John The Ripper

Password Hash Cracking

Original Author: *Pavandeep Singh*

Table of Contents

Abstract

We know the importance of John the ripper in penetration testing, as it is quite popular among password cracking tool. In this report, we are introducing John the ripper and its various usage for beginners.

Additionally, we will use John the Ripper to crack the password hashes of some of the file formats like zip, rar, pdf and much more.

Disclaimer: This report is provided for educational and informational purpose only (Penetration Testing). Penetration Testing refers to legal intrusion tests that aim to identify vulnerabilities and improve cybersecurity, rather than for malicious purposes.

Introduction

What is John the Ripper?

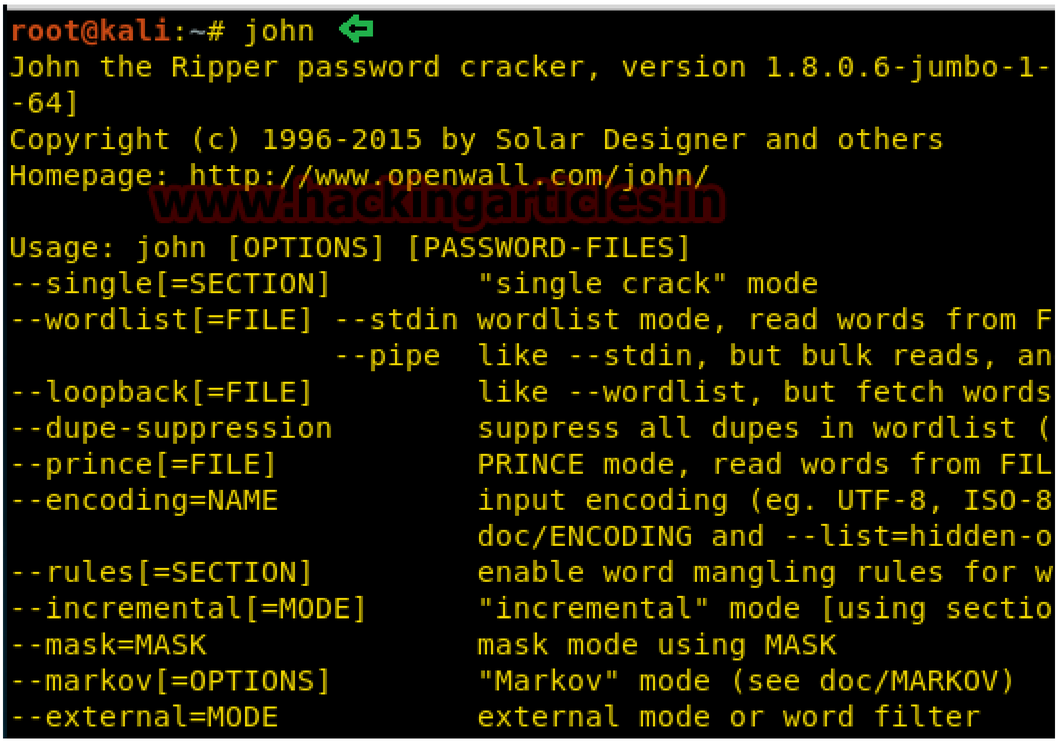
John the Ripper is a free password cracking software tool developed by Openwall. Originally developed for Unix Operating Systems but later on developed for other platforms as well. It is one of the most popular password testings and breaking programs as it combines a number of password crackers into one package, autodetects password hash types, and includes a customizable cracker. It can be run against various encrypted password formats including several crypt password hash types commonly found in Linux or Windows. It can also be to crack passwords of Compressed files like ZIP and also Documents files like PDF.

Where to get John the Ripper?

John the Ripper can be downloaded from Openwall's Website, or from the Official John the Ripper Github Repo.

Usage

John the Ripper comes pre-installed in Linux Kali and can be run from the terminal as shown below:



```
root@kali:~# john
John the Ripper password cracker, version 1.8.0.6-jumbo-1-
-64]
Copyright (c) 1996-2015 by Solar Designer and others
Homepage: http://www.openwall.com/john/

Usage: john [OPTIONS] [PASSWORD-FILES]
--single[=SECTION]        "single crack" mode
--wordlist[=FILE] --stdin wordlist mode, read words from F
                  --pipe  like --stdin, but bulk reads, an
--loopback[=FILE]         like --wordlist, but fetch words
--dupe-suppression        suppress all dupes in wordlist (
--prince[=FILE]           PRINCE mode, read words from FIL
--encoding=NAME           input encoding (eg. UTF-8, ISO-8
                          doc/ENCODING and --list=hidden-o
--rules[=SECTION]         enable word mangling rules for w
--incremental[=MODE]      "incremental" mode [using sectio
--mask=MASK               mask mode using MASK
--markov[=OPTIONS]        "Markov" mode (see doc/MARKOV)
--external=MODE           external mode or word filter
```

John the Ripper works in 3 distinct modes to crack the passwords:

1. Single Crack Mode
2. Wordlist Crack Mode
3. Incremental Mode

John the Ripper Single Crack Mode

In this mode John the ripper makes use of the information available to it in the form of a username and other information. This can be used to crack the password files with the format of

```
Username:Password
```

For Example: If the username is “Hacker” it would try the following passwords:

hacker

HACKER

hacker1

h-acker

hacker=

We can use john the ripper in Single Crack Mode as follows:

Here we have a text file named crack.txt containing the username and password, where the password is encrypted in SHA1 encryption so to crack this password we will use:

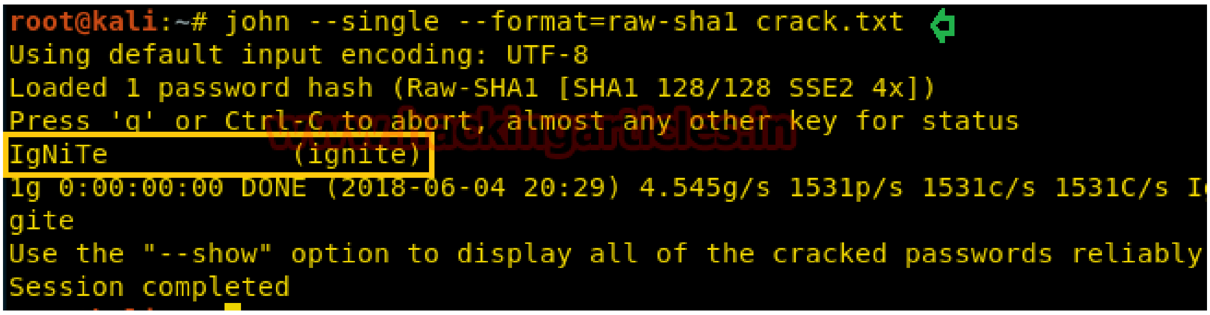
Syntax: john [mode/option] [password file]

```
john --single --format=raw-sha1 crack.txt
```

As you can see in the screenshot that we have successfully cracked the password.

Username**: ignite**

Password: **IgNiTe**



```
root@kali:~# john --single --format=raw-sha1 crack.txt
Using default input encoding: UTF-8
Loaded 1 password hash (Raw-SHA1 [SHA1 128/128 SSE2 4x])
Press 'q' or Ctrl-C to abort, almost any other key for status
IgNiTe           (ignite)
1g 0:00:00:00 DONE (2018-06-04 20:29) 4.545g/s 1531p/s 1531c/s 1531C/s I
gite
Use the "--show" option to display all of the cracked passwords reliably
Session completed
```

Wordlist Crack Mode

In this mode John the ripper uses a wordlist that can also be called a Dictionary and it compares the hashes of the words present in the Dictionary with the password hash. We can use any desired wordlist. John also comes in build with a password.lst which contains most of the common passwords.

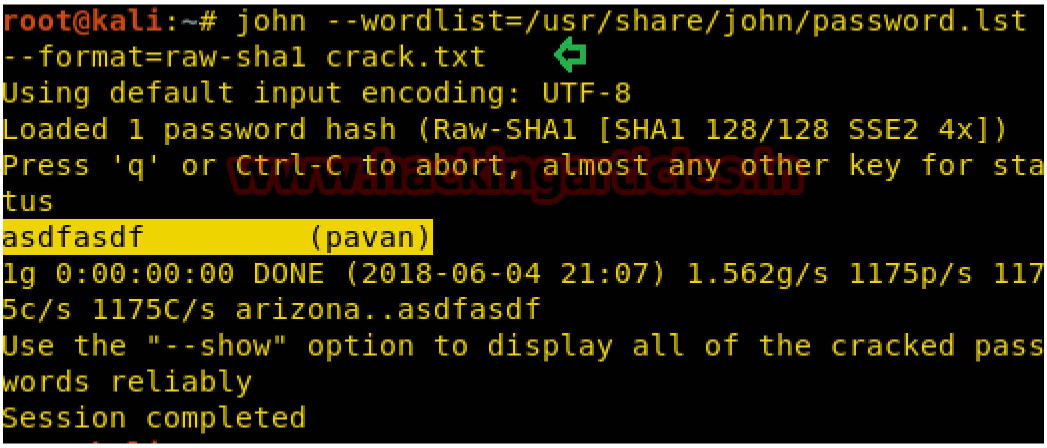
Let’s see how John the Ripper cracks passwords in Wordlist Crack Mode:

Here we have a text file named crack.txt containing the username and password, where the password is encrypted in SHA1 encryption so to crack this password we will use:

Syntax: john [wordlist] [options] [password file]

```
john --wordlist=/usr/share/john/password.lst --format=raw-sha1 crack.txt
```

As you can see in the screenshot, john the Ripper have cracked our password to be **asdfasdf**



```
root@kali:~# john --wordlist=/usr/share/john/password.lst
--format=raw-sha1 crack.txt
Using default input encoding: UTF-8
Loaded 1 password hash (Raw-SHA1 [SHA1 128/128 SSE2 4x])
Press 'q' or Ctrl-C to abort, almost any other key for sta
tus
asdfasdf         (pavan)
1g 0:00:00:00 DONE (2018-06-04 21:07) 1.562g/s 1175p/s 117
5c/s 1175C/s arizona..asdfasdf
Use the "--show" option to display all of the cracked pass
words reliably
Session completed
```

Cracking the User Credentials

We are going to demonstrate two ways in which we will crack the user credentials of a Linux user.

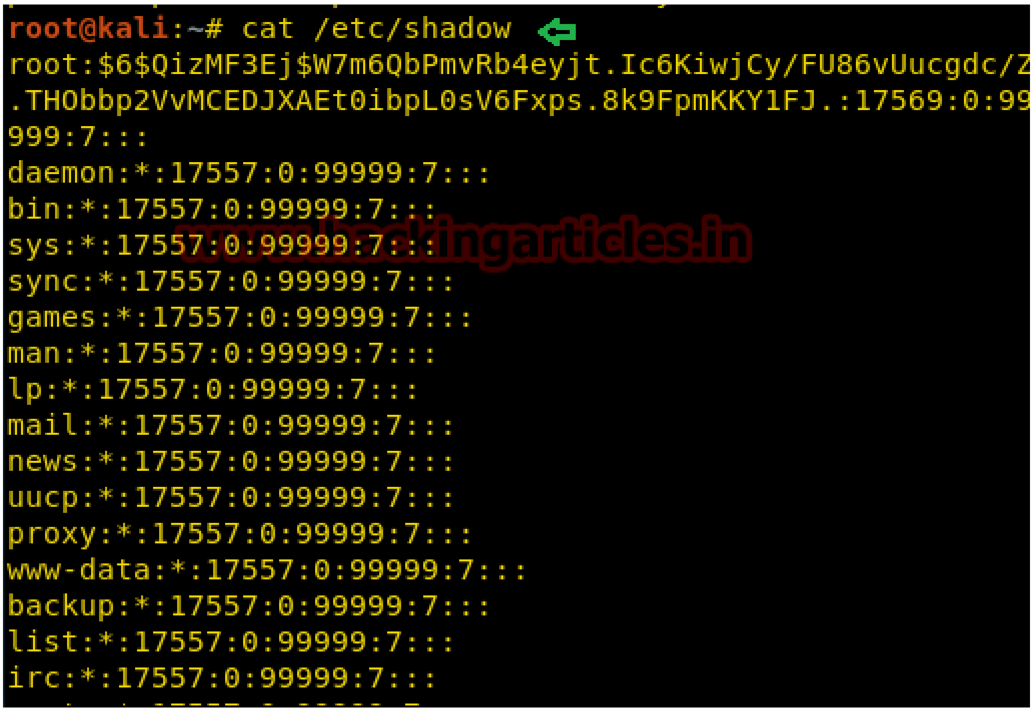
Before that we will have to understand, what is a shadow file?

In the Linux operating system, a shadow password file is a system file in which encrypted user password is stored so that they are not available to the people who try to break into the system. It is located at /etc/shadow.

First Method

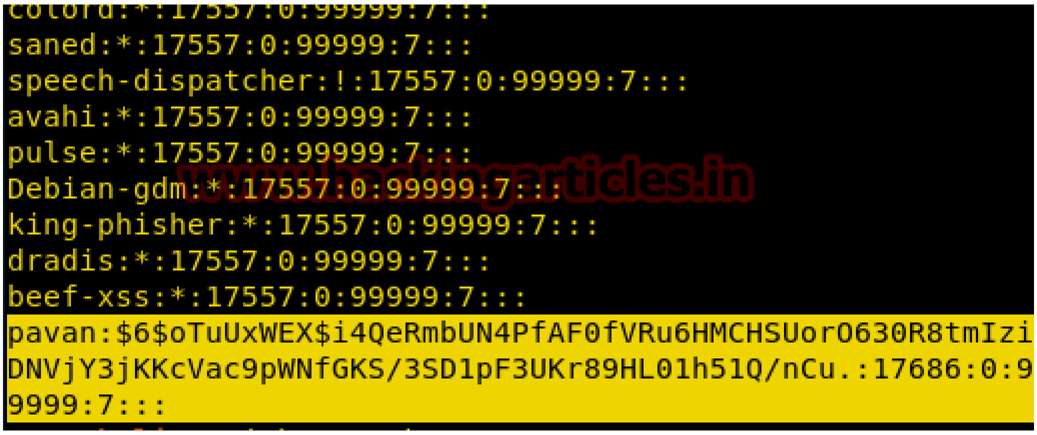
Now, for the first method, we will crack the credentials of a particular user “pavan”.

Now to do this First we will open the shadow file as shown in the image.



```
root@kali:~# cat /etc/shadow
root:$6$QizMF3Ej$W7m6QbPmvRb4eyjt.Ic6KiwjCy/FU86vUucgdc/Z
.THObbp2VvMCEDJXAEt0ibpL0sV6Fxps.8k9FpmKKY1FJ.:17569:0:99
999:7:::
daemon:*:17557:0:99999:7:::
bin:*:17557:0:99999:7:::
sys:*:17557:0:99999:7:::
sync:*:17557:0:99999:7:::
games:*:17557:0:99999:7:::
man:*:17557:0:99999:7:::
lp:*:17557:0:99999:7:::
mail:*:17557:0:99999:7:::
news:*:17557:0:99999:7:::
uucp:*:17557:0:99999:7:::
proxy:*:17557:0:99999:7:::
www-data:*:17557:0:99999:7:::
backup:*:17557:0:99999:7:::
list:*:17557:0:99999:7:::
irc:*:17557:0:99999:7:::
```

And we will find the credentials of the user pavan and copy it from here and paste it into a text file. Here we have the file named crack.txt.

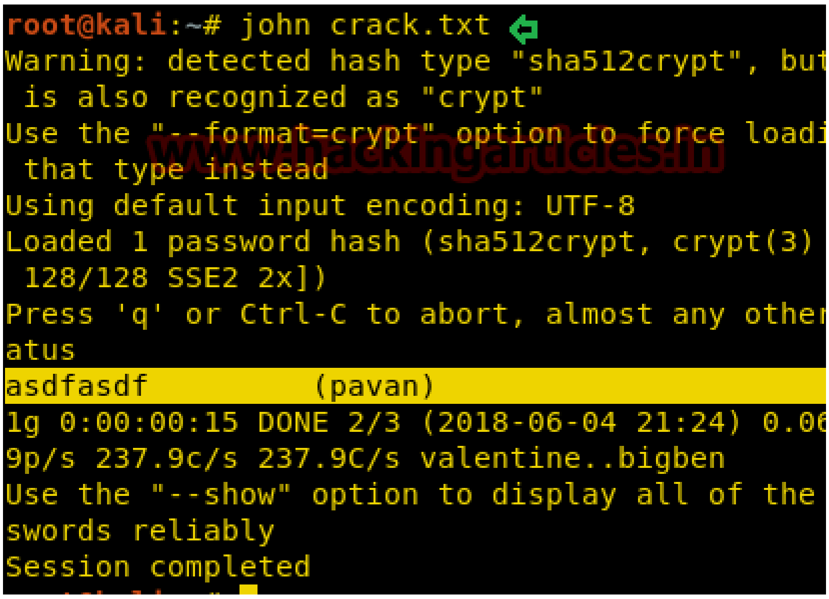


```
colord:*:17557:0:99999:7:::
saned:*:17557:0:99999:7:::
speech-dispatcher:!:17557:0:99999:7:::
avahi:*:17557:0:99999:7:::
pulse:*:17557:0:99999:7:::
Debian-gdm:*:17557:0:99999:7:::
king-phisher:*:17557:0:99999:7:::
dradis:*:17557:0:99999:7:::
beef-xss:*:17557:0:99999:7:::
pavan:$6$oTuUxWEX$i4QeRmbUN4PfAF0fVRu6HMCHSUor0630R8tmIzi
DNVjY3jKKcVac9pWNfGKS/3SD1pF3UKr89HL01h51Q/nCu.:17686:0:9
9999:7:::
```

Now we will use john the ripper to crack it.

```
john crack.txt
```

As you can see in the image below that john the ripper has successfully cracked the password for the user pavan.



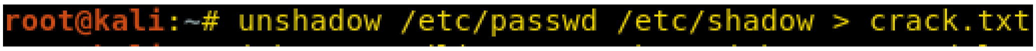
```
root@kali:~# john crack.txt
Warning: detected hash type "sha512crypt", but
 is also recognized as "crypt"
Use the "--format=crypt" option to force loadi
 that type instead
Using default input encoding: UTF-8
Loaded 1 password hash (sha512crypt, crypt(3)
 128/128 SSE2 2x])
Press 'q' or Ctrl-C to abort, almost any other
atus
asdfasdf         (pavan)
1g 0:00:00:15 DONE 2/3 (2018-06-04 21:24) 0.06
9p/s 237.9c/s 237.9C/s valentine..bigben
Use the "--show" option to display all of the
swords reliably
Session completed
```

Second Method

Now, for the second method, we will collectively crack the credentials for all the users.

To do this we will have to use John the ripper utility called “unshadow”.

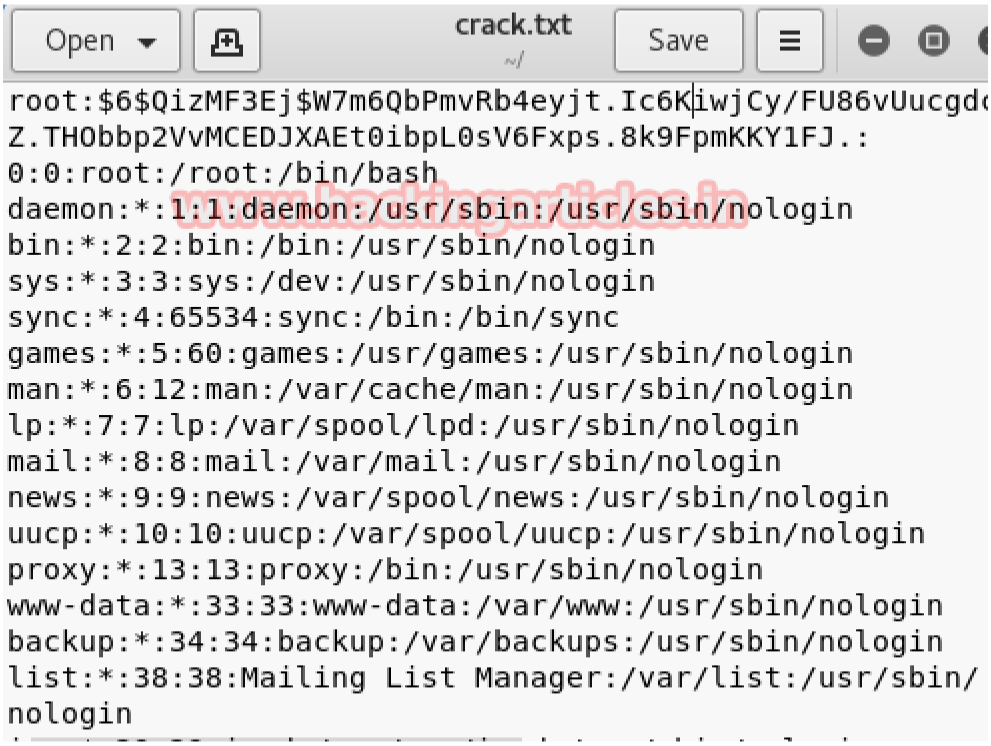
```
unshadow /etc/passwd /etc/shadow > crack.txt
```



```
root@kali:~# unshadow /etc/passwd /etc/shadow > crack.txt
```

Here the unshadow command is combining the /etc/passwd and /etc/shadow files so that John can use them to crack them. We are using both files so that John can use the information provided to efficiently crack the credentials of all users.

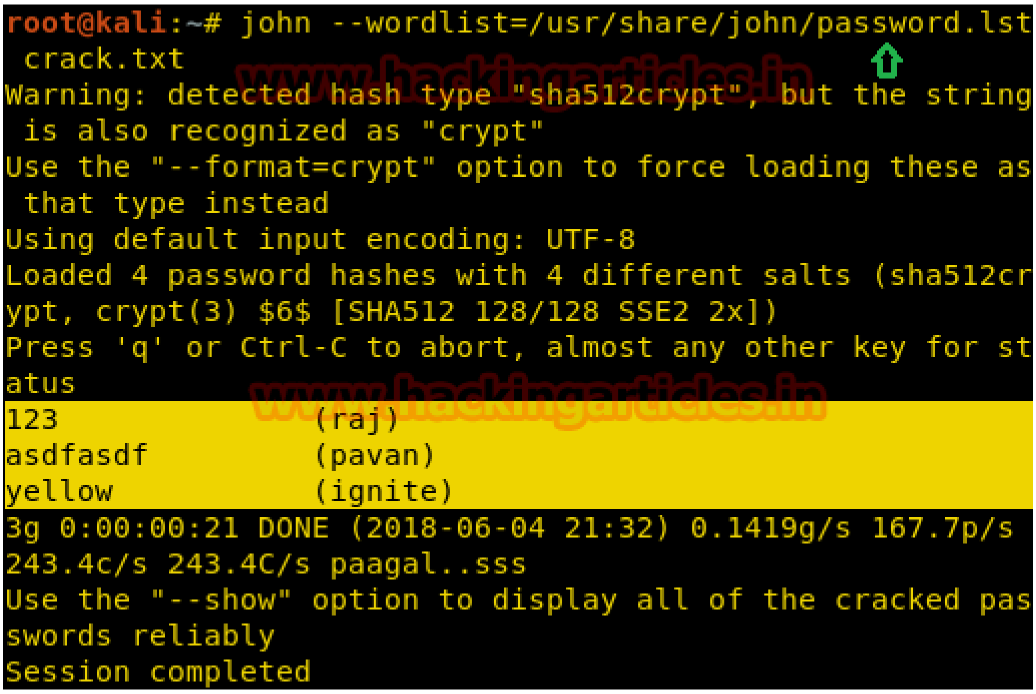
Here is how the crack file looks after unshadow command.



```
root:$6$QizMF3Ej$W7m6QbPmvRb4eyjt.Ic6KiwjCy/FU86vUucgd
Z.THObbp2VvMCEDJXAEt0ibpL0sV6Fxps.8k9FpmKKY1FJ.:
0:0:root:/root:/bin/bash
daemon:*:1:1:daemon:/usr/sbin:/usr/sbin/nologin
bin:*:2:2:bin:/bin:/usr/sbin/nologin
sys:*:3:3:sys:/dev:/usr/sbin/nologin
sync:*:4:65534:sync:/bin:/bin/sync
games:*:5:60:games:/usr/games:/usr/sbin/nologin
man:*:6:12:man:/var/cache/man:/usr/sbin/nologin
lp:*:7:7:lp:/var/spool/lpd:/usr/sbin/nologin
mail:*:8:8:mail:/var/mail:/usr/sbin/nologin
news:*:9:9:news:/var/spool/news:/usr/sbin/nologin
uucp:*:10:10:uucp:/var/spool/uucp:/usr/sbin/nologin
proxy:*:13:13:proxy:/bin:/usr/sbin/nologin
www-data:*:33:33:www-data:/var/www:/usr/sbin/nologin
backup:*:34:34:backup:/var/backups:/usr/sbin/nologin
list:*:38:38:Mailing List Manager:/var/list:/usr/sbin/
nologin
```

Now we will use john to crack the user credentials of all the users collectively.

```
john --wordlist=/usr/share/john/password.lst crack.txt
```



```
root@kali:~# john --wordlist=/usr/share/john/password.lst
 crack.txt
Warning: detected hash type "sha512crypt", but the string
 is also recognized as "crypt"
Use the "--format=crypt" option to force loading these as
 that type instead
Using default input encoding: UTF-8
Loaded 4 password hashes with 4 different salts (sha512cr
ypt, crypt(3) $6$ [SHA512 128/128 SSE2 2x])
Press 'q' or Ctrl-C to abort, almost any other key for st
atus
123              (raj)
asdfasdf         (pavan)
yellow           (ignite)
3g 0:00:00:21 DONE (2018-06-04 21:32) 0.1419g/s 167.7p/s
243.4c/s 243.4C/s paagal..sss
Use the "--show" option to display all of the cracked pas
swords reliably
Session completed
```

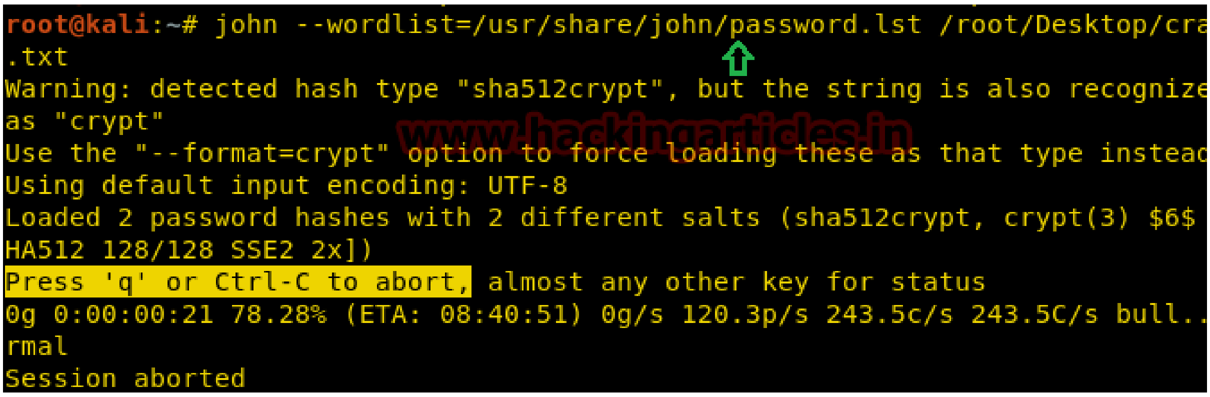
As you can see from the provided image that we have discovered the following credentials:

User	Password
Raj	123
Pavan	Asdfasdf
Ignite	Yellow

Stopping and Restoring Cracking

While John the ripper is working on cracking some passwords we can interrupt or pause the cracking and Restore or Resume the Cracking again at our convenience.

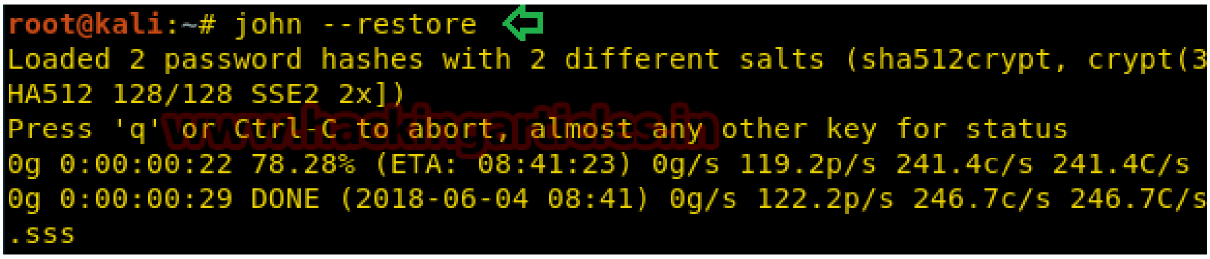
So, while John the Ripper is running you can interrupt the cracking by Pressing “q” or Crtl+C as shown in the given image.



```
root@kali:~# john --wordlist=/usr/share/john/password.lst /root/Desktop/cra
.txt
Warning: detected hash type "sha512crypt", but the string is also recognize
as "crypt"
Use the "--format=crypt" option to force loading these as that type instead
Using default input encoding: UTF-8
Loaded 2 password hashes with 2 different salts (sha512crypt, crypt(3) $6$
HA512 128/128 SSE2 2x])
Press 'q' or Ctrl-C to abort, almost any other key for status
0g 0:00:00:21 78.28% (ETA: 08:40:51) 0g/s 120.3p/s 243.5c/s 243.5C/s bull..
rmal
Session aborted
```

Now to resume or restore the cracking process we will use the –restore option of John the ripper as shown:

```
john --restore
```



```
root@kali:~# john --restore
Loaded 2 password hashes with 2 different salts (sha512crypt, crypt(3
HA512 128/128 SSE2 2x])
Press 'q' or Ctrl-C to abort, almost any other key for status
0g 0:00:00:22 78.28% (ETA: 08:41:23) 0g/s 119.2p/s 241.4c/s 241.4C/s
0g 0:00:00:29 DONE (2018-06-04 08:41) 0g/s 122.2p/s 246.7c/s 246.7C/s
.sss
```

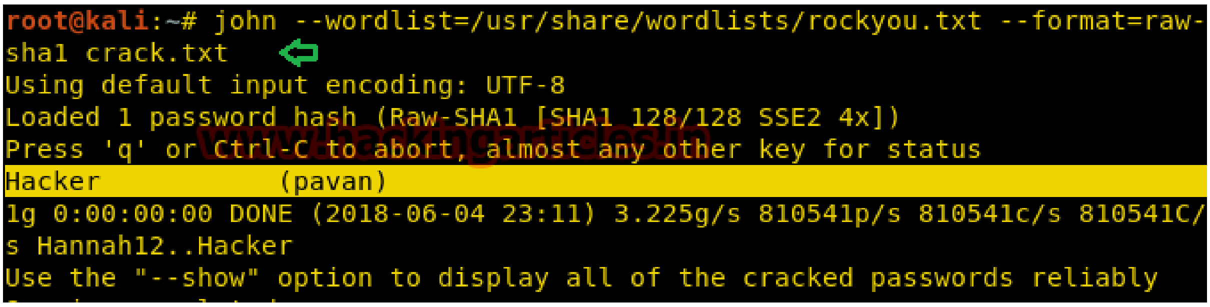
Now we will decrypt various hashes using John the Ripper

SHA1

To decrypt SHA1 encryption we will use RockYou as wordlist and crack the password as shown below:

```
john --wordlist=/usr/share/wordlists/rockyou.txt --format=raw-sha1 crack.txt
```

As you can see in the given image that we have the username pavan and password as Hacker



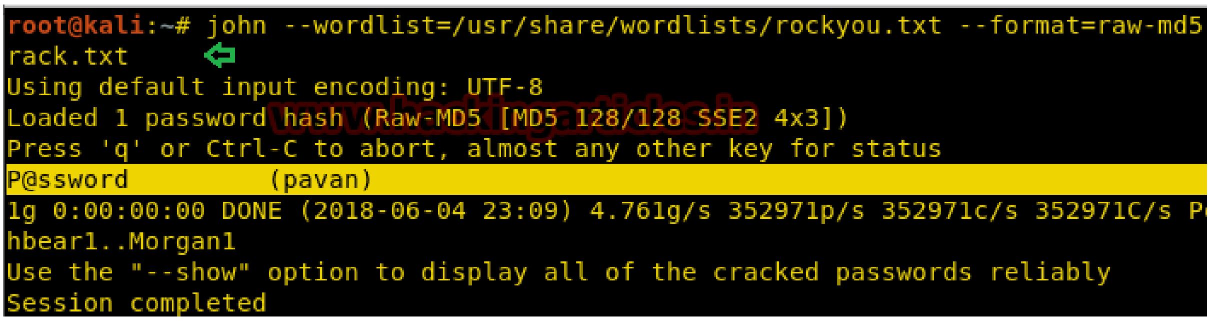
```
root@kali:~# john --wordlist=/usr/share/wordlists/rockyou.txt --format=raw-
sha1 crack.txt
Using default input encoding: UTF-8
Loaded 1 password hash (Raw-SHA1 [SHA1 128/128 SSE2 4x])
Press 'q' or Ctrl-C to abort, almost any other key for status
Hacker           (pavan)
1g 0:00:00:00 DONE (2018-06-04 23:11) 3.225g/s 810541p/s 810541c/s 810541C/
s Hannah12..Hacker
Use the "--show" option to display all of the cracked passwords reliably
```

MD5

To decrypt MD5 encryption we will use RockYou as wordlist and crack the password as shown below:

```
john --wordlist=/usr/share/wordlists/rockyou.txt --format=raw-md5 rack.txt
```

As you can see in the given screenshot that we have the username pavan and password as P@ssword.



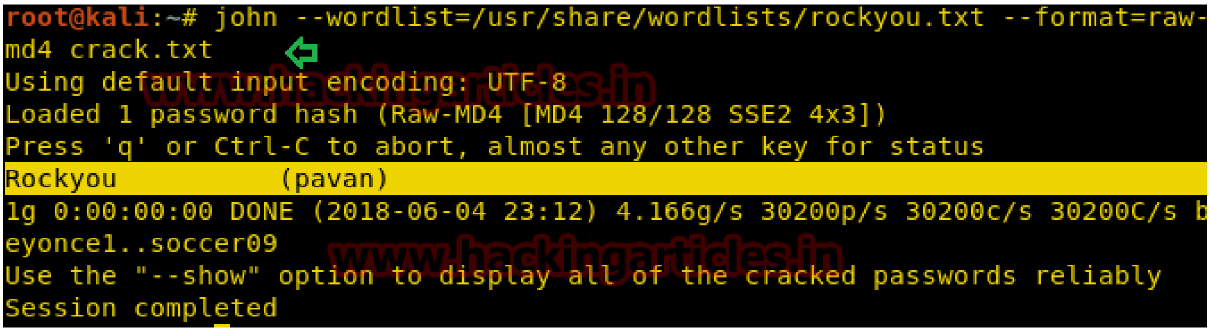
```
root@kali:~# john --wordlist=/usr/share/wordlists/rockyou.txt --format=raw-md5
rack.txt
Using default input encoding: UTF-8
Loaded 1 password hash (Raw-MD5 [MD5 128/128 SSE2 4x3])
Press 'q' or Ctrl-C to abort, almost any other key for status
P@ssword         (pavan)
1g 0:00:00:00 DONE (2018-06-04 23:09) 4.761g/s 352971p/s 352971c/s 352971C/s P
hbear1..Morgan1
Use the "--show" option to display all of the cracked passwords reliably
Session completed
```

MD4

To decrypt MD4 encryption we will use RockYou as wordlist and crack the password as shown below:

```
john --wordlist=/usr/share/wordlists/rockyou.txt --format=raw-md4 crack.txt
```

As you can see in the given screenshot that we have the username pavan and password as Rockyou



```
root@kali:~# john --wordlist=/usr/share/wordlists/rockyou.txt --format=raw-
md4 crack.txt
Using default input encoding: UTF-8
Loaded 1 password hash (Raw-MD4 [MD4 128/128 SSE2 4x3])
Press 'q' or Ctrl-C to abort, almost any other key for status
Rockyou          (pavan)
1g 0:00:00:00 DONE (2018-06-04 23:12) 4.166g/s 30200p/s 30200c/s 30200C/s b
eyonce1..soccer09
Use the "--show" option to display all of the cracked passwords reliably
Session completed
```

SHA256

To decrypt SHA256 encryption we will use RockYou as wordlist and crack the password as shown below:

```
john --wordlist=/usr/share/wordlists/rockyou.txt --format=raw-sha256 crack.txt
```

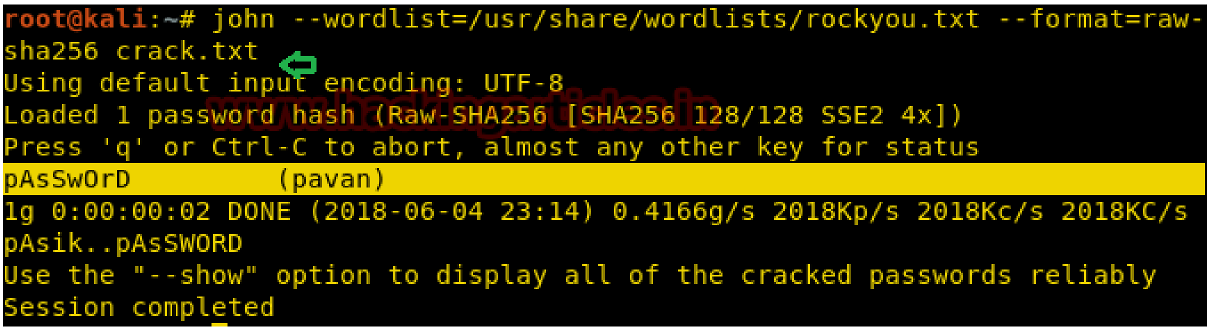
As you can see in the given screenshot that we have the username pavan and password as pAsSwOrD

RIPEMD128

To decrypt RIPEMD128 encryption we will use RockYou as wordlist and crack the password as shown below:

```
john --wordlist=/usr/share/wordlists/rockyou.txt --format=ripemd-128 crack.txt
```

As you can see in the given image that we have the username pavan and password as password123



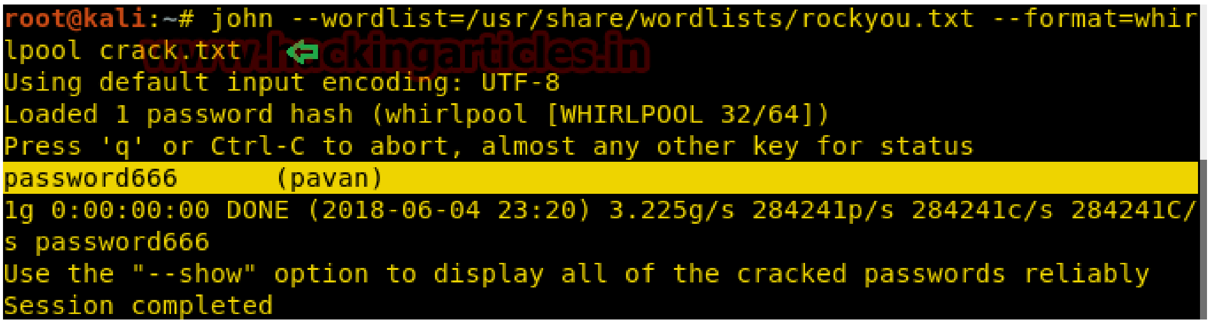
```
root@kali:~# john --wordlist=/usr/share/wordlists/rockyou.txt --format=raw-
sha256 crack.txt
Using default input encoding: UTF-8
Loaded 1 password hash (Raw-SHA256 [SHA256 128/128 SSE2 4x])
Press 'q' or Ctrl-C to abort, almost any other key for status
pAsSwOrD         (pavan)
1g 0:00:00:02 DONE (2018-06-04 23:14) 0.4166g/s 2018Kp/s 2018Kc/s 2018KC/s
pAsik..pAsSWORD
Use the "--show" option to display all of the cracked passwords reliably
Session completed
```

Whirlpool

To decrypt whirlpool encryption, we will use RockYou as wordlist and crack the password as shown below:

```
john --wordlist=/usr/share/wordlists/rockyou.txt --format=whirlpool crack.txt
```

As you can see in the given screenshot that we have the username pavan and password as password666



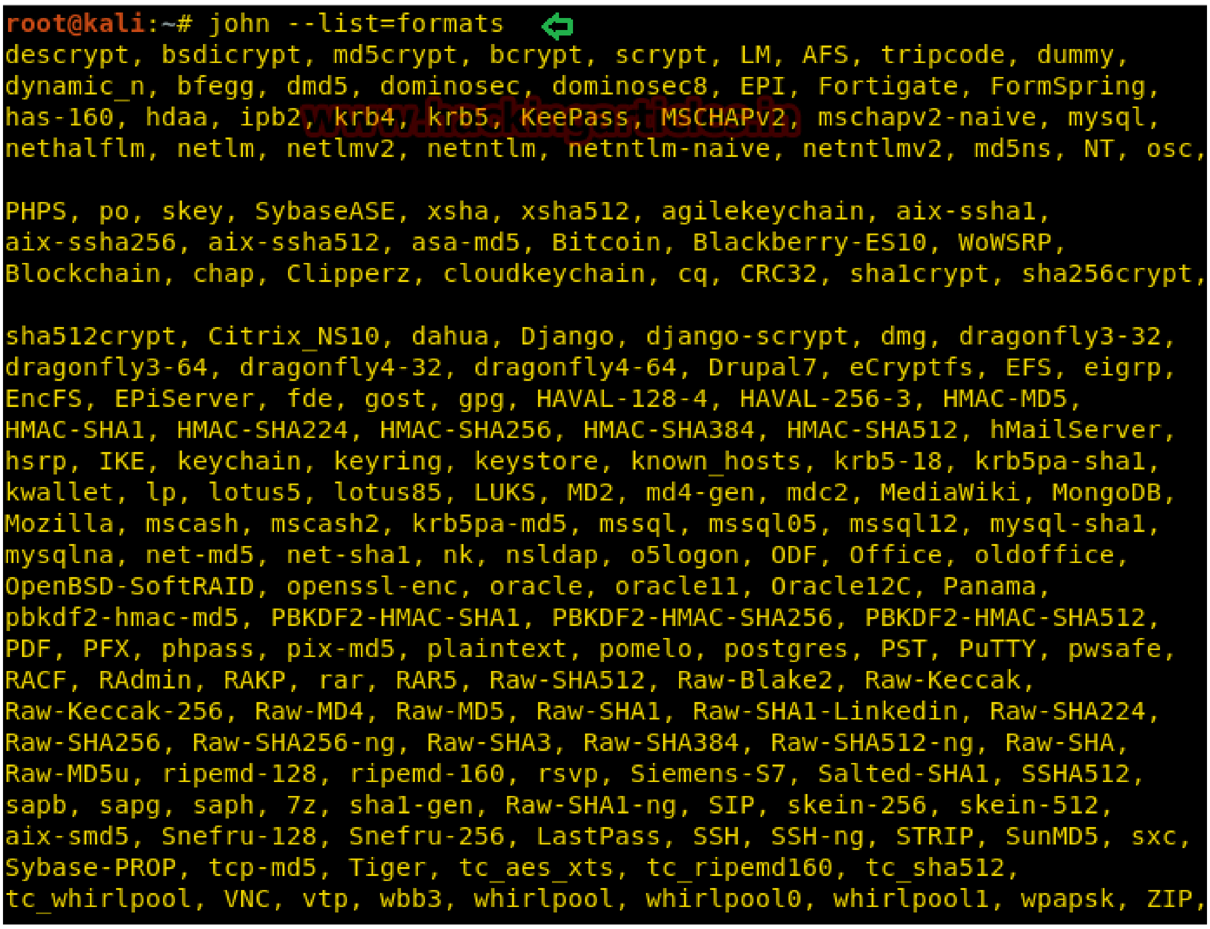
```
root@kali:~# john --wordlist=/usr/share/wordlists/rockyou.txt --format=whir
lpool crack.txt
Using default input encoding: UTF-8
Loaded 1 password hash (whirlpool [WHIRLPOOL 32/64])
Press 'q' or Ctrl-C to abort, almost any other key for status
password666      (pavan)
1g 0:00:00:00 DONE (2018-06-04 23:20) 3.225g/s 284241p/s 284241c/s 284241C/
s password666
Use the "--show" option to display all of the cracked passwords reliably
Session completed
```

View All Formats

John the Ripper supports much encryption some of which we showed above. To view all the formats, it supports:

```
john --list=formats
```

Hope, you can take reference of this article while using John the ripper, More on John the Ripper will be in the Next Part.



```
root@kali:~# john --list=formats
descrypt, bsdicrypt, md5crypt, bcrypt, scrypt, LM, AFS, tripcode, dummy,
dynamic_n, bfegg, dmd5, dominosec, dominosec8, EPI, Fortigate, FormSpring,
has-160, hdaa, ipb2, krb4, krb5, KeePass, MSCHAPv2, mschapv2-naive, mysql,
nethalflm, netlm, netlmv2, netntlm, netntlm-naive, netntlmv2, md5ns, NT, osc,

PHPS, po, skey, SybaseASE, xsha, xsha512, agilekeychain, aix-ssha1,
aix-ssha256, aix-ssha512, asa-md5, Bitcoin, Blackberry-ES10, WoWSRP,
Blockchain, chap, Clipperz, cloudkeychain, cq, CRC32, sha1crypt, sha256crypt,

sha512crypt, Citrix_NS10, dahua, Django, django-scrypt, dmg, dragonfly3-32,
dragonfly3-64, dragonfly4-32, dragonfly4-64, Drupal7, eCryptfs, EFS, eigrp,
EncFS, EPiServer, fde, gost, gpg, HAVAL-128-4, HAVAL-256-3, HMAC-MD5,
HMAC-SHA1, HMAC-SHA224, HMAC-SHA256, HMAC-SHA384, HMAC-SHA512, hMailServer,
hsrp, IKE, keychain, keyring, keystore, known_hosts, krb5-18, krb5pa-sha1,
kwallet, lp, lotus5, lotus85, LUKS, MD2, md4-gen, mdc2, MediaWiki, MongoDB,
Mozilla, mscash, mscash2, krb5pa-md5, mssql, mssql05, mssql12, mysql-sha1,
mysqlna, net-md5, net-sha1, nk, nsldap, o5logon, ODF, Office, oldoffice,
OpenBSD-SoftRAID, openssl-enc, oracle, oracle11, Oracle12C, Panama,
pbkdf2-hmac-md5, PBKDF2-HMAC-SHA1, PBKDF2-HMAC-SHA256, PBKDF2-HMAC-SHA512,
PDF, PFX, phpass, pix-md5, plaintext, pomelo, postgres, PST, PuTTY, pwsafe,
RACF, RAdmin, RAKP, rar, RAR5, Raw-SHA512, Raw-Blake2, Raw-Keccak,
Raw-Keccak-256, Raw-MD4, Raw-MD5, Raw-SHA1, Raw-SHA1-Linkedin, Raw-SHA224,
Raw-SHA256, Raw-SHA256-ng, Raw-SHA3, Raw-SHA384, Raw-SHA512-ng, Raw-SHA,
Raw-MD5u, ripemd-128, ripemd-160, rsvp, Siemens-S7, Salted-SHA1, SSHA512,
sapb, sapg, saph, 7z, sha1-gen, Raw-SHA1-ng, SIP, skein-256, skein-512,
aix-smd5, Snefru-128, Snefru-256, LastPass, SSH, SSH-ng, STRIP, SunMD5, sxc,
Sybase-PROP, tcp-md5, Tiger, tc_aes_xts, tc_ripemd160, tc_sha512,
tc_whirlpool, VNC, vtp, wbb3, whirlpool, whirlpool0, whirlpool1, wpapsk, ZIP,
```

Abbreviating the Options

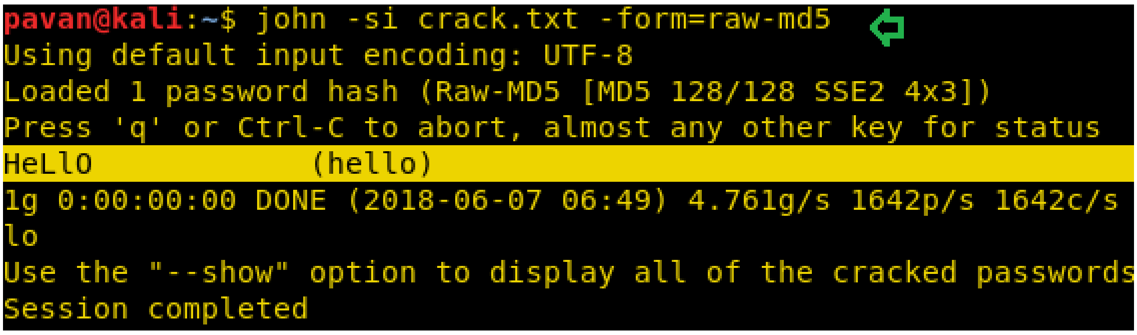
We don’t have to type complete option every time we use john the ripper, Developers have given users the option to abbreviate the options like

–single can be written as -si

–format can be written as -form

Shown below is an example of how to use these abbreviations.

```
john -si crack.txt -form=raw-md5
```

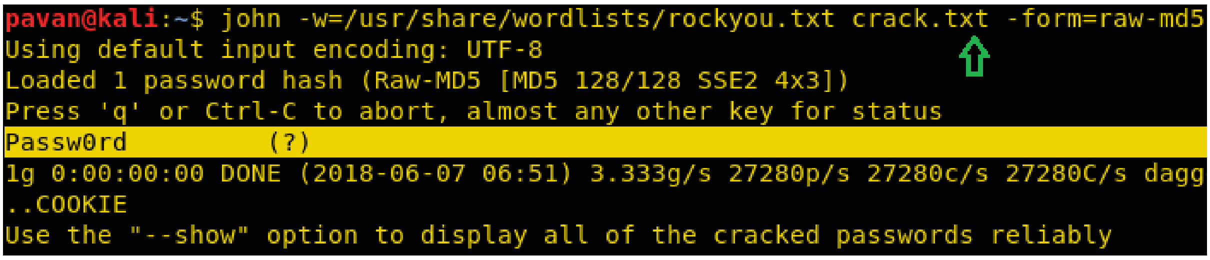


```
pavan@kali:~$ john -si crack.txt -form=raw-md5
Using default input encoding: UTF-8
Loaded 1 password hash (Raw-MD5 [MD5 128/128 SSE2 4x3])
Press 'q' or Ctrl-C to abort, almost any other key for status
HeLlO              (hello)
1g 0:00:00:00 DONE (2018-06-07 06:49) 4.761g/s 1642p/s 1642c/s
lo
Use the "--show" option to display all of the cracked passwords
Session completed
```

Another abbreviation we can use is:

–wordlist can be written as -w

```
john -w=/usr/share/wordlists/rockyou.txt crack.txt -form=raw-md5
```



```
pavan@kali:~$ john -w=/usr/share/wordlists/rockyou.txt crack.txt -form=raw-md5
Using default input encoding: UTF-8
Loaded 1 password hash (Raw-MD5 [MD5 128/128 SSE2 4x3])
Press 'q' or Ctrl-C to abort, almost any other key for status
Passw0rd          (?)
1g 0:00:00:00 DONE (2018-06-07 06:51) 3.333g/s 27280p/s 27280c/s 27280C/s dagg
..COOKIE
Use the "--show" option to display all of the cracked passwords reliably
```

Cracking Multiple Files

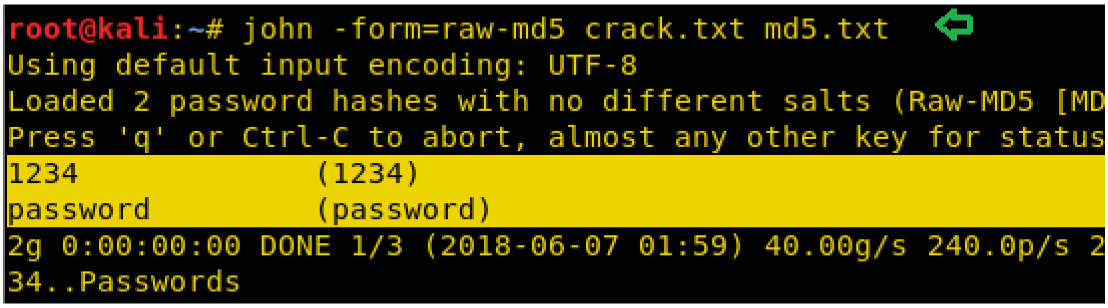
We can also crack multiple hash files if they have the same encryption. Let’s take an example, we have two files.

1. crack.txt
2. md5.txt

Both contain md5 hashes, so to crack both files in one session, we will run john as follows:

Syntax: john [file 1][file 2]

```
john -form=raw-md5 crack.txt md5.txt
```



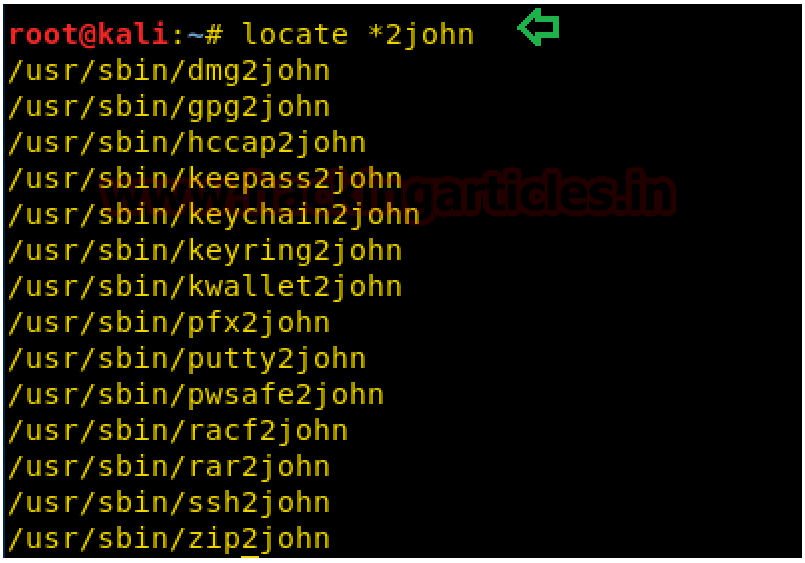
```
root@kali:~# john -form=raw-md5 crack.txt md5.txt
Using default input encoding: UTF-8
Loaded 2 password hashes with no different salts (Raw-MD5 [MD
Press 'q' or Ctrl-C to abort, almost any other key for status
1234             (1234)
password         (password)
2g 0:00:00:00 DONE 1/3 (2018-06-07 01:59) 40.00g/s 240.0p/s 2
34..Passwords
```

Password Hash Cracking

To crack these password hashes, we are going to use some of the inbuilt and some other utilities which extract the password hash from the locked file. There are some utilities that come inbuilt with John which can be found using the following command.

```
locate *2john
```

As you can see that we have the following utilities, we will demonstrate some of them here.

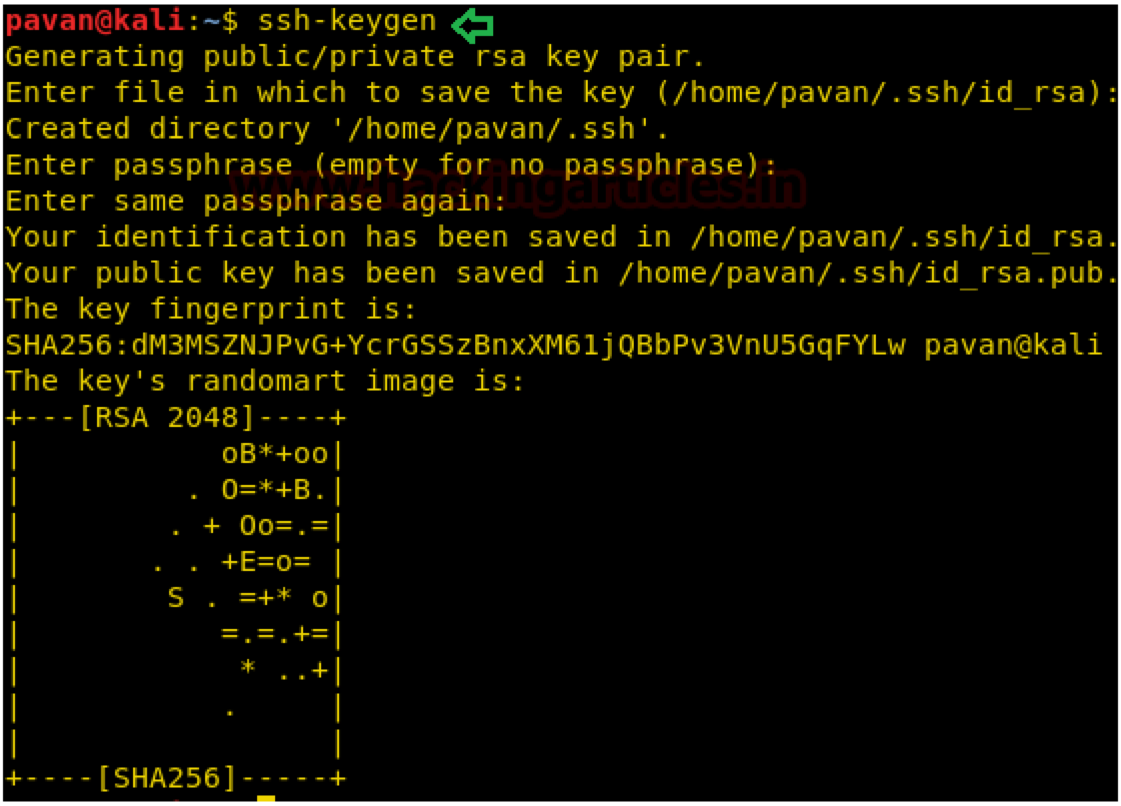


```
root@kali:~# locate *2john
/usr/sbin/dmg2john
/usr/sbin/gpg2john
/usr/sbin/hccap2john
/usr/sbin/keepass2john
/usr/sbin/keychain2john
/usr/sbin/keyring2john
/usr/sbin/kwallet2john
/usr/sbin/pfx2john
/usr/sbin/putty2john
/usr/sbin/pwsafe2john
/usr/sbin/racf2john
/usr/sbin/rar2john
/usr/sbin/ssh2john
/usr/sbin/zip2john
```

Cracking the SSH Password Hash

John the Ripper can crack the SSH private key which is created in RSA Encryption. To test the cracking of the private key, first, we will have to create a set of new private keys. To do this we will use a utility that comes with ssh, called “ssh-keygen”.

```
ssh-keygen
```

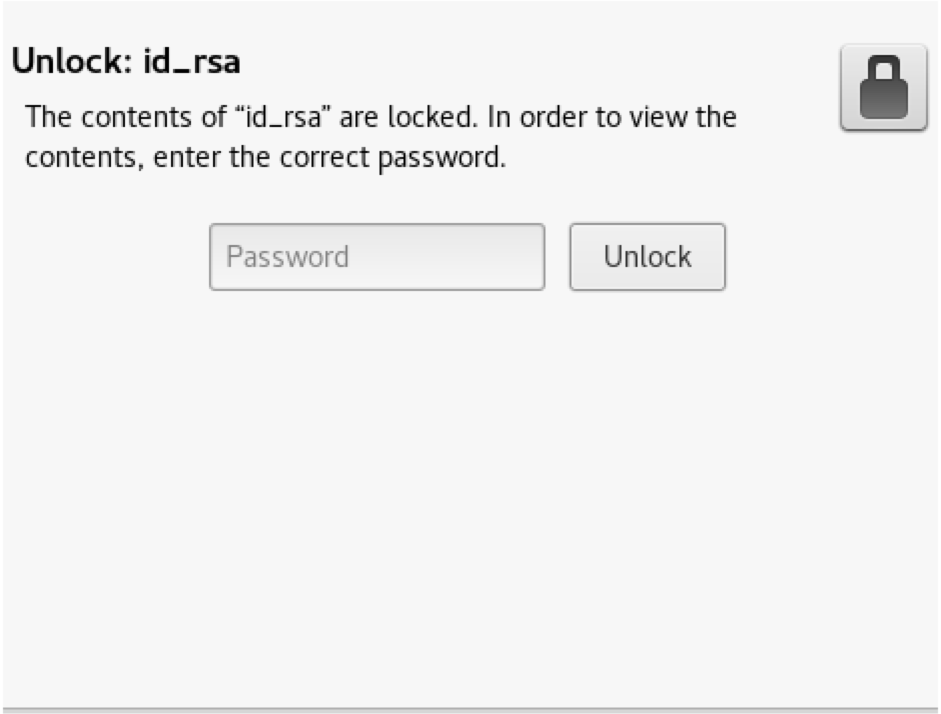


```
pavan@kali:~$ ssh-keygen
Generating public/private rsa key pair.
Enter file in which to save the key (/home/pavan/.ssh/id_rsa):
Created directory '/home/pavan/.ssh'.
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /home/pavan/.ssh/id_rsa.
Your public key has been saved in /home/pavan/.ssh/id_rsa.pub.
The key fingerprint is:
SHA256:dM3MSZNJPvG+YcrGSSzBnxXM61jQBbPv3VnU5GqFYLw pavan@kali
The key's randomart image is:
+---[RSA 2048]----+
|           oB*+oo|
|         . O=*+B.|
|        . + Oo=.=|
|       . . +E=o= |
|        S . =+* o|
|           =.=.+=|
|            * ..+|
|           .     |
|                 |
+----[SHA256]-----+
```

After opening, it asks for the location at which we want the public/private RSA key pair to store? You can use any location or you can leave it as default.

After that it asks for the passphrase, after entering the password again, we successfully generate the RSA private key. (Refer the image)

When you will try to open the file, you will be greeted by the following prompt.



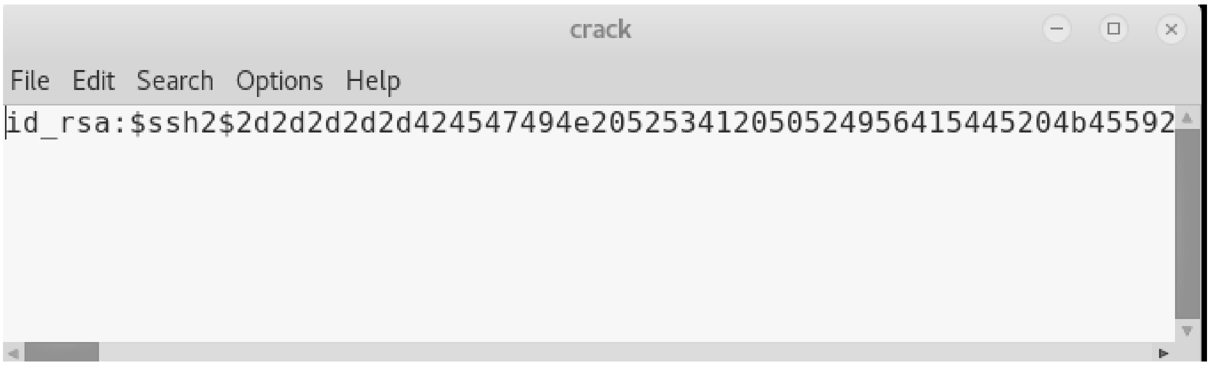
Unlock: id_rsa

The contents of “id_rsa” are locked. In order to view the contents, enter the correct password.

Now John cannot directly crack this key, first, we will have to change its format, which can be done using a john utility called “ssh2john”.

Syntax: ssh2john [location of key]

```
ssh2john /home/pavan/.ssh/id_rsa > crack.txt
```



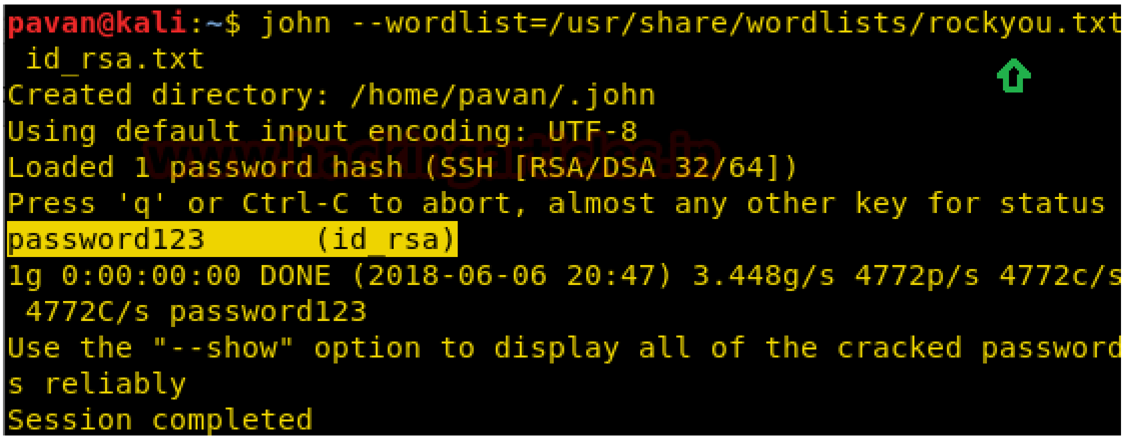
```
id_rsa:$ssh2$2d2d2d2d2d424547494e205253412050524956415445204b45592
```

You can see that we converted the key to a crackable hash and then entered it into a text file named id_rsa.txt.

Now let’s use John the Ripper to crack this hash.

```
john --wordlist=/usr/share/wordlists/rockyou.txt id_rsa.txt
```

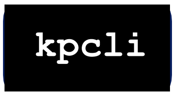
Great! We have successfully cracked the passphrase used to create the private ssh key to be "password123"



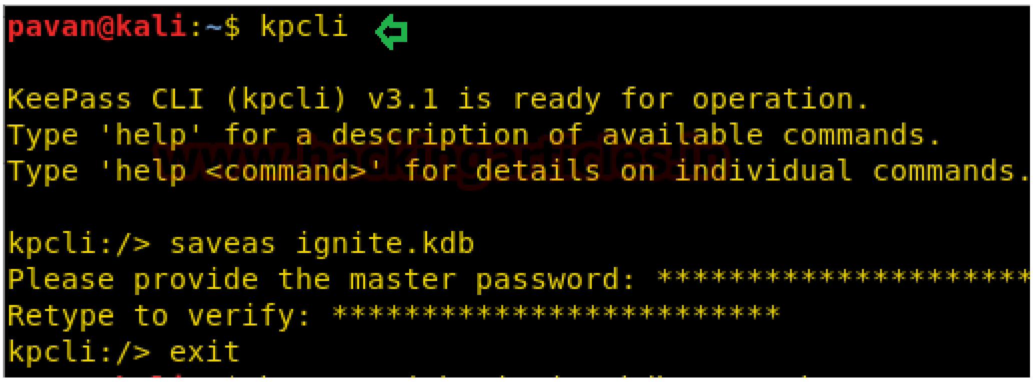
```
pavan@kali:~$ john --wordlist=/usr/share/wordlists/rockyou.txt
 id_rsa.txt
Created directory: /home/pavan/.john
Using default input encoding: UTF-8
Loaded 1 password hash (SSH [RSA/DSA 32/64])
Press 'q' or Ctrl-C to abort, almost any other key for status
password123      (id_rsa)
1g 0:00:00:00 DONE (2018-06-06 20:47) 3.448g/s 4772p/s 4772c/s
 4772C/s password123
Use the "--show" option to display all of the cracked password
s reliably
Session completed
```

Cracking the KeepPass2 Password Hash

John the Ripper can crack the KeepPass2 key. To test the cracking of the key, first, we will have to create a set of new keys. To do this we will use a utility that is called "kpcli".



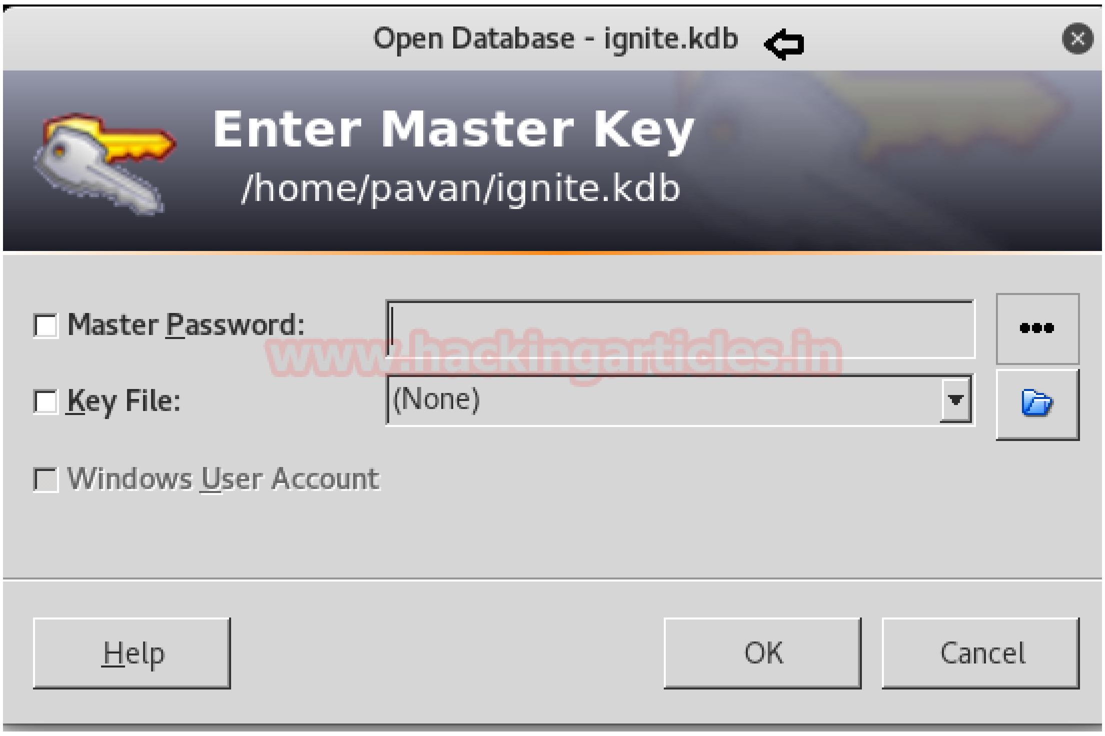
```
kpcli
```



```
pavan@kali:~$ kpcli

KeePass CLI (kpcli) v3.1 is ready for operation.
Type 'help' for a description of available commands.
Type 'help <command>' for details on individual commands.

kpcli:/> saveas ignite.kdb
Please provide the master password: *********************
Retype to verify: *************************
kpcli:/> exit
```

Now we will create a database file using the command "save as" and naming the database file as ignite.kdb and entering a passcode to secure it.

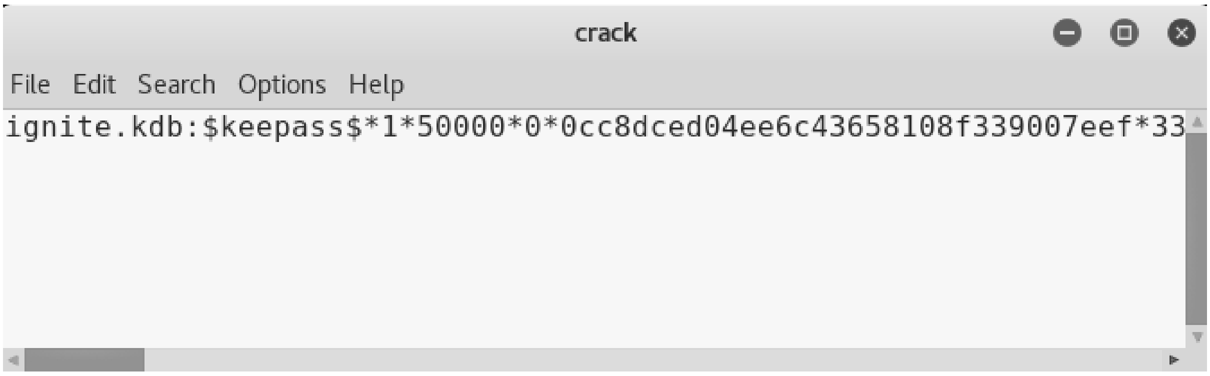
When you will try to open the file, you will be greeted by the following prompt.



Now John cannot directly crack this key, first, we will have to change its format, which can be done using a john utility called "keepass2john".

Syntax: keepass2john [location of key]

```
keepass2john ignite.kdb > crack.txt
```

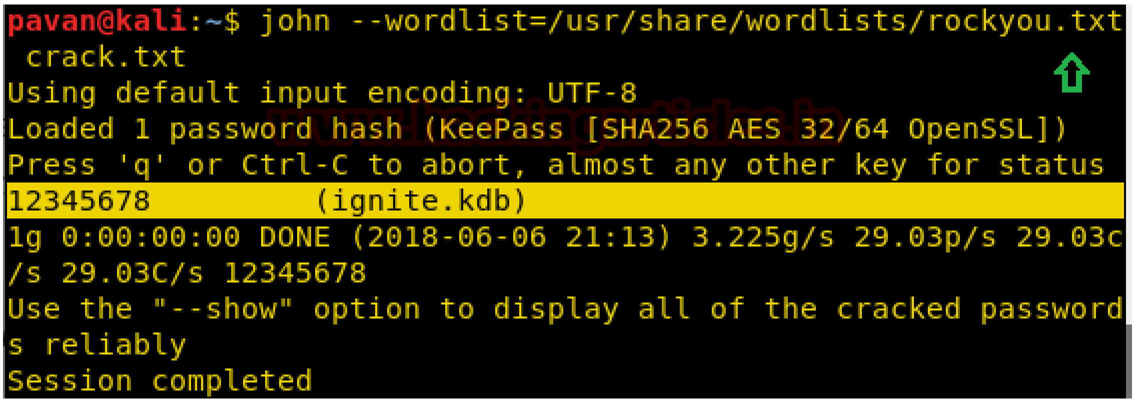


```
ignite.kdb:$keepass$*1*50000*0*0cc8dced04ee6c43658108f339007eef*33
```

Now let's use John the Ripper to crack this hash.

```
john --wordlist=/usr/share/wordlists/rockyou.txt crack.txt
```

Great! We have successfully cracked the passphrase used to create the key to be "12345678"



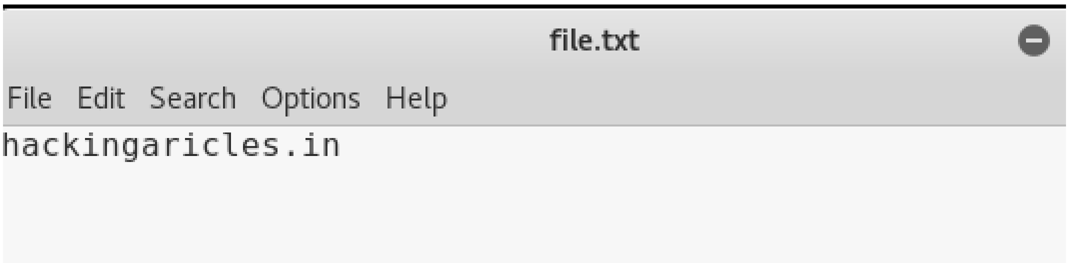
```
pavan@kali:~$ john --wordlist=/usr/share/wordlists/rockyou.txt
 crack.txt
Using default input encoding: UTF-8
Loaded 1 password hash (KeePass [SHA256 AES 32/64 OpenSSL])
Press 'q' or Ctrl-C to abort, almost any other key for status
12345678         (ignite.kdb)
1g 0:00:00:00 DONE (2018-06-06 21:13) 3.225g/s 29.03p/s 29.03c
/s 29.03C/s 12345678
Use the "--show" option to display all of the cracked password
s reliably
Session completed
```

Cracking the RAR Password Hash

Now we will crack some compressed files, to do that we will have to create a file to be compressed so let’s do that using echo command as shown in the given screenshot.

You can see that we created a file.txt which we will be using to create compressed files.

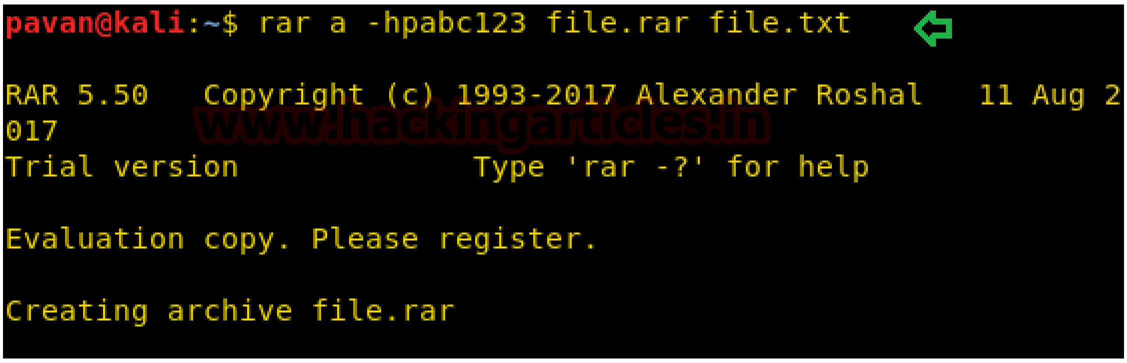
```
echo hackingarticles.in > file.txt
```



```
hackingaricles.in
```

John the Ripper can crack the RAR file passwords. To test the cracking of the password, first, let’s create a compressed encrypted rar file.

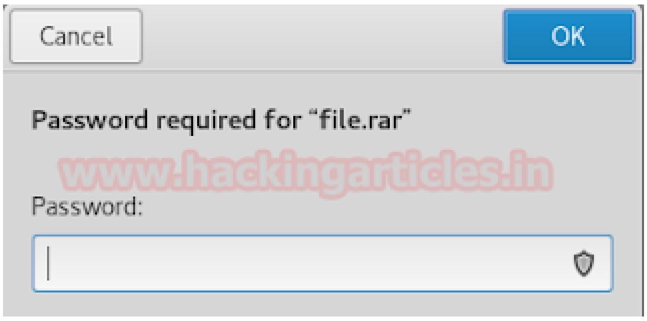
```
rar a -hpabc123 file.rar file.txt
```



```
pavan@kali:~$ rar a -hpabc123 file.rar file.txt

RAR 5.50   Copyright (c) 1993-2017 Alexander Roshal   11 Aug 2
017
Trial version             Type 'rar -?' for help

Evaluation copy. Please register.

Creating archive file.rar
```

- a = Add files to archive
- hp[password] = Encrypt both file data and headers

This will compress and encrypt our file.txt into a file.rar. So, when you will try to open the file, you will be greeted by the following prompt.

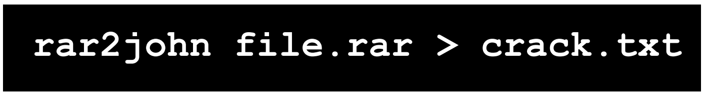


Password required for "file.rar"

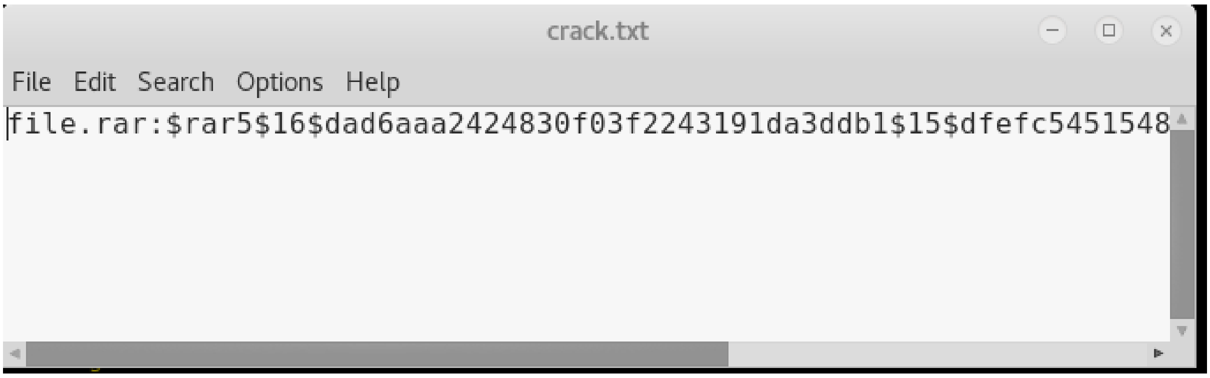
Password:

Now John cannot directly crack this key, first, we will have to change it format, which can be done using a john utility called "rar2john".

Syntax: rar2john [location of key]



```
rar2john file.rar > crack.txt
```

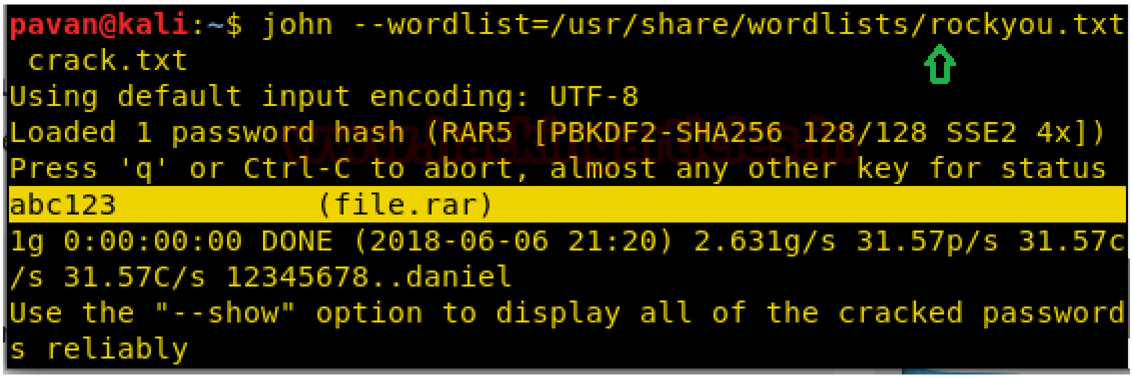


```
file.rar:$rar5$16$dad6aaa2424830f03f2243191da3ddb1$15$dfefc5451548
```

Now let's use John the Ripper to crack this hash.

```
john --wordlist=/usr/share/wordlists/rockyou.txt crack.txt
```

Great! We have successfully cracked the passphrase used to create the key to be "abc123"

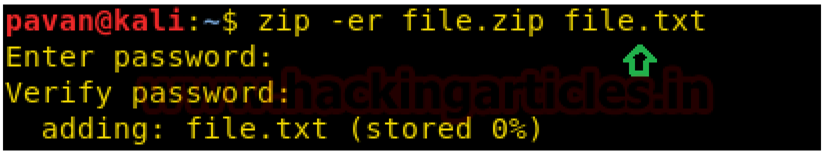


```
pavan@kali:~$ john --wordlist=/usr/share/wordlists/rockyou.txt
 crack.txt
Using default input encoding: UTF-8
Loaded 1 password hash (RAR5 [PBKDF2-SHA256 128/128 SSE2 4x])
Press 'q' or Ctrl-C to abort, almost any other key for status
abc123           (file.rar)
1g 0:00:00:00 DONE (2018-06-06 21:20) 2.631g/s 31.57p/s 31.57c
/s 31.57C/s 12345678..daniel
Use the "--show" option to display all of the cracked password
s reliably
```

Cracking the ZIP Password Hash

John the Ripper can crack the ZIP file passwords. To test the cracking of the password, first, let's create a compressed encrypted zip file.

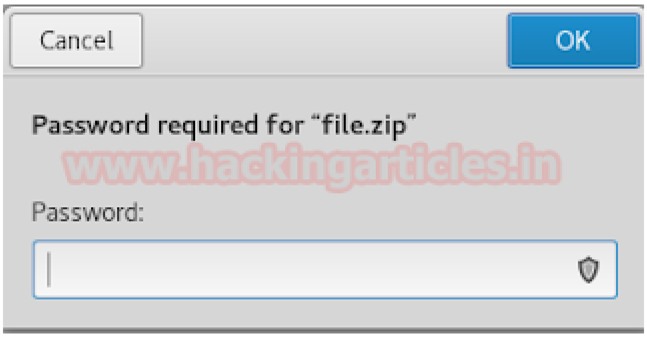
```
zip -er file.zip file.txt
```



```
pavan@kali:~$ zip -er file.zip file.txt
Enter password:
Verify password:
  adding: file.txt (stored 0%)
```

- e = Encrypt
- r = Recurse into directories

This will compress and encrypt our file.txt into a file.zip. So, when you will try to open the file, you will be greeted by the following prompt.



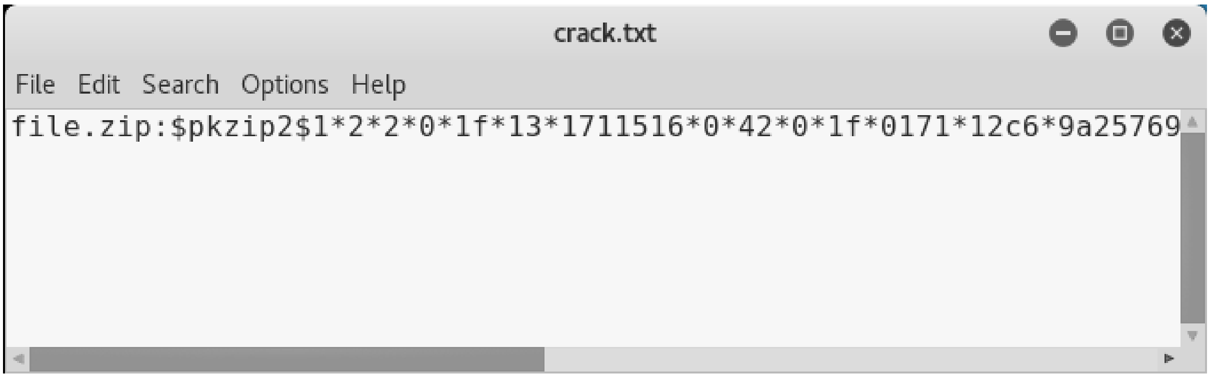
Password required for “file.zip”

Password:

Now John cannot directly crack this key, first, we will have to change its format, which can be done using a john utility called “zip2john”.

Syntax: zip2john [location of key]

```
zip2john file.zip > crack.txt
```

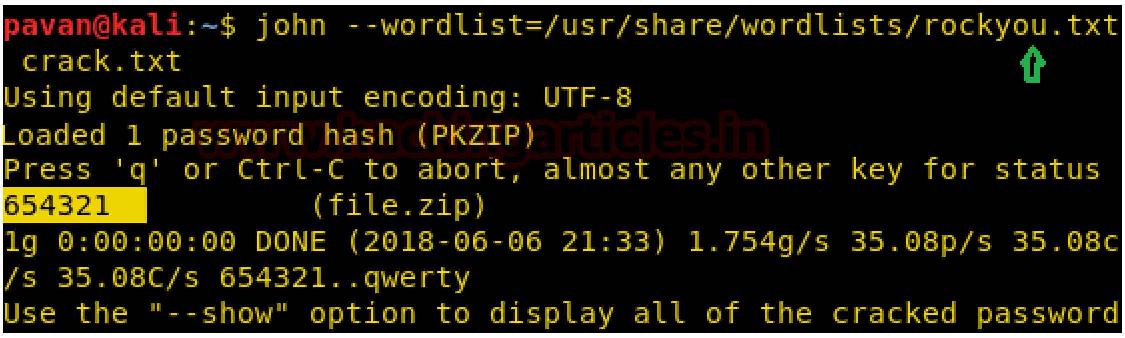


```
file.zip:$pkzip2$1*2*2*0*1f*13*1711516*0*42*0*1f*0171*12c6*9a25769
```

Now let’s use John the Ripper to crack this hash.

```
john --wordlist=/usr/share/wordlists/rockyou.txt crack.txt
```

Great! We have successfully cracked the passphrase used to create the key to be “654321”

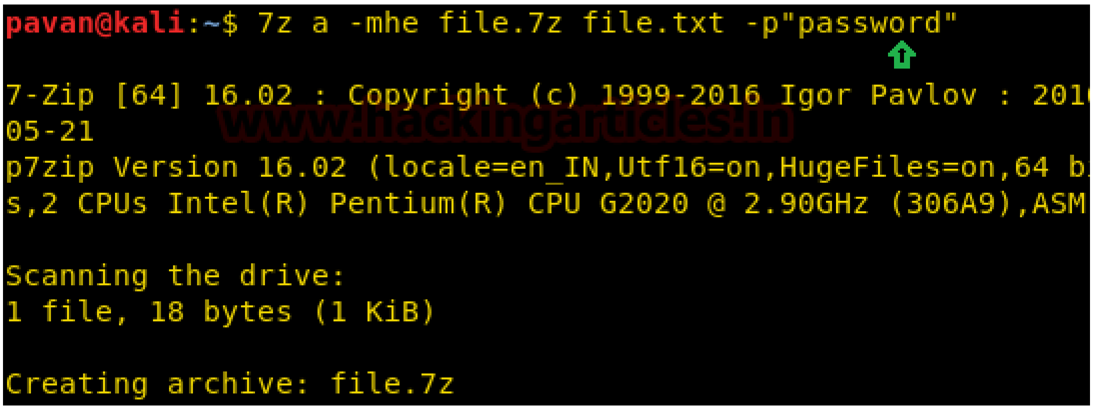


```
pavan@kali:~$ john --wordlist=/usr/share/wordlists/rockyou.txt
 crack.txt
Using default input encoding: UTF-8
Loaded 1 password hash (PKZIP)
Press 'q' or Ctrl-C to abort, almost any other key for status
654321           (file.zip)
1g 0:00:00:00 DONE (2018-06-06 21:33) 1.754g/s 35.08p/s 35.08c
/s 35.08C/s 654321..qwerty
Use the "--show" option to display all of the cracked password
```

Cracking the 7-Zip Password Hash

John the Ripper can crack the 7-Zip file passwords. To test the cracking of the password, first, let's create a compressed encrypted 7z file.

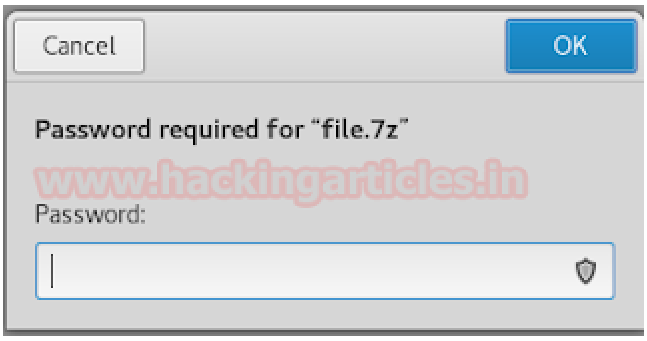
```
7z a -mhe file.7z file.txt -p"password"
```



```
pavan@kali:~$ 7z a -mhe file.7z file.txt -p"password"

7-Zip [64] 16.02 : Copyright (c) 1999-2016 Igor Pavlov : 201
05-21
p7zip Version 16.02 (locale=en_IN,Utf16=on,HugeFiles=on,64 b
s,2 CPUs Intel(R) Pentium(R) CPU G2020 @ 2.90GHz (306A9),ASM

Scanning the drive:
1 file, 18 bytes (1 KiB)

Creating archive: file.7z
```

- a = Add files to archive
- m = Set compression Method
- h = Calculate hash values for files
- e = Encrypt file
- p = set Password

This will compress and encrypt our file.txt into a file.7z. So, when you will try to open the file, you will be greeted by the following prompt.



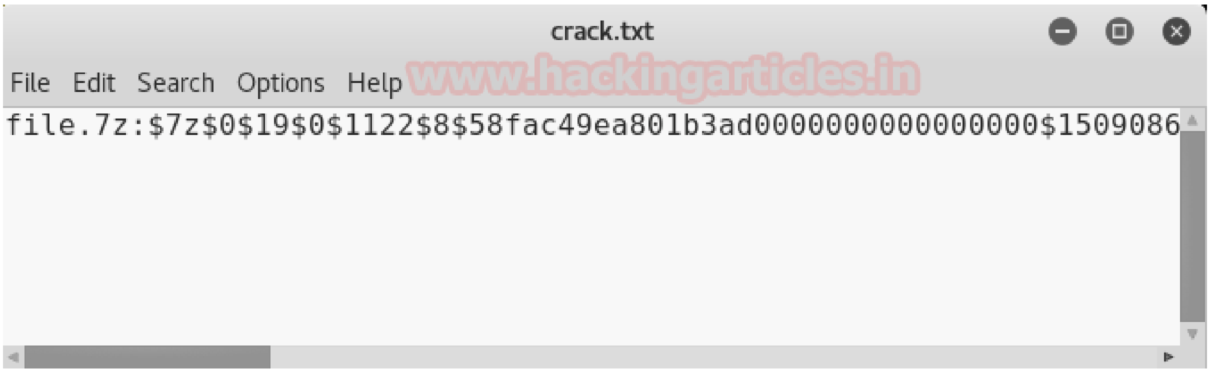
Password required for "file.7z"

Password:

Now John cannot directly crack this key, first, we will change its format, which can be done using a john utility called "7z2john". This is not inbuilt utility, It can be downloaded from the Github Repo.

Syntax: zip2john [location of key]

```
python 7z2john.py file.7z > crack.txt
```

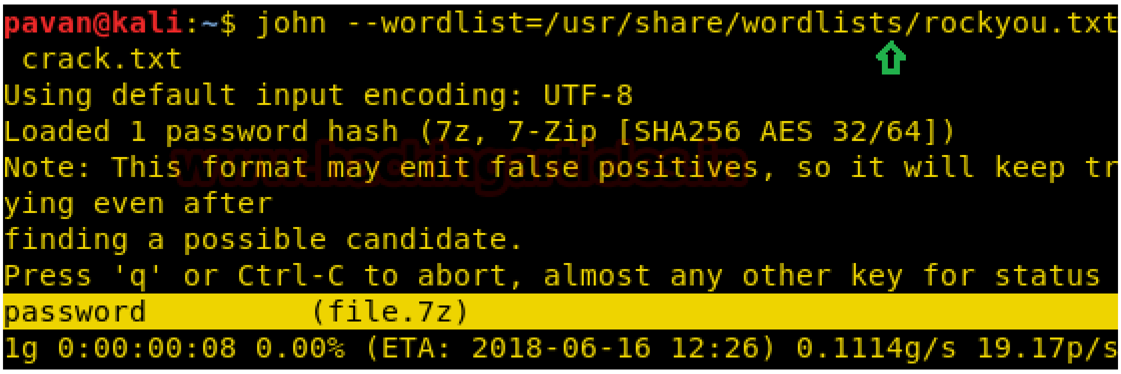


```
file.7z:$7z$0$19$0$1122$8$58fac49ea801b3ad0000000000000000$1509086
```

Now let’s use John the Ripper to crack this hash.

```
john –wordlist=/usr/share/wordlists/rockyou.txt crack.txt
```

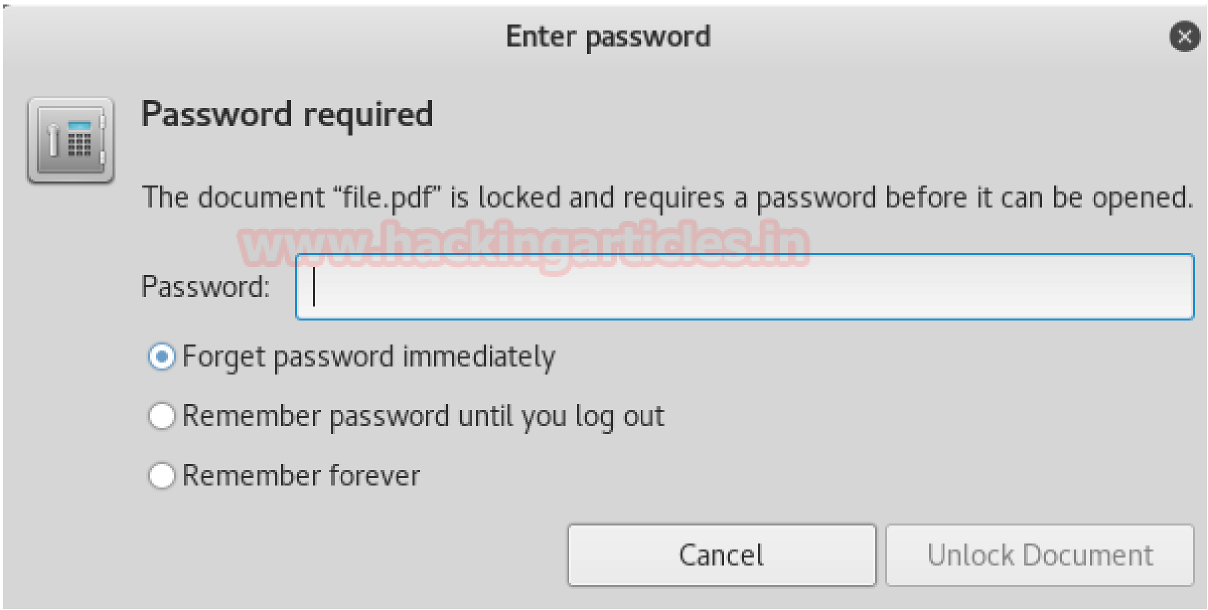
Great! We have successfully cracked the passphrase used to create the key to be “password”



```
pavan@kali:~$ john --wordlist=/usr/share/wordlists/rockyou.txt
 crack.txt
Using default input encoding: UTF-8
Loaded 1 password hash (7z, 7-Zip [SHA256 AES 32/64])
Note: This format may emit false positives, so it will keep tr
ying even after
finding a possible candidate.
Press 'q' or Ctrl-C to abort, almost any other key for status
password         (file.7z)
1g 0:00:00:08 0.00% (ETA: 2018-06-16 12:26) 0.1114g/s 19.17p/s
```

Cracking the PDF Password Hash

John the Ripper can crack the PDF file passwords. You can encrypt your pdf online by using Soda PDF website. This will compress and encrypt our pdf into a password protected file.pdf. So, when you will try to open the file, you will be greeted by the following prompt.



Enter password

Password required

The document “file.pdf” is locked and requires a password before it can be opened.

Password:

● Forget password immediately

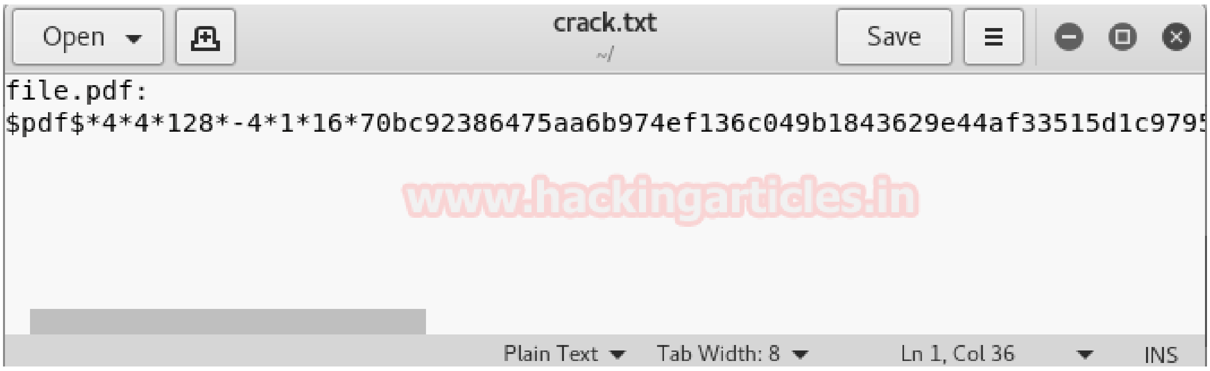
○ Remember password until you log out

○ Remember forever

Now John cannot directly crack this key, first, we will have to change its format, which can be done using a john utility called “pdf2john”. This is not an inbuilt utility, it can be downloaded from the link provided among the references.

Syntax: pdf2john [location of key]

```
python pdf2john.py file.pdf > crack.txt
```

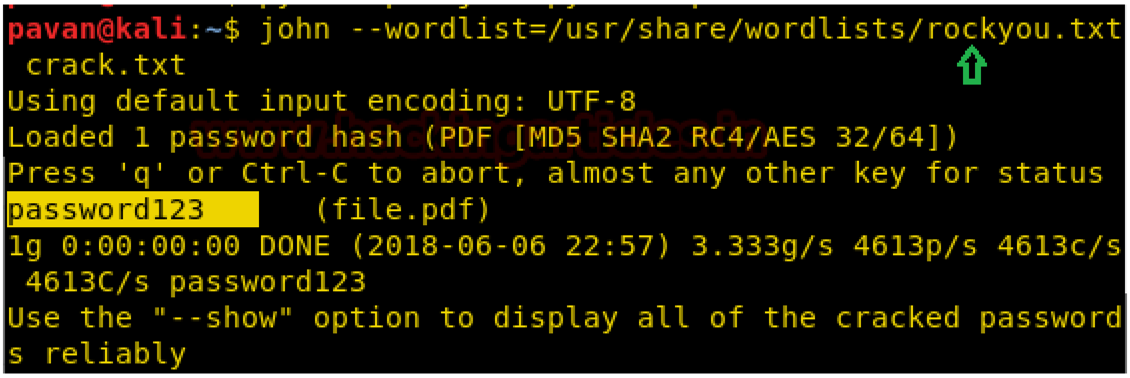


```
file.pdf:
$pdf$*4*4*128*-4*1*16*70bc92386475aa6b974ef136c049b1843629e44af33515d1c9795
```

Now let’s use John the Ripper to crack this hash.

```
john -wordlist=/usr/share/wordlists/rockyou.txt crack.txt
```

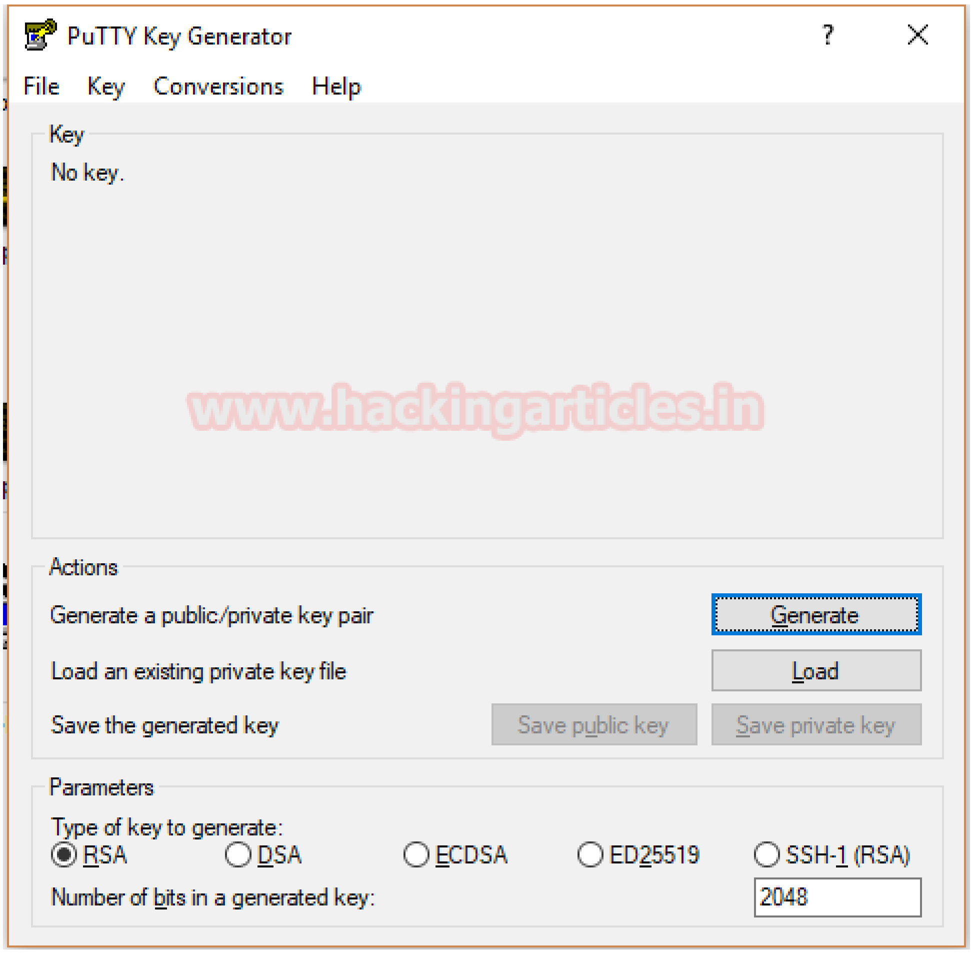
Great! We have successfully cracked the passphrase used to create the key to be “password123”.



```
pavan@kali:~$ john --wordlist=/usr/share/wordlists/rockyou.txt
 crack.txt
Using default input encoding: UTF-8
Loaded 1 password hash (PDF [MD5 SHA2 RC4/AES 32/64])
Press 'q' or Ctrl-C to abort, almost any other key for status
password123      (file.pdf)
1g 0:00:00:00 DONE (2018-06-06 22:57) 3.333g/s 4613p/s 4613c/s
 4613C/s password123
Use the "--show" option to display all of the cracked password
s reliably
```

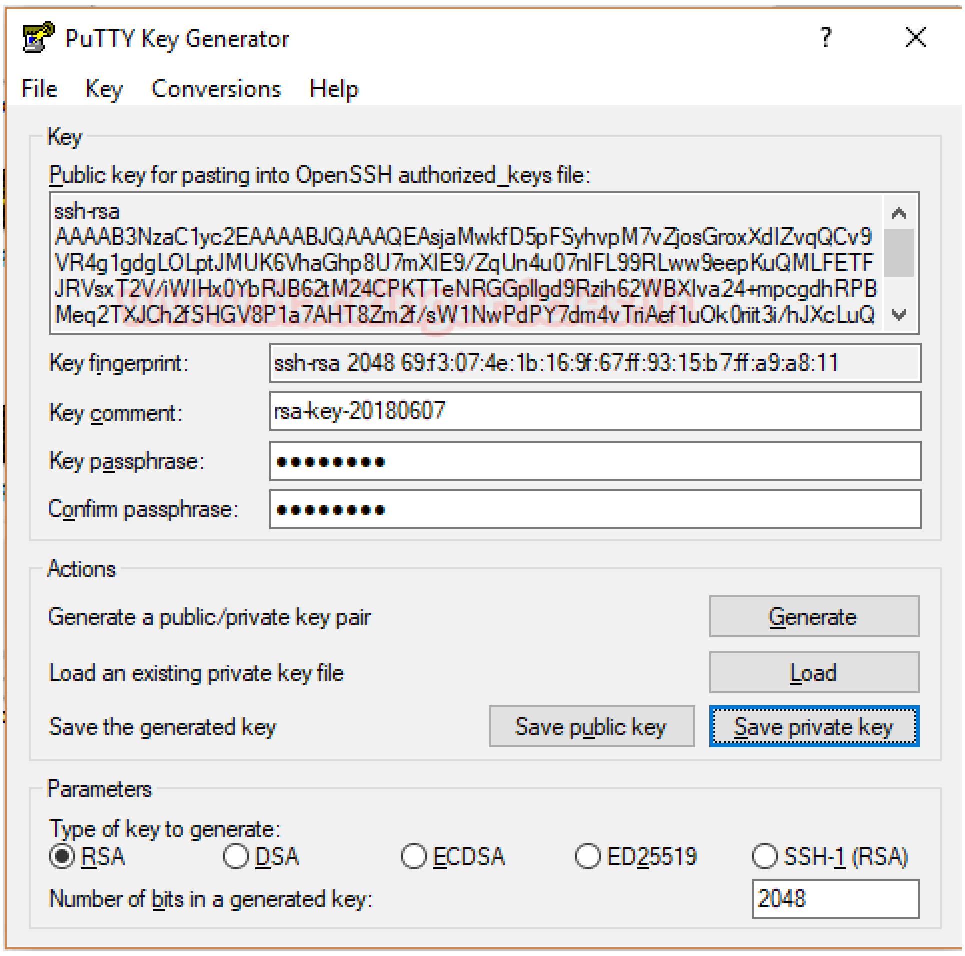
Cracking the PuTTY Password Hash

John the Ripper can crack the PuTTY private key which is created in RSA Encryption. To test the cracking of the private key, first, we will have to create a set of new private keys. To do this we will use a utility that comes with PuTTY, called “PuTTY Key Generator”.



Type of key to generate:
● RSA ○ DSA ○ ECDSA ○ ED25519 ○ SSH-1 (RSA)
Number of bits in a generated key: 2048

Click on “Generate”. After Generating the key, we get a window where we will input the key passphrase as shown in the image.



Public key for pasting into OpenSSH authorized_keys file:

```
ssh-rsa
AAAAB3NzaC1yc2EAAAABJQAAAQEAsjaMwkfD5pFSyhvpM7vZjosGroxXdIZvqQCv9
VR4g1gdgLOLptJMUK6VhaGhp8U7mXIE9/ZqUn4u07nIFL99RLww9eepKuQMLFETF
JRVsxT2V/iWIHx0YbRJB62tM24CPKTTeNRGGpllgd9Rzih62WBXIva24+mpcgdhRPB
Meq2TXJCh2fSHGV8P1a7AHT8Zm2f/sW1NwPdPY7dm4vTriAef1uOk0nit3i/hJXcLuQ
```

Key fingerprint: ssh-rsa 2048 69:f3:07:4e:1b:16:9f:67:ff:93:15:b7:ff:a9:a8:11

Key comment: rsa-key-20180607

Key passphrase: ●●●●●●●●

Confirm passphrase: ●●●●●●●●

Generate a public/private key pair — Generate

Load an existing private key file — Load

Save the generated key — Save public key — Save private key

Type of key to generate:
● RSA ○ DSA ○ ECDSA ○ ED25519 ○ SSH-1 (RSA)

Number of bits in a generated key: 2048

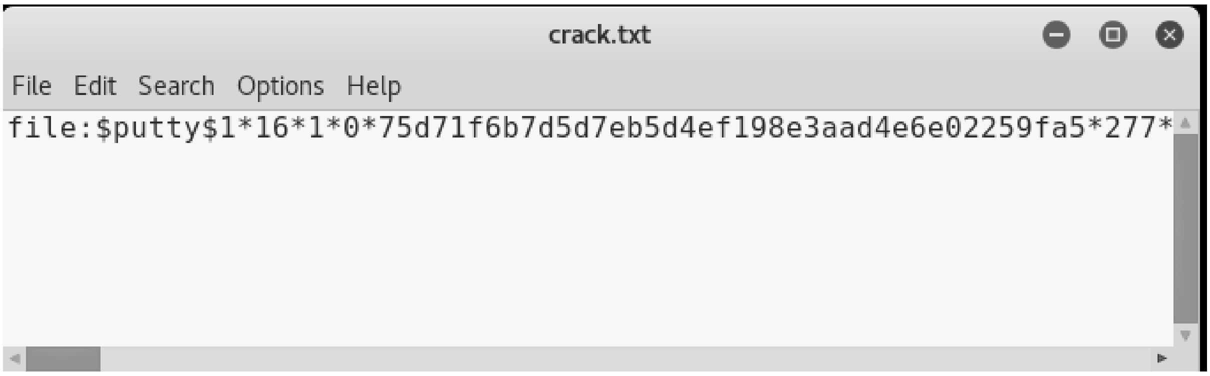
After entering the passphrase, click on Save private key to get a private key in the form of a .ppk file

After generating transfer this .ppk file to Kali Linux.

Now John cannot directly crack this key, first, we will have to change its format, which can be done using a john utility called “putty2john”.

Syntax: putty2john [location of key]

```
putty2john file.ppk > crack.txt
```



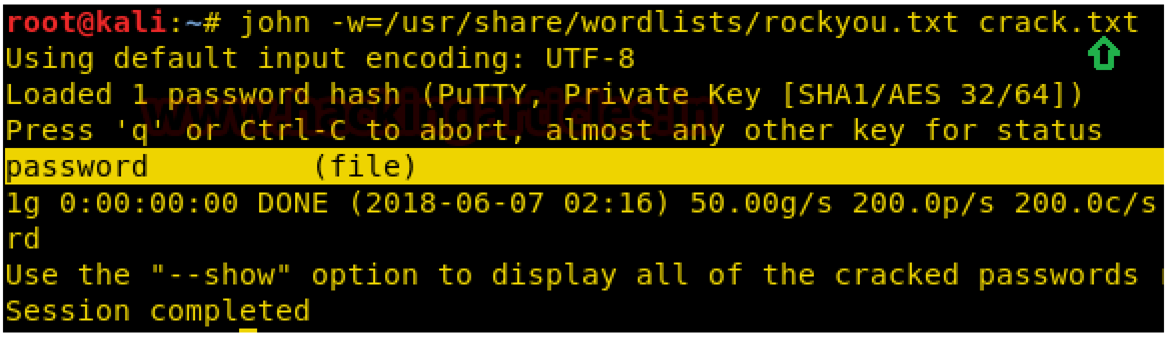
```
file:$putty$1*16*1*0*75d71f6b7d5d7eb5d4ef198e3aad4e6e02259fa5*277*
```

You can see that we converted the key to a crackable hash and then entered it into a text file named crack.txt.

Now let's use John the Ripper to crack this hash.

```
john -w=/usr/share/wordlists/rockyou.txt crack.txt
```

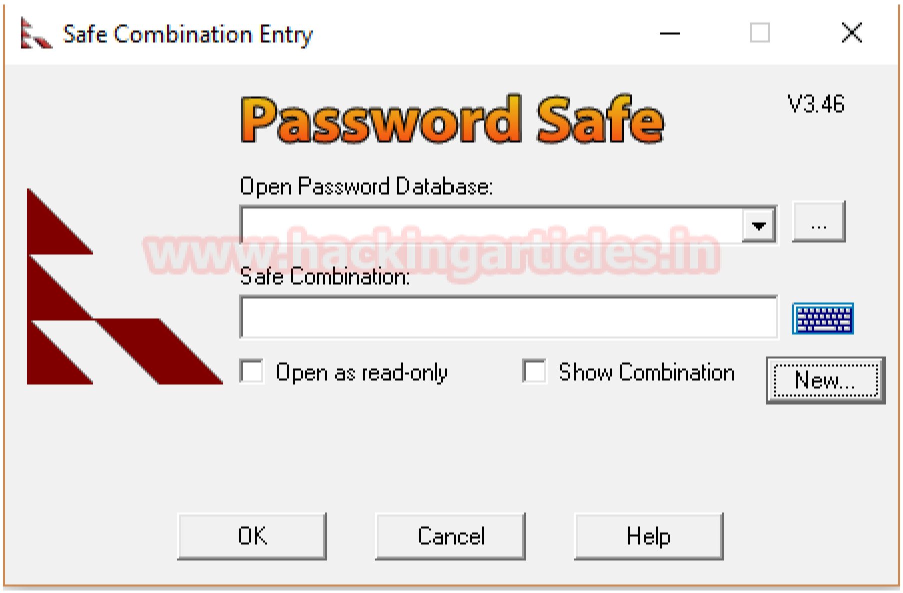
Great! We have successfully cracked the passphrase used to create the private PuTTY key to be "password".



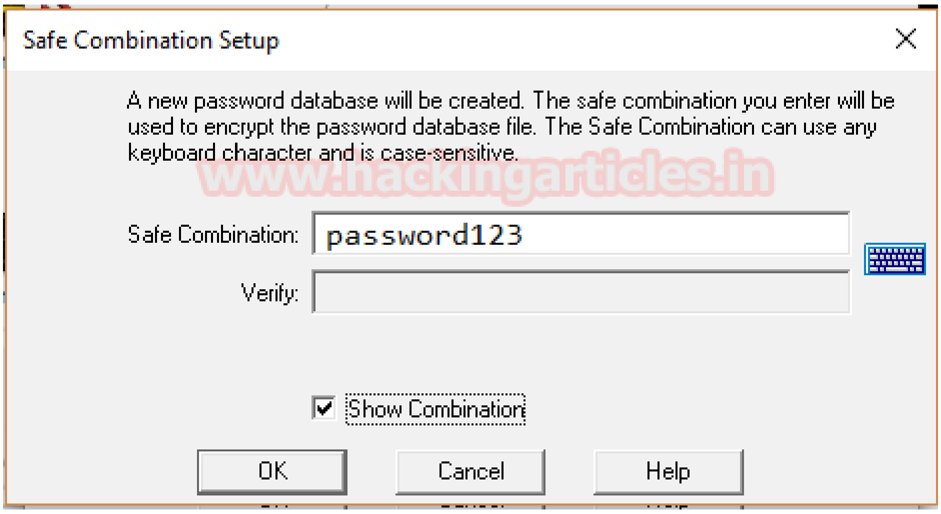
```
root@kali:~# john -w=/usr/share/wordlists/rockyou.txt crack.txt
Using default input encoding: UTF-8
Loaded 1 password hash (PuTTY, Private Key [SHA1/AES 32/64])
Press 'q' or Ctrl-C to abort, almost any other key for status
password         (file)
1g 0:00:00:00 DONE (2018-06-07 02:16) 50.00g/s 200.0p/s 200.0c/s
rd
Use the "--show" option to display all of the cracked passwords
Session completed
```

Cracking the "Password Safe" Password Hash

John the Ripper can crack the Password Safe Software's key. To test the cracking of the key, first, we will have to create a set of new keys. To do this we will install the Password Safe Software on our Windows 10 System.



To get a new key, Click on "New"



Safe Combination Setup

A new password database will be created. The safe combination you enter will be used to encrypt the password database file. The Safe Combination can use any keyboard character and is case-sensitive.

Safe Combination: password123

Verify:

☑ Show Combination

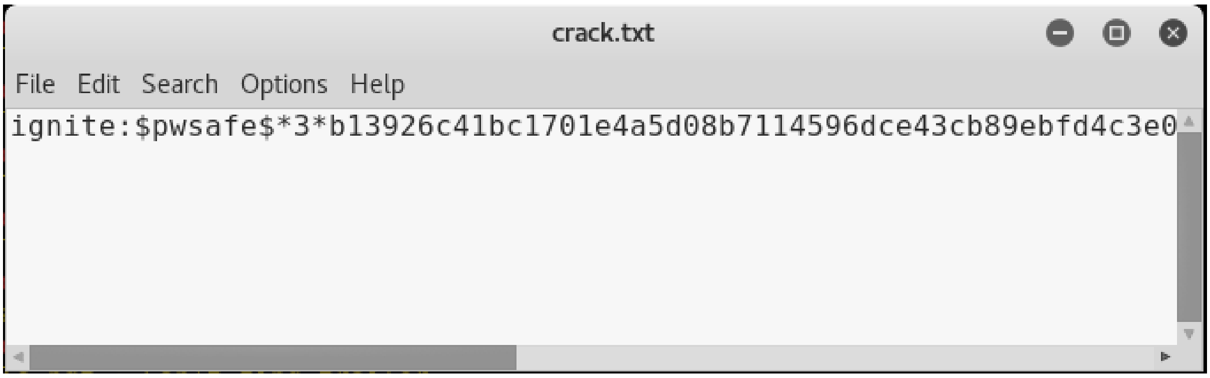
In this prompt, check the Show Combination Box. After that Enter the passphrase you want to use to generate the key. This will generate a .psafe3 file.

After generating transfer this .safe3 file to Kali Linux.

Now John cannot directly crack this key, first, we will have to change its format, which can be done using a john utility called "pwsafe2john".

Syntax: pwsafe2john [location of key]

```
pwsafe2john ignite.psafe3 > crack.txt
```



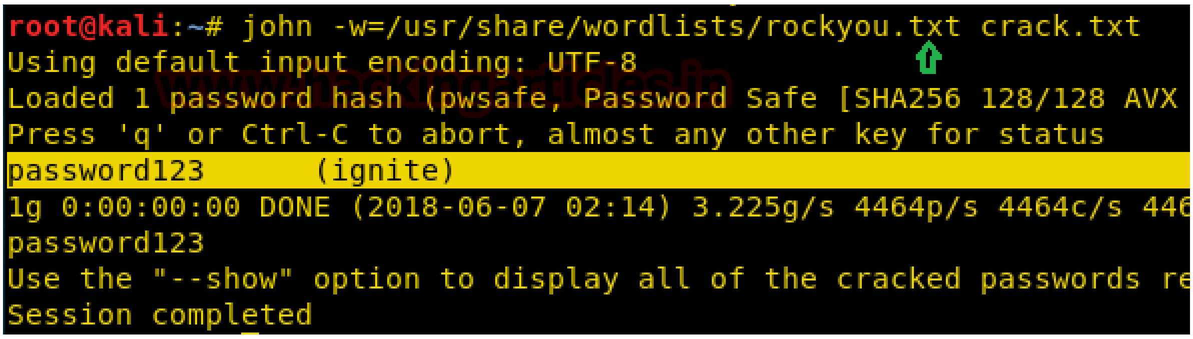
```
ignite:$pwsafe$*3*b13926c41bc1701e4a5d08b7114596dce43cb89ebfd4c3e0
```

You can see that we converted the key to a crackable hash and then entered it into a text file named crack.txt.

Now let’s use John the Ripper to crack this hash.

```
john -w=/usr/share/wordlists/rockyou.txt crack.txt
```

Great! We have successfully cracked the passphrase used to create the private pwsafe key to be “password123”



```
root@kali:~# john -w=/usr/share/wordlists/rockyou.txt crack.txt
Using default input encoding: UTF-8
Loaded 1 password hash (pwsafe, Password Safe [SHA256 128/128 AVX
Press 'q' or Ctrl-C to abort, almost any other key for status
password123      (ignite)
1g 0:00:00:00 DONE (2018-06-07 02:14) 3.225g/s 4464p/s 4464c/s 446
password123
Use the "--show" option to display all of the cracked passwords re
Session completed
```

Conclusion

Hence, one can make use of these commands as a cybersecurity professional to assess vulnerabilities on systems and keep these systems away from threat.

References

- https://www.hackingarticles.in/beginner-guide-john-the-ripper-part-1/
- https://www.hackingarticles.in/beginners-guide-for-john-the-ripper-part-2/
- https://www.openwall.com/john/
- https://github.com/openwall/john
- https://www.sodapdf.com/password-protect-pdf/
- https://github.com/truongkma/ctf-tools/blob/master/John/run/7z2john.py