# Содержание

*Моей любимой жене, которая была рядом со мной на протяжении всего пути. Без нее было бы невозможно завершить этот проект. Моим родителям и родственникам за их постоянную поддержку и мотивацию. Моему псу за то, что бодрствовал вместе со мной во время бессонных ночей*

# Соавторы

## Об авторе

Монаппа К. А. работает в Cisco Systems в качестве следователя по вопросам информационной безопасности и занимается аналитикой угроз и расследованием целевых кибератак. Он является членом наблюдательного совета Black Hat, создателем изолированной среды Limon Linux, победителем конкурса плагинов Volatility 2016 и соучредителем исследовательского сообщества Cysinfo по кибербезопасности. Он представлял и проводил учебные занятия на различных конференциях по безопасности, включая Black Hat, FIRST, OPCDE и DSCI, регулярно проводит тренинги на Конференции по безопасности Black Hat в США, Азии и Европе.

*Я хотел бы выразить свою благодарность Дэниелу Катберту (Daniel Cuthbert) и доктору Майклу Шпрайценбарту (Michael Spreitzenbarth) за то, что они нашли время в своем плотном графике, чтобы рецензировать книгу. Благодарю Шарон Радж (Sharon Raj), Прашанта Чаудхари (Prashant Chaudhari), Шрилеху Инани (Shrilekha Inani) и остальную часть команды Packt за их поддержку. Спасибо Майклу Шеку (Michael Scheck), Крису Фраю (Chris Fry), Скотту Хайдеру (Scott Heider) и моим сотрудникам из Cisco CSIRT за их поддержку. Спасибо Майклу Хейлу Лайю (Michael Hale Ligh), Эндрю Кейсу (Andrew Case), Джейми Леви (Jamie Levy), Аарону Уолтерсу (Aaron Walters), Мэтту Суич (Matt Suiche), Ильфаку Гильфанову (Ilfak Guilfanov) и Ленни Зельцеру (Lenny Zeltser), которые вдохновляли и мотивировали меня своей работой. Благодарю Саджана Шетти (Sajan Shetty), Виджая Шарму (Vijay Sharm), Гэвина Рейда (Gavin Reid), Леви Гундерта (Levi Gundert), Джоанну Кретович (Joanna Kretowicz), Марту Стрелец (Marta Strzelec), Венкатеша Мурти (Venkatesh Murthy), Амита Малика (Amit Malik) и Эшвина Патила (Ashwin Patil) за их бесконечную поддержку. Спасибо авторам других книг, веб-сайтов, блогов и инструментального ПО, которые внесли свой вклад в мои знания, а следовательно, и в эту книгу.*

# О РЕЦЕНЗЕНТАХ

Дэниел Катберт (Daniel Cuthbert) – глава отдела исследований безопасности в Банко Сантандер.

За свою более чем 20-летнюю карьеру как в атакующей, так и в оборонительной сферах он был свидетелем эволюции взлома от небольших групп пытливых умов до организованных криминальных сетей и национальных государств, которые мы видим сегодня. Он заседает в наблюдательном совете Black Hat и является соавтором Руководства по тестированию OWASP (2003) и Стандарта проверки безопасности приложений OWASP (ASVS).

Доктор Майкл Шпрайценбарт (Michael Spreitzenbarth) работает фрилансером в секторе информационной безопасности уже несколько лет после защиты дипломной работы, основной темой которой была мобильная криминалистика. В 2013 году он защитил кандидатскую диссертацию в области криминалистического анализа устройств, работающих под управлением Android, и анализа мобильных вредоносных программ. Затем он начал работать в международной компьютерной группе реагирования на чрезвычайные ситуации и во внутренней группе белых хакеров. Он ежедневно занимается безопасностью мобильных систем, криминалистическим анализом смартфонов и подозрительных мобильных приложений, а также расследованием инцидентов, связанных с безопасностью и симуляцией кибератак.

# Предисловие

Развитие компьютерных и интернет-технологий изменило наши жизни и коренным образом изменило способ ведения бизнеса организациями. Тем не менее развитие технологий и цифровизация вызывали рост киберпреступности. Растущая угроза кибератак на критическую инфраструктуру, дата-центры, а также частный/общественный, оборонный, энергетический, государственный и финансовый секторы бросает невиданный вызов каждому, начиная от отдельного человека и заканчивая крупными корпорациями. В ходе этих кибератак используются вредоносные программы с целью финансовых краж, шпионажа, саботажа, кражи интеллектуальной собственности и по политическим мотивам.

По мере того как противники становятся все более изощренными и внедряют атаки с использованием сложного вредоносного ПО, обнаружение таких вторжений и реагирование на них имеет решающее значение для специалистов по кибербезопасности. Анализ вредоносных программ стал обязательным навыком для борьбы со сложным вредоносным ПО и целевыми атаками. Анализ вредоносных программ требует сбалансированного владения различными навыками и темами. Другими словами, он требует времени и терпения.

Эта книга учит концепциям, инструментам и методам, чтобы понять поведение и характеристики вредоносных программ Windows, используя анализ вредоносного ПО. Эта книга начинается со знакомства с основными понятиями анализа вредоносных программ и постепенно углубляется в более продвинутые концепции анализа кода и анализа дампа памяти.

Для лучшего восприятия в примерах данной книги используются различные реальные образцы вредоносного ПО, зараженные образы памяти и визуальные диаграммы. В дополнение к этому дается достаточно информации, чтобы помочь вам понять необходимые принципы, и, где это возможно, приводятся ссылки на дополнительные ресурсы для дальнейшего чтения.

Если вы новичок в области анализа вредоносных программ, эта книга должна помочь вам, или, если у вас уже есть опыт в этой области, эта книга дополнит ваши знания. Изучаете ли вы анализ вредоносных программ для криминалистического расследования, чтобы отреагировать на компьютерный инцидент, или просто чтобы развлечься, эта книга позволит вам достичь цели.

## Для кого эта книга

Если вы член группы реагирования на компьютерные инциденты, эксперт по кибербезопасности, системный администратор, аналитик вредоносных программ, судебно-медицинский эксперт, студент или любопытный специалист по безопасности, который интересуется анализом вредоносных программ или хочет расширить свои познания в этой области, то эта книга для вас.

## Что рассматривается в этой книге

Глава 1 *«Введение в анализ вредоносных программ»* знакомит читателей с концепцией анализа вредоносного ПО, типами анализа вредоносного ПО и настройкой изолированной тестовой среды для анализа вредоносного ПО.

Глава 2 *«Статический анализ»* обучает инструментам и методам извлечения информации о метаданных из вредоносного двоичного файла. Показывает, как сравнивать и классифицировать образцы вредоносных программ. Вы узнаете, как определить различные аспекты двоичного файла без его выполнения.

Глава 3 «Динамический анализ» обучает инструментам и методам определения поведения вредоносного ПО и показывает его взаимодействие с системой. Вы узнаете, как получить сетевые и хостовые индикаторы, связанные с вредоносным ПО.

Глава 4 *«Язык ассемблера и дизассемблирование для начинающих»* дает основное понимание языка ассемблера и учит необходимым навыкам для выполнения анализа кода.

Глава 5 *«Дизассемблирование с использованием IDA»* описывает свойства дизассемблера IDA Pro, вы узнаете, как использовать IDA Pro для статического анализа кода (дизассемблирование).

Глава 6 *«Отладка вредоносных двоичных файлов»* обучает технике отладки двоичного файла с использованием x64dbg и отладчика IDA Pro. Вы узнаете, как использовать отладчик, чтобы контролировать выполнение программы и манипулировать её поведением.

Глава 7 *«Функциональные возможности и персистенция вредоносных программ»* описывает различные функциональные возможности вредоносных программ с использованием реверс-инжиниринга. В этой главе также рассматриваются различные методы персистенции, используемые вредоносными программами.

Глава 8 *«Внедрение кода и перехват»* рассказывает о распространенных методах внедрения кода, используемых вредоносными программами для выполнения вредоносного кода в контексте доверенного процесса. В этой главе также описываются методы подключения, используемые вредоносной программой для передачи контроля вредоносному коду для мониторинга, блокировки или выходных данных API. Вы узнаете, как анализировать вредоносные программы, использующие методы внедрения кода и перехвата.

Глава 9 *«Методы обфускации вредоносных программ»* рассказывает о кодировании и методах упаковки, используемых вредоносными программами для сокрытия информации. Данная глава обучает различным стратегиям декодирования/дешифрования данных и распаковки вредоносного бинарного файла.

Глава 10 *«Охота на вредоносные программы с использованием криминалистического анализа дампов памяти»* рассказывает, как обнаруживать вредоносные компоненты, используя криминалистический анализ дампа памяти. Вы познакомитесь с различными плагинами Volatility для обнаружения и идентификации артефактов форензики в памяти.

Глава 11 *«Обнаружение сложных вредоносных программ с использованием анализа дампа памяти»* рассказывает о скрытых методах, используемых сложными вредоносными программами для сокрытия от инструментов форензики. Вы научитесь исследовать и распознавать руткиты в пользовательском режиме и режиме ядра.

## Чтобы получить максимальную отдачу от этой книги

Знание языков программирования, таких как C и Python, будет полезно (особенно для понимания концепций, изложенных в главах 5, 6, 7, 8 и 9). Если вы написали несколько строк кода и имеете общее представление о концепциях программирования, то сможете получить максимальную отдачу от этой книги.

Если у вас нет знаний в области программирования, вы все равно сможете получить основные принципы анализа вредоносного ПО, описанные в главах 1, 2 и 3. Однако вам может оказаться немного трудно понять концепции, изложенные в остальных частях. Чтобы вы ускорились, на этот случай достаточно информации и дополнительных ресурсов есть в каждой главе. Возможно, вам придется прочитать дополнительную литературу, чтобы полностью понять эти принципы.

## Скачать цветные изображения

Мы также предоставляем PDF-файл с цветными изображениями скриншотов/диаграмм, используемых в этой книге. Вы можете скачать его здесь: www.packtpub.com/sites/default/files/downloads/LearningMalwareAnalysis_ColorImages.pdf.

## Используемые условные обозначения

В этой книге используется ряд текстовых обозначений.

`Моноширинный шрифт` используется для примеров кода, имен папок, имен файлов, ключа реестра и значений, расширения файлов, путей, фиктивных URL, пользовательского ввода, имен функций и имен пользователей в Твиттере. Например: «Смонтируйте скачанный файл образа диска `WebStorm-10*.dmg` как еще один диск в вашей системе».

Любой ввод командной строки выделен **полужирным шрифтом**, и пример выглядит следующим образом:

```
$ sudo inetsim
INetSim 1.2.6 (2016-08-29) by Matthias Eckert & Thomas Hungenberg
Using log directory: /var/log/inetsim/
Using data directory: /var/lib/inetsim/
```

Когда мы хотим обратить ваше внимание на определенную часть кода или вывода, соответствующие строки или элементы выделены **полужирным шрифтом**:

```
$ python vol.py -f tdl3.vmem --profile=WinXPSP3x86 ldrmodules -p 880
Volatility Foundation Volatility Framework 2.6
Pid Process Base InLoad InInit InMem MappedPath
--- ----------- -------- ----- ------- ----- --------------------------
880 svchost.exe 0x10000000 False False False \WINDOWS\system32\TDSSoiqh.dll
880 svchost.exe 0x01000000 True  False True  \WINDOWS\system32\svchost.exe
880 svchost.exe 0x76d30000 True  True  True  \WINDOWS\system32\wmi.dll
880 svchost.exe 0x76f60000 True  True  True  \WINDOWS\system32\wldap32.dll
```

*Курсив*: используется для нового термина, важного слова или слов, названия вредоносного ПО.

**Текст на экране**: слова в меню или диалоговых окнах появляются в тексте следующим образом. Например: Выберите **Системная информация на панели администрирования**.

Предупреждения или важные заметки выглядят так.

Советы и подсказки выглядят так.

# Глава 1

# Введение в анализ вредоносных программ

Количество кибератак, несомненно, растет, нацеливаясь на правительственный, военный, государственный и частный секторы. Эти кибератаки направлены на физических лиц или организации, стремясь извлечь ценную информацию. Иногда они якобы связаны с киберпреступностью или группами, финансируемыми государством, но могут также выполняться отдельными группами для достижения своих целей. В ходе большинства этих кибератак используется вредоносное программное обеспечение (также называемое вредоносные программы) для заражения своих потенциальных жертв. Знания, навыки и инструменты, необходимые для анализа вредоносных программ, нужны для обнаружения, расследования и защиты от таких атак.

Из этой главы вы узнаете:

- что означает вредоносное ПО и какова его роль в кибератаках;
- об анализе вредоносных программ и его значении в компьютерной криминалистике;
- о различных видах анализа вредоносных программ;
- о настройке тестовой среды;
- о различных источниках для получения образцов вредоносных программ.

## 1.1 Что такое вредоносное ПО?

Вредоносное ПО – это код, который выполняет вредоносные действия; он может принять форму исполняемого файла, скрипта, кода или любого другого программного обеспечения. Злоумышленники используют вредоносное ПО для кражи конфиденциальной информации, чтобы шпионить за зараженной системой или с целью взять систему под контроль.

Обычно оно попадает в вашу систему без вашего согласия и может быть доставлено через различные каналы связи, такие как электронная почта, интернет или USB-накопители.

Ниже приведены некоторые вредоносные действия, выполняемые вредоносными программами:

- нарушение работы компьютера;
- кража конфиденциальной информации, в том числе личных, деловых и финансовых данных;
- несанкционированный доступ к системе жертвы;
- шпионаж;
- отправка спам-писем;
- участие в распределенных атаках типа «отказ в обслуживании» (DDOS);
- блокировка файлов на компьютере и удержание их с целью выкупа.

Вредоносное ПО – это широкий термин, относящийся к различным типам вредоносных программ, таким как вирусы, черви и руткиты. Выполняя анализ вредоносных программ, вы часто будете сталкиваться с различными типами вредоносных программ; некоторые из них классифицируются в зависимости от их функциональности и векторов атаки, как указано далее.

- **Вирус или червь**: вредоносное ПО, способное копировать себя и распространять на другие компьютеры. Вирус требует вмешательства пользователя, тогда как червь может распространяться без вмешательства пользователя.
- **Троян**: вредоносная программа, которая маскируется под обычную программу, чтобы обманом заставить пользователей установить её на своих системах. После установки она может выполнять вредоносные действия, такие как кража конфиденциальных данных, загрузка файлов на сервер злоумышленника или мониторинг веб-камер.
- **Бэкдор/троян удаленного доступа (RAT):** это тип троянца, который позволяет злоумышленнику получить доступ и выполнить команды во взломанной системе.
- **Рекламное ПО**: вредоносное ПО, которое показывает нежелательные рекламные объявления пользователю. Оно обычно доставляется с помощью бесплатных загрузок и может принудительно установить программное обеспечение на вашей системе.
- **Ботнет**: это группа компьютеров, зараженных одним и тем же вредоносным ПО (называемых ботами), ожидающих получения инструкций от командно-контрольного сервера, контролируемого злоумышленником. Затем злоумышленник может передать команду этим ботам, которые могут выполнять вредоносные действия, такие как DDOS-атаки или рассылка спам-писем.
- **Похититель информации**: вредоносное ПО, предназначенное для кражи конфиденциальных данных, таких как банковские учетные данные, или нажатия клавиш из зараженной системы. Некоторые примеры этих вредоносных программ включают в себя кейлогеры, шпионское ПО, снифферы и формграбберы.

- **Вирус-вымогатель**: вредоносная программа, которая удерживает систему с целью выкупа, блокируя пользователей из своего компьютера или путем шифрования своих файлов.
- **Руткит**: вредоносная программа, предоставляющая злоумышленнику привилегированный доступ к зараженной системе, скрывает свое наличие или наличие другого программного обеспечения.
- **Загрузчик или дроппер**: вредоносное ПО, предназначенное для загрузки или установки дополнительных вредоносных компонентов.

Удобный ресурс для понимания терминологии и определений вредоносного ПО доступен по адресу Blog.malwarebytes.com/glossary/.

Классификация вредоносных программ на основе функциональности не всегда возможна, потому что одна вредоносная программа может содержать несколько функций, которые могут вылиться во множество категорий, упомянутых только что. Например, вредоносное ПО может включать в себя компонент червя, который сканирует сеть в поисках уязвимых систем и может применить другой вредоносный компонент, такой как бэкдор или вирус-вымогатель при успешной эксплуатации. Классификация вредоносных программ также может быть проведена, основываясь на мотивах злоумышленника.

Например, если вредоносное ПО используется для кражи личной, коммерческой или патентованной информации для получения прибыли, то вредоносные программы могут быть классифицированы как криминальное программное обеспечение или товарное вредоносное ПО. Если вредоносная программа нацелена на конкретную организацию или отрасль промышленности, чтобы украсть информацию / собрать разведданные с целью шпионажа, тогда это может быть классифицировано как целевое или шпионское вредоносное ПО.

## 1.2 Что такое анализ вредоносных программ?

Анализ вредоносных программ – это изучение поведения вредоносного ПО. Цель анализа вредоносного ПО – понять работу вредоносных программ и методы их обнаружения и устранения. Он включает в себя анализ подозрительного двоичного файла в безопасной среде для определения его характеристик и функциональных возможностей, чтобы можно было выстроить лучшую оборонительную стратегию для защиты сети организации.

## 1.3 Почему анализ вредоносных программ?

Основным мотивом проведения анализа вредоносных программ является извлечение информации из образца вредоносного ПО, которая может помочь в реагировании на вредоносный инцидент. Целью анализа вредоносных программ является определение возможностей вредоносного ПО, его обнаружение и содержание. Это также помогает в определении идентифицируемых

моделей, которые могут быть использованы для лечения и предотвращения будущих инфекций. Вот некоторые из причин, почему вы будете выполнять анализ вредоносных программ:

- чтобы определить характер и назначение вредоносного ПО. Например, это может помочь определить, является ли вредоносное ПО средством для кражи информации, HTTP-ботом, спам-ботом, руткитом, кейлоггером или RAT и т. д.;
- чтобы получить представление о том, как система была взломана и каковы последствия;
- для выявления сетевых индикаторов, связанных с вредоносным ПО, которые могут затем быть использованы для обнаружения аналогичных инфекций с помощью сетевого мониторинга. Например, во время анализа, если вы определите, что вредоносная программа связывается с конкретным доменным/IP-адресом, вы можете использовать этот доменный/IP-адрес для создания подписи и отслеживать сетевой трафик для идентификации всех хостов, связавшись с этим доменным/IP-адресом;
- чтобы извлечь хостовые индикаторы, такие как имена файлов и ключи реестра, которые, в свою очередь, могут быть использованы для определения аналогичной инфекции с использованием хостового мониторинга. Например, если вы узнаете, что вредоносная программа создает раздел реестра, вы можете использовать этот ключ реестра в качестве индикатора для создания подписи или сканирования вашей сети, чтобы определить хосты, которые имеют одинаковый раздел реестра;
- определить намерение и мотив злоумышленника. Например, во время вашего анализа, если вы обнаружите, что вредоносное ПО крадет банковские учетные данные, то можно сделать вывод, что мотив злоумышленника – денежная выгода.

Группы, занимающиеся анализом угроз, очень часто используют индикаторы, установленные на основе анализа вредоносных программ, чтобы классифицировать атаку, и приписывают их известным угрозам. Анализ вредоносных программ может помочь вам получить информацию о том, кто может стоять за атакой (конкурент, спонсируемая государством группа и т. д.).

## 1.4 Типы анализа вредоносных программ

Чтобы понять работу и характеристики вредоносного ПО и оценить его влияние на систему, вы будете часто использовать различные методы анализа. Ниже приводится классификация этих методов.

- **Статический анализ**: это процесс анализа двоичного файла без его выполнения. Его проще всего осуществить, и он позволяет извлечь метаданные, связанные с подозрительным двоичным файлом. Статический анализ может не выявить всех необходимых сведений, но иногда может предоставить интересную информацию, которая помогает сосредо-

точить ваши последующие усилия по анализу. В главе 2 «Статический анализ» рассказывается об инструментальных средствах и методах извлечения полезной информации из вредоносного бинарного кода с использованием статического анализа.

- **Динамический анализ (поведенческий анализ)**: это процесс выполнения подозрительного бинарного файла в изолированной среде и отслеживание его поведения. Этот метод анализа прост в выполнении и дает ценную информацию о деятельности двоичного файла при его выполнении. Этот метод полезен, но не раскрывает всех функциональных возможностей враждебной программы. В главе «Динамический анализ» рассказывается об инструментальных средствах и методах определения поведения вредоносного ПО с использованием динамического анализа.
- **Анализ кода**: это продвинутый метод, который фокусируется на анализе кода, чтобы понять внутреннюю работу файла. Он раскрывает информацию, которую невозможно выявить только в ходе статического и динамического анализа. Анализ кода далее делится на статический и динамический. Статический анализ кода включает в себя разборку подозрительного двоичного файла и визуальный просмотр кода, чтобы понять программу поведения, тогда как динамический анализ кода включает в себя отладку подозрительного двоичного файла в контролируемой форме, чтобы понять его функциональность. Анализ кода требует понимания языка программирования и концепций операционной системы. Предстоящие главы (главы с 4 по 9) содержат сведения об инструментальных средствах и методах, необходимых для выполнения анализа кода.
- **Анализ памяти (криминалистика памяти)**: это метод анализа оперативной памяти компьютера для артефактов форензики. Обычно это метод компьютерной криминалистики, но включение его в ваш анализ вредоносных программ поможет получить представление о поведении вредоносных программ после заражения. Анализ памяти особенно полезен, чтобы выявить уловки и хитрости вредоносного ПО. Вы узнаете, как выполнить анализ памяти, в последующих главах (главы 10 и 11).

Интеграция различных методов при выполнении анализа вредоносных программ может выявить множество контекстной информации, которая окажется полезной для вашего расследования.

## 1.5 Настройка тестовой среды

Анализ враждебной программы требует безопасной и надежной тестовой среды. Ведь вы же не хотите заразить свою систему или систему компании. Тестовая среда для работы с вредоносным ПО может быть очень простой или сложной в зависимости от доступных вам ресурсов (оборудование, программное обеспечение для виртуализации, лицензия Windows и т. д.). Этот раздел

поможет вам настроить простую среду на одной физической системе, состоящей из виртуальных машин (ВМ). Если вы хотите создать похожую среду, следуйте изложенным далее инструкциям или перейдите к следующему разделу (раздел 6 «Источники вредоносного ПО»).

### 1.5.1 Требования к среде

Прежде чем приступить к настройке тестовой среды, вам потребуется несколько компонентов: физическая система под управлением базовой операционной системы Linux, Windows или macOS X с программным обеспечением для виртуализации (например, VMware или VirtualBox). При анализе вредоносного ПО вы будете выполнять его на виртуальной машине на базе Windows (Windows VM). Преимущество использования виртуальной машины заключается в том, что, завершив анализ вредоносного ПО, вы можете вернуть её в чистое состояние.

VMware Workstation для Windows и Linux доступна для скачивания на странице www.vmware.com/products/workstation/workstation-valuation.html, а VMware Fusion для MacOS X доступна для загрузки по адресу www.vmware.com/products/fusion/fusion-evaluation.html. VirtualBox для различных операционных систем доступна для скачивания на странице www.virtualbox.org/wiki/Downloads. Чтобы создать безопасную тестовую среду, вы должны принять необходимые меры предосторожности, дабы избежать утечки вредоносных программ из виртуальной среды и заражения вашей физической (хост-) системы. Ниже приведено несколько моментов, которые следует помнить при настройке виртуализированной среды. Постоянно обновляйте программное обеспечение для виртуализации. Это необходимо, потому что вредоносные программы могут использовать уязвимость в программном обеспечении. Они могут вырваться из виртуальной среды и заразить вашу хост-систему.

Установите свежую копию операционной системы внутри виртуальной машины (ВМ) и не храните там конфиденциальную информацию. При анализе вредоносного ПО, если вы не хотите, чтобы оно получило доступ к интернету, вы должны рассмотреть возможность использования режима конфигурации сети host-only или ограничить сетевой трафик в вашей среде с использованием моделированных служб. Не подключайте съемные носители, которые впоследствии могут быть использованы на физических машинах, такие как USB-накопители.

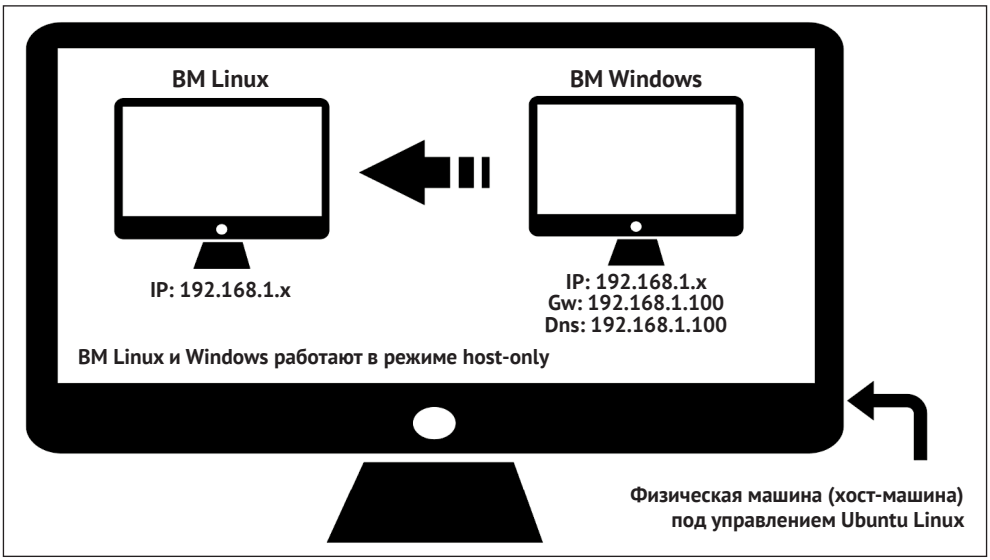
Поскольку вы будете анализировать вредоносные программы Windows (обычно исполняемые или DLL), рекомендуется выбирать базовую операционную систему, такую как Linux или macOS X, для вашего хост-устройства вместо Windows. Потому что даже если вредоносная программа покинет виртуальную машину, она все равно не сможет заразить главный компьютер.

### 1.5.2 Обзор архитектуры тестовой среды

Архитектура среды, которую я буду использовать на протяжении всей книги, состоит из физической машины (называемой хост-машиной) под управлением

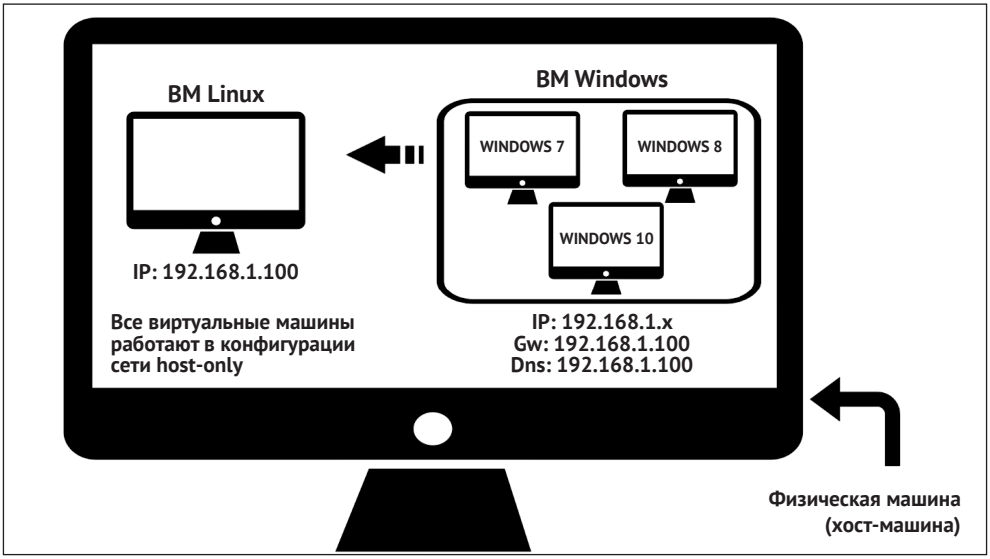
Ubuntu Linux с экземплярами виртуальной машины Linux (Ubuntu Linux VM) и виртуальной машины Windows (Windows VM). Эти виртуальные машины будут настроены так, чтобы быть частью одной сети и использовать режим конфигурации сети host-only, дабы вредоносной программе не разрешалось выходить в интернет и сетевой трафик содержался в изолированной тестовой среде. Виртуальная машина Windows – место, где вредоносная программа будет выполняться во время анализа, а виртуальная машина Linux используется для мониторинга сетевого трафика и будет настроена на моделирование интернет-сервисов (DNS, HTTP и т. д.), чтобы обеспечить надлежащий ответ, когда вредоносная программа будет запрашивать их. Например, виртуальная машина Linux будет настроена таким образом, что когда вредоносная программа будет запрашивать такой сервис, как DNS, Linux VM предоставит правильный ответ DNS. Глава 3 «Динамический анализ» детально описывает эту концепцию.

На следующем рисунке показан пример архитектуры простой среды, которую я буду использовать в этой книге. В этой настройке виртуальная машина Linux будет предварительно настроена на IP-адрес `192.168.1.100`, а IP-адрес виртуальной машины Windows будет установлен на `192.168.1.x` (где x – любое число от `1` до `254`, кроме `100`). Шлюз по умолчанию и DNS виртуальной машины Windows будет настроен на IP-адрес виртуальной машины Linux (`192.168.1.100`), так что весь сетевой трафик Windows будет направляться через виртуальную машину Linux. Следующий раздел поможет вам настроить виртуальную машину Linux и виртуальную машину Windows, чтобы соответствовать вышеприведенным параметрам.



Вам не нужно ограничивать себя архитектурой, показанной на предыдущем рисунке; возможны разные конфигурации, и не представляется возможным предоставить инструкции по всем возможным вариантам. В этой книге я покажу вам, как настроить и использовать архитектуру среды, показанную на предыдущем рисунке.

Также можно создать среду, состоящую из нескольких виртуальных машин, работающих на разных версиях Windows; это позволит вам проанализировать образец вредоносного ПО на различных версиях операционных систем Windows. Пример конфигурации с несколькими виртуальными машинами Windows будет выглядеть так же, как показано на следующей диаграмме:




## 1.5.3 Установка и настройка виртуальной машины Linux

Для настройки виртуальной машины Linux я буду использовать дистрибутив Linux Ubuntu 16.04.2 (releases.ubuntu.com/16.04). Я выбрал Ubuntu по той причине, что большинство инструментов, описанных в этой книге, либо уже предустановлено, либо доступно через apt-get менеджер пакетов. Ниже приведена пошаговая процедура настройки Ubuntu 16.04.2 LTS на VMware и VirtualBox.

Не стесняйтесь следовать приведенным здесь инструкциям в зависимости от программного обеспечения для виртуализации (VMware или VirtualBox), установленного на вашей системе. Если вы незнакомы с установкой и настройкой виртуальных машин, обратитесь к руководству VMware по адресу pubs.vmware.com/workstation-12/topic/com.vmware.ICbase/PDF/workstation-pro-12-user-guide.pdf или руководству пользователя для VirtualBox (www.virtualbox.org/manual/UserManual.html).

1. Загрузите Ubuntu 16.04.2 LTS на странице releases.ubuntu.com/16.04/ и установите его на VMware Workstation/Fusion или VirtualBox. Можно установить любую другую версию Ubuntu Linux, решать вам, если вам удобно устанавливать пакеты и решать проблемы с зависимостями.
2. Установите средства виртуализации в Ubuntu; это позволит разрешению экрана произвести автоматическую настройку в соответствии с геометрией вашего монитора и дополнительно усовершенствовать такие возможности, как расшаривание содержимого буфера обмена и копирование/вставка или перетаскивание файлов на вашем главном компьютере и виртуальной машине Linux. Чтобы установить инструменты виртуализации на VMware Workstation или VMware Fusion, вы можете следовать процедуре, описанной здесь: kb.vmware.com/selfservice/microsites/search.do?language=en_US&cmd=displayKC&externalId=1022525, – или посмотрите видео по адресу youtu.be/ueM1dCk3o58. После установки перезагрузите систему.
3. Если вы используете VirtualBox, то должны установить программное обеспечение Guest Additions. Для этого в меню VirtualBox выберите **Devices | Insert guest additions CD image** (Устройства | Вставить CD от Guest Additions). Появится диалоговое окно Guest Additions. Затем нажмите **Выполнить**, чтобы запустить программу установки с виртуального компакт-диска. Подтвердите свой пароль, когда будет предложено, и перезагрузите компьютер.
4. После установки операционной системы Ubuntu и инструментов виртуализации запустите виртуальную машину Ubuntu и установите следующие инструменты и пакеты.
5. Установите pip. Это система управления пакетами, используемая для установки и управления пакетов, написанных на Python. В этой книге я буду использовать ряд скриптов, написанных на Python; некоторые из них используют сторонние библиотеки. Для автоматизации установки сторонних пакетов вам необходимо установить pip. Запустите следующую команду в терминале для установки и обновления pip:

```
$ sudo apt-get update
$ sudo apt-get install python-pip
$ pip install --upgrade pip
```

Ниже приведен ряд инструментов и пакетов Python, которые будут использованы в этой книге. Чтобы установить их, запустите эти команды в терминале:

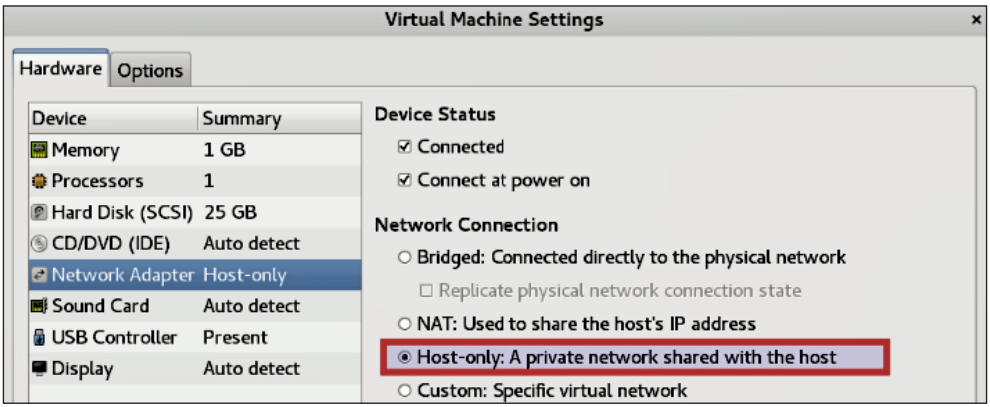
```
$ sudo apt-get install python-magic
$ sudo apt-get install upx
$ sudo pip install pefile
$ sudo apt-get install yara
$ sudo pip install yara-python
$ sudo apt-get install ssdeep
```

```
$ sudo apt-get install build-essential libffi-dev python python-dev \ libfuzzydev
$ sudo pip install ssdeep
$ sudo apt-get nstall wireshark
$ sudo apt-get install tshark
```

6. INetSim (www.inetsim.org/index.html) – мощная утилита, позволяющая моделировать различные интернет-службы (такие как DNS и HTTP), с которыми, как ожидается, вредоносные программы часто взаимодействуют. Позже вы поймете, как настроить INetSim для моделирования служб. Чтобы установить INetSim, используйте данные ниже команды. Использование INetSim будет подробно рассмотрено в главе 3 «Динамический анализ». Если у вас возникли проблемы с установкой INetSim, обратитесь к документации (www.inetsim.org/packages.html):

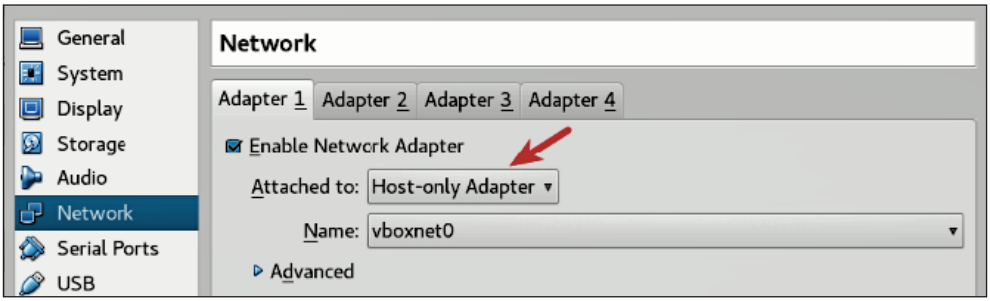
```
$ sudo su
# echo "deb http://www.inetsim.org/debian/inary /"> \
/etc/apt/sources.list.d/inetsim.list
# wget -O - http://www.inetsim.org/inetsim-archive-signing-key.asc | \
apt-key add -
# apt update
# apt-get install inetsim
```

7. Теперь вы можете изолировать виртуальную машину Ubuntu в своей среде, настроив виртуальное приложение для использования сетевого режима host-only. На VMware зайдите в **Network Adapter Settings** (Настройки сетевого адаптера) и выберите режим host-only, как показано на следующем рисунке. Сохраните настройки и перезагрузитесь.




В VirtualBox выключите виртуальную машину Ubuntu, а затем зайдите в раздел **Settings** (Настройки). Выберите сеть и измените настройки адаптера на Host-only Adapter, как показано на следующей диаграмме. Нажмите кнопку **OK**.

В VirtualBox иногда при выборе варианта **Host-only Adapter** имя интерфейса может отображаться как **Not selected** (Не выбрано). В этом случае вам необходимо сначала создать хотя бы один интерфейс host-only, перейдя в раздел **File| Preferences | Network | Host-only networks | Add host-only network** (Файл | Предпочтения | Сеть | Сети host-only | Добавить сеть host-only). Нажмите кнопку **OK**; затем откройте раздел **Settings. Select Network** (Настройки. Выбрать сеть) и измените настройки адаптера на host-only, как показано ниже. Нажмите кнопку **OK**.




8. Теперь мы назначим статический IP-адрес `192.168.1.100` для виртуальной машины Ubuntu Linux. Для этого запустите виртуальную машину, откройте окно терминала, введите команду `ifconfig` и запишите имя интерфейса. В моем случае имя интерфейса `ens33`. В вашем случае имя интерфейса может отличаться. Если оно отличается, вам нужно внести изменения при выполнении следующих шагов соответственно. Откройте файл `/etc/network/interfaces` с помощью команды:

```
$ sudo gedit /etc/network/interfaces
```

Добавьте следующие записи в конец файла (обязательно замените `ens33` на имя интерфейса в вашей системе) и сохраните его:

```
auto ens33
iface ens33 inet static
address 192.168.1.100
netmask 255.255.255.0
```

Файл `/etc/network/interfaces` должен теперь выглядеть так, как показано здесь. Недавно добавленные записи выделены:

```
# interfaces(5) file used by ifup(8) and ifdown(8)
auto lo
iface lo inet loopback

auto ens33
iface ens33 inet static
address 192.168.1.100
netmask 255.255.255.0
```

Затем перезапустите виртуальную машину Ubuntu Linux. На данный момент ее IP-адрес должен быть установлен как `192.168.1.100`. Вы можете проверить это, запустив следующую команду:

```
$ ifconfig
ens33 Link encap:Ethernet HWaddr 00:0c:29:a8:28:0d
inet addr:192.168.1.100 Bcast:192.168.1.255 Mask:255.255.255.0
inet6 addr: fe80::20c:29ff:fea8:280d/64 Scope:Link
UP BROADCAST RUNNING MULTICAST MTU:1500 Metric:1
RX packets:21 errors:0 dropped:0 overruns:0 frame:0
TX packets:49 errors:0 dropped:0 overruns:0 carrier:0
collisions:0 txqueuelen:1000
RX bytes:5187 (5.1 KB) TX bytes:5590 (5.5 KB)
```

9. Следующим шагом является настройка INetSim, чтобы он мог слушать и моделировать все службы на настроенном IP-адресе `192.168.1.100`. По умолчанию он использует локальный интерфейс (`127.0.0.1`), который необходимо изменить на `192.168.1.100`. Для этого откройте файл конфигурации, расположенный по адресу `/etc/inetsim/inetsim.conf`, используя следующую команду:

```
$ sudo gedit /etc/inetsim/inetsim.conf
```

Перейдите в раздел `service_bind_address` в файле конфигурации и добавьте эту запись:

```
service_bind_address 192.168.1.100
```

Добавленная запись (выделенная) в файле конфигурации должна выглядеть так:

```
# service_bind_address
#
# IP address to bind services to
#
# Syntax: service_bind_address <IP address>
#
# Default: 127.0.0.1
#
# service_bind_address 10.10.10.1
service_bind_address 192.168.1.100
```

По умолчанию DNS-сервер INetSim разрешит все доменные имена на `127.0.0.1`. Вместо этого мы хотим, чтобы доменное имя разрешалось на `192.168.1.100` (IP-адрес виртуальной машины Linux). Для этого перейдите в раздел `dns_default_ip` в файле конфигурации и добавьте запись, как показано здесь:

```
dns_default_ip 192.168.1.100
```

Добавленная запись (выделена в следующем коде) в конфигурации файла должна выглядеть так:

```
# dns_default_ip
#
# Default IP address to return with DNS replies
#
# Syntax: dns_default_ip <IP address>
#
# Default: 127.0.0.1
#
# dns_default_ip 10.10.10.1
dns_default_ip 192.168.1.100
```

После внесения изменений сохраните файл конфигурации и запустите основную программу INetSim. Убедитесь, что все службы работают, а также проверьте, слушает ли inetsim `192.168.1.100`, так, как выделено в следующем коде. Вы можете остановить службу, нажав сочетание клавиш **Ctrl**+**C**:

```
$ sudo inetsim
INetSim 1.2.6 (2016-08-29) by Matthias Eckert & Thomas Hungenberg
Using log directory: /var/log/inetsim/
Using data directory: /var/lib/inetsim/
Using report directory: /var/log/inetsim/report/
Using configuration file: /etc/inetsim/inetsim.conf
=== INetSim main process started (PID 2640) ===
Session ID: 2640
Listening on: 192.168.1.100
Real Date/Time: 2017-07-08 07:26:02
Fake Date/Time: 2017-07-08 07:26:02 (Delta: 0 seconds)
Forking services...
* irc_6667_tcp - started (PID 2652)
* ntp_123_udp - started (PID 2653)
* ident_113_tcp - started (PID 2655)
* time_37_tcp - started (PID 2657)
* daytime_13_tcp - started (PID 2659)
* discard_9_tcp - started (PID 2663)
* echo_7_tcp - started (PID 2661)
* dns_53_tcp_udp - started (PID 2642)
[..........REMOVED.............]
* http_80_tcp - started (PID 2643)
* https_443_tcp - started (PID 2644)
done.
Simulation running.
```

10. В какой-то момент вам понадобится передавать файлы между хостом и виртуальной машиной. Чтобы сделать это на VMware, выключите виртуальную машину и откройте раздел **Settings** (Настройки). Выберите **Options** | **Guest Isolation** (Параметры | Гостевая изоляция) и установите флажки напротив опций **Enable drag and drop** и **Enable copy** (Включить перетаскивание) и (Включить копирование и вставку). Сохраните настройки. На Virtualbox, пока виртуальная машина выключена, откройте раздел **Settings** | **General** | **Advanced** (Настройки | Общие | Дополнительно)

и убедитесь, что и общий буфер обмена, и Drag'n'Drop установлены в положении **Bidirectional** (Двунаправленный). Нажмите кнопку **OK**.

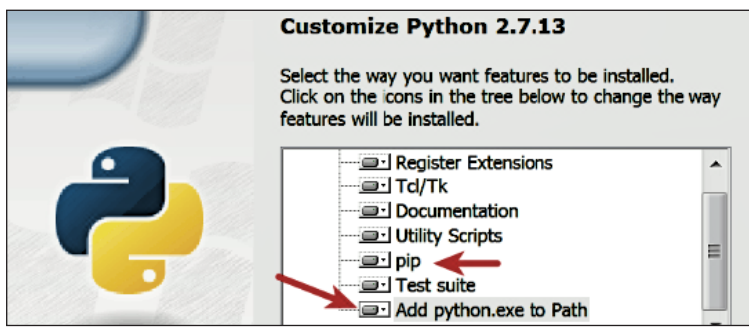
11. На этом этапе виртуальная машина Linux настроена на использование режима host-only, а INetSim настроен для модулирования всех служб. Последний шаг – сделать снимок файловой системы (чистый снимок) и назвать его на ваше усмотрение, чтобы вы могли при необходимости вернуть его в рабочее состояние. Для этого на рабочей станции VMware нажмите на **VM | Snapshot | Take Snapshot** (ВМ | Снапшот | Сделать снапшот). На Virtualbox можно сделать то же самое, нажав **Machine | Take Snapshot** (Машина | Сделать снапшот).

Помимо функции перетаскивания, также можно передавать файлы с главного компьютера на виртуальную машину с использованием общих папок; посетите эту страницу для VirtualBox (www.virtualbox.org/manual/ch04.html#sharedfolders), а для VMware (docs.vmware.com/en/VMware-Workstation-Pro/14.0/com.vmware.ws.using.doc/GUID-AACE0935-4B43-43BA-A935-FC71ABA17803.html).

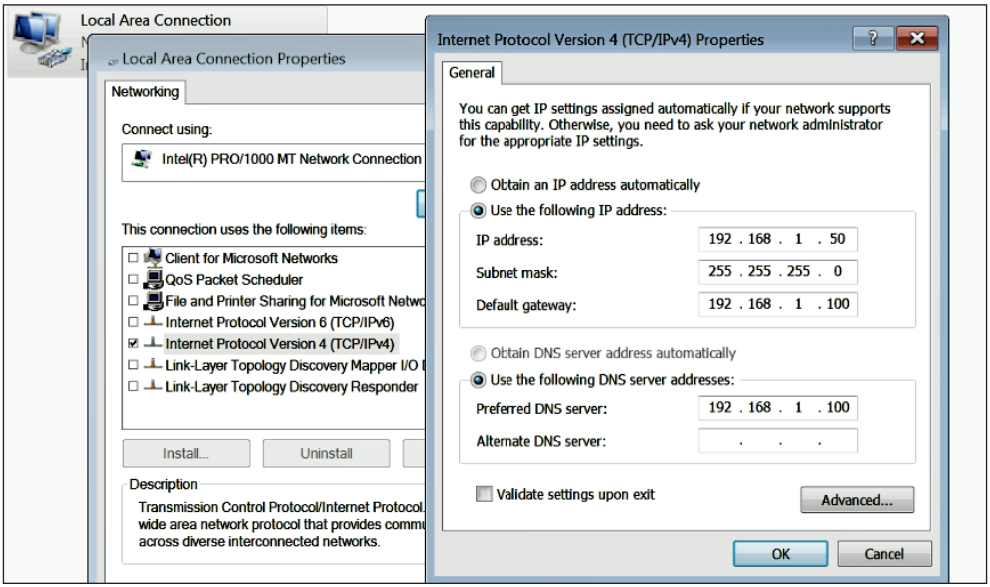
## 1.5.4 Установка и настройка виртуальной машины Windows

Перед настройкой виртуальной машины Windows сначала необходимо установить операционную систему Windows (Windows 7, Windows 8 и т. д.) на ваш выбор с использованием программного обеспечения для виртуализации (например, VMware или VirtualBox). После того как Windows будет установлена, выполните следующие действия.

1. Загрузите Python на странице www.python.org/downloads/. Не забудьте скачать Python 2.7.x (например, 2.7.13); большинство сценариев, используемых в этой книге, написано для запуска на версии Python 2.7 и может работать неправильно на Python 3. После того как вы скачали файл, запустите установщик. Убедитесь, что вы отметили опцию для установки pip и опцию **Add python.exe to Path** (Добавить python.exe в переменную Path), как показано на рисунке. Установка pip облегчит установку любой сторонней библиотеки Python, а добавление Python в переменную **Path** облегчит запуск Python из любого места.



2. Настройте виртуальную машину Windows для работы в режиме конфигурации сети host only. Чтобы сделать это в VMware или VirtualBox, откройте **Настройки сети** и выберите режим Host-only; сохраните настройки и перезагрузите компьютер (подобный шаг описан в разделе «Установка и настройка виртуальной машины Linux»).
3. Настройте IP-адрес виртуальной машины Windows на `192.168.1.x` (выберите любой IP-адрес, кроме `192.168.1.100`, потому что виртуальная машина Linux настроена на использование этого IP) и настройте шлюз по умолчанию и DNS-сервер на IP-адрес виртуальной машины Linux (то есть `192.168.1.100`), как показано ниже. Эта конфигурация необходима, чтобы при запуске вредоносной программы на виртуальной машине Windows весь сетевой трафик направлялся через виртуальную машину Linux.

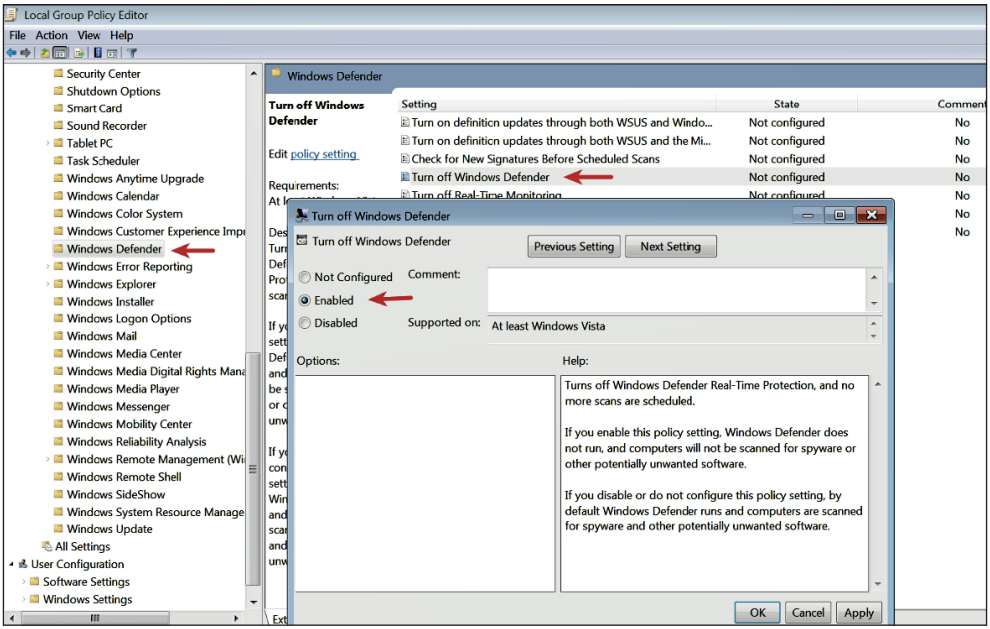



4. Включите виртуальные машины Linux и Windows и убедитесь, что они могут общаться друг с другом. Можно проверить подключение, запустив команду ping, как показано на этом скриншоте:

```
C:\Users\test>ping 192.168.1.100

Pinging 192.168.1.100 with 32 bytes of data:
Reply from 192.168.1.100: bytes=32 time<1ms TTL=64
Reply from 192.168.1.100: bytes=32 time<1ms TTL=64
Reply from 192.168.1.100: bytes=32 time<1ms TTL=64
Reply from 192.168.1.100: bytes=32 time<1ms TTL=64
```

5. Службу Защитника Windows необходимо отключить на виртуальной машине Windows, так как она может помешать, когда вы будете выполнять образец вредоносного ПО. Для этого нажмите сочетание клавиш **Windows**+**R**, чтобы открыть меню **Выполнить**, введите gpedit.msc и нажмите клавишу **Enter**, чтобы запустить редактор локальной групповой политики. В левой панели перейдите в раздел **Computer Configuration** | **Administrative Templates** | **Windows Components** | **Windows Defender** (Конфигурации компьютера | Административные шаблоны | Компоненты Windows | Защитник Windows). В правой панели дважды щелкните на **Turn off Windows Defender** (Выключить Защитника Windows), чтобы внести правки; затем выберите опцию **Enabled** (Включено) и нажмите кнопку **OK**:



6. Чтобы можно было передавать файлы (перетаскивать) и копировать содержимое буфера обмена между главным компьютером и виртуальной машиной Windows, следуйте инструкциям пункта 7 раздела «Установка и настройка виртуальной машины Linux».
7. Сделайте снимок файловой системы, чтобы вы могли возвращаться в исходное/чистое состояние после каждого анализа. Процедура создания снимка была описана в пункте 10 раздела «Установка и настройка виртуальной машины Linux».

На этом этапе ваша тестовая среда должна быть готова. Виртуальные машины Linux и Windows на вашем снимке файловой системы должны находиться в режиме сети host-only и быть способны общаться друг с другом. На протяжении всей этой книги я буду рассказывать о различных инструментах анализа вредоносных программ. Если вы хотите использовать их, то можете скопировать их на чистый снапшот на виртуальных машинах. Чтобы иметь постоянно обновленную информацию о снапшоте, просто перенесите/установите эти инструменты на виртуальных машинах и сделайте новый снимок.

## 1.6 Источники вредоносных программ

После настройки среды вам понадобятся образцы вредоносного ПО для выполнения анализа. В этой книге я использовал различные образцы вредоносных программ в качестве примеров. Поскольку эти образцы взяты из реальных атак, я решил не заниматься их распространением, так как это затрагивает правовые вопросы. Вы можете найти их (или аналогичные образцы) в различных хранилищах вредоносных программ. Ниже приведено несколько источников, из которых вы можете получить образцы вредоносных программ для вашего анализа. Некоторые из них позволяют скачать образцы вредоносных программ бесплатно (или после бесплатной регистрации), а некоторые требуют связаться с владельцем, чтобы создать учетную запись, после чего вы сможете получить их:

- Hybrid Analysis: www.hybrid-analysis.com/;
- KernelMode.info: www.kernelmode.info/forum/viewforum.php?f=16;
- VirusBay: beta.virusbay.io/;
- Contagio malware dump: contagiodump.blogspot.com/;
- AVCaesar: avcaesar.malware.lu/;
- Malwr: malwr.com/;
- VirusShare: virusshare.com/;
- theZoo: thezoo.morirt.com/.

Вы можете найти ссылки на другие источники вредоносного ПО в блоге Ленни Зельцера zeltser.com/malware-sample-sources/.

Если ни один из вышеупомянутых методов вам не подходит и вы хотите получить образцы вредоносных программ, используемые в этой книге, пожалуйста, не стесняйтесь связаться с автором.

## Резюме

Настройка изолированной тестовой среды крайне важна перед анализом вредоносных программ.

Выполняя анализ вредоносного ПО, вы обычно запускаете враждебный код для наблюдения за его поведением, поэтому наличие изолированной тестовой среды предотвратит случайное распространение вредоносного кода на вашу систему или системы производства вашей сети. В следующей главе вы узнаете об инструментах и методах извлечения ценной информации из образца вредоносного ПО с использованием статического анализа.