

Computer Forensic: John the Ripper(JTR)

Madan Baduwal

Department of Computer Science, University of Texas Permian Basin
baduwal_m63609@utpb.edu


Hash(group_4)	Password
\$y\$j9T\$EsFrFARBYNL6wbmnCa7e1\$p6MMZX5PuL0Rlo/dm7U/wepFs.xQU0A/ProMzcXlGO3	shogun
\$y\$j9T\$QiO5j0r3J26wQob1eX82a/\$eYF77HBqyeNauyp/T5iN8iKxTz944ofNPKGSGuGFig5	Peaches
\$y\$j9T\$SG0vQ7c9k8CWOnXWzHPVA/\$i5CZWNiRufYkBNaMVfSJ0Afjs8aDA5cYFxP.O5A/4X8	Liber7y
\$y\$j9T\$Hq/9KUrele5FBfAGkcZ/O.\$4rYieUbr.8SqG6PWDMtyLjZ50faLkylz8nRvXiwnJB6	Fe8ru4ry
\$y\$j9T\$HnEt.Cm4Fd0BeUwmayOMb/\$Y5zXJIYZaXhZ8oxghbg6TwYsKUTcNV5l7ZQWASmZEw/	

Table 1: Passwords Cracking John the Ripper(jtR), group_4

Github Code: <https://github.com/madanbaduwal/jtR>

Contents

1 Project 3: Analysis, JTR, Group_4	2
1.1 Input files	2
1.2 Hint Patterns	2
1.3 Scripts	3
1.3.1 grp4_0: 6-letter word, all lowercase wordlist generator	3
1.3.2 grp4_1: 7-letter word, first letter caps wordlist generator	3
1.3.3 grp4_2: 7-letter word, one numerical substitution wordlist generator	4
1.3.4 grp4_3: 8-letter word, two subs, first letter caps wordlist generator	5
1.3.5 grp4_4: 8-10 letters, two subs, first letter caps, one additional caps padding up to 12 char total with !@#%&* -+=0123456789, and one of !@#%&* -+= may! appear in the middle wordlist generator	6
1.4 Password screenshot	7
1.4.1 grp4_0_password : shogun	7
1.4.2 grp4_1_password : Peaches	7
1.4.3 grp4_2_password : liber7y	7
1.4.4 grp4_3_password : Fe8ru4ry	8
2 Other Group password cracking	8
2.1 Group_2	8
2.2 Group_5	8
3 Password Cracking John the ripper step-by-step process	9
3.1 Help	9
3.2 Wordlist and hash file	9
3.3 Hash file	10
3.4 Check Hash support by john	11
3.5 Unhash	12
3.6 See the password	12

1 Project 3: Analysis, JTR, Group_4

1.1 Input files

We've supplied the following file.

```
grp4_0:$y$j9T$EsFrrFARBYNL6wbmnCa7e1$P6MMZX5PuL0R1o/dm7U/wepFs.xQU0A/ProMzcX1G03:1021:1021:,,,:/home/grp4_0:/bin/bash
grp4_1:$y$j9T$Q105j0r3J26wQob1eX82a/$eYF77HBqyeNauyp/T5iN8iKxTz944ofNPKGSGuGFig5:1024:1024:,,,:/home/grp4_1:/bin/bash
grp4_2:$y$j9T$SG0vQ7c9k8CWOnXWzHPVA/$i5CZwNiRuFYk8NaMVfSJ0Afjs8aDA5cYFxp.O5A/4X8:1027:1027:,,,:/home/grp4_2:/bin/bash
grp4_3:$y$j9T$Hq/9KUrEle5FBfAGkcZ/O.$4rYieUbr.8SqG6PwDMtyLjZ50faLkYtz8nRvXiwnJ86:1030:1030:,,,:/home/grp4_3:/bin/bash
grp4_4:$y$j9T$HnEt.Cm4Fd0BeUwmayOMb/$Y5zXJIYzaXhZ8oxghbg6TwYsKUTcNV517ZQWASMZEw/:1033:1033:,,,:/home/grp4_4:/bin/bash
```

Figure 1: Input hash file

1.2 Hint Patterns

Here are the patterns provided by the professor.

grp4_0: 6-letter word, all lowercase

eg: attila
tigger
qwerty
carmen

grp4_1: 7-letter word, first letter caps

eg: Volley
Service
Letmein
Mustang
Michael
Patrick

grp4_2: 7-letter word, one numerical substitution

eg: p4ckrat
packr4t
packra7

grp4_3: 8-letter word, two subs, first letter caps

eg: Valh411a
V4lha11a
V41h411a
V41ha11a

grp4_4: 8-10 letters, two subs, first letter caps, one additional caps padding

up to 12 char total with !@#\$\$%&* -+=0123456789, and one of !@#\$\$%&* -+= may! appear in the middle,

eg: H4rryPot7er!

6ignatur8
6ignatur9
7ignatur0
7ignatur1
7ignatur2
7ignatur3
7ignatur4
7ignatur

1.3 Scripts

1.3.1 grp4_0: 6-letter word, all lowercase wordlist generator

Listing 1: 6-letter word, all lowercase wordlist generator

```
1
2 def create_subset_wordlist(input_file, output_file):
3     with open(input_file, 'r') as f:
4         wordlist = f.read().splitlines()
5
6     subset_wordlist = [word.lower() for word in wordlist if len(word) == 6]
7
8     with open(output_file, 'w') as f:
9         f.write('\n'.join(subset_wordlist))
10
11 # Usage example
12 input_file = 'wordlist.txt' # Replace with your input wordlist file
13 output_file = 'grp4_0_wordlist.txt' # Replace with the desired output file name
14 create_subset_wordlist(input_file, output_file)
```

```
123456
abc123
tigger
qwerty
carmen
mickey
secret
summer
a1b2c3
canada
ranger
shadow
...
```

1.3.2 grp4_1: 7-letter word, first letter caps wordlist generator

Listing 2: 7-letter word, first letter caps wordlist generator

```
1 def create_subset_wordlist(input_file, output_file):
2     with open(input_file, 'r') as f:
3         wordlist = f.read().splitlines()
4
5     subset_wordlist = [word.capitalize() for word in wordlist if len(word) == 7]
6
7     with open(output_file, 'w') as f:
8         f.write('\n'.join(subset_wordlist))
9
10 # Usage example
11 input_file = 'wordlist.txt' # Replace with your input wordlist file
12 output_file = 'grp4_1_wordlist.txt' # Replace with the desired output file name
13 create_subset_wordlist(input_file, output_file)
```

```
Service
Letmein
Mustang
Michael
```

Patrick
Diamond
Fuckyou
Matthew
Chelsea
Freedom
Gandalf
Newyork
Dorothy
Fishing
...

1.3.3 grp4_2: 7-letter word, one numerical substitution wordlist generator

Listing 3: 7-letter word, one numerical substitution wordlist generator

```
1 def generate_substituted_words(word):
2     substituted_words = []
3     for i in range(len(word)):
4         if word[i].isalpha():
5             for digit in '0123456789':
6                 substituted_word = word[:i] + digit + word[i+1:]
7                 substituted_words.append(substituted_word)
8     return substituted_words
9
10 def create_subset_wordlist(input_file, output_file):
11     with open(input_file, 'r') as f:
12         wordlist = f.read().splitlines()
13
14     subset_wordlist = []
15     for word in wordlist:
16         if len(word) == 7:
17             subset_wordlist.extend(generate_substituted_words(word))
18
19     with open(output_file, 'w') as f:
20         f.write('\n'.join(subset_wordlist))
21
22 # Usage example
23 input_file = 'wordlist.txt' # Replace with your input wordlist file
24 output_file = 'grp4_2_wordlist.txt' # Replace with the desired output file name
25 create_subset_wordlist(input_file, output_file)
```

servic4
servic5
servic6
servic7
servic8
servic9
0etmein
1etmein
2etmein
3etmein
4etmein
5etmein
6etmein
7etmein

```
8etmein
9etmein
10tmein
11tmein
12tmein
13tmein
...
```

1.3.4 grp4_3: 8-letter word, two subs, first letter caps wordlist generator

Listing 4: 8-letter word, two subs, first letter caps wordlist generator

```
1 import itertools
2
3 def generate_substituted_words(word):
4     substituted_words = []
5     for indices in itertools.combinations(range(len(word)), 2):
6         for replacement in itertools.product('0123456789', repeat=2):
7             substituted_word = list(word)
8             for index, digit in zip(indices, replacement):
9                 substituted_word[index] = digit
10            substituted_words.append(''.join(substituted_word))
11    return substituted_words
12
13 def create_subset_wordlist(input_file, output_file):
14     with open(input_file, 'r') as f:
15         wordlist = f.read().splitlines()
16
17     subset_wordlist = []
18     for word in wordlist:
19         if len(word) == 8:
20             subset_wordlist.extend(generate_substituted_words(word.capitalize()))
21
22     with open(output_file, 'w') as f:
23         f.write('\n'.join(subset_wordlist))
24
25 # Usage example
26 input_file = 'wordlist.txt' # Replace with your input wordlist file
27 output_file = 'grp4_3_wordlist.txt' # Replace with the desired output file name
28 create_subset_wordlist(input_file, output_file)
```

```
00ssword
01ssword
02ssword
03ssword
04ssword
05ssword
06ssword
07ssword
08ssword
09ssword
10ssword
11ssword
12ssword
13ssword
14ssword
```

15ssword

...

1.3.5 grp4_4: 8-10 letters, two subs, first letter caps, one additional caps padding up to 12 char total with !@#%&* -+=0123456789, and one of !@#%&* -+= may! appear in the middle wordlist generator

Listing 5: 8-10 letters, two subs, first letter caps, one additional caps padding up to 12 char total with !@#%&* -+=0123456789, and one of !@#%&* -+= may! appear in the middle wordlist generator

```
1 import itertools
2
3 import itertools
4
5 def generate_substituted_words(word):
6     substituted_words = []
7     for indices in itertools.combinations(range(len(word)), 2):
8         for replacement in itertools.product('0123456789', repeat=2):
9             substituted_word = list(word)
10            for index, digit in zip(indices, replacement):
11                substituted_word[index] = digit
12            substituted_words.append(''.join(substituted_word))
13    return substituted_words
14
15 def generate_wordlist(input_file):
16     characters = 'abcdefghijklmnopqrstuvwxyz'
17     digits = '0123456789'
18     specials = '!@#%&* -+= '
19     padding = 'abcdefghijklmnopqrstuvwxyz0123456789!@#%&* -+= '
20
21     wordlist = []
22     with open(input_file, 'r') as f:
23         for line in f:
24             word = line.strip()
25             if 8 <= len(word) <= 10:
26                 for word_chars in itertools.combinations(characters, len(word) - 2):
27                     for caps_padding in itertools.combinations(padding, 2):
28                         for special in specials:
29                             word = ''.join(word_chars)
30                             word = word.capitalize() + ''.join(caps_padding) + special
31                             wordlist.extend(generate_substituted_words(word))
32     return wordlist
33
34 def create_subset_wordlist(input_file, output_file):
35     wordlist = generate_wordlist(input_file)
36
37     with open(output_file, 'w') as f:
38         f.write('\n'.join(wordlist))
39
40 # Usage example
41 input_file = 'wordlist.txt' # Replace with your input wordlist file
42 output_file = 'grp4_4_wordlist.txt' # Replace with the desired output file name
43 create_subset_wordlist(input_file, output_file)
```

Signatur8

Signatur9

Signatur0

Signatur1

```

6signatur2
6signatur3
6signatur4
6signatur5
6signatur6
6signatur7
6signatur8
6signatur9
7signatur0
7signatur1
7signatur2
7signatur3
7signatur4
7signatur5
...

```

1.4 Password screenshot

1.4.1 grp4_0_password : shogun

```

madanbaduwal@pop-os: ~/Desktop/jtR$ john --show grp4_0_hash.txt
grp4_0:shogun:1021:1021:,,,:/home/grp4_0:/bin/bash

madanbaduwal@pop-os: ~/Desktop/jtR$ john --format=crypt --wordlist=password.lst grp4_0_hash.txt

```

Figure 2: grp4_0_password : shogun

1.4.2 grp4_1_password : Peaches

```

madanbaduwal@pop-os: ~/Desktop/jtR$ john --show grp4_1_hash.txt
grp4_1:Peaches:1024:1024:,,,:/home/grp4_1:/bin/bash

1 password hash cracked, 0 left
madanbaduwal@pop-os: ~/Desktop/jtR$ █

```

Figure 3: grp4_1_password : Peaches

1.4.3 grp4_2_password : liber7y

```

madanbaduwal@pop-os: ~/Desktop/jtR$ john --format=crypt --wordlist=subset_wordlist.txt ./input-files/grp4_2_hash.txt
Loaded 1 password hash (crypt, generic crypt(3) [?/64])
Will run 8 OpenMP threads
Press 'q' or Ctrl-C to abort, almost any other key for status
0g 0:00:00:04 2% 0g/s 232.2p/s 232.2c/s 232.2C/s 2ichard..sc7oter
0g 0:00:00:17 8% 0g/s 231.3p/s 231.3c/s 231.3C/s rebe6ca..1xcvbnm
0g 0:00:00:33 16% 0g/s 229.2p/s 229.2c/s 229.2C/s ja4mine..jeff9ey
0g 0:00:00:47 23% 0g/s 228.4p/s 228.4c/s 228.4C/s bi4gles..biol9gy
liber7y (grp4_2)
1g 0:00:00:59 100% 0.01677g/s 227.1p/s 227.1c/s 227.1C/s l0berty..lin5say
Use the "--show" option to display all of the cracked passwords reliably
Session completed
madanbaduwal@pop-os: ~/Desktop/jtR$ john --show ./input-files/grp4_2_hash.txt
grp4_2:liber7y:1027:1027:,,,:/home/grp4_2:/bin/bash

1 password hash cracked, 0 left

```

Figure 4: grp4_2_password : liber7y

1.4.4 grp4_3_password : Fe8ru4ry

```

madanbaduwal@pop-os:~/Desktop/jtR$ john --format=crypt --wordlist=./scripts/subset_wordlist.txt ./input-files/grp4_3_hash.txt
Loaded 1 password hash (crypt, generic crypt(3) [?/64])
Will run 8 OpenMP threads
Press 'q' or Ctrl-C to abort, almost any other key for status
0g 0:00:00:02 0% 0g/s 229.3p/s 229.3c/s 229.3C/s 7asswor2..P67sword
0g 0:00:00:10 0% 0g/s 230.2p/s 230.2c/s 230.2C/s Pass9or6..Passw91d
0g 0:00:00:25 0% 0g/s 229.3p/s 229.3c/s 229.3C/s 6o0puter..5om5uter
0g 0:00:00:50 0% 0g/s 226.5p/s 226.5c/s 226.5C/s 2ase0all..1aseb5ll
0g 0:00:02:00 1% 0g/s 212.7p/s 212.7c/s 212.7C/s 3oot6all..3ootb1ll
0g 0:00:03:04 2% 0g/s 206.5p/s 206.5c/s 206.5C/s Po1hbe6r..Po1hbea1
0g 0:00:04:48 4% 0g/s 206.2p/s 206.2c/s 206.2C/s 2hithea4..S19thead
0g 0:00:08:12 7% 0g/s 201.5p/s 201.5c/s 201.5C/s Bl60bird..Bl5e5ird
0g 0:00:09:23 8% 0g/s 202.8p/s 202.8c/s 202.8C/s Secu40ty..Secu3i5y
0g 0:00:10:36 9% 0g/s 201.5p/s 201.5c/s 201.5C/s Flet5h2r..Flet4he7
0g 0:00:13:30 12% 0g/s 207.0p/s 207.0c/s 207.0C/s Penel12e..Penel0p7
0g 0:00:26:55 26% 0g/s 216.0p/s 216.0c/s 216.0C/s Jen56fer..Jen5i1er
0g 0:00:29:36 29% 0g/s 216.7p/s 216.7c/s 216.7C/s Be60ardo..Be5n5rdo
0g 0:00:34:20 33% 0g/s 217.3p/s 217.3c/s 217.3C/s 32oggies..2r7ggies
0g 0:00:38:15 37% 0g/s 214.9p/s 214.9c/s 214.9C/s 4ateri4a..3aterin9
0g 0:00:41:27 40% 0g/s 214.0p/s 214.0c/s 214.0C/s 1apol6on..1apole1n
0g 0:00:44:19 42% 0g/s 211.8p/s 211.8c/s 211.8C/s 3edsk2ns..2edski7s
0g 0:00:46:24 44% 0g/s 211.1p/s 211.1c/s 211.1C/s Scoote04..Scoote99
0g 0:00:50:27 47% 0g/s 210.2p/s 210.2c/s 210.2C/s W7lf6ang..W7lfg1ng
0g 0:00:52:20 49% 0g/s 209.9p/s 209.9c/s 209.9C/s Bl2zz0rd..Bl1zza5d
0g 0:00:55:35 52% 0g/s 209.1p/s 209.1c/s 209.1C/s 3oftb2ll..2oftba7l
0g 0:00:58:29 55% 0g/s 208.7p/s 208.7c/s 208.7C/s 1q8w344r..1q7w3e9r
0g 0:01:02:40 58% 0g/s 206.7p/s 206.7c/s 206.7C/s Cri0t4na..Cri9t9na
0g 0:01:07:43 62% 0g/s 204.3p/s 204.3c/s 204.3C/s Le0nar0o..Le9nar5o
0g 0:01:09:38 64% 0g/s 204.1p/s 204.1c/s 204.1C/s Sp6rt8ng..Sp6rti3g
0g 0:01:13:53 67% 0g/s 203.1p/s 203.1c/s 203.1C/s Za7efro6..Zac71ron
Fe8ru4ry
(grp4_3)
1g 0:01:18:53 100% 0.000211g/s 204.4p/s 204.4c/s 204.4C/s Fe8ru4ry..Fe7rua9y
Use the "--show" option to display all of the cracked passwords reliably
Session completed
madanbaduwal@pop-os:~/Desktop/jtR$ john --show ./input-files/grp4_3_hash.txt
grp4_3:Fe8ru4ry:1030:1030:,,,:/home/grp4_3:/bin/bash

1 password hash cracked, 0 left
madanbaduwal@pop-os:~/Desktop/jtR$ 

```

Figure 5: grp4_3_password : Fe8ru4ry

2 Other Group password cracking

2.1 Group_2


Hash(group_2)	Password
\$y\$j9T\$NV8mcQw/ZGuYvEvfEDqpy0\$rf9DmCLdprb73GCJfMdE8M4joKicjl6jLtIn9W6Mrh8	speedo
\$y\$j9T\$zvB6Hjg1zvGzqM1GmV/hI0\$hLPPWkfMPYOkSLH3CA2b6kQeT38CZNbOMQk2Fn2CUq4	Sabbath
\$y\$j9T\$g3WGvqZOUhVBe2MSEm7vj/\$heB2ACtGBS8z7s4.Sev62x1AeFE1OceLWVVFHtjrso6	carn4ge
\$y\$j9T\$Y/1ribmq86M/2uZ2HN1pf0\$yz6rluQHSY8ZUDzf4LvP6YRS5lw1S8k/hLfBd4KJyy.	M0ntre4l
\$y\$j9T\$5xIOFnDWWvQJ1WXQy.Trwk/\$rJvtVrm4br74qBAdczRyBPruzWY44L5WL8/e7LULGd6	

Table 2: Passwords Cracking John the Ripper(jtR), group_2

2.2 Group_5


Hash(group_5)	Password
\$y\$j9T\$NunaZCpRsJeQp5zYH7qwr0\$.1feCokdBhDiGhMHMQSswC2m23MFR0AW6nY/0JdkP86	trophy
\$y\$j9T\$HmNxkfkIuPEsMz4MuQgsz1\$8ba7B4lfszNTlhvSZyrwcmvkUvfZJmMz/LSScpszmx7	Swimmer
\$y\$j9T\$Igt24b2mPkI6TH5L8xMNo1\$OLN6fTUmqfhLVDtgwvrcv4ozHRZLkHBLMFQpHkNkEJO	upsil0n
\$y\$j9T\$4IQQFvwi6w50tUgAZpElD0\$5.xoDWt1as9pNCFdrj8htONITWuroCi3o8LlwhYbLZC	Eins5te1n
\$y\$j9T\$trJivi4QYHqmHTo27kbYw.\$waieWSP3zXxf0pieJ2sLTkMNLyiwM9cnnLx9GzLumS5	

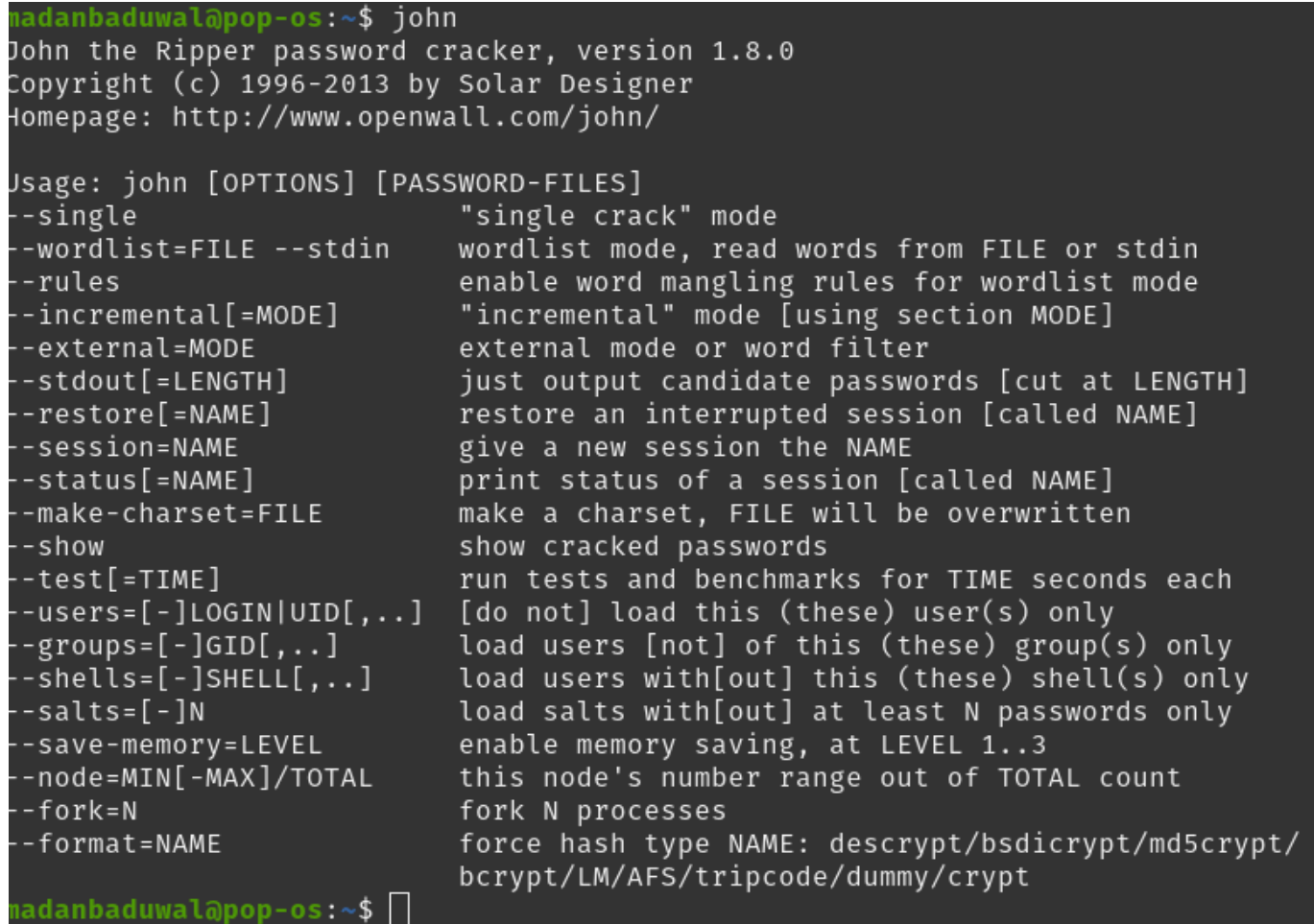
Table 3: Passwords Cracking John the Ripper(jtR), group_5

3 Password Cracking John the ripper step-by-step process

3.1 Help

Check the version and supported format by the john the ripper(hash algorithm support by the version):

```
1 $ John or john --help
```



```
nadanbaduwal@pop-os:~$ john
John the Ripper password cracker, version 1.8.0
Copyright (c) 1996-2013 by Solar Designer
Homepage: http://www.openwall.com/john/

Usage: john [OPTIONS] [PASSWORD-FILES]
--single                "single crack" mode
--wordlist=FILE --stdin  wordlist mode, read words from FILE or stdin
--rules                 enable word mangling rules for wordlist mode
--incremental[=MODE]    "incremental" mode [using section MODE]
--external=MODE         external mode or word filter
--stdout[=LENGTH]       just output candidate passwords [cut at LENGTH]
--restore[=NAME]        restore an interrupted session [called NAME]
--session=NAME          give a new session the NAME
--status[=NAME]         print status of a session [called NAME]
--make-charset=FILE     make a charset, FILE will be overwritten
--show                 show cracked passwords
--test[=TIME]           run tests and benchmarks for TIME seconds each
--users=[-]LOGIN|UID[,..] [do not] load this (these) user(s) only
--groups=[-]GID[,..]    load users [not] of this (these) group(s) only
--shells=[-]SHELL[,..]  load users with[out] this (these) shell(s) only
--salts=[-]N            load salts with[out] at least N passwords only
--save-memory=LEVEL     enable memory saving, at LEVEL 1..3
--node=MIN[-MAX]/TOTAL  this node's number range out of TOTAL count
--fork=N               fork N processes
--format=NAME          force hash type NAME: descript/bsdictcrypt/md5crypt/
                      bcrypt/LM/AFS/tripcode/dummy/crypt

nadanbaduwal@pop-os:~$
```

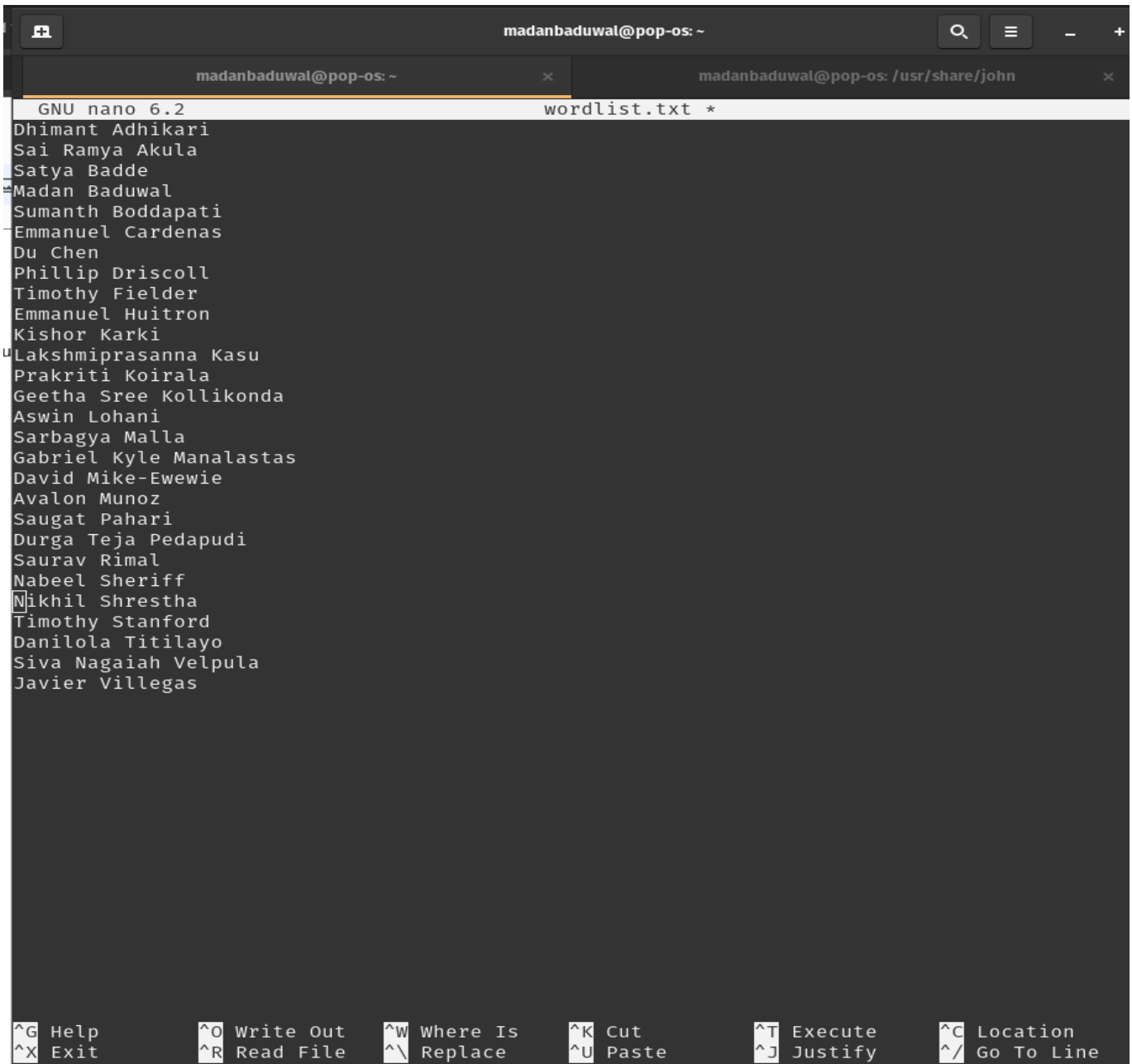
Figure 6: John help

- You can see the version 1.8.0 and format descript/bsdictcrypt/md5crypt/bcrypt/LM/AFS/tripcode/dummy/crypt
- Note: John the ripper can't unhash file which is not supported by this version

3.2 Wordlist and hash file

- You can use your custom wordlist or you can use the wordlist provided by john the ripper
- I am creating custom wordlistst here
- Third item

```
1 $ touch wordlist.txt
2 $ nano touchlist.txt
```



The image shows a terminal window with a dark background. At the top, there are two tabs: 'madanbaduwal@pop-os: ~' and 'madanbaduwal@pop-os: /usr/share/john'. The active tab is the first one. Below the tabs, the terminal shows the output of the 'cat' command on 'wordlist.txt'. The list of names is as follows:

```
GNU nano 6.2 wordlist.txt *
Dhimant Adhikari
Sai Ramya Akula
Satya Badde
Madan Baduwal
Sumanth Boddapati
Emmanuel Cardenas
Du Chen
Phillip Driscoll
Timothy Fielder
Emmanuel Huitron
Kishor Karki
Lakshmiprasanna Kasu
Prakriti Koirala
Geetha Sree Kollikonda
Aswin Lohani
Sarbagya Malla
Gabriel Kyle Manalastas
David Mike-Ewewie
Avalon Munoz
Saugat Pahari
Durga Teja Pedapudi
Saurav Rimal
Nabeel Sheriff
Nikhil Shrestha
Timothy Stanford
Danilola Titilayo
Siva Nagaiah Velpula
Javier Villegas
```

At the bottom of the terminal, there is a status bar with various keyboard shortcuts and their functions:

^G Help	^O Write Out	^W Where Is	^K Cut	^T Execute	^C Location
^X Exit	^R Read File	^_ Replace	^U Paste	^J Justify	^_ Go To Line

Figure 7: John touch

3.3 Hash file

Create hash file which is supported by this version: There are lot of online tools which helps to create hash value, I am using this: [Bcrypt](#)

```
1 $ touch hash.txt
2 $ nano hash.txt
```



Figure 8: John Bcrypt

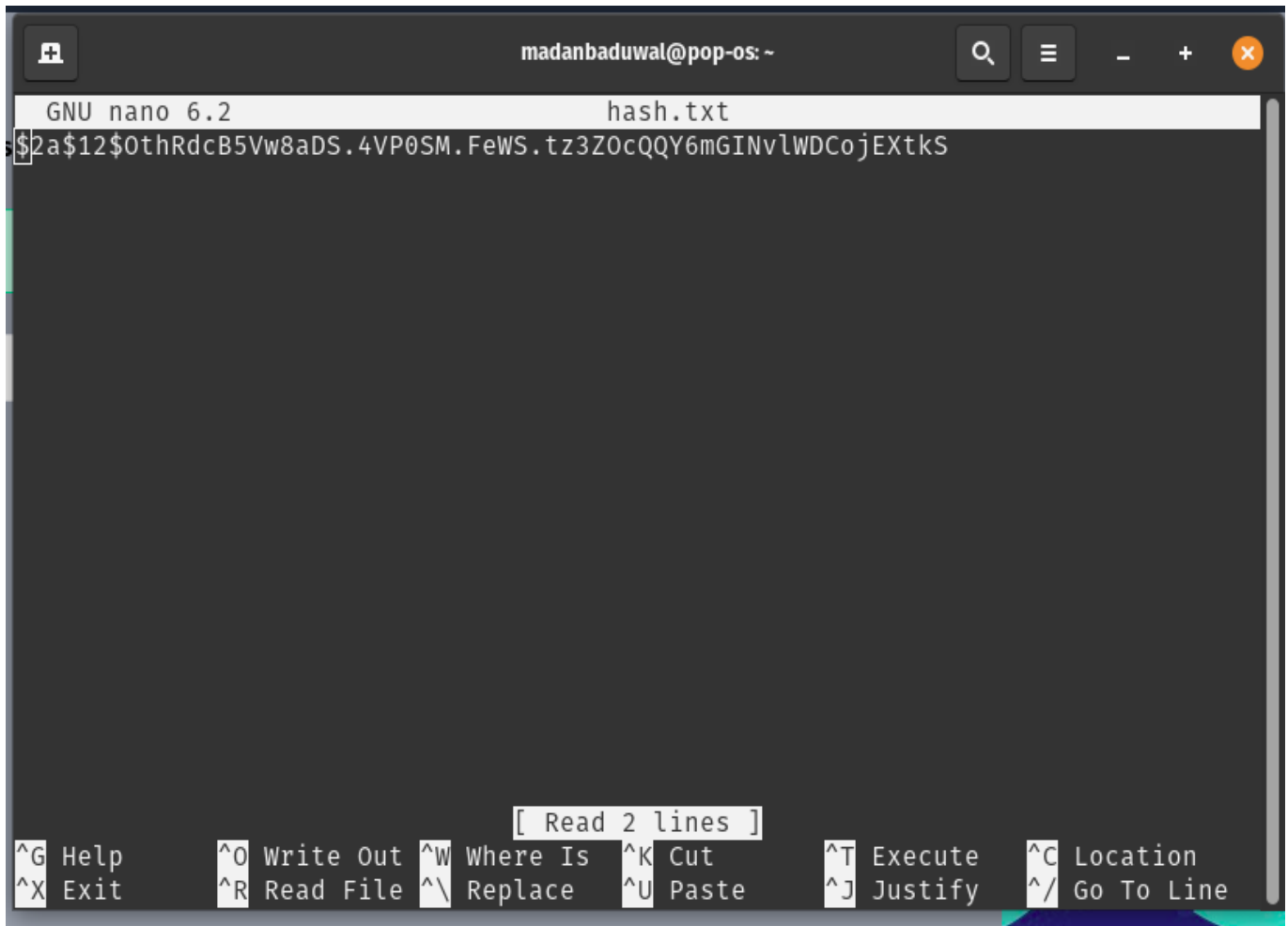


Figure 9: John Bcrypt

3.4 Check Hash support by john

Make sure which kind of hash is used by your hash file or hash

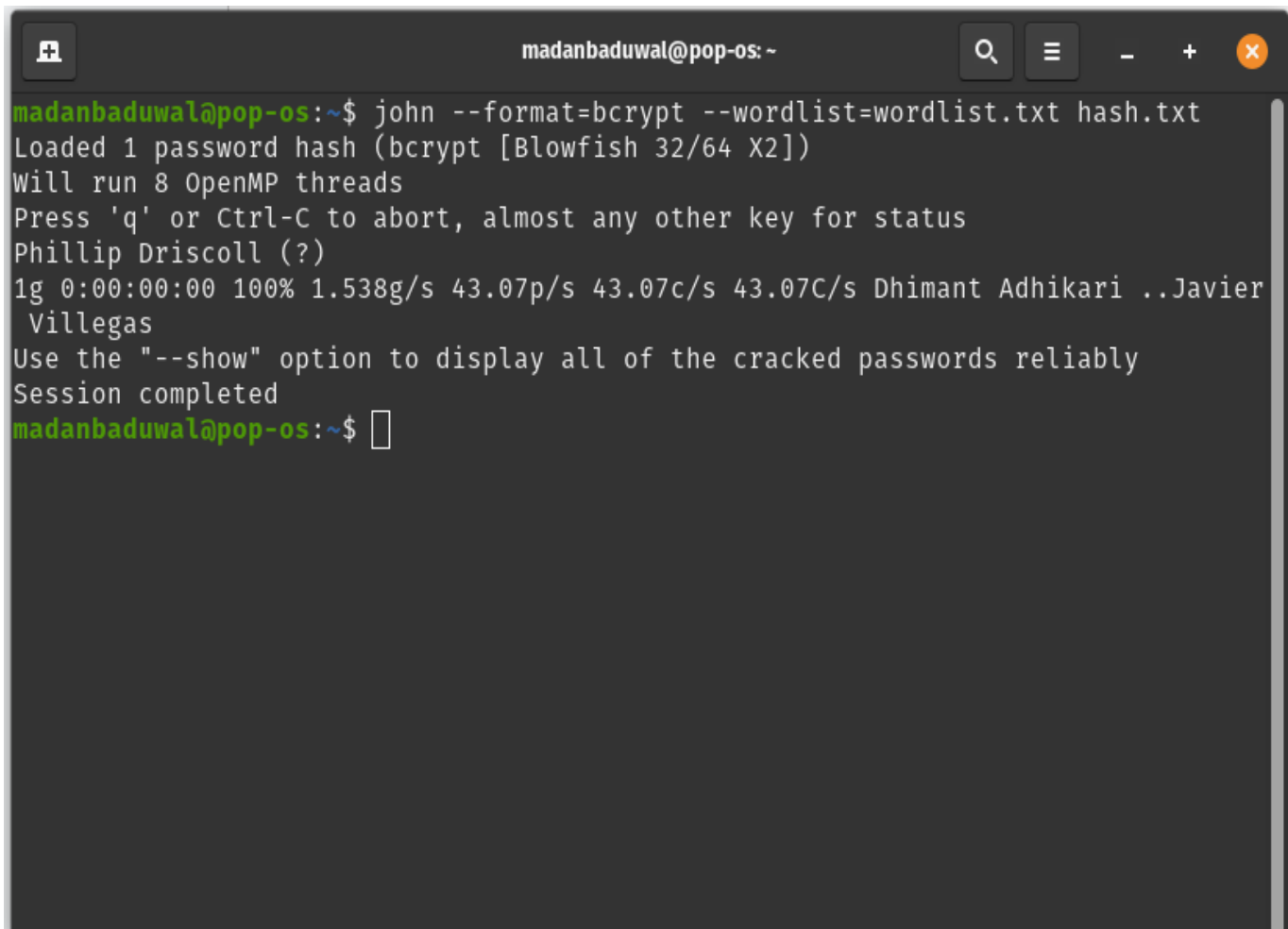
- hashid ‘hashfile’ or ‘hash value’ (If you don’t have hashid, then you can install using command ‘sudo apt-get install hashid’)
- Example : hashid

```
$'$2a$12$OthRdcB5Vw8aDS.4VP0SM.FeWS.tz3ZOcQQY6mGINvIWDCojEXtkS"
[+] Blowfish(OpenBSD)
[+] Woltlab Burning Board 4.x
[+] bcrypt
```

3.5 Unhash

Unhash the password using john the ripper

```
1 $ john --format=bcrypt --wordlist=wordlist.txt hash.txt
```

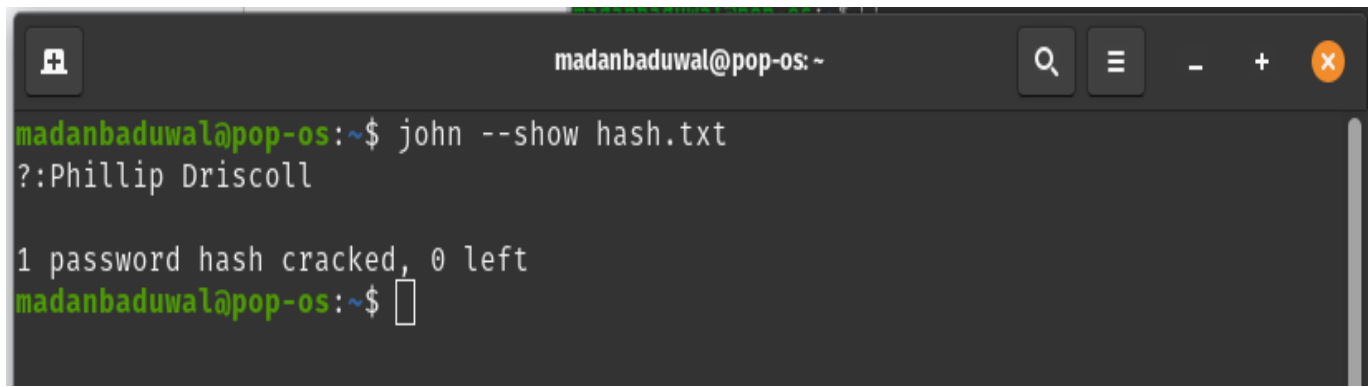


```
madanbaduwal@pop-os:~$ john --format=bcrypt --wordlist=wordlist.txt hash.txt
Loaded 1 password hash (bcrypt [Blowfish 32/64 X2])
Will run 8 OpenMP threads
Press 'q' or Ctrl-C to abort, almost any other key for status
Phillip Driscoll (?)
1g 0:00:00:00 100% 1.538g/s 43.07p/s 43.07c/s 43.07C/s Dhimant Adhikari ..Javier
Villegas
Use the "--show" option to display all of the cracked passwords reliably
Session completed
madanbaduwal@pop-os:~$
```

Figure 10: John Unhash

3.6 See the password

```
1 $ john --show hash.txt
```

A terminal window with a dark background. The title bar shows the user 'madanbaduwal' on a 'pop-os' machine. The terminal text shows the command 'john --show hash.txt' being executed, followed by the output '?:Phillip Driscoll' and '1 password hash cracked, 0 left'. The prompt returns to the user's shell.

```
madanbaduwal@pop-os: ~$ john --show hash.txt
?:Phillip Driscoll

1 password hash cracked, 0 left
madanbaduwal@pop-os: ~$
```

Figure 11: John See Password