ГЛАВА 1

Основы Kali Linux

Kali Linux — это специализированный дистрибутив операционной системы Linux, основанный на Ubuntu Linux, которая, в свою очередь, базируется на Debian Linux. Данный дистрибутив ориентирован на людей, занимающихся вопросами безопасности, будь то ее тестирование, разработка эксплойтов, реверс-инжиниринг или цифровая криминалистика. Следует помнить о том, что дистрибутивы Linux отличаются друг от друга. Операционная система Linux представляет собой ядро дистрибутивов, каждый из которых накладывает поверх этого ядра дополнительное программное обеспечение, делая его уникальным. В случае с Kali речь идет не только об основных утилитах, но и о сотнях программных пакетов, предназначенных для решения задач, связанных с безопасностью.

Одна из весьма приятных особенностей Linux, отличающих эту ОС от прочих, заключается в высокой степени настраиваемости. Она позволяет выбрать оболочку, из которой будут запускаться программы, терминал для ввода команд, а также графическое окружение рабочего стола. Кроме того, после выбора среды вы можете изменить внешний вид каждого из этих элементов. Linux позволяет вам подстроить систему под свои потребности, вместо того чтобы менять свой стиль работы в соответствии с ее особенностями.

У Linux длинная история, ознакомление с которой помогает понять, почему эта ОС стала именно такой, какая она есть. Особенно это касается нескольких загадочных команд, которые используются для управления данной системой, манипулирования файлами и решения повседневных задач.

Наследие Linux

Когда-то давно, во времена динозавров или, точнее, компьютеров размером с холодильник, существовала операционная система под названием *Multics*. Этот начатый в 1964 году проект был разработан МТИ совместно с General Electric (GE) и Bell Labs. Цель Multics заключалась в создании операционной системы, рассчитанной на множество пользователей, и разделении процессов и файлов между ними. В конце концов, в ту эпоху компьютерное оборудование, необходимое для работы ОС, подобных Multics, стоило миллионы. Минимальная стоимость компьютерного оборудования исчислялась сотнями тысяч долларов. Для сравнения: система, стоившая 7 млн долларов в то время, в апреле 2023 года стоила бы около

62 млн долларов. Система, способная обслуживать только одного пользователя за раз, была просто нерентабельна, поэтому производители компьютеров, такие как GE, были заинтересованы в разработке Multics совместно с такими исследовательскими организациями, как МТИ и Bell Labs.

Из-за сложностей и конфликтующих интересов участников проект постепенно развалился, хотя операционная система все-таки была выпущена. Один из работавших над проектом программистов Bell Labs, вернувшись на постоянную работу, решил создать собственную версию операционной системы, чтобы иметь возможность играть в игру, написанную им для Multics, на компьютере PDP-7, который использовался в Bell Labs. Эта игра называлась Space Travel, и для ее перенесения на PDP-7 программисту Кену Томпсону требовалась подходящая среда. В те дни системы были несовместимы друг с другом. У них были совершенно разные аппаратные инструкции (коды операций), а иногда и разные размеры слов памяти, которые сегодня часто называются *разрядностью шины данных*. Из-за этого программы, написанные для одной среды, особенно с использованием языков очень низкого уровня, не работали в другой среде. Разработанная Томпсоном среда получила название *Unics*. Со временем к проекту присоединились другие программисты Bell Labs, и она была переименована в *Unix*.

Unix была устроена очень просто. Разработанная как среда программирования для одного пользователя, она применялась сначала внутри Bell Labs, а затем стала применяться и за пределами этой организации. Одним из самых больших преимуществ Unix перед другими операционными системами стало то, что в 1972 году ее ядро было переписано на языке программирования C. Использование языка более высокого уровня, чем ассемблер, который в то время был очень распространен, сделало эту ОС переносимой на другие аппаратные системы. Благодаря этому Unix могла работать не только на компьютере PDP-7, но и на любой другой системе, предусматривающей компилятор для языка C, на котором был написан исходный код Unix. Это позволило создать стандартную операционную систему, рассчитанную на множество аппаратных платформ.

Язык ассемблера максимально приближен к двоичному коду, который понимает компьютер. Этот язык состоит из мнемоник, с помощью которых люди обозначают операции, понятные процессору. Как правило, мнемоника представляет собой короткое слово, описывающее операцию. Например, инструкция *CMP* сравнивает (compare) два значения, а инструкция *MOV* перемещает (move) данные из одного места в другое. Язык ассемблера предоставляет полный контроль над работой программы, поскольку написанный на нем код переводится непосредственно на машинный язык, то есть преобразуется в двоичные значения, обозначающие операции процессора и адреса памяти.

Помимо простоты, преимуществом Unix было свободное распространение ее исходного кода, что позволяло исследователям не только читать, но и дорабатывать его. Язык ассемблера, который широко использовался до этого, очень сложен для восприятия, поэтому для изучения написанного на нем кода требуется много

времени и большой опыт. Языки более высокого уровня, такие как C, существенно облегчают чтение исходного кода. Unix породила множество дочерних операционных систем, которые вели себя так же, как она, и обладали схожей функциональностью. В одних случаях дистрибутивы других операционных систем создавались на основе исходного кода Unix, предоставленного компанией AT&T. В других случаях Unix подвергалась реверс-инжинирингу, в результате чего появились две популярные Unix-подобные операционные системы: BSD и Linux.

Как будет показано далее, одно из преимуществ Unix заключается в возможности создания цепочек программ, позволяющих подавать выходные данные одной простой программы на вход другой. Один из распространенных способов применения этого конструктивного решения — получение списка процессов путем передачи вывода одной утилиты другой утилите, которая обрабатывает этот вывод, выполняя поиск одной конкретной записи или удаляя фрагменты вывода, чтобы сделать его более понятным.

Появление Linux

Простота дизайна Unix, ее позиционирование как среды программирования, но главным образом доступность исходного кода со временем привели к тому, что работе с ней начали обучать в рамках курсов по информатике по всему миру. В 1980-х годах на основе дизайна Unix был написан ряд книг, посвященных проектированию операционных систем. Хотя использование оригинального исходного кода нарушало авторские права, обширная документация и простота дизайна позволяли создавать клоны этой операционной системы. Одна из таких реализаций была написана Эндрю Таненбаумом для его книги *Operating Systems: Design and Implementation*[1]. На основе этой реализации под названием *Minix* Линус Торвальдс разработал ядро операционной системы Linux. Оно позволяло управлять аппаратными устройствами, включая процессор, то есть запускать процессы через центральное процессорное устройство (ЦПУ). При этом оно не давало пользователям возможности взаимодействовать с операционной системой, то есть выполнять программы.

Проект GNU, основанный в конце 1970-х годов Ричардом Столлманом, предусматривал коллекцию программ, которые либо полностью дублировали стандартные утилиты Unix, либо выполняли те же функции, но имели другие названия. В рамках проекта GNU программы писались в основном на языке C, что обеспечивало их переносимость. В результате Торвальдс, а позднее и другие разработчики объединили утилиты проекта GNU с ядром Linux, чтобы создать полноценный дистрибутив программного обеспечения, которое каждый человек мог доработать и установить на свою компьютерную систему. Набор утилит GNU иногда (по крайней мере, так было исторически) называется *userland* и определяет способ взаимодействия пользователей с системой.

---

[1] *Таненбаум Э.* Операционные системы: разработка и реализация. — СПб.: Питер, 2006.

Система Linux унаследовала большинство особенностей дизайна Unix, прежде всего потому, что данная ОС создавалась как нечто функционально идентичное стандартной операционной системе Unix, которая была разработана AT&T и переработана небольшой группой сотрудников Калифорнийского университета в Беркли в систему BSD (Berkeley Systems Distribution). Это означало, что любой человек, знакомый с Unix или BSD, мог сразу же начать эффективно применять Linux. За десятилетия, прошедшие с момента выпуска Торвальдсом ядра Linux, было создано множество проектов, направленных на расширение функциональности и повышение удобства использования этой ОС. К ним относятся несколько окружений рабочего стола, каждое из которых базируется на системе X Window System, разработанной в МТИ (который принимал участие и в создании Multics).

Создание Linux, то есть ее ядра, изменило сам подход к процессу разработки ПО. Например, Торвальдс был недоволен репозиториями, которые позволяли разработчикам одновременно работать над одними и теми же файлами. В результате он возглавил разработку системы контроля версий *Git*, которая вытеснила практически все остальные подобные системы, используемые для разработки ПО с открытым исходным кодом. Если вам понадобится текущая версия исходного кода современной свободно распространяемой программы, то в большинстве случаев вы будете получать доступ к ней именно через Git. Кроме того, в настоящее время существуют публичные репозитории для хранения кода проектов, которые позволяют использовать менеджер исходного кода Git для получения доступа к нему. Помимо разработчиков проектов с открытым исходным кодом, многие, если не большинство компаний применяют Git в качестве системы контроля версий благодаря ее современному децентрализованному подходу к управлению исходным кодом.

Сравнение монолитных и микроядерных ОС

Ядро Linux *монолитно*. Этим данная ОС отличается от Minix и других Unix-подобных реализаций, использующих *микроядра*. Разница между монолитным ядром и микроядром заключается в том, что в монолитное ядро встроена вся функциональность. Сюда входит любой код, необходимый для поддержки работы аппаратных устройств. В свою очередь, микроядро включает в себя минимально необходимый код для поддержания работоспособности операционной системы. Любая дополнительная функциональность реализуется в виде модуля и загружается в пространство ядра по мере необходимости. Это не значит, что в Linux нет модулей, однако ядро, которое обычно включается в дистрибутивы Linux, не относится к микроядрам. Поскольку в основе Linux не лежит идея реализации в ядре лишь основных сервисов, эта операционная система считается не микроядерной, а монолитной.

Linux поставляется в виде дистрибутивов, которые, как правило, распространяются бесплатно. *Дистрибутив* Linux представляет собой коллекцию программных пакетов, отобранных людьми, отвечающими за сопровождение этого дистрибутива. Сами пакеты собираются определенным образом и оснащаются функциями, отбираемыми сопровождающими пакета. Эти программные пакеты поставляются в виде исходного кода, и многие из них предусматривают различные параметры (например, связанные с поддержкой базы данных или шифрованием), которые задаются при настройке и сборке пакета. Сопровождающий пакета, предназначенного для одного дистрибутива, может выбрать параметры, отличные от тех, которые выбрал сопровождающий пакета для другого дистрибутива.

Разные дистрибутивы предусматривают также разные форматы пакетов. Например, дистрибутив Red Hat и такие родственные ему дистрибутивы, как RedHat Enterprise Linux (RHEL) и Fedora Core, используют формат RPM (Red Hat Package Manager). Кроме того, для управления пакетами Red Hat задействует как утилиту RPM, так и инструмент YUM (Yellowdog Updater Modified). Другие дистрибутивы могут применять другие утилиты для управления пакетами, используемые Debian, в частности APT (Advanced Package Tool). Вне зависимости от дистрибутива и формата цель пакетов заключается в том, чтобы собрать все файлы, необходимые для работы программного обеспечения, и облегчить их использование по назначению. Поскольку Kali Linux базируется на Debian, этот дистрибутив также задействует APT для управления пакетами как в плане поддерживаемых форматов, так и в плане применяемых для этого инструментов.

С годами еще одним различием между дистрибутивами стала среда рабочего стола, которая предоставляется дистрибутивом по умолчанию. В последнее время дистрибутивы стали предусматривать собственные разновидности таких сред. Все среды, включая GNU Object Model Environment (GNOME), K Desktop Environment (KDE) или Xfce, допускают настройку с помощью тем, обоев и особой организации меню и панелей. Дистрибутивы часто предлагают свои варианты среды рабочего стола. А некоторые из них даже предусматривают собственную среду. В качестве примера можно привести среду Pantheon системы ElementaryOS.

Несмотря на то что результат работы всех менеджеров пакетов одинаков, иногда выбор такого менеджера и даже среды рабочего стола может повлиять на пользовательский опыт. Кроме того, для некоторых пользователей может иметь значение глубина репозитория пакетов. Они могут предпочесть наличие большого выбора программ, которые можно установить через репозиторий, процессу ручной сборки и установки программного обеспечения. Одни дистрибутивы могут предусматривать репозитории меньшего размера, чем другие, даже будучи основанными на тех же утилитах управления пакетами и форматах. Из-за программных зависимостей, установка которых необходима для функционирования программного обеспечения, пакеты не всегда сочетаются друг с другом, даже являясь компонентами родственных дистрибутивов.

Некоторые дистрибутивы являются не универсальными, а ориентированными на определенные группы пользователей. Кроме того, такие дистрибутивы, как Ubuntu,

даже предусматривают отдельные версии для установки на сервер и на обычный настольный компьютер. Настольная версия обычно включает графический интерфейс пользователя (GUI), в то время как серверная версия его не предусматривает и, как следствие, позволяет установить гораздо меньше пакетов. Чем меньше пакетов, тем меньше вероятность подвергнуться атаке. Это очень важно, учитывая то, что на серверах часто хранится конфиденциальная информация. Кроме того, серверы больше подвержены атакам со стороны неавторизованных пользователей, поскольку они, в отличие от настольных систем, нередко предоставляют сетевые услуги.

Kali Linux — это дистрибутив, специально разработанный для пользователей, которые интересуются вопросами обеспечения информационной безопасности. Однако он относится к категории настольных систем. Это говорит о том, что его разработчики не пытаются ограничить количество устанавливаемых пакетов в стремлении сделать Kali менее подверженным атакам. Специалисту, занимающемуся тестированием безопасности, необходим широкий спектр программных пакетов, и дистрибутив Kali предоставляет к ним доступ. Это может показаться несколько ироничным, учитывая то, что дистрибутивы, ориентированные на защиту от атак (иногда их ошибочно называют *безопасными*), обычно ограничивают количество пакетов с помощью процесса, называемого *укреплением* (hardening). Однако дистрибутив Kali ориентирован на тестирование безопасности, а не на защиту от атак.

Kali Linux поддерживается компанией Offensive Security, которая занимается консалтингом и обучением в сфере безопасности. Кроме того, эта организация предлагает программу сертификации Offensive Security Certified Professional (OSCP) для людей, интересующихся наступательной безопасностью, например тестированием на проникновение и организацией деятельности «красной» команды.

Загрузка и установка Kali Linux

Самый простой способ получить дистрибутив Kali Linux — это посетить официальный веб-сайт (https://oreil.ly/TPahL). Там вы можете найти дополнительную информацию об этом программном обеспечении, в частности списки устанавливаемых пакетов, и загрузить ISO-образ, который можно использовать для установки на виртуальную машину, или записать на DVD для дальнейшей установки на физический компьютер.

Kali Linux базируется на Debian, но так было не всегда. Когда-то дистрибутив Kali назывался *BackTrack Linux*. BackTrack был основан на Knoppix Linux, который представлял собой «живой» (live) дистрибутив, то есть был разработан для загрузки с CD, DVD или USB-накопителя и запуска с исходного носителя, а не для установки на жесткий диск. Knoppix, в свою очередь, наследовал от Debian. Дистрибутив BackTrack, как и Kali Linux, был ориентирован на решение задач, связанных с тестированием на проникновение и цифровой криминалистикой.

Последняя версия BackTrack была выпущена в 2012 году, после чего команда Offensive Security воссоздала идею BackTrack на основе Debian Linux. Одна из особенностей Kali, унаследованная от BackTrack, — возможность «живой» загрузки. При использовании загрузочного носителя Kali вы можете выбрать либо установку, либо «живую» загрузку этого дистрибутива (рис. 1.1).

Рис. 1.1. Загрузочный экран Kali Linux

Выбор подходящего варианта зависит только от вас. При загрузке с DVD и отсутствии домашнего каталога, хранящегося на каком-либо носителе, вы не сможете сохранять данные между загрузками системы. Если у вас нет носителя для хранения информации, то при каждой загрузке все будет начинаться с нуля. Это преимущество, если вы не хотите, чтобы действия, предпринятые вами во время работы операционной системы, оставили какие-либо следы. Если же вы настраиваете SSH-ключи или другие учетные данные или хотите сохранить их, придется установить систему на локальный носитель.

Процесс установки Kali очень прост. В отличие от других дистрибутивов, которые позволяют выбирать категории устанавливаемых пакетов, Kali предусматривает их определенный набор. Вы можете что-то добавить или убрать, но изначально получаете исчерпывающий набор инструментов для тестирования безопасности и решения задач цифровой криминалистики. Процесс настройки сводится к выбору диска для установки, его разбиению на разделы и форматированию. Вам также необходимо настроить сеть, в том числе указать имя хоста и то, используете ли вы

статический IP-адрес или протокол DHCP (протокол динамической настройки узла). После того как вы все это настроите, установите часовой пояс и зададите некоторые другие основные параметры конфигурации, пакеты будут обновлены и все будет готово к загрузке Linux.

Виртуальные машины

Описанный подход может отлично работать на выделенном компьютере, но такие машины могут быть весьма дорогостоящими. Даже бюджетные компьютеры стоят немало, кроме того, для их работы требуется место и питание. А для некоторого оборудования может потребоваться еще и система охлаждения.

К счастью, Kali не требует использования выделенного компьютера. Она прекрасно работает на виртуальной машине (ВМ). Если вы собираетесь заниматься тестированием безопасности, в частности тестированием на проникновение, рекомендую создать виртуальную лабораторию. По моим наблюдениям, Kali отлично работает при использовании 4 Гбайт оперативной памяти и около 20 Гбайт дискового пространства. Если вы собираетесь хранить большое количество артефактов, полученных в ходе тестирования, вам может понадобиться дополнительное дисковое пространство. Для выполнения основных задач достаточно будет 2 Гбайт оперативной памяти, но очевидно, что чем больше памяти вы сможете выделить, тем выше будет производительность. Некоторые программы требуют больше памяти для эффективной работы.

Вы можете выбрать один из множества гипервизоров в зависимости от своей хостовой операционной системы. VMware предусматривает гипервизоры как для Mac, так и для ПК. Parallels работает на компьютерах Mac. В свою очередь, VirtualBox (https://oreil.ly/GpOJz) работает на ПК, компьютерах Mac, системах Linux и даже Solaris. Программа VirtualBox разработана в 2007 году, а в 2008-м была приобретена компанией Sun Microsystems. Затем компанию Sun приобрела корпорация Oracle, которая в настоящее время поддерживает VirtualBox. Тем не менее эту программу можно загрузить и использовать бесплатно. Если вы только начинаете знакомиться с миром виртуальных машин, это может стать для вас отличной отправной точкой. В плане взаимодействия с пользователями различные гипервизоры работают немного по-разному. Это выражается в нажатии разных клавиш для выхода из ВМ, разных уровнях взаимодействия с операционной системой, а также различной поддержке гостевых операционных систем, для которых гипервизор должен предоставлять драйверы. В конечном итоге выбор зависит от доступного вам бюджета и удобства использования.

К слову, одним из главных разработчиков BSD был Билл Джой, выпускник Калифорнийского университета в Беркли. Именно Джой разработал первую реализацию протоколов TCP/IP в Berkeley Unix. В 1982 году он стал соучредителем компании Sun Microsystems и, работая там, опубликовал статью о более совершенном языке программирования, чем C++, которая послужила толчком для разработки языка Java.

Один из критериев выбора гипервизора — предоставляемые им инструменты: драйверы, которые устанавливаются в ядро для обеспечения лучшей интеграции с хостовой операционной системой. К ним могут относиться драйверы печати, драйверы для совместного использования файловой системы хостовой и гостевой ОС, а также улучшенная поддержка видео. Система VMware может задействовать инструменты VMware с открытым исходным кодом, доступные в репозитории Kali Linux. Из этого же репозитория вы можете получить инструменты VirtualBox. В то же время программа Parallels предоставляет свои собственные инструменты. Одно из преимуществ использования VMware заключается в том, что драйверы с открытым исходным кодом доступны в большинстве дистрибутивов Linux, если не во всех. В прошлом у меня были некоторые проблемы с установкой Parallels Tools в некоторых версиях Linux, хотя в целом мне нравится эта программа. Если вы не хотите автоматически масштабировать экран в виртуальной машине Kali или обеспечивать обмен документами между хост-машиной и гостевой виртуальной машиной, то можете не обращать внимания на эти инструменты ВМ.

Дешевые вычисления

Если вы не хотите выполнять установку с нуля, но заинтересованы в использовании виртуальной машины, то можете загрузить образ VMware или VirtualBox. Kali поддерживает не только виртуальные среды, но и такие устройства на базе ARM (Advanced RISC Machine — усовершенствованная RISC-машина), как Raspberry Pi и BeagleBone. Преимущество образов ВМ заключается в том, что они позволяют приступить к работе, не тратя времени на установку. Для этого достаточно скачать образ и загрузить его в выбранный гипервизор. Если вы предпочитаете предварительно настроенную виртуальную машину, можете найти соответствующие образы на сайте Kali (https://oreil.ly/rM5lh).

Еще один бюджетный вариант запуска Kali Linux — очень недорогой и малогабаритный компьютер Raspberry Pi. Вы можете загрузить образ, предназначенный именно для него. В отличие от настольных компьютеров, Pi не использует процессор Intel или AMD. Вместо них в нем задействуется процессор ARM. Такие процессоры предусматривают меньший набор инструкций и потребляют меньше энергии, чем процессоры, обычно установленные в настольных компьютерах. Устройство Pi поставляется в виде платы размером с ладонь. Можете приобрести для него несколько корпусов и оснастить такими периферийными устройствами, как клавиатура, мышь и монитор.

Одно из преимуществ Pi — небольшой размер, позволяющий использовать его для осуществления физических атак. Вы можете установить Kali на этот компьютер и оставить его в том месте, где проводите тестирование, правда, придется обеспечить его питание и сетевое подключение. Pi предусматривает встроенный порт Ethernet, интерфейс Wi-Fi и порты USB. Установив на него Kali, вы сможете проводить локальные атаки удаленно, получая доступ к Pi через сеть. Мы рассмотрим некоторые из этих возможностей далее в книге.

Подсистема Windows для Linux

Многие люди используют Windows в качестве основной операционной системы, и это справедливо, учитывая ее удобство в плане решения большинства задач, выполняемых на настольном компьютере. В то время как виртуальные машины позволяют установить Linux на любую систему, Windows предусматривает для этого более простой способ. В 2016 году компания Microsoft выпустила подсистему *Windows Subsystem for Linux* (WSL), позволяющую запускать исполняемые двоичные файлы в ELF (Executable and Linkable Format) — стандартном формате исполняемых файлов для Linux непосредственно в Windows. Было создано две версии WSL. Первая представляла собой способ реализации системных вызовов Linux непосредственно в ядре Windows. Поскольку аппаратная архитектура не играет роли, так как исполняемые файлы Windows и Linux основаны на архитектуре процессоров Intel, основное внимание уделяется тому, как именно ядро Linux управляет аппаратным обеспечением. Это делается с помощью системных вызовов. Системные вызовы в Windows отличаются от системных вызовов в Linux. Реализация системных вызовов Linux в Windows — важный шаг к обеспечению работы исполняемых файлов Linux в Windows как нативных.

Недавно компания Microsoft изменила реализацию. Теперь настольные версии Windows включают облегченный гипервизор, являющийся реализацией Hyper-V, ранее доступного в качестве нативного гипервизора в Windows Server. В настоящее время система WSL реализуется с помощью ядра Linux, запущенного на машине Hyper-V. Приложения Linux обращаются напрямую к ядру Linux, а не к ядру Windows. Это объясняется тем, что некоторые системные вызовы Linux оказалось сложно реализовать в Windows. Реализация WSL в виртуализированной среде обеспечивает изоляцию, благодаря которой приложения Linux не могут повлиять на приложения Windows, так как выполняются в отдельных областях памяти.

Устанавливается подсистема WSL очень просто. Используя командную строку, будь то PowerShell или более старый командный процессор, выполните команду `wsl --install`. На рис. 1.2 видно, что Windows по умолчанию устанавливает версию ядра Ubuntu. Если у вас уже установлена старая версия WSL, можете преобразовать ее в WSL2 с помощью команды `wsl --upgrade`. О том, что применяется старая версия, вам будет сообщено при попытке взаимодействия с WSL, однако можно проверить это заранее, введя команду `wsl -l -v` в окно PowerShell или командную строку. После установки среды вы сможете запустить ее из меню Windows. Среда Ubuntu в этом меню по умолчанию будет называться Ubuntu.

Однако нас интересует не Ubuntu, а Kali. Вы можете найти Kali в приложении Microsoft Store. Если введете «Kali» в поисковой строке, то увидите результат, изображенный на рис. 1.3. Однако установка этого приложения не приводит к установке всего дистрибутива, а лишь предоставляет возможность запуска Kali Linux. При первом открытии Kali Linux в Windows будет установлен образ и вам предложат ввести имя пользователя и пароль. При последующем запуске Kali Linux вы будете автоматически авторизовываться в оболочке командной строки.

```
Installing: Virtual Machine Platform
Virtual Machine Platform has been installed.
Installing: Windows Subsystem for Linux
Windows Subsystem for Linux has been installed.
Installing: Ubuntu
[=                          2.0%                          ]
```

Рис. 1.2. Установка подсистемы WSL с помощью оболочки PowerShell

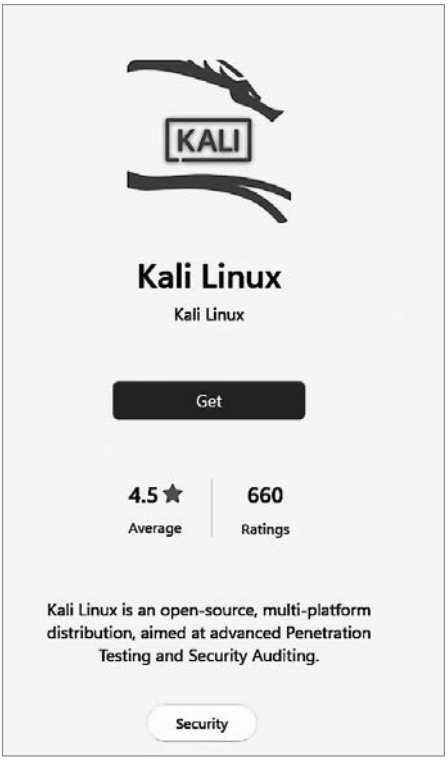

Рис. 1.3. Kali Linux в Microsoft Store

Базовый образ Kali в подсистеме WSL весьма компактен и предусматривает установку небольшого количества компонентов. Одно из преимуществ WSL2 — возможность запуска графических приложений непосредственно в Windows. Так было не всегда. В Windows можно было запускать программы командной строки, но запуск графических программ требовал использования дополнительного программного обеспечения для размещения этих приложений. Сегодня Windows предусматривает функциональность для размещения графических программ. В связи с этим вы, вероятно, решите установить ряд метапакетов для получения дополнительных приложений. В этом случае можете начать с `kali-linux-default`.

Он установит сотни дополнительных пакетов, которые помогут приступить к работе. У вас не будет полноценного графического рабочего стола, но вы сможете запускать графические программы непосредственно в Windows. На рис. 1.4 показан пример приложения Ettercap, запущенного в качестве графической программы из Linux на рабочем столе Windows.

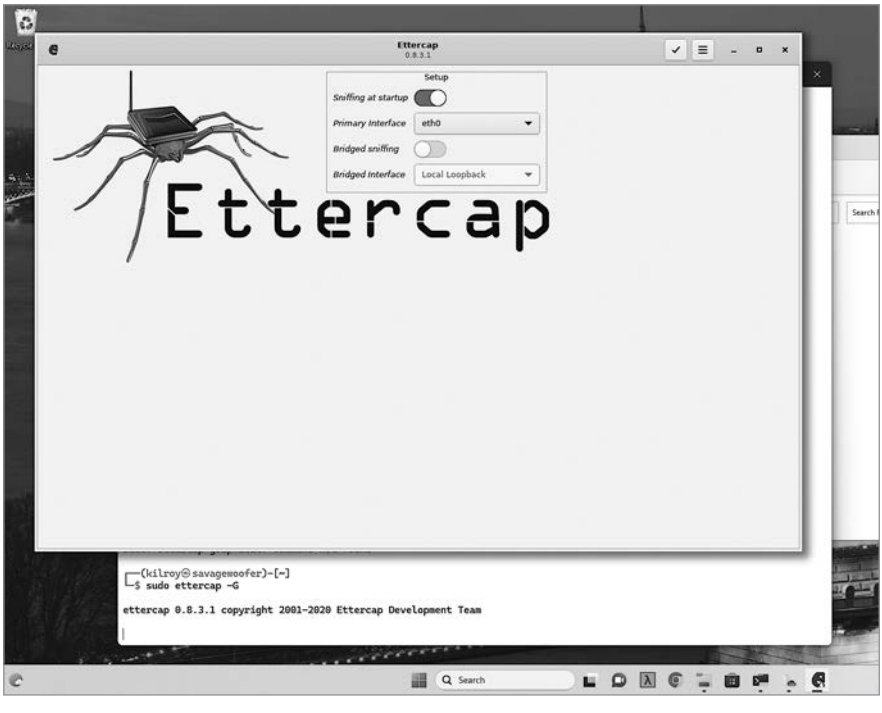

Рис. 1.4. Программа Ettercap, запущенная на рабочем столе Windows

Однако у подсистемы WSL есть некоторые ограничения. Вы столкнетесь с ними, когда приступите к тестированию беспроводных сетей. Вам понадобится физическая или виртуальная система, через которую вы сможете передать интерфейс.

Учитывая такое разнообразие вариантов, установка не должна занять много времени. После завершения установки нужно будет ознакомиться со средой рабочего стола, если выбранный вами вариант ее предусматривает.

Среды рабочего стола

Вы будете проводить много времени в среде рабочего стола, поэтому постарайтесь выбрать ту, с которой вам будет комфортно взаимодействовать. В отличие от таких проприетарных операционных систем, как Windows и macOS, Linux

предусматривает множество сред рабочего стола. Kali поддерживает использование популярных вариантов из их репозиториев, не требуя добавления дополнительных репозиториев. Если не устраивает среда рабочего стола, установленная по умолчанию, ее легко заменить. Поскольку вы, скорее всего, будете проводить в ней много времени, постарайтесь выбрать такие среду и инструменты, которые обеспечат вам не только комфорт, но и продуктивность.

Xfce

В настоящее время рабочий стол Kali по умолчанию — *Xfce*. Эта среда представляет собой довольно популярную альтернативу в силу своей легковесности и, как следствие, большей отзывчивости. Многие знакомые мне пользователи Linux выбирали Xfce в качестве предпочтительной среды, когда им требовалось рабочее окружение. Причина этого заключалась в простоте дизайна и высокой степени настраиваемости. На рис. 1.5 показана базовая конфигурация Xfce. Панель в верхней части рабочего стола полностью настраиваемая. Вы можете менять ее расположение и поведение, добавлять или удалять элементы в соответствии со своими предпочтениями. Эта панель включает в себя меню приложений, содержащее те же папки или категории, что и меню GNOME.

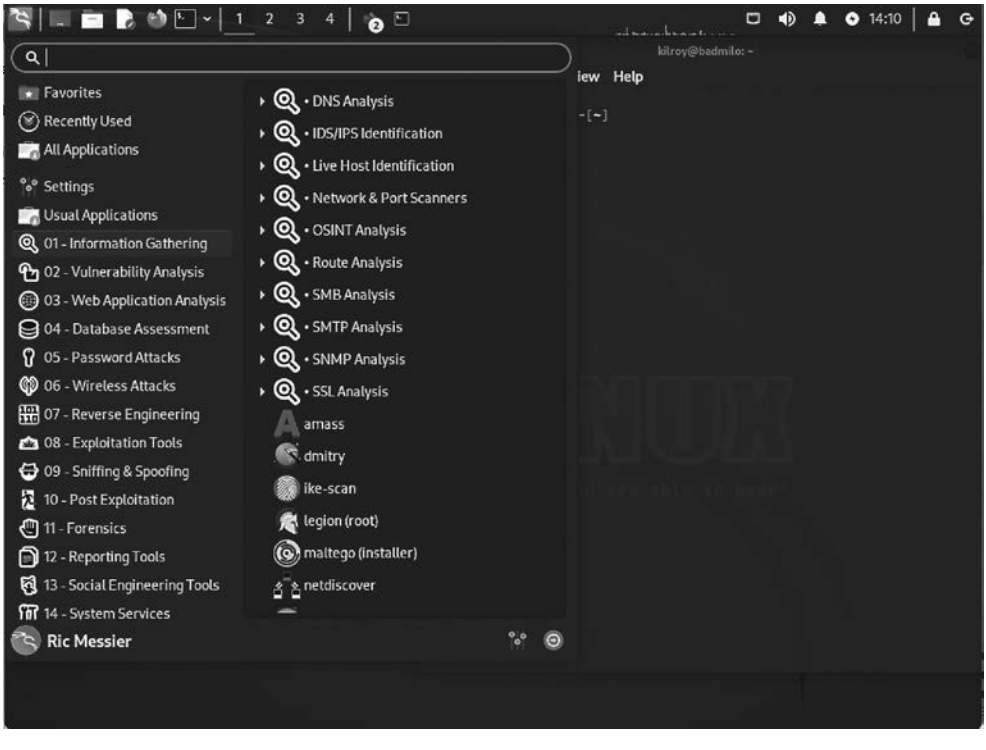

Рис. 1.5. Рабочий стол Xfce с меню приложений