Network and System Security- Reflection - Week 03

Key Learning Outcomes

The Critical Importance of Password Security

The workshop transformed the understanding of password security by emphasising that password strength depends on the length and that passwords must never be stored in plaintext or with fast hashes. Through hands-on implementation, it became clear that secure password handling relies on slow, adaptive hashing algorithms like bcrypt, which are specifically designed to resist brute-force attacks.

Salt and Pepper: Defence Against Pre-computation Attacks

Using salts and peppers in password hashing is essential for defending against rainbow table and database attacks, as they ensure hash uniqueness and add extra layers of protection, demonstrating the importance of layered security design.

Two-Factor Authentication Implementation

Implementing TOTP-based two-factor authentication demonstrated how combining a password with a time-based code strengthens security, while also showing that TOTP is more secure than SMS-based 2FA because it avoids vulnerabilities like SIM swapping and interception.

Brute-Force Attack Simulation

Brute-force simulation showed how fast hashing algorithms like MD5 and SHA-256 are easily cracked, making them insecure for passwords, while bcrypt's deliberate slowness provides strong protection by making brute-force attacks impractical.

Technical Skills Developed

- Practical experience with Python's bcrypt library for secure password hashing
- Implementation of TOTP using pyotp for two-factor authentication
- Understanding of cryptographic concepts: entropy, salting, peppering, and adaptive hashing
- Socket programming knowledge from previous weeks now connects to authentication in client-server architectures

Challenges Overcome

Initially, I struggled to understand why bcrypt stores the salt alongside the hash if the salt is meant to be "secret." The workshop clarified that salts don't need to be secret—they just need to be unique per user to prevent rainbow table attacks. The actual secret in the system is the pepper, which remains outside the database.

Real-World Applications

This workshop has practical implications for any application I build. I now understand:

- Why major platforms enforce minimum password requirements
- How companies like Google implement their authenticator apps

- Why data breaches with "hashed passwords" can still be catastrophic if weak hashing was used
- The importance of the "work factor" in bcrypt—as computers get faster, we can increase computational cost to maintain security

Moving Forward

This workshop has given me the knowledge to implement secure authentication systems and critically evaluate the security posture of existing applications. I am now equipped to explore more advanced topics like OAuth, OpenID Connect, and biometric authentication. The hands-on experience has transformed abstract security principles into practical skills I can apply immediately in personal projects and future professional work.