Password Profiling

About me

- **Ex. Java Developer.**
- **Security Consultant.**
- **IDDO**
  - **First BMX trick sensor in World.**

Objectives

- You get better with time ?!
- What is an **efficient** wordlist.
- Wordlist generators.

Issues

- Blind Brute-force takes to much time ?
- What's the **best** wordlist ?
- Word **permutation** ?
    - What are they ?
    - How can i use them?
- What tool should i use ?

Efficiency

Check list

- The **D.R.Y** philosophy.
- What’s the **best method** of this case.
- What’s not the password.

Don’t repeat your self.

- Repeating the same word = **more time**.
- Separate wordlists by **length**.
- **Easier** remove duplicates.

Check list (cont)

- Information about the hash’s **owner**/’s.
- Do password restrictions apply?
- One or more hashes ?

The target

- Personal information.
- Interests.
- Native language.
- Blogs.
- Visited sites.

Password restrictions

- What’s not the password = less words = less time.
- The most efficient tool for that case.

Multiple hashes

- The source is important.
  - Source of words.
  - Target information.

Single hash

- Relevant information?
    - Blog, twitter, social media.
    - Contact info.
    - Personal information.

Use case.

15 MD5 hashes provided by the website!

Data Sample

- ID: 1
- Username:
- Password: 4c89d332b2fa5a1684dccbcafe881c07
- Nome:
- Instituição:
- Email:
- Telefone:

Step 1

- Identify the hashing algorithm.
- Identify the source.
    - Create wordlist **form the source**.
    - Create wordlist from **user’s info**.

- Tools and resources.
  - Storage space and speed.
    - Is SSD + rainbow tables a possibility ?
    - Can you use your GPU.
    - What google says about the hash ?

Tools (cont)

- Hashcat.
- oclHashcat.
- John the Ripper password cracker.
  - Support for OpenCL and GPU.

Hashcat / oclHashcat

- Advantages.
    - Brute-force masks.
    - GPU/Multicore.
    - Word permutation.

Efficient Brute-force

- L/U - 2 - 8 : 1h.
- D - 2 - 9: 1.
- L/U and D 2 - 7: 1m.
- L/U and D 8: 31m.

- l - lower case.
- U - Upper Case.
- D - Numbers.

Efficient Brute-force (cont)

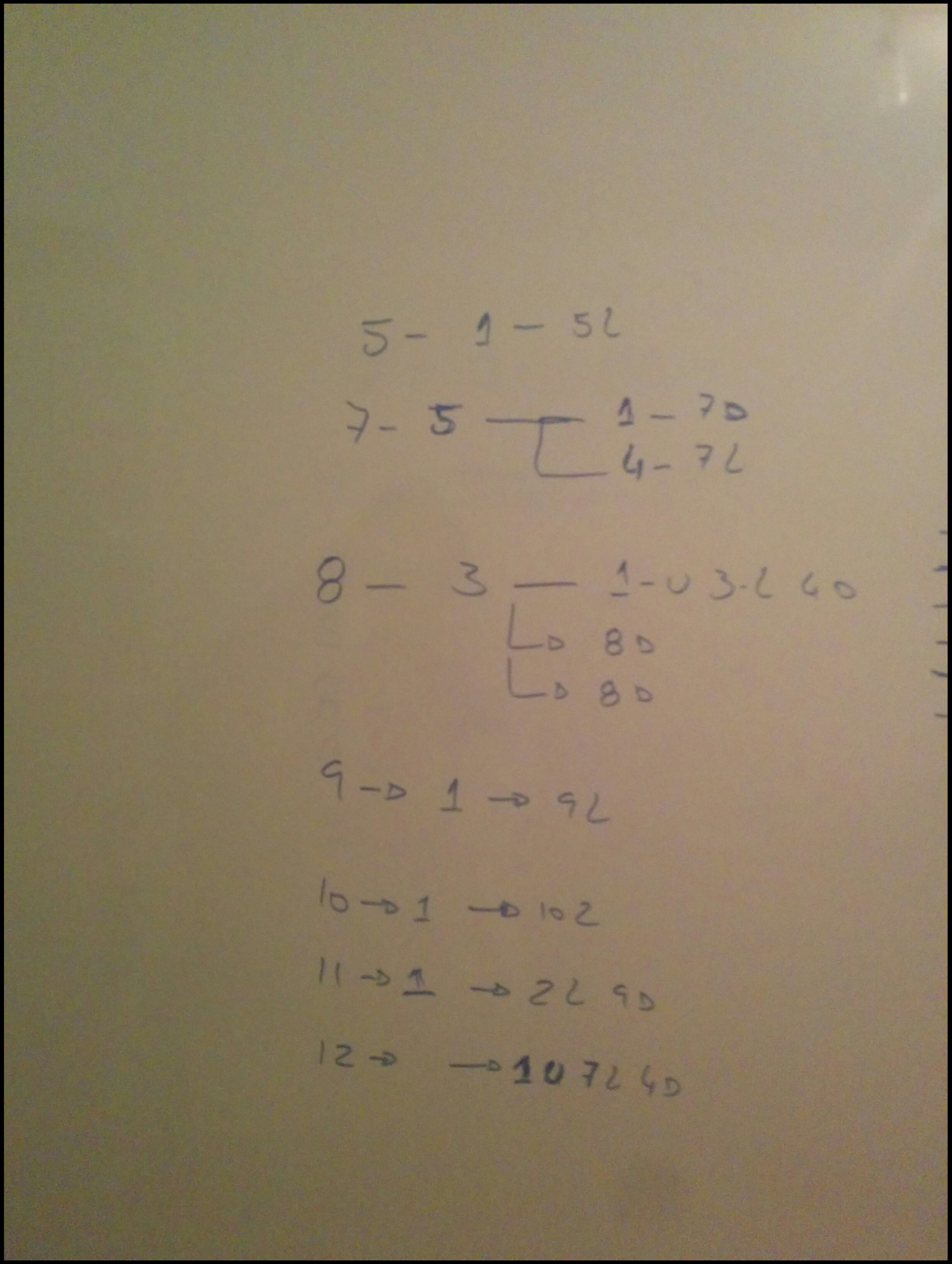
- Reduce number of words
  - Eg?U?L?L?L?L?d?d
  - Combine Dictionaries with masks.

Step 2

Result sample

- 5375907
- arquivo
- teclado
- sepanas
- fcporto
- 14947531
- 15304560
- Arli3266

Profile board.



5- 1 - 5L

7- 5 — 1 - 7D
└ 4 - 7L

8 — 3 — 1-U 3-L 4D
└→ 8D
└→ 8D

9 → 1 → 9L

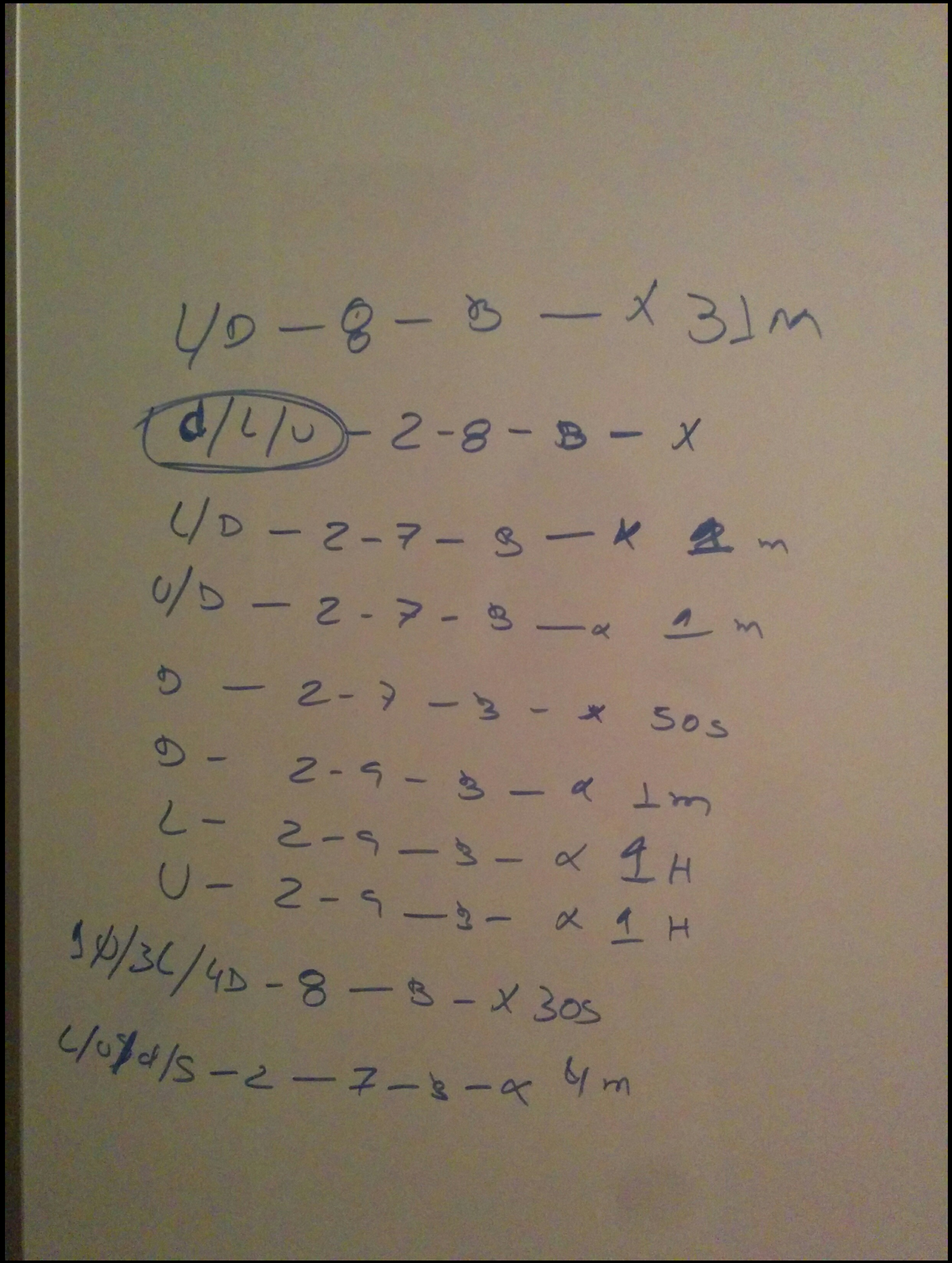
10 → 1 → 10L

11 → 1 → 2L 9D

12 → → 1U 7L 4D

Step 3

What’s not The password



4D – 8 – 3 – x 31m

d/L/U – 2-8-3 – x

L/D – 2-7 – 3 – x 1m

U/D – 2-7-3 – α 1m

D – 2-7 – 3 – x 50s

D – 2-9 – 3 – α 1m

L – 2-9 – 3 – α 1H

U – 2-9 – 3 – α 1H

1D/3L/4D - 8 – 3 – x 30s

L/U/d/S – 2 – 7 – 3 – α 4m

Most used

- Pattern identification.
- Collect external wordlists.
  - Remove what’s not the password!

Step 4

If everything fails, get a coffee and relax.

Word Permutation

- What is?
- Tools.
- How does it work.

Word Permutation (cont)

- Hashcat-Tools.
  - Word Generation tools
    - maskprocessor - Wordlist Generation by mask.
    - statsprocessor - Wordlist Generation by mask with markov-attack.
  - Word list processing.
    - combinator - Combines 2 wordlists in to one.
    - hcstatgen - Statistics file generator for markov-attack.
    - len - Filter wordlists by length.
    - permute - Word permutations.
    - req - Filter wordlists by rule eg: all words that include numbers.
    - rli - Compare wordlists are remove duplicates.
    - splitlen - Split Wordlist by length.

Word permutation (cont)

- Candidates.
    - Cracked passwords.
    - Used wordlists.

Results 13/15

- Ilnnnnnnnnn
- Ulllllllnnnn

User information (cont)

- Web Footprint.
  - Social media.
  - Interests.

15/15
Game over

Links

- http://hashcat.net/wiki/doku.php?id=hashcat_utils#permute

- http://blog.thireus.com/cracking-story-how-i-cracked-over-122-million-sha1-and-md5-hashed-passwords

- https://www.question-defense.com/2010/08/15/automated-password-cracking-use-oclhashcat-to-launch-a-fingerprint-attack