ХАКИНГ [ЛЕКЦИЯ 5]

ОСВОЕНИЕ KALI LINUX ДЛЯ ХАКИНГА

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.

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/sda1), 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 ls -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*.

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.

Командная строка Linux

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*).

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.

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:

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).

```
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:~#
```

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.

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.(dpkg).
- /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.

Полезные команды Linux

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!
```

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.

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)).*

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 stickybit

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.

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).

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.

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.

Основные команды Linux. Нужно знать как таблицу умножения!

Привилегии

sudo command — запустить команду как **root**
sudo -s — открыть оболочку **root**
sudo -s -u user — открыть оболочку как пользователь
sudo -k — восстановить пароль **sudo**
gksudo command — визуальный диалог **sudo (GNOME)**
kdesudo command — визуальный диалог **sudo (KDE)**
sudo visudo — редактировать **/etc/sudoers**
gksudo nautilus — корневой файловый менеджер **(GNOME)**
kdesudo konqueror — корневой файловый менеджер **(KDE)**
passwd — изменить ваш пароль

Сеть

ifconfig — показать информацию о сети
iwconfig — показать информацию о беспроводной сети
sudo iwlist scan — поиск беспроводных сетей
sudo /etc/init.d/networking restart — перезапустить сеть
/etc/network/interfaces — файл для ручной настройки сети
ifup interface — включить интерфейс
ifdown interface — отключить интерфейс

ping host - пропинговать **host** и вывести результат
whois domain - получить информацию **whois** для **domain**
wget file - скачать **file**
ifconfig eth0 - показать конфигурацию сетевого интерфейса **eth0**
ifup eth0 - активировать интерфейс **eth0**
ifdown eth0 - деактивировать интерфейс **eth0**
ifconfig eth0 192.168.1.1 netmask 255.255.255.0 - выставить интерфейсу eth0 ip-адрес и маску подсети

ifconfig eth0 promisc - перевести интерфейс **eth0** в **promiscuous**-режим для "отлова" пакетов **(sniffing)**

ifconfig eth0 -promisc - отключить promiscuous-режим на интерфейсе **eth0**

dhclient eth0 - активировать интерфейс **eth0** в **dhcp**-режиме.

route -n - вывести локальную таблицу маршрутизации
route add -net 0/0 gw IP_Gateway - задать **ip**-адрес шлюза по умолчанию **(default gateway)**

route add -net 192.168.0.0 netmask 255.255.0.0 gw 192.168.1.1 - добавить статический маршрут в сеть **192.168.0.0/16** через шлюз с **ip**-адресом **192.168.1.1**

route del 0/0 gw IP_gateway - удалить **ip**-адрес шлюза по умолчанию **(default gateway)**

echo "1" > /proc/sys/net/ipv4/ip_forward - разрешить пересылку пакетов **(forwarding)**

hostname - отобразить имя компьютера
ip link show - отобразить состояние всех интерфейсов
mii-tool eth0 - отобразить статус и тип соединения для интерфейса **eth0**
ethtool eth0 - отображает статистику интерфейса **eth0** с выводом такой информации, как поддерживаемые и текущие режимы соединения

netstat-tupn - отображает все установленные сетевые соединения по протоколам TCP и UDP без разрешения имён в ip-адреса и PID'ы и имена процессов, обеспечивающих эти соединения

netstat-tupln - отображает все сетевые соединения по протоколам **TCP** и **UDP** без разрешения имён в **ip**-адреса и **PID**'ы и имена процессов, слушающих порты

tcpdump tcp port 80 - отобразить весь трафик на **TCP**-порт **80** (обычно - **HTTP**)

iwlist scan - просканировать эфир на предмет, доступности беспроводных точек доступа

iwconfig eth1 - показать конфигурацию беспроводного сетевого интерфейса **eth1**

cat /proc/net/dev - показать сетевые интерфейсы и статистику по ним
dig domain – получить **DNS** информацию domain

Дисплей

sudo /etc/init.d/gdm restart — перезапустить **X** и вернуться к авторизации (**GNOME**)
sudo /etc/init.d/kdm restart — перезапустить **X** и вернуться к авторизации (**KDE**)

/etc/X11/xorg.conf — файл настроек экрана
sudo dexconf — сбросить конфигурацию **xorg.conf**
Ctrl+Alt+Bksp — перезапустить **X**-сервер, если завис
Ctrl+Alt+FN — переключиться на интерфейс командной строки

Ctrl+Alt+F7 — переключиться обратно на графический интерфейс пользователя

Системные службы

start service — начать работу службы (**Upstart**)
stop service — остановить работу службы (**Upstart**)
status service — проверить, запущена ли служба (**Upstart**)
/etc/init.d/service start — запустить службу (**SysV**)
/etc/init.d/service stop — остановить службу (**SysV**)
/etc/init.d/service status — проверить статус службы (**SysV**)

/etc/init.d/service restart — перезапустить службу (**SysV**)
runlevel — получить текущий уровень запуска

Брандмауэр

ufw enable — включить брандмауэр
ufw disable — выключить брандмауэр
ufw default allow — разрешить все соединения по умолчанию
ufw default deny — запретить все соединения по умолчанию
ufw status — текущий статус и правила
ufw allow port — разрешить трафик на порт
ufw deny port — заблокировать порт
ufw deny from ip — заблокировать **IP**-адрес

Управление пакетами

apt-get update — обновить доступные обновления
apt-get upgrade — обновить все пакеты
apt-get dist-upgrade — обновить версию **Ubuntu**
apt-get install pkg — установить пакет (**pkg**)
apt-get purge pkg — удалить пакет (**pkg**)
apt-get autoremove — удалить устаревшие пакеты
apt-get -f install — попробовать исправить битые пакеты
dpkg —configure -a — попробовать исправить битые пакеты
dpkg -i pkg.deb — установить файл **pkg.deb**
/etc/apt/sources.list — файл со списком **APT** репозиториев

Имена приложений

nautilus — файловый менеджер (**GNOME**)
dolphin — файловый менеджер (**KDE**)
konqueror — веб-браузер (**KDE**)
kate — текстовый редактор (**KDE**)
gedit — текстовый редактор (**GNOME**)

Система

Восстановление — нажмите и удерживайте **Alt+SysRq (PrintScrn)**, затем с паузами в одну секунду, нажимайте клавиши **R, E, I, S, U, B** для безопасной перезагрузки системы
lsb_release -a — получить версию ОС
uname -r — получить версию ядра
uname -a — получить всю информацию о ядре

Системная информация

arch - отобразить архитектуру компьютера
cat /proc/cpuinfo - показать информацию о ЦПУ
cat /proc/meminfo - проверить использование памяти
df - информация об использовании дисков
hdparm -i /dev/hda - вывести характеристики жесткого диска
lspci -tv - показать в виде дерева PCI устройства
lsusb -tv - показать в виде дерева USB устройства

uptime - показать время работы с момента включения
uname -a - показать информацию о ядре
clock -w - сохранить системное время в BIOS
shutdown -h now - Остановить систему
shutdown -r now - перегрузить систему
logout - выйти из системы

Файловые команды

cd /home - перейти в директорию '**/home**'
cd .. - перейти в директорию уровнем выше
cd ../.. - перейти в директорию двумя уровнями выше
cd - перейти в домашнюю директорию
cd ~user - перейти в домашнюю директорию пользователя **user**
cd - - перейти в директорию, в которой находились до перехода в текущую директорию
pwd - показать текущюю директорию
ls - отобразить содержимое текущей директории
ls -F - отобразить содержимое текущей директории с добавлением к именам символов, храктеризующих тип
ls -l - показать детализированое представление файлов и директорий в текущей директории
ls -a - показать скрытые файлы и директории в текущей директории
ls *[0-9]* - показать файлы и директории содержащие в имени цифры
tree - показать дерево файлов и директорий, начиная от корня **(/)**
mkdir dir1 - создать директорию с именем '**dir1**'
mkdir dir1 dir2 - создать две директории одновременно
mkdir -p /tmp/dir1/dir2 - создать дерево директорий
rm -f file1 - удалить файл с именем '**file1**'
rmdir dir1 - удалить директорию с именем '**dir1**'
rm -rf dir1 - удалить директорию с именем '**dir1**' и рекурсивно всё её содержимое
rm -rf dir1 dir2 - удалить две директории и рекурсивно их содержимое
mv dir1 new_dir - переименовать или переместить файл или директорию
cp file1 file2 - сопировать файл file1 в файл **file2**
cp dir/* - копировать все файлы директории **dir** в текущую директорию
cp -a /tmp/dir1 - копировать директорию **dir1** со всем содержимым в текущую директорию
cp -a dir1 dir2 - копировать директорию **dir1** в директорию **dir2**

Пользователи и группы

whoami - имя, под которым вы залогинены
groupadd group_name - создать новую группу с именем group_name
groupdel group_name - удалить группу group_name

groupmod -n new_group_name old_group_name - переименовать группу **old_group_name** в **new_group_name**

useradd -c "Nome Cognome" -g admin -d /home/user1 -s /bin/bash user1 - создать пользователя **user1**, назначить ему в качестве домашнего каталога **/home/user1**, в качестве **shell'a /bin/bash**, включить его в группу **admin** и добавить комментарий **Nome Cognome.**

useradd user1 - создать пользователя **user1**
userdel -r user1 - удалить пользователя **user1** и его домашний каталог

usermod -c "User FTP" -g system -d /ftp/user1 -s /bin/nologin user1 - изменить атрибуты пользователя

passwd - сменить пароль
passwd user1 - сменить пароль пользователя user1 (только root)
chage -E 2005-12-31 user1 - установить дату окончания действия учётной записи пользователя **user1**
pwck - проверить корректность системных файлов учётных записей. Проверяются файлы **/etc/passwd** и **/etc/shadow**

grpck - проверяет корректность системных файлов учётных записей. Проверяется файл **/etc/group**

newgrp [-] group_name - изменяет первичную группу текущего пользователя. Если указать "-", ситуация будет идентичной той, в которой пользователь вышил из системы и снова вошёл. Если не указывать группу, первичная группа будет назначена из **/etc/passwd.**

Установка пакетов

apt-get istall application_name - установить приложение application_name

Установка из исходников:

```
./configure
make
make install
```

dpkg -i pkg.deb - установить пакет (Debian)

Привилегированный запуск приложений

sudo <command> - запуск команды под именем привилегированного пользователя
gksu <command> - тоже самое, разница в том что появляется графическое окно с просьбой ввести пароль в обоих случаях вводится пароль вашего текущего пользователя.