Evaluating PostgreSQL Database Security Through Automated Brute-Force Attacks Using Metasploit

*Project by **D.James Chrishan***

The `scanner/postgres/postgres_login` module in Metasploit is designed to perform brute-force attacks against PostgreSQL databases. It attempts to log in to a PostgreSQL database using a list of usernames and passwords, helping security professionals identify weak credentials.

How It Works:

- **Brute-Force Attack:** The module tries various combinations of usernames and passwords to authenticate against a PostgreSQL database. If successful, it provides the attacker with access to the database.
- **Protocol:** The module targets the PostgreSQL protocol, commonly used in databases for web applications and other services.

Configuration and Usage:

Set the Module:

```
use auxiliary/scanner/postgres/postgres_login
```

Set the Target Host:

```
set RHOSTS <target IP>
```

Set the Target Port (Optional):

```
set RPORT <port number>
```

- The default port for PostgreSQL is 5432.

Set the Username List (Optional):

```
set USERNAME <username>
```

- You can also use `USER_FILE` to specify a file containing multiple usernames.

Set the Password List (Optional):

```
set PASSWORD <password>
```

- You can also use `PASS_FILE` to specify a file containing multiple passwords.

Set Number of Threads (Optional):

```
set THREADS <number of threads>
```

- Increasing the number of threads can speed up the brute-force process.

Run the Module:

```
run
```

Example Command:

```
use auxiliary/scanner/postgres/postgres_login
set RHOSTS 192.168.1.100
set RPORT 5432
set USERNAME postgres
set PASS_FILE /path/to/passwordlist.txt
set THREADS 5
run
```

Explanation:

- **RHOSTS:** Specifies the target IP address.
- **RPORT:** Specifies the target port (default is 5432 for PostgreSQL).
- **USERNAME:** Defines a single username to try.
- **PASS_FILE:** Defines the path to a file containing a list of passwords to try.
- **THREADS:** Sets the number of concurrent threads for the brute-force attack.

Use Cases:

- **Database Security Testing:** Assess the strength of PostgreSQL database credentials.
- **Identifying Weak Passwords:** Help database administrators identify and replace weak passwords with stronger ones.
- **Penetration Testing:** Used as part of a larger penetration test to assess the security posture of a network.

Important Considerations:

- **Ethical Use:** Ensure you have explicit permission to perform brute-force attacks on the target database. Unauthorized use of this module is illegal and unethical.
- **Account Lockout:** Be aware of any account lockout policies that may be in place to avoid detection or unintended consequences.
- **Impact on Database:** Brute-force attacks can impact the performance of the target database, so use with caution in production environments.

The `scanner/postgres/postgres_login` module is a crucial tool for security professionals aiming to evaluate and strengthen the security of PostgreSQL databases by identifying weak or default credentials.