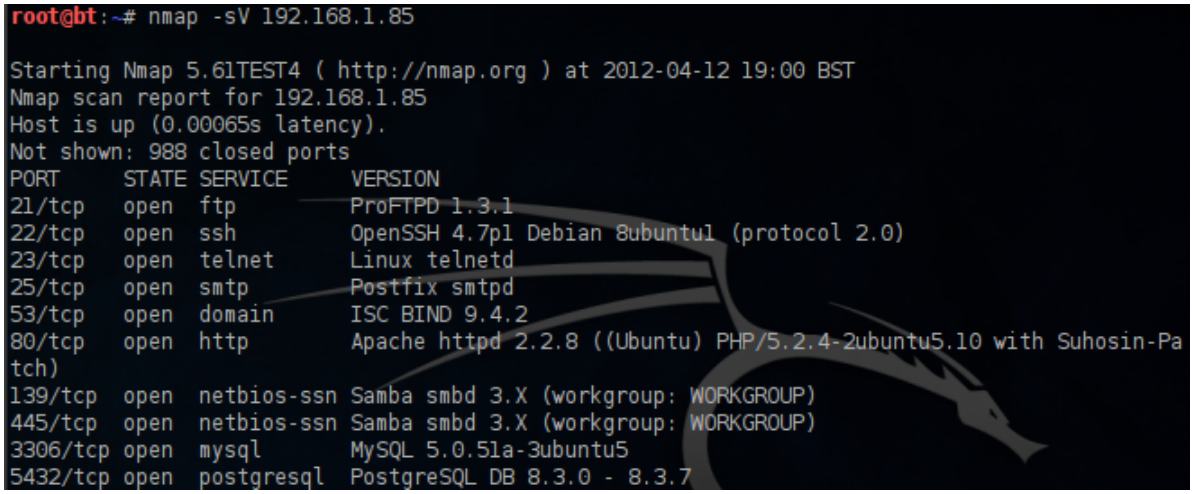
Attacking PostgreSQL

pentestlab.blog/category/exploitation-techniques/page/9

April 13, 2012

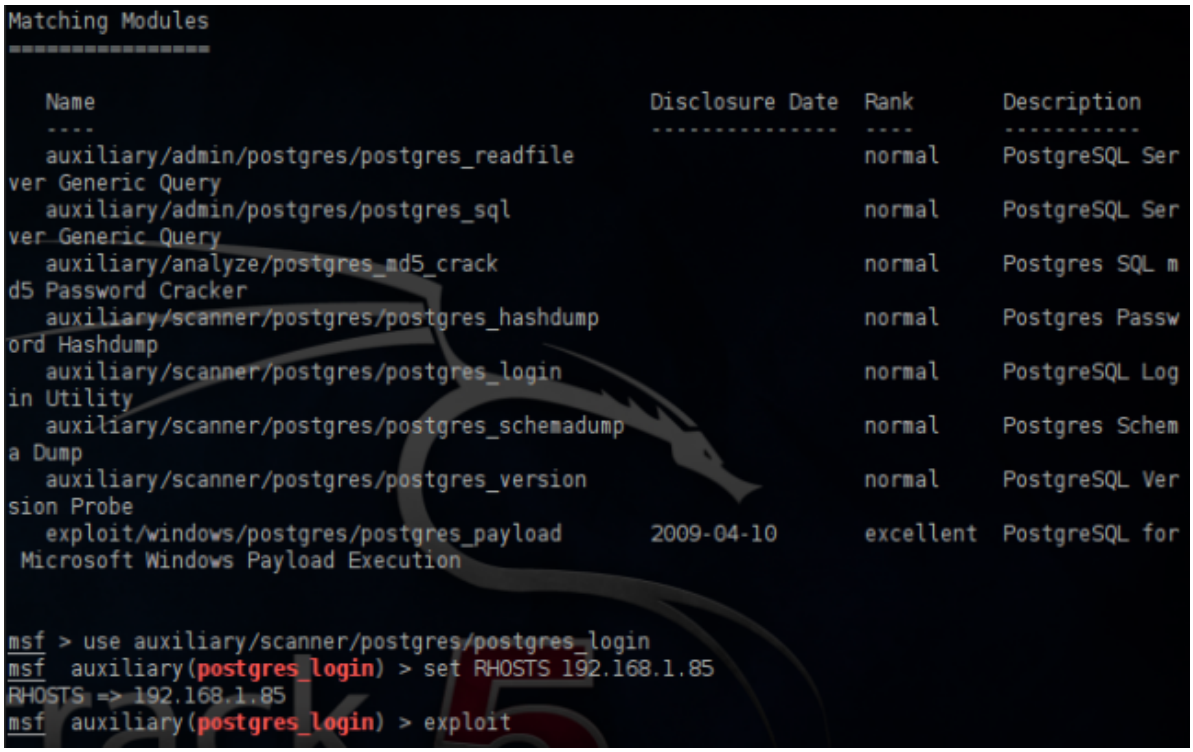
PostgreSQL is a database that comes with MacOS X Lion as a default standard database.Also according to wikipedia the majority of Linux distributions have the PostgreSQL in the supplied packages.So besides the regular databases (Oracle,MySQL etc.) there will be times as a penetration tester that we will need to assess and this database.In this article we will see how we can attack a system that contains a PostgreSQL database.

Lets say that we have perform a port scan on a server and we have identify that is running a PostgreSQL database at port 5432.



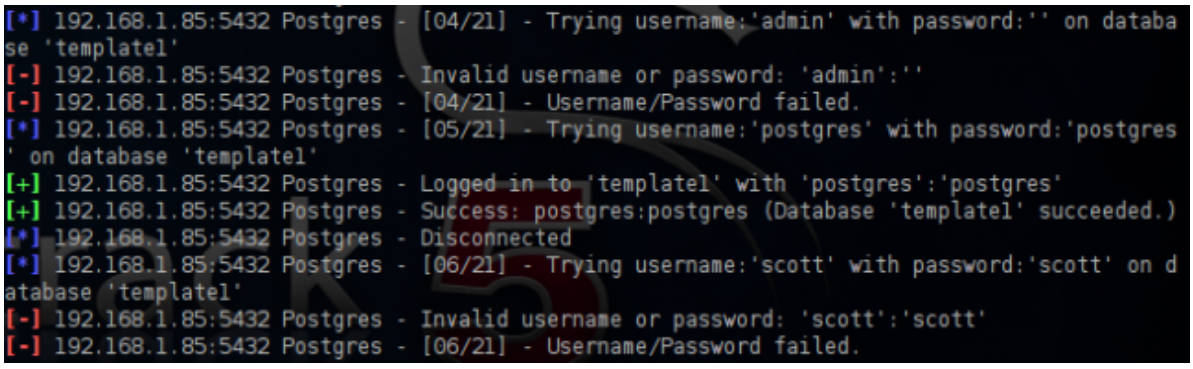
Discovery of PostgreSQL Database

We will try a brute force attack in order to discover any weak credentials that will allow us then to connect to the database.We will open the metasploit framework and we will use the **postgres_login scanner.**



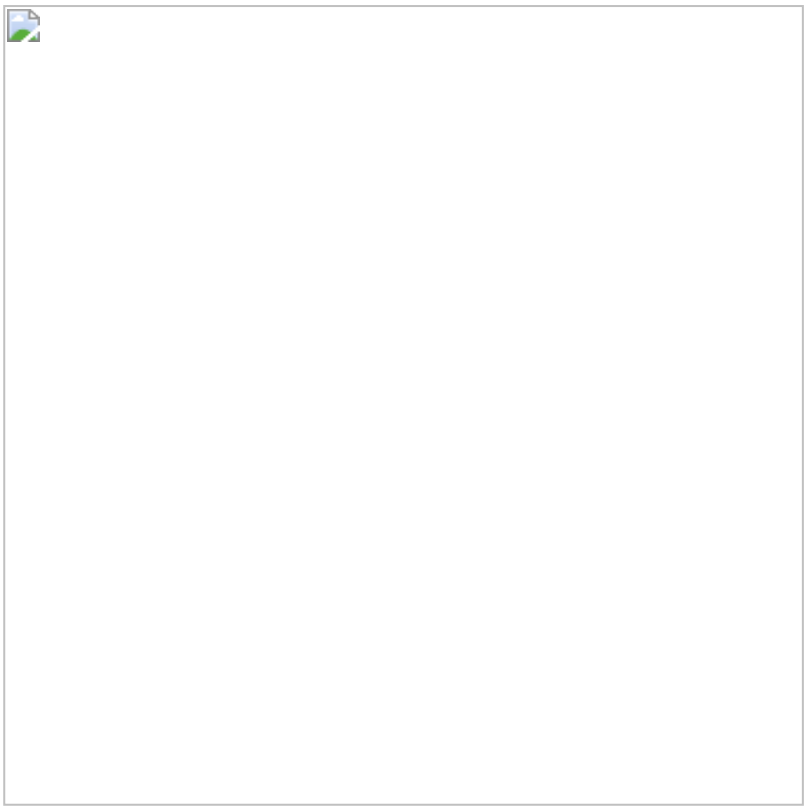
Choosing and configuring the postgres scanner

This scanner is already configured to use the default wordlists about postgreSQL databases of metasploit framework so we will use them in this case.As you can see from the next image we have successfully discovered some valid credentials after the execution of the scanner.



Valid credentials discovered on postgreSQL database

Now that we have a valid username and password we can use that to connect to the database by using a psql client.The first query that we want to execute is the ***select usename, passwd from pg_shadow;*** because it will return to us the password hashes of the database from the **pg_shadow** table.



Connecting to the PostgreSQL Database

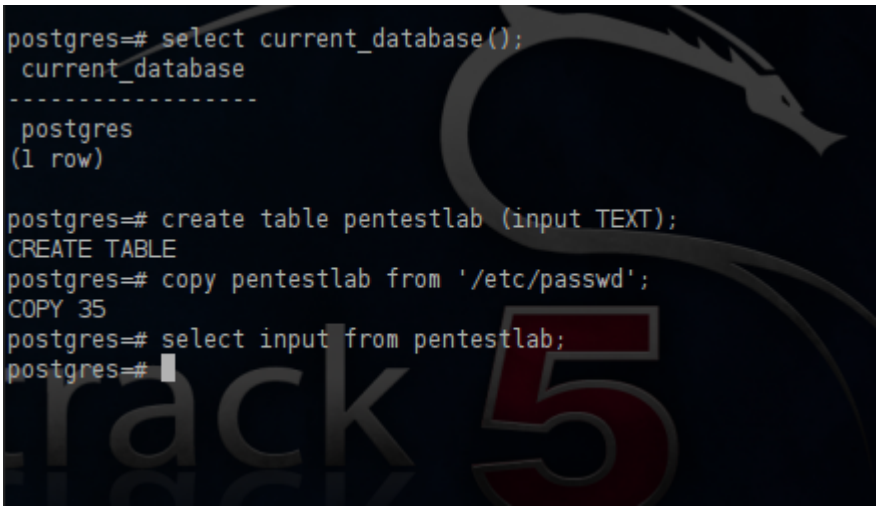
Another option is to look at the databases that exist with the command **\l**

```
postgres=# \l
                              List of databases
   Name    |  Owner   | Encoding |   Access privileges
-----------+----------+----------+-----------------------
 postgres  | postgres | UTF8     |
 template0 | postgres | UTF8     | =c/postgres
                                 : postgres=CTc/postgres
 template1 | postgres | UTF8     | =c/postgres
                                 : postgres=CTc/postgres
(3 rows)

postgres=#
```

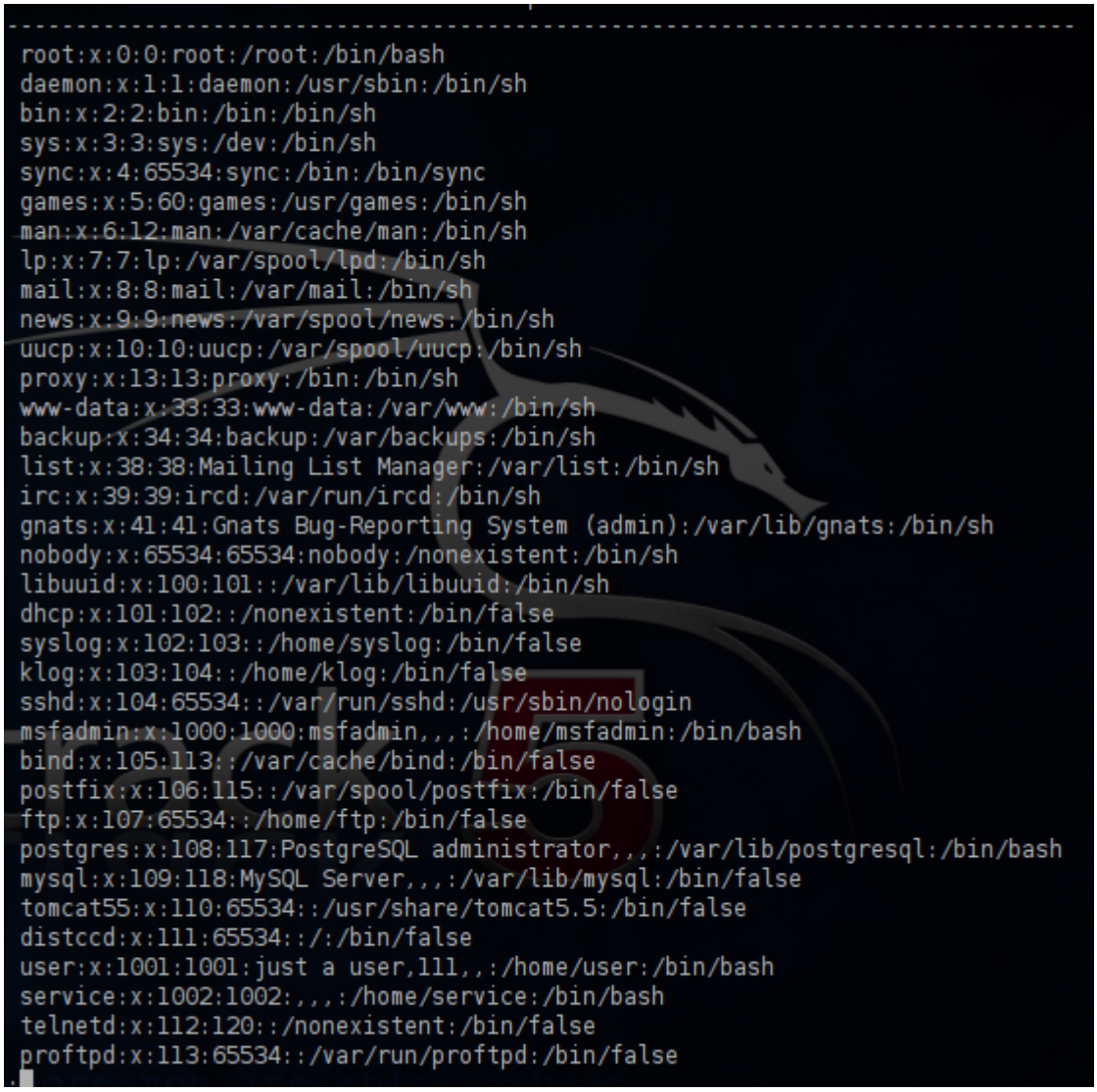
List the current databases

As you can see there are 3 databases in place.What we will try to do here is to select one of the databases and then we will create a new table called **pentestlab** and we will copy the contents of the **/etc/passwd** file to this table.



Creating a new table and copying the contents of /etc/passwd

We have now retrieved all the existing passwords of the remote server.



Conclusion

All databases from the moment that are installed in a system containing default credentials.So we need to be aware about these default accounts in order to remove them or change them.Also as we already saw in that article the first thing that we did

when we took access to the database was to check the available databases.Then we copied the contents of passwd to a new table that we have created in order to obtain passwords for other services as well.It is also important not to forget to delete anything that you will create (tables,users,new databases) in order to revert the postgreSQL to its previous state.