимость от более новой версии *libc6:*

```
# apt full-upgrade
[...]
You might want to run 'apt-get -f install' to correct these.
The following packages have unmet dependencies:
rdesktop: Depends: libc6 (>= 2.5) but 2.3.6.ds1-13etch7 is installed
E: Unmet dependencies. Try using -f.
```

Если же вы являетесь отважным администратором, который уверен в правильности своего анализа, то вы можете игнорировать зависимость или конфликт, и использовать параметр -force- *. В этом случае, если вы хотите продолжать использовать apt или aptitude, вы должны отредактировать /var/lib/dpkg/status, чтобы удалить или изменить зависимость или конфликт, которые вы выбрали для переопределения.

Эта манипуляция является очень дурным тоном и никогда не должна использоваться, кроме как в самом крайнем случае.

Довольно часто более подходящим решением является перекомпиляция пакета, который вызывает проблему, или использование новой версии (потенциально исправленной) из репозитория, предоставляющего backports (backports - это более новые версии, специально перекомпилированные для работы в более старой среде).

8.2.7 Внешние интерфейсы: aptitude и synaptic

APT является программой, работающей на C ++, код которой в основном находится в общей библиотеке libapt-pkg. Как раз именно эта общая библиотека открыла дверь для создания пользовательских интерфейсов (front-ends), поскольку код разделяемой библиотеки можно легко использовать повторно. Исторически, apt-get был разработан только как тестовый интерфейс для libapt-pkg, но ввиду его ошеломляющего успеха, этот факт тщательно умалчивается.
Со временем, несмотря на популярность интерфейсов командной строки, таких как apt и apt-get, были разработаны различные графические интерфейсы. Мы рассмотрим два из этих интерфейсов в этом разделе: aptitude и synaptic.

Aptitude

Aptitude, показанный на рис. 8.1, «Менеджер пакетов aptitude» [стр. 191], представляет собой интерактивную программу, которая может использоваться в полуграфическом режиме на консоли. Вы можете просмотреть список установленных и доступных пакетов, просмотреть всю информацию и выбрать пакеты для установки или удаления. Программа разработана специально для использования администраторами, поэтому ее поведение по умолчанию намного более интеллектуально, чем APT, и его интерфейс намного легче понять.

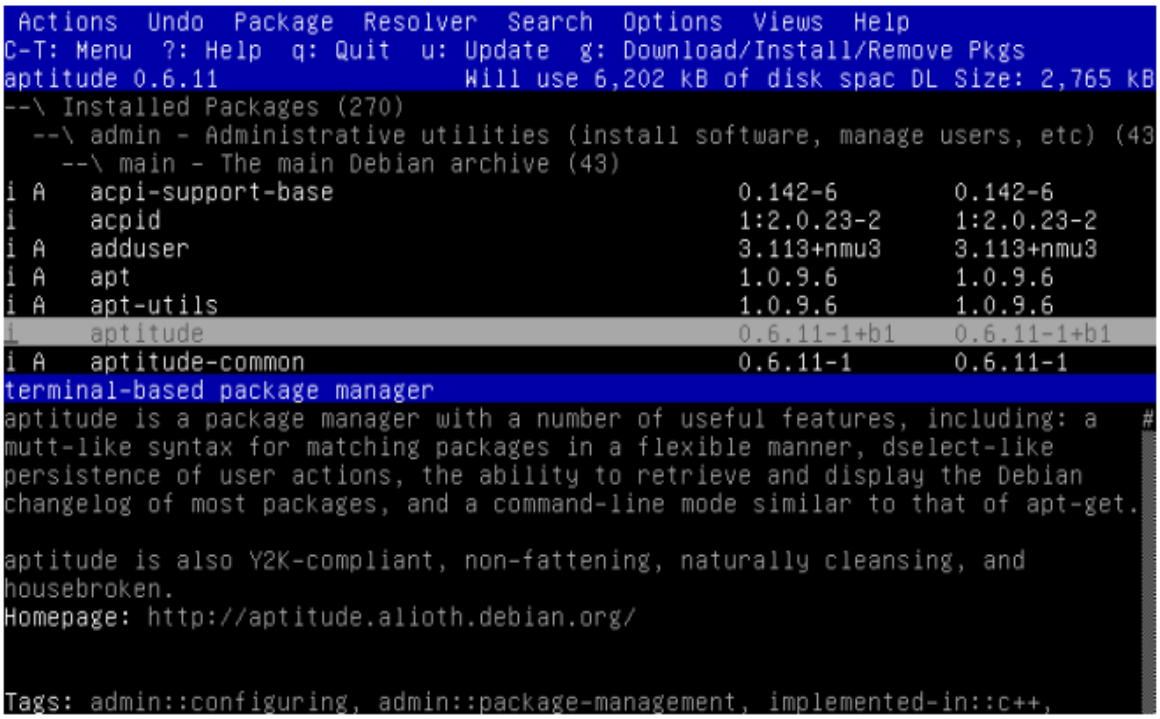

Рисунок 8.1 *Менеджер пакетов aptitude*

Когда вы запустите aptitude, вам будет показан список пакетов, отсортированных по состоянию (установленный, не установленный или установленный, но недоступный на зеркалах), тогда как в других разделах отображаются задачи, виртуальные пакеты и новые пакеты, появившиеся недавно на зеркалах. Для облегчения тематического просмотра доступны разнообразные режимы.

Во всех случаях aptitude отображает список, объединяющий категории и пакеты на экране. Категории организованы через древовидную структуру, ветви которой можно развернуть или свернуть с помощью клавиши Enter. Клавишу «+» следует использовать для того, чтобы маркировать пакет, который вы хотите установить, «-» следует использовать для маркирования пакетов, которые вы хотите удалить, а «_» для их очистки. Обратите внимание, что эти ключи могут также быть использованы для категорий, и в этом случае соответствующие действия будут применяться ко всем пакетам категории. Клавиша u обновляет списки доступных пакетов, в то время как Shift+u подготавливает полное системное обновление. Ключ g переключается на сводный режим просмотра запрошенных изменений (повторный ввод g применит изменения), а ключ q

завершает текущее представление. Если вы находитесь в исходном режиме просмотра, то это закроет aptitude.

Документация aptitude

В этом разделе не рассматриваются более тонкие нюансы использования aptitude, он скорее фокусируется на предоставление аварийного пакета пользователя. Aptitude достаточно хорошо документирована и мы рекомендуем использовать полное руководство, которое доступно в пакете *aptitude-doc-en.*

file:///usr/share/doc/aptitude/html/en/index.html

Для поиска пакета вы можете ввести шаблон поиска или следовать ему. Этот шаблон соответствует имени пакета, но может также применяться к описанию (если ему предшествует ~ d), к разделу (с ~ s) или другим характеристикам, указанным в документации. Те же шаблоны могут фильтровать список отображаемых пакетов: введите ключ l (как в *limit*) и введите шаблон.

Управление *автоматической пометкой* пакетов Debian (смотри раздел 8.3.4 “Автоматическое отслеживание установленных пакетов” [стр. 199]) является безумно простым с aptitude. Вы можете просматривать список установленных пакетов и отмечать пакеты как автоматические с помощью Shift + m или вы можете удалить отметку с помощью клавиши m. Автоматические пакеты отображаются с «A» в списке пакетов. Эта функция также предлагает простой способ визуализации пакетов, используемых на машине, без всех библиотек и зависимостей, которые вам абсолютно не нужны. Связанный шаблон, который может быть использован с l (limit) (для активации режима фильтра) является ~i!~M. Он обозначает, что вы хотите увидеть только установленные пакеты (~i), не отмеченные как автоматические (!~M).

Использование aptitude в Интерфейсе командной строки

Большинство особенностей Aptitude доступно через интерактивный интерфейс также как и через командную строку.

Эти командные строки покажутся очень знакомыми постоянным пользователям apt-get и apt-cache.

Расширенные функции aptitude также доступны в командной строке. Вы можете использовать те же шаблоны поиска пакетов, что и в интерактивной версии. Например, если вы хотите очистить список установленных вручную пакетов, и если вы знаете, что ни одна из локально установленных программ не требует каких-либо конкретных библиотек или модулей Perl, вы можете пометить соответствующие пакеты как автоматические с помощью одной команды:

```
# aptitude markauto '~slibs|~sperl'
```

Здесь вы можете четко увидеть всю силу системы шаблонов поиска aptitude, которая позволяет мгновенно выбирать все пакеты в Tibs и perl разделах.

Остерегайтесь случаев, когда некоторые пакеты отмечены как автоматические, но никакие другие пакеты не зависят от них. Такие пакеты будут немедленно удалены (после запроса на подтверждение).

Управление рекомендациями, предложениями и задачами
Еще одна интересная особенность aptitude заключается в том, что она учитывает рекомендации между пакетами, предоставляя пользователям возможность не устанавливать их в каждом конкретном случае. Например, пакет *gnome* рекомендует *gdebi* (среди других). Когда вы выбираете первый для установки, последний также будет выбран (и будет отмечен как автоматический, если он еще не установлен в системе). Ввод g сделает это очевидным: *gdebi* появляется в окне сводке ожидающих действий в списке пакетов, установленных автоматически для удовлетворения зависимостей. Однако, вы можете решить не устанавливать его, отменив его выбор, прежде чем подтверждать операции.

Обратите внимание, что эта функция отслеживания рекомендаций не применяется к обновлениям. Например, если новая версия

gnome рекомендует пакет, который ранее не рекомендовался, пакет не будет помечен для установки. Однако он будет указан на экране обновления, чтобы администратор в случае необходимости мог выбрать его для установки.

Также учитываются предложения между пакетами, но в соответствии с их конкретным статусом. Например, поскольку *gnome* предлагает *dia-gnome*, последний будет отображаться в окне сводки ожидающих действий (в разделе пакетов, предложенных другими пакетами). Таким образом, администратор сможет решить, принимать во внимание это предложение или нет. Поскольку это всего лишь предложение, а не зависимость или рекомендация, пакет не будет выбран автоматически - его выбор требует ручного вмешательства (таким образом, пакет не будет отмечен как автоматический).

В этом же духе, помните что aptitude разумно использует концепцию задач. Поскольку задачи отображаются в виде категорий на экранах списков пакетов, вы можете выбрать как полную задачу для установки или удаления, так и просто просмотреть список пакетов, включенных в задачу, чтобы выбрать меньшую подгруппу.

Более эффективные алгоритмы В заключение этого раздела отметим, что aptitude имеет более сложные алгоритмы по сравнению с apt, когда дело доходит до решения сложных ситуаций. Когда запрашивается набор действий и когда эти совместные действия приводят к некогерентной системе, aptitude оценивает несколько возможных сценариев и представляет их в порядке уменьшения важности. Однако, эти алгоритмы не являются надежными. К счастью, всегда есть возможность вручную выбрать действия для выполнения. Когда текущие выбранные действия приводят к противоречиям, в верхней части экрана указывается количество неисправных пакетов (вы можете напрямую перейти к этим пакетам, нажав b). Затем вы можете вручную создать решение. В частности, вы можете получить доступ к различным доступным версиям, выбрав пакет с помощью Enter. Если выбор одной из этих версий решает проблему, вы не должны сомневаться в верности решения и использовать эту функцию. Когда количество сломанных снижается до нуля, вы

можете безопасно перейти к окну сводки ожидающих действий для последней проверки перед их применением.

Журнал Aptitude

Как и dpkg, aptitude хранит след выполненных действий в своем файле журнала (/var/log/aptitude). Однако, поскольку обе команды работают на совершенно разном уровне, вы не можете найти ту же информацию в своих соответствующих файлах журналов. Хотя dpkg регистрирует все операции, выполняемые на отдельных пакетах шаг за шагом, aptitude дает более широкий обзор операций высокого уровня, таких как общесистемное обновление.

Но будьте очень внимательны, т.к. этот файл журнала содержит только сводку операций, выполненных aptitude. Если используются другие интерфейсы (или даже сам dpkg), то журнал aptitude будет содержать только частичный вид операций, поэтому вы не можете полагаться на него, чтобы выстроить достоверную историю системы.

Synaptic

Synaptic является графическим менеджером пакетов, который имеет чистый и эффективный графический интерфейс (см. Рис. 8.2, «Менеджер пакетов Synaptic» [стр. 194]) на основе GTK + и GNOME. В нем существует множество готовых к использованию фильтров, которые обеспечивают быстрый доступ к новым пакетам, установленным пакетам, обновляемым пакетам, устаревшим пакетам и т. д. Если вы просматриваете эти списки, вы можете выбрать операции, которые должны выполняться в пакетах (установка, обновление, удаление, очищение); эти операции выполняются не сразу, но они заносятся в список задач. Один щелчок на кнопке запускает проверку операции, и они выполняются сразу же.

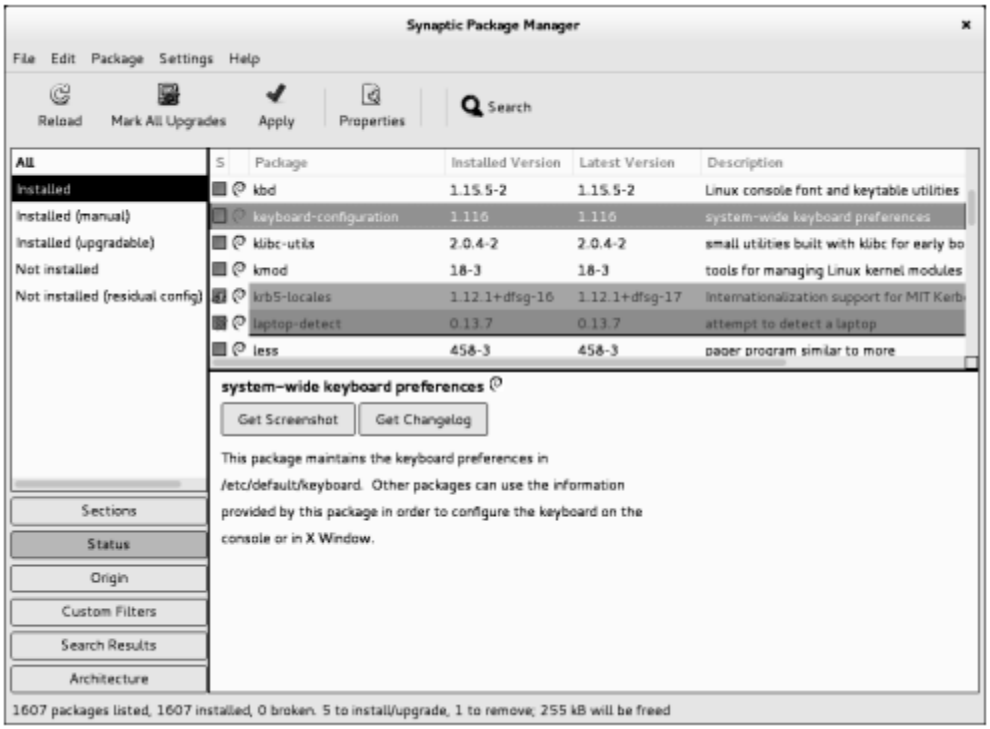

Рисунок 8.2 *Менеджер пакетов synaptic*

8.3 Дополнительная настройка и использование APT

Теперь пришло время погрузиться в более сложные темы. Во-первых, мы рассмотрим расширенную конфигурацию APT, которая позволит вам установить более постоянные параметры, которые будут применяться к инструментам APT. Затем мы покажем вам, как управлять приоритетами пакетов, что откроет нам дверь для дополнительной настройки, пользовательских обновлений и модернизаций. Мы также покажем, как обращаться с несколькими дистрибутивами, чтобы вы могли начать экспериментировать с пакетами, поступающими из других дистрибутивов. Затем мы рассмотрим, как отслеживать автоматически установленные пакеты, что позволит вам управлять пакетами, которые устанавливаются через зависимости. Мы также объясним, каким образом поддержка multiarch дает нам возможность запуска пакетов, созданных для различных аппаратных архитектур. И последнее, но не менее важное: мы обсудим криптографические протоколы и утилиты, которые позволят вам проверить подлинность каждого пакета.

8.3.1 Настройка APT

Перед тем как мы углубимся в изучение того, как настроить APT, давайте потратим немного времени и обсудим механизм конфигурации системы Debian. В течение длительного времени настройка проводилась специально заточенными под это файлами. Однако в современных Linux системах, вроде Debian и Kali, директории конфигурации с суффиксом .d используются все чаще и чаще. Каждая директория представляет собой файл конфигурации, который в свою очередь разбит на множество файлов. В этом смысле все файлы в /etc/apt/apt.conf.d/ используются для настройки APT. APT обрабатывает файлы в алфавитном порядке, таким образом, что более поздние файлы могут изменять элементы конфигурации, определенные в более ранних файлах.

Эта структура предоставляет определенную гибкость администратору и тем, кто занимается поддержкой пакетов, позволяя им производить изменения в настройках программного обеспечения с помощью добавления файлов без необходимости изменения уже существующего файла. Это особенно полезно для тех, кто занимается поддержкой пакетов, потому что они могут использовать этот подход для адаптации конфигурации другого программного обеспечения, чтобы гарантировать, что оно без проблем сможет сосуществовать с другим программным обеспечением, не нарушая политики Debian, которая в свою очередь строго запрещает изменение конфигурации файлов других пакетов. Благодаря механизму .d конфигурации вам не нужно вручную следовать множеству инструкций по настройке пакетов, которые обычно находятся в файле пакета /usr/share/doc/package /README.Debian, т.к. установщик может обращаться к файлу конфигурации.

Будьте осторожны с файлами конфигурации, которые были созданы в .d директории

Хотя APT имеет встроенную поддержку своего каталога /etc/apt/apt.conf.d, это не всегда так. Для некоторых приложений (например, для exim), директория .d является особым дополнением Debian, которое используется в качестве входа для

динамического создания канонического файла конфигурации, используемого приложением. В таких случаях пакеты предоставляют команду “update-*” (например: update-exim4.conf), которая последовательно соединит файлы из .d директории и перезапишет основной файл конфигурации.
В подобных ситуациях вы ни в коем случае не должны вручную редактировать основной файл конфигурации, т.к. ваши изменения будут потеряны после следующего запуска команды update-*, а также вы никогда не должны забывать запускать предыдущую команду после редактирования файла из .d директории (или ваши изменения не будут использоваться).

Теперь вооружившись пониманием механизма .d конфигурации, давайте поговорим о том, как вы можете использовать это для настройки APT. Как мы уже обсудили, вы можете изменять поведение APT с помощью dpkg аргументов командной строки, как в этом примере, который выполняет принудительную перезапись установки zsh:

```
# apt -o Dpkg::Options::="--force-overwrite" install zsh
```

Очевидно, что это довольно громоздко, особенно если вы используете опции часто, но вы также можете использовать структуру конфигурации .d директории для настройки определенных аспектов APT, путем добавления директив в файлы в директории /etc/apt/apt.conf.d/. Например, эта (и многие другие) директивы могут с легкостью быть добавлены к файлу в /etc/apt/apt.conf.d/. Имя этого файла несколько произвольно, но общим условным обозначением является использование либо local, либо 991ocal:

```
$ cat /etc/apt/apt.conf.d/99local
Dpkg::Options {
   "--force-overwrite";
}
```

Существует довольно много других полезных опций конфигурации и мы, к сожалению, не сможем затронуть их все. Тем не менее, одну из них мы обсудим. Ту, которая относится к связанности узлов в сети. Например, если вы имеете доступ к интернету только через прокси, добавьте строку вроде **Acquire::http::proxy ”http-**

Hyourproxy:3128". Для прокси FTP используйте **Acquire::ftp::proxy "ftp■***.Hyourproxy".*

Чтобы узнать больше об опциях настройки, мы рекомендуем вам прочитать страницу руководства к apt.conf(5) с помощью команды man apt. Conf.

8.3.2 Управление приоритетами пакетов

Одним из самых важных аспектов в конфигурации APT является управление приоритетами, связанными с каждым источником пакетов. Например, вы можете захотеть расширить свою Kali Rolling систему на один или два более новых пакета Debian Unstable или Debian Experimental. Также возможно назначить приоритет для каждого доступного пакета (один и тот же пакет может иметь несколько приоритетов в зависимости от его версии или от дистрибутива, предоставившего его). Эти приоритеты будут влиять на поведение APT: для каждого пакета он всегда будет выбирать версию с наивысшим приоритетом (за исключением случаев, когда эта версия старше установленной, а ее приоритет меньше 1000).

APT определяет несколько приоритетов по умолчанию. Каждый установленная версия пакета обладает приоритетом равным 100. Неустановленная версия имеет приоритет 500 по умолчанию, но она может измениться на 990, если является частью целевого выпуска (определенной с помощью опции командной строки **–t** или **APT::Default-Release** директивой конфигурации).

Вы можете изменять приоритеты путем добавления записей в файл /etc/apt/preferences с именами поврежденных пакетов, их версией, их производителем и их новым приоритетом.

APT никогда не установит более старую версию пакета (имеется ввиду пакет, чья версия является более старой по отношению к существующей версии установленного пакета) за исключением тех случаев, когда её приоритет выше 1000. APT всегда будет устанавливать пакет с самым высоким приоритетом, который выше указанной цифры. Если у двух пакетов одинаковый

приоритет, APT установит самый новый (чей номер версии самый высокий). Если два пакета имеют одинаковую версию и приоритет, но отличаются своим содержанием, APT установит ту версию, которая еще не была установлена (это правило было создано для того, чтобы затрагивать случаи, когда пакет обновился, но не обновился номер его версии).

Более конкретно, пакет, приоритет которого меньше 0, никогда не будет установлен. Пакет с приоритетом в диапазоне от 0 до 100 будет установлен только в том случае, если другая версия пакета еще не была установлена. При приоритете от 100 до 500 пакет будет установлен только в том случае, если в другом дистрибутиве нет другой более новой версии, установленной или доступной. Пакет приоритетов между 501 и 990 будет установлен только в том случае, если новая версия не установлена или недоступна в целевом дистрибутиве. Пакет с приоритетом от 990 до 1000 будет установлен в случае, если существующая версия является более старой. Приоритет, превышающий 1000, всегда приведет к установке пакета, даже если он заставит APT перейти на более раннюю версию.

Когда APT проверяет /etc/apt/preferences, он сначала учитывает наиболее конкретные записи (часто указывающие соответствующий пакет), затем более общие (в том числе, например, все пакеты дистрибутива). Если существует несколько общих записей, используется первое соответствие. Доступный критерий выбора включает в себя имя пакета и источник, который его предоставил. Каждый источник пакета идентифицирован информацией содержащейся в файле Release, который APT загружает вместе с файлами Packages. Эти файлы обычно указывают источник, “Kali” для пакетов с официальных зеркал Kali и «Debian »для пакетов с официальных зеркал Debian, но источником также может быть название организации, имя человека или же название каких-либо других репозиториев. Файл Release также предоставляет название дистрибутива вместе с его версией. Теперь, давайте познакомимся с его синтаксисом на примере реального случая, для более полного понимания механизма.

Приоритет Kali-Bleeding-Edge и Debian Experimental

Если вы указали kali-bleeding-edge или Debian experimental в вашем файле sources.list, то соответствующий пакет практически никогда не будет установлен, потому что их APT приоритет по умолчанию 1. Это, безусловно, особенный случай, который был задан специально для того, чтобы предотвратить пользователей от ошибочной установки bleeding edge пакетов. Пакеты могут быть установлены только путем ввода apt install package/kali-bleeding-edge, предполагая, конечно, что вы полностью осознаете риски и потенциальные проблемы «жизни на краю» (life on the edge). По-прежнему возможно (хотя и не рекомендуется) обрабатывать пакеты kali-bleeding-edge/experimental, аналогично пакетам из других дистрибутивов, предоставляя им приоритет 500. Это делается с помощью конкретной записи в /etc/apt/preferences:

```
Package: *
Pin: release a=kali-bleeding-edge
Pin-Priority: 500
```

Давайте предположим, что вы всего лишь хотите использовать пакеты Kali и желаете, чтобы пакеты Debian устанавливались только при явном запросе. Вы можете написать следующую запись в файле /etc/apt/ preferences (или в любом другом файле в /etc/apt/preferences.d/):

```
Package: *
Pin: release o=Kali
Pin-Priority: 900

Package: *
Pin: release o=Debian
Pin-Priority: -10
```

В последних двух примерах вы встречали a=kali-bleeding-edge, что обозначает имя выбранного дистрибутива, а также вы могли видеть o=Kali и o=Debian, что в свою очередь ограничивает сферу деятельности пакетов, чьим источником являются Kali и Debian, соответственно.

Теперь давайте предположим, что у вас есть сервер с несколькими локальными программами, которые находятся в зависимости от версии Perl 5.22 и вы, соответственно, хотите убедиться в том, что

обновление не установит другую версию. Вы можете использовать следующую запись.

```
Package: perl
Pin: version 5.22*
Pin-Priority: 1001
```

Справочная документация для этого файла конфигурации доступна на странице руководства apt_pref erences (5), которую вы можете вывести на экран с помощью команды man apt_preferences.

Добавление комментариев в /etc/apt/preferences
В /etc/apt/preferences не существует официального синтаксиса для комментариев, но некоторые текстовые описания могут быть предоставлены путем добавления одного или нескольких полей Explanation в каждую запись:

```
Explanation: The package xserver-xorg-video-intel provided
Explanation: in experimental can be used safely
Package: xserver-xorg-video-intel
Pin: release a=experimental
Pin-Priority: 500
```

8.3.3 Работа с несколькими дистрибутивами

Учитывая, что apt является таким замечательным инструментом, вы, скорее всего, захотите погрузиться и начать экспериментировать с пакетами, поступающими из других дистрибутивов. Например, после установки системы Kali Rolling, вы можете попробовать пакет программного обеспечения, доступный в Kali Dev, Debian Unstable или Debian Experimental, не слишком сильно отдаляясь от исходного состояния системы.

Даже если вы время от времени сталкиваетесь с проблемами, когда вы смешиваете пакеты из различных дистрибутивов, apt отлично справляется с такого рода сосуществованием и очень эффективно ограничивает риски (при условии, что зависимости пакетов точны). Сначала перечислите все дистрибутивы, используемые в /etc/apt/sources.list, и затем определите

дистрибутив, на который вы ссылаетесь с помощью параметра APT::Default-Release (смотри раздел 8.2.3, «Обновление Kali Linux» [стр. 179])

Давайте предположим, что Kali Rolling является вашим ссылочным дистрибутивом, но Kali Dev и Debian Unstable также указаны в вашем файле sources.list. В этом случае вы можете использовать apt install *package*/unstable для установки пакета из Debian Unstable. Если установка была прервана ввиду некоторых неудовлетворяемых зависимостей, позвольте ей разрешить эти зависимости в Unstable путем добавления параметра -t unstable.

В данной ситуации, обновления (upgrade и full-upgrade) выполнены в пределах Kali Rolling за исключением пакетов, которые уже были обновлены в другом дистрибутиве: они будут продолжать следовать обновлениям, доступным в других дистрибутивах. Мы объясним это поведение с помощью приоритетов по умолчанию выставленных APT ниже. Не стесняйтесь использовать apt-cache policy (см. вставку «Использование apt-cache policy» [стр. 199]) для проверки заданных приоритетов.

Все опирается на то, что рассматривает лишь пакеты, которые выше или равны версии установленного пакета (при условии, что /etc/apt/preferences не используется для форсирования приоритетов выше 1000 для некоторых пакетов).

Использование apt-cache policy

Для получения лучшего понимания механизма приоритетности, не стесняйтесь выполнять apt-cache policy для того, чтобы отобразить приоритетность по умолчанию связанную с каждым источником пакета. Вы также можете использовать apt-cache policy *package* для отображения приоритетов всех доступных версий заданного пакета.

Давайте предположим, что вы установили версию 1 первого пакета из ***Kali Rolling*** и что версия 2 и 3 доступны соответственно в ***Kali Dev*** и ***Debian Unstable.*** Установленная версия обладает приоритетом 100, а версия доступная в ***Kali Rolling*** (абсолютно такая же) обладает приоритетом 990 (ввиду того, что она является

частью целевого выпуска). Пакеты в ***Kali Dev*** и ***Debian Unstable*** обладают приоритетом 500 (приоритет по умолчанию не установленной версии). Таким образом, победителем в этой ситуации выйдет версия 1 с приоритетом в 990. Пакет остается в ***Kali Rolling***.

Давайте приведем другой пример с пакетов версии 2, который был установлен из ***Kali Dev.*** Версия 1 является доступной в ***Kali Rolling*** и версия 3 доступна в ***Debian Unstable.*** Версия 1 (с приоритетом 990 – таким образом ниже чем 1000) отбрасывается, потому что она обладает более низким приоритетом, чем установленная версия. Остаются две версии 2 и 3, обе с приоритетом 500. В таких случаях APT выбирает самую новую версию, ту которая доступна в ***Debian Unstable***. Если вы не хотите чтобы пакет установленный с ***Kali Dev*** переместился в ***Debian Unstable,*** вам нужно назначить приоритет меньше 500 (например, 490) для пакетов, поступающих с ***Debian Unstable.*** Вы можете изменить /etc/apt/preferences для этого эффекта:

```
Package: *
Pin: release a=unstable
Pin-Priority: 490
```

8.3.4 Автоматическое отслеживание установленных пакетов

Одной из существенных функций apt является отслеживание пакетов, установленных только через зависимость. Такие пакеты называются ***автоматическими (automatic)*** и часто включают в себя библиотеки.

С этой информацией, когда пакеты удалены, менеджеры пакетов могут вычислить список автоматических пакетов, которые больше не нужны (потому что не существует вручную установленных пакетов, которые бы полагались на них). Команда apt autoremove избавится от этих пакетов. Aptitude не имеет подобной команды, т.к. она удаляет эти пакеты автоматически, как только они будут идентифицированы. Во всех случаях инструменты отображают точное сообщение с указанием затронутых пакетов.

Очень полезная привычка отмечать как автоматический любой пакет, который вам не нужен напрямую, чтобы они автоматически удалялись, когда они больше не нужны. Вы можете использовать команду apt-mark auto *package* для маркировки данного пакета как автоматического, тогда как команда apt-mark manual *package* делает обратное. aptitude markauto и aptitude unmarkauto работают точно также, хотя они предлагают больше возможностей для маркировки сразу нескольких пакетов (см. раздел 8.2.7.1, «Aptitude» [стр. 190]). Консольный интерактивный интерфейс aptitude также упрощает просмотр отметки «автоматический» на многих пакетах.

Возможно, вам захочется узнать, почему в системе присутствуют автоматически установленные пакеты. Чтобы получить эту информацию из командной строки, вы можете использовать aptitude why *package* (apt и apt-get не имеют подобной функции):

```
$ aptitude why python-debian
i   aptitude         Recommends apt-xapian-index
i A apt-xapian-index Depends    python-debian (>= 0.1.15)
```

8.3.5 Использование поддержки Multi-Arch Support

Все пакеты Debian имеют поле «Архитектура» (Architecture) в их управляющей информации. Это поле может содержать либо «все» (“all” (для тех пакетов, которые не зависят от архитектуры систем)), либо имя архитектуры, на которую он нацелен (например, amd64 или armhf). В последнем случае, по умолчанию, dpkg будет устанавливать пакет только в том случае, если его архитектура соответствует архитектуре хоста, что видно на выводе команды dpkg -print-architecture.

Это ограничение гарантирует, что вы не получите исполняемые файлы, скомпилированные для неправильной архитектуры. Все было бы идеально, за исключением того, что (некоторые) компьютеры могут запускать исполняемые файлы для нескольких архитектур, как стандартным способом (система amd64 может запускать исполняемые файлы i386), так и через эмуляторы.

Включение Multi-Arch

Multi-arch поддержка dpkg позволяет пользователям определять внешние архитектуры, которые могут быть установлены в текущей системе. Это легко сделать с помощью dpkg --add-architecture, как в примере ниже, где архитектура i386 должна быть добавлена в систему amd64 для запуска приложений Windows с использованием Wine[30]. Существует соответствующая команда dpkg --remove-architecture для отказа от поддержки внешней архитектуры, но ее можно использовать только тогда, когда на вашей системе не осталось установленных пакетов, относящихся к этой архитектуры.

```
# dpkg --print-architecture
amd64
```

```
# wine
it looks like wine32 is missing, you should install it.
multiarch needs to be enabled first.  as root, please
execute "dpkg --add-architecture i386 & apt-get update &
apt-get install wine32"
Usage: wine PROGRAM [ARGUMENTS...]   Run the specified program
       wine --help                   Display this help and exit
       wine --version                Output version information and exit
# dpkg --add-architecture i386
# dpkg --print-foreign-architectures
i386
# apt update
[...]
# apt install wine32
[...]
Setting up libwine:i386 (1.8.6-5) ...
Setting up vdpau-driver-all:i386 (1.1.1-6) ...
Setting up wine32:i386 (1.8.6-5) ...
Setting up libasound2-plugins:i386 (1.1.1-1) ...
Processing triggers for libc-bin (2.24-9)
# wine
Usage: wine PROGRAM [ARGUMENTS...]   Run the specified program
     wine --help                   Display this help and exit
     wine --version                Output version information and exit
# dpkg --remove-architecture i386
dpkg: error: cannot remove architecture 'i386' currently in use by the database
# dpkg --print-foreign-architectures
i386
```

[30] https://www.winehq.org/

APT автоматически определит, когда dpkg был настроен для поддержки внешних архитектур и начнет скачивание соответствующих файлов пакетов на протяжении процесса обновления.

Затем могут быть установлены внешние пакеты с помощью команды apt install package-.architecture.

Использование патентованных i386 исполняемых файлов на amd64

Существует множество вариантов использования multi-arch, но наиболее популярным из них является возможность выполнения 32-разрядных бинарных файлов (i386) в 64-битных системах (amd64), в частности, благодаря тому, что несколько популярных фирменных приложений (таких как Skype) предоставляются только в 32 разрядных версиях.

Изменения, связанные с Multi-Arch

Чтобы сделать multi-arch реально полезным и удобным в использовании, библиотеки должны быть перепакетированы и перемещены в специальную директорию для архитектур таким образом, чтобы несколько экземпляров (ориентированных на разные архитектуры) могли успешно сосуществовать друг с другом. Такие обновленные пакеты содержат Multi-Arch: одно и то же поле заголовка для того, чтобы сообщить системе пакетирования, что различные архитектуры пакета могут безопасно сосуществовать (и что эти пакеты могут удовлетворять только зависимости пакетов такой же архитектуры).

```
$ dpkg -s libwine
dpkg-query: error: --status needs a valid package name but 'libwine' is not: ambiguous
    ➥ package name 'libwine' with more than one installed instance

Use --help for help about querying packages.
$ dpkg -s libwine:amd64 libwine:i386 | grep ^Multi
Multi-Arch: same
Multi-Arch: same
$ dpkg -L libgcc1:amd64 |grep .so
[...]
/usr/lib/x86_64-linux-gnu/wine/libwine.so.1
$ dpkg -S /usr/share/doc/libwine/copyright
libwine:amd64, libwine:i386: /usr/share/doc/libwine/copyright
```

Также стоит отметить, что в Multi-Arch одинаковые пакеты должны иметь названия в соответствии с их архитектурой для того, чтобы идентифицироваться без проблем. Эти пакеты также могут совместно использовать файлы с другими экземплярами одного и того же пакета; dpkg гарантирует, что все пакеты обладают одинаковыми поразрядными файлами, когда последние совместно используются. Также, все экземпляры пакета должны иметь одну и ту же версию, поэтому их необходимо обновлять вместе.

Поддержка Multi-Arch приносит некоторые интересные трудности, вызванные способом, которым обрабатываются зависимости. Удовлетворение зависимости требует либо пакета отмеченного как Multi-Arch: foreign или пакета, архитектура которого соответствует одному из пакетов, объявляющих зависимость (в этом процессе восстановления зависимостей предполагается, что независимые от архитектуры пакеты имеют такую же архитектуру что и хост). Зависимость также может быть ослаблена для того, чтобы позволить любой другой архитектуре выполнить её с помощью ***package:***любой синтаксис, но внешние пакеты могут удовлетворить подобную зависимость, если они отмечены как Multi-Arch: allowed.

8.3.6 Проверка подлинности пакета

Обновление системы представляет собой очень уязвимую операцию, так что вы должны полностью убедиться, что вы устанавливаете пакеты лишь из официальных репозиториев Kali. Если зеркало Kali, которое вы использовали, было взломано, то

человек, проделавший это, захочет добавить вредоносный код во вполне себе официальный пакет. Подобного рода пакет, в случае его установки, сможет делать абсолютно все, что в него было заложено взломщиком, включая хищение паролей или конфиденциальной информации. Для того чтобы обойти этот риск Kali предоставляет защищенный от злонамеренного вмешательства изолирующий слой – на время установки – который поможет удостовериться в том, что пакет реально происходит из официального источника и не был изменен кем-либо со стороны.

Изолирующий слой работает с цепочкой криптографической хэш-функции и подписью. Подписанный файл является Release файлом, предоставленным зеркалами Kali. Он содержит список файлов файловых пакетов (включая из сжатые формы, Packages.gz and Packages.xz, и инкрементные версии) вместе с их MD5, SHA1, и SHA256 хэшами, что в свою очередь гарантирует, что файлы еще не были подвергнуты вмешательству. Эти файлы пакетов содержат список пакетов Debian, доступных на зеркале, вместе с их хэшами, которые гарантируют в свою очередь, что пакеты сами по себе еще не были изменены.

Доверенные ключи управляются командой apt-key, которая находится в ***apt*** пакете. Эта программа поддерживает общественные ключи keyring GnuPG, которые используются для проверки подписей в файле Release.gpg, доступном на зеркалах. Он может использоваться для добавления новых ключей вручную (когда вам необходимы неофициальные зеркала). Однако, как правило, нужны только официальные ключи. Эти ключи автоматически поддерживаются в актуальном состоянии пакетом ***kali-archive-keyring*** (который помещает соответствующий keyrings в /etc/apt/trusted.gpg.d). Однако, первая установка отдельных пакетов, требует особого внимания: даже если пакет подписан как любой другой, подпись не может быть проверена извне. Осторожный администратор должен всегда идентифицировать импортируемые ключи, перед тем как доверять им и начинать установку новых пакетов.

```
# apt-key fingerprint
/etc/apt/trusted.gpg.d/debian-archive-jessie-automatic.gpg
-----------------------------------------------------
pub   4096R/2B90D010 2014-11-21 [expires: 2022-11-19]
      Key fingerprint = 126C 0D24 BD8A 2942 CC7D  F8AC 7638 D044 2B90 D010
uid                  Debian Archive Automatic Signing Key (8/jessie) <ftpmaster@debian.org>

/etc/apt/trusted.gpg.d/debian-archive-jessie-security-automatic.gpg
-----------------------------------------------------------------
pub   4096R/C857C906 2014-11-21 [expires: 2022-11-19]
      Key fingerprint = D211 6914 1CEC D440 F2EB  8DDA 9D6D 8F6B C857 C906
uid                  Debian Security Archive Automatic Signing Key (8/jessie) <ftpmaster@debian.org>

/etc/apt/trusted.gpg.d/debian-archive-jessie-stable.gpg
-----------------------------------------------------
pub   4096R/518E17E1 2013-08-17 [expires: 2021-08-15]
      Key fingerprint = 75DD C3C4 A499 F1A1 8CB5  F3C8 CBF8 D6FD 518E 17E1
uid                  Jessie Stable Release Key <debian-release@lists.debian.org>

/etc/apt/trusted.gpg.d/debian-archive-squeeze-automatic.gpg
-----------------------------------------------------
pub   4096R/473041FA 2010-08-27 [expires: 2018-03-05]
      Key fingerprint = 9FED 2BCB DCD2 9CDF 7626  78CB AED4 B06F 4730 41FA
uid                  Debian Archive Automatic Signing Key (6.0/squeeze) <ftpmaster@debian.org>

/etc/apt/trusted.gpg.d/debian-archive-squeeze-stable.gpg
-----------------------------------------------------
pub   4096R/B98321F9 2010-08-07 [expires: 2017-08-05]
      Key fingerprint = 0E4E DE2C 7F3E 1FC0 D033  800E 6448 1591 B983 21F9
uid                  Squeeze Stable Release Key <debian-release@lists.debian.org>

/etc/apt/trusted.gpg.d/debian-archive-wheezy-automatic.gpg
-----------------------------------------------------
pub   4096R/46925553 2012-04-27 [expires: 2020-04-25]
      Key fingerprint = A1BD 8E9D 78F7 FE5C 3E65  D8AF 8B48 AD62 4692 5553
uid                  Debian Archive Automatic Signing Key (7.0/wheezy) <ftpmaster@debian.org>

/etc/apt/trusted.gpg.d/debian-archive-wheezy-stable.gpg
-----------------------------------------------------
pub   4096R/65FFB764 2012-05-08 [expires: 2019-05-07]
      Key fingerprint = ED6D 6527 1AAC F0FF 15D1  2303 6FB2 A1C2 65FF B764
uid                  Wheezy Stable Release Key <debian-release@lists.debian.org>
```

```
/etc/apt/trusted.gpg.d/kali-archive-keyring.gpg
-----------------------------------------------
pub   4096R/7D8D0BF6 2012-03-05 [expires: 2018-02-02]
      Key fingerprint = 44C6 513A 8E4F B3D3 0875  F758 ED44 4FF0 7D8D 0BF6
uid                   Kali Linux Repository <devel@kali.org>
sub   4096R/FC0D0DCB 2012-03-05 [expires: 2018-02-02]
```

Когда в sources.list добавляется пакет из неофициального источника, APT должен быть уведомлен о том, что он может доверять соответствующему GPG ключу аутентификации (в противном случае он будет продолжать жаловаться, что он не может гарантировать подлинность пакетов, поступающих из этого репозитория). Первым шагом, безусловно, является получение публичного ключа. Чаще всего ключ будет предоставлен в виде небольшого текстового файла, который мы будем называть key.asc в следующих примерах.

Для того чтобы добавить ключ к доверенному keyring, администратор должен запустить run apt-key add < key.asc.

Другим способом является использование графического интерфейса synaptic: его вкладка Authentication в меню Settings - Repositories предоставляет возможность импорта ключей из файла key.asc.

Для людей, которые предпочитают специализированное приложение и более подробную информацию о доверенных ключах, можно использовать gui-apt-key (в пакете с тем же именем), небольшой пользовательский графический интерфейс, который управляет доверенным ключом.

После того, как соответствующие ключи находятся в keyring, APT проверит сигнатуры перед любой рискованной операцией, чтобы интерфейсы отображали предупреждение, если было предложено установить пакет, подлинность которого не может быть установлена.

8.4 Справка по пакетам: углубление в систему пакетов Debian

Наконец, пришло время углубиться в систему пакетов Debian и Kali. На этом этапе мы собираемся выйти за рамки инструментов и синтаксиса и сосредоточиться на сути системы пакетирования. Этот закулисный взгляд поможет вам понять основы работы APT и предоставит вам реальный взгляд на то, каким образом вы можете настроить и рационализировать вашу систему Kali. Вам не обязательно запоминать абсолютно весь материал, изложенный в этом разделе, но в любом случае полученная вами здесь информация послужит вам хорошую службу по мере вашего профессионального роста в освоении системы Kali Linux.

До сих пор вы взаимодействовали с данными пакета APT с помощью различных инструментов, предназначенных для взаимодействия с ним. Далее мы углубимся и заглянем внутрь пакетов и рассмотрим внутреннюю *метаинформацию* (или информацию о другой информации), используемую инструментами управления пакетами.

Эта комбинация файлового архива и метаинформации непосредственно видна в структуре файла .deb, который является просто архивом, объединяющим три файла:

```
$ ar t /var/cache/apt/archives/apt_1.4~beta1_amd64.deb
debian-binary
control.tar.gz
data.tar.xz
```

Двоичный файл debian содержит один номер версии, описывающий формат архива:

```
$ ar p /var/cache/apt/archives/apt_1.4~beta1_amd64.deb debian-binary
2.0
```

Архив control.tar.gz содержит метаинформацию:

```
$ ar p /var/cache/apt/archives/apt_1.4~beta1_amd64.deb control.tar.gz | tar -tzf -
./
./conffiles
./control
./md5sums
./postinst
./postrm
./preinst
./prerm
./shlibs
./triggers
```

И наконец, архив data.tar.xz (формат сжатия может отличаться) содержит фактические файлы, которые необходимо установить в файловой системе:

```
$ ar p /var/cache/apt/archives/apt_1.4~beta1_amd64.deb data.tar.xz | tar -tJf -
./
./etc/
./etc/apt/
./etc/apt/apt.conf.d/
./etc/apt/apt.conf.d/01autoremove
./etc/apt/preferences.d/
./etc/apt/sources.list.d/
./etc/apt/trusted.gpg.d/
./etc/cron.daily/
./etc/cron.daily/apt-compat
./etc/kernel/
./etc/kernel/postinst.d/
./etc/kernel/postinst.d/apt-auto-removal
./etc/logrotate.d/
./etc/logrotate.d/apt
./lib/
./lib/systemd/
[...]
```

Обратите внимание, что в данном примере, вы рассматриваете .deb пакет в архиве кэша APT, и что ваш архив может содержать файлы с номерами версий, которые могут отличаться от приведенных в примере.

В этом разделе, мы представим эту метаинформацию, которая содержится в каждом пакете, и мы покажем вам как её использовать.

8.4.1 Контрольный файл

Давайте начнем рассмотрение контрольного файла, который содержится в архиве control.tar.gz. Контрольный файл содержит самую необходимую информацию о пакете. Он использует структуру, похожую на заголовки электронных писем, и может быть просмотрен с помощью команды dpkg –I. Например, контрольный файл ***apt*** выглядит следующим образом:

```
$ dpkg -I apt_1.4~beta1_amd64.deb control
Package: apt
Version: 1.4~beta1
Architecture: amd64
Maintainer: APT Development Team <deity@lists.debian.org>
Installed-Size: 3478
Depends: adduser, gpgv | gpgv2 | gpgv1, debian-archive-keyring, init-system-helpers (>=
    ➥  1.18~), libapt-pkg5.0 (>= 1.3~rc2), libc6 (>= 2.15), libgcc1 (>= 1:3.0),
    ➥ libstdc++6 (>= 5.2)
Recommends: gnupg | gnupg2 | gnupg1
Suggests: apt-doc, aptitude | synaptic | wajig, dpkg-dev (>= 1.17.2), powermgmt-base,
    ➥ python-apt
Breaks: apt-utils (<< 1.3~exp2~)
Replaces: apt-utils (<< 1.3~exp2~)
Section: admin
Priority: important
Description: commandline package manager
 This package provides commandline tools for searching and
 managing as well as querying information about packages
 as a low-level access to all features of the libapt-pkg library.
 .
 These include:
  * apt-get for retrieval of packages and information about them
    from authenticated sources and for installation, upgrade and
    removal of packages together with their dependencies
  * apt-cache for querying available information about installed
    as well as installable packages
  * apt-cdrom to use removable media as a source for packages
  * apt-config as an interface to the configuration settings
  * apt-key as an interface to manage authentication keys
```

В этом разделе мы представим вам контрольный файл и разъясним вам различные области. Каждая из них даст вам лучшее понимание системы пакетирования, даст вам более точную настройку управления конфигурацией и предоставит вам информацию, необходимую для устранения проблем, которые могут возникнуть.

Зависимости: поле Depends

Зависимости пакета определяются в поле Depends в заголовке пакета. Это список условий для корректной работы пакета - эта информация используется такими инструментами, как apt для установки необходимых библиотек, в соответствующих версиях,

выполняющих зависимости устанавливаемого пакета. Для каждой зависимости вы можете ограничить диапазон версий, соответствующих этому условию Другими словами, можно выразить тот факт, что пакет *libc6* вам нужен в версии, равной или большей, чем «2.15» (пишется libc6 (>= 2.15)”).

Знаки операции сравнения выглядят следующим образом:

- <<: меньше чем;
- <=: меньше чем или равно;
- =: равно (но помните, что “2.6.1” не будет равняться “2.6.1-1”);
- >=: больше чем или равно;
- >>: больше чем.

В списке условий, которые должны выполняться, запятая служит разделителем, интерпретируемым как логическое «И». В условиях вертикальная черта (“|”) обозначает логическое «ИЛИ» (это включающее ИЛИ, не исключающее и/или (“either/or”)). Указывая больший приоритет, чем просто «И», вы можете использовать его столько раз, сколько необходимо. Таким образом, зависимость «(A ИЛИ B) и C» записывается A | B, C. Напротив, выражение «A ИЛИ (B И C)» должно быть записано как «(A ИЛИ B) И (A ИЛИ C)», поскольку поле Depends не допускает круглые скобки, которые изменяют порядок приоритетов между логическими операторами «ИЛИ» и «И». Следователь запись будет выглядеть следующим образом A | B, A | C. Для получения большей информации рекомендуем вам посетить сайт http://www.debian.org/doc/ debian-policy/ch- relationships.html

Система зависимостей является хорошим механизмом для гарантии работы программы, но она также имеет другие сферы применение с мета пакетами. Это пустые пакеты, которые описывают только зависимости. Они облегчают установку согласованной группы программ, предварительно выбранных создателем мета-пакета; как таковая команда apt install *meta-package* будет автоматически устанавливать все эти программы, используя зависимости мета-пакета. Пакеты *gnome, kde-full* и *kali-linux* - это примеры мета-пакетов.

Pre-Depends, более требовательные зависимости

Pre-dependencies (Предварительные зависимости), которые перечислены в поле Pre-Depends в заголовках пакетов, дополняют обычные зависимости; их синтаксис идентичный. Обычная зависимость указывает на то, что пакет, о котором идет речь, должен быть распакован и настроен перед настройкой пакета, объявляющего зависимость. Предварительная зависимость предусматривает, что пакет, о котором идет речь, должен быть распакован и настроен перед выполнением сценария предварительной установки пакета, объявляющего предварительную зависимость, которая идет перед его установкой.

Предварительная зависимость является очень требовательной к apt, потому что она добавляет строгое ограничение на порядок пакетов для установки. По сути, предварительные зависимости как таковые не используются до тех пор, пока в них не стаёт острая необходимость. Мы даже порекомендуем вам проконсультироваться с другими разработчиками на debian-devel@lists.debian.org перед тем, как добавлять предварительную зависимость, если все еще есть вероятность разрешения вашей проблемы без их использования.

Поля Recommends (Рекомендуемые), Suggests (Предложенные), и Enhances (Улучшенные)

Поля Recommends и Suggests описывают зависимости, которые не являются обязательными. Рекомендуемые зависимости, что является самым важным, значительно улучшают функциональность, предлагаемую пакетом, но не являются обязательными для его работы. Предложенные зависимости, вторичной важности, указывают на то, что определенные пакеты могут улучшить и дополнить свою соответственную полезность, но довольно разумным является установка одних без других.
Вам следует всегда устанавливать рекомендуемые пакеты, пока конкретно не будете уверены в том, что более не нуждаетесь в

них. И, наоборот, у вас нет необходимости устанавливать предложенные пакеты, пока вы не будете уверены в том, что они вам действительно нужны.

Поле Enhances также описывает предложение, но немного в другом контексте. Оно действительно находится в предложенном пакете , а не в пакете, который получит преимущество от этого предложения. Его интерес лежит в том, что существует возможность добавить предложение без изменения, связанного с ним пакета. Таким образом, все надстройки, плагины и другие расширения программы могут затем отображаться в списке предложений, связанных с программным обеспечением. Не смотря на то, что данное поле существует уже несколько лет, он до сих пор в значительной степени игнорируется программами, такими как apt или synaptic. Первоначальная цель состояла в том, чтобы позволить пакету, такому как *xul-ext-adblock-plus* (Firefox расширение) объявлять Enhances (Улучшения): firefox, firefox-esr, таким образом, появлялись в списке предложенных пакетов связанных с *firefox* и *firefox-esr.*

Конфликты: поле Conflicts

Поле Conflicts обозначает случаи, когда пакет нельзя установить одновременно вместе с другими. Самой часто распространенной причиной для этого является то, что оба пакета включают в себя файл с одним и тем же именем, предоставляют одну и ту же службу на одном и том же порту протокола управления передачей (transmission control protocol (TCP)) или же из-за того, что они препятствуют работе друг друга.

Если возникает случай, когда появляется конфликт с уже установленным пакетом, dpkg откажется устанавливать пакет, за исключением случаев, когда новый пакет указывает, что он заменит установленный пакет, и в этом случае dpkg решит заменить старый пакет на новый. APT всегда следует вашим инструкциям: если вы решите установить новый пакет, он автоматически предложит удалить пакет, представляющий проблему.

Несовместимости: Поле Breaks

Поле Breaks обладает похожим эффектом, подобным эффекту поля Conflicts, но в отличие от последнего оно имеет особое значение. Оно оповещает о том, что установка пакета навредит другому пакету (или конкретной версии пакета). В общем, эта несовместимость между двумя пакетами носит временный характер, а отношение Breaks конкретно относятся к несовместимым версиям.

Когда пакет нарушает работу уже установленного пакета, dpkg откажется устанавливать его, а apt попробует решить проблему путем обновления пакета, работа которого будет нарушена, до более новой версии (которая считается совместимой).

Такая ситуация может возникнуть в случае обновлений без обратной совместимости: имеется ввиду, что новая версия больше не работает с более старой версией и вызывает неисправность в другой программе без создания особых условий. Поле Breaks помогает предотвратить эти проблемы.

Предусмотренные пункты: Поле Provides

В этом поле представлена очень интересная концепция ***виртуального пакета (virtual package).*** Он имеет много ролей, но две из них особенно важны. Первая роль заключается в использовании пакета для связывания с ним общей службы (пакет предоставляет службу). Вторая роль указывает на то, что пакет полностью заменяет другой и что для достижения этой цели он может удовлетворять зависимости, которые также удовлетворяют другие пакеты. Таким образом можно создать пакет замещения без использования одного и того же имени пакета.

Мета пакет и виртуальный пакет

Очень важно четко отличать мета пакеты от виртуальных пакетов. Последние являются реальными пакетами (включая реально существующие .deb файлы), чья задача состоит в выражении зависимостей.

Но виртуальные пакеты физически не существуют; они являются лишь средством идентификации реальных пакетов на основе общих логических критериев (например, предоставляемых услуг или совместимости со стандартной программой или уже существующим пакетом).

Предоставление услуги (Providing a Service)
Давайте обсудим первый случай более детально на конкретном примере: все почтовые серверы, такие как ***postfix*** или ***sendmail*** должны предоставлять виртуальный пакет ***mail-transport-agent.*** Таким образом, любой пакет, для которого эта служба должна быть функциональной (например, диспетчер списка рассылки, такой как smartlist или sympa), просто заявляет в своих зависимостях, что для этого требуется ***mail-transport-agent*** вместо указания большого, но неполного списка возможных решений. Кроме того, абсолютно бессмысленно устанавливать два почтовых сервера на одной и той же машине, именно поэтому каждый из этих пакетов заявляет о конфликте с ***mail-transport-agent*** виртуальным пакетом. Конфликт между пакетом и им самим игнорируется системой, но этот метод будет запрещать установку двух почтовых серверов на одном компьютере.

Взаимозаменяемость с другим пакетом

Поле Provides также является интересным, когда содержание пакета включено в более крупный пакет. Например, модуль Perl ***libdigest-md5-perl*** выступает необязательным модулем в Perl 5.6, и был встроен в качестве стандартного только в Perl 5.8. Таким образом, начиная с версии 5.8, пакет ***perl*** заявляет Iibdigest-md5-perl таким образом, что все зависимости на этом пакете удовлетворяются, если на системе установлен Perl 5.8 (или более новый). Пакет ***libdigest-md5-perl*** сам по себе был удален, т.к. у него более нет никаких задач, с которыми бы не могли справиться другие пакеты.

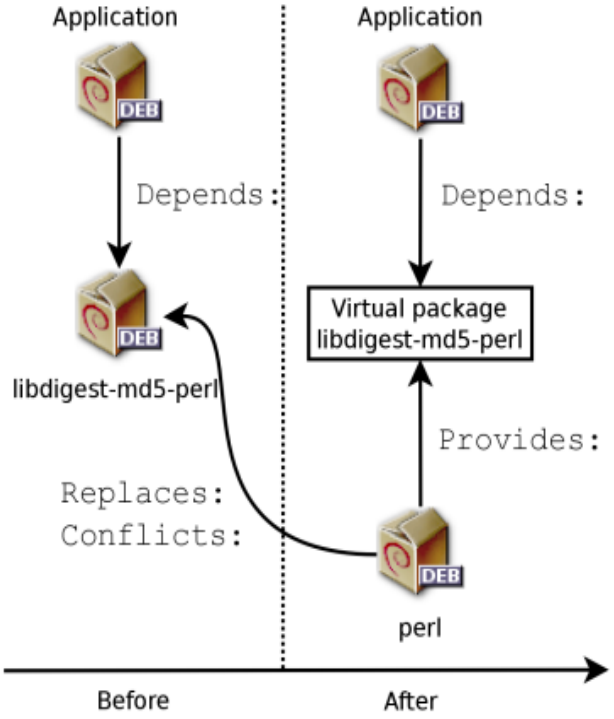

Рисунок 8.3 *Использование поля Provides для того, чтобы не нарушить зависимость.*

Это свойство является очень полезным, т.к. никогда не знаешь, какие причуды можно ожидать от процесса разработки и необходимо иметь возможность приспосабливаться к переименованию и другой автоматической замене устаревшего программного обеспечения.

Замена файлов: Поле Replaces

Поле Replaces обозначает, что пакет содержит файлы, которые также присутствуют в другом пакете и что данный пакет законно обладает правами на их замену. Без этой спецификации, dpkg не сможет быть выполнен, утверждая, что он не может перезаписать файлы другого пакета (технически возможно заставить его сделать это с помощью опции -force-overwrite, но это не является стандартной операцией). Это позволяет выявить потенциальные проблемы и требует от специалиста по обслуживанию изучить вопрос пред тем, как принять решение о добавлении такого поля.

Использование этого поля оправдано, когда имена пакетов изменяются или когда пакет включен в другой. Это также может происходить, когда специалист по обслуживанию решает распространять файлы по-разному среди различных двоичных пакетов, созданных одним и тем же источником пакетов: замещенный файл больше не принадлежит старому пакету, теперь он принадлежит только новому.

Если все файлы в установленных пакетах были заменены, пакет считается удаленным. И наконец, это поле также способствует удалению замещенных пакетов с помощью dpkg в случае возникновения конфликта.

8.4.2 Скрипты конфигурации

В дополнение к управляющему файлу, control.tar.gz архив для каждого пакета Debian может содержать определенное количество скриптов (postinst, postrm, preinst, prerm), вызываемых dpkg на разных этапах обработки пакета. Мы можем использовать dpkg -I для отображения этих файлов, поскольку они находятся в архиве пакета .deb:

```
$ dpkg -I /var/cache/apt/archives/zsh_5.3-1_amd64.deb | head
 new debian package, version 2.0.
 size 814486 bytes: control archive=2557 bytes.
     838 bytes,    20 lines      control
    3327 bytes,    43 lines      md5sums
     969 bytes,    41 lines   *  postinst             #!/bin/sh
     348 bytes,    20 lines   *  postrm               #!/bin/sh
     175 bytes,     5 lines   *  preinst              #!/bin/sh
     175 bytes,     5 lines   *  prerm                #!/bin/sh
 Package: zsh
 Version: 5.3-1
$ dpkg -I zsh_5.3-1_amd64.deb preinst
#!/bin/sh
set -e
# Automatically added by dh_installdeb
dpkg-maintscript-helper symlink_to_dir /usr/share/doc/zsh zsh-common 5.0.7-3 -- "$@"
# End automatically added section
```

Debian Policy детально описывает каждый из этих файлов, указывая вызванные скрипты и аргументы, которые они получают. Эти последовательности могут быть довольно сложными, т.к. если один из этих скриптов потерпит неудачу, dpkg попробует вернуться к удовлетворительному состоянию путем отмены идущего на данный момент процесса установки или удаления (насколько это будет возможным).

База данных dpkg

Вы можете перемещать базу данных dpkg в файловой системе в / var / lib / dpkg /. Этот каталог содержит текущую запись обо всех пакетах, которые были установлены в системе. Все скрипты конфигурации для установленных пакетов хранятся в каталоге / var / lib / dpkg / info / в виде файла с префиксом имени пакета:

```
$ ls /var/lib/dpkg/info/zsh.*
/var/lib/dpkg/info/zsh.list
/var/lib/dpkg/info/zsh.md5sums
/var/lib/dpkg/info/zsh.postinst
/var/lib/dpkg/info/zsh.postrm
/var/lib/dpkg/info/zsh.preinst
/var/lib/dpkg/info/zsh.prerm
```

Эта директория также включает в себя файл с расширением .list для каждого пакета, который содержит список файлов принадлежащих этому пакету:

```
$ head /var/lib/dpkg/info/zsh.list
/.
/bin
/bin/zsh
/bin/zsh5
/usr
/usr/lib
/usr/lib/x86_64-linux-gnu
/usr/lib/x86_64-linux-gnu/zsh
/usr/lib/x86_64-linux-gnu/zsh/5.2
/usr/lib/x86_64-linux-gnu/zsh/5.2/zsh
[...]
```

Файл /var/lib/dpkg/status содержит серию блоков данных (в формате известного запроса заголовков почты для комментария RFC 2822), описывающих состояние каждого пакета. Информация из управляющего файла установленных пакетов также дублируется там.

```
$ more /var/lib/dpkg/status
Package: gnome-characters
Status: install ok installed
Priority: optional
Section: gnome
Installed-Size: 1785
Maintainer: Debian GNOME Maintainers <pkg-gnome-
    ➥ maintainers@lists.alioth.debian.org>
Architecture: amd64
Version: 3.20.1-1
[...]
```

Давайте обсудим файлы конфигурации и посмотрим, как они взаимодействуют. В общем, скрипт preinst выполняется до установки пакета, в то время как postins следует за ней. Подобно этому prerm вызывается перед удалением пакета, а postrm после этого. Обновление пакета является эквивалентным по отношению к удалению предыдущей версии и установки новой. К сожалению, невозможно детально описать все сценарии возможные здесь, но, тем не менее, мы обсудим два самых распространенных: установка/обновление и удаление.

Эти последовательности могут быть довольно запутывающими, но визуальное представление может помочь. Manoj Srivastava сделал эти схемы (графики), объясняя то, как именно вызываются

скрипты конфигурации с помощью dpkg. Похожие схемы (графики) были также разработаны проектом Debian Women; они являются более простыми для понимания, но менее полными.

https://people.debian.org/~srivasta/MaintainerScripts.html
https://wiki.debian.org/MaintainerScripts

ПРЕДОСТЕРЕЖЕНИЕ Символические имена скриптов

Последовательности, описанные в этом разделе, вызывают скрипты конфигурации по конкретным именам, например old-prerm или new-postinst. Это, соответственно, скрипт prerm, содержащийся в старой версии пакета (установленный перед обновлением) и postinst-скрипт, содержащийся в новой версии (установленной обновлением).

Установка и обновление последовательности скриптов

Ниже описано то, что происходит во время процесса установки (или обновления):

1. Для обновления, dpkg вызывает old-prerm upgrade *new-version.*
2. Также для обновления, dpkg выполняет new-preinst upgrade *old-version;* для первой установки, он выполняет new-preinst install. Он может добавлять старую версию в последний параметр, если пакет уже был установлен и удален (но не очищен, файлы конфигурации были сохранены).
3. Далее новые файлы пакеты распаковываются. Если файл уже существует, он заменяется, но все равно создается резервная копия и хранится во временных файлах.
4. Для обновления, dpkg выполняет old-postrm upgrade *new-version.*
5. dpkg обновляет все внутренние данные (список файлов, скрипты конфигурации, и т.д.) и удаляет все резервные копии замещенных файлов. В данном случае это является точкой невозврата: dpkg больше не имеет доступа ко всем тем

элементам, которые необходимы для возвращения к предыдущему состоянию.
6. dpkg обновит файлы конфигурации, предложив вам решить, не может ли он автоматически управлять этой задачей. Детали данной процедуры подробно изложены в разделе 8.4.3, “Сигнатуры и файлы конфигурации” [page 214].
7. И наконец, dpkg настраивает пакет путем выполнения new-postinst configure *last- version-configured.*

Удаление пакета

Ниже описано то, что происходит во время удаления пакета

1. dpkg вызывает prerm remove.
2. dpkg удаляет все файлы пакета, за исключением файлов конфигурации и сценариев конфигурации.
3. dpkg выполняет postrm remove. Все сценарии конфигурации, кроме postrm, удаляются. Если вы не использовали опцию purge, то процесс останавливается на этом этапе.
4. Для полной очистки пакета (команда, заданная с dpkg --purge or dpkg -P), необходимо удалить и файлы конфигурации, также как и определенное количество копий (*.dpkg- tmp, *.dpkg-old, *.dpkg-new) и временных файлов; dpkg затем выполняет postrm purge.

В некоторых случаях пакет может использовать debconf для получения информации о конфигурации от вас: четыре сценария, описанные выше, затем дополняются скриптом config, предназначенным для получения этой информации. Во время установки этот скрипт подробно определяет, какие вопросы задает debconf. Ответы записываются в базу данных debconf для дальнейшего использования. Сценарий обычно выполняется apt перед последовательной установкой пакетов, чтобы сгруппировать все вопросы вместе в начале процесса. Затем pre- и post- скрипты установки могут использовать эту информацию для работы в соответствии с вашими пожеланиями.

Инструмент debconf

Инструмент debconf был создан для решения повторяющейся проблемы в Debian. Все пакеты Debian не могут функционировать без конфигурации, которая используется для постановки вопросов с вызовами echo и read команд в оболочке скриптов postinst (и других похожих скриптов). Это заставляло установщик присматривать за крупными установками или обновлениями для того, чтобы отвечать на различные конфигурационные запросы по мере их возникновения. На сегодняшний день, благодаря debconf, мы можем почти полностью обходиться без подобного рода ручного взаимодействия.

Инструмент debconf обладает большим количеством интересных функций: он требует от разработчика указать взаимодействие с пользователем; он позволяет локализовать все отображаемые строки (все переводы хранятся в файле шаблонов, описывающих взаимодействия); он предоставляет различные интерфейсы (текстовый режим, графический режим, не интерактивный); и он разрешает создание центральной базы данных ответов для того, чтобы иметь возможность делиться одной и той же конфигурацией с несколькими компьютерами. Самым важным свойством является то, что все вопросы могут быть представлены в одном ряде, все вместе, перед началом длительного процесса установки или обновления. Теперь вы можете смело заниматься своими делами, пока система абсолютно самостоятельно проводит установку. Теперь у вас нет никакой необходимости оставаться на месте, уставившись в экран в ожидании вопросов, которые могут возникнуть в любую минуту.

8.4.3 Сигнатуры и конфигурационные файлы (conffiles) (внести правки в общее содержание)

Кроме скриптов поддержки и данных управления уже упомянутых а предыдущем разделе, архив control.tar.gz пакета Debian может содержать другие интересные файлы.

```
# ar p /var/cache/apt/archives/bash_4.4-2_amd64.deb control.tar.gz | tar -tzf -
```

```
./
./conffiles
./control
./md5sums
./postinst
./postrm
./preinst
./prerm
```

Первый—md5sums—содержит сигнатур MD5 для всех файлов пакета. Главным его преимуществом является то, что он допускает dpkg –verify для того, чтобы проверить были ли эти файлы изменены с момента их установки. Обратите внимание: если этот файл не существует, то dpkg будет генерировать его динамически во время установки (и хранить его в базе данных dpkg, как и другие управляющие файлы).

Conffiles перечисляет файлы пакетов, которые должны быть обработаны в качестве файлов конфигурации. Файлы конфигурации могут быть изменены администратором, а dpkg, в свою очередь, постарается сохранить эти изменения во время обновления пакета.

Фактически, в этой ситуации dpkg ведет себя настолько разумно насколько это возможно: если стандартный файл конфигурации не изменился между двумя версиями, он ничего не делает. Если же файл был изменен, он попытается обновить этот файл. Возможны два случая: либо администратор в принципе не касался этого файла конфигурации, и в этом случае dpkg автоматически устанавливает новую версию; или файл был изменен, и в этом случае dpkg запрашивает у администратора версию, которую он хочет использовать (старую версию с модификациями или новую, предоставленную пакетом). Чтобы помочь в принятии подобного рода решения, dpkg предлагает отобразить diff, который показывает разницу между двумя версиями. Если вы решите сохранить старую версию, новая будет сохранена в том же месте в файле с суффиксом .dpkg-dist. Если же вы выберете оставить новую версию, старая будет сохранена в файле с суффиксом .dpkg - old. Другое доступное действие состоит из кратковременного прерывания работы dpkg для того, чтобы отредактировать файл и попытаться восстановить соответствующие изменения (ранее идентифицированные с помощью diff).

dpkg обрабатывает обновления файла конфигурации, но при этом регулярно прерывает свою работу, запрашивая ввод от администратора. Это часто может быть довольно таки трудоемким и неудобным. К счастью, вы можете дать команду dpkg ответить на эти запросы автоматически. Параметр --force-confold сохраняет старую версию файла, а --force-confnew будет использовать новую версию. Эти варианты соблюдаются, даже если файл не был изменен администратором, что довольно редко дает желаемый эффект. Добавление опции --force-confdef говорит dpkg решать вопрос самостоятельно там, где это возможно (другими словами, когда исходный файл конфигурации не был затронут). Опции --force-confnew или --force-confold следует использовать для других случаев.

Эти опции используют dpkg, но большую часть времени администратор будет работать непосредственно с программами aptitude или apt. Тем не менее, необходимо знать синтаксис, используемый для указания параметров, передаваемых команде dpkg (их интерфейсы командной строки очень похожи).

```
# apt -o DPkg::options::="--force-confdef" -o DPkg::options::="--force-confold" full-
    ➥ upgrade
```

Эти опции могут храниться непосредственно в конфигурациях apt. Для того чтобы сделать это просто напишите следующую строку в файле /etc/apt/apt.conf.d/local:

```
DPkg::options { "--force-confdef"; "--force-confold"; }
```

Включение этой опции в файл конфигурации означает, что она также будет использована в графическом интерфейсе, таком как aptitude.

И наоборот, вы также можете заставить dpkg задавать вопросы конфигурационного файла. Опция --force-confask поручает dpkg отобразить вопросы о файлах конфигурации, даже в тех случаях, когда это не является необходимым. Таким образом, при переустановке пакета с этой опцией dpkg снова задаст вопросы для всех файлов конфигурации, измененных администратором.

Это очень удобно, особенно для переустановки исходного файла конфигурации, если он был удален, и у вас нет другой копии: обычная переустановка не сработает, потому что dpkg воспринимает удаление, как форму легального изменения и, таким образом, не устанавливает необходимый файл конфигурации.

8.5 Подведем итоги

В этом разделе, мы узнали больше о системе пакетов Debian, обсудили Advanced Package Tool (APT) и dpkg, узнали больше о базовом взаимодействии пакетов, дополнительной настройке и использовании APT, и углубились в изучение системы пакетов Debian с краткой ссылкой файла формата .deb. Мы рассмотрели такие понятия как файл управления, скрипты конфигурации, сигнатуры и файлы конфигурации (conffiles).

Основные моменты:

- Пакет Debian представляет собой сжатый архив программного приложения. Он содержит файлы приложения, а также другие метаданные, включая имена зависимостей, которые требуются приложениям, а также скрипты, которые позволяют выполнять команды на разных этапах существования пакета (установка, удаление, обновление).
- Инструмент dpkg, в отличие от apt и apt-get (семейства APT), не имеет необходимой информации о всех доступных пакетах, которые могут быть использованы для удовлетворения зависимостей пакетов. Таким образом, для управления пакетами Debian, вы, вероятно, будете использовать последние инструменты, поскольку они могут автоматически разрешать проблемы с зависимостями.
- Вы можете использовать APT для установки и удаления приложений, обновления пакетов и даже для обновления всей вашей системы. Ниже приведены основные моменты, которые вам необходимо знать об APT и его конфигурациях:
- Файл sources.list является ключевым файлом конфигурации для определения источников пакетов (или репозиториев, содержащих пакеты);

- Debian и Kali используют три раздела для дифференциации пакетов в соответствии с лицензиями, выбранными авторами каждой работы: содержит все пакеты, которые полностью соответствуют Директиве по свободному программному обеспечению Debian (Debian Free Software Guidelines[31]); non-free содержит программное обеспечение, которое не полностью соответствует Директиве по свободному программному обеспечению, но тем не менее может быть распространено без ограничений; и contrib (contributions) включает программное обеспечение с открытым исходным кодом, которое не может функционировать без каких-либо non-free элементов;
- Kali поддерживает несколько репозиториев, в том числе: kali-rolling, который является основным хранилищем для конечных пользователей и всегда должен содержать устанавливаемые и последние пакеты; kali-dev, который используется разработчиками Kali и не предназначен для публичного использования; и kali-bleeding-edge, который часто содержит непроверенные пакеты, автоматически создаваемые из репозитория Git (или Subversion), менее чем через двадцать четыре часа после их загрузки;
- Работая с APT, вам необходимо сначала загрузить список доступных на данный момент пакетов с помощью apt update;
- Вы можете добавить пакет в систему с помощью apt install *package.* APT автоматически установит все необходимые зависимости;
- Для того чтобы удалить пакет, используйте apt remove *package.* Он также устранит обратную зависимость пакета (то есть пакеты, которые зависят от пакета, который нужно удалить);
- Для того чтобы удалить все данные связанные с пакетом, вы можете «очистить» пакет с помощью команды apt purge *package*. В отличие от процесса удаления, это не только удалит пакет, но также удалит и его файлы конфигурации и иногда даже связанные с ним данные пользователя.

Мы рекомендуем регулярные обновления для установки последних обновлений безопасности. Чтобы обновить, используйте apt update, за которым следуют apt upgrade, apt-get upgrade или aptitude safe-upgrade. Эти команды ищут

[31] https://www.debian.org/social_contract#guidelines

установленные пакеты, которые можно обновить без удаления каких-либо пакетов.
Для более важных обновлений, таких как обновление основных версий, используйте apt full-upgrade. С помощью этой команды apt завершит обновление, даже если ему нужно удалить некоторые устаревшие пакеты или установить новые зависимости. Это также команда, которую вы должны использовать для регулярных обновлений вашей системы Kali Rolling. Просмотрите все плюсы и минусы обновлений, описанных в соответствующей главе.

Для проверки пакетов Debian можно использовать несколько инструментов:

- dpkg --listfiles *package* (или -L) перечисляет файлы, которые были установлены в указанных пакетах.
- dpkg --search *file* (или -S) находит любые пакеты, которые содержат файл или путь указанный в аргументе.
- dpkg --list (или -I) выводит на экран список пакетов известных систем, а также их статус установки.
- dpkg --contents *file.deb* (или -c) перечисляет все файлы в конкретном .deb файле.
- dpkg --info *file.deb* (или-I) выводит на экран заголовки указанного .deb файла.
- Различные apt-cache подкоманды большую часть информации, хранящейся во внутренней базе данных APT.

Чтобы избежать чрезмерного использования диска, вы должны регулярно сортировать /var/cache/apt/archives/. Для этого можно использовать две команды: apt clean (или apt-get clean), которая полностью опустошает каталог; apt autoclean (apt-get autoclean), которая удаляет только те пакеты, которые больше не могут быть загружены, т.к. они исчезли из зеркала, и поэтому являются бесполезными.

Aptitude - это интерактивная программа, которая может использоваться в полуграфическом режиме на консоли. Это чрезвычайно надежная программа, которая может помочь вам установить и устранить неполадки в пакетах.

synaptic - графический менеджер пакетов, который имеет чистый и эффективный графический интерфейс.

Как продвинутый пользователь, вы можете создавать файлы в /etc/apt/apt.conf.d/ для настройки определенных аспектов APT. Вы также можете управлять приоритетами пакетов, отслеживать автоматически установленные пакеты, работать с несколькими дистрибутивами или архитектурами одновременно, использовать криптографические подписи для проверки пакетов и обновлять файлы, используя методы, описанные в соответствующей главе.

Несмотря на все усилия, предпринимаемые разработчиками Kali / Debian, обновление системы не всегда проходит так гладко, как мы надеемся. Если вы столкнулись с каким-либо проблемами, вы можете посетить Kali bug tracker[32] или Debian bug tracking system[33] на https://bugs.debian.org/package для проверки отчетов о решении данного рода проблемы. Вы также можете попытаться понизить в статусе пакет или отладить и восстановить неудачный сценарий поддержки пакета.

[32]http://bugs.kali.org
[33]https://bugs.debian.org

Часть 9: Расширенное использование системы

Содержание:

Ключевые слова главы:

- Пользовательские пакеты;
- Пользовательское ядро;
- Пользовательские изображения;
- live-build;
- Persistence;

Kali был разработан как высокомодульный настраиваемый фреймворк для тестирования на проникновение, который позволяет довольно продвинутую настройку и использование. Настройка может происходить на нескольких уровнях, начиная с уровня исходного кода. Источники всех пакетов Kali являются общедоступными. В этой главе мы покажем, как вы можете извлекать пакеты, изменять их и как создавать из них ваши собственные настраиваемые пакеты. Ядро Linux заслуживает отдельного внимания, и именно поэтому мы посвятили ему целый раздел (раздел 9.2, “Перекомпиляция ядра Linux” [стр. 232]), в котором подробно обсуждается, где можно найти источники, как настроить сборку ядра и, наконец, как скомпилировать его и как создать связанные с ядром пакеты.

Второй уровень настройки являет собой процесс создания живого образа ISO. Мы покажем вам, каким образом инструмент live-build предлагает множество различных приемов и опций конфигурации для настройки итогового образа ISO, включая возможность использовать пользовательские пакеты Debian вместо пакетов доступных на зеркалах.

Мы также обсудим, как вы можете создать постоянную живую сборку ISO на USB накопители, который будет сохранять файлы и изменения, внесенные на операционной системе между перезагрузками.

9.1 Модифицируем пакеты Kali

Изменение пакетов Kali обычно является задачей для сотрудников и разработчиков Kali: они обновляют пакеты на более новые версии, они настраивают конфигурацию по умолчанию для лучшей интеграции в дистрибутив, а также они решают проблемы, указанные пользователями в поданных отчетах об ошибках. Но все это не отрицает тот факт, что у вас может возникнуть особая необходимость, которая не была выполнена официальными пакетами и, в связи с этим, знание того, как модифицировать и изменять пакеты может быть очень полезным для вас.

Вы можете поинтересоваться, зачем вообще вам необходимо беспокоиться по поводу пакетов? В конце концов, если вам приходиться изменять какую-либо часть программного обеспечения, вы всегда можете получить его исходный код (обычно с помощью git) и запустить измененную и модифицированную версию прямо из исходного кода. Это, безусловно, очень удобно, если это возможно, и когда вы используете свою домашнюю директорию для этих целей, но если же ваше приложение требует общесистемной установки (например, с помощью шага make install), то оно засорит вашу файловую систему различными файлами, которые неизвестны dpkg и которые в скором времени создадут проблемы, не решаемые зависимостями пакетов. Кроме того, с помощью верных пакетов вы сможете делиться своими изменениями и с легкостью запускать их на множестве других компьютеров или же обращать изменения, если они работают не так, как вам хотелось бы.

Итак, когда бы вы хотели изменить пакет? Давайте рассмотрим несколько примеров. Для начала, мы предположим, что вы являетесь уверенным пользователем SET и вы заметили более новый выпуск, но так сложилось, что все разработчики Kali заняты на конференциях, а вы хотите опробовать этот новый выпуск как можно скорее. Вы желаете обновить пакеты самостоятельно. В другом случае, мы предположим, что вы всеми силами пытаетесь сделать так, чтобы ваша MIFARE NFC заработала, и вы хотите восстановить «libfreefare» для того, чтобы включить отладочные сообщения, чтобы предоставить данные о действиях для указания их в отчете об ошибке, который вы сейчас готовите. И в последнем случае мы предположим, что программа “pyrit” выдает сообщение об ошибке cryptic. После поиска в интернете, вы находите исправление в репозитории GitHub, которое может помочь вам решить возникшую проблему, и хотите изменить пакет с использованием этого исправления.

Мы детально пройдемся по всем этим примерам в следующем разделе. Мы попытаемся обобщить наши разъяснения таким образом, чтобы вы смогли лучше применять инструкции в различных случаях, но, к сожалению, невозможно затронуть все ситуации, с которыми вы можете столкнуться. Если вы столкнулись с проблемой, примените свои навыки для разрешения ситуации или попробуйте найти решение в интернете, также

рекомендуем вам искать ответы на самых подходящих форумах (смотри главу 6 «Получение помощи» [стр. 124]).

Какие бы изменения вы не хотели бы произвести, общий процесс всегда будет выглядеть одинаково: получить источник пакета, запустить его, произвести желаемые изменения, а затем создать пакет. Но для каждого шага, всегда существует множество инструментов, которые могут помочь справиться с задачей. Мы затронули самые релевантные и самые популярные инструменты, но наш обзор не является исчерпывающим.

9.1.1 Получение источников

Переделывание пакета Kali начинается с получения его исходного кода. Исходный пакет состоит из нескольких файлов: главный файл *.dsc *(Debian Source Control)* содержит список всех других файлов, которые могут быть в формате *.tar. *gz, bz2, xz,* иногда в *.diff.gz, или *.debian.tar. *gz, bz2,xz*.

Исходные пакеты хранятся на зеркалах Kali, которые доступны через HTTP. Вы можете использовать браузер для загрузки всех необходимых файлов, но самый простой способ сделать это заключается в использовании команды apt source *source_package_name*. Эта команда требует строчку deb-src в файле /etc/apt/sources.tist и обновленные индекс файлы (что можно сделать путем выполнения команды apt update). По умолчанию, Kali не добавляет требуемую строку, так как немногие пользователи Kali действительно должны извлекать исходные пакеты, но вы можете легко добавить их (см. пример файла в разделе 8.1.3, «Kali репозитории» [стр. 173] и соответствующие объяснения в разделе 8.1.2 «Правильное понимание файла sources.tist» [стр. 172]).

```
$ apt source libfreefare
Reading package lists... Done
NOTICE: 'libfreefare' packaging is maintained in the 'Git' version control system at:
git://anonscm.debian.org/collab-maint/libnfc.git
Please use:
git clone git://anonscm.debian.org/collab-maint/libnfc.git
to retrieve the latest (possibly unreleased) updates to the package.
Need to get 119 kB of source archives.
Get:1 http://archive-2.kali.org/kali kali-rolling/main libfreefare 0.4.0-2 (dsc) [2,090 B]
Get:2 http://archive-2.kali.org/kali kali-rolling/main libfreefare 0.4.0-2 (tar) [113 kB]
Get:3 http://archive-2.kali.org/kali kali-rolling/main libfreefare 0.4.0-2 (diff) [3,640 B]
Fetched 119 kB in 1s (63.4 kB/s)
gpgv: keyblock resource '/home/rhertzog/.gnupg/trustedkeys.gpg': file open error
gpgv: Signature made Tue 04 Mar 2014 06:57:36 PM EST using RSA key ID 40AD1FA6
gpgv: Can't check signature: public key not found
dpkg-source: warning: failed to verify signature on ./libfreefare_0.4.0-2.dsc
dpkg-source: info: extracting libfreefare in libfreefare-0.4.0
dpkg-source: info: unpacking libfreefare_0.4.0.orig.tar.gz
dpkg-source: info: unpacking libfreefare_0.4.0-2.debian.tar.xz
$ cd libfreefare-0.4.0
$ ls
AUTHORS    CMakeLists.txt  COPYING   HACKING            m4           README
ChangeLog  configure.ac    debian    libfreefare        Makefile.am  test
cmake      contrib         examples  libfreefare.pc.in  NEWS         TODO
$ ls debian
changelog  copyright               libfreefare-dev.install  rules
compat     libfreefare0.install    libfreefare-doc.install  source
control    libfreefare-bin.install README.Source            watch
```

В этом примере, в то время как мы получили исходный пакет из зеркала Kali, пакет является таким же, как и в Debian, т.к. строка версии не содержит “kali.” Это означает, что никаких изменений, связанных с кали, не было произведено.

Если вам нужна конкретная версия исходного пакета, которая, по тем или иным причинам, на данный момент недоступна в репозиториях, перечисленных в /etc/apt/sources.list, то самым простым способом для загрузки будет найти URL необходимого файла .dsc путем поиска на http://pkg.kali.org и затем передача этого URL через dget (из пакета devscripts package).

После поиска URL исходного пакета libreefare в kali-bleeding-edge, вы можете скачать его с помощью dget. Сначала оно скачает файл .dsc, затем проанализирует его, чтобы узнать, на что ссылаются другие файлы, и затем загрузит их из того же места:

```
$ dget http://http.kali.org/pool/main/libf/libfreefare/libfreefare_0.4.0+0~
    ➥ git1439352548.ffde4d-1.dsc
dget: retrieving http://http.kali.org/pool/main/libf/libfreefare/libfreefare_0.4.0+0~
    ➥ git1439352548.ffde4d-1.dsc
  % Total    % Received % Xferd  Average Speed   Time    Time     Time  Current
                                 Dload  Upload   Total   Spent    Left  Speed
100   364  100   364    0     0    852      0 --:--:-- --:--:-- --:--:--   854
100  1935  100  1935    0     0   2650      0 --:--:-- --:--:-- --:--:-- 19948
dget: retrieving http://http.kali.org/pool/main/libf/libfreefare/libfreefare_0.4.0+0~
    ➥ git1439352548.ffde4d.orig.tar.gz
[...]
dget: retrieving http://http.kali.org/pool/main/libf/libfreefare/libfreefare_0.4.0+0~
    ➥ git1439352548.ffde4d-1.debian.tar.xz
[...]
libfreefare_0.4.0+0~git1439352548.ffde4d-1.dsc:
dscverify: libfreefare_0.4.0+0~git1439352548.ffde4d-1.dsc failed signature check:
gpg: Signature made Wed Aug 12 06:14:03 2015 CEST
gpg:                using RSA key 43EF73F4BD8096DA
gpg: Can't check signature: No public key
Validation FAILED!!
$ dpkg-source -x libfreefare_0.4.0+0~git1439352548.ffde4d-1.dsc
gpgv: Signature made Wed Aug 12 06:14:03 2015 CEST
gpgv:                using RSA key 43EF73F4BD8096DA
gpgv: Can't check signature: No public key
dpkg-source: warning: failed to verify signature on ./libfreefare_0.4.0+0~git1439352548
    ➥ .ffde4d-1.dsc
dpkg-source: info: extracting libfreefare in libfreefare-0.4.0+0~git1439352548.ffde4d
dpkg-source: info: unpacking libfreefare_0.4.0+0~git1439352548.ffde4d.orig.tar.gz
dpkg-source: info: unpacking libfreefare_0.4.0+0~git1439352548.ffde4d-1.debian.tar.xz
```

Стоит отметить, что dget автоматически не извлекал исходный пакет, потому что он не мог проверить подпись PGP в исходном пакете. Таким образом, мы сделали этот шаг вручную с помощью dpkg- source -x *dsc-file* .Вы также можете принудительно удалить исходный пакет, указав параметр -allow-unauthenticated или опцию -u. И наоборот, вы можете использовать -download-only, чтобы пропустить шаг извлечения исходного пакета.

Извлечение источника из Git

Возможно, вы заметили, что вызов apt source указывает вам возможный репозиторий Git, который используется для поддержки пакета. Он может указывать на репозиторий Debian Git или на репозиторий Kali Git.

Все пакеты, специфичные для Kali, хранятся в репозиториях Git, размещенных на git.kali.org[34]. Вы можете получить источники из этих репозиториев с помощью git clone git://git.kali.org/packages/source-package. Если операция не дает ожидаемых источников, попробуйте переключиться на ветку kali / master с помощью git checkout kali / master.

В отличие от того, что вы получаете с помощью источника apt, полученное дерево не будет автоматически применять изменения. Загляните в debian/patches/, чтобы узнать о возможных изменениях, внесенных Kali.

```
$ git clone git://git.kali.org/packages/kali-meta
Cloning into 'kali-meta'...
remote: Counting objects: 760, done.
remote: Compressing objects: 100% (614/614), done.
remote: Total 760 (delta 279), reused 0 (delta 0)
Receiving objects: 100% (760/760), 141.01 KiB | 0 bytes/s,
    ➥ done.
Resolving deltas: 100% (279/279), done.
Checking connectivity... done.
$ cd kali-meta
$ ls
debian
$ ls debian
changelog  compat  control  copyright  rules  source
```

Вы можете использовать репозитории git как другой способ извлечения источников и, следовательно, (в большинстве случаев) следовать другим инструкциям из этого раздела. Но когда разработчики Kali работают с этими репозиториями, они используют другой процесс пакетирования и используют инструменты из пакета git-buildpackage, которые мы здесь не будем затрагивать. Вы можете узнать больше об этих инструментах здесь:
https://honk.sigxcpu.org/piki/projects/git-buildpackage/

[34] http://git.kali.org

9.1.2 Установка зависимостей сборки

Теперь, когда у вас есть источники, вам все равно нужно установить зависимости сборки. Они будут необходимы для создания желаемых бинарных пакетов, но также, вероятно, необходимы для частичных сборок, которые вы, возможно, захотите запустить, чтобы протестировать изменения по мере их создания.

Каждый исходный пакет заявляет свои зависимости сборки в поле Build-Depends файла debian/control. Давайте обратимся к apt для того, чтобы установить их (при условии, что вы находитесь в директории, содержащей распакованный исходный пакет):

```
$ sudo apt build-dep ./
Note, using directory './' to get the build dependencies
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following NEW packages will be installed:
  autoconf automake autopoint autotools-dev debhelper dh-autoreconf
  dh-strip-nondeterminism gettext intltool-debian libarchive-zip-perl
  libfile-stripnondeterminism-perl libtool po-debconf
0 upgraded, 13 newly installed, 0 to remove and 0 not upgraded.
Need to get 4 456 kB of archives.
After this operation, 14,6 MB of additional disk space will be used.
Do you want to continue? [Y/n]
[…]
```

В этом примере все зависимости сборки могут быть удовлетворены пакетами, доступными для APT. Это может быть не всегда так, поскольку инструмент kali-roll не обеспечивает возможность установки зависимостей сборки (учитываются только зависимости двоичных пакетов). На практике бинарные зависимости и зависимости сборки часто тесно связаны, и у большинства пакетов зависимости сборки будут удовлетворены.

9.1.3 Производим изменения

Мы не можем затронуть в данном разделе все возможные изменения, которые вы захотите произвести с пакетом. Здесь мы

постараемся разъяснить вам все необходимые детали [35] пакетирования Debian. Однако, мы затронем три самых распространенных случая использования, которые мы уже упоминали ранее, а также объясним некоторые из самых важных частей (вроде поддержки файла changelog).

Первая вещь, которую необходимо сделать, это изменить версию пакета таким образом, чтобы измененный пакет мог отличаться от оригинального пакета предоставленного Kali или Debian. Чтобы сделать это, мы обычно добавляем суффикс идентифицирующий сущность-объект (человека или компанию), применяющую изменения. Учитывая тот факт, что buxy является моим IRC ником, я буду использовать его в качестве суффикса. Такое изменение лучше всего выполнять с помощью команды dch *(Debian CHangelog)* из пакета *devscripts*, которая будет выглядеть следующим образом dch --local buxy. Это вызовет текстовый редактор (sensible-editor, который запускает редактор, назначенный в переменных сред VISUAL или EDITOR, или /usr/bin/editor), который позволит вам документировать различия, представленные в изменении. Этот редактор показывает, что dch действительно изменил файл debian/changelog:

```
$ head -n 1 debian/changelog
libfreefare (0.4.0-2) unstable; urgency=low
$ dch --local buxy
[...]
$ head debian/changelog
libfreefare (0.4.0-2buxy1) UNRELEASED; urgency=medium

  * Enable --with-debug configure option.

 -- Raphael Hertzog <buxy@kali.org>  Fri, 22 Apr 2016 10:36:00 -0400

libfreefare (0.4.0-2) unstable; urgency=low

  *  Update debian/copyrtight.
     Fix license to LGPL3+.
```

Если вы производите подобные изменения довольно часто, вам может понадобиться выставить значение переменных сред DEBFULLNAME и DEBEMAIL на ваше полное имя и ваш адрес электронной почты соответственно. Их значения будут

[35] https://www.debian.org/doc/manuals/maint-guide/

использоваться многими инструментами пакетирования, включая dch, которые будут вставлять их последнюю строчку, как показано выше (начиная с «-»).

Применение модификаций

В одном из наших случаев использования мы загрузили исходный пакет pyrit, и мы хотим применить модификации, которые мы обнаружили в соответственном git репозитории. Это довольно распространенная операция, и она всегда должна быть простой. К сожалению, модификации могут обрабатываться по-разному в зависимости от формата исходного пакета и рабочего процесса пакетирования Git (когда Git используется для поддержки пакета).

С нераспакованным исходным пакетом Вы выполнили apt source pyrit и у вас есть pyrit-0. 4.0 directory. Вы можете применить ваши изменения или модификации напрямую с помощью patch -p1 < patch-file:

```
$ apt source pyrit
[...]
$ cd pyrit-0.4.0
$ wget https://github.com/JPaulMora/Pyrit/commit/14
    ➥ ec997174b8e8fd20d22b6a97c57e19633f12a0.patch -O /tmp/pyrit-patch
[...]
$ patch -p1 </tmp/pyrit-patch
patching file cpyrit/pckttools.py
Hunk #1 succeeded at 53 (offset -1 lines).
$ dch --local buxy "Apply patch to work with scapy 2.3"
```

На этой стадии, вы вручную модифицируете исходный код и уже можете создавать двоичные пакеты вашей измененной версии (смотри раздел 9.1.4 «Запуск сборки» [стр. 230]). Но если вы попробуете создать обновленный исходный пакет, то процесс потерпит неудачу, выдавая ошибку «незапланированные изменения» (“unexpected upstream changes”). Это происходит из-за того, что pyrit (как и большинство исходных пакетов) использует исходный формат (смотри файл debian/source/format), также известный как 3.0 (quilt), где изменения в соответствующем коде должны записываться в отдельные патчи (модификации),

хранящиеся в debian/patches/, и где файл debian/patches/series указывает порядок, в котором должны применяться модификация. Вы можете зарегистрировать свои изменения в новом патче (модификации) путем запуска dpkg-source --commit:

```
$ dpkg-source --commit
dpkg-source: info: local changes detected, the modified files are:
 pyrit-0.4.0/cpyrit/pckttools.py
Enter the desired patch name: fix-for-scapy-2.3.patch
dpkg-source: info: local changes have been recorded in a new patch: pyrit-0.4.0/debian/
    ➥ patches/fix-for-scapy-2.3.patch
$ tail -n 1 debian/patches/series
fix-for-scapy-2.3.patch
```

Серия патчей (модификаций) Quilt

Это условное обозначение в сфере управления патчами получило распространение благодаря инструменту под названием quilt и таким образом формат исходного пакета “3.0 (quilt)” является совместимым с этим инструментом – с небольшим отклонением, которое оно использует, а именно debian/patches вместо патчей. Этот инструмент доступен в пакете с тем же именем, и вы можете найти хорошее руководство здесь: https://raphaelhertzog.com/2012/08/08/how- to - use-quilt-to-manage-patches-in-debian- packages/

Если исходный пакет использует 1.0 или 3.0 (родной) исходный формат, тогда нет никакой необходимости регистрировать ваши изменения в отдельный патч. Они автоматически собраны в итоговом исходном пакете.

С помощью Git Репозитория Если вы использовали Git для извлечения исходного пакета, ситуация еще сложнее. Существует несколько рабочих процессов Git и связанных с ними инструментов, и, очевидно, что не все пакеты Debian используют одни и те же рабочие процессы и инструменты. Различие, ранее разъясненное в отношении исходного формата, по-прежнему актуально, но вы также должны проверить, применяются ли предварительно патчи (модификации) в исходном дереве, или они сохраняются только в debian/patches (в этом случае они применяются во время создания).

Самым популярным инструментом является *git-buildpackage.* Это тот инструмент, который мы используем для управления всеми репозиториями на git.kali.org. Когда вы используете его, патчи не применяются предварительно в исходном дереве, а они хранятся в debian/patches. Вы можете вручную добавить патчи в этот каталог и перечислить их в debian/patches/series, но пользователи git-buildpackage имеют тенденцию использовать gbp pq для редактирования всей серии патчей как отдельной ветки, которую вы можете расширить или перестроить по своему вкусу. Проверьте gbp-pq(1) для того, чтобы узнать, как его вызвать.

git-dpm (с соответствующей командой с таким же именем) является другим инструментом пакетирования git, который вы можете использовать. Он записывает метаданные в debian/.git-dpm сохраняет патчи, применяемые в исходном дереве, путем слияния постоянно переустанавливаемой ветки, которая создается из содержимого debian/patches.

Настройка опций сборки

Обычно вам приходится настраивать опции сборки, когда вы хотите подключить некоторые дополнительные функции или поведение, которое не активировано в официальном пакете, или же когда вы хотите настроить параметры, которые выставлены во время создания через ./configure или через переменные, установленные в среде сборки.

В подобных случаях, изменения обычно ограничиваются debian/rules, которые управляют шагами в процессе создания пакета. В простейших случаях строки, касающиеся начальной конфигурации (./configure ...) или фактической сборки ($ (MAKE) ... или make ...), довольно легко обнаружить. Если эти команды явно не вызваны, они, вероятно, являются побочным эффектом другой явной команды, и в этом случае, пожалуйста, обратитесь к их документации, чтобы узнать больше о том, как изменить поведение по умолчанию. С пакетами, использующими dh, вам может потребоваться добавить переопределение для команд

dh_auto_configure или dh_auto_build (см. Их соответствующие страницы руководства для объяснения того, как это сделать).

Для того чтобы сделать наше разъяснение более конкретным, давайте применим его к одному из случаев, приведенных нами в качестве примера. Вы решили модифицировать libfreefare для того, чтобы передать параметр -enable-debug в сценарий ./configure, чтобы вы могли получить более подробный вывод из ваших инструментов связи с ближним полем (near field communication (NFC)) и создать лучший отчет об ошибках в отношении вашей non-recognized Mifare NFC карты.

Поскольку пакет использует dh для управления процессом сборки, вы добавляете (или в данном случае модифицируете) объект override_dh_auto_configure. Ниже приведена соответствующая часть из файла libfreefare debian/rules:

```
override_dh_auto_configure:
        dh_auto_configure -- --without-cutter --disable-silent-rules --enable-debug
```

Пакетирование обновленной версии

Давайте рассмотрим пример с точки зрения обсуждения пакетирования обновленной версии. Предположим, что вы являетесь уверенным пользователем SET, и вы обнаружили новую версию (7.4.5), которая еще не является доступной в Kali (в которой есть только версия 7.4.4). Вы хотите создать обновленный пакет и опробовать его в деле. Это является незначительным вмешательством, и вы не ожидаете, что обновление потребует каких-либо изменений на уровне пакетирования.

Для того чтобы обновить исходный пакет, вы извлекаете новый исходный tarball рядом с текущим исходным пакетом и копируете каталог debian из текущего исходного пакета в новый. Затем вы используете версию в debian/changelog.

```
$ apt source set
Reading package lists... Done
NOTICE: 'set' packaging is maintained in the 'Git' version control system at:
git://git.kali.org/packages/set.git
Please use:
git clone git://git.kali.org/packages/set.git
to retrieve the latest (possibly unreleased) updates to the package.
Need to get 42.3 MB of source archives.
[...]
dpkg-source: warning: failed to verify signature on ./set_7.4.4-0kali1.dsc
dpkg-source: info: extracting set in set-7.4.4
dpkg-source: info: unpacking set_7.4.4.orig.tar.gz
dpkg-source: info: unpacking set_7.4.4-0kali1.debian.tar.xz
dpkg-source: info: applying edit-config-file
dpkg-source: info: applying fix-path-interpreter.patch
$ wget https://github.com/trustedsec/social-engineer-toolkit/archive/7.4.5.tar.gz -O
    ➥ set_7.4.5.orig.tar.gz
[...]
$ tar xvf set_7.4.5.orig.tar.gz
[...]
social-engineer-toolkit-7.4.5/src/wireless/wifiattack.py
$ cp -a set-7.4.4/debian social-engineer-toolkit-7.4.5/debian
$ cd social-engineer-toolkit-7.4.5
$ dch -v 7.4.5-0buxy1 "New upstream release"
```

Вот и все. Теперь вы можете создавать обновленный пакет.
В зависимости от типа изменений, которые представлены в новой версии, вам также может потребоваться изменить зависимости сборки и зависимости необходимые для корректной работы, а также установить новые файлы. Это довольно сложные и обширные операции, которые, к сожалению, не затронуты в этой книге.

9.1.4 Запуск сборки

Когда все необходимые изменения были применены к источникам, вы можете начать генерировать реальный двоичный пакет или .deb файл. Весь процесс управляется командой dpkg-buitdpackage command и выглядит следующим образом:

```
$ dpkg-buildpackage -us -uc -b
dpkg-buildpackage: source package libfreefare
dpkg-buildpackage: source version 0.4.0-2buxy1
dpkg-buildpackage: source distribution UNRELEASED
dpkg-buildpackage: source changed by Raphael Hertzog <buxy@kali.org>
dpkg-buildpackage: host architecture amd64
[...]
   dh_builddeb
dpkg-deb: building package 'libfreefare0-dbgsym' in '../libfreefare0-dbgsym_0.4.0-2buxy1_amd64.deb'.
dpkg-deb: building package 'libfreefare0' in '../libfreefare0_0.4.0-2buxy1_amd64.deb'.
dpkg-deb: building package 'libfreefare-dev' in '../libfreefare-dev_0.4.0-2buxy1_amd64.deb'.
dpkg-deb: building package 'libfreefare-bin-dbgsym' in '../libfreefare-bin-dbgsym_0.4.0-2buxy1_amd64.deb'.
dpkg-deb: building package 'libfreefare-bin' in '../libfreefare-bin_0.4.0-2buxy1_amd64.deb'.
dpkg-deb: building package 'libfreefare-doc' in '../libfreefare-doc_0.4.0-2buxy1_all.deb'.
 dpkg-genchanges -b >../libfreefare_0.4.0-2buxy1_amd64.changes
dpkg-genchanges: binary-only upload (no source code included)
 dpkg-source --after-build libfreefare-0.4.0
dpkg-buildpackage: binary-only upload (no source included)
```

Параметры -us –uc отключают подписи на некоторых сгенерированных файлах (.dsc, .changes), потому что эта операция завершится неудачей, если у вас нет ключа GnuPG, связанного с идентификатором, который вы поместили в файл changelog. Параметр –b запрашивает «только двоичные сборки». В этом случае исходный пакет (.dsc) не будет создан, а вместо этого будет создан только двоичный пакет (.deb). Используйте эту опцию для того, чтобы избежать неудач во время создания исходного пакета: если вы не записали ваши изменения должным образом в системе управления внесения изменений (patch management system), она может жаловаться и прерывать процесс сборки.

Как было предложено сообщениями dpkg-deb, созданные двоичные пакеты теперь доступны в родительской директории (в той, которой находится директория исходного пакета). Вы можете установить их с помощью команд dpkg -i или apt install.

```
$ sudo apt install ../libfreefare0_0.4.0-2buxy1_amd64.deb \
  ../libfreefare-bin_0.4.0-2buxy1_amd64.deb
Reading package lists... Done
Building dependency tree
Reading state information... Done
Note, selecting 'libfreefare0' instead of '../libfreefare0_0.4.0-2buxy1_amd64.deb'
Note, selecting 'libfreefare-bin' instead of '../libfreefare-bin_0.4.0-2buxy1_amd64.deb'
The following packages will be upgraded:
  libfreefare-bin libfreefare0
2 upgraded, 0 newly installed, 0 to remove and 0 not upgraded.
Need to get 0 B/69,4 kB of archives.
After this operation, 2 048 B of additional disk space will be used.
[...]
```

Мы предпочитаем использовать apt install вместо dpkg –i, поскольку она очень тонко работает с отсутствующими зависимостями. Но не так давно, вам приходилось работать с dpkg, т.к. apt не могла работать с файлами .deb за пределами любого репозитория.

Упаковщик dpkg-buildpackage wrappers

В наше время разработчики Debian чаще всего используют программу более высокого уровня, такую как debuild;

Она, как обычно, запускает dpkg-buildpackage, но она также добавляет вызов программы, которая запускает множество проверок для подтверждения соответствия сгенерированного пакета политике Debian[36]. Этот скрипт также очищает среду, чтобы локальные переменные среды не загрязняли сборку пакета. Команда debuild является одним из инструментов в пакете *devscripts*, который имеет определенную согласованность и конфигурацию, чтобы облегчить задачу специалистам по поддержке.

9.2 Перекомпиляция ядра Linux

Ядра, предоставленные Kali, включают в себя максимально возможное количество функций, а также максимальное количество драйверов, чтобы охватить самый широкий спектр существующих аппаратных конфигураций. Вот почему некоторые пользователи предпочитают перекомпилировать ядро, чтобы включить только то, что им нужно. Для этого есть две причины. Во-первых, это способ оптимизации потребления памяти, поскольку весь код ядра, даже если он никогда не используется, занимает физическую память. Поскольку статически скомпилированные части ядра никогда не перемещаются в место подкачки, общее снижение производительности системы будет вызвано созданием драйверов и встроенных функций, которые никогда не используются. Во-вторых, сокращение количества драйверов и функций ядра снижает риск проблем с

[36]https://www.debian.org/doc/debian- policy/

безопасностью, поскольку выполняется только часть доступного кода ядра.

Это важно знать

Если вы решите скомпилировать собственное ядро, вы должны принять следующие условия: Kali не может обеспечить обновления безопасности для вашего пользовательского ядра, т.е. ядра, которое вы скомпилировали под свои нужды. Сохраняя ядро, предоставленное Kali, вы получаете преимущества от обновлений, подготовленных проектом Debian.

Перекомпиляция ядра, также необходима, если вы хотите использовать определенные свойства, которые только доступны в качестве патчей (которые не включены в стандартную версию ядра).

Справочник ядра Debian (Debian Kernel Handbook)

Команда Debian поддерживает справочник ядра Debian (*Debian Kernel Handbook* (также доступный в пакете *debian-kernel-handbook*)) с обширной документацией о большинстве связанных с ядром заданиях и о том, каким образом обрабатываются официальные пакеты ядра Debian. Это то место, к которому вам следует обращаться в первую очередь, если вам нужно получить больше информации, чем было предоставлено в этом разделе.
http://kernet-handbook.atioth.debian.org

9.2.1 Введение и необходимые знания

Неудивительно, что Debian и Kali управляют ядром в форме пакета, который традиционно не компилируется и не устанавливается. Поскольку ядро остается под контролем системы пакетирования, оно может затем быть чисто удалено или развернуто на нескольких машинах. Кроме того, скрипты, связанные с этими пакетами, автоматизируют взаимодействие с начальным загрузчиком и initrd генератором.

В исходных Linux-источниках содержится все необходимое для создания пакета ядра Debian, но вам все равно нужно установить пакет ***build-essential***, чтобы убедиться, что у вас есть инструменты, необходимые для сборки пакета Debian. Кроме того, этап конфигурации ядра требует пакет *libncurses5-dev*. И наконец, пакет *fakeroot* позволит создать пакет Debian без каких-либо административных привилегий.

```
# apt install build-essential libncurses5-dev fakeroot
```

9.2.2 Получение источников

Поскольку исходные файлы ядра Linux доступны в виде пакета, вы можете извлечь их с помощью установки *linux-source-version* package. Команда apt-cache search ~linux-source должна показать последнюю версию ядра пакетированную Kali. Обратите внимание, что исходный код, который содержится в этих пакетах не соответствует тому, который был опубликован Линусом Торвальдсом (Linus Torvalds) и разработчиками ядра[37]; как и все дистрибутивы Debian и Kali применяет определенное количество исправлений, которые могут (или не могут) найти свой путь в обновленную версию Linux. Эти изменения включают в себя резервные копии исправлений/функций/драйверов из более новых версий ядра, новые функции, которые еще не были полностью объединены в соответствующее дерево Linux, а иногда даже особые изменения Debian или Kali.

Оставшаяся часть данного раздела будет сосредоточена на версии ядра Linux 4.9, но данный пример, безусловно, может быть адаптирован к конкретной версии ядра, которая вас интересует.

В данном примере, мы предположим, что двоичный пакет *linux-source-4.9* был установлен. Обратите внимание, что мы устанавливаем двоичный пакет, который содержит исходные источники, но не извлекает исходный пакет Kali под названием *linux*.

[37] https://kernel.org/

```
# apt install linux-source-4.9
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following additional packages will be installed:
  bc libreadline7
Suggested packages:
  libncurses-dev | ncurses-dev libqt4-dev
The following NEW packages will be installed:
  bc libreadline7 linux-source-4.9
0 upgraded, 3 newly installed, 0 to remove and 0 not upgraded.
Need to get 95.4 MB of archives.
After this operation, 95.8 MB of additional disk space will be used.
Do you want to continue? [Y/n] y
[...]
# ls /usr/src
linux-config-4.9  linux-patch-4.9-rt.patch.xz  linux-source-4.9.tar.xz
```

Обратите внимание, что пакет содержит /usr/src/linux- source-4.9.tar.xz, сжатый архив источников ядра. Вы должны извлечь эти файлы в новую директорию (не напрямую в / usr / src /, так как нет необходимости в специальных разрешениях для компиляции ядра Linux). Вместо этой, более подходящей директорией будет - /kernel/

```
$ mkdir ~/kernel; cd ~/kernel
$ tar -xaf /usr/src/linux-source-4.9.tar.xz
```

9.2.3 Настройка ядра

Следующий шаг состоит в настройке ядра в соответствии с вашими потребностями. Точная процедура зависит от целей, которые вы преследуете.

Сборка ядра зависит от файла конфигурации ядра. В большинстве случаев вы, скорее всего, будете максимально придерживаться предложенной Kali версии, которая, как и все дистрибутивы Linux, устанавливается в каталог /boot. В этом случае вместо того, чтобы настраивать все с нуля, достаточно сделать копию файла /boot/config- *version*. (Версия должна совпадать с версией ядра используемой на данный момент, которую можно найти с помощью команды uname -r.) Поместите копию в файл .config в каталог, содержащий источники ядра.

```
$ cp /boot/config-4.9.0-kali1-amd64 ~/kernel/linux-source-4.9/.config
```

В качестве альтернативы, поскольку ядро предоставляет конфигурации по умолчанию в arch/arch/configs /* _ defconfig, вы можете разместить выбранную конфигурацию с помощью команды make x86_64_defconfig (в случае 64-разрядного ПК) или make i386_defconfig (в случай 32-разрядного ПК).

Если вам не нужно изменять конфигурацию, вы можете остановиться здесь и сразу перейти к разделу 9.2.4 “Компиляция и создание пакета” [стр. 235]. Если же вам необходимо внести изменения или если вы решите перенастроить все с нуля, вы должны уделить достаточное количество времени на настройку своего ядра. В исходном каталоге ядра есть различные выделенные интерфейсы, которые можно использовать, вызывая команду make *target*, где *target* является одним из значений, описанных ниже.

make menuconfig компилирует и запускает текстовый режим интерфейса конфигурации ядра (здесь требуется пакет *libncurses5-dev*), который позволяет перемещать множество доступных параметров ядра в иерархическую структуру. Нажатие клавиши «пробел» изменяет значение выбранного параметра, а «Ввод» подтверждает правильность кнопки, выбранную в нижней части экрана; Select возвращает выбранное подменю; Выход (Exit) закрывает текущий экран и возвращается в иерархию; Справка (Help) будет отображать более подробную информацию о роли выбранного варианта. Клавиши со стрелками позволяют перемещаться по списку опций и кнопок. Чтобы выйти из программы настройки, выберите «Выход» из главного меню. Затем программа предлагает сохранить сделанные вами изменения; принимайте, если вы удовлетворены своими выборами и хотите сохранить их.

Другие интерфейсы обладают похожими функциями, но они работают в более современных графических интерфейсах, таких как make xconfig, который использует графический интерфейс Qt и make gconfig, который использует GTK +. Первый требует *libqt4-dev*, а последний зависит от *libglade2-dev* и *libgtk2.0-dev*.

Работа с устаревшими .config файлами

Когда вы предоставляете файл .config, который был сгенерирован с другой (обычно более старой) версией ядра, вам придется обновить его. Вы можете сделать это с помощью make oldconfig, который будет интерактивно задавать вам вопросы, соответствующие новым настройкам конфигурации. Если вы хотите использовать ответ по умолчанию для всех этих вопросов, вы можете использовать make olddefconfig. Если вы используете make oldnoconfig, то он будет принимать отрицательный ответ на все вопросы.

9.2.1. Компиляция и создание пакета

Очистка перед внесением изменений

Если вы уже скомпилировали ядро в директории и хотите перестроить все с нуля (например, из-за того, что вы существенно изменили конфигурации ядра), то вам необходимо будет запустить make clean для того, чтобы удалить скомпилированные файлы, make distclean удаляет даже больше сгенерированных файлов, включая .config файл, так что убедитесь, что вы предварительно сделали резервную копию.

Как только конфигурации ядра будут готовы, простая команда make deb-pkg сгенерирует около пяти пакетов Debian в стандартном формате .deb: *linux-image-version,* который содержит образ ядра и связанные с ним модули; *linux-headers-version,* который содержит файлы заголовков, необходимые для создания внешнего модуля; *linux-firmware-image-version,* который содержит файлы прошивки, необходимые для корректной работы некоторых драйверов (этот пакет может отсутствовать в том случае, если вы выполняете создание из источников предоставленных Debian или Kali); *linux-image-version-dbg,* который содержит символы отладки для образа ядра и его модулей; и *linux-libc-dev,* который содержит заголовки, относящиеся к некоторым библиотекам пользовательского пространства вроде библиотеки GNU’s C library (glibc).

Пункт *version* определяется связью с соответствующей версией (как это определено переменной VERSION, PATCHLEVEL, SUBLEVEL, и EXTRAVERSION в Makefile) конфигурационного параметра LOCALVERSION, и переменной среды LOCALVERSION.

Версия пакета повторно использует ту же строку версии с добавленной ревизией, которая регулярно увеличивается (и сохраняется в .version), за исключением случаев, когда вы переопределяете ее с помощью переменной среды KDEB_PKGVERSION.

```
$ make deb-pkg LOCALVERSION=-custom KDEB_PKGVERSION=$(make kernelversion)-1
[...]
$ ls ../*.deb
../linux-headers-4.9.0-kali1-custom_4.9.2-1_amd64.deb
../linux-image-4.9.0-kali1-custom_4.9.2-1_amd64.deb
../linux-image-4.9.0-kali1-custom-dbg_4.9.2-1_amd64.deb
../linux-libc-dev_4.9.2-1_amd64.deb
```

Для того чтобы, наконец, использовать созданное ядро, последним шагом, который необходимо проделать является установка требуемых пакетов с помощью dpkg -i *file.deb*. Вам потребуется пакет “linux-image”; также вам нужно будет установить пакет “linux-headers” если вам необходимо будет создать некоторые внешние модули, что обычно происходит, когда у вас установлены некоторые “*-dkms” пакеты (это легко проверить с помощью dpkg -l "*-dkms" | grep ~ii). В большинстве случаев вам не понадобятся никакие другие пакеты (До тех пор, пока вы четко не будете уверены в том, что они вам действительно понадобятся!).

9.3 Создание живого пользовательского Kali ISO образа

Kali Linux по умолчанию обладает обширной функциональностью и достаточно высокой степенью гибкости. Как только Kali будет установлен, вы сможете выполнять всевозможные удивительные вещи со всем вашим творчеством, терпением и практикой. Однако вы можете также настроить сборку Kali так, чтобы она содержала определенные файлы или пакеты (чтобы увеличить или уменьшить производительность и количество предоставляемых функций) и могла выполнять определенные функции

автоматически. Например, Kali ISO of Doom[38] и Kali Evil Wireless Access Point[39] являются прекрасными проектами, основанными на индивидуальной реализации Kali Linux. Давайте посмотрим на процесс создания пользовательского образа ISO Kali Linux.

Официальные образы ISO Kali построены с помощью live-build[40], который представляет собой набор сценариев, который позволяет полностью автоматизировать и настроить все аспекты создания образа ISO. *live-build* suite использует всю структуру каталогов в качестве входных данных для его конфигурации. Мы сохраняем эту конфигурацию и некоторые связанные с ней вспомогательные скрипты в репозитории live-build-config Git. Мы будем использовать этот репозиторий в качестве основы для создания индивидуальных пользовательских образов.

9.3.1 Установка необходимых компонентов

Первый шаг заключается в том, чтобы установить пакеты, которые необходимы для извлечения Git репозитория с конфигурацией Kali *live-build:*

```
# apt install curl git live-build
[...]
# git clone git://git.kali.org/live-build-config.git
[...]
# cd live-build-config
# ls
auto  build_all.sh  build.sh  kali-config  README
```

На этой стадии вы уже можете создать обновленный (но не модифицированный) образ Kali ISO просто путем выполнения ./build.sh --verbose. Процесс создания займет довольно много времени, т.к. он сначала скачает все пакеты для включения их в этот процесс. После завершения этого процесса, вы найдете новый образ ISO в директории images.

[38]https://www.offensive-security.com/kali-linux/kali-linux-iso-of-doom
[39]https://www.offensive-security.com/kali-linux/kali-linux-evil-wireless-access- point/
[40]http://debian-live.alioth.debian.org/live-build/

9.3.2 Создание живых образов с использованием различных сред рабочего стола

Оболочка buitd.sh live-build, которую мы предоставляем, является ответственной за настройку должным образом директории config, которую рассчитывает найти live-build. Она может создавать различные конфигурации в зависимости от ее значений указанных в —variant option (вариант опция).

Оболочка создает config директорию путем соединения файлов из kali-config/common и kali-config/variant-X, где *X* это имя варианта, заданного в параметре variant. Когда опция конкретно не задана, она использует default в качестве имени варианта

Директория kali-config содержит директории для самых распространенных сред рабочего стола.

- e17 для Enlightenment;
- gnome для GNOME;
- i3wm для соответствующего менеджера управления окнами;
- kde для KDE;
- lxde для LXDE;
- mate для среды рабочего стола Mate (Mate Desktop Environment);
- xfce для XFCE.

Вариант light является немного особенным; он основан на XFCE[41] и используется для создания официального “light” ISO образа, который содержит сокращенный набор приложений.

Вы можете с легкостью создать Kali живой образ, используя KDE в качестве среды рабочего стола с помощью одиночной команды:

```
# ./build.sh --variant kde --verbose
```

Эта концепция варианта (*variant*) допускает некоторые предопределенные настройки высокого уровня, но если вы

[41] https://www.xfce.org/

потратите время на ознакомление с руководством живой системы Debian (Debian Live System Manual[42]), вы обнаружите множество других способов для настройки образов, просто изменив содержание соответствующей подкатегории kali-config. В следующих разделах приведены некоторые примеры.

9.3.3 Изменение набора установленных пакетов

После запуска live-build устанавливает все пакеты, перечисленные в файлах package-lists/*.list.chroot. Конфигурация по умолчанию, которую мы предоставляем, включает в себя файл package-lists/kali.list.chroot, в котором перечисляется *kali-linux-full* (основной мета-пакет, который стягивает все пакеты Kali для включения). Вы можете закомментировать этот пакет и поместить другой мета-пакет по вашему выбору или включить точный набор других пакетов. Вы также можете комбинировать оба подхода, начиная с мета-пакета и добавляя дополнительные пакеты по вашему выбору.

С помощью package-lists, вы можете только включать пакеты, которые являются уже доступными на официальных репозиториях Kali. Но если у вас есть пользовательские пакеты, вы можете включить их в живой образ, поместив файлы .deb в директорию package.chroot (например, kali-config/config-gnome/packages. Chroot, если вы создаете вариант для GNOME).

Мета-пакеты - это пустые пакеты, единственная цель которых иметь много зависимостей от других пакетов. Они упрощают установку наборов пакетов, которые вы часто хотите установить вместе. Исходный пакет kali-meta создает все метапакеты, предоставляемые Kali Linux:

- kali-linux: базовая система (он извлекается всеми другими мета пакетами)
- kali-linux-full: инсталляция Kali Linux по умолчанию
- kali-linux-all: мета пакет всех мета пакетов и других пакетов (практически все, что предоставляет Kali так что имейте ввиду, что он действительно огромен!)

[42]http://debian-live.alioth.debian.org/live-manual/unstable/manual/html/live-manual.en.html

- kali-linux-sdr: инструменты Software Defined Radio (SDR)
- kali-linux-gpu: Инструменты с графическим процессором (инструменты, использующие вычислительную мощность, доступную на вашей графической карте)
- kali-linux-wireless: беспроводная оценка и инструменты анализы
- kali-linux-web: инструменты оценки веб приложений
- kali-linux-forensic: криминалистические инструменты (поиск данных о том, что произошло)
- kali-linux-voip: инструменты Voice Over IP
- kali-linux-pwtools: инструменты взлома пароля
- kali-linux-top10: десять самых популярных инструментов
- kali-linux-rfid: RFID инструменты

Вы можете использовать эти мета пакеты, когда вы создаете список пользовательских пакетов для live-build. Полный список доступных мета пакетов и инструментов, которые включены в них, можно найти, перейдя по соответствующей ссылке http://tools.kali. org/kali-metapackages

Debconf пресидинг установленных пакетов

Вы можете предоставить Debconf preseed файлы (смотри раздел 4.3.2, “Создание Preseed файла” [стр. 93] для получения разъяснений) в качестве preseed/*.cfg файлов. Они будут использованы для настройки пакетов установленных в живой файловой системе.

9.3.4 Использование различных хуков для настройки содержимого образа

live-build предлагает различные хуки, которые могут быть выполнены на различных стадиях процесса сборки. Хуки Chroot являются исполняемыми скриптами, которые вы устанавливаете как файлы hooks/live/*.chroot в своем дереве конфигурации и выполняются внутри chroot. Хотя chroot - это команда, которая позволяет временно изменить корневую директорию операционной системы на выбранную вами, она также используется расширением для обозначения каталога, в котором

размещается полное (альтернативное) дерево файловой системы. В этом случае с live-build, каталог chroot является каталогом, в котором готовится живая файловая система. Поскольку приложения, запущенные в chroot, не видны за пределами этого каталога, тоже самое происходит с хуками chroot: вы можете использовать и изменять всё доступное в этой среде chroot. Мы полагаемся на те хуки, которые используются для выполнения нескольких специальных настроек Kali (см. kali-config/common/ hooks/live/kali-hacks.chroot).

Бинарные хуки (hooks/live/*.binary) выполняются в контексте процесса сборки (они не могут быть вызваны с помощью chroot), а именно на стадии его завершения. Вы можете модифицировать содержимое сборки ISO образа, но не саму живую файловую систему, т.к. на этом этапе она была уже сгенерирована. Мы используем эту особенность в Kali для проведения некоторых изменений в конфигурации по умолчанию isolinux, которая была сгенерирована live-build. Например, ознакомьтесь с kali-config/common/hooks/live/persistence.binary, где мы добавляем пункты загрузочного меню, предназначенные для включения постоянного хранилища данных.

9.3.5 Добавление файлов в образ ISO или в файловую систему

Еще одним очень распространенным способом настройки является добавление файлов в живую файловую систему или в образ ISO.

Вы можете добавлять файлы в живую файловую систему, помещая их в ожидаемое местоположение в директорию includes.chroot config. Например, есть файл kali-config/common/ includes.chroot/usr/lib/live/config/0031-root-password, который оказывается по адресу /usr/lib/ live/config/0031- root-password в живой файловой системе.

Live-Boot хуки

Скрипты, установленные как /lib/live/config/XXXX-***name*** выполняются сценарием init пакета live-boot. Они

реконфигурируют многие аспекты системы, подходящие для живой системы. Вы можете добавить ваши собственные скрипты для настройки вашей живой системы во время её работы: в частности, они используются, например, для реализации пользовательских параметров загрузки.

Вы можете добавить файлы в ISO образ путем размещения их в ожидаемом месте конфигурационной директории includes.binary. Например, есть файл kali-config/common/includes.binary/isolinux/splash.png который переопределяет фоновое изображение, используемое загрузчиком Isolinux (который хранится в /isolinux/splash.png в файловой системе образа ISO).

9.4 Добавление постоянного хранилища данных в живой образ ISO с помощью USB накопителя (Необходима правка общего содержания)

9.4.1 Особенности постоянного хранилища информации: Разъяснение основных моментов (Необходима правка общего содержания)

Далее, мы обсудим шаги, которые необходимо проделать для добавления постоянного хранилища информации в Kali USB накопитель. Вся суть живой системы заключается в её эфемерности. Все данные, которые хранятся в живой системе и все изменения, произведенные в ней, теряются после перезагрузки.

Чтобы исправить это, вы можете использовать функцию live-boot, называемую постоянным хранилищем информации (persistence), которая активируется, когда в параметры загрузки включено ключевое слово persistence.Поскольку внесение изменений в загрузочное меню является довольно непростой задачей, Kali включает в себя два пункта меню по умолчанию, которые позволяют включить постоянное хранилище: Live USB Persistence и Live USB Encrypted Persistence, как показано на Рисунке 9.1, “Пункты Persistence в меню” [стр. 240].

Рисунок 9.1 *Пункты Persistence в меню*

Когда данное свойство будет активно, *live-boot* просканирует все разделы в поисках файловых систем, помеченных как persistence (что может быть изменено с помощью параметра загрузки persistence-label=v‚a/ue), и установщик создаст постоянное хранилище директорий, которые перечислены в файле persistence.conf, расположенном в этом разделе (одна директория указывается в одной строчке). Специальное параметр "/ union" позволит полное сохранение всех директорий с помощью *union mount,* специальный слой, который хранит лишь изменения, вносимые в данные базовой файловой системы. Данные директорий с постоянным хранилищем данных хранятся в файловой системе, которая содержит соответствующий файл persistence.conf.

9.4.2 Создание незашифрованного хранилища на USB накопителе

В данном разделе, мы предположим, что вы подготовили Kali Live USB накопитель, следуя инструкциям в разделе 2.1.4, "Копирование образа на DVD-ROM или USB накопитель" [стр. 19],

и что вы использовали USB ключ с достаточным количеством памяти (хотя бы 3 GB) для хранения ISO образа и данных директорий, которые попадут в постоянное хранилище. Мы также предполагаем, что USB накопитель воспринимается Linux как /dev/sdb и что он содержит два раздела, которые являются частью ISO образа по умолчанию (/dev/sdb1 and /dev/sdb2). Будьте очень осторожны, выполняя эту процедуру. Вы можете с легкостью уничтожить важные данные, если случайно выполните разделение на части не того диска.

Для добавления нового раздела, вы должны знать размер образа, который вы скопировали для того чтобы новый раздел начинался на том месте, где заканчивается живой образ. Далее, используйте команду parted для того, чтобы создать новый раздел. Команды, приведенные ниже, анализируют ISO образ под названием kati-tinux-2016.1-amd64.iso, который, как предполагается, также присутствует на USB накопителе:

```
# parted /dev/sdb print
Model: SanDisk Cruzer Edge (scsi)
Disk /dev/sdb: 32,0GB
Sector size (logical/physical): 512B/512B
Partition Table: msdos
Disk Flags:

Number  Start   End     Size    Type     File system  Flags
 1      32,8kB  2852MB  2852MB  primary               boot, hidden
 2      2852MB  2945MB  93,4MB  primary
# start=$(du --block-size=1MB kali-linux-2016.1-amd64.iso | awk '{print $1}')
# echo "Size of image is $start MB"
Size of image is 2946 MB
# parted -a optimal /dev/sdb mkpart primary "${start}MB" 100%
Information: You may need to update /etc/fstab.

# parted /dev/sdb print
Model: SanDisk Cruzer Edge (scsi)
Disk /dev/sdb: 32,0GB
Sector size (logical/physical): 512B/512B
Partition Table: msdos
Disk Flags:

Number  Start   End     Size    Type     File system  Flags
 1      32,8kB  2852MB  2852MB  primary               boot, hidden
 2      2852MB  2945MB  93,4MB  primary
 3      2946MB  32,0GB  29,1GB  primary
```

После того, как вы создали новый раздел /dev/sdb3, отформатируйте его в файловой системе ext4 и назначьте ему метку «persistence» с помощью команды mkfs.ext4 (и её опции -L для назначения метки). Затем раздел монтируется в /mnt директорию, и вы добавляете необходимый файл конфигурации persistence.conf. Как обычно, не стоит забывать о том, что необходимо быть очень внимательным во время форматирования диска, поскольку вы можете навсегда потерять ценную информацию, если вы отформатируете неверный диск или раздел.

```
# mkfs.ext4 -L persistence /dev/sdb3
mke2fs 1.43-WIP (15-Mar-2016)
Creating filesystem with 7096832 4k blocks and 1777664 inodes
Filesystem UUID: dede20c4-5239-479a-b115-96561ac857b6
Superblock backups stored on blocks:
        32768, 98304, 163840, 229376, 294912, 819200, 884736, 1605632, 2654208,
        4096000

Allocating group tables: done
Writing inode tables: done
Creating journal (32768 blocks): done
Writing superblocks and filesystem accounting information: done
# mount /dev/sdb3 /mnt
# echo "/ union" >/mnt/persistence.conf
# ls -l /mnt
total 20
drwx------ 2 root root 16384 May 10 13:31 lost+found
-rw-r--r-- 1 root root     8 May 10 13:34 persistence.conf
# umount /mnt
```

USB накопитель теперь готов и вы можете загружаться с нее с использованием пункта меню загрузки “Live USB Persistence”.

9.4.3 Создание зашифрованного хранилища на USB накопителе (необходимы правки названия в общем содержании)

live-boot также может обрабатывать постоянные хранилища файловых систем на зашифрованных разделах. Таким образом, вы можете защитить данные путем создания зашифрованного раздела LUKS, в котором они и будут находиться.

Начальные шаги одинаковы вплоть до создания раздела, но вместо форматирования его в файловой системой ext4 используйте cryptsetup для инициализации его как контейнера LUKS. Затем откройте этот контейнер и настройте файловую систему ext4 так же, как и при создании нешифрованного хранилища, но вместо использования раздела /dev/sdb3 используйте виртуальный раздел, созданный cryptsetup. Этот виртуальный раздел представляет дешифрованное содержимое зашифрованного раздела, который доступен в /dev/mapper под именем, которое вы ему назначили ранее. В приведенном ниже примере мы будем использовать имя kali_persistence. Опять же, убедитесь, что вы используете правильный диск и раздел.

```
# cryptsetup --verbose --verify-passphrase luksFormat /dev/sdb3

WARNING!
========
This will overwrite data on /dev/sdb3 irrevocably.

Are you sure? (Type uppercase yes): YES
Enter passphrase:
Verify passphrase:
Command successful.
# cryptsetup luksOpen /dev/sdb3 kali_persistence
Enter passphrase for /dev/sdb3:
# mkfs.ext4 -L persistence /dev/mapper/kali_persistence
mke2fs 1.43-WIP (15-Mar-2016)
Creating filesystem with 7096320 4k blocks and 1774192 inodes
Filesystem UUID: 287892c1-00bb-43cb-b513-81cc9e6fa72b
Superblock backups stored on blocks:
        32768, 98304, 163840, 229376, 294912, 819200, 884736, 1605632, 2654208,
        4096000

Allocating group tables: done
Writing inode tables: done
Creating journal (32768 blocks): done
Writing superblocks and filesystem accounting information: done

# mount /dev/mapper/kali_persistence /mnt
# echo "/ union" >/mnt/persistence.conf
# umount /mnt
# cryptsetup luksClose /dev/mapper/kali_persistence
```

9.4.4 Использования нескольких постоянных хранилищ информации

Если вы используете вашу живую Kali систему в различных ситуациях, вы можете использовать несколько файловых систем с различными отметками и указывать в командной строке загрузки, какая файловая система должны быть использована в конкретном сеансе работы: это делается с помощью параметра загрузки persistence-label= label.

Давайте предположим, что вы профессиональный тестировщик на проникновение. Когда вы работаете с клиентом, вы используете постоянное хранилище на зашифрованном разделе для того, чтобы защитить конфиденциальность ваших данных, в случае если ваш USB накопитель будет украден или взломан. В то же самое время, возможно, вы захотите продемонстрировать Kali и какие-либо рекламные материалы, которые хранятся на незашифрованном разделе того же самого USB накопителя. Поскольку вы не захотите вручную редактировать параметры загрузки каждый раз, вам, скорее всего, захочется создать свой собственный живой образ со специальными пунктами загрузочного меню.

Первый шаг заключается в том, чтобы создать пользовательский live ISO (смотрите соответствующий раздел 9.3, “Создание живого пользовательского Kali ISO образа” [стр. 236] а также раздел 9.3.4, “Использование различных хуков для настройки содержимого образа ” [стр. 238]).

Самое главное, что вам необходимо сделать - это модифицировать kali-config/common/hooks/tive/ persistence-menu.binary для того, чтобы оно выглядело следующим образом (обратите внимание на параметры persistence-label):

```
#!/bin/sh

if [ ! -d isolinux ]; then
    cd binary
```

```
fi

cat >>isolinux/live.cfg <<END

label live-demo
    menu label ^Live USB with Demo Data
    linux /live/vmlinuz
    initrd /live/initrd.img
    append boot=live username=root hostname=kali persistence-label=demo persistence

label live-work
    menu label ^Live USB with Work Data
    linux /live/vmlinuz
    initrd /live/initrd.img
    append boot=live username=root hostname=kali persistence-label=work persistence-
        ➥ encryption=luks persistence

END
```

Затем мы создаем наш пользовательский ISO и скопируем его на USB-накопитель. Затем мы создаем и инициализируем два раздела и файловые системы, которые будут использоваться для организации постоянного хранилища данных. Первый раздел является незашифрованным (и помечен как «demo»), а второй будет зашифрованным (и помечен как “work”). Предполагая, что /dev/sdb это и есть наш USB-ключ, а размер нашего пользовательского ISO-образа - 3000 МБ, он будет выглядеть так:

```
# parted /dev/sdb mkpart primary 3000 MB 55%
# parted /dev/sdb mkpart primary 55% 100%
# mkfs.ext4 -L demo /dev/sdb3
[...]
# mount /dev/sdb3 /mnt
# echo "/ union" >/mnt/persistence.conf
# umount /mnt
# cryptsetup --verbose --verify-passphrase luksFormat /dev/sdb4
[...]
# cryptsetup luksOpen /dev/sdb4 kali_persistence
[...]
# mkfs.ext4 -L work /dev/mapper/kali_persistence
[...]
# mount /dev/mapper/kali_persistence /mnt
# echo "/ union" >/mnt/persistence.conf
# umount /mnt
# cryptsetup luksClose /dev/mapper/kali_persistence
```

И на этом все. Теперь вы можете загружаться с USB накопителя и использовать новые пункты в загрузочном меню, как вам будет удобно.

Добавление пароля самоуничтожения для получения дополнительной безопасности

Kali модифицировал cryptsetup и реализовал в нем новое свойство: вы можете установить *пароль самоуничтожения (nuke password)*, который в случае использования уничтожает все ключи, которые используются для управления зашифрованным разделом.

Это очень полезно в тех случаях, если вы много путешествуете, и вам нужен быстрый способ обеспечить невозможность доступа к вашим данным. Во время загрузки, просто введите пароль самоуничтожения вместо реального, и после этого никто больше (даже вы) не сможет получить доступ к вашим данным.

Перед использованием этой особенности, довольно умным шагом будет сделать резервную копию ваших ключей шифрования и хранить её в надежном месте.

Следуя примеру, приведенному в этой секции, вы можете добавить пароль самоуничтожения с помощью этой команды:

```
# cryptsetup luksAddNuke /dev/sdb4
Enter any existing passphrase:
Enter new passphrase for key slot:
Verify passphrase:
```

Больше информации об этой особенности вы сможете найти, в указанном ниже руководстве:

https://www.kali.org/tutorials/nuke- kali-linux-luks/

9.5 Подведем итоги

В этой главе мы узнали больше о том, как модифицировать исходные пакеты Kali, которые являются основными строительными блоками всех приложений, поставляемых в Kali. Мы также разъяснили, каким образом настроить и установить ядро Kali. Затем мы рассмотрели среду live-build и обсудили, как создать пользовательский образ ISO Kali Linux. Мы также продемонстрировали, как создавать как зашифрованные, так и незашифрованные установки USB Kali.

9.5.1 Основные рекомендации по модификации пакетов Kali

Изменение Kali пакетов обычно является задачей для сотрудников Kali и разработчиков, но у вас могут возникнуть некие особые нужды, которые не были удовлетворены официальными пакетами, и знание того, как создавать модифицированный пакет, может быть очень ценным, особенно, если вы хотите поделиться своими изменениями, использовать их вовне или же просто откатить все изменения к их предыдущему состоянию.

Когда вам нужно модифицировать какую-либо часть программного обеспечения, может возникнуть соблазн загрузить источник, внести изменения и использовать модифицированное программное обеспечение. Однако, если вашему приложению требуется общесистемная настройка (например, с помощью шага make install), то оно будет только *загрязнять* вашу файловую систему файлами, неизвестными dpkg, и вскоре создаст множество проблемы, которые не могут быть решены зависимостями пакетов. Кроме того, этот тип модификации программного обеспечения является более утомительным.

При создании модифицированного пакета общий процесс всегда один и тот же: возьмите исходный пакет, извлеките его, внесите изменения и затем создайте пакет. Для каждого шага часто используются несколько инструментов, которые могут отдельно обрабатывать каждую задачу.

Чтобы начать перестраивать пакет Kali, сначала скачайте исходный пакет, который состоит из файла *.dsc *(Debian Source*

Control) и дополнительных файлов, на которые ссылается этот файл управления.

Исходные пакеты хранятся на зеркалах, доступных через HTTP. Самый эффективный способ получить их - это использовать команду apt source *source-package-name,* которая требует от вас добавить строчку deb-src в файл /etc/ apt/sources.tist и обновить файлы индекса с помощью apt update.

Дополнительно вы можете использовать dget (из пакета *devscripts* package) для того, чтобы скачать .dsc файл напрямую вместе с его сопровождающими файлами.

Для специальных Kali пакетов, чей источник находится в репозитории Git на git.kali.org[43], вы можете получить источники с помощью git clone git://git.kali.org/ packages/ *source-package* (если вы ничего не видите в вашем репозитории, попытайтесь переключиться на ветку kali/ master с помощью git checkout kali/master).

После скачивания источников, установите пакеты, перечисленные в зависимостях сборки исходного пакета с помощью sudo apt build-dep ./. Эта команда должна быть запущена из исходной директории пакета.

Обновления исходного пакета состоят из комбинации следующих шагов:

- Требуемый первый шаг - изменить номер версии, чтобы отличить ваш пакет от оригинала с помощью dch --local *version-identifier,* или модифицировать другие детали пакета с помощью dch.
- Применение патча с помощью patch -p1 < *patch-file* или модификация quilt серии патча.
- Тонкая настройка опций сборки, обычно встречаются в файле пакета debian/rules или в других файлах директории debian/.

[43]http://git.kali.org

После модификации исходного пакета, вы можете создать бинарный пакет с помощью команды dpkg-buildpackage -us -uc –b из исходной директории, которая создаст неподписанный бинарный пакет. Затем пакет может быть установлен с помощью dpkg -i *package-name_version_arch.deb.*

9.5.2 Основные рекомендации по рекомпиляции ядра Linux

Как продвинутый пользователь, вы можете перекомпилировать ядро Kali. Вы можете захотеть уменьшить стандартное ядро Kali, которое загружено многими функциями и драйверами, добавить нестандартные драйверы или функции или применить различные патчи ядра. Однако обратите внимание: неправильно сконфигурированное ядро может дестабилизировать вашу систему, и вы должны быть готовы к тому, что Kali не сможет обеспечить обновления безопасности для вашего пользовательского ядра.

Для большинства модификаций ядра, вам понадобится установить несколько пакетов с помощью команды apt install build- essential libncurses5-dev fakeroot.

Команда apt-cache search ~linux-source должна перечислить последнюю версию ядра Kali, и apt install linux-source-version-number устанавливает сжатый архив источников ядра в /usr/src.

Исходные файлы должны быть извлечены с помощью tar –xaf отличный от /usr/src (например, - / kernel).
Когда пришло время настроить ваше ядро, всегда помните об этих моментах:

- Если вы не являетесь продвинутым пользователем, вы должны сначала заполнить файл конфигурации ядра. Предпочтительным способом является заимствование стандартной конфигурации Kali путем копирования/загрузки config-*version- string* to ~/kernel/linux-source-*version-number/.config.* Альтернативно, вы можете использовать make architecture_defconfig для получения подходящей конфигурации для данной архитектуры.

- Средство конфигурирования ядра menuconfig на текстовой основе прочитает файл .config и представит вам все элементы конфигурации в огромном меню, которое вы можете перемещать. Выбор элемента показывает его документацию, возможные значения и позволяет ввести новое значение.

Когда вы запускаете из исходного каталога ядра команду make clean, она удалит ранее скомпилированные файлы, а make deb-pkg сгенерирует до пяти пакетов Debian. Файл *linux-image-version* .deb содержит образ ядра и связанные с ним модули.

Чтобы на самом деле использовать сборку ядра, установите все необходимые пакеты с помощью dpkg -i *file.deb.* Требуется пакет “linux- image”; вам придется установить пакет “linux-headers” если у вас есть некоторые внешние модули ядра для сборки, что обычно необходимо, если у вас установлены пакеты “*-dkms” (checkwith dpkg -l "*-dkms" | grep ~ii). Остальные пакеты вам, как правило, не понадобятся (пока вы наверняка не будете уверены в том, что они вам действительно нужны!).

9.5.3 Основные рекомендации по созданию пользовательского живого ISO образа Kali

Официальный ISO образ Kali создается с помощью live-build[11], который является набором скриптов, позволяющих полностью автоматизировать и настраивать все аспекты создания образа ISO.

Ваша система Kali должна быть полностью обновлена, перед тем как использовать live-build.

Конфигурация Kali live-build может быть получена из Kali Git репозиториев с помощью двух команд: apt install curl git live-build а затем git clone git://git.kali.org/ live-build-config.git

Для создания обновленного, но не модифицированного ISO образа Kali, просто запустите ./build.sh --verbose. Процесс создания займет довольно много времени, пока он будет скачивать все пакеты. По завершению, вы найдете новый ISO

образ в директории, где хранятся образы. Если вы добавите --variant *variant* в командную строку, то будет создан данный вариант образа Kali ISO. Различные варианты определяются их конфигурационными каталогами kali-config/variant-*. Основным образом является вариант gnome.

Существует несколько способов настроить ваш ISO образ с помощью модификации конфигураций директории live-build:

- Пакеты могут быть добавлены к живому ISO (или удалены) путем модифицирования файлов package-lists/*. list.chroot
- Пользовательские пакеты могут быть добавлены в живой образ путем помещения файлов .deb в директорию packages.chroot
- Вы можете добавлять файлы в их файловую систему путем помещения их в логически ожидаемое место в конфигурационную директорию includes.chroot
- Вы можете выполнять скрипты во время chroot установочного процесса живой системы путем их установки в качестве файлов hooks/live/*.chroot. Вы также можете выполнять скрипты во время загрузки сгенерированного живого изображения: вы должны обеспечить их установку в usr/lib/live/config/XXXX-*name,* например, опираясь на каталог конфигурации include.chroot.
- Руководство Debian Live Systems Manual[12] является замечательной ссылкой для получения дополнительной информации о тестировании и настройке live-build.

Установка зашифрованного и незашифрованного постоянного хранилища информации на USB накопитель: всегда очень просто создать Kali Live USB устройство. Хотя процесс может казаться синтаксически сложным, довольно просто добавить как зашифрованную, так и незашифрованное постоянное хранилище к вашему переносному устройству, чтобы значительно расширить его функциональность.

В следующей главе, мы обсудим различные масштабы применения Kali на предприятиях. Мы обсудим управление конфигурацией и покажем вам, как расширить и настроить Kali Linux таким образом, чтобы его было легко установить, как на пару машин, так и на несколько тысяч.

Часть 10: Kali Linux в действии

Содержание:

Ключевые слова главы:

- PXE installation Управление конфигурацией
- Saltstack
- Разветвление пакетов Kali
- Пакеты конфигурации
- Репозиторий пакетов

До сих пор мы видели, что Kali является чрезвычайно надежной и безопасной операционной системой на основе Debian, обеспечивающей серьезный уровень безопасности и шифрования, расширенное управление пакетами, мультиплатформенные возможности и (что наиболее известно) арсенал инструментов мирового класса для профессионалов в сфере безопасности. Пока что для нас не совсем является очевидным тот факт, в каких именно масштабах может применяться Kali. В этом разделе мы покажем вам, каким образом можно использовать Kali для обеспечения централизованного управления в масштабах предприятия и предоставления контроля над множеством устройств, работающих с операционной системой Kali Linux. Вкратце, после прочтения этой главы вы сможете быстро развертывать высокозащищенные системы Kali, предварительно сконфигурированные для ваших конкретных потребностей, и синхронизировать их благодаря Kali установке (полуавтоматической) обновлений пакетов.

Этот уровень масштаба требует нескольких шагов, включая инициирование загрузки сети PXE, использование специального инструмента управления конфигурацией (SaltStack), возможность разветвления и настройки пакетов и развертывание их репозитория. Мы затронем каждый из этих вопросов более детально, покажем вам, каким образом с легкостью справиться с кажущимися на первый взгляд трудными моментами, а также как развертывать, управлять и поддерживать множество пользовательских устройств Kali Linux с относительной легкостью.

10.1 Установка Kali Linux по сети (PXE Boot)

Как мы видели в предыдущих главах, базовый процесс установки Kali Linux достаточно прост, если вы знаете, что именно вы хотите, и что вам нужно для этого делать. Но если вам нужно установить Kali на нескольких машинах, стандартная установка может быть довольно утомительной. К счастью, вы можете запустить процедуру установки Kali, загрузив компьютер через сеть. Это позволяет вам быстро и легко устанавливать Kali на многих машинах одновременно.

Для начала, вам нужно будет загрузить ваш целевой компьютер из сети. Этому способствует среда выполнения Preboot (Preboot execution Environment (PXE)), интерфейс клиента/сервера, предназначенный для загрузки любого сетевого устройства из сети, даже если на нем нет установленной операционной системы. Для настройки сетевой загрузки PXE требуется настроить, по крайней мере, trivial file transfer protocol (TFTP) и сервер DHCP/BOOTP. Вам также понадобится веб-сервер, если вы хотите разместить ***debconf*** preseeding файл, который будет автоматически использоваться в процессе установки.

К счастью, ***dnsmasq*** обрабатывает как *DHCP,* так и *TFTP,* так чтобы вы могли положиться на одну службу для настройки всего, что вам необходимо. Веб сервер Apache установлен (но не активен) по умолчанию на системах Kali.

Отдельные DHCP и TFTP демоны

Для более сложных установок, набор свойств *dnsmasq* может быть слишком ограниченным, или вы можете включить загрузку PXE в своей основной сети, которая уже запускает DHCP-демона.

В обоих случаях вам придется настроить отдельные DHCP и TFTP демоны.

Руководство по установке Debian затрагивает установку *isc-dhcp-server* и *tftpd-hpa* для PXE загрузки.

Для того чтобы настроить dnsmasq, вы должны сначала настроить его через/etc/dnsmasq.conf. Базовые конфигурации состоят всего лишь из нескольких ключевых строк:

```
# Network interface to handle
interface=eth0
# DHCP options
# IP range to allocate
dhcp-range=192.168.101.100,192.168.101.200,12h
# Gateway to announce to clients
dhcp-option=option:router,192.168.101.1
# DNS servers to announce to clients
dhcp-option=option:dns-server,8.8.8.8,8.8.4.4
# Boot file to announce to clients
dhcp-boot=pxelinux.0
# TFTP options
enable-tftp
# Directory hosting files to serve
tftp-root=/tftpboot/
```

После настройки/etc/dnsmasq.conf вам понадобится поместить установочные файлы загрузки в директорию /tftpboot/. Именно для этой цели Kali Linux предоставляет файловый архив, который может быть непосредственно распакован в директорию /tftpboot/. Просто сделайте выбор между 32-bit (i386) и 64-bit (amd64), а также между стандартным или графическим (gtk) методом установки для вашей целевой машины и выберите соответствующий архив:

- http://http.kali.org/dists/kali-rolling/main/installer-amd64/current/images/ netboot/gtk/netboot.tar.gz
- http://http.kali.org/dists/kali-rolling/main/installer-amd64/current/images/ netboot/netboot.tar.gz
- http://http.kali.org/dists/kali-rolling/main/installer-i386/current/images/ netboot/gtk/netboot.tar.gz
- http://http.kali.org/dists/kali-rolling/main/installer-i386/current/images/ netboot/netboot.tar.gz

Как только вы выбрали архив, создайте /tftpboot/, скачайте архив и распакуйте его в эту директорию:

```
# mkdir /tftpboot
# cd /tftpboot
# wget http://http.kali.org/dists/kali-rolling/main/installer-amd64/current/images/
    ➥ netboot/netboot.tar.gz
# tar xf netboot.tar.gz
# ls -l
total 25896
drwxrwxr-x 3 root root     4096 May  6 04:43 debian-installer
lrwxrwxrwx 1 root root       47 May  6 04:43 ldlinux.c32 -> debian-installer/amd64/boot
    ➥ -screens/ldlinux.c32
-rw-r--r-- 1 root root 26507247 May  6 04:43 netboot.tar.gz
lrwxrwxrwx 1 root root       33 May  6 04:43 pxelinux.0 -> debian-installer/amd64/
    ➥ pxelinux.0
lrwxrwxrwx 1 root root       35 May  6 04:43 pxelinux.cfg -> debian-installer/amd64/
    ➥ pxelinux.cfg
-rw-rw-r-- 1 root root       71 May  6 04:43 version.info
```

В распакованные файлы входит загрузчик *pxelinux*, который использует те же файлы конфигурации, что и *syslinux* и *isolinux.*. Благодаря этому, вы сможете настраивать файлы загрузки в debian- installer/amd64/ boot - screens/ как это было бы, если бы вы создавали пользовательские живые ISO образы Kali Linux.

Например, предположим, что вы выбрали текстовый режим установки, вы можете добавить параметры загрузки для того, чтобы подвергнуть процедуре пресидинга значения языка, страны, раскладки клавиатуры, имени хоста и имени домена. Вы также можете указать установщику на внешнюю preseed URL и настроить лимит времени таким образом, что загрузка начнется автоматически, если в течение 5 секунд не будет нажата какая-либо клавиша. Для того чтобы выполнить это, вам сначала нужно модифицировать файл debian - installer/amd64/txt.cfg:

```
label install
        menu label ^Install
        kernel debian-installer/amd64/linux
        append vga=788 initrd=debian-installer/amd64/initrd.gz --- quiet language=en
            ➥ country=US keymap=us hostname=kali domain= url=http://192.168.101.1/
            ➥ preseed.cfg
```

Затем, вы модифицируете файл debian - installer/amd64/syslinux.cfg для того, чтобы настроить лимит времени:

```
# D-I config version 2.0
# search path for the c32 support libraries (libcom32, libutil etc.)
path debian-installer/amd64/boot-screens/
include debian-installer/amd64/boot-screens/menu.cfg
default debian-installer/amd64/boot-screens/vesamenu.c32
prompt 0
timeout 50
```

Теперь вооружившись возможностью загрузить любую машину через сеть с помощью PXE, вы можете использовать в своих интересах все свойства и функции, описанные в разделе 4.3, “Автоматические Установки” [стр. 91], которые позволят вам проводить полную загрузку, процедуру пресидинга и автоматическую установку на множестве компьютеров без наличия физического загрузочного носителя. Также, не забывайте о гибкости параметра загрузки preseed/url=http://serv‘er/preseed. Cfg (но не использован url альтернативного имени), который позволяется вам установить preseed файл на основе сети.

10.2 Использование управление конфигурацией

Имея возможность довольно таки быстро устанавливать Kali на множество компьютеров, вам в дальнейшем понадобится помощь для управления этими машинами. Вы можете использовать инструменты управления конфигурацией для дальнейшего управления этими машинами.

Kali Linux содержит множество популярных инструментов управления конфигурацией, которые вы можете захотеть использовать ***ansible, chef, puppet, saltstack,*** и т. д.), но в этом разделе мы затронем лишь ***SaltStack.***
https://saltstack.com

10.2.1 Настройка SaltStack

SaltStack является централизованной службой управления конфигурацией: ***salt master*** управляет множеством ***salt minions.*** Вам следует установить пакет ***salt-master*** на сервер, который

является достижимым для всех хостов, которыми вы хотите управлять и установить ***salt-minion*** непосредственно на те хосты, которыми вы хотите управлять. Каждый миньон (minion) должен знать, где он может найти своего хозяина (master). Для этого просто отредактируйте /etc/salt/minion и выставьте master key для имени DNS (или IP адреса) Salt master. Обратите внимание, что Salt использует YAML в качестве формата для своего файла конфигурации.

```
minion# vim /etc/salt/minion
minion# grep ^master /etc/salt/minion
master: 192.168.122.105
```

Каждый миньон имеет уникальный идентификатор, хранящийся в файле / etc / salt / minion_id, который по умолчанию соответствует его имени хоста. Этот идентификатор миньона будет использоваться в правилах конфигурации и очень важно правильно установить его, прежде чем миньон откроет свое соединение с хозяином:

```
minion# echo kali-scratch >/etc/salt/minion_id
minion# systemctl enable salt-minion
minion# systemctl start salt-minion
```

Когда запущена ***salt-minion*** служба, она будет пробовать соединиться с Salt master для обмена некоторыми криптографическими ключами. Со стороны master, вам придется принять ключ, который использует minion для самоидентификации, чтобы продолжить соединение. Последующее соединение произойдет автоматически:

```
master# systemctl enable salt-master
master# systemctl start salt-master
master# salt-key --list all
Accepted Keys:
Denied Keys:
Unaccepted Keys:
kali-scratch
Rejected Keys:
master# salt-key --accept kali-scratch
The following keys are going to be accepted:
```

```
Unaccepted Keys:
kali-scratch
Proceed? [n/Y] y
Key for minion kali-scratch accepted.
```

10.2.2 Выполнение команд на миньонах (Minions)

Как только миньоны будут подключены, вы можете выполнять команды на них со стороны мастера:

```
master# salt '*' test.ping
kali-scratch:
    True
kali-master:
    True
```

Эта команда запрашивает все миньоны («*» - это групповой символ для всех миньонов) для выполнения функции ping из модуля выполнения теста. Эта функция возвращает значение True при успешном запуске и представляет собой простой способ убедиться, что соединение работает между мастером и различными миньонами.

Вы также можете настроить настроиться на определенный миньон, указав его идентификатор в первом параметре или, возможно, подгруппу миньонов, используя менее общий групповой символ (например, «* -scratch» или «kali- *»). Вот пример того, как выполнить произвольную команду оболочки на minion kali-scratch:

```
master# salt kali-scratch cmd.shell 'uptime; uname -a'
kali-scratch:
     05:25:48 up 44 min,  2 users,  load average: 0.00, 0.01, 0.05
    Linux kali-scratch 4.5.0-kali1-amd64 #1 SMP Debian 4.5.3-2kali1 (2016-05-09) x86_64
        ➥ GNU/Linux
```

Справочный материал касательно модуля Salt

Существует множество различных модулей выполнения для различных случаев употребления. К сожалению, мы не сможем затронуть их все, но их полный список доступен по ссылке https://docs.saltstack.com/en/latest/ref/modules/all/index.html. Вы

также можете получить описание всех модулей выполнения и их доступных функций на данном миньоне с помощью команды salt *minion* sys.doc. Запуск этой команды выведет довольно большой список функций, но вы можете отфильтровать данный список, указав имя функции или модуля с предстоящим родительским модулем в качестве параметра

```
master# salt kali-scratch sys.doc disk.usage
disk.usage:

    Return usage information for volumes mounted on this
        ➥ minion
```

Одним из наиболее полезных модулей является pkg, который представляет собой абстракцию диспетчера пакетов (package manager abstraction), полагающуюся на соответствующий менеджер пакетов для системы (apt-get для Debian и его производных, таких как Kali).

Команда pkg.refresh_db обновляет список пакетов (то есть выполняет apt-get update), а pkg.upgrade устанавливает все доступные обновления (он выполняет apt-get upgrade или apt-get dist-upgrade, в зависимости от полученных параметров). Команда pkg.tist_upgrades перечисляет операции, ожидающие обновления (которые будут выполняться командой pkg.upgrade dist_upgrade = True).

Сервисный модуль является абстракцией менеджера служб (systemd в случае с Kali), который позволяет выполнять все обычные операции systemctt: service.enabte, service.disabte, service. start, service.stop, service.restart и service.retoad:

```
master# salt '*' service.enable ssh
kali-scratch:
    True
kali-master:
    True
master# salt '*' service.start ssh
kali-master:
    True
kali-scratch:
    True
master# salt '*' pkg.refresh_db
kali-scratch:
    ----------
kali-master:
    ----------
master# salt '*' pkg.upgrade dist_upgrade=True
kali-scratch:
    ----------
    changes:
        ----------
        base-files:
            ----------
            new:
                1:2016.2.1
            old:
                1:2016.2.0
[...]
        zaproxy:
            ----------
            new:
                2.5.0-0kali1
            old:
                2.4.3-0kali3
    comment:
    result:
        True
```

В качестве более конкретного примера вы можете легко настроить сканирование Nmap с помощью dnmap. После установки пакета на всех миньонах вы запустите сервер в первом терминале:

```
server# salt '*' pkg.install dnmap
[...]
server# vim dnmap.txt
server# dnmap_server -f dnmap.txt
```

Предположив, что IP-сервера это 1.2.3.4, вы можете сообщить всем миньонам, чтобы они запустили клиентский процесс, который подключается к серверу:

```
server# salt '*' cmd.run_bg template=jinja 'dnmap_client -s 1.2.3.4 -a {{ grains.id }}'
kali-scratch:
    ----------
    pid:
        17137
[...]
```

Обратите внимание, что в примере используется cmd.run_bg для запуска команды dnmap_client в фоновом режиме. Не ждите, пока он закончится, так он является очень длительным процессом. К сожалению, он не завершает себя должным образом, когда вы прерываете работу сервера, поэтому вам придется его очистить:

```
server# salt '*' cmd.shell 'pkill -f dnmap_client'
```

10.2.3 Salt States и другие особенности

Хотя удаленное выполнение является довольно таки важным блоком, это лишь малая часть того, что может сделать SaltStack.

Каждый раз, настраивая новую машину, вы запускаете множество различных команд и тестов для определения деталей системы перед установкой.

Эти операции могут быть формализованы в повторно используемых шаблонах конфигурации, называемых *state files*. Затем операции, описанные в файлах state, могут быть выполнены с помощью одной команды state.apply salt.

Для того чтобы сэкономить некоторое время, вы можете полагаться на множество готовых к использованию state файлов, которые были созданы сообществом и которые распространяются в “Salt formulas”:

https://docs.saltstack.com/en/latest/topics/development/conventions/formulas.html

Также существует множество других особенностей, которые можно комбинировать:

- Запланированное выполнение действий;
- Определение действий в ответ на события запускаемые миньонами;
- Сбор данных с миньонов;
- Гармоничное сочетание последовательности операций в нескольких миньонах;
- Применение состояний через SSH без установки службы salt-minion;
- Системы предоставления облачных инфраструктур и управления ими;
- И многое другое.

SaltStack является довольно обширным, и мы, к сожалению, не можем затронуть все функции здесь. Фактически, есть книги, которые полностью посвящены SaltStack, а также онлайн-документация представляет собой огромный пласт информации. Перейдите по этой ссылке для того, чтобы ознакомиться с дополнительными источниками информации:

https://docs.saltstack.com/en/latest/

Если вы управляете значительным количеством машин, вам будет полезно узнать больше о SaltStack, так как вы можете сэкономить довольно много времени при подключении новых машин, а также вы сможете поддерживать согласованную конфигурацию в своей сети.

Чтобы дать вам представление о том, как выглядит работа со state файлами, мы рассмотрим простой пример: как активировать репозиторий APT и установить пакет, который вы создадите в разделе 10.3.3, «Создание репозитория пакетов для APT», [стр. 269] и в разделе 10.3.2 «Создание пакетов конфигурации» [стр. 263]. Вы также зарегистрируете SSH-ключ в учетной записи root, чтобы вы могли войти в систему удаленно в случае возникновения проблем.

По умолчанию, файлы state хранятся в /srv/salt на master; это файлы со структурой YAML и расширением .sls. Точно так же, как для запуска команд, применение state зависит от многих state модулей:

- https://docs.saltstack.com/en/latest/topics/tutorials/starting_states.html
- https://docs.saltstack.com/en/latest/ref/states/all/

Ваш /srv/salt/offsec.sls файл вызовет один из трех следующих модулей:

```
offsec_repository:
  pkgrepo.managed:
    - name: deb http://pkgrepo.offsec.com offsec-internal main
    - file: /etc/apt/sources.list.d/offsec.list
    - key_url: salt://offsec-apt-key.asc
    - require_in:
      - pkg: offsec-defaults

offsec-defaults:
  pkg.installed

ssh_key_for_root:
  ssh_auth.present:
    - user: root
    - name: ssh-rsa AAAAB3NzaC1yc2...89C4N rhertzog@kali
```

offsec_repository state опирается на pkgrepo state module. В примере используется управляемая функция в этом state модуле для регистрации репозитория пакетов. С помощью атрибута key_url, вы дадите знать salt (вооружившись ASCII), что GPG ключ, необходимый для проверки подписи репозитория, может быть извлечен из /srv/salt/offsec-apt-key.asc у salt master. Атрибут require_in гарантирует, что это что этот state будет обработан до offsec-defaults, так как последний должен правильно настроить репозиторий для установки пакета.

offsec-defaults state устанавливает пакет с таким же самым именем. Это показывает, что имя ключа очень часто является довольно важной величиной для state, хотя оно всегда может быть переопределено атрибутом name (как это было сделано для предыдущего state). Для простых случаев, вроде этого, оно и читаемое и довольно краткое.

Последний state (ssh_key_for_root) добавляет SSH ключ, заданный в атрибуте name в /root/.ssh/ authorized_keys (целевой пользователь указан в атрибуте user). Обратите внимание, что мы

сократили ключ для повышения читаемости здесь, но вам следует поместить полный ключ в атрибут name.

Этот state файл далее может быть применен к заданному миньону:

```
server# salt kali-scratch state.apply offsec
kali-scratch:
----------
          ID: offsec_repository
    Function: pkgrepo.managed
        Name: deb http://pkgrepo.offsec.com offsec-internal main
      Result: True
     Comment: Configured package repo 'deb http://pkgrepo.offsec.com offsec-internal
         ➥ main'
     Started: 06:00:15.767794
    Duration: 4707.35 ms
     Changes:
              ----------
              repo:
                  deb http://pkgrepo.offsec.com offsec-internal main
----------
          ID: offsec-defaults
    Function: pkg.installed
      Result: True
     Comment: The following packages were installed/updated: offsec-defaults
     Started: 06:00:21.325184
    Duration: 19246.041 ms
     Changes:
              ----------
              offsec-defaults:
                  ----------
                  new:
                      1.0
                  old:
----------
          ID: ssh_key_for_root
    Function: ssh_auth.present
```

```
        Name: ssh-rsa AAAAB3NzaC1yc2...89C4N rhertzog@kali
      Result: True
     Comment: The authorized host key AAAAB3NzaC1yc2...89C4N for user root was added
     Started: 06:00:40.582539
    Duration: 62.103 ms
     Changes:
              ----------
              AAAAB3NzaC1yc2...89C4N:
                  New

Summary for kali-scratch
------------
Succeeded: 3 (changed=3)
Failed:    0
------------
Total states run:     3
Total run time:  24.015 s
```

Он также может быть постоянно связан с миньоном, если его записать в файле /srv/salt/top.sls, который используется командой state.highstate для применения всех соответствующих states за один ввод:

```
server# cat /srv/salt/top.sls
base:
  kali-scratch:
    - offsec
server# salt kali-scratch state.highstate
kali-scratch:
----------
          ID: offsec_repository
    Function: pkgrepo.managed
        Name: deb http://pkgrepo.offsec.com offsec-internal main
      Result: True
     Comment: Package repo 'deb http://pkgrepo.offsec.com offsec-internal main' already
         ➥  configured
     Started: 06:06:20.650053
    Duration: 62.805 ms
     Changes:
----------
          ID: offsec-defaults
    Function: pkg.installed
      Result: True
     Comment: Package offsec-defaults is already installed
     Started: 06:06:21.436193
    Duration: 385.092 ms
     Changes:
----------
          ID: ssh_key_for_root
```

```
    Function: ssh_auth.present
        Name: ssh-rsa AAAAB3NzaC1yc2...89C4N rhertzog@kali
      Result: True
     Comment: The authorized host key AAAAB3NzaC1yc2...89C4N is already present for
         ➥ user root
     Started: 06:06:21.821811
    Duration: 1.936 ms
     Changes:

Summary for kali-scratch
------------
Succeeded: 3
Failed:    0
------------
Total states run:     3
Total run time: 449.833 ms
```

10.3 Расширение и настройка Kali Linux

Иногда вам необходимо изменить Kali Linux для того, чтобы он соответствовал вашим локальным нуждам. Лучший способ сделать это – поддерживать ваш собственный репозиторий пакетов, в котором размещены модифицированные версии Kali пакетов, которые вам придётся разветвить, также как и дополнительные пакеты, которые предоставляют настраиваемую конфигурацию и дополнительное программное обеспечение (не предосталяются Kali Linux).

10.3.1 Разветвление пакетов Kali

Пожалуйста, ознакомьтесь с разделом 9.1 «Модификация пакетов Kali» [стр. 222] для получения основной информации на эту тему.

Все пакеты могут быть разветвлены, если у вас есть на то веская причина, но вы должны понимать, что разветвление пакета имеет свою стоимость, поскольку вам придется обновлять его каждый раз, когда Kali опубликует обновление. Ниже приведены несколько причин, зачем вам может понадобиться разветвлять пакет:

- Для добавления патча, чтобы исправить неполадку или

добавить новую особенность. Хотя в большинстве случаев вам захочется передать этот патч соответствующим разработчикам для того, чтобы они исправили данную неполадку у себя или добавили новую особенность в источник.

- Для компиляции его с различными параметрами (при условии, что у вас есть веские причины, по которым Kali ранее не сделала этого их с этими параметрами, иначе было бы лучше обсудить это с разработчиками Kali, чтобы узнать, могут ли они активировать нужные параметры).
- В отличие от вышеперечисленных причин, ниже мы привели несколько не очень приятных моментов, которые могут вынудить вас прибегнуть к разветвлению пакета. Также, мы указали несколько рекомендации по поводу того, как разрешить эти проблемы:
- Для того чтобы модифицировать файл конфигурации. У вас также есть несколько более удобных вариантов сделать это, например, использовать управление конфигурацией для автоматической установки модифицированного файла конфигурации или установки пакета конфигурации, который поместит файл в директорию конфигурации (когда это возможно) или направит в другую сторону исходный файл конфигурации.
- Для обновления до последней новой версии. И снова хотим обратить ваше внимание на то, что лучше работать с разработчиками для обновления пакета напрямую в Debian или Kali. С моделью rolling release обновления намного быстрее достигнут конечных пользователей.

Среди всех доступных пакетов, существуют некоторые, которые являются основными строительными блоками Kali Linux, и в некоторых ситуациях их разветвление может быть довольно интересным:

- *kali-meta:* этот исходный пакет создает все мета пакеты kali-linux-* и в особенности *kali-linux- full*, который определяет, какие пакеты установлены в ISO образе Kali Linux по умолчанию.
- *desktop-base:* этот исходный пакет содержит множество разнообразных файлов, которые используются по умолчанию в устройствах рабочего стола. Рассмотрите возможность разветвления этого пакета, если вы хотите показать бренд

вашей организации в фоновом режиме по умолчанию или изменить тему рабочего стола.
- *kali-menu:* этот пакет определяет структуру Kali меню и предоставляет .desktop файлы для всех приложений, которые должны быть перечислены в Kali меню.

10.3.2 Создание пакетов конфигурации

Теперь, когда мы коснулись загрузки PXE и обсудили управление конфигурацией с помощью Salt-Stack, а также затронули вопрос о разветвлении пакетов, настало время перевести эти процессы в практический пример и расширить сценарий, создав собственный конфигурационный пакет для развертывания настраиваемой конфигурации на нескольких машинах полуавтоматически.

В этом примере вы создадите настраиваемый пакет, который устанавливает и использует ваш собственный репозиторий пакетов и ключ подписи GnuPG, распределяет конфигурацию SaltStack, предоставляет единым способом настройки рабочего стола по умолчанию для всех ваших устройств Kali.

Это может показаться сложной задачей (особенно, если вы заглянете в руководство Debian New Maintainer Guide[44]), но, к счастью, для нас пакет конфигурации - это в основном сложный файловый архив и превращение его в пакет довольно просто.

Изучение примера пакета

Если вы хотите изучить реальный пакет, который является в основном пакетом конфигурации, рассмотрите пакет *kali-defaults*. Он не так прост, как пример приведенный в этом разделе, но он имеет все соответствующие характеристики и даже использует некоторые передовые методы (например, dpkg-divert) для замены файлов, уже предоставленных другими пакетами.

Пакет *offsec-defaults* будет содержать несколько файлов:

- /etc/apt/sources.list.d/offsec.list: sources.list запись для APT,

[44]https://www.debian.org/doc/manuats/maint-guide/

которая делает доступным внутренний репозиторий пакетов компании.
- /etc/apt/trusted.gpg.d/offsec.gpg: ключ GnuPG, который используется для подписи внутреннего репозитория пакетов компании.
- /etc/salt/minion.d/offsec.conf: файл конфигурации SaltStack, который используется для того, чтобы определить, где найти Salt master.

10.3.3 Создание репозитория пакетов для APT

Теперь, когда у вас есть собственный пакет, вы можете распространять его через репозиторий пакетов APT. Используйте reprepro для создания желаемого репозитория и его наполнения. Этот инструмент довольно мощный, и его справочную страницу, безусловно, стоит прочитать.

Репозиторий пакетов обычно размещается на сервере. Чтобы правильно отделить его от других служб, запущенных на сервере, лучше всего создать пользователя, специально предназначенного для этой службы. В этой специальной учетной записи пользователя вы сможете размещать файлы репозитория, а также ключ GnuPG, который будет использоваться для подписи репозитория пакетов:

```
# apt install reprepro gnupg
[...]
# adduser --system --group pkgrepo
Adding system user ‘pkgrepo’ (UID 136) ...
Adding new group ‘pkgrepo’ (GID 142) ...
Adding new user ‘pkgrepo’ (UID 136) with group ‘pkgrepo’ ...
Creating home directory ‘/home/pkgrepo’ ...
# chown pkgrepo $(tty)
# su - -s /bin/bash pkgrepo
$ gpg --gen-key
gpg (GnuPG) 2.1.11; Copyright (C) 2016 Free Software Foundation, Inc.
This is free software: you are free to change and redistribute it.
There is NO WARRANTY, to the extent permitted by law.

gpg: directory ’/home/pkgrepo/.gnupg’ created
gpg: new configuration file ’/home/pkgrepo/.gnupg/dirmngr.conf’ created
gpg: new configuration file ’/home/pkgrepo/.gnupg/gpg.conf’ created
gpg: keybox ’/home/pkgrepo/.gnupg/pubring.kbx’ created
Note: Use ”gpg --full-gen-key” for a full featured key generation dialog.

GnuPG needs to construct a user ID to identify your key.

Real name: Offensive Security Repository Signing Key
Email address: repoadmin@offsec.com
You selected this USER-ID:
    ”Offensive Security Repository Signing Key <repoadmin@offsec.com>”

Change (N)ame, (E)mail, or (O)kay/(Q)uit? o
We need to generate a lot of random bytes. It is a good idea to perform
some other action (type on the keyboard, move the mouse, utilize the
disks) during the prime generation; this gives the random number
generator a better chance to gain enough entropy.
[...]
gpg: /home/pkgrepo/.gnupg/trustdb.gpg: trustdb created
gpg: key B4EF2D0D marked as ultimately trusted
gpg: directory ’/home/pkgrepo/.gnupg/openpgp-revocs.d’ created
gpg: revocation certificate stored as ’/home/pkgrepo/.gnupg/openpgp-revocs.d/
    ➥ F8FE22F74F1B714E38DA6181B27F74F7B4EF2D0D.rev’
public and secret key created and signed.

gpg: checking the trustdb
gpg: marginals needed: 3  completes needed: 1  trust model: PGP
gpg: depth: 0  valid:   1  signed:   0  trust: 0-, 0q, 0n, 0m, 0f, 1u
pub   rsa2048/B4EF2D0D 2016-06-17 [S]
      Key fingerprint = F8FE 22F7 4F1B 714E 38DA  6181 B27F 74F7 B4EF 2D0D
uid         [ultimate] Offensive Security Repository Signing Key <repoadmin@offsec.com>
sub   rsa2048/38035F38 2016-06-17 []
```

Обратите внимание, что когда вам будет предложено ввести идентификационную фразу, вы должны ввести пустое значение (и подтвердить, что вы не хотите защищать свой закрытый ключ),

поскольку вы хотите иметь возможность подписать репозиторий не интерактивно. Также обратите внимание, что gpg требует доступ для записи к терминалу, чтобы иметь возможность безопасно запрашивать идентификационную фразу: именно поэтому вы изменили права собственности на виртуальный терминал (который принадлежит root, так как вы первоначально были подключены как этот пользователь), прежде чем запускать оболочку как pkgrepo.

Теперь вы можете начать настройку репозитория. Для reprepro необходима специальная директория, и внутри этой директории вам необходимо создать файл conf/distributions, в котором описаны, какие дистрибутивы доступны в репозитории пакетов:

```
$ mkdir -p reprepro/conf
$ cd reprepro
$ cat >conf/distributions <<END
Codename: offsec-internal
AlsoAcceptFor: unstable
Origin: Offensive Security
Description: Offsec's Internal packages
Architectures: source amd64 i386
Components: main
SignWith: F8FE22F74F1B714E38DA6181B27F74F7B4EF2D0D
END
```

Обязательные поля: Codename, в котором указывается имя дистрибутива «Архитектура» (Architectures), которое указывает, какие архитектуры будут доступны в дистрибутиве (и утверждены во время ввода), и «Компоненты» (Components), что указывает на различные компоненты, доступные в дистрибутиве (компоненты своего рода подразделение дистрибутива, которое может быть включено отдельно в sources.list APT).

Поля «Источник» (Origin) и «Описание» (Description) являются чисто информативными, и они копируются как есть из файла Release. Поле SignWith просит reprepro подписать репозиторий с помощью ключа GnuPG, чей идентификатор включен в список (поместите здесь полную контрольную сумму файла, чтобы убедиться, что вы используете правильный ключ, а не другой с коротким идентификатором). Параметр AlsoAcceptFor не требуется, но позволяет обрабатывать файлы .changes, чье поле

Distribution имеет значение, указанное здесь (без этого оно будет принимать только кодовое имя дистрибутива в этом поле).

Используя эту базовую настройку, вы можете позволить reprepro сгенерировать пустой репозиторий:

```
$ reprepro export
Exporting indices...
$ find .
.
./db
./db/version
./db/references.db
./db/contents.cache.db
./db/checksums.db
./db/packages.db
./db/release.caches.db
./conf
./conf/distributions
./dists
./dists/offsec-internal
./dists/offsec-internal/Release.gpg
./dists/offsec-internal/Release
./dists/offsec-internal/main
./dists/offsec-internal/main/source
./dists/offsec-internal/main/source/Release
./dists/offsec-internal/main/source/Sources.gz
./dists/offsec-internal/main/binary-amd64
./dists/offsec-internal/main/binary-amd64/Packages
./dists/offsec-internal/main/binary-amd64/Release
./dists/offsec-internal/main/binary-amd64/Packages.gz
./dists/offsec-internal/main/binary-i386
./dists/offsec-internal/main/binary-i386/Packages
./dists/offsec-internal/main/binary-i386/Release
./dists/offsec-internal/main/binary-i386/Packages.gz
./dists/offsec-internal/InRelease
```

Как вы можете видеть, reprepro создал метаданные репозитория в подкаталоге dists. Он также инициализировал внутреннюю базу данных в подкаталоге db.

Настало время добавить ваш первый пакет. Сначала скопируйте файлы, сгенерированные сборкой пакета offsec-defaults (offsec-defaults_1.0.dsc, offsec-defaults_1.0.tar.xz, offsec-defaults_1.0_all.deb и offsec-defaults_1.0_amd64.changes) в /tmp на сервере, на котором размещен репозиторий пакетов, и скажите reprepro включить пакет:

```
$ reprepro include offsec-internal /tmp/offsec-defaults_1.0_amd64.changes
Exporting indices...
$ find pool
pool
pool/main
pool/main/o
pool/main/o/offsec-defaults
pool/main/o/offsec-defaults/offsec-defaults_1.0.dsc
pool/main/o/offsec-defaults/offsec-defaults_1.0.tar.xz
pool/main/o/offsec-defaults/offsec-defaults_1.0_all.deb
```

Как вы можете видеть, он добавил файлы в свой собственный пул пакетов в подкаталоге пула.

Каталоги dists и pool - это две директории, которые необходимо сделать (обще) доступными через HTTP, чтобы завершить настройку вашего репозитория APT. Они содержат все файлы, которые APT захочет загрузить.

Предполагая, что вы хотите разместить это на виртуальном хосте под названием pkgrepo.offsec.com, вы можете создать соответствующий файл конфигурации Apache, сохранить его в /etc/apache2/sites- available/pkgrepo. offsec.com.conf, и активировать его с помощью a2ensite pkgrepo.offsec.com):

```
<VirtualHost *:80>
    ServerName pkgrepo.offsec.com
    ServerAdmin repoadmin@offsec.com

    ErrorLog /var/log/apache2/pkgrepo.offsec.com-error.log
    CustomLog /var/log/apache2/pkgrepo.offsec.com-access.log "%h %l %u %t \"%r\" %>s %O"

    DocumentRoot /home/pkgrepo/reprepro

    <Directory "/home/pkgrepo/reprepro">
        Options Indexes FollowSymLinks MultiViews
        Require all granted
        AllowOverride All
    </Directory>
</VirtualHost>
```

И соответствующая запись sources.list для добавления на машинах, которые нуждаются в пакетах из этого репозитория, будет выглядеть так:

```
deb http://pkgrepo.offsec.com offsec-internal main

# Enable next line if you want access to source packages too
# deb-src http://pkgrepo.offsec.com offsec-internal main
```

Теперь ваш пакет опубликован и должен быть доступен для ваших сетевых хостов.

Хотя это была довольно таки длительная установка, «тяжелый труд» наконец завершен. Вы можете загружать свои сетевые машины через PXE, устанавливать индивидуальную версию Kali Linux без непосредственного взаимодействия благодаря предоставленному сетью preseed, настраивать SaltStack для управления вашими конфигурациями (и контролю над миньонами!), создавать разветвленные пользовательские пакеты и распространять эти пакеты через ваш собственный пакетный репозиторий. Это обеспечивает централизованное управление и контроль уровня предприятия на нескольких устройствах Kali Linux. Короче говоря, теперь вы можете быстро развертывать высокозащищенные системы Kali, предварительно сконфигурированные для ваших конкретных потребностей, и синхронизировать их благодаря (полуавтоматической) установке всех обновлений пакета Kali.

10.4 Подведем итоги

Kali Linux может использоваться в различных целях и на различных уровнях, начиная от одиночного пользователя и заканчивая уровнем предприятия. В этой главе мы затронули вопросы относительно того, каким образом централизовать управление множеством устройств Kali с помощью SaltStack, позволяя вам быстро разворачивать высокозащищенные системы Kali, предварительно настроенные под ваши конкретные нужды. Мы также показали, как вы можете синхронизировать их благодаря (полуавтоматической) установке обновлений пакетов Kali.

Мы обсудили разветвление пакетов, которое позволяет создавать собственные настраиваемые дистрибутивные исходные пакеты.

Вкратце, давайте рассмотрим основные шаги, необходимые для создания Salt мастеров и миньонов, которые позволяют осуществлять дистанционное управление и настройку удаленных хостов.

Основные моменты:

- Загрузите машину из сети с PXE, с TFTP файловым сервером, DHCP/BOOTP сервером (и веб-сервером с debconf preseeding). ***dnsmasq*** обрабатывает как DHCP и TFTP, а ***apache2*** веб-сервер установлен заранее (но заблокирован) на Kali.
- Руководство по установке Debian затрагивает установку ***isc-dhcp-server*** и ***tftpd-hpa*** для загрузки PXE:
- https://www.debian.org/releases/stable/amd64/ch04s05.html
- ***dnsmasq*** настроен через /etc/dnsmasq.conf. Базовая конфигурация состоит всего из нескольких ключевых строк:

```
# Network interface to handle
interface=eth0
# DHCP options
# IP range to allocate
dhcp-range=192.168.101.100,192.168.101.200,12h
# Gateway to announce to clients
dhcp-option=option:router,192.168.101.1
# DNS servers to announce to clients
dhcp-option=option:dns-server,8.8.8.8,8.8.4.4
# Boot file to announce to clients
dhcp-boot=pxelinux.0
# TFTP options
enable-tftp
# Directory hosting files to serve
tftp-root=/tftpboot/
```

- Распакуйте 32-разрядные (i386), 64-разрядные (amd64), стандартные или графические (gtk) установочные файлы загрузки из архива Kali в /tftpboot/. Архивы можно найти здесь:

http://http.kali.org/dists/kali-rolling/main/installer-amd64/current/images/netboot/gtk/netboot.tar.gz

http://http.kali.org/dists/kali-rolling/main/installer-amd64/current/images/netboot/netboot.tar.gz

http://http.kali.org/dists/kali-rolling/main/installer-i386/current/images/netboot/gtk/netboot.tar.gz

http://http.kali.org/dists/kali-rolling/main/installer-i386/current/images/netboot/netboot.tar.gz

```
# mkdir /tftpboot
# cd /tftpboot
# wget http://http.kali.org/dists/kali-rolling/main/installer-amd64/current/
    ➥ images/netboot/netboot.tar.gz
# tar xf netboot.tar.gz
```

- В случае необходимости измените txt.cfg чтобы подвергнуть пресидингу параметры или пользовательский лимит времени. Смотри раздел 4.3, “Автоматические установки” [стр. 91]. Затем вы можете использовать инструменты управления конфигурацией для управления машинами или настройки удаленных компьютеров до любого желаемого вам состоянии.
- SaltStack является централизованной службой управления конфигурацией: Salt master управляет множеством Salt миньонов. Установите пакет ***salt-master*** на доступный сервер и ***salt-minion*** управляемых компьютерах.
- Отредактируйте файл конфигурации /etc/salt/minion YAML-formatted и установите master ключ для DNS имени (или IP адреса) Salt мастера.
- Установите уникальный идентификатор миньонов в /etc/salt/minion_id:

```
minion# echo kali-scratch >/etc/salt/minion_id
minion# systemctl enable salt-minion
minion# systemctl start salt-minion
```

- Далее идет обмен ключами. master, принимает ключ аутентификации миньона. Последующее соединение пройдет автоматически.

```
master# systemctl enable salt-master
master# systemctl start salt-master
master# salt-key --list all
```

```
Accepted Keys:
Denied Keys:
Unaccepted Keys:
kali-scratch
Rejected Keys:
master# salt-key --accept kali-scratch
The following keys are going to be accepted:
Unaccepted Keys:
kali-scratch
Proceed? [n/Y] y
Key for minion kali-scratch accepted.
```

- Как только миньон подключен, вы можете выполнять команды на них с master компьютера. Примеры:

```
master# salt '*' test.ping
kali-scratch:
True
kali-master:
True
master# salt kali-scratch cmd.shell 'uptime; uname -a'
master# salt kali-scratch sys.doc'
master# salt '*' service.enable ssh
[...]
master# salt '*' service.start ssh
[...]
master# salt '*' pkg.refresh_db
[...]
master# salt '*' pkg.upgrade dist_upgrade=True
server# salt '*' cmd.shell 'pkill -f dnmap_client'
```

Часть 11: Введение в оценку безопасности

Содержание:

Ключевые слова главы:

- Типы оценок
- Оценка уязвимости
- Тестирование на проникновение на основе соответствия
- Традиционное тестирование на проникновение
- Оценка приложения
- Типы атак
- DOS атака
- Повреждение памяти Веб-уязвимости
- Атаки взлома пароля
- Атаки на клиента

Мы осветили достаточно особых свойств Kali Linux, поэтому у вас должно быть хорошее понимание того, что делает Kali Linux особенным, и как справиться с рядом сложных задач.

Существует несколько концепций относительно оценки безопасности, которые вам следует понимать перед началом использования Kali. В этой главе мы представим эти концепции и предоставим ссылки, которые смогут оказать помощь, если вам нужно использовать Kali для выполнения оценки безопасности. Для начала требуется какое-то время, чтобы четко понять, что означает «безопасность» при работе с информационными системами. При попытке обезопасить информационную систему вы концентрируетесь на трех первичных свойствах системы:

- *Конфиденциальность:* могут ли действующие лица, у которых нет доступа к системе или информации, получить доступ?
- *Целостность:* может ли система или данные быть изменены тем или иным способом, если это не было запланировано?
- *Доступность:* можно ли получить доступ к данным или системе привычным способом?

Все вместе эти три компонента образуют CIA триаду ((Confidentiality, Integrity, Availability) или КЦД триаду (*Конфиденциальность, Целостность, Доступность*)) и, по большей части, являются первичными пунктами, на которых вы сосредоточите свое внимание при оценке безопасности, как части стандартного развертывания системы, её поддержки или оценки.

Важно отметить, что в некоторых случаях вы можете быть больше обеспокоены одним компонентом триады CIA, чем другими. Например, если у вас есть личный журнал, содержащий ваши самые секретные мысли, то соответственно конфиденциальность журнала может быть более важна для вас, чем, к примеру, его целостность и доступность. Другими словами, вас не будет беспокоить, сможет ли кто-либо написать что-то в журнале, или будет ли он всегда являться доступным. С другой стороны, если вы защищаете систему, которая содержит и отслеживает медицинские предписания, то целостность данных будет очень важна. Т.к. очень важно, чтобы другие люди не смогли получить доступ к данным касательно того, какие медикаменты применяет тот или иной пациент, а также очень важно, чтобы вы были

уверены, что лично обладаете доступом к списку медикаментов; если же кто-то сможет изменить содержание системы (нарушая целостность), это может привести к результатам, угрожающим жизни человека.

Когда вы пытаетесь обезопасить систему, и у вас возник вопрос, вам следует определиться к какой из концепций или их сочетанию относится возникший вопрос. Это позволит понять проблему более широко, и отнести вопрос в соответствующую категорию, а также разобраться с ним. Также вы сможете идентифицировать уязвимости, которые влияют на один или несколько пунктов триады CIA. Давайте используем веб-приложение вместе с уязвимостью типа SQL инъекции в качестве примера:

- *Конфиденциальность:* Уязвимость типа «SQL инъекция» позволяет атакующей стороне извлечь все содержимое веб-приложения, что дает злоумышленнику возможность полного доступа к считыванию всех данных, но не дает возможность изменения информации или блокирования доступа к базе данных.
- *Целостность:* Уязвимость типа «SQL инъекция» позволяет атакующей стороне изменить существующую информацию в базе данных. Атакующая сторона не может считывать информацию или запрещать доступ другим пользователям к базе данных.
- *Доступность*: Уязвимость типа «SQL инъекция» инициирует длительный запрос, потребляющий большое количество ресурсов на сервере. Этот запрос, в том случае если он инициируется несколько раз, приводит к ситуации отказа в обслуживании (DoS). Атакующая сторона не имеет возможности доступа или изменения данных, но может ограничить доступ законным пользователям к веб-приложению.
- *Многократность*: Уязвимость типа «SQL инъекция» приводит к полному интерактивному доступу оболочки к операционной системе хоста, запускающего веб-приложение. Имея такой доступ, атакующая сторона сможет нарушить конфиденциальность системы, как им захочется, получив доступ к данным; скомпрометировать целостность системы, изменяя данные; но если они все же решатся на такое, вам следует удалить, которое может привести к компрометации доступности системы.

Концепции, включенные в триаду CIA, не являются очень сложными, и, скорее всего, вы постоянно работаете с ними интуитивно, даже если вы сами этого не осознаете. Однако, важно внимательно взаимодействовать с концепцией, т.к. она помогает вам распознать, куда будет лучше всего направить свои усилия. Такая концептуальная основа поможет вам определить критические компоненты системы и количество усилий и ресурсов, которые стоит затратить на исправление выявленных проблем.

Другая концепция, на которую мы будем ссылаться,- это ***риск,*** и то, каким именно образом он состоит из ***угроз*** и ***уязвимостей***. Эти концепции не очень сложные, но их легко понять неверным образом. Позже мы детально рассмотрим эти концепции. Но на более высоком уровне лучше думать о ***рисках***, которые вы пытаетесь предотвратить, ***угрозах*** – а именно, о том кто может их вам сделает, и ***уязвимостях*** - а именно, что может использовать их.

Например, посещая различные стороны света, вы можете подвергнуть себя ***риску*** заболеть малярией. Это потому, что ***угроза*** укуса комаров (москитов) очень высока в некоторых регионах, а вы не привиты от малярии. К счастью, вы можете контролировать ***уязвимость*** при помощи медикаментов и попытаться контролировать угрозу, используя средство, отпугивающее насекомых или москитные сетки. Контролируя одновременно и угрозу и уязвимость, вы можете гарантировать, что вы застрахованы от этого риска.

11.1 Kali Linux в оценке

При подготовке к работе Kali Linux вы должны удостовериться, что у вас чистая, рабочая версия. Довольно часто распространенная ошибка, которую делают профессионалы, это использование одной версия для проведения многочисленных оценок. Это проблема возникает по двум причинам:

- В ходе оценки вы вручную устанавливаете, настраиваете или,

в любом случае, изменяете свою систему. Эти единоразовые изменения помогут вам быстро приступить к работе или решить проблему, но их, тем не менее, довольно сложно отслеживать; это осложняет поддержку вашей системы и будущих конфигураций;
- Каждая оценка безопасности уникальна. Если вы не учитываете примечания, коды и другие изменения, это может привести к путанице или, еще хуже, к полному загрязнению данных клиента.

Вот почему настоятельно рекомендуется начинать с чистой версии Kali, а также именно поэтому специализированная версия Kali Linux, готовая к автоматической установке, довольно таки быстро окупается. При необходимости обратитесь к разделу 9.3 " Создание живого пользовательского ISO образа Kali" [стр. 236] и разделу 4.3, "Автоматические установки" [стр. 91], чтобы вспомнить, как это делать, т.к. чем больше вы сможете автоматизировать сегодня, тем меньше времени вы потратите завтра.

У каждого свои требования относительно того, как именно им нравится настроить Kali Linux для работы, но, тем не менее, есть универсальные рекомендации, которым мы бы, действительно, порекомендовали вам следовать. Во-первых, помните о зашифрованной установке, о которой говорилось в разделе 4.2.2 "Установка на полностью зашифрованную файловую систему» [стр. 85]. Это защитит ваши данные на реальной машине, которая будет являться для вас спасением в том случае, если ваш ноутбук будет украден.

Для дополнительной безопасности во время путешествия, возможно, вы захотите уничтожить ключ дешифрования (см. "Добавление пароля самоуничтожения для получения дополнительной безопасности" [стр. 245]) после того, как отправите своему сотруднику в офисе (зашифрованную) копию ключа. Таким образом, ваши данные будут находиться в безопасности, пока вы не вернетесь в офис, где вы можете восстановить свой ноутбук при помощи ключа дешифрования.

Другой пункт, на который вам следует дважды обратить внимание, это список пакетов, которые вы установили. Продумайте, какие

инструменты вам могут понадобиться для завершения работы. Например, если вы начинаете оценку беспроводной безопасности, вы можете учитывать установку метапакета ***kali-linux-wireless***, который содержит все инструменты беспроводной оценки, доступной в Kali Linux; или в случае необходимости оценки веб-приложения, вы можете установить все доступные инструменты тестирования веб-приложения в метапакете ***kali-linux-web***. Лучше предположить, что у вас не будет легкого доступа к интернету при проведении оценки безопасности, поэтому подготовьте все заранее.

По этой же причине вам, возможно, захочется просмотреть настройки сети (см. раздел 5.1, “Настройка сети” [стр. 104] и раздел 7.3, “Защита сетевых служб” [стр. 153]). Проверьте дважды настройки вашего DHCP и \службы, которые перечислены в вашем IP адресе. Эти настройки могут сильно повлиять на ваш успех. Вы не можете оценивать то, что не видите, а чрезмерные службы прослушивания ослабят вашу систему и закроют ее прежде, чем вы начнете работать.

Если ваша роль заключается в исследовании проникновения в сеть, то важнее будет обратить внимание на настройки сети, а также нужно избегать изменения системы, которая подвергается воздействию. Специализированная версия Kali с метапакетом ***kali-linux-forensic***, загруженная изначально в криминалистическом режиме, не будет автоматически устанавливаться на диск или использовать раздел диска. Таким образом, вы можете поддерживать анализ целостности системы во время использования многих криминалистических инструментов, доступных на Kali Linux.

Очень важно, чтобы вы тщательно подготовили версию Kali Linux к работе. Вы обнаружите, что чистая, эффективная среда Kali всегда будет делать все, чтобы в дальнейшем у вас не возникало проблем, и работа проходила максимально эффективно.

11.2 Типы оценки

Теперь, когда вы уверены, что ваша среда Kali готова, следующим шагом будет определение того, какой именно сорт оценки вы собираетесь проводить. На самом высоком уровне мы можем описать четыре типа оценок: ***оценка уязвимости***, ***проверка надежности***, ***традиционное тестирование на проникновение***, а также ***оценка приложения.*** Взаимодействие может включать в себя различные элементы каждого типа оценки, но, безусловно, их лучше подробно описать и объяснить их актуальность для вашей сборки и среды Kali Linux.

Прежде чем углубляться в рассмотрение различных типов оценок, очень важно сначала обратить ваше внимание на различие между уязвимостью и эксплойтом.

Уязвимостью мы называем недостаток либо изъян, который при его использовании может взломать или скомпрометировать конфиденциальность, целостность, а также доступность информационной системы. Существует множество различных типов уязвимостей, которые вам могут встречаться, включая:

- Вложение файла (File Inclusion): Уязвимость с вложением файла[1] в веб приложение позволяет вам ***включать*** содержимое локального или удаленного файла в вычислительные процессы программы. Например, веб-приложение может иметь функцию «Сообщение дня», которая считывает содержимое файла и включает его на веб-странице для отображения его пользователю. Если этот тип функции запрограммирован неправильно, он может позволить злоумышленнику изменить веб-запрос для того, чтобы заставить сайт включать в себя содержимое выбранного файла.
- SQL инъекция (SQL Injection): Атака с применением SQL инъекции[45] [46] является тем типом атаки, в котором процедуры проверки ввода для программы обойдены, что в свою очередь позволяет атакующему задавать SQL команды для целевой программы, которую она будет выполнять. Это является той формой исполнения команды, которая может привести к потенциальным проблемам безопасности.
- Переполнение буфера (Buffer Overflow): Переполнение

[45] https://en.wikipedia.org/wiki/File_inclusion_vu'lnerabi'lity
[46] https://en.wikipedia.org/wiki/SQL_injection

буфера[47] - это тип уязвимости, который обходит процедуры проверки ввода для записи данных в соседнюю память буфера. В некоторых случаях это смежное расположение памяти может иметь решающее значение для работы целевой программы, и управление выполнением кода может быть получено посредством тщательной манипуляции с перезаписанными данными памяти.
- Состояние гонки (Race Conditions): Состоянием гонки[48] - тип уязвимости, который использует хронометраж зависимостей в программе. В некоторых случаях рабочий процесс программы зависит от конкретной последовательности событий. Если вы можете изменить эту последовательность событий, это может привести к уязвимости.

С другой же стороны ***эксплойтом*** мы называем программное обеспечение, которое в случае применения, использует конкретную зависимость, хотя не все уязвимости подвержены действию эксплойта. В связи с тем, что эксплойт должен изменить запущенный процесс, заставляя его совершить незапланированные действие, создание эксплойта может быть довольно таки сложным. Кроме того, в современных вычислительных платформах существует ряд анти-эксплойтных технологий, которые были разработаны для того, чтобы затруднить использование уязвимостей, таких как предотвращение выполнения данных (ПВД) или Data Execution Prevention [49] (DEP) и рандомизация размещения адресного пространства (Address Space Layout Randomization [50] (ASLR)). Однако из-за того, что для конкретной уязвимости не существует общеизвестного эксплойта, это не означает, что его не существует (или его нельзя создать). Например, множество организаций занимаются продажей эксплойтов, которые никогда не публикуются, поэтому все уязвимости должны рассматриваться как потенциально доступные.

11.2.1 Оценка уязвимости

[47] https://en.wikipedia.org/wiki/Buffer_overflow
[48] https://en.wikipedia.org/wiki/Race_condition
[49] https://en.wikipedia.org/wiki/Executable_space_protection#Windows
[50] https://en.wikipedia.org/wiki/Address_space_layout_randomization

Уязвимостью является недостаток или изъян, который в той или иной степени может быть использован для того, чтобы скомпрометировать конфиденциальность, целостность или доступность информационной системы. В процессе оценки уязвимости вашей основной целью является создание простого списка обнаруженных уязвимостей в *целевой среде*. Данная концепция целевой среды является чрезвычайно важной. Вы должны быть уверены, что остаетесь в пределах целевой сети своего клиента и требуемых целей. Если вы выпадаете за пределы оценки, то это может к прерыванию обслуживания, нарушению доверия с вашим клиентом или судебным искам против вас и вашего работодателя.

Ввиду своей относительной простоты, тест на уязвимость часто завершается в более совершенных средах на регулярной основе как часть демонстрации их должной осмотрительности. В большинстве случаев, автоматизированный инструмент, такой как те, что содержаться в категориях анализа уязвимости (Vulnerability Analysis[51]) и веб приложения (Web Applications[52]) сайта Инструменты Кали (Kali Tools) и приложения меню рабочего стола Кали, используется для того, чтобы обнаружить живые системы в целевой среде, определить службы прослушивания и перечислить их с целью обнаружения максимального количества информации, такой как программное обеспечение сервера, его версия, платформа и т.д.

Затем эта информация проверяется на наличие известных сигнатур потенциальных проблем или уязвимостей. Эти сигнатуры состоят из комбинаций точек начала отсчета, которые предназначены для представления известных проблем. Используется довольно большое количество точек начала отсчета, т.к. чем более точек вы используете, тем более точной будет идентификация. Существует довольно большое количество потенциальных точек начала отсчета, которые включают в себя, но не ограничиваются ими:

- Версия операционной системы: Довольно распространенным явлением для программного обеспечения является то, что оно может быть уязвимым на одной версии операционной системы,

[51]http://tools.kali.org/category/vulnerability-analysis
[52]http://tools.kali.org/category/web-applications

а на другой – нет. Из-за этого сканер попытается как можно точнее определить, какая версия операционной системы размещает целевое приложение.

- Уровень патча: Довольно часто патчи для операционной системы не увеличивают информацию о версии, но все равно меняют способ реагирования на уязвимость или даже полностью устраняют эту уязвимость.
- Архитектура процессора: Многие программные приложения доступны для нескольких процессорных архитектур, таких как Intel x86, Intel x64, несколько версий ARM, UltraSPARC и т. д.
- В некоторых случаях уязвимость будет существовать только в определенной архитектуре, поэтому знание этого бита информации может иметь решающее значение для точной сигнатуры.
- Версия программного обеспечения: Версия целевого программного обеспечения практически всегда является одним из самых основных моментов, которые необходимо знать, для того, чтобы распознать уязвимость.
- Эти и многие другие исходные точки данных будут использованы для того, чтобы создать сигнатуру, как часть процесса сканирования уязвимостей. Вполне логично, что чем больше исходных точек данных совпадает, тем более точной будет сигнатура. При работе с сигнатурными совпадениями вы можете иметь несколько различных потенциальных результатов:
- Положительный результат: Сигнатура совпадает и захватывает реальную уязвимость. Вам необходимо следовать этим результатам и выполнить необходимые исправления в соответствии с ними, т.к. любой злоумышленник может использовать эти недостатки и изъяны для того, чтобы нанести вред вашей организации (или организации вашего клиента).
- Ложноположительный результат: Сигнатура совпадает; однако обнаруженная проблема не является реальной уязвимостью. Во время оценки, они могут создавать довольно много шума, что само по себе сильно раздражает. Вы никогда не должны отбрасывать положительный результат как ложноположительный без более детальной проверки.
- Отрицательный результат: Сигнатура не совпадает и соответственно уязвимостей нет. Этот сценарий является идеальным для того, чтобы проверить, что уязвимости не существует на цели.

- Ложноотрицательный результат: Сигнатура не совпадает, но, тем не менее, существует уязвимость. Не смотря на то, что ложноположительный результат является довольно плохим, ложноотрицательный результат намного хуже. В данном случае проблема существует, но сканер не может определить её, поэтому у вас не существует никаких указаний про её существование.

Как вы можете себе представить, точность сигнатур чрезвычайно важна для получения максимально точных результатов. Чем больше данных предоставлено, тем больше вероятность того, что вы получите точные результаты от автоматизированного сканирования на основе сигнатур, поэтому часто используются аутентифицированные сканирования.

При аутентифицированном сканировании, сканирующее программное обеспечение использует предоставленные учетные данные для аутентификации цели. Это предоставляет более глубокий уровень видимости в цели, чем каким либо другим способом. Например, при обычном сканировании вы можете обнаруживать информацию о системе, которая может быть получена из служб прослушивания и предоставляемых ими функций. Иногда это может быть совсем немного информации, но в данном случае обычное сканирование никак не может конкурировать в уровне и глубине данных, которые будут получены, если вы выполните аутентификацию в системе и всесторонне просмотрите все установленное программное обеспечение, примененные патчи, запущенные процессы и т. д. Подобная обширность полученных данных является довольно полезной для обнаружения уязвимостей, которые, в противном случае, могут остаться необнаруженными.

Хорошо проведенная оценка уязвимости представляет собой реальное отображение потенциальных проблем в организации и предоставляет показатели для измерения изменений с течением времени. Это довольно легкая оценка, но, тем не менее, множество организаций будут регулярно проводить автоматизированное сканирование уязвимостей в нерабочее время во избежание потенциальных проблем на протяжении дня, когда доступность служб и пропускная способность являются наиболее важными.

Как было упомянуто ранее, при сканировании уязвимостей необходимо проверить множество различных исходных точек данных для того, чтобы получить точные результаты. Все эти различные проверки могут создавать нагрузку на целевую систему, а также сокращать пропускную способность. К сожалению, очень сложно точно определить насколько много ресурсов цели будет потребляться, т.к. это зависит от количества открытых служб и типа проверок, которые будут связаны с этими службами. Таковой является цель проведения сканирования; в любом случае оно будет потреблять системные ресурсы. Иметь общее представление о ресурсах, которые будут потребляться, и насколько будет загружена целевая система, является очень важным при запуске этих инструментов.

Потоки сканирования

Большинство сканеров уязвимостей включают в себя опцию установки *потоков на сканирование*, что соответствует количеству одновременных проверок, которые происходят в данный момент. Увеличение этого числа будет иметь прямое влияние на нагрузку на платформу оценки, а также на сети и цели, с которыми вы взаимодействуете. Это очень важно иметь ввиду, когда вы используете эти сканеры. Всегда есть большой соблазн увеличить количество потоков, чтобы ускорить сканирование, но помните о значительном увеличении нагрузки, которая связана с этим.

Когда сканирование уязвимостей завершено, обнаруженные проблемы, как правило, привязываются к отраслевым стандартным идентификаторам, таким как номера CVE [53], EDB-ID[54] [55], информационным сообщениям о поставщике. Эта информация, вместе с оценкой CVSS уязвимости[11], в дальнейшем используется для определения уровня риска. Наряду с ложноположительными ложноотрицательными сообщениями об уязвимости, эти условные уровни риска являются общими проблемами, которые необходимо учитывать при анализе результатов сканирования.

[53] https://cve.mitre.org
[54] https://www.exploit-db.com/about/
[55] https://www.first.org/cvss

Поскольку автоматизированные инструменты используют базы данных сигнатур для определения уязвимостей, любое незначительное отклонение от известной сигнатуры может изменить результат, а также достоверность воспринимаемой уязвимости. Ложноположительный результат неверно отмечает уязвимость, которая не существует, в то время как ложноотрицательный результат довольно успешно следит за проблемой, но не сообщает об этом. Ввиду этого, сканер чаще всего является настолько хорошим, насколько хороша его база правил сигнатур. По этой причине, множество поставщиков предлагают несколько наборов сигнатур: один, который является бесплатным для домашнего использования, и другой набор, довольно дорогой, но, тем не менее, всесторонний, и обычно его приобретают корпорации и компании.

Другой проблемой, с которой часто встречаются во время сканирования уязвимостей, является достоверность предполагаемых уровней риска. Эти уровни риска определяются на общей основе, учитывая множество различных факторов, таких как уровень привилегий, тип программного обеспечения и до или после аутентификации. В зависимости от вашей среды, эти уровни могут быть или не быть приемлемы для вас, так что они не должны приниматься вами вслепую. Только те специалисты, которые хорошо осведомлены в различных системах и уязвимостях могут должным образом проверять уровни рисков.

Несмотря на то, что не существует общепринятого соглашения об уровнях рисков, рекомендуется использовать специальную публикацию NIST 800-30 (NIST Special publication 800-30[56]) в качестве основы для оценки уровней рисков и их точности в вашей среде. NIST SP 800-30 определяет реальный риск обнаруженной уязвимости, как *комбинацию вероятности возникновения и потенциального воздействия*.

Вероятность возникновения угрозы

Согласно Национальному институту стандартов и технологий (National Institute of Standards and Technology (NIST)), возможность возникновения угрозы базируется на вероятности того, насколько конкретная угроза способна использовать

[56]http://csrc.nist.gov/publications/PubsSPs.html#800-30

конкретную уязвимость с вероятными рейтингами – низкий, средний или высокий.

- Высокий: потенциальный злоумышленник является высококвалифицированным и мотивированным, а меры, которые были созданы для защиты от уязвимости, являются недостаточными.
- Средний: потенциальный злоумышленник является квалифицированным и мотивированным, но меры, которые были созданы для защиты от уязвимости, могут воспрепятствовать действиям злоумышленника.
- Низкий: потенциальный злоумышленник является неквалифицированным и/или слабо мотивированным, а те меры, которые существуют по защите от уязвимости, являются частично или полностью эффективными.

Воздействие

Уровень воздействия определяется оценкой количества вреда, который может быть нанесен в том случае, если уязвимость, о которой идет речь, была использована или эксплуатировалась иным образом.

- Высокий: использование уязвимости может привести к очень значительным финансовым потерям, серьезному вреду для выполнения задания или репутации организации или даже к серьезным травмам, включая потерю жизни.
- Средний: использование уязвимости может привести к финансовым потерям, нанесению ущерба миссии или репутации организации или травмам людей.
- Низкий: использование уязвимости может привести к некоторой степени финансовых потерь или воздействия на миссию и репутацию организации.

Общий риск

После того как вероятность возникновения, и степень воздействия были определены, вы можете определить уровень общего риска, который определяется в результате действия двух групп. Общий риск может оцениваться как низкий, средний или высокий, и

соответственно каждый из этих уровней представляет собой руководство к действию для лиц, ответственных за обеспечение безопасности и поддержание соответствующих систем.

- Высокий: Существует острая необходимость в принятии дополнительных мер для защиты от определенной уязвимости. В некоторых случаях системе может быть разрешено продолжить работу, но план действий должен быть разработан и реализован как можно скорее.
- Средний: Существует необходимость в принятии дополнительных мер для защиты от определенной уязвимости. План по реализации необходимых мер безопасности должен быть выполнен своевременно.
- Низкий: Владелец системы определит, стоит ли реализовывать дополнительные меры безопасности для защиты от определенной уязвимости или же риски являются допустимыми и систему можно оставить без изменений.

Подведем итоги

Учитывая столь большое количество факторов реальных рисков, которые исходят от обнаруженных уязвимостей, предопределение уровней риска из результатов работы соответствующего инструмента, должно быть использовано как отправная точка для определения реального риска для всей организации.

Грамотно созданные отчеты об оценке уязвимости, когда они анализируются профессионалом, могут служить основой для других оценок, таких как восприимчивость к тестированию на проникновение. Таким образом, важно понять, как получить наилучшие результаты из этой первоначальной оценки.

Kali является идеальной платформой для проведения оценки уязвимости и не требует никаких дополнительных конфигураций. В меню приложений Kali (Kali Applications menu), вы найдете множество различных инструментов для оценки уязвимостей в таких категориях как: сбор информации (Information Gathering), анализ уязвимостей (Vulnerability Analysis) и анализ веб приложений (Web Application Analysis). Несколько сайтов, включая ранее упомянутые Kali Linux Tools Listing[57], The Kali Linux

[57] http://tools.kali.org/tools-listing

Official Documentation [58] и курс free Metasploit Unleashed [59] предоставляют отличные ресурсы для использования Kali Linux во время проведения оценки безопасности.

11.2.2 Тестирование на проникновение на основе соответствия

Следующий тип оценки в порядке сложности - это тестирование на проникновение на основе соответствия. Это наиболее распространенные тесты на проникновение, поскольку они отвечают требованиям правительства и промышленности, а также они являются основанными на фреймворке соответствия, в которой работает вся организация.

Не смотря на то, что существует множество специализированных для различных отраслей фреймворков соответствия, самым распространенным скорее всего будет Payment Card Industry Data Security Standard [60] (PCI DSS), стандарт, который разработан компаниями платежных карт. Однако, также существует определенное количество других стандартов таких как Defense Information Systems Agency Security Technical Implementation Guides[61] (DISA STIG), Federal Risk and Authorization Management Program [62] (FedRAMP), Federal Information Security Management Act [63] (FISMA), и многие другие. В некоторых случаях корпоративный клиент может запросить оценку или же попросить посмотреть на результаты самой недавней оценки, ввиду различных причин. Вне зависимости от того является ли этот тип оценки узкоспециализированным или нет, в общем он называется тестированием на проникновение на основе соответствия или же просто «оценка соответствия» или «проверка соответствия».

Проверка соответствия обычно начинается с оценки уязвимости. В случае с проверкой соответствия PCI (PCI compliance auditing[64]), оценка уязвимости, в случае правильного проведения, может

[58] http://docs.kali.org
[59] https://www.offensive-security.com/metasptoit- unleashed/
[60] https://www.pcisecuritystandards.org/documents/Penetration_Testing_Guidance_March_2015.pdf
[61] http://iase.disa.mil/stigs/Pages/index.aspx
[62] https://www.fedramp.gov/about- us/about/
[63] http://csrc.nist.gov/groups/SMA/fisma/
[64] https://www.pcisecuritystandards.org/documents/PCIDSS_QRGv3_2.pdf

удовлетворить несколько базовых требований, включая «Не используйте предоставленные поставщиком значения по умолчанию для системных паролей и других параметров безопасности » (например, с инструментами из категории меню атака взлома паролей (Password Attacks)), «11.Регулярно проверяйте системы безопасности и процессы» (с инструментами из категории «Оценка базы данных (Database Assessment)) и другими. Некоторые требования, такие как «9. Ограничивайте физический доступ к данным владельцев кредитных карт» и «12. Поддерживайте политику, направленную на защиту информации для всего персонала» похоже, не поддаются традиционной оценке уязвимости с использованием инструментов и требуют дополнительной креативности и тестирования.

Несмотря на тот факт, что на первый взгляд использование Kali Linux может показаться нецелесообразным для некоторых элементов тестирования соответствия, Kali идеально подходит в этой среде, и это не только из-за широкого спектра инструментов, связанных с безопасностью, но, в первую очередь, из-за среды Debian с открытым исходным кодом, на которой и построен Kali, что позволяет устанавливать широкий спектр инструментов. Поиск менеджера пакетов с тщательно подобранными ключевыми словами, вне зависимости от того какой фреймворк соответствия вы используете, в большинстве случаев выдаст вам множество результатов. В данных условиях, многие организации используют Kali Linux в качестве стандартной платформы для этих точных видов оценок.

11.2.3 Традиционное тестирование на проникновение

Само понятие традиционное тестирование на проникновение стало довольно сложным для определения, которое в первую очередь зависит от пространства применения самого тестирования. Данная путаница вызвана тем фактом, что сам термин «Тестирование на проникновение» стал более часто использоваться для ранее упомянутого тестирования на проникновение на основе соответствия (или даже для оценки уязвимости), где, по сути, вы не слишком углубляетесь в саму

оценку, поскольку это выходит за рамки минимальных требований.

В целях освещения вопроса в этом разделе, мы отойдем в сторону от этой дискуссии для того, чтобы затронуть вопрос оценки, которая выходит за рамки минимальных требований; оценки, которая была разработана для того, чтобы действительно улучшить общую безопасность организации.
В отличие от ранее рассмотренного типа оценки, тестирование на проникновение обычно не начинается с определения области, но вместо этого определяется цель вроде, «имитировать, что произойдет, если внутренний пользователь будет скомпрометирован», или «определить, что произойдет, если организация подвергалась целенаправленной атаке со стороны внешней злонамеренной стороны». Ключевым дифференциатором этого сорта оценки является тот факт, что она не просто ищет подтвержденные уязвимости, но вместо этого использует найденные проблемы для того, чтобы определить самый худший сценарий развития событий. Вместо того чтобы опираться исключительно на набор инструментов для сканирования уязвимостей, вы должны следить за проверкой результатов с помощью эксплойтов или тестов для исключения ложноположительных результатов и сделать все от вас возможное для определения скрытых или ложнонегативных уязвимостей. Это часто связано с использованием обнаруженных ранее уязвимостей, изучения уровня доступа к эксплойтам и использования этого расширенного доступа в качестве рычага для дополнительных атак против цели.

Это требует критического пересмотра целевой среды наряду с ручным поиском, креативностью, внешним анализом и нестандартным мышлением, чтобы обнаружить другие возможности потенциальной уязвимости и, в конечном итоге, использовать другие инструменты и тесты кроме тех, которые были обнаружены сканерами уязвимостей. Как только это будет завершено, всегда необходимо начинать полный процесс снова и снова для того, чтобы окончательно выполнить работу.

Даже с таким подходом, вы все рано часто будете замечать, что множество оценок состоят из различных этапов. Kali облегчает поиск программ для каждого этапа с помощью меню Kali:

- Сбор информации (Information Gathering): На данном этапе, вы будете сосредоточены на изучении максимального количества информации о целевой среде. Как правило, это действие неинвазивное и довольно похоже на стандартную активность пользователя. Эти действия лягут в основу остальной части оценки и поэтому должны быть выполнены максимально полно. В категории «Сбор информации Кали» есть десятки инструментов, позволяющих выявить как можно больше информации об оцениваемой среде.
- Обнаружение уязвимостей (Vulnerability Discovery): Этот этап часто будет называться «активный сбор информации», где вы не атакуете, а занимаетесь нестандартным поведением пользователей, пытаясь определить потенциальные уязвимости в целевой среде. Именно на этом этапе будет происходить сканирование уязвимостей, обсуждаемое ранее. Те программы, которые были ранее перечислены в разделах Анализ уязвимостей, Анализ веб-приложений, Оценка базы данных и Обратная инженерия, будут очень полезны на этом этапе.
- Эксплуатация (Exploitation): Обнаружив потенциальные уязвимости, на этом этапе вы попытаетесь использовать их для закрепления в цели. Инструменты, которые смогут помочь вам на этом этапе, вы сможете найти в следующих разделах: Анализ веб-приложений, Оценка базы данных, Атака взлома пароля и Инструменты для эксплуатации.
- Переход в другие системы и фильтрация данных (Pivoting and Exfiltration): После того, как будет установлена первоначальная точка опоры, необходимо выполнить дальнейшие шаги. Они довольно часто повышают права доступа до уровня необходимого вам для достижения ваших целей, в качестве атакующего, например, перехода в другие системы, что, возможно, вам не было доступно ранее и фильтрация конфиденциальной информации из целевых систем. Для получения помощи на данном этапе обратитесь следующим разделам Атака взлома пароля, Инструменты эксплуатации, Сниффинг и спуффинг и Постэксплуатация.
- Составление отчета (Reporting): Как только активная часть оценки будет завершена, вы должны будете задокументировать и сообщить о проведенных мероприятиях. Этот этап часто бывает не таким техническим, как предыдущие этапы, однако очень важно, чтобы ваш клиент получил полную отдачу от выполненной работы. В разделе «Инструменты отчетности»

содержится ряд инструментов, которые окажутся полезными на данном этапе.

В большинстве случаев, эти оценки будут уникальными в своем составлении, т.к. каждая организация будет работать с различными угрозами, и будет защищать различные активы. Kali Linux создает довольно универсальную базу для этих видов оценок, и именно здесь вы можете действительно воспользоваться многими функциями индивидуальной настройки Kali Linux. Многие организации, которые проводят эти типы оценок, будут поддерживать индивидуальные версии Kali Linux для внутреннего использования, чтобы ускорить развертывание систем до новой оценки.

Индивидуальные настройки, которые организации смогут выполнять со своими инсталляциями Kali Linux, чаще всего будут включать в себя:

- Предварительная установка коммерческих пакетов с лицензией. Например, у вас может быть такой пакет, как коммерческий сканер уязвимостей, который вы хотели бы использовать. Чтобы не устанавливать этот пакет с каждой сборкой, вы можете сделать это один раз[65] и использовать его при каждом развертывании Kali, которое вы делаете.
- Предварительно настроенные и подключенные виртуальные частные сети (virtual private networks (VPN)). Они очень полезны для устройств, которые позволяют проводить «дистанционные внутренние» оценки. В большинстве случаев эти системы подключаются к системе, контролируемой оценщиком, создавая туннель, который оценщик может использовать для доступа к внутренним системам. The Kali Linux ISO of Doom[66] является примером именно этого типа настройки.
- Предварительно установленное, внутренне разработанное программное обеспечение и инструменты. Многие организации будут обладать частным набором инструментов, таким образом, их настройка все лишь единожды в пользовательской установке Kali[67] очень хорошо экономит время.
- Предварительно настроенные конфигурации операционной

[65] http://docs.kali.org/kali-dojo/02 - mastering-live-build
[66]https://www.offensive-security.com/kali-linux/kali- rolling-iso-of-doom/
[67]http://docs.kali.org/development/live- build-a-custom-kali-iso

системы, такие как отображение хостов, обои рабочего стола, настройки прокси-сервера и т.д. Многие пользователи Kali имеют особые настройки[68], которые они хотели бы настроить именно так. Если вы собираетесь регулярно переустанавливать Kali, сохранение и запоминание этих изменений имеет большой смысл.

11.2.4 Оценка приложения

В то время как большинство оценок обладают широким охватом, оценка приложения является специализацией, которая узко сконцентрирована на одном приложении. Этот тип оценки становится все более распространенными из-за сложности критически важных приложений, которые используются организациями, многие из которых являются внутрифирменными. Оценка приложения может быть добавлена к более широкой оценке, в случае необходимости. Приложения, которые могут быть оценены таким образом, включают в себя, но не ограничиваются этим списком:

- Веб-приложения: Веб-приложения довольно часто подвергаются атакам, хотя бы просто потому, что они представляют собой отличную цель ввиду своей доступности. Обычно, стандартная оценка найдет базовые проблемы в веб-приложении, однако, более детальное рассмотрение обычно стоит потраченного на него времени, для определения проблем, связанных с рабочим процессом приложения. Метапакет *kali-linux-web* обладает большим количеством инструментов, которые смогут помочь вам в этой оценке;
- Скомпилированные десктоп-приложения (Compiled desktop applications): Серверное программное обеспечение не всегда являются единственной целью; десктоп-приложения также представляют собой прекрасную цель для атаки. В прошедшие годы, множество десктоп-приложений, таких как PDF редактор или веб-видеопрограммы, довольно часто являлись целями атак. Однако, на данный момент существует не меньшее количество десктоп-приложений, которые при более детально изучении обладают большим количеством уязвимостей.

[68]https://www.offensive-security.com/kali-linux/kali-linux- recipes/

- Мобильные приложения: По мере того, как мобильные устройства становятся все более популярными, мобильные приложения становятся все более интересной целью для проведения атаки. Это довольно таки быстро движущаяся цель, а методология в этой области все еще созревает, что приводит к появлению новых разработок практически каждую неделю. Инструменты, связанные с анализом мобильных приложение, могут быть найдены в разделе меню Обратная разработка.

Оценка приложений может проводиться различными способами. В качестве простого примера для определения в приложении потенциальных проблем или угроз может быть применен специальный автоматизированный инструмент. Этот инструмент будет использовать специфическую для приложения логику в попытках определить неизвестную проблему, вместо того, чтобы полагаться в поиске на набор уже известных сигнатур. Этот инструмент должен обладать встроенным пониманием поведения приложения. Довольно распространенным примером этого будет сканер уязвимостей веб-приложений, такой как Burp Suite[69], направленный на работу с приложениями, которые сначала определяют различные поля ввода, а затем отправляет распространенные атаки SQL-инъекций в эти поля, отслеживая реакцию приложения на индикацию успешной атаки.

В более сложных сценариях, оценка приложения может быть проведена интерактивно как с помощью способа ***black box,*** так и с помощью способа ***white box.***

- Black Box Оценка: Инструмент (или оценщик) взаимодействует с приложением не имея специальных знаний или доступа, кроме доступа обычного пользователя. Например, в случае с веб-приложением, оценщик может иметь доступ лишь к функциям и свойствам, которые доступны пользователю, не вошедшему в систему. Любые учетные записи пользователя будут такими, в которых обычный пользователь может самостоятельно зарегистрировать аккаунт. Это не даст возможности атакующему видеть или просматривать какие-либо функции, которые доступны только пользователям, которые должны быть созданы администратором.

[69]https://portswigger.net/burp/

- White Box Оценка: Инструмент (или оценщик) чаще всего будет иметь полный доступ к исходному коду, доступ администратора к платформе, на которой работает приложение и т. д. Это гарантирует, что полный и всесторонний обзор всех функциональных возможностей приложения будет завершен независимо от того, где эта функциональность находится в приложении. Компромисс всего этого заключается в том, что оценка ни в коем случае не является симуляцией реальных вредоносных действий.

Безусловно, между этими способами оценок есть свои плюсы и минусы. Как правило, решающим фактором является цель оценки. Если целью является определение того, что будет происходить в случае, если данное приложение будет подвержено атаке, то black box оценка подойдет идеально в подобной ситуации. Если же целью является определить выявление и устранение как можно большего числа вопросов безопасности за относительно короткий период времени, то white box оценка будет более эффективной в данном случае.

В других случаях может возникнуть гибридный подход, когда оценщик не имеет полного доступа к исходному коду платформы, на которой запущено приложение, но учетные записи пользователей предоставляются администратором, чтобы обеспечить доступ к максимально возможной функциональности приложения.

Kali является идеальной платформой для всех типов оценки приложения. При установке по умолчанию доступно множество различных специальных сканеров приложений. Для более продвинутой оценки существует более широкий диапазон инструментов, исходных редакторов и сценариев сред. Следующие разделы, такие как Веб-приложение[70] и Обратная разработка[71], взятые с сайта Kali Tools[72], будут вам очень полезны.

[70]http://tools.kali.org/category/web-applications
[71]http://tools.kali.org/category/reverse-engineering
[72]http://tools.kali.org

11.3 Формализация оценки

Когда вы определились с типом оценки и подготовили вашу среду Kali, считайте, что вы уже практически готовы к началу работы. Последним шагом перед началом будет формализация той работы, которую вам необходимо будет проделать. Это является критически важным, т.к. это определяет ваши ожидания от работы и дает вам разрешение на проведение нелегальной активности в случае, если эти ожидания не будут оправданы. Мы затронем этот вопрос на высоком уровне, но, тем не менее, это является довольно таки сложным и важным шагом, так что вам, скорее всего, будет необходимо обратиться к юридическому представителю вашей организации за помощью.

В рамках процесса формализации вам необходимо будет определить правила участия в работе. Это затрагивает такие вопросы, как:

- С какими системами вы можете или вам разрешено взаимодействовать? Очень важно гарантировать, что вы случайно не вмешаетесь в какую-либор систему, которая будет иметь критическое значение для проведения бизнес-операций.
- В какое время дня, и через какое окно атаки, возможно, проводить оценку? Некоторые организации предпочитают ограничить время, в которое может проводиться оценка.
- Когда вы обнаружите потенциальную уязвимость, будет ли вам разрешено использовать её? Если нет, то каков процесс утверждения? Есть некоторые организации, которые предпочитают прибегать к очень контролируемому подходу к каждой попытке эксплуатации, тогда как другие хотели бы использовать более реалистичный подход. Лучше всего четко определить эти ожидания перед началом работы.
- В случае если была обнаружена существенная проблема, каким образом следует её решать? Иногда, организации хотят быть проинформированы о ней сразу же, в противном случае, проблема рассматривается в конце оценки.
- В случае неотложных ситуаций, с кем вам необходимо связываться? Всегда необходимо знать, с кем вам нужно связаться, если возникает какая-либо проблема требующая неотложного решения.

- Кто будет в курсе проводимой вами деятельности? Как эти данные будут переданы им? В некоторых случаях организации захотят проверить свою реакцию на возникшую проблему и эффективность обнаружения в рамках оценки. Это всегда является хорошей идеей, иметь эти данные заранее, так что вы будете знать, должны ли вы проявлять какую-либо степень скрытности в подходе к оценке.

Каковыми являются ожидания в конце оценки? Каким образом будут переданы результаты? Вы должны точно знать, что именно все стороны ожидают в конце оценки. Определение результатов и способа их передачи - лучший способ сохранить все стороны счастливыми после завершения работы.

Несмотря на то, что данный список не является полным, он дает вам представление о деталях, которые необходимо учитывать. Тем не менее, вы должны понимать, что не существует хорошей замены для законных действий. Как только вы определились с вышеуказанными пунктами, вам необходимо получить соответствующее разрешение для проведения оценки, поскольку большая часть деятельности, которую вы будете выполнять в ходе оценки, может быть не законной без надлежащих полномочий от кого-либо, обладающего полномочиями на предоставление этого разрешения.

После того как вы определились со всеми вышеуказанными пунктами и проделали все необходимые шаги, тем не менее остается еще один момент, который вам будет необходимо разрешить перед началом работы: проверка. Никогда не доверяйте тому диапазону (области) работы, который вам предоставили – всегда проверяйте его. Используйте множество источников информации для подтверждения того, что система в пределах вашего диапазона действительно принадлежит клиенту и, что она также обслуживается вашим клиентом. С преобладанием облачных сервисов организация может забыть, что они фактически не владеют системами, предоставляющими им услуги. Вы можете обнаружить, что вам на самом деле необходимо получить специальное разрешение от провайдера облачного сервиса, перед тем, как приступить к работе. Кроме того, всегда проверяйте узлы IP адреса. Никогда не стоит полагаться на предположение организации о том, что им принадлежат все узлы

IP, даже если они готовы подписаться под ними, как под реальной жизнеспособной целью. Например, мы видели примеры организаций, которые запрашивают оценку всего диапазона сети класса C, когда на самом деле они принадлежали только подмножеству этих адресов. Напав на все адресное пространство класса C, мы столкнулись бы с атакой на соседей сети организации. Подкатегория Анализа OSINT (OSINT Analysis), а именно меню Сбор Информации (Information Gathering) содержит довольно-таки большое количество различных инструментов, которые могут помочь вам с процессом проверки.

11.4 Типы Атак

Как только вы сможете приступить к работе, какие именно конкретные типы атак, которые вы собираетесь проводить? Каждый тип уязвимости[73] имеет свою конкретную связанную с ним технику. Этот раздел затронет различные классы уязвимостей, с которыми вы будете взаимодействовать в большинстве случаев.

Не имеет никакого значения, с какой категорией уязвимостей вы сталкиваетесь, Kali помогает вам с легкостью найти подходящие инструменты и эксплойты. Kali меню на вашем графическом пользовательском интерфейсе разделено на категории для того, чтобы помочь вам с лёгкостью найти подходящий инструмент. Кроме того на вебсайте Kali Tools [74] имеется исчерпывающий список различных инструментов, доступных в Kali, которые организованы в категории и отмечены соответствующим образом для более легкого просмотра. Каждая запись содержит детальную информацию об инструменте, также как и о примерах его использования.

11.4.1 DoS атака (Denial of Service)

DoS атаки используют уязвимость для того, чтобы создать потерю обслуживания, чаще всего прерыванием уязвимого процесса. Категория Стресс-тестирования (The Stress Testing) меню Kali

[73]https://www.cvedetaits.com/vutnerabitities-by-types.php
[74]http://tools.kati.org/tools-tisting

Linux содержит определенное количество инструментов для этой цели.

Когда большинство людей слышат термин «DoS атака», они моментально начинают думать об атаках потребления ресурсов, которые исходят из множества источников одновременно против одной цели. Этот тип атаки будет называться ***распределенная*** DoS атака (***distributed*** denial of services attack), или DDoS. Подобные атаки редко являются частью профессиональной оценки безопасности.

На самом деле одиночная DoS атака чаще всего является результатом неправильной попытки использовать уязвимость. Если автор эксплойта выпускает частично функциональный или концептуально верный (proof-of-concept PoC) код, и он используется непосредственно в деле, это может создать условия для DoS атаки. Даже верно закодированный эксплойт способен работать только в очень специфических обстоятельствах, но он может начать DoS атаку и при меньших обстоятельствах. Может показаться, что самым верным решением будет использовать только надежный и проверенный эксплойт или же написать свой собственный. Даже в таких условиях и при таком решении нет никаких гарантий, и это довольно сильно ограничивает оценщика, т.к. вызывает чрезмерные ограничения, что в свою очередь выливается в худшие результаты оценки. Решением в данной ситуации является компромисс. Избегайте концептуально верных (PoC) кодов и непроверенных эксплойтов в реальной работе и всегда заручайтесь поддержкой юриста, который сможет прикрыть вас, в случае неудачи.

Как правило, большинство DoS атак не запускаются намеренно. Большинство автоматизированных инструментов уязвимости отнесут уязвимости к DoS атакам к группе меньшего риска, ввиду того факта, что хотя вы и можете удалить службу из операции, эта служба не может быть использована для выполнения кода. Однако, очень важно помнить, что не все эксплойты выпускаются публично и любая уязвимость к DoS атакам может замаскироваться намного глубже, что представляет собой намного большую угрозу. Код выполнения эксплойта для DoS атаки может существовать, но не быть общедоступным. Основная мысль заключается в том, чтобы вы обращали внимание на уязвимости к

DoS атакам и убедите своего клиента установить соответствующий патч независимо от их (чаще всего низкого) уровня угрозы.

11.4.2 Повреждение памяти (Memory Corruption)

Повреждение памяти происходит, когда местоположение процесса в пространстве памяти случайно изменяется из-за ошибок программирования. Неполадки, связанные с повреждением памяти, приводят к непредсказуемому поведению программы, однако, в большинстве случаев, эти неполадки позволяют манипулировать памятью процесса таким образом, при котором поток выполнения программы можно контролировать, что позволяет определять активность, зависящую от атакующего.

Эти атаки обычно относят к переполнению буфера, хотя использование этого термина является чрезмерным упрощением. Наиболее распространенные типы повреждения памяти значительно отличаются друг от друга, т.к. каждая из них обладает собственной тактикой и техникой, необходимой для успешной эксплуатации.

- Переполнение буфера стэка (Stack Buffer Overflow): когда программа записывает больше данных в буфер в стеке, чем пространство, которое для него доступно, смежная память может быть повреждена, что часто приводит к сбою программы.
- Разрушение хипа (Heap Corruption): Хип память обычно распределяется во время выполнения задач и обычно содержит данные запущенной программы. Разрушение хипа происходят путем манипулирования данными для перезаписи с помощью связанного с ним списка хипов памяти.
- Целочисленное переполнение (Integer Overflow): Эти переполнения возникают, когда приложение пытается создать числовое значение, которое не может содержаться в пределах выделенного пространства для хранения.
- Строка форматирования (Format String): Когда программа принимает ввод пользователя и форматирует его без проверки, ячейки памяти могут быть обнаружены или перезаписаны в зависимости от используемых обозначений (маркеров)

формата.

11.4.3 Веб-Уязвимости (Web Vulnerabilities)

В связи с тем, что современные веб-сайты больше не являются статическими страницами, но вместо этого динамически генерируются для пользователя, среднестатистический веб-сайт является довольно-таки сложным. Веб-уязвимости используют эту сложность в попытке атаковать как back end логику страницы, так и презентацию для посетителя сайта.

Эти типы атаки довольно популярны, так как в наше время многие организации достигли той точки, когда у них очень мало внешних служб. Двумя наиболее распространенными типами атаки веб-приложений являются SQL-инъекция и межсайтовый скриптинг (XSS).

- SQL-инъекция (SQL injection): Эти атаки используют некорректно запрограммированные приложения, которые неправильно обрабатывают пользовательский ввод, что приводит к возможности запуска информации из базы данных или даже к полному захвату сервера.
- Межсайтовый скриптинг (Cross-site scripting): Как и в случае с SQL-инъекцией, атаки XSS являются результатом неправильной обработки ввода пользователя, что позволяет злоумышленникам манипулировать пользователем или сайтом с помощью выполнения кода в контексте их собственного сеанса браузера.

Комплексные, разноплановые и сложные веб-приложения являются очень распространёнными в наше время, предоставляя идеальные условия для атаки различным злоумышленникам. Вы найдете довольно много ценной и полезной для вас информации и инструментов в разделе меню Анализ Веб Приложений (Web Application Analysis) и в метапакете *kali-linux-web.*

11.4.4 Атака взлома пароля (Password Attacks)

Атаки взлома паролей - это атаки на систему аутентификации службы. Эти атаки довольно часто превращаются в онлайн атаки взлома паролей и офлайн атаки взлома паролей, с чем вы можете ознакомиться в разделе меню Атаки взлома паролей (Password Attacks). Во время онлайн атаки взлома пароля, множество различных паролей перебираются против запущенной системы. Во время офлайн атаки взлома пароля, сначала атакующий получает хэшированные или зашифрованные значения паролей, а затем пытается получить чистое текстовое значение. Защитой от данного типа атак является тот факт, что вычислительно довольно тяжело работать в этом процессе, ограничивая количество попыток, которые могут быть сгенерированы в секунду. Тем не менее, существует обходные пути, как, например, использование графических процессоров (GPU) для ускорения количества попыток, которые могут быть сделаны. Метапакет ***kali-linux-gpu*** содержит определенное количество инструментов, которые используют эту мощность.

Чаще всего атаки взлома паролей, нацелены на пароли, установленные поставщиком по умолчанию. В связи с тем, что эти пароли хорошо всем известны, атакующий проведет сканирование в надежде на удачу. Другая, не менее распространённая атака, это атака перебор по словарю (dictionary attack), во время которой создается список слов, который был адаптирован к целевой среде. Затем проводится онлайн атака взлома пароля против распространенных, по умолчанию или же просто известных учетных записей, где каждое слово перебирается в определенной последовательности.

Во время оценки, очень важно понимать потенциальные последствия данного типа атаки. Во-первых, они всегда создают много шума, ввиду повторяющихся попыток аутентификации. Во-вторых, подобные атаки часто могут привести к блокированию аккаунта из-за огромного количества неудачных попыток входа, проведенных против данного аккаунта. И наконец, производительность данных атак довольно медленная, что приводит к трудностям при попытке использовать исчерпывающий список слов.

11.4.5 Атаки на клиентов (Client-Side Attacks)

Большинство атак проводится против серверов, но т.к. сервисы становится все сложнее атаковать, более легкие цели, в данном случае, уже были выбраны. Результатом подобного выбора стали атаки на клиентов, где целью атакующего будет являться различные приложения, которые установлены на устройстве сотрудника целевой организации. В разделе Инструменты Социальной инженерии (The Social Engineering Tools) имеется большое количество прекрасных приложений, которые помогут провести вам этот тип атаки.

Данный способ атаки лучше всего применяется против Flash, Acrobat Reader, и Java, которые были довольно широко распространены в ранних 2000ых. В этих случаях злоумышленники будут пытаться запросить цель посетить вредоносную веб-страницу. Эти страницы будут содержать специализированный код, который запустит уязвимости в приложениях? установленных у клиента, что приведет к возможности запустить код на целевой системе.

Атаки на клиента всегда очень сложно предотвратить, это потребует качественного обучения пользователей, постоянного обновления приложений и сетевых средств управления для эффективного снижения риска.

11.5 Подведем итоги

В этой главе мы кратко рассмотрели роль Kali в сфере безопасности информации. Мы обсудили важность чистой, рабочей инсталляции и использование шифрования перед тем, как непосредственно начать работу, чтобы защитить информацию клиентов, а также важность законности ваших действий для защиты ваших интересов и интересов вашего клиента.

Компоненты триады CIA (конфиденциальность, целостность и доступность) являются первичными пунктами, на которые вы обращаете внимание, кouда пытаетесь обезопасить систему как часть стандартного развертывания системы, её поддержки или

оценки. Эта концептуальная основа поможет вам идентифицировать критические компоненты вашей системы и количество усилий или ресурсов, которые необходимо затратить для решения возникших проблем.

Мы обсудили несколько типов уязвимостей, таких как выполнения файла (file inclusion), SQL инъекция, переполнение буфера и состояние гонки (race conditions).

Точность сигнатуры чрезвычайно важна для получения полезных результатов оценки уязвимости. Чем больше представлено данных, тем выше шанс получить точные результаты от автоматизированного сканирования на основе сигнатуры, и именно поэтому данный тип сканирования сейчас очень популярен.

В связи с тем, что автоматизированные инструменты используют сигнатуры для выявления уязвимостей, любое незначительное отклонение от известной сигнатуры может изменить результат и обоснованность воспринимаемой уязвимости.

Мы также обсудили четыре типа оценки:

1. ***оценка уязвимости,***
2. ***тестирование на проникновение на основе соответствия,***
3. ***традиционное тестирование на проникновение***
4. ***оценка приложения.***

Несмотря на то, что каждый тип оценки использует ключевой набор инструментов, многие инструменты и технологии переплетаются.

Оценка уязвимости относительно проста по сравнению с другими типами оценок и часто состоит из автоматизированной записи обнаруженных вопросов в целевой среде. В этом разделе мы обсудили, что уязвимость – это неполадка или изъян, который при обнаружении может подвергнуть риску конфиденциальность, целостность или доступность информационной системы. В связи с тем, что этот тип оценки основан на сигнатуре, он базируется на точных сигнатурах и может представлять ложноположительные или ложноотрицательные результаты. Вы можете найти ключевые

инструменты для этого типа оценки в разделе меню Kali Linux - Анализ уязвимости (Vulnerability Analysis) и инструменты эксплуатации (Exploitation Tools).

Тестирование на проникновение на основе соответствия базируется на правительственных и санкционированных промышленностью требованиях (таких как PCI DSS, DISA STIG, и FISMA), которые, в свою очередь, основаны на стандарте соответствия. Этот тест обычно начинается с оценки уязвимости.

Традиционное тестирование на проникновение – это глубокая оценка безопасности, которая создана для того, чтобы улучшить общую ситуацию безопасности организации, основанной на определенных реальных угрозах. Этот тип тестирования включает несколько шагов (отраженных в структуре меню Kali Linux) и завершается использованием уязвимостей и переключением доступа с одной машины к другим машинам и сетям в пределах целевого диапазона.

Оценки приложения (обычно white-box или black-box) концентрируются на одном приложении и используют специализированные инструменты, которые были обнаружены в категориях меню - Анализ Веб Приложения (Web Application Analysis), Оценка Базы данных (Database Assessment), Обратная Разработка (Reverse Engineering) и Инструменты Эксплуатации (Exploitation Tools).

Обсуждались несколько видов атак, включая: DoS атаку (Denial of Service), которая нарушает поведение приложения и делает его недоступным; повреждение памяти, которая приводит к манипуляции доступной памятью, что часто позволяет выполнение кода атакующей стороной злоумышленника; веб-атаки, которые атакуют веб-сервисы, используя технологии типа SQL инъекции и межсайтовый скриптинг (XSS атаки); и атаки взлома пароля, которые часто используют список паролей для атаки сервисных учетных данных.

Часть 12: Заключение

Поздравляем! Надеемся, сейчас вы намного лучше познакомились и узнали систему Kali Linux, и вам не следует бояться экспериментировать с ней. Вы обнаружили ее самые интересные свойства, но также вы знаете пределы ее возможностей, а также способы, с помощью которых их можно обойти.

Если вы пока еще не использовали все свойства, держите эту книгу поблизости, чтобы при необходимости заглянуть в нее и освежить свою память. Помните, что нет ничего лучше практики (или настойчивости), чтобы развивать новые навыки. «Старайтесь упорнее[75]», - как не устают повторять специалисты Offensive Security.

12.1 Продолжаем следить за обновлениями

С постоянно изменяющимся дистрибутивом kali-rolling *некоторые разделы книги обязательно станут устаревшими. Мы сделаем все возможное, чтобы идти в ногу со временем (по крайней мере, относительно онлайн версии), а для некоторых разделов мы попытались дать общее объяснение, которое будет действительно долгое время.*

В нем говорится, что вы должны быть готовы принимать изменения и искать решение возникающих проблем. При лучшем понимании Kali Linux и ее отношения к Debian вы можете рассчитывать на сообщества Kali Linux и Debian и их многочисленные ресурсы (баг трекеры, форумы, списки рассылки и т.д.), если у вас возникли сложности.

Не бойтесь документировать возникшие ошибки (см.раздел 6.3, “Подача грамотно составленного отчета об ошибке” [стр.129])! Если вы, как и я, завершили заполнение хорошего отчета об ошибках (у меня это заняло какое-то время), вы уже решили проблему или, по крайней мере, нашли пути ее обхода. А

[75]https://www.offensive-security.com/offsec/say-try- harder/

действительным составлением отчета об ошибке, вы поможете другим, которые столкнулись с такой проблемой.

12.2 Подтвердите полученные вами

Вы гордитесь новыми навыками Kali Linux? Вы можете с уверенностью сказать, что помните по-настоящему важные вещи? Если вы ответите да, то вам стоит обратиться к программе Kali Linux Certified Professional.

Это хорошее подтверждение, которое гарантирует, что вы знаете, как разворачивать (устанавливать) и использовать Kali Linux в различных случаях. Это хорошее дополнение к вашему резюме, которое доказывает, что вы можете двигаться дальше.

12.3 Движемся дальше

Эта книга научила вас многому, а именно большинству вещей, которые должен знать любой пользователь Kali Linux, но, тем не менее, нам пришлось сделать множество нелегких выборов, для того, чтобы придерживаться приемлемого объема книги. Ввиду этого, нам не удалось затронуть множество не менее важных вопросов и тем.

12.3.1 Относительно системного администрирования

Если вы хотите узнать больше о системном администрировании, то в таком случае мы можем порекомендовать вам, ознакомиться со Справочником администратора Debian (Debian Administrator’s Handbook): https://debian-handbook.info/get/

Вы найдете там множество дополнительных глав о распространённых службах Unix, которые нам, к сожалению, пришлось полностью опустить в нашей книге. И даже в тех главах, которые мы использовали в этой книге, вы найдете множество дополнительных советов, в основном относительно системы

пакетирования (которая также рассматривается более детально на своем самом низком уровне)

Книга Debian (Debian book) очевидно представит более глубоко сообщество Debian и то, каким образом оно организовано. Не смотря на то, что эти знания не являются жизненно важными, тем не менее, это действительно полезно знать особенно, когда вам нужно взаимодействовать со специалистами Debian, например, когда вы вместе работаете над отчетом об ошибке.

12.3.2 Относительно тестирования на проникновение

Возможно, сейчас вы заметили, что эта книжка не учит вас непосредственно проведению тестирования на проникновение, но, тем не менее, те вещи, которые вы изучили, являются очень важными. Теперь вы готовы полностью использовать всю мощь Kali Linux, лучше платформы для проведения тестирования на проникновение. И вы получили базовые навыки в системе Linux, которые необходимы для начала тренировок с Offensive Security.

Если вы чувствуете, что все еще не готовы к тому, чтобы начать проходить платные курсы, вы можете начать бесплатные онлайн курсы Metasploit Unleashed [76] . Metasploit является очень популярным инструментом для проведения тестирования на проникновение, и вам просто необходимо ознакомиться с ним, если вы серьезно настроены изучить тестирование на проникновение.

Следующим логическим шагом будет изучение онлайн курса Penetration Testing with Kali Linux[77] , который проведет вас по известному пути получение сертификации “Offensive Security Certified Professional”. Этот онлайн-курс можно выполнять в своем собственном темпе, но сертификация на самом деле представляет собой сложное, 24-часовое, реальное, практическое тестирование на проникновение, которое проходит в изолированной сети VPN.

Вы готовы к новым вызовам и препятствиям?

[76]https://www.offensive-security.com/metasptoit- unleashed/
[77]https://www.offensive-security.com/information- security-training/