Using Metasploit with nessus bridge on Unbuntu

David J. Dodd

Ever wondered how to use the autopwn feature in Metasploit on Unbuntu? Want to run nessus from within metasploit? What database should I use; sqlite3 or postgres? I will explain the benefits of both.

Nessus is a vulnerability scanner program, it is free for personal use using the nessus for home. They also have a nessus for business which requires a fee. I will be discussing the nessus for home use and using it with the popular metasploit framework. Acquire the latest release of nessus homefeed Nessus-4.4.1-ubuntu1010_i386.deb and register for the activation code. Follow the instructions listed in the document ion for installing with Ubuntu and start to configure. Nessus daemon cant be started until nessus has been registered and the plugin download has occurred.

$ sudo /opt/nessus/bin/nessus-fetch –register 'registration code from nessus'

Add user

$ sudo /opt/nessus/sbin/nessus-adduser

Make cert

$ sudo /opt/nessus/sbin/nessus-mkcert

Start the nessus Daemon

$ sudo /etc/init.d/nessusd start

Open up web browser to https://localhost:8834, login and complete a policy for your scans. I would create a number of policies based on the different systems that you will be scanning. If your scanning a windows environment then having the plugin for Linux and BSD are pointless. Also make sure that you have safe checks enabled, select a port scanner to use, select credentials, select plugins (remember not to enable ones that will bounce the box), and select preferences. When finished you should have a number of different policies that will be numbered 1 – however many you have and you can give them names for example for scanning windows environment you can label them as windows. Now you can logout of nessus and close the web browser.

Now open up a terminal and browse to where metasploit is installed and run an update.

$ cd /opt/framework-3.6.0/msf3
$ sudo svn update

Before we start the msfconsole lets get our database in proper order. Now I have used sqlite3 in the past and even did a tutorial on my website using sqlite3 http://pbnetworks.net/?cmd=bbs&id=35 which worked fine but sometimes it may not work and give error warning 'Note that sqlite is not supported due to numerous issues. It may work, but don't count on it.' Postgres is the recommended database for Metasploit. So lets install the postgres database and libraries.

$ sudo apt-get install postgresql-8.4

$ sudo apt-get install rubygems libpq-dev

$ sudo gem install pg

$ sudo apt-get install libreadline-dev

$ sudo apt-get install libssl-dev

$ sudo apt-get install libpq5

$ sudo apt-get install ruby-dev

You will need to become the system postgres user

$ sudo -s

 # su postgres

Now you will need to create a database user:

$ createuser <user account name> -P

Enter password for new role:

Enter it again:

Shall the new role be a superuser? (y/n) *n*

Shall the new role be allowed to create databases? (y/n) *n*

shall the new role be allowed to create more new roles? (y/n) *n*

Next we need to crate a database:

$ createdb –owner=<user account name> msf_database

Now we can start up metasploit:

:/opt/framework-3.6.0/msf3*$ sudo ./msfconsole*

Enter in the following commands:

msf> db_driver postgresql

msf> db_connect <user account name>:<password>@127.0.0.1:5432/msf_database

msf> db_hosts

Now before, when using sqlite3, creating and connecting to the database was easy. I would start up metasploit and issue the following commands:

msf> db_driver sqlite3

msf> db_connect

To verify if the database was connected I would issue the following command:

msf> db_hosts

If everything looked good I would have no errors and I could use the db_nmap command. But sometimes I would encounter errors and it would crash. Using postgres is more reliable than sqlite3 but is still useful as I will describe later. Finally go ahead and enable the database on startup by issuing the following commands:

$ cat > ~/.msf3/msfconsole.rc

db_driver postgresql

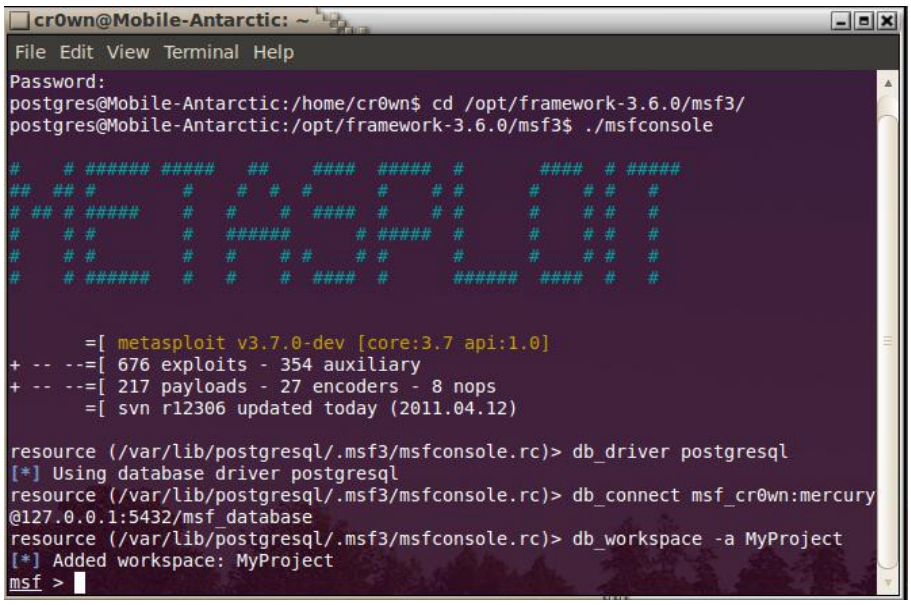
db_connect <user name account>:<password>@127.0.0.1:5432/msf_database

db_workspace -a MyProject

^D

Now the next time you fire up metasploit your database will automatically be up and you will be connected to it. Just make sure that you have postgres running, I run postgres manually before I start metasploit. (see Figure #1)

Figure 1 Notice that postgresql loads when first starting the msfconsole



$ sudo /etc/init.d/postgresql-8.4 start

$ su postgres

Now just change directory over to /opt/framework-3.6.0/msf3 and start the msfconsole.

Now that we have postgres as the database for metasploit lets start using nessus from within metasploit. Open up a second terminal and make sure nessus is running if not load the daemon. Now from the msfconsole load nessus (see figure #2)

msf > load nessus

Now let see what kind of commands the Nessus Bridge for Metasploit 1.1 has given us, type nessus_help (see figure #3)

Figure 2 loading nessus from the msfconsole

```
cr0wn@Mobile-Antarctic: /etc/init.d
File Edit View Terminal Help

resource (/var/lib/postgresql/.msf3/msfconsole.rc)> db_driver postgresql
[*] Using database driver postgresql
resource (/var/lib/postgresql/.msf3/msfconsole.rc)> db_connect msf_cr0wn:mercury
@127.0.0.1:5432/msf_database
NOTICE:  CREATE TABLE will create implicit sequence "sessions_id_seq" for serial
 column "sessions.id"
NOTICE:  CREATE TABLE / PRIMARY KEY will create implicit index "sessions_pkey" f
or table "sessions"
NOTICE:  CREATE TABLE will create implicit sequence "session_events_id_seq" for
serial column "session_events.id"
NOTICE:  CREATE TABLE / PRIMARY KEY will create implicit index "session_events_p
key" for table "session_events"
resource (/var/lib/postgresql/.msf3/msfconsole.rc)> db_workspace -a MyProject
[*] Added workspace: MyProject
msf > load nessus
[*] Nessus Bridge for Metasploit 1.1
[+] Type nessus_help for a command listing
[*] Creating Exploit Search Index - (/var/lib/postgresql/.msf3/nessus_index) - t
his wont take long.
[*]
[*] It has taken : 5.813429 seconds to build the exploits search index
[*] Successfully loaded plugin: nessus
msf >
```

Figure 3 nessus_help

```
cr0wn@Mobile-Antarctic: /etc/init.d
File Edit View Terminal Help
msf > nessus_help

Command                   Help Text
-------                   ---------
Generic Commands
----------------          ----------------
nessus_connect            Connect to a nessus server
nessus_save               Save nessus login info between sessions
nessus_logout             Logout from the nessus server
nessus_help               Listing of available nessus commands
nessus_server_status      Check the status of your Nessus Server
nessus_admin              Checks if user is an admin
nessus_server_feed        Nessus Feed Type
nessus_find_targets       Try to find vulnerable targets from a report
nessus_server_prefs       Display Server Prefs

Reports Commands
----------------          ----------------
nessus_report_list        List all Nessus reports
nessus_report_get         Import a report from the nessus server in Nessus v2 f
ormat
nessus_report_hosts       Get list of hosts from a report
nessus_report_host_ports  Get list of open ports from a host from a report
nessus_report_host_detail Detail from a report item on a host

Scan Commands
----------------          ----------------
nessus_scan_new           Create new Nessus Scan
nessus_scan_status        List all currently running Nessus scans
nessus_scan_pause         Pause a Nessus Scan
nessus_scan_pause_all     Pause all Nessus Scans
nessus_scan_stop          Stop a Nessus Scan
nessus_scan_stop_all      Stop all Nessus Scans
nessus_scan_resume        Resume a Nessus Scan
nessus_scan_resume_all    Resume all Nessus Scans

Plugin Commands
----------------          ----------------
nessus_plugin_list        Displays each plugin family and the number of plugins
nessus_plugin_family      List plugins in a family
nessus_plugin_details     List details of a particular plugin

User Commands
----------------          ----------------
nessus_user_list          Show Nessus Users
nessus_user_add           Add a new Nessus User
nessus_user_del           Delete a Nessus User
nessus_user_passwd        Change Nessus Users Password

Policy Commands
----------------          ----------------
```

msf > nessus_help

The commands are divided up into different sections labeled Generic, Reports, Scan, Plugin, User, and Policy commands. Before we can run a scan we need to connect to the nessus server by using the nessus_connect command

msf > nessus_connect <nessus username>:<password>@localhost:8834 ok

This should connect and authenticate you. From here you can run the scans, review the results, and load the scan results into the database and use autopwn feature. Or you can view the results and find a vulnerability with a system you scanned and throw a single exploit and get a meterpreter shell. Depending on the environment you may want to review the results of your nessus output and find the appropriate exploit to use instead of generating the noise of running autopwn. Now lets start our scan by issuing nessus_scan_new command as follows nessus_scan_new <policy id> (this was set in your nessus policy settings) <scan name> (generic) <target> (ip address)

msf > nessus_scan_new 1 winXP_home 192.168.1.124

To check up on the status of our scan use the nessus scan status feature (see figure #4)

Figure 4 nessus_scan_status

```
cr0wn@Mobile-Antarctic: /etc/init.d
File  Edit  View  Terminal  Help
[*] Authenticated
msf > nessus_scan_new 1 192.168.1.124
[*] Usage:
[*]        nessus_scan_new <policy id> <scan name> <targets>
[*]        use nessus_policy_list to list all available policies
msf > nessus_scan_new 1 winXP_home 192.168.1.124
[*] Creating scan from policy number 1, called "winXP_home" and scanning 192.168.1.1
24
[*] Scan started.  uid is 092da411-1f14-2e7f-4c06-5515a802026c809fa8837565190d
msf > nessus_scan_status
[+] Running Scans

Scan ID                                          Name        Owner  Started
        Status   Current Hosts  Total Hosts
-------                                          ----        -----  -------
        ------   -------------  -----------
092da411-1f14-2e7f-4c06-5515a802026c809fa8837565190d  winXP_home  cr0wn  16:26 Apr 1
2 2011  running  0              1


[*] You can:
[+]             Import Nessus report to database :      nessus_report_get <reportid>
[+]             Pause a nessus scan :                   nessus_scan_pause <scanid>
msf >
```

msf > nessus_scan_status

When the scan has completed you can view the results using the following commands

msf > nessus_report_list

We can view a list of hosts from the report with the following command

msf > nessus_report_hosts UID

To view further information issue the following command

msf > nessus_report_host_ports <ip address> UID (see Figure #5)

Figure 5 nessus_report_host_ports 192.168.1.124 UID

```
cr0wn@Mobile-Antarctic: /etc/init.d
File  Edit  View  Terminal  Help
msf > nessus_report_hosts 092da411-1f14-2e7f-4c06-5515a802026c809fa8837565190d
[+] Report Info

Hostname       Severity  Sev 0  Sev 1  Sev 2  Sev 3  Current Progress  Total Progres
s
--------       --------  -----  -----  -----  -----  ----------------  -------------
-
192.168.1.124  16        3      16     0      0      42521             42521

[*] You can:
[*]         Get information from a particular host:          nessus_report_host_port
s <hostname> <report id>
msf > nessus_report_host_ports 192.168.1.124 092da411-1f14-2e7f-4c06-5515a802026c809
fa8837565190d
[+] Host Info

Port  Protocol  Severity  Service Name  Sev 0  Sev 1  Sev 2  Sev 3
----  --------  --------  ------------  -----  -----  -----  -----
0     icmp      1         general       0      2      0      0
0     tcp       1         general       0      6      0      0
0     udp       1         general       0      1      0      0
23    tcp       1         telnet        1      1      0      0
135   tcp       0         epmap         1      0      0      0
137   udp       1         netbios-ns    0      1      0      0
139   tcp       1         smb           1      4      0      0
1900  udp       1         upnp-client   0      1      0      0

[*] You can:
[*]         Get detailed scan infromation about a specfic port: nessus_report_host_d
etail <hostname> <port> <protocol> <report id>
msf >
```

To see a list of hosts issue the db_host command. If you want to remove hosts from the db_hosts file then issue the db_del_host command (see Figure #6)

Now with the scan complete and the host listed in the db_hosts file you can run the autopwn tool or find an exploit that will work against the box. More on this in another article next month.

Now lets take a look at using nmap within the metasploit framework.

To use the nmap command from within the metasploit framework use the 'db_nmap' command to run nmap scans against targets and have the scan results stored in the database. When running on Back|Track I can issue many different nmap commands such as db_nmap -sS -sV -T 3 -P0 -O <ip address> -D RND --packet-trace. Which show the results: -sS TCP SYN stealth scan, -sV version scan, -T 3 normal scan, -O find the operating system, -D RND use a decoy and generate a random, non-reserved IP address, and finally --packet-trace will trace packets and data sent and received. I like to use the

packet-trace feature on large scans because if it fails you can see it. Now this is great feature to use while in the msf console but I cant do this when using Unbuntu and connected to the postgres database as the postgres user. Why? Because I get an error saying that only the root user has the ability to use this nmap option (see Figure #7). I can use 'db_nmap -v -sV 192.168.15.0/24 --packet-trace' and the scan runs and produces an output. I have view the results with the following commands (Figure 8)

msf > db_hosts

msf > db_services -c port,state

Figure 6 db_del_host command

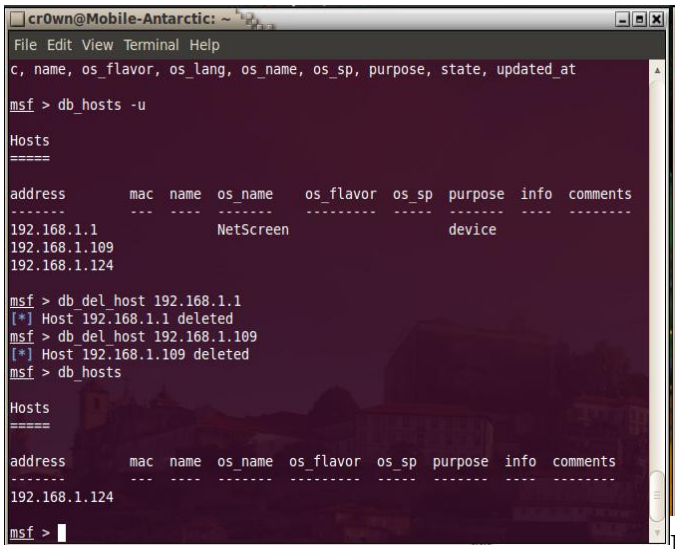
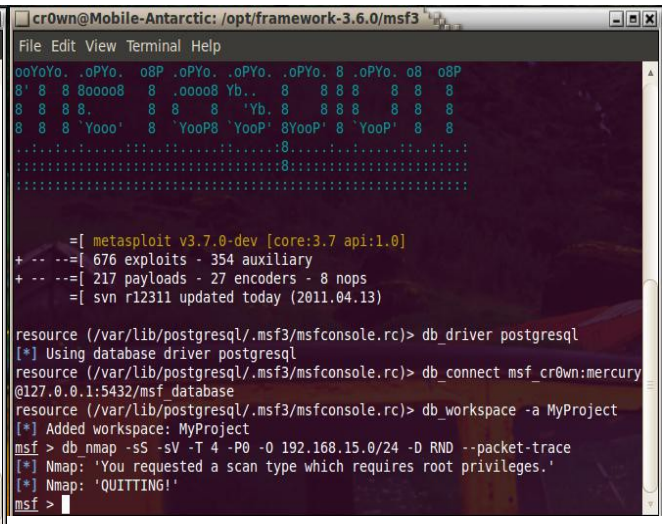


Figure 7 nmap error with postgres



Now if I want to issue complex nmap scans I can exit out of the msf prompt, exit out of postgres, stop the database and login with sudo and use the sqlite3 database. The same command that the OS didn't allow me to use now can be used with no problem (Figure #9)

msf > db_nmap -sS -sV -T 4 -P0 -O 192.168.15.0/24 -D RND --packet-trace

Look at the difference in results we now have after viewing information in the db_hosts and db_services -c port,state commands. Compare difference between figure #10 & figure #8 below.

Figure 9 db_nmap using sqlite3

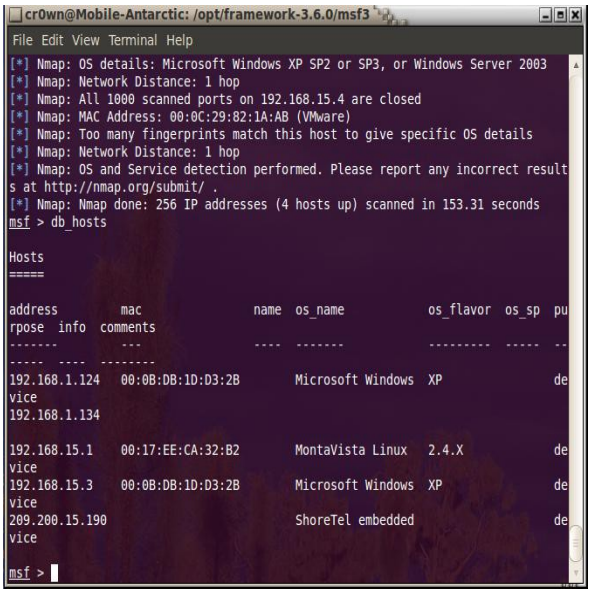


Figure 10 nmap results using sqlite3

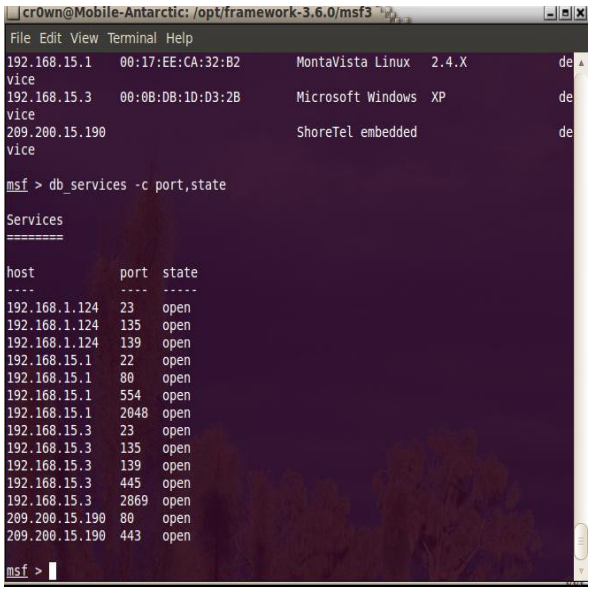


Figure #8 db_nmap using postgres database

```
cr0wn@Mobile-Antarctic: /opt/framework-3.6.0/msf3
File  Edit  View  Terminal  Help

Hosts
=====

address       mac  name  os_name          os_flavor  os_sp  purpose  info  co
mments
-------       ---  ----  -------          ---------  -----  -------  ----  --
------
192.168.1.124
192.168.15.1             Linux                                 device
192.168.15.2
192.168.15.3             Microsoft Windows                     server

msf > db_sevices -c port,state
[-] Unknown command: db_sevices.
msf > db_services -c port,state

Services
========

host          port  state
----          ----  -----
192.168.1.124 23    open
192.168.1.124 135   open
192.168.1.124 139   open
192.168.15.1  22    open
192.168.15.1  80    open
192.168.15.1  554   open
192.168.15.1  2048  open
192.168.15.2  22    open
192.168.15.2  135   open
192.168.15.2  139   open
192.168.15.2  445   open
192.168.15.2  1024  open
192.168.15.3  23    open
192.168.15.3  135   open
192.168.15.3  139   open
192.168.15.3  445   open
192.168.15.3  2869  open

msf >
```

Conclusion

This information can be useful in checking the integrity and strength of your network if you are the Network Security Engineer for your workplace, and have permission to do so. Doing this to networks that you have no authorization to be on is against the law in many if not all countries. For more information and some video tutorial please visit my website at http://pbnetworks.net

On the 'Net

Link to postgres setup: http://dev.metasploit.com/redmine/projects/framework/wiki/Postgres_setup

Link to video tutorials: http://pbnetworks.net/?cmd=bbs

About the Author

David J. Dodd is currently in the United States and holds a current 'Secret' DoD Clearance and is available for consulting on various Information Assurance projects. A former U.S. Marine with Avionics background in Electronic Countermeasures Systems. David has given talks at the San Diego Regional Security Conference and SDISSA, is a member of InfraGard, and contributes to Secure our eCity http://securingourecity.org. He works for pbnetworks Inc. http://pbnetworks.net a small service disabled veteran owned business located in San Diego, CA and can be contacted by emailing: dave@pbnetworks.net.

Let pbnetworks get your pen test on target

Visit us and learn how http://pbnetworks.net
How secure is your network?