

Homework

Three tasks to deepen your understanding of dictionary attacks, rule writing, and combinator strategies.

Complete [Lab 2](#) before starting the homework. These tasks build on the skills and files from the lab.

Task 1 — Attack Mode Benchmark

Objective

Benchmark three attack approaches against a set of hashes and compare their effectiveness and speed.

Setup

Create a file `hw_hashes.txt` with the following MD5 hashes:

```
5f4dcc3b5aa765d61d8327deb882cf99
e10adc3949ba59abbe56e057f20f883e
d8578edf8458ce06fbc5bb76a58c5ca4
0d107d09f5bbe40cade3de5c71e9e9b7
5baa61e4c9b93f3f0682250b6cf8331b7ee68fd8
```

The last hash is SHA-1 (`-m 100`). The others are MD5 (`-m 0`). Run each attack type separately for MD5 hashes first.

Instructions

- Plain dictionary:** Run `hashcat -m 0 -a 0 --potfile-disable hw_hashes.txt rockyou.txt`. Record: time, speed, hashes cracked.
- Dictionary + rules:** Run with `-r rules/best64.rule`. Record the same metrics.

3. **Combinator:** Split `rockyou.txt` into two halves (first 7M lines and last 7M lines) and run a combinator attack. Record the metrics. Note: this generates a very large candidate space — you may want to use smaller subsets.

Deliverable

A comparison table:

Attack Mode	Candidates	Time	Speed (H/s)	Hashes Cracked
Dictionary (<code>-a 0</code>)	(your value)	(your value)	(your value)	(your value)
Dictionary + best64.rule	(your value)	(your value)	(your value)	(your value)
Combinator (<code>-a 1</code>)	(your value)	(your value)	(your value)	(your value)

Write a short paragraph (100-150 words) explaining which approach you would use first in a real audit and why.

Task 2 — Custom Wordlist Creation

Objective

Create a targeted custom wordlist for a fictional company and test it against generated hashes.

Instructions

1. Choose a fictional company name (e.g., "SolarTech") and create a wordlist of 50+ entries that an employee might use as a password. Include:
 - Company name variations (solartech, SolarTech, SOLARTECH, solar, tech)
 - Product names (make up 2-3)
 - City where the company is "based"

- Common suffixes: numbers (1-99, 2024, 2025, 2026), special characters (!, @, #)
- Seasonal patterns (Summer2025, Winter2026)

2. Generate MD5 hashes of 5 passwords from your wordlist:

```
echo -n "SolarTech2025" | md5sum  
echo -n "solartech!" | md5sum  
# ... etc.
```

3. Run a dictionary attack using your custom wordlist against those hashes.

4. Run the same hashes against `rockyou.txt`. How many does rockyou crack?

Deliverable

Your custom wordlist file, the generated hashes, and a written comparison: how does a targeted wordlist perform vs. a generic one for organization-specific passwords?

Tip: In real-world audits, custom wordlists combined with rules are often more effective than massive generic wordlists for cracking organization-specific passwords.

Task 3 — Rule Syntax Deep Dive

Objective

Learn hashcat's rule syntax by writing and testing 5 custom rules.

Instructions

1. Read the [hashcat rule syntax documentation](#).
2. Create a file called `custom5.rule` containing exactly 5 rules. Each rule must use a different operation. Suggested operations:
 - Capitalize first letter + append a year (`c $2 $0 $2 $6`)
 - Leet-speak substitutions (`sa@ se3 sil so0`)

- Reverse + uppercase all (`r u`)
- Duplicate the word + append "!" (`d $!`)
- Truncate to first 6 characters + append "123" (`]]]]] $1 $2 $3` — note: `]` removes the last character; adjust count based on word length, or use `'6` to truncate at position 6)

3. Test your rules by running:

```
# Preview what rules generate (without cracking)
echo "password" | hashcat -r custom5.rule --stdout
```

4. Document each rule: explain what it does and give an example of the input/output transformation.

Deliverable

The `custom5.rule` file and a table documenting each rule:

#	Rule Code	Description	Input	Output
1	<code>c \$2 \$0 \$2</code> <code>\$6</code>	Capitalize + append "2026"	password	Password2026
2	(your rule)	(your description)	(input)	(output)
3	(your rule)	(your description)	(input)	(output)
4	(your rule)	(your description)	(input)	(output)
5	(your rule)	(your description)	(input)	(output)

Submission Checklist

- Task 1: Benchmark table with 3 attack modes + written analysis
- Task 2: Custom wordlist (50+ entries) + hash comparison results + written comparison
- Task 3: `custom5.rule` file created + documentation table with 5 rules + tested with `--stdout`

- All commands ran successfully and results are documented