

Lab Guide

Five hands-on exercises covering mask attacks, incremental mode, custom charsets, and hybrid attacks.

Prerequisites

- Hashcat installed and working (verify with `hashcat --version`)
- A working directory for this lab (e.g., `~/hashcat-labs/lab06`)
- A small wordlist (e.g., `rockyou.txt` or a custom test wordlist)

Exercise 1 — Create Test Hashes

Goal

Generate MD5 hashes of known passwords so you can verify that your attacks are working correctly.

Step 1 — Generate Hashes

Create MD5 hashes for the following test passwords using `echo -n` and `md5sum` (or `md5` on macOS):

```
# Linux
echo -n "1234" | md5sum | awk '{print $1}' >> hashes.txt
echo -n "5678" | md5sum | awk '{print $1}' >> hashes.txt
echo -n "Alex42" | md5sum | awk '{print $1}' >> hashes.txt
echo -n "John99" | md5sum | awk '{print $1}' >> hashes.txt
echo -n "pass" | md5sum | awk '{print $1}' >> hashes.txt
echo -n "password123" | md5sum | awk '{print $1}' >> hashes.txt

# macOS
echo -n "1234" | md5 >> hashes.txt
echo -n "5678" | md5 >> hashes.txt
echo -n "Alex42" | md5 >> hashes.txt
echo -n "John99" | md5 >> hashes.txt
```

```
echo -n "pass" | md5 >> hashes.txt
echo -n "password123" | md5 >> hashes.txt
```

Step 2 — Verify

Check that `hashes.txt` contains 6 lines, each a 32-character hex string:

```
cat hashes.txt
wc -l hashes.txt
```

Tip: Use `echo -n` (no trailing newline) to get the correct hash. Without `-n`, the newline character is included in the hash input, producing a different result.

Exercise 2 — 4-Digit PIN Mask Attack

Goal

Crack the 4-digit PIN hashes ("1234" and "5678") using a simple digit mask.

Step 1 — Run the Attack

```
hashcat -m 0 -a 3 -o cracked_pins.txt hashes.txt ?d?d?d?d
```

Step 2 — Examine Results

```
cat cracked_pins.txt
```

You should see the hashes for "1234" and "5678" with their cracked plaintext values.

Step 3 — Review the Output

Note in the Hashcat output:

- **Keyspace:** should be 10,000 (10^4)
- **Speed:** how many hashes per second your hardware achieves
- **Recovered:** at least 2 out of 6 hashes should be cracked

Note: If Hashcat reports that some hashes were already found in a potfile, add `--potfile-disable` to force re-cracking, or remove the potfile with `rm ~/.local/share/hashcat/hashcat.potfile`.

Exercise 3 — Complex Mask Attack

Goal

Crack the "Alex42" and "John99" hashes using the pattern: uppercase + 3 lowercase + 2 digits.

Step 1 — Run the Attack

```
hashcat -m 0 -a 3 -o cracked_names.txt hashes.txt ?u?l?l?l?d?d
```

Step 2 — Examine Results

```
cat cracked_names.txt
```

Both "Alex42" and "John99" should appear in the results.

Step 3 — Calculate the Keyspace

Verify the keyspace manually:

- Position 1: `?u` = 26 choices
- Positions 2-4: `?l` = 26 choices each
- Positions 5-6: `?d` = 10 choices each
- Total: $26 \times 26 \times 26 \times 26 \times 10 \times 10 = 45,697,600$

Compare this with the keyspace shown in the Hashcat status output.

Exercise 4 — Incremental Mode

Goal

Use incremental mode to crack the "pass" hash (4 characters) without knowing the exact length in advance.

Step 1 — Run the Incremental Attack

```
hashcat -m 0 -a 3 -i --increment-min 4 --increment-max 8 \  
-o cracked_inc.txt hashes.txt ?a?a?a?a?a?a?a
```

Step 2 — Monitor Progress

Press **s** during the attack to see the status. Observe:

- The current mask length being tested
- The progress percentage for the current length
- Estimated time remaining

Warning: This attack will take a very long time beyond length 5 with **?a**. After "pass" is found at length 4, press **q** to quit gracefully, or let it run if you have time. The point is to observe the incremental behavior.

Step 3 — Verify

```
cat cracked_inc.txt
```

The hash for "pass" should be cracked. It will be found during the length-4 pass.

Exercise 5 — Hybrid Attack

Goal

Crack "password123" using a hybrid attack that appends digits to dictionary words.

Step 1 — Prepare a Small Wordlist

If you don't have rockyou.txt, create a small test wordlist:

```
cat > wordlist.txt <<EOF
password
admin
summer
welcome
dragon
EOF
```

Step 2 — Run the Hybrid Attack (Mode 6)

```
hashcat -m 0 -a 6 -o cracked_hybrid.txt \
hashes.txt wordlist.txt ?d?d?d
```

Step 3 — Examine Results

```
cat cracked_hybrid.txt
```

"password123" should appear in the cracked results. Hashcat combined "password" from the wordlist with the mask `?d?d?d` which generated "123" among its candidates.

Step 4 — Try Prepend Mode (Mode 7)

For comparison, try a prepend attack:

```
hashcat -m 0 -a 7 -o cracked_hybrid7.txt \
hashes.txt ?d?d?d wordlist.txt
```

This generates candidates like "000password", "001password", etc. It won't crack "password123" (since the digits are at the end, not the beginning), but it demonstrates the mode 7 syntax.

Tip: In real-world scenarios, run both `-a 6` and `-a 7` since you often don't know whether the user appended or prepended characters.

Completion Checklist

- Ex 1: hashes.txt created with 6 MD5 hashes
- Ex 2: "1234" and "5678" cracked with `?d?d?d?d` mask
- Ex 3: "Alex42" and "John99" cracked with `?u?1?1?1?d?d` mask
- Ex 4: "pass" cracked using incremental mode
- Ex 5: "password123" cracked using hybrid mode 6