Sneak peak @ the Metasploit framework

Author: Karthik R aka 3psil0nLaMbDa, INDIA
Blog: www.epsilonlambda.wordpress.com

Metasploit is a free, open source pen-testing framework developed by the Open source community and Rapid7 team. The use of metasploit ranges from defending your own system by breaking into them to learning about the vulnerabilities that pose a real risk. For more information and to download the tool please visit http://www.metasploit.com.

The information provided in the series of articles is attributed to Metasploit Unleashed section in http://www.offensive-security.com and few videos by Mr. Vivek Ramachandra,founder of Security tube. This article covers the basic structure of metasploit and the need for a framework like MSF and also different techniques of Information gathering and Vulnerability scans using this great tool. The challenges posed by writing individual exploits, like time consumption and the level of skill required in coding and testing the exploit calls for a framework like metasploit.

The filesystem and libraries is quite self explanatory after installation. For example, tools folder would contain various command line utilities and the modules folder would contain different MSF modules. MSF is based on scripting language, so the script folder contains meterpreter and other scripts actually needed by the framework. Other directories like plug-in, external, lib all are intuitively explainable.

The disadvantage of using specific payloads while exploitation is that, when a new process is started in the target system, then, this event may trigger an alarm. An idea of ideal payload should avoid creation of new process, and whatever happens should happen within the scope of the payload. It should also allow us for writing scripts, and not create any new files on HD, because this may trigger the Antivirus.

The answer to these drawbacks is Meterpreter. It is a post exploitation tool; the principle behind is the use of **'In memory DLL injection'**. This avoids all the drawbacks of using specific payloads and enables us to write our own commands, and also uses an encrypted communication. **DLL injection** is basically making the target run our dll i.e. make a new process in the target which will calls our dll and then run it. For this to happen, we would need a dll injector, a target system and the dll that is to be injected. The coverage of dll injection as a topic in itself is out of the scope of this article.

The lab setup is Winxp attacker system with metasploit framework installed and winxp vulnerable system, both on vmware, and the payload that I shall be setting here to conceptualize meterpreter would be as shown in figure1.

The exploit being used here is the windows RPC DCOM exploit, to perform a buffer overflow in the target system.

Figure1 : payload-> windows/meterpreter/bind_tcp

This is going to bind to port 4444 of 192.168.13.30

When exploitation is done, we get a meterpreter console to the remote system. The actual process that happens during this phase is described in the figure2.

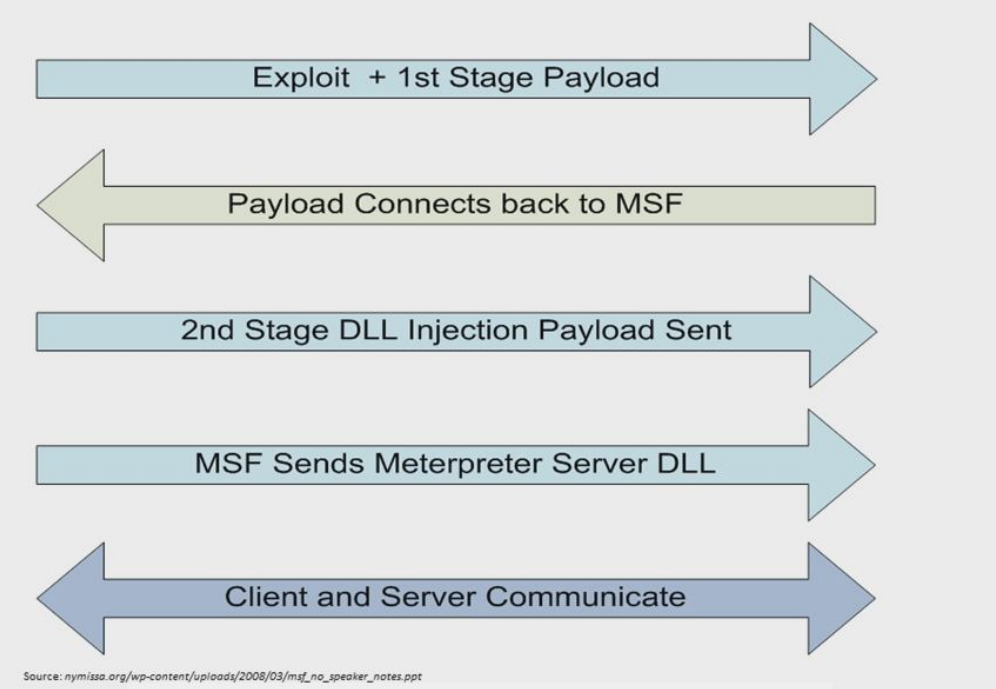

Figure2- Workflow of how meterpreter works (Self-explanatory)

Meterpreter's command set includes core commands, stdapi commands and Privilege escalation commands. Figure3 gives a sneak peak at the command set available under stdapi category.

Figure3- Stdapi networking commands and system commands.

The command set can be got by typing '?' in the meterpreter console.

Server side support dll is running on the victim under the stdapi module which is loaded by default, with meterpreter. Migrate command helps us to shift the work environment from one process to next on the victim. This comes in handy if in the remote system, the service on which initially the payload is bound is stopped in an unexpected manner.

In the similar way there are networking commands and system commands. Keystroke capturing can very easily be done using stdapi UI commandset. Keyscan_start will start the service, and keyscan_dump will show the captured key strokes.

Stealing windows tokens and impersonation

Windows security model assigns every user on the system with a unique SID (Security Identifier). Every thread for each user has a primary token associated which contains the information regarding privileges, groups etc. Impersonation token is a mechanism by which temporarily a process or a thread can assume identity of some other user on the same system. Once this is used up, then, the primary token is assumed back by the thread.

Attacks based on Impersonation tokens

1. Local privilege escalation
   Suppose a low privilege process runs in the system that has an admin authentication, there would be an impersonation token available for the admin. Now, if an attacker breaks in using some exploit to the process, then he would have access to the impersonation token for the admin and this can be dangerous.
2. Domain privilege escalation
   The difference here is that the attacker hops to other machines over the network using the impersonation token.

This can be accomplished in metasploit **using incognito**, in the meterpreter console. Figure4 shows the incognito.

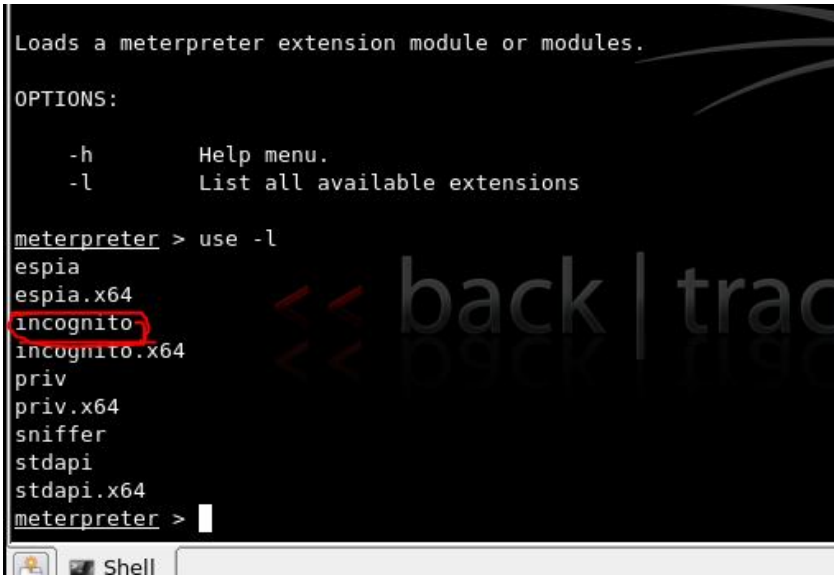

Figure4- Lists all available extensions for meterpreter including incognito.

Use commands like, List_tokens, steal_tokens and impersonate_token intuitively to carry out operations.

Performing client side exploits – Victim is behind a firewall/NAT:

Here, since the user is behind a firewall or NAT, it's up to the attacker to do a social engineering and present the victim with a link that will redirect him to the attacker's machine, which is in fact a metasploit instance. This is done coz, directly probing the target is not possible.

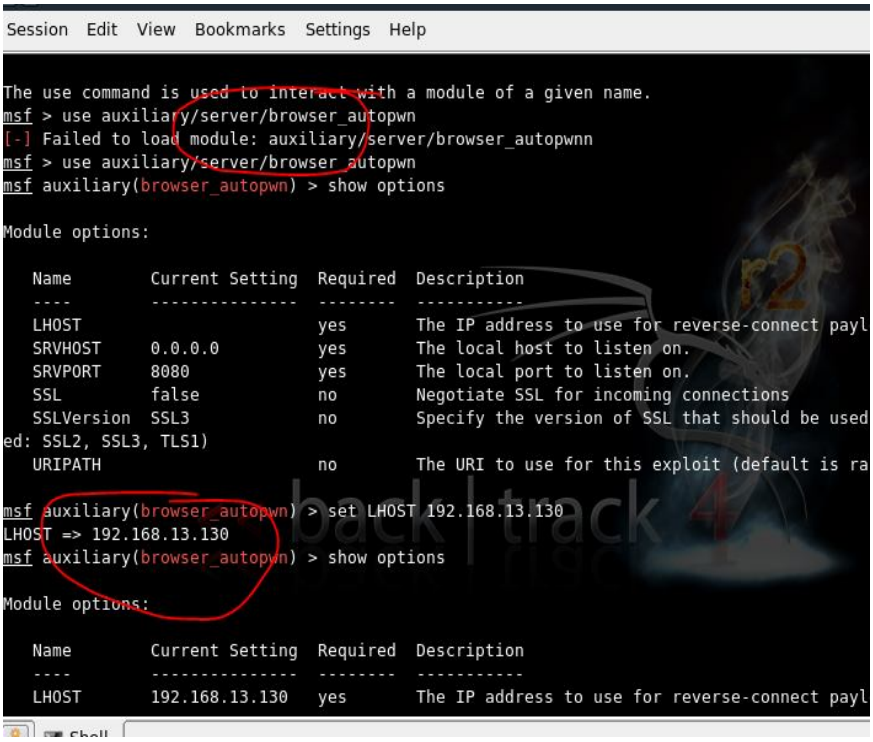

Figure5- Using the auxiliary module browser_autopwn we try to gain information about a victim behind the firewall.

After setting values, we give the **run** command. The server gets started after some time and exploits gets loaded for different browsers.
While sending a link to the victim, it should redirect to the attackers msf instance. Once the victim clicks on the link, a meterpreter session starts in the attacker's machine granting access to the victim's machine.