**TALAKUNCHI**

**BATCH: 03**

**NAME: VIBHOR KUMAR DEWANGAN**

**INTERNSHIP PROJECT 01:**

**System Hacking:**

1. **Hydra**
2. **auxiliary Module**
3. **NSE Scripts**
4. **John the ripper**
5. **Password generating using Crunch**

We can do System Hacking by following these steps:

1. **Hydra:**

Step 01) We have to do Brute-Forcing Activity when someone is maintain the default username

and passwords.

**Brute-Forcing Activity:** It represents just doing trial and error method by giving different kind

of username and password and trying to find out our username and

password.

Step 02) Download the default username and password from the goggle from the github.

After that, it will do the Bruit-Forcing activity .

Step 03) Give some random username (fig.01) and passwords (fig.02) in your kali-linux

Kali-Linux Operating System as an input to hydra tool.

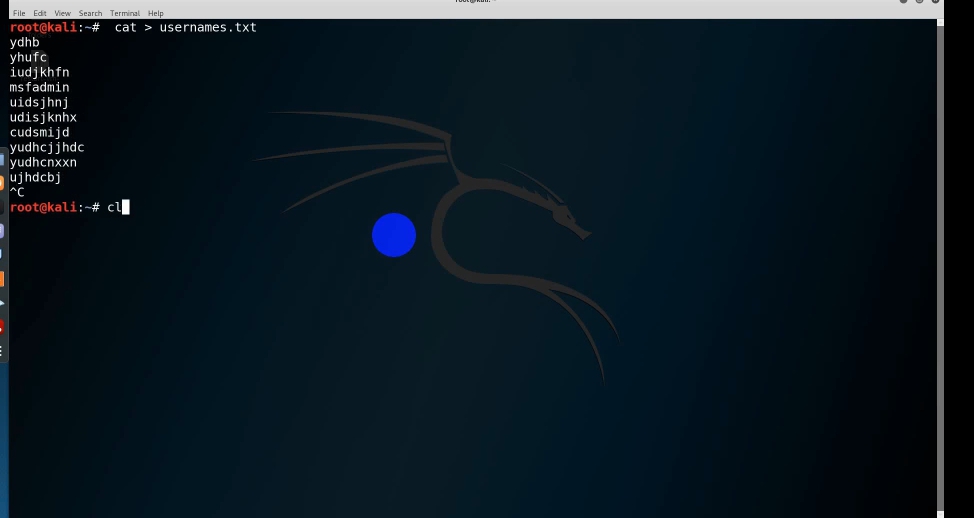


Fig.01 (Giving some random username)

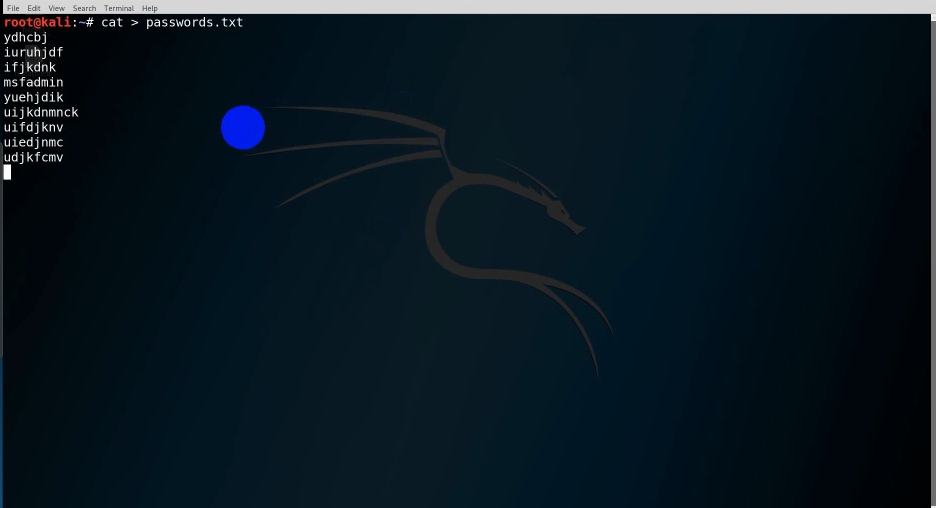


Fig.02 ( Giving some random passwords)

Step 05) Now we can do Bruit-Forcing activity to any open port. Let’s take an example of

“telnet”.

hydra -L [path uname] -P [path pass] <telnet://Target> ipaddress

Now, we use the tool “Hydra” as:

# hydra -L /root/usernames.txt -P/root/passwords.txt <telnet://192.168.114.130>

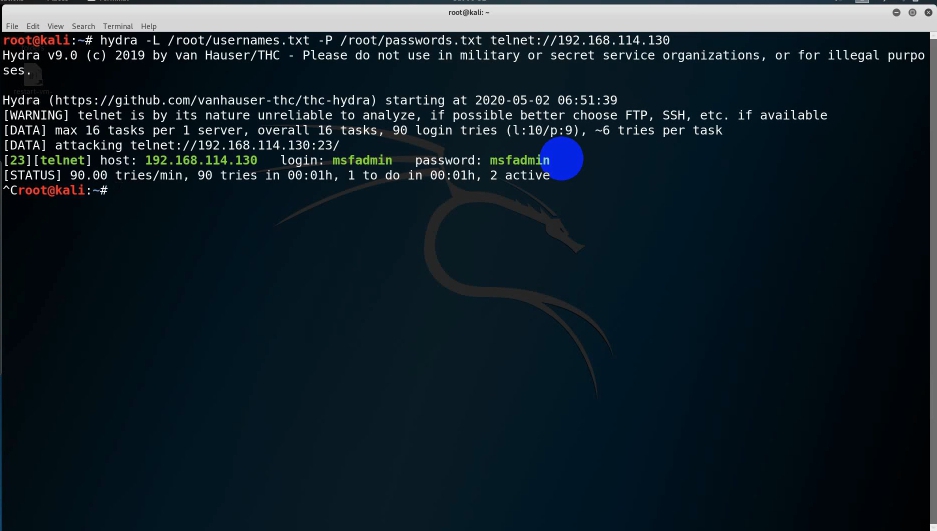


fig. 03(Checking for particular username & passwords given by admin)

1. **auxiliary Module:**

Step 01) We can go to our Kali-Linux Operating system, the **auxiliary Module** is itself

present in the meta-exploitable framework.

Step 02) Now, we have to go in the meta-exploitable framework where we can do lot

of activities (like brute-forcing, information gathering , few attack etc) by giving

command, “msfconsole”.

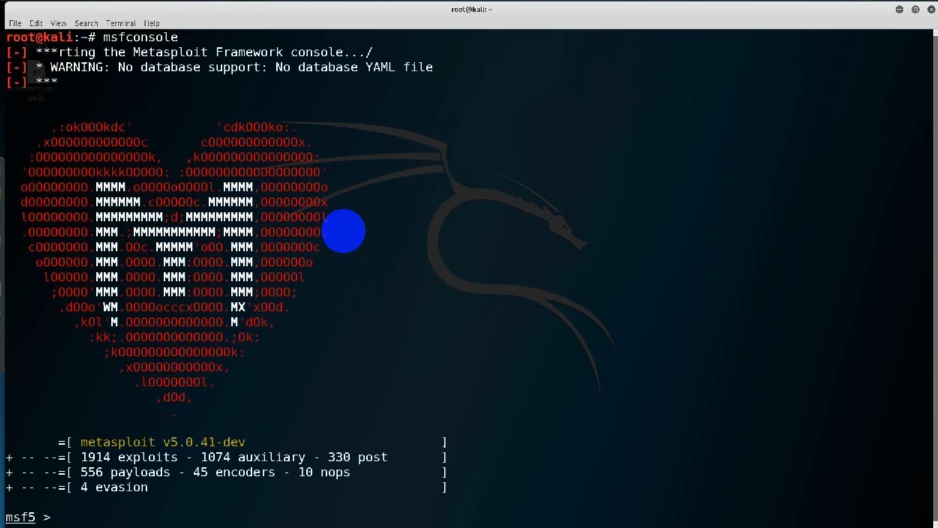


fig.04(Major attacks done from the particular framework)

Here, it is providing a module called “auxiliary module”.It is giving so many options to each

and every protocol. Maybe, that is for gathering information or doing attacks, whatever it

may be according to the requirements.

Step 03) Go in “**msf5 > use auxillary/scanner/ssh/ssh**”. There we can view “**ssh login**” where

We can do brute-forcing activity by utilising this module and can find the username and

passwords.. Likewise, we can do for telnet, ssh etc.

Step 04) Go into **“msf5 auxiliary (scanner/ssh/ssh\_login) >”** . It is providing the login option for

doing the brute-forcing activities. Atleast one port will be in open state, we have to go

to that module. After that, give a command called “**clea**r”.

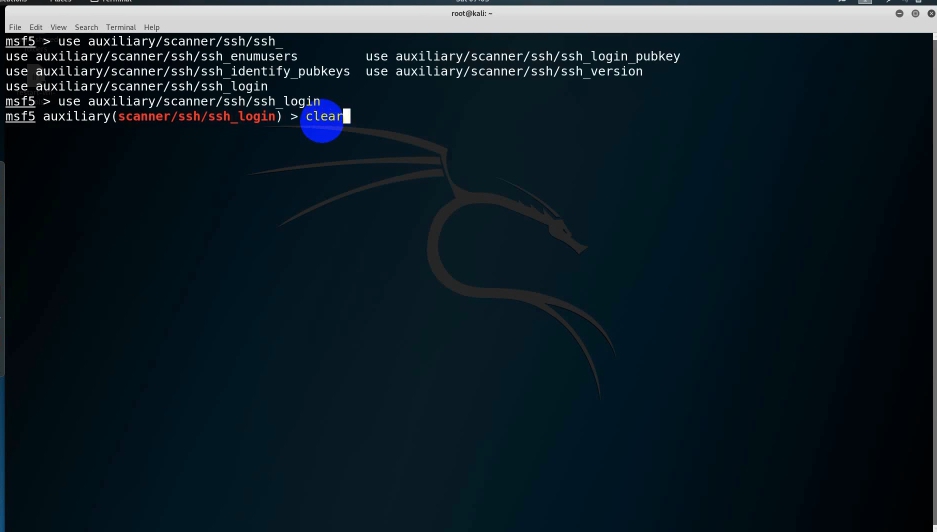


fig.05( Providing login option for doing Brute-forcing activity)

Step 05) After that, give “**show option**”. When we give show option, here it shows “**USER\_FILE**”

i.e file containing usernames, one per line and “**PASS\_FILE**” i.e file containing passwords

per line. Give the path as:

msf5 auxiliary(scanner/ssh/ssh\_login) > set USER\_FILE/ root/usernames.txt

msf5 auxiliary(scanner/ssh/ssh\_login) > set PASS\_FILE/root/passwords.txt

msf5 auxiliary(scanner/ssh/ssh\_login) > set RHOSTS 192.168.114.130

By utilising these usernames & passwords, it will do the brute-forcing activity and show us the output.

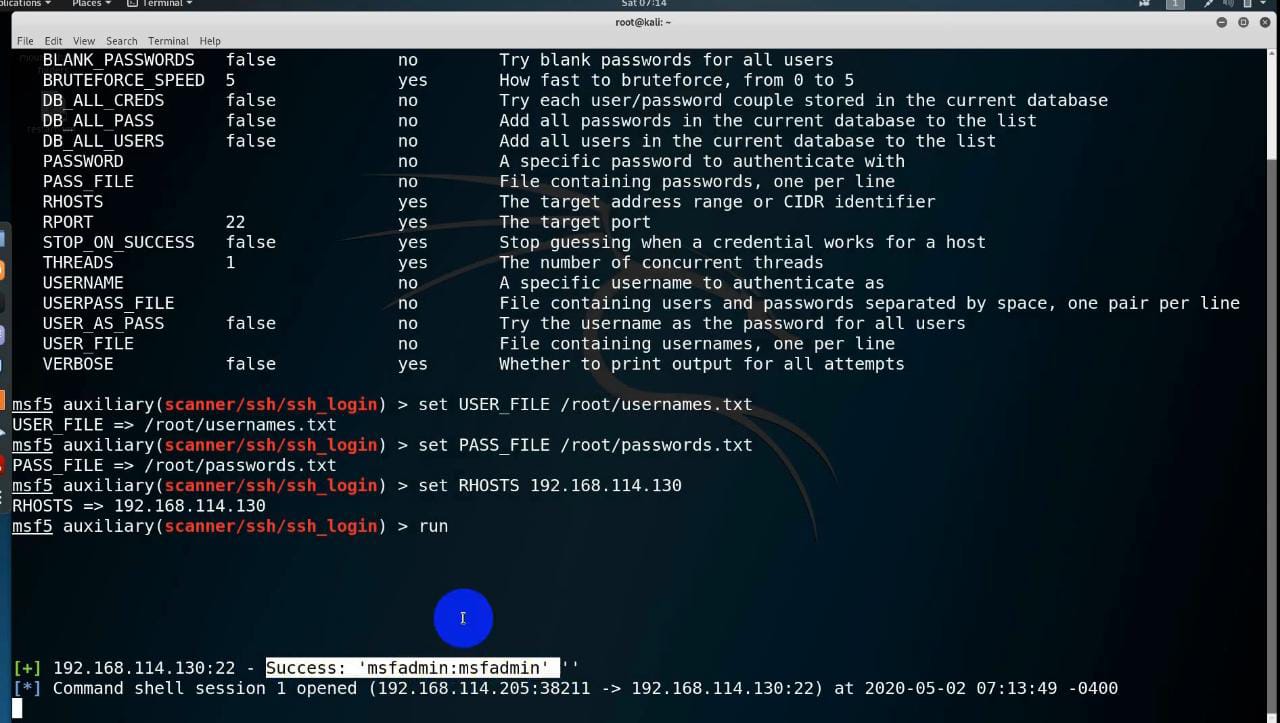


fig.06(Output of the brute-forcing activity done)

1. **NSE SCRIPTS:**

Here there is no need to give input like hydra and auxiliary module. We can do the Brute-

Forcing activity just by utilizing that automated script. There are a lot of automated scripts present

in our Kali-Linux Operating System. For doing NSE Scripts system hacking , follow these steps:

Step 01) We have to simply open our Kali-Linux Operating System. We have to just go to the path where

the automated scripts are present i.e “cd/usr/share/nmap /scripts” .For running any script, we

will get at least some information.

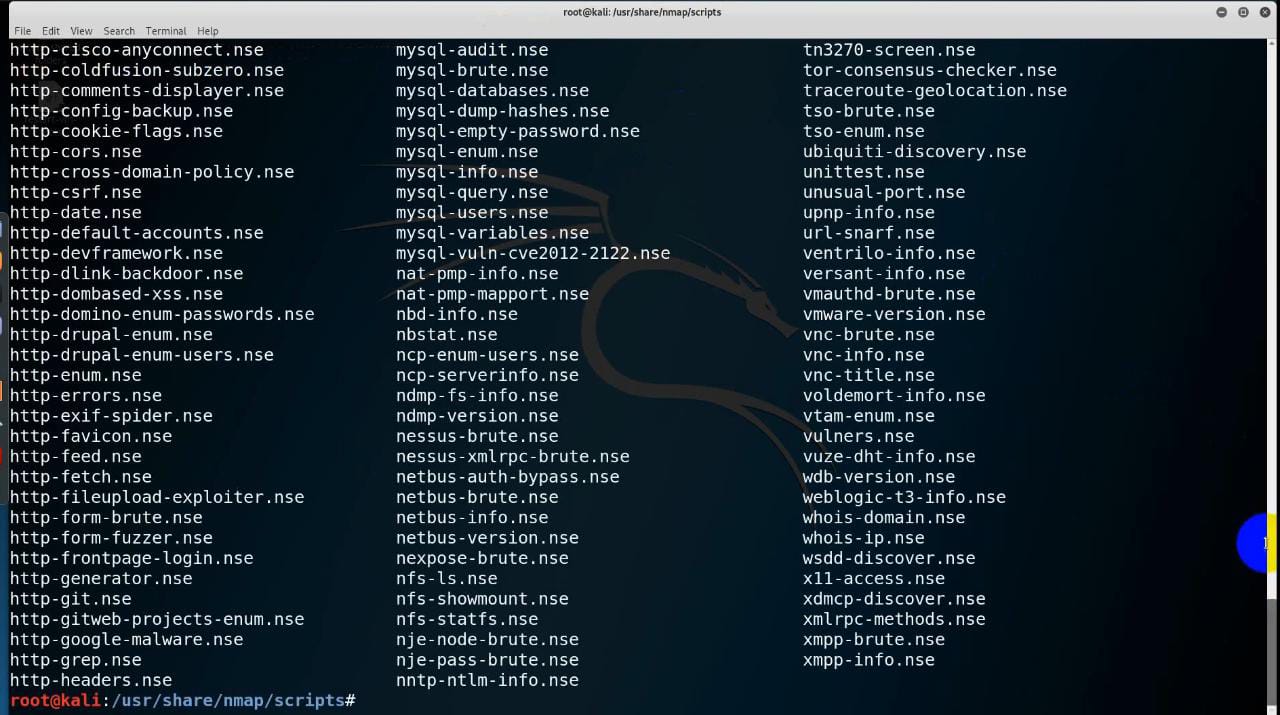


fig. 07( List of scripts provided by Kali-Linux operating system)

Step 02) To filter whatever script we need, give the command as “# **ls -l | grep ssh**” . After that it

will do the brute-forcing activity itself.

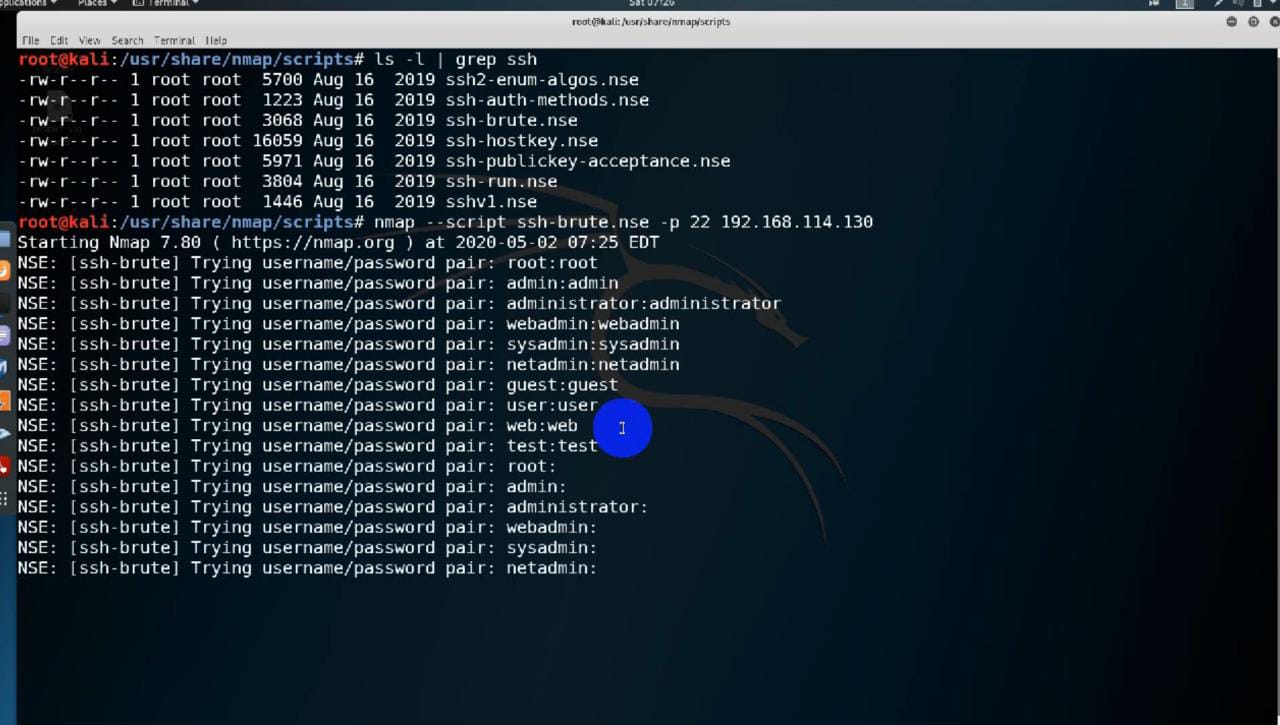
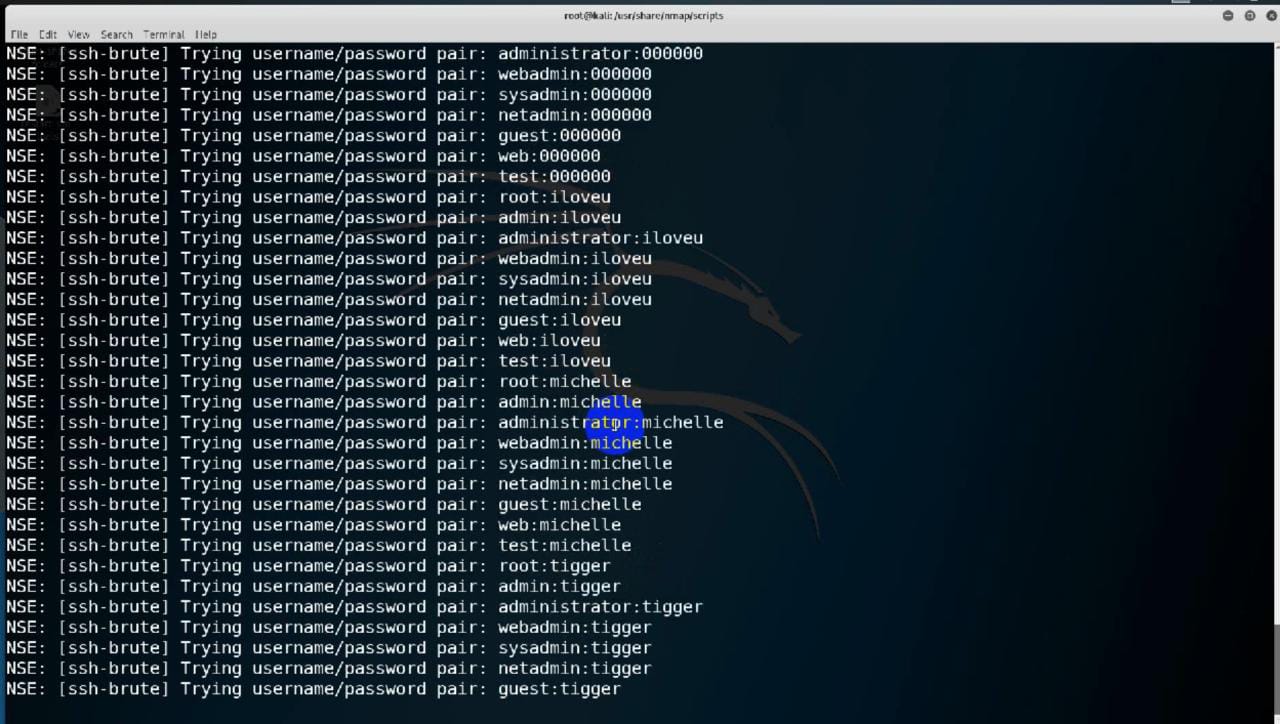


fig.08 & 09( Checking for username & password by itself)



1. **JOHN THE RIPPER:**

John the Ripper is a fast password cracker, it is used to crack the hashes. After

taking the server into control, we will open the shadow file. There will be passwords

like the usernames and passwords. Here the passwords will be shown in the form of

“hash- values”. We will be copying all the hash values . After cracking the hash-values

we will get the correct passwords.

Step 01) Go to Kali-Linux Operating System, there type a command “**# cat /etc/shadow**”.

After getting those hash-values, we will copy all the hash-values.

Step 02) Give a command as “ **cat > hashcrack.txt**”. Now, we can see all the copied

hash-values inside “**ls command**”

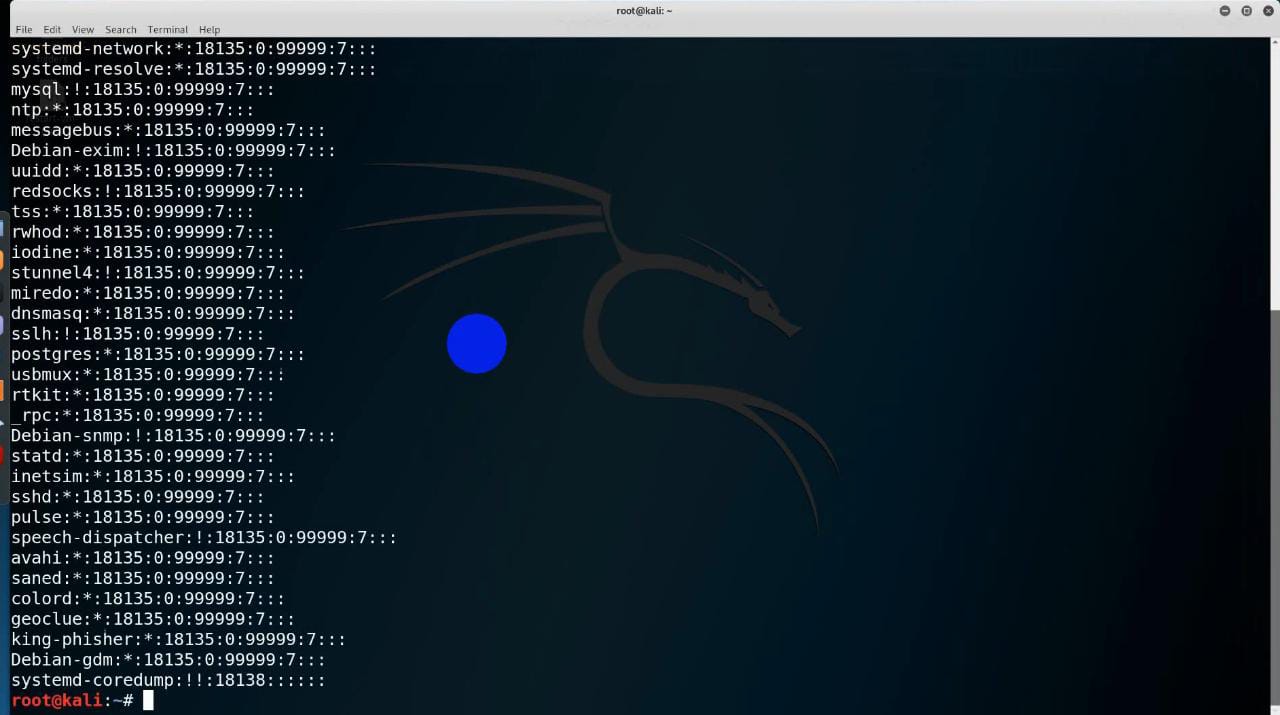


fig.10 (Various Hash-values shown by Kali-Linux Operating System)

Step 5) Now after taking any device into our control, open the Shadow file . In that shadow file

all the passwords will be stored in the form of hash-values. Just copy the hash-values and

paste it in the text-file. Now give this input to “**John hashcrack.txt**”. At some point it is

saying like “There is a username called “**root**” and for that the password is “**too**r”. So,

it is having a has-value for the password. Therefore ,the session is completed.

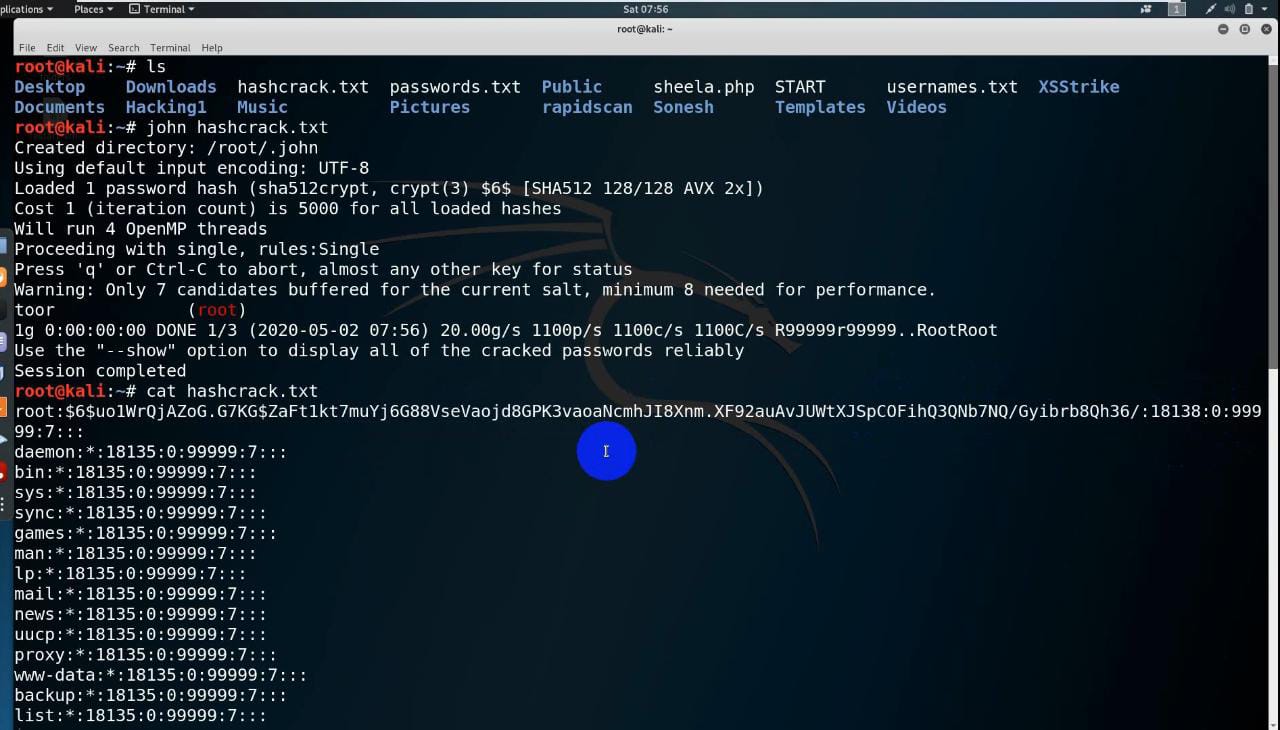


Fig.11 (It shows the completed session)

1. **PASSWORD GENERATING USING CRUNCH:**

Password brute-force is a technique of generating all possible combinations of characters and

using them for cracking. **Crunch** is one of the powerful tool to create such list of passwords.

For example- To create a wordlist containing the characters 0-9 and A-F, we enter commands

“# **crunch 6 6 0123456ABCDEF -0 filename.txt**”.

Crunch can also be used to generate more customized password lists. Example- Abc@123.

Here **,** is Uppercase

**@** is lowercase

**^** is a special char

**%** is numeric.

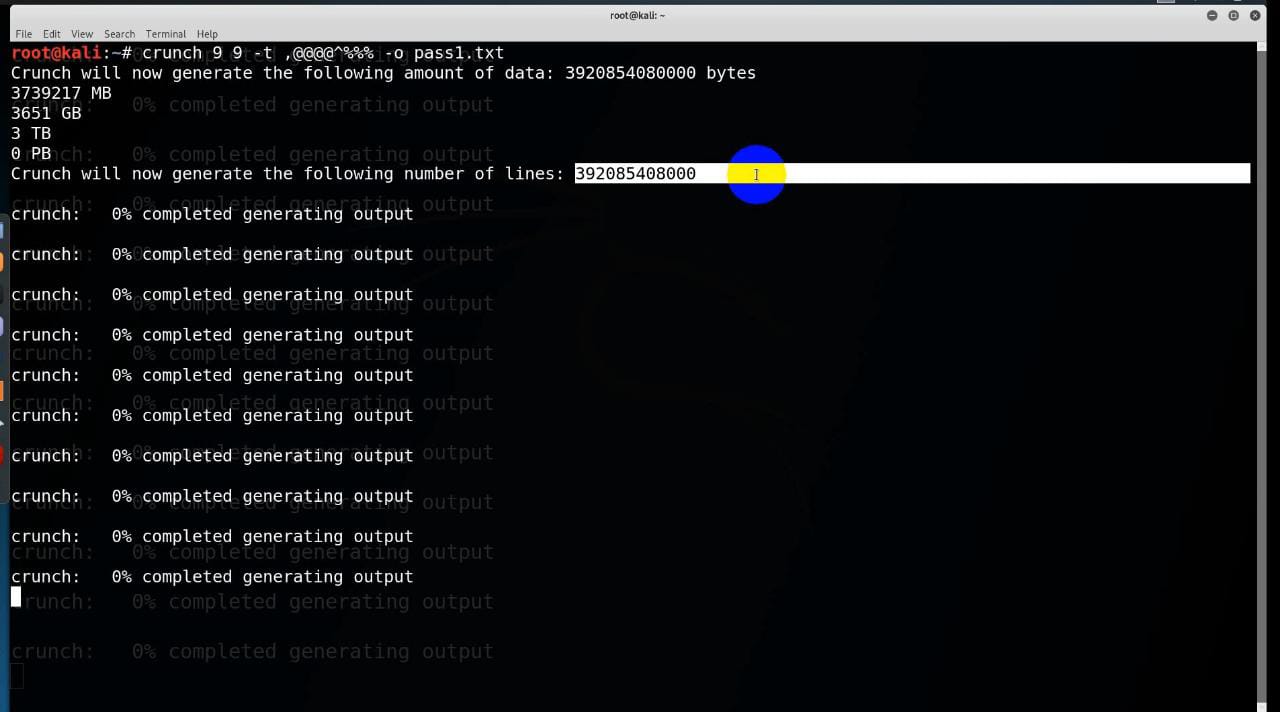


fig.12 (Brute-forcing activity done by Crunch)

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*