LAB MANUAL
ON
PENETRATION TESTING OF PASSWORD PROTECTED DOCUMENTS

ESTABLISHMENT OF ADVANCED LABORATORY FOR CYBER SECURITY TRAINING TO TECHNICAL TEACHERS

DEPARTMENT OF INFORMATION MANAGEMENT AND EMERGING ENGINEERING

MINISTRY OF ELECTRONICS AND INFORMATION TECHNOLOGY

GOVERNMENT OF INDIA

Principal Investigator: Prof. Maitreyee Dutta

Co Investigator: Prof. Shyam Sundar Pattnaik

PREPARED BY:

Prof. Maitreyee Dutta and Ms. Shweta Sharma (Technical Assistant)

Table of Contents

MANUAL-6: PENETRATION TESTING OF PASSWORD PROTECTED DOCUMENTS

INTRODUCTION TO PENETRATION TESTING

- Penetration testing or PenTest is a practice of testing a computer system, network or web application.
- It finds security vulnerabilities that an attacker could exploit.
- Penetration testing should not be confused with vulnerability testing. The intention of Vulnerability Testing is just to identify the potential problems, whereas Pen-Testing is to attack those problems.

PASSWORDS IN PROTECTED DOCUMENTS

- Passwords are used to protect the documents from an unauthorized access.
- Documents such as PDF, MS Word, MS excel, Zip, and RAR files can be protected by password to secure the files.
- Password cracking is a process to recover passwords of these protected documents.

- The purpose of password cracking is to recover forgotten password. The forensic team can perform password cracking on these protected documents to recover the data after getting the password.
- Penetration testing can be performed with John-the-Ripper tool to access password protected documents.

PASSWORD CRACKING TECHNIQUES

The password cracking techniques are discussed as follows:

- **BRUTE FORCE**: A brute force technique is an attempt to crack passwords using permutation and combination approach. This method takes a lot of time and memory consumption depending on the length and complexity of password.

- **DICTIONARY**: A dictionary technique is an attempt to store in-build passwords in a file known as dictionary. Instead of trying all combination of passwords, it creates a word-list of most common passwords and calculates the hash values while cracking the passwords. It will only able to crack the password if it is stored in dictionary file. This technique takes less time as compared to brute-force technique to crack the password.

- **RAINBOW TABLES**: This technique is same as dictionary, but instead of calculating hash vales during password cracking; it stores the in-built hash values of password in the tables. Thus, this technique takes less time as compared to brute-force and dictionary technique to crack the password.

JOHN-THE-RIPPER TOOL

- John-the-ripper tool [1] is an open-source application and penetration testing tool that allows users to view authentication credentials of password protected documents.

- This tool provides hashes of password protected documents.

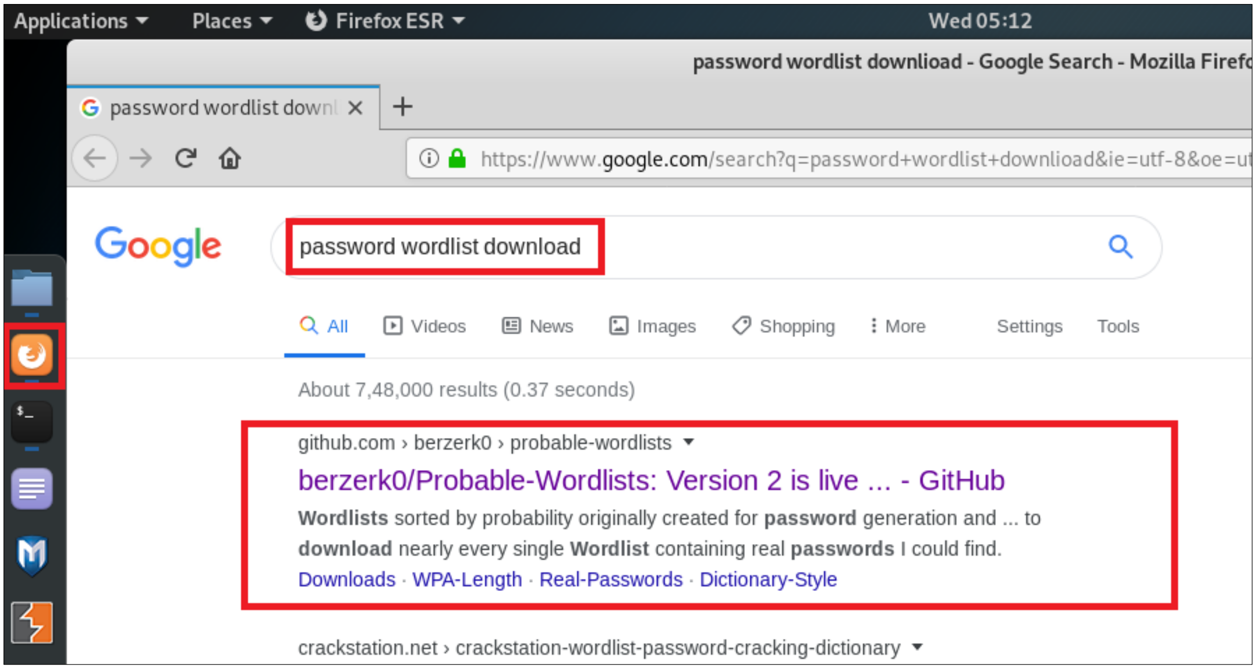
- The forensics team can use John-the-ripper tool to get the password in plain text and pass it to the password protected documents to open it.

- This tool can be used in both Kali Linux operating system and Windows operating system.

DOWNLOAD DICTIONARY

The following steps are followed to download dictinary:

Step 1: Search the password wordlist by browsing Google search engine as shown in Figure 1.



password wordlist download - Google Search - Mozilla Firefo

https://www.google.com/search?q=password+wordlist+downlioad&ie=utf-8&oe=ut

password wordlist download

About 7,48,000 results (0.37 seconds)

github.com › berzerk0 › probable-wordlists

berzerk0/Probable-Wordlists: Version 2 is live ... - GitHub

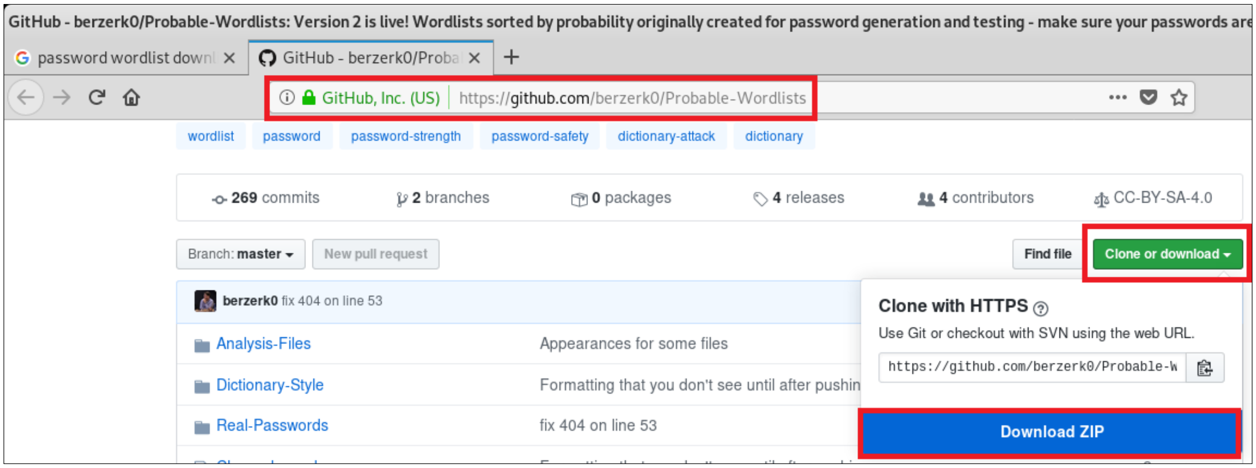
Wordlists sorted by probability originally created for **password** generation and ... to **download** nearly every single **Wordlist** containing real **passwords** I could find.

Downloads · WPA-Length · Real-Passwords · Dictionary-Style

crackstation.net › crackstation-wordlist-password-cracking-dictionary

Figure 1: Search password wordlist

Step 2: Open the GitHub website and download the ZIP file as shown in Figure 2.



GitHub - berzerk0/Probable-Wordlists: Version 2 is live! Wordlists sorted by probability originally created for password generation and testing - make sure your passwords are

Clone with HTTPS

Use Git or checkout with SVN using the web URL.

```
https://github.com/berzerk0/Probable-W
```

Figure 2: Download password wordlist

Step 3: Save and open the downloaded file as shown in Figure 3.

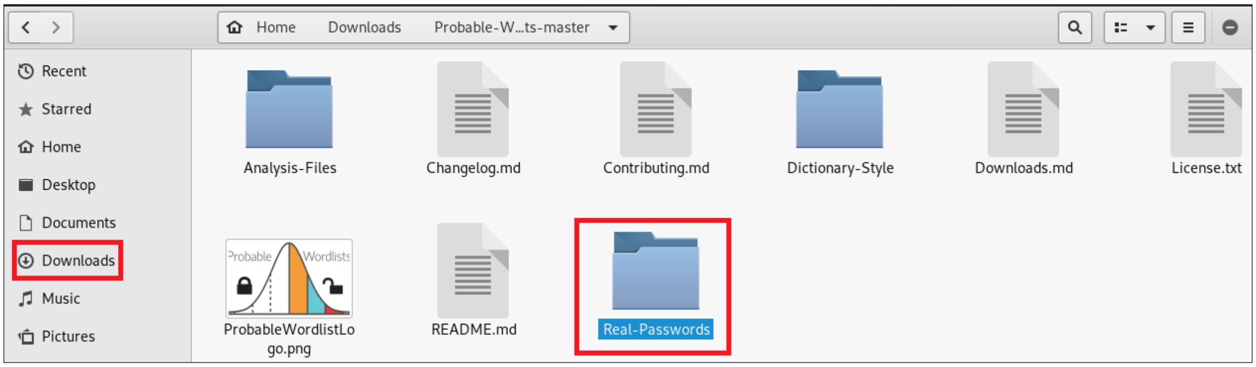


Figure 3: Password folder in downloaded file

Step 4: Open the "Real-Passwords" folder to see the passwords wordlist as shown in Figure 4.

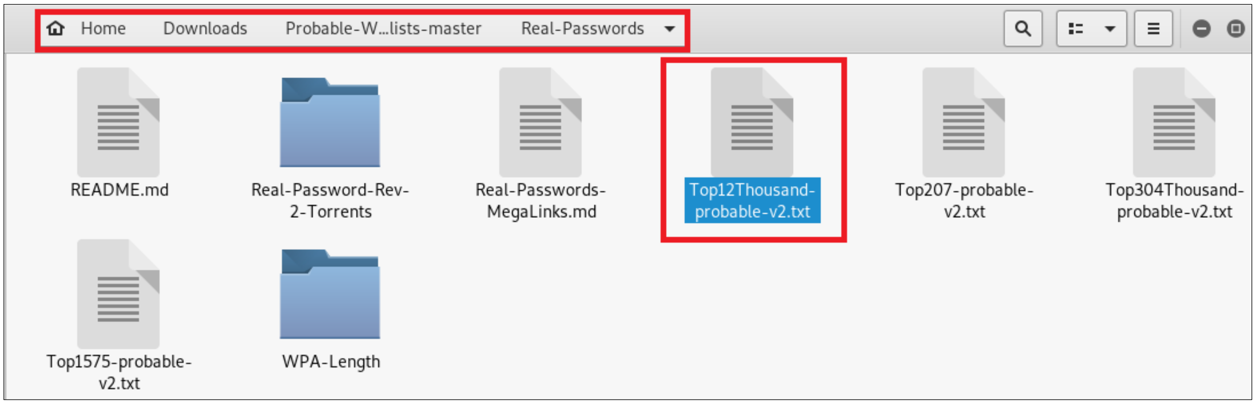
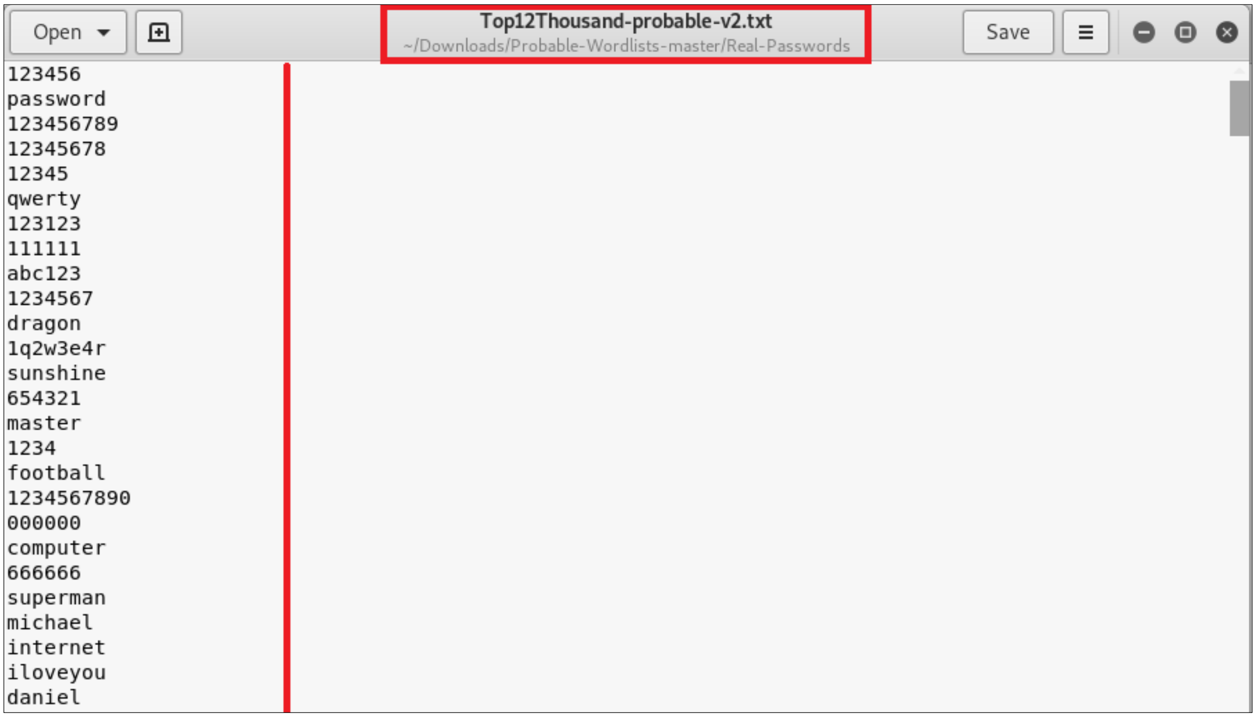

Home
Downloads
Probable-W...lists-master
Real-Passwords
README.md
Real-Password-Rev-2-Torrents
Real-Passwords-MegaLinks.md
Top12Thousand-probable-v2.txt
Top207-probable-v2.txt
Top304Thousand-probable-v2.txt
Top1575-probable-v2.txt
WPA-Length

Figure 4: Password wordlist

<u>Step 5:</u> Open any password wordlist (e.g., Top12Thousand-probable-v2.txt file) as shown in Figure 5.


Open
Top12Thousand-probable-v2.txt
~/Downloads/Probable-Wordlists-master/Real-Passwords
Save

```
123456
password
123456789
12345678
12345
qwerty
123123
111111
abc123
1234567
dragon
1q2w3e4r
sunshine
654321
master
1234
football
1234567890
000000
computer
666666
superman
michael
internet
iloveyou
daniel
```

Figure 5: Top 12 thousand most frequently used passwords

<u>Step 6:</u> Copy this file in Home directory and rename as “wordlist.txt” as shown in Figure 6.

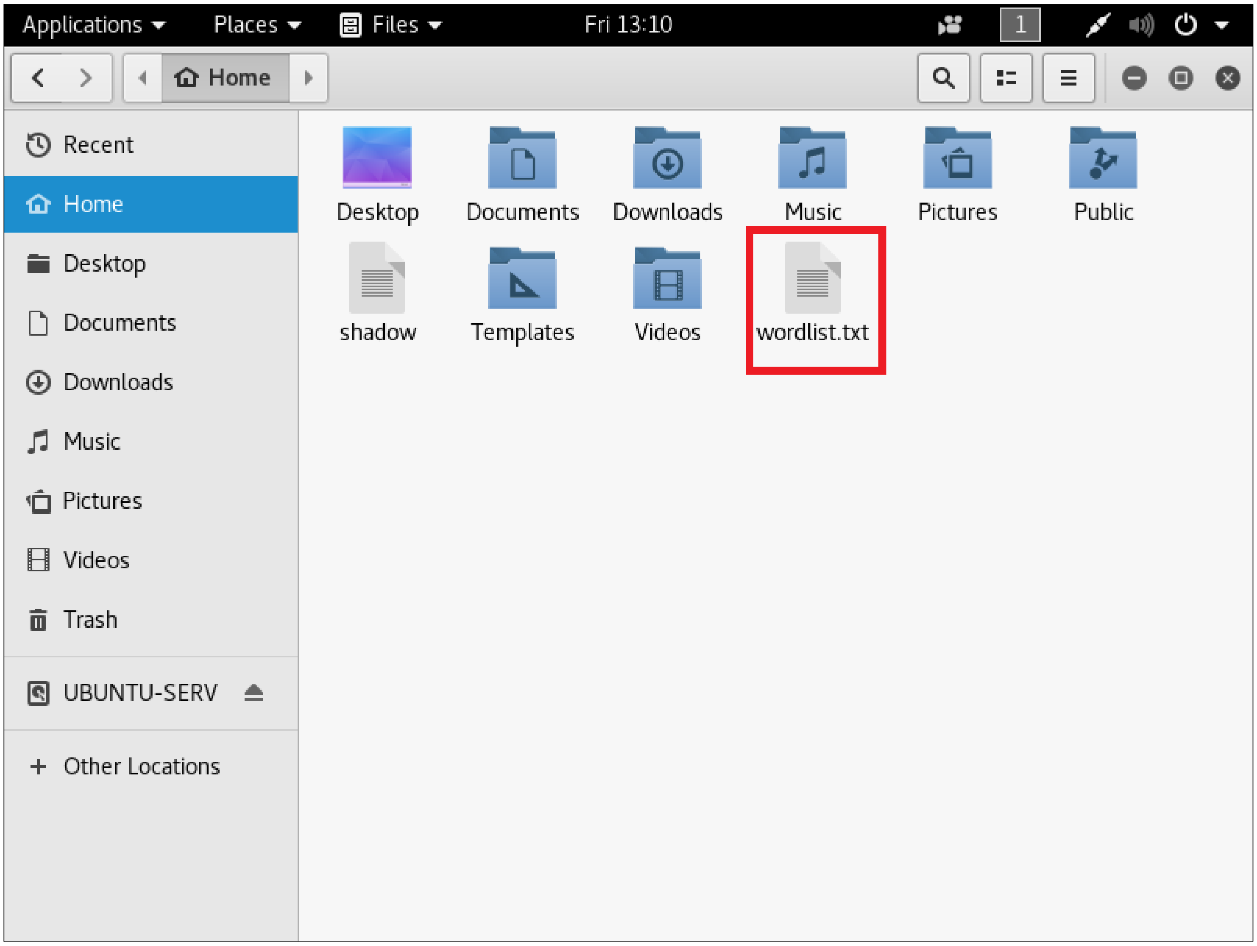


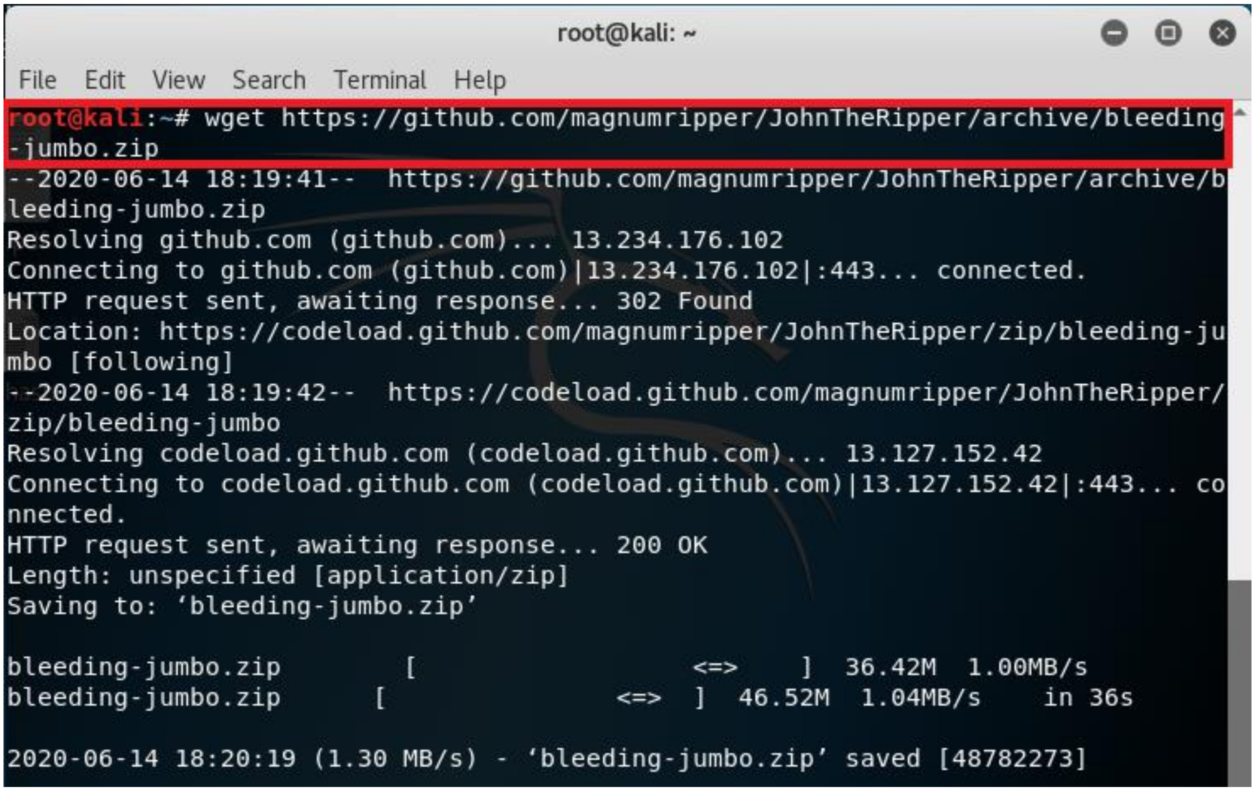
Figure 6: Wordlist file in Home directory

TASK-I: PENETRATION TESTING OF PROTECTED PDF DOCUMENTS IN KALI LINUX OPERATING SYSTEM

John-the-Ripper tool is available in Kali Linux operating system and Windows operating system. This manual shows practical of penetration testing of password protected documents in Kali Linux and Windows operating system.

The penetration testing of protected PDF documents with John-the-ripper tool can be done with the following steps:

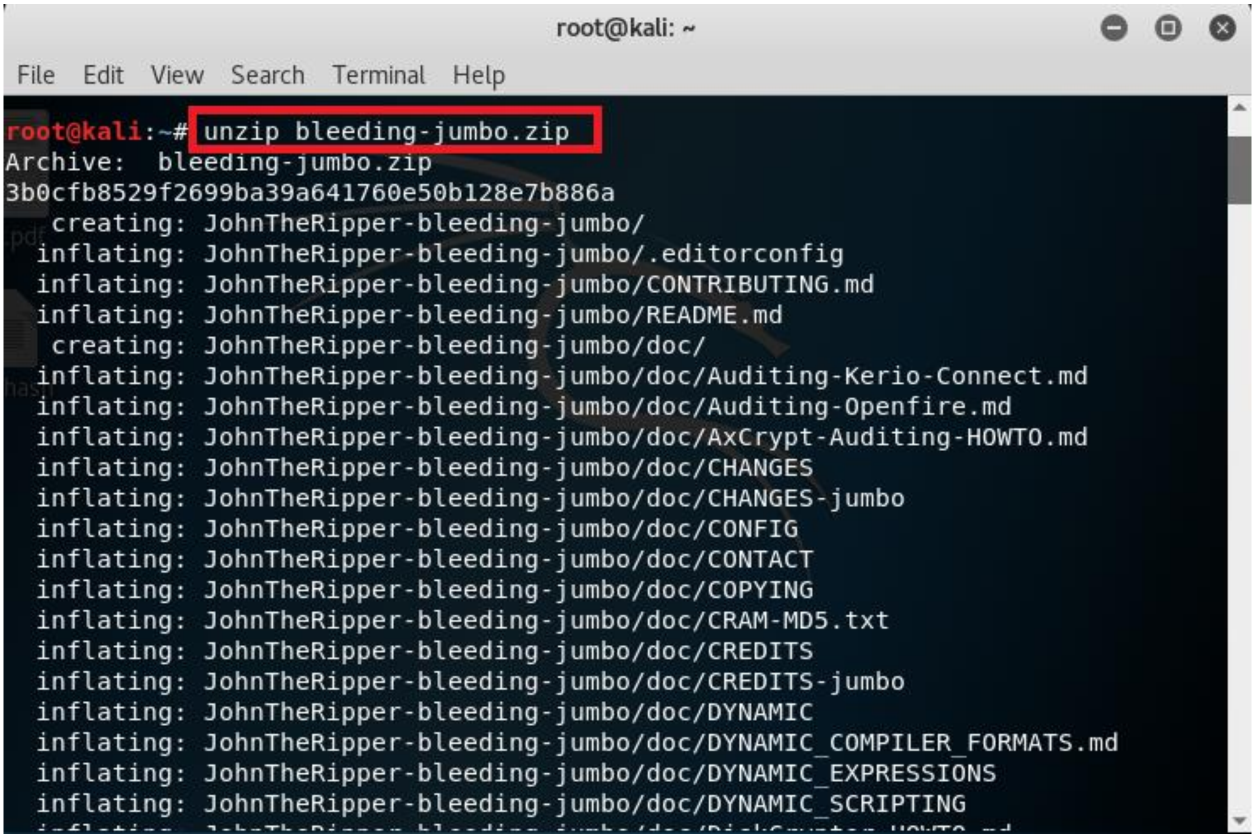
Step 7: Download bleeding-jumbo.zip from github in Kali Linux operating system with command "wget https://github/magnumripper/JohnTheRipper/archive/bleeding-jumbo.zip" as shown in Figure 7.



```
root@kali:~# wget https://github.com/magnumripper/JohnTheRipper/archive/bleeding
-jumbo.zip
--2020-06-14 18:19:41--  https://github.com/magnumripper/JohnTheRipper/archive/b
leeding-jumbo.zip
Resolving github.com (github.com)... 13.234.176.102
Connecting to github.com (github.com)|13.234.176.102|:443... connected.
HTTP request sent, awaiting response... 302 Found
Location: https://codeload.github.com/magnumripper/JohnTheRipper/zip/bleeding-ju
mbo [following]
--2020-06-14 18:19:42--  https://codeload.github.com/magnumripper/JohnTheRipper/
zip/bleeding-jumbo
Resolving codeload.github.com (codeload.github.com)... 13.127.152.42
Connecting to codeload.github.com (codeload.github.com)|13.127.152.42|:443... co
nnected.
HTTP request sent, awaiting response... 200 OK
Length: unspecified [application/zip]
Saving to: 'bleeding-jumbo.zip'

bleeding-jumbo.zip       [                  <=>    ]  36.42M  1.00MB/s
bleeding-jumbo.zip      [              <=>  ]  46.52M  1.04MB/s    in 36s

2020-06-14 18:20:19 (1.30 MB/s) - 'bleeding-jumbo.zip' saved [48782273]
```

Figure 7: Download bleeding-jumbo.zip from github

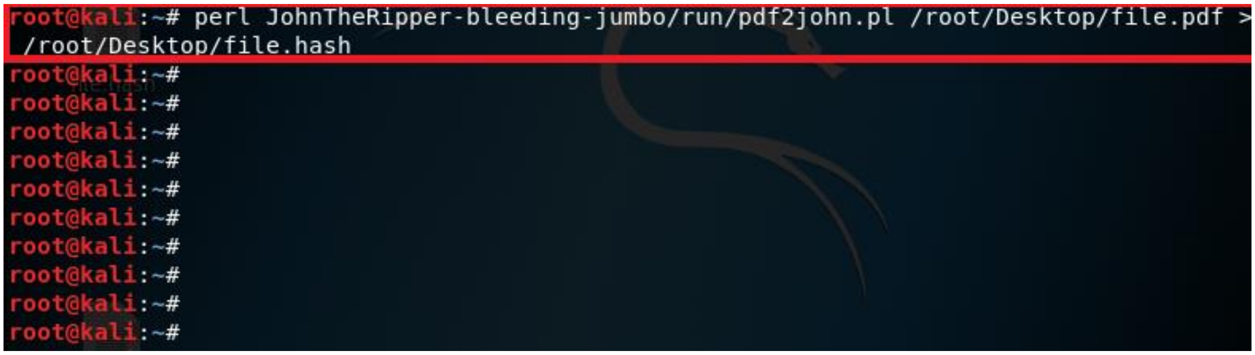
Step 8: Unzip bleeding-jumbo.zip file with command "unzip bleeding-jumbo.zip" as shown in Figure 8.



```
root@kali:~# unzip bleeding-jumbo.zip
Archive:  bleeding-jumbo.zip
3b0cfb8529f2699ba39a641760e50b128e7b886a
   creating: JohnTheRipper-bleeding-jumbo/
  inflating: JohnTheRipper-bleeding-jumbo/.editorconfig
  inflating: JohnTheRipper-bleeding-jumbo/CONTRIBUTING.md
  inflating: JohnTheRipper-bleeding-jumbo/README.md
   creating: JohnTheRipper-bleeding-jumbo/doc/
  inflating: JohnTheRipper-bleeding-jumbo/doc/Auditing-Kerio-Connect.md
  inflating: JohnTheRipper-bleeding-jumbo/doc/Auditing-Openfire.md
  inflating: JohnTheRipper-bleeding-jumbo/doc/AxCrypt-Auditing-HOWTO.md
  inflating: JohnTheRipper-bleeding-jumbo/doc/CHANGES
  inflating: JohnTheRipper-bleeding-jumbo/doc/CHANGES-jumbo
  inflating: JohnTheRipper-bleeding-jumbo/doc/CONFIG
  inflating: JohnTheRipper-bleeding-jumbo/doc/CONTACT
  inflating: JohnTheRipper-bleeding-jumbo/doc/COPYING
  inflating: JohnTheRipper-bleeding-jumbo/doc/CRAM-MD5.txt
  inflating: JohnTheRipper-bleeding-jumbo/doc/CREDITS
  inflating: JohnTheRipper-bleeding-jumbo/doc/CREDITS-jumbo
  inflating: JohnTheRipper-bleeding-jumbo/doc/DYNAMIC
  inflating: JohnTheRipper-bleeding-jumbo/doc/DYNAMIC_COMPILER_FORMATS.md
  inflating: JohnTheRipper-bleeding-jumbo/doc/DYNAMIC_EXPRESSIONS
  inflating: JohnTheRipper-bleeding-jumbo/doc/DYNAMIC_SCRIPTING
```

Figure 8: Unzip bleeding-jumbo.zip

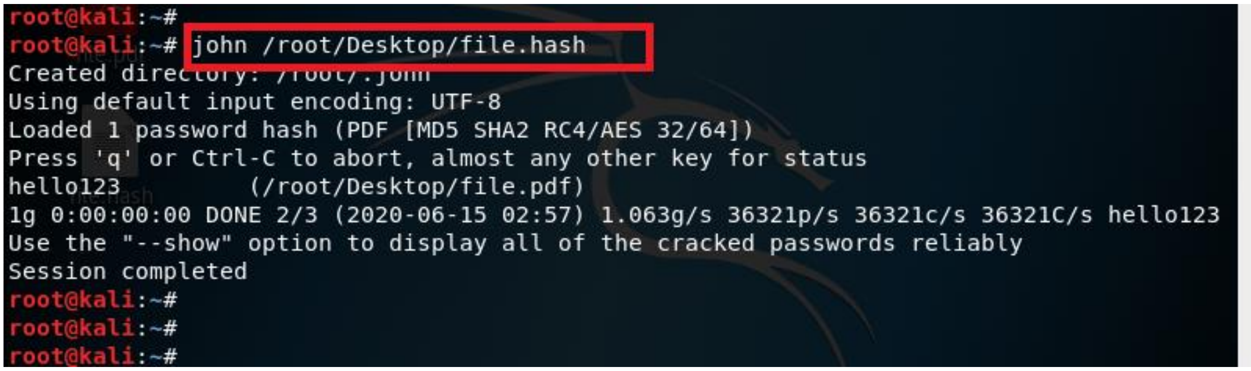
__Step 9:__ Create hash of PDF document with the command "perl JohnTheRipper-bleeding-jumbo/run/pdf2john.pl /root/Desktop/file.pdf > /root/Desktop/file.hash" as shown in Figure 9.



```
root@kali:~# perl JohnTheRipper-bleeding-jumbo/run/pdf2john.pl /root/Desktop/file.pdf >
 /root/Desktop/file.hash
root@kali:~#
root@kali:~#
root@kali:~#
root@kali:~#
root@kali:~#
root@kali:~#
root@kali:~#
root@kali:~#
root@kali:~#
root@kali:~#
```

Figure 9: Create hash of PDF document

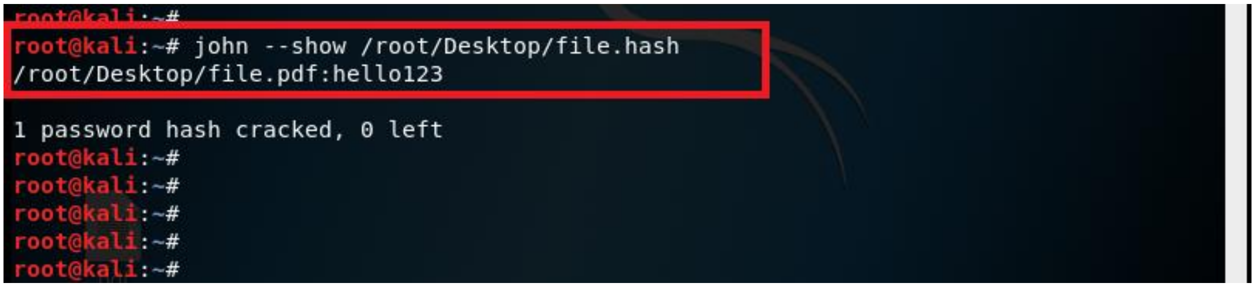
<u>Step 10:</u> Use John-the-ripper tool to get password from hash file via brute-force password cracking technique with command “john /root/Desktop/file.hash” as shown in Figure 10.



```
root@kali:~#
root@kali:~# john /root/Desktop/file.hash
Created directory: /root/.john
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
hello123         (/root/Desktop/file.pdf)
1g 0:00:00:00 DONE 2/3 (2020-06-15 02:57) 1.063g/s 36321p/s 36321c/s 36321C/s hello123
Use the "--show" option to display all of the cracked passwords reliably
Session completed
root@kali:~#
root@kali:~#
root@kali:~#
```

Figure 10: Use John-the-ripper to get password of hash file

<u>Step 11:</u> To show the cracked password of protected PDF document, type command “john --show /root/Desktop/file.hash”. The cracked password of protected PDF file is “hello123” as shown in Figure 11.

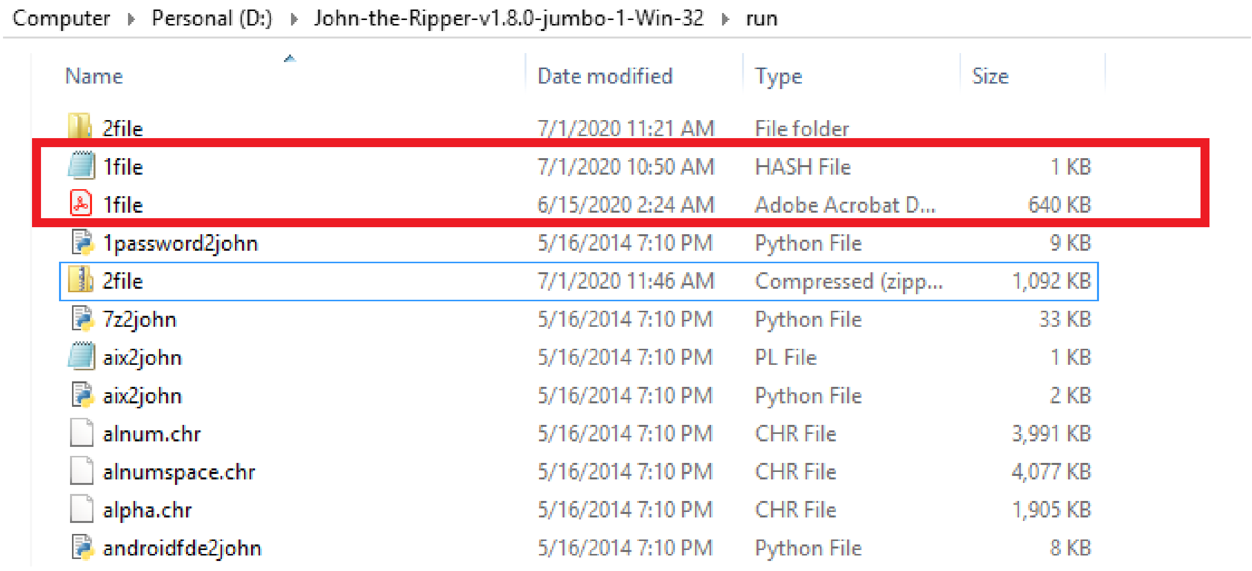


```
root@kali:~#
root@kali:~# john --show /root/Desktop/file.hash
/root/Desktop/file.pdf:hello123

1 password hash cracked, 0 left
root@kali:~#
root@kali:~#
root@kali:~#
root@kali:~#
root@kali:~#
```

Figure 11: Cracked password

TASK-II: PENETRATION TESTING OF PROTECTED PDF DOCUMENTS IN WINDOWS OPERATING SYSTEM

Step 12: Download John-the-Ripper-v1.8.0-jumbo-1-Win-32 or 64 bit from website (https://www.openwall.com/john/). Place the password protected PDF document to crack in the run folder of downloaded John-the-Ripper tool as shown in Figure 12.



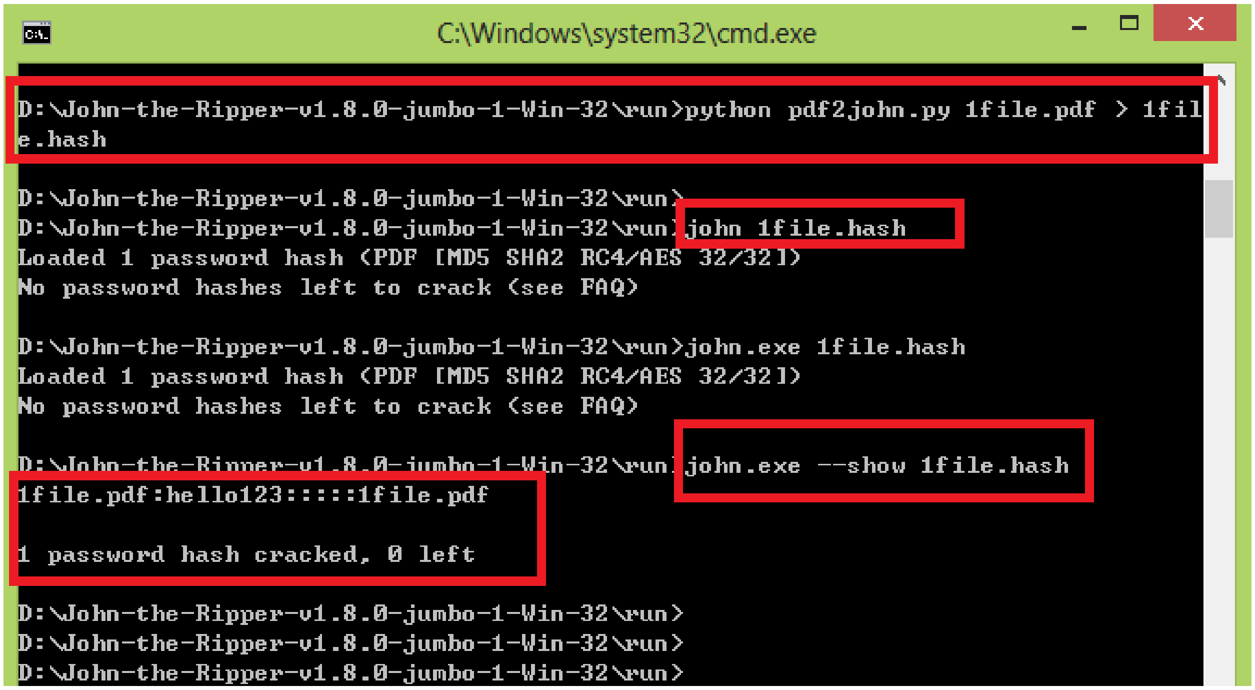
Computer ▸ Personal (D:) ▸ John-the-Ripper-v1.8.0-jumbo-1-Win-32 ▸ run

Name	Date modified	Type	Size
2file	7/1/2020 11:21 AM	File folder	
1file	7/1/2020 10:50 AM	HASH File	1 KB
1file	6/15/2020 2:24 AM	Adobe Acrobat D...	640 KB
1password2john	5/16/2014 7:10 PM	Python File	9 KB
2file	7/1/2020 11:46 AM	Compressed (zipp...	1,092 KB
7z2john	5/16/2014 7:10 PM	Python File	33 KB
aix2john	5/16/2014 7:10 PM	PL File	1 KB
aix2john	5/16/2014 7:10 PM	Python File	2 KB
alnum.chr	5/16/2014 7:10 PM	CHR File	3,991 KB
alnumspace.chr	5/16/2014 7:10 PM	CHR File	4,077 KB
alpha.chr	5/16/2014 7:10 PM	CHR File	1,905 KB
androidfde2john	5/16/2014 7:10 PM	Python File	8 KB

Figure 12: Password protected PDF document

Step 13: Type the command to create hash file of password protected PDF file with command “python pdf2john.py

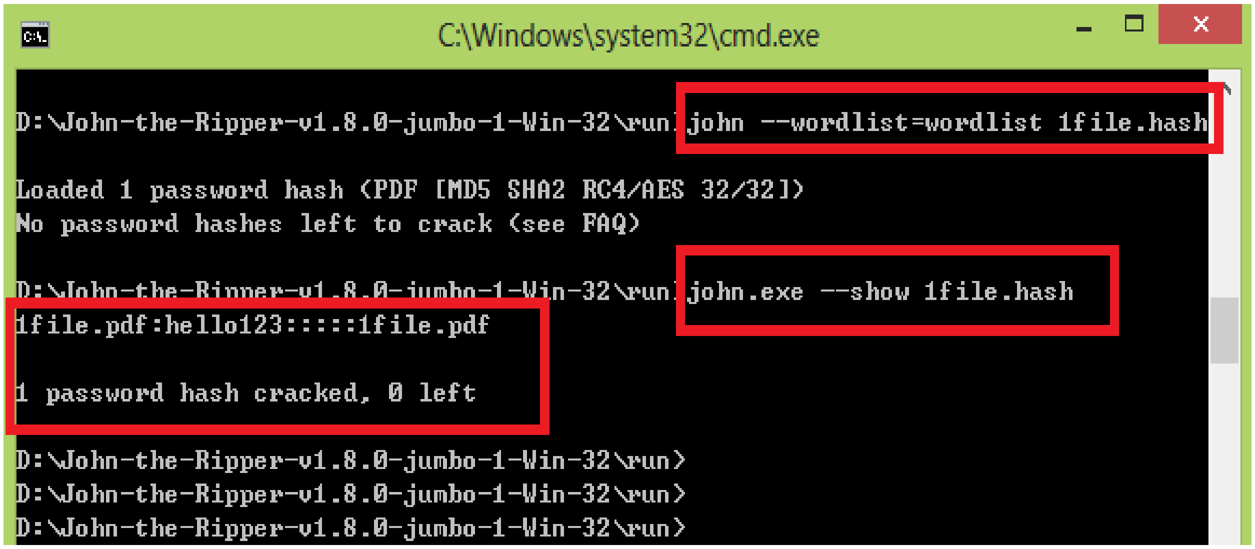
1file.pdf > 1file.hash” as shown in Figure 13 where 1file is the name of the PDF file. Type “john 1file.hash” to get the password in plain text via brute force password cracking technique. Type the command “john.exe --show 1file.hash” to show the password. The password of the protected file is “hello123” as shown in Figure 13.



```
D:\John-the-Ripper-v1.8.0-jumbo-1-Win-32\run>python pdf2john.py 1file.pdf > 1fil
e.hash

D:\John-the-Ripper-v1.8.0-jumbo-1-Win-32\run>
D:\John-the-Ripper-v1.8.0-jumbo-1-Win-32\run>john 1file.hash
Loaded 1 password hash (PDF [MD5 SHA2 RC4/AES 32/32])
No password hashes left to crack (see FAQ)

D:\John-the-Ripper-v1.8.0-jumbo-1-Win-32\run>john.exe 1file.hash
Loaded 1 password hash (PDF [MD5 SHA2 RC4/AES 32/32])
No password hashes left to crack (see FAQ)

D:\John-the-Ripper-v1.8.0-jumbo-1-Win-32\run>john.exe --show 1file.hash
1file.pdf:hello123:::::1file.pdf

1 password hash cracked, 0 left

D:\John-the-Ripper-v1.8.0-jumbo-1-Win-32\run>
D:\John-the-Ripper-v1.8.0-jumbo-1-Win-32\run>
D:\John-the-Ripper-v1.8.0-jumbo-1-Win-32\run>
```

Figure 13: Cracking password in Windows operating system (via brute-force)

<u>Step 14:</u> The other way of cracking the password is by using dictionary mode. Write the command “john --wordlist= wordlist 1file.hash” to compare the hash of PDF file with dictionary. Write the command “*john.exe --show 1file.hash*” to display the passwords in plaintext of protected PDF document as shown in Figure 14. The passwords in plaintext are displayed in the Figure 14 and highlighted in red rectangular box.



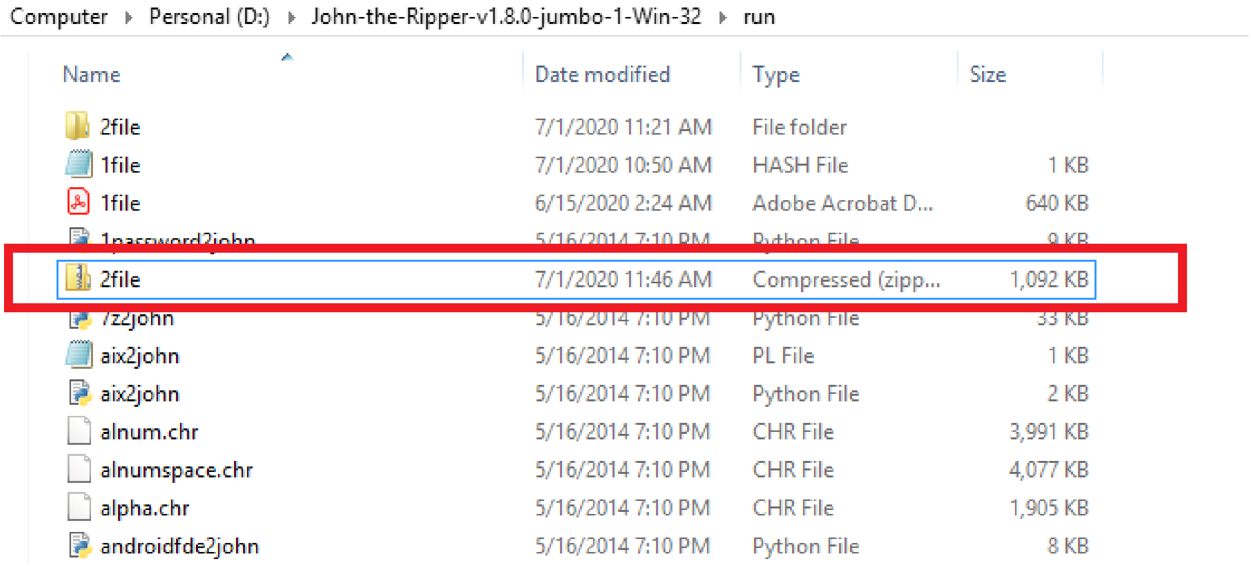
```
D:\John-the-Ripper-v1.8.0-jumbo-1-Win-32\run>john --wordlist=wordlist 1file.hash

Loaded 1 password hash (PDF [MD5 SHA2 RC4/AES 32/32])
No password hashes left to crack (see FAQ)

D:\John-the-Ripper-v1.8.0-jumbo-1-Win-32\run>john.exe --show 1file.hash
1file.pdf:hello123:::::1file.pdf

1 password hash cracked, 0 left

D:\John-the-Ripper-v1.8.0-jumbo-1-Win-32\run>
D:\John-the-Ripper-v1.8.0-jumbo-1-Win-32\run>
D:\John-the-Ripper-v1.8.0-jumbo-1-Win-32\run>
```

Figure 14: Cracking password of protected PDF document in Windows operating system (via dictionary)

TASK-III: PENETRATION TESTING OF PROTECTED ZIP DOCUMENTS IN WINDOWS OPERATING SYSTEM

The penetration testing of protected ZIP documents with John-the-ripper tool can be done with the following steps:

Step 15: Place the password protected ZIP document to crack in the run folder of downloaded John-the-Ripper tool as shown in Figure 15.

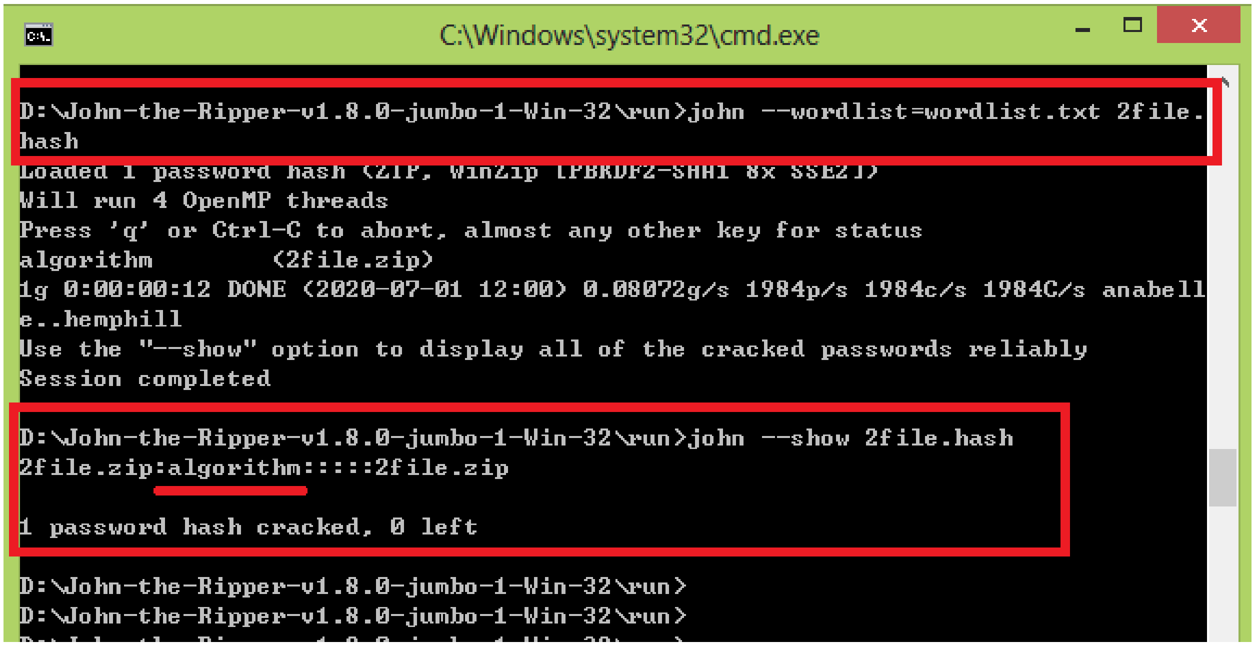


Computer ▸ Personal (D:) ▸ John-the-Ripper-v1.8.0-jumbo-1-Win-32 ▸ run

Name	Date modified	Type	Size
2file	7/1/2020 11:21 AM	File folder	
1file	7/1/2020 10:50 AM	HASH File	1 KB
1file	6/15/2020 2:24 AM	Adobe Acrobat D...	640 KB
1password2john	5/16/2014 7:10 PM	Python File	9 KB
2file	7/1/2020 11:46 AM	Compressed (zipp...	1,092 KB
7z2john	5/16/2014 7:10 PM	Python File	33 KB
aix2john	5/16/2014 7:10 PM	PL File	1 KB
aix2john	5/16/2014 7:10 PM	Python File	2 KB
alnum.chr	5/16/2014 7:10 PM	CHR File	3,991 KB
alnumspace.chr	5/16/2014 7:10 PM	CHR File	4,077 KB
alpha.chr	5/16/2014 7:10 PM	CHR File	1,905 KB
androidfde2john	5/16/2014 7:10 PM	Python File	8 KB

Figure 15: Password protected ZIP document

<u>Step 16:</u> Write the command "john --wordlist= wordlist.txt 2file.hash" to compare the hash of PDF file with dictionary. Write the command "*john.exe --show 2file.hash*" to display the passwords in plaintext of protected ZIP document as shown in Figure 16. The passwords in plaintext are displayed in the Figure 16 and highlighted in red rectangular box.



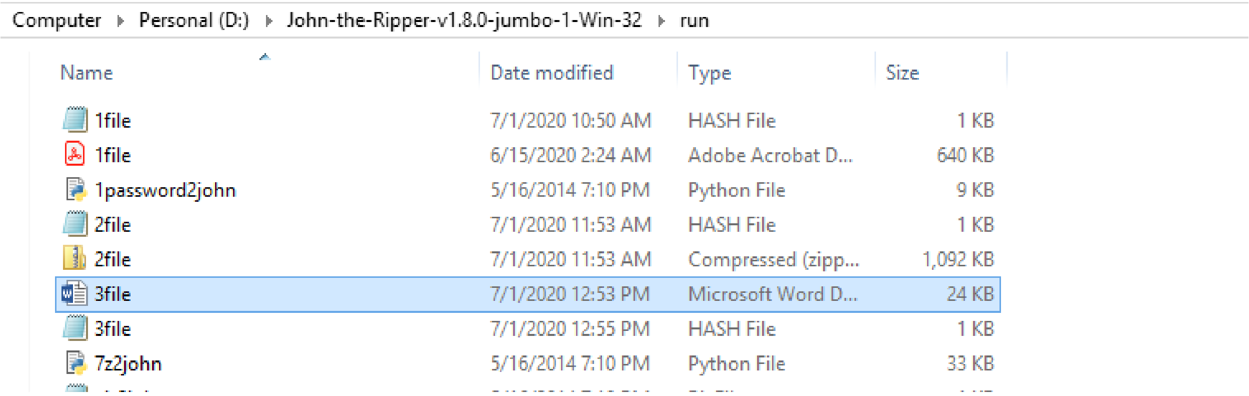
```
D:\John-the-Ripper-v1.8.0-jumbo-1-Win-32\run>john --wordlist=wordlist.txt 2file.
hash
Loaded 1 password hash (ZIP, WinZip [PBKDF2-SHA1 8x SSE2])
Will run 4 OpenMP threads
Press 'q' or Ctrl-C to abort, almost any other key for status
algorithm        (2file.zip)
1g 0:00:00:12 DONE (2020-07-01 12:00) 0.08072g/s 1984p/s 1984c/s 1984C/s anabell
e..hemphill
Use the "--show" option to display all of the cracked passwords reliably
Session completed

D:\John-the-Ripper-v1.8.0-jumbo-1-Win-32\run>john --show 2file.hash
2file.zip:algorithm:::::2file.zip

1 password hash cracked, 0 left

D:\John-the-Ripper-v1.8.0-jumbo-1-Win-32\run>
D:\John-the-Ripper-v1.8.0-jumbo-1-Win-32\run>
```

Figure 16: Cracking password of protected ZIP document in Windows operating system (via dictionary)

TASK-IV: PENETRATION TESTING OF PROTECTED DOCX DOCUMENTS IN WINDOWS OPERATING SYSTEM

The penetration testing of protected DOCX documents with John-the-ripper tool can be done with the following steps:

<u>Step 17:</u> Place the password protected DOCX document to crack in the run folder of downloaded John-the-Ripper tool as shown in Figure 17.

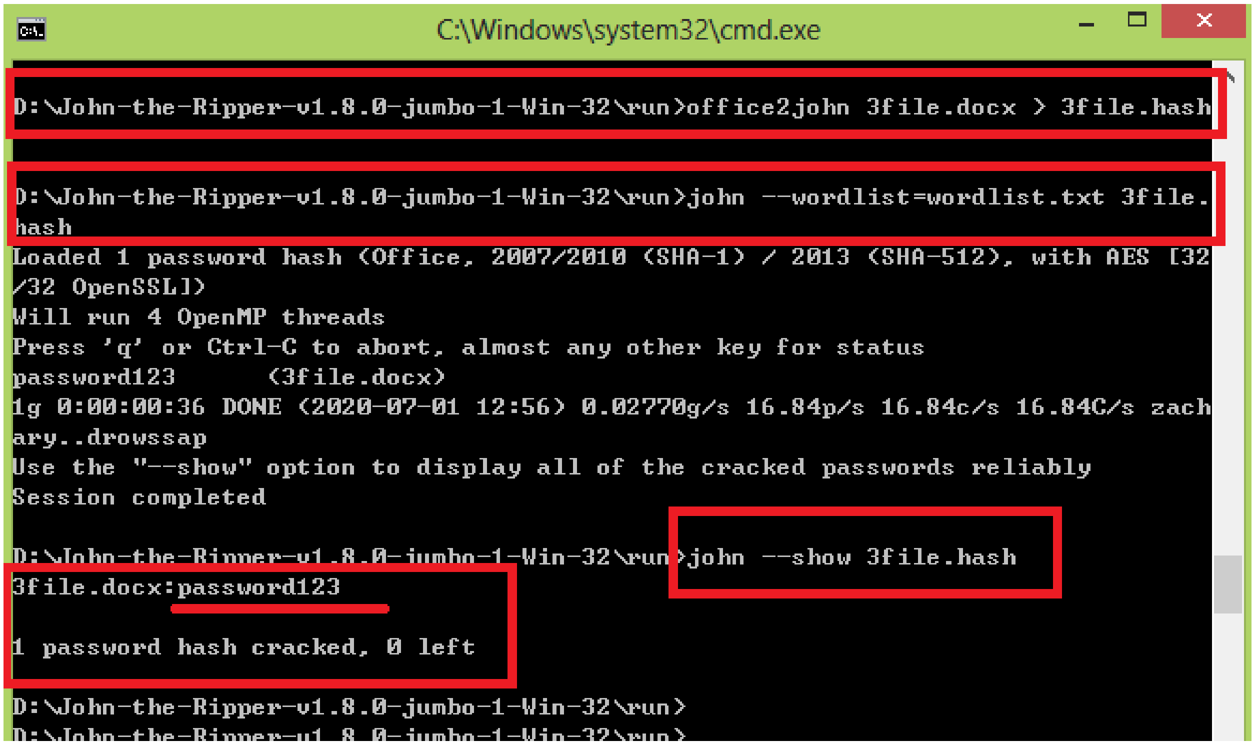


Computer ▸ Personal (D:) ▸ John-the-Ripper-v1.8.0-jumbo-1-Win-32 ▸ run

Name	Date modified	Type	Size
1file	7/1/2020 10:50 AM	HASH File	1 KB
1file	6/15/2020 2:24 AM	Adobe Acrobat D...	640 KB
1password2john	5/16/2014 7:10 PM	Python File	9 KB
2file	7/1/2020 11:53 AM	HASH File	1 KB
2file	7/1/2020 11:53 AM	Compressed (zipp...	1,092 KB
3file	7/1/2020 12:53 PM	Microsoft Word D...	24 KB
3file	7/1/2020 12:55 PM	HASH File	1 KB
7z2john	5/16/2014 7:10 PM	Python File	33 KB

Figure 17: Password protected DOCX document

<u>Step 18:</u> Write the command *"john --wordlist= wordlist.txt 3file.hash"* to compare the hash of PDF file with dictionary. Write the command "*john.exe --show 3file.hash*" to display the passwords in plaintext of protected DOCX document as shown in Figure 18. The passwords in plaintext are displayed in the Figure 18 and highlighted in red rectangular box.



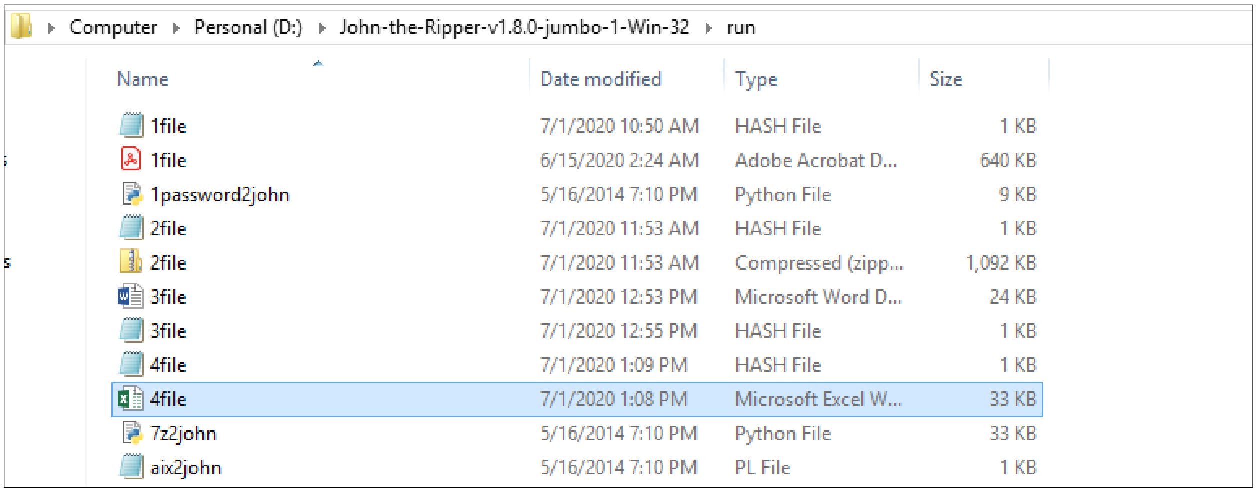
```
D:\John-the-Ripper-v1.8.0-jumbo-1-Win-32\run>office2john 3file.docx > 3file.hash

D:\John-the-Ripper-v1.8.0-jumbo-1-Win-32\run>john --wordlist=wordlist.txt 3file.
hash
Loaded 1 password hash (Office, 2007/2010 (SHA-1) / 2013 (SHA-512), with AES [32
/32 OpenSSL])
Will run 4 OpenMP threads
Press 'q' or Ctrl-C to abort, almost any other key for status
password123      (3file.docx)
1g 0:00:00:36 DONE (2020-07-01 12:56) 0.02770g/s 16.84p/s 16.84c/s 16.84C/s zach
ary..drowssap
Use the "--show" option to display all of the cracked passwords reliably
Session completed

D:\John-the-Ripper-v1.8.0-jumbo-1-Win-32\run>john --show 3file.hash
3file.docx:password123

1 password hash cracked, 0 left

D:\John-the-Ripper-v1.8.0-jumbo-1-Win-32\run>
D:\John-the-Ripper-v1.8.0-jumbo-1-Win-32\run>
```

Figure 22: Cracking password of protected DOCX document in Windows operating system (via dictionary)

TASK-V: PENETRATION TESTING OF PROTECTED EXCEL DOCUMENTS IN WINDOWS OPERATING SYSTEM

The penetration testing of protected EXCEL documents with John-the-ripper tool can be done with the following steps:

Step 19: Place the password protected EXCEL document to crack in the run folder of downloaded John-the-Ripper tool as shown in Figure 19.

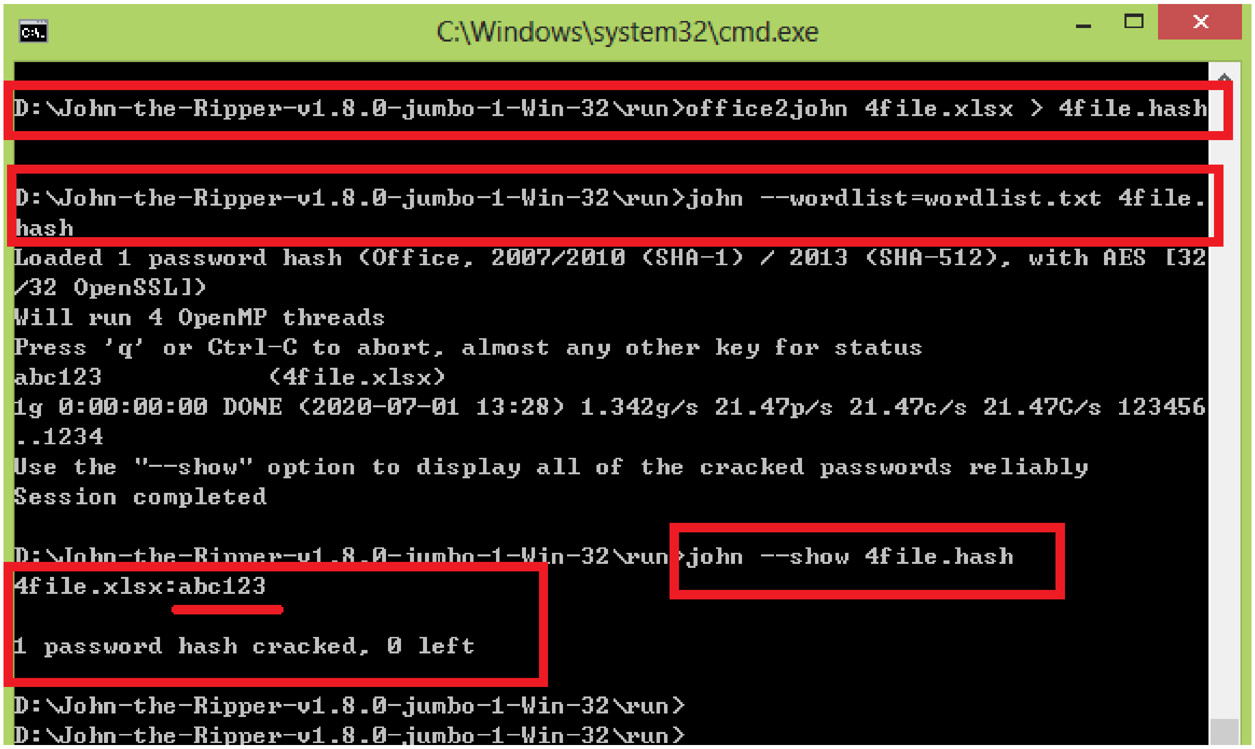


Computer ▸ Personal (D:) ▸ John-the-Ripper-v1.8.0-jumbo-1-Win-32 ▸ run

Name	Date modified	Type	Size
1file	7/1/2020 10:50 AM	HASH File	1 KB
1file	6/15/2020 2:24 AM	Adobe Acrobat D...	640 KB
1password2john	5/16/2014 7:10 PM	Python File	9 KB
2file	7/1/2020 11:53 AM	HASH File	1 KB
2file	7/1/2020 11:53 AM	Compressed (zipp...	1,092 KB
3file	7/1/2020 12:53 PM	Microsoft Word D...	24 KB
3file	7/1/2020 12:55 PM	HASH File	1 KB
4file	7/1/2020 1:09 PM	HASH File	1 KB
4file	7/1/2020 1:08 PM	Microsoft Excel W...	33 KB
7z2john	5/16/2014 7:10 PM	Python File	33 KB
aix2john	5/16/2014 7:10 PM	PL File	1 KB

Figure 19: Password protected EXCEL document

Step 20: Write the command "john --wordlist= wordlist.txt 4file.hash" to compare the hash of PDF file with dictionary. Write the command "*john.exe --show 4file.hash*" to display the passwords in plaintext of protected EXCEL document as shown in Figure 20. The passwords in plaintext are displayed in the Figure 20 and highlighted in red rectangular box.



```
D:\John-the-Ripper-v1.8.0-jumbo-1-Win-32\run>office2john 4file.xlsx > 4file.hash

D:\John-the-Ripper-v1.8.0-jumbo-1-Win-32\run>john --wordlist=wordlist.txt 4file.
hash
Loaded 1 password hash (Office, 2007/2010 (SHA-1) / 2013 (SHA-512), with AES [32
/32 OpenSSL])
Will run 4 OpenMP threads
Press 'q' or Ctrl-C to abort, almost any other key for status
abc123           (4file.xlsx)
1g 0:00:00:00 DONE (2020-07-01 13:28) 1.342g/s 21.47p/s 21.47c/s 21.47C/s 123456
..1234
Use the "--show" option to display all of the cracked passwords reliably
Session completed

D:\John-the-Ripper-v1.8.0-jumbo-1-Win-32\run>john --show 4file.hash
4file.xlsx:abc123

1 password hash cracked, 0 left

D:\John-the-Ripper-v1.8.0-jumbo-1-Win-32\run>
D:\John-the-Ripper-v1.8.0-jumbo-1-Win-32\run>
```

Figure 20: Cracking password of protected EXCEL document in Windows operating system (via dictionary)

REFERENCES

[1] O. S. Limited, “john Package Description,” 2020. https://tools.kali.org/password-attacks/john (accessed May 20, 2020).