John the Ripper — Professional Cheat Sheet

One-page professional reference for **John the Ripper (JtR)** — password-cracking suite (core + Jumbo). Commands, modes, file formats, examples, tuning tips, and OPSEC notes for pentesters and forensic analysts.

---

1) At-a-glance

- **Tool:** `john` (John the Ripper) — versatile password cracker with multiple attack modes: single, wordlist, incremental, mask, and rule-based. `john-jumbo` adds many formats and enhancements.
- **Primary uses:** Offline hash cracking (passwd/NTLM/MD5/sha*/bcrypt/etc.), auditing password strength, processing /etc/shadow, and integrating with wordlists and rules.
- **Note:** Choose `john` (core) for simple tasks; install `john-jumbo` for widest format support.

---

2) Install / check

```
# Kali / Debian
sudo apt update && sudo apt install john
# For Jumbo features, install john-data and jumbo package if available or build from source
# From source (recommended for latest jumbo)
git clone https://github.com/openwall/john.git
cd john/src
./configure && make -s && sudo make install

# Check version / formats
john --version
john --list=formats | less
```

---

3) Common files & helpers

- **Potfile (cracked passwords):** `~/.john/john.pot` (default) — stores cracked passwords.
- **Unshadow / unafs / unshadowing:** `unshadow passwd shadow > unshadowed.txt` to combine `/etc/passwd` & `/etc/shadow` for cracking.
- **Unique helpers:** `zip2john`, `rar2john`, `pdf2john`, `ssh2john`, `keepass2john`, etc., to extract hashes from containers.

---

4) Core commands & options

- `john [options] <hashfile>` — main command.
- `--wordlist=<file>` or `-w:<file>` — use wordlist/wordlist mode.
- `--rules` or `--rules=<name>` — apply word mangling rules (e.g., `--rules=Jumbo` or `--rules:wordlist`).
- `--incremental[=mode]` — brute-force incremental mode (fast, charset-based). Modes listed in `john --list=build-info` or `john --list=incremental`.
- `--mask=<mask>` — mask mode (highly efficient for structured passwords), supports placeholders: `?l?l?d?d` etc.
- `--format=<format>` — force hash format (e.g. `--format=NT`, `--format=md5crypt`, `--format=raw-md5`, `--format=bcrypt`).
- `--stdout` — pipe-mode to generate candidate words (useful with `rules` or `mangling`).
- `--show` — show cracked passwords from potfile for a hashfile: `john --show hashes.txt`.
- `--restore[=name]` — resume saved session.
- `--session=<name>` — name the session for restore and parallel runs.
- `--fork=N` — run N parallel processes (useful on multi-core), only for some modes (e.g., `--fork=4 --incremental`).
- `--format=dynamic_...` — dynamic formats (Jumbo) for exotic hashes.
- `--pipe` — read candidates from stdin (e.g., `hashcat --stdout | john --stdin` or `wordlist | john --stdin --rules`).

---

5) Practical examples

5.1 Prepare / unshadow local passwd

```
sudo unshadow /etc/passwd /etc/shadow > local_unshadow.txt
john --wordlist=/usr/share/wordlists/rockyou.txt local_unshadow.txt --rules --format=sha512crypt
```

5.2 Wordlist + rules (fast)

```
john --wordlist=/usr/share/wordlists/rockyou.txt --rules --format=NT hashes.txt
# or shorthand
john -w:rockyou.txt --rules hashes.txt
```

5.3 Mask attack (targeted brute force)

```
# 1 uppercase, 5 lowercase, 2 digits
john --mask='?u?l?l?l?l?l?d?d' --format=raw-md5 hashes.txt --fork=4 --session=maskrun
```

5.4 Incremental brute force (charset defined in john.conf)

```
john --incremental=All --format=raw-md5 hashes.txt --fork=8
```

5.5 Use specialized extractor for archive/pfile

```
zip2john secret.zip > secret.hash
john secret.hash -w:wordlist.txt --format=zip
```

5.6 Pipe candidates from stdout (rule mangling)

```
# generate candidates via rules; useful to chain or test rules
john --wordlist=rockyou.txt --rules --stdout | head -n 1000
# pipe into john stdin mode (reads candidates)
john --stdin --format=NT hashes.txt < candidates.txt
```

5.7 Show cracked / restore / status

```
john --show hashes.txt      # display cracked accounts
john --status               # show running status for default session
john --restore=mysess       # restore session named mysess
```

---

6) Tuning & performance tips

- **Use** `--format` to avoid format detection overhead and select optimized kernels in Jumbo.
- **Prefer mask attacks** over full incremental where you know structure (mask is much faster).
- **Use** `--fork` to utilize multiple cores — for CPU-bound modes like `--incremental` or `--mask`.
- **Offload to GPU:** John's `jumbo` supports OpenCL (`john --format=raw-md5-opencl`) — use GPU builds where available (but Hashcat is often faster for GPUs).
- **Use** `--session` **and** `--restore` for long runs to resume after interruptions.
- **Optimize wordlists:** pre-process (remove duplicates, sort by probability) and use `--rules` carefully to expand effectively.

---

7) Integration & workflow

- **Hash extraction:** use format-specific `*2john` tools (e.g., `ssh2john`, `pdf2john`) then feed to `john`.
- **Hybrid attacks:** generate mutated candidates via `--stdout` and pipe into `john --stdin` or `hashcat` for GPU work.

- **Password policy checks:** use `--show` and `john --list=rules` to analyze weak passwords.

8) Potfile, logging & forensic handling

- **Potfile location:** `~/.john/john.pot` — back it up for evidence. Use `--pot=<file>` to specify alternative.
- **Export results:** `john --show=left hashes.txt > cracked.txt` or parse `john.pot`.
- **Chain-of-custody:** preserve original hash files, timestamp actions, and include exact commands in reports.

9) OPSEC / legal & ethical notes

- Always have explicit authorization and scope for offline cracking.
- Cracking passwords of accounts outside scope is illegal and unethical.
- Keep sensitive cracked credentials secure; only include necessary artifacts in deliverables.

10) Quick cheats (copy-paste)

```
# Combine passwd+shadow and run rockyou+rules
sudo unshadow /etc/passwd /etc/shadow > unsh.txt
john -w:/usr/share/wordlists/rockyou.txt --rules unsh.txt

# Zip file
zip2john secret.zip > z.hash
john z.hash -w:rockyou.txt

# Mask attack (example)
john --mask='?u?l?l?l?l?d?d?d' --format=raw-md5 hashes.txt --fork=4

# Show cracked
john --show hashes.txt

# Use specific format
john --format=raw-sha1 -w:pwdlist.txt hashes.txt
```

Further reading

- John the Ripper docs and `john.conf` (rules & incremental definitions).
- John Jumbo README for OpenCL/GPU and dynamic formats.

This cheat sheet is designed for professional, authorized use in penetration testing and forensic analysis.