Lab 4: Passwords and Logins

Step 1: Create a lab user and set a password

Create the user:

- Command used:

```
❯ sudo useradd -m labuser
[sudo] password for markus:
```

Set the password:

- Command used:

```
❯ sudo passwd labuser
New password:
Retype new password:
passwd: password updated successfully
```

Verify account existence:

- `getent passwd labuser` output:

```
❯ getent passwd labuser
labuser:x:1001:1001::/home/labuser:/bin/sh
```

Step 2: Observe /etc/passwd and /etc/shadow

Deliverable 1:

- The `labuser` line from `/etc/passwd`:

```
❯ grep '^labuser:' /etc/passwd
labuser:x:1001:1001::/home/labuser:/bin/sh
```

Deliverable 2:

- The `labuser` line from `/etc/shadow` (Redacted):
  - Format: `username:$id$salt$...` (Keep the ID and Salt, replace the rest of the hash with `...`)
  - Line:

```
❯ sudo grep '^labuser:' /etc/shadow
labuser:$y$j9T$...
```

> After switching to `SHA512`

```
❯ sudo grep '^labuser:' /etc/shadow
labuser:$6$5AnXfOatz9PMlE/b$...
```

Identify the hash scheme:

- Hash scheme ID found : `$y$`, After switching `$6$`

Step 3: Extract the hash for cracking

Deliverable 3:
Contents of `hash.txt` (Paste the single line here):

```
❯ cat hash.txt
$y$j9T$nUeFtyViZWM42bei0l3jtO/$iyLznu6lMPKOq9isz1ZQQPD8OyU.bDGDD1uVJ7iXtn3
```

> After switching to `SHA512`

```
❯ cat hash.txt
$6$5AnXfOatz9PMlE/b$vUkU.RTDABLSKjlN3eLS0qrkaaA5wJTZFcp1cQmHzKlBctVPvXACAxV9
9Chs2GPi0.K3uCFRkZcOoBAxJPMMQ0
```

Step 4: Dictionary cracking with Hashcat

Setup:

- Wordlist used:

```
❯ curl
https://raw.githubusercontent.com/danielmiessler/SecLists/refs/heads/master/
Passwords/Common-Credentials/Pwdb_top-10000000.txt >> wordlist
```

- Hashcat mode selected:
  - Couldn't find a hashcat mode since the password is hashed using `yescrypt` which hashcat doesn't support.
    - Correction, seems like there is support but set up is quite in depth, beyond what I think is reasonable for this lab, some more info here
  - Had to go back and change the hash type for the password
  - Looks like it should be 1800, so going with that

```
❯ hashcat -h | grep "SHA512"
   ...
   7100 | macOS v10.8+ (PBKDF2-SHA512)    | Operating System
   1800 | sha512crypt $6$, SHA512 (Unix)  | Operating System
```

- Full command line used:

```
❯ hashcat -m 1800 -a 0 hash.txt wordlist
```

Results::

- Did it crack?
  - Yes
- Password recovered?
  - Yes, it discovered it was `password`.

Deliverable 4 (Short Paragraph):

- What does this demonstrate about common-password risk?
  - Since it common password, it's very likely that the million word list we used contains the password. This concept also applied for any other authentication system, not just linux, any password anywhere.
- What does it demonstrate about choosing passwords that humans can reproduce?
  - Humans aren't good at creating passwords, often using birthdays, related words etc. that're easily discover able by an attacker.

Step 5: Brute-force mask demo

Constraints:

- Search space: Digits only, length 4-6.

Command used:

```
❯ hashcat -m 1800 -a 3 hash.txt \?d\?d\?d\?d\?d\?d --increment
```

Deliverable 5 (One Sentence):

- Compare dictionary cracking vs. brute force:
  - Brute force took much longer, around 37 seconds, whereas dictionary was like 1 second.

Step 6: Cleanup

Lock and Remove Account:

- Commands used:

```
❯ sudo passwd -l labuser

❯ sudo userdel -r labuser
```

Deliverable 6 (Evidence):

- Run `getent passwd labuser`.
- Output (should be empty):