## JTR CHEAT SHEET

This cheat sheet presents tips and tricks for using JtR.

### JtR Community Edition - Linux

Download the JtR Bleeding Jumbo edition with improved capabilities and other goodies.

```
git clone https://github.com/magnumripper/JohnTheRipper -b bleeding-jumbo
```

Compile JtR and enable/disable required features

```
cd JohnTheRipper/
cd src/
./configure
make clean && make
```

Enable bash completion. add the following line to your ~/.bashrc

```
> PATH=/usr/local/bin:/usr/local/git/bin:
> /usr/local/jtR/bin:
> $HOME/.john/bin:
> $HOME/.john/bin:jtR:
> 
> export PATH
```

### Cracking Modes

**Wordlist Mode** (dictionary attack)

```
./john --wordlist=password.lst hashfile
```

**Mangling Rules Mode** (hybrid)

```
./john --wordlist=password.lst -rules:run rulename hashfile
```

**Incremental mode** (Brute Force)

```
./john --incremental hashfile
```

**External mode** (use a program to generate guesses)

```
./john --external: <rulename> hashfile
```

**Loopback mode** (use POT as wordlist)

```
./john --loopback hashfile
```

**Mask mode** (read MASK under /doc)

```
./john --mask=1111111111111111 -l=A-Z hashfile -min-len=8
```

**Hybrid Mask mode**

```
./john -w=password.lst -mask='11111111' hashfile
```

**Markov mode** (Read MARKOV under /doc).

First generate Markov stats:

```
./calc.stat wordlist markovstats
```

Then run:

```
./john -markov:200 -max-len:12 hashfile --mkv-stats=markovstats
```

**Prince mode** (Read PRINCE under /doc)

```
./john --prince=wordlist hashfile
```

Most modes have MaxLen=13 in John.conf but it can be overwritten with -max-len=N up to 24

### Multiple CPU or GPU

List OpenCL devices and get the device id

```
> ./john --list=opencl-devices
```

List formats supported by OpenCL

```
> ./john --list=formats --format=opencl
```

**Multiple GPU's**

```
> ./john hashfile --format:<openclformat> --wordlist:<> --rules:<> --dev=0,1 --fork=2
```

**Multiple CPU's** (e.g., 4 cores)

```
> ./john hashfile --wordlist:<> --rules:<> --dev=2 --fork=4
```

### Rules

**--rules:Single**

**--rules:wordlist**

**--rules:Extra**

**--rules:Jumbo (all the above)**

**--rules:KoreLogic**

**--rules:All (all the above)**

### Incremental Modes (Brute Force)

**--incremental:Lower (26 char)**

**--incremental:Alpha (52 char)**

**--incremental:Digits (10 char)**

**--incremental:Alnum (62 char)**

### Incremental mode with new charsets

Create a new charset based on john.pot

```
> ./john --make-charset=charset.chr
```

Create a new entry in John.conf to accommodate the new charset

```
# Incremental modes
[Incremental:charset]
File = $JOHN/charset.chr
MinLen = 0
MaxLen = 31
CharCount = 95
```

Run JtR with the new charset

```
> ./john --incremental=charset hashfile
```

### Wordlists

Sort a wordlist to use with wordlist rule mode

```
$tr A-Z a-z < SOURCE | sort -u > TARGET
```

Use a POT file to generate a new wordlist

```
cut -d: -f2 john.pot | sort -u > pot_dic
```

Generate candidate passwords for slow hashes.

```
> ./john --wordlist= password.lst --stdout --rules:jumbo | ./unique --mem=25 wordlistuniq
```

### Use external mode for complex rules


Generate a wordlist that meets the complexity specified in the complex filter

```
> ./john --wordlist=[path to word list] --stdout --external:[filter name] > [path to output list]
```

Try sequences of adjacent keys on a keyboard as candidate passwords

```
john --external:Keyboard hashfile
```

### Configuration Items on John.conf

When using both CPU and GPU set this flag

```
Idle = N
```

### Hidden Options

```
> ./john --list=hidden-options
```

### Display guesses

```
> ./john --incremental:Alpha --stdout --session=s1
```

### Generate guesses with external program

```
crunch 1 6 abcdefg | ./john hashes -stdin --session=s1
```

### Session and Restore

```
> ./john hashes --session=name
> ./john --restore:name
```

### Show cracked passwords

```
> ./john hashes --pot=<>
```

### Resources

- John-Users Mailing List
- JtR Community Edition
- Resources

---

*Authors by Luis Rocha. This cheat sheet was reviewed by John-Users. It’s distributed according to the Creative Commons v3 “Attribution” License. You’re looking at version 1.0 of this document.*
Mangling Rules Generation

- Insert and Delete Wordlist Rules
  - # Remove the first char from the word
  - # Remove the last char from the word
  - # Delete the character in position N
  - # Extract substring from position N for up to M characters
  - # Insert character X in position N and shift the rest right
  - # Overstrike character in position N with character X

- Simple Wordlist Rules
  - # Lowercase the first character, and uppercase the rest
  - # Toggle case of all characters in the word
  - # Toggle case of the character in position N
  - # Reverse "Fred" -> "derF"
  - # Duplicate "Fred" -> "FredFred"
  - # Reflect: "Fred" -> "RedFred"
  - # Rotate the word left: "jsmith" -> "smithj"
  - # Rotate the word right: "smithj" -> "jsmith"
  - # Append character X to the word
  - # Prefix the word with character X

- Charset and Conversion Wordlist Rules
  - # Shift case: "Crack96" -> "cRACK(^"
  - # Lowercase vowels, uppercase consonants: "Crack96" -> "CraCK96"
  - # Shift each character right, by keyboard: "Crack96" -> "Vtsvl07"
  - # Shift each character left, by keyboard: "Crack96" -> "XeaXj85"

- Length Control
  - # Reject the word unless it is less than N characters long
  - # Reject the word unless it is greater than N characters long
  - # Truncate the word at length N

- Dictionaries
  - Generate wordlists from Wikipedia pages: wget https://raw.githubusercontent.com/zombie/wikigen/master/wwg.py
  - Generate wordlists from Aspell Dict's:
    - aspell dump dicts
    - sudo apt-get install aspell-es
    - aspell -d es dump master | aspell -l es expand | awk '{RS="\n"} > Spanish.dic

- Resources
  - Password Analysis and Cracking Kit https://thesprawl.org/projects/pack/

Authored by Luis Rocha. This cheat sheet was reviewed by John-Users. It's distributed according to the Creative Commons v3 "Attribution" License. You’re looking at version 1.1 of this document.