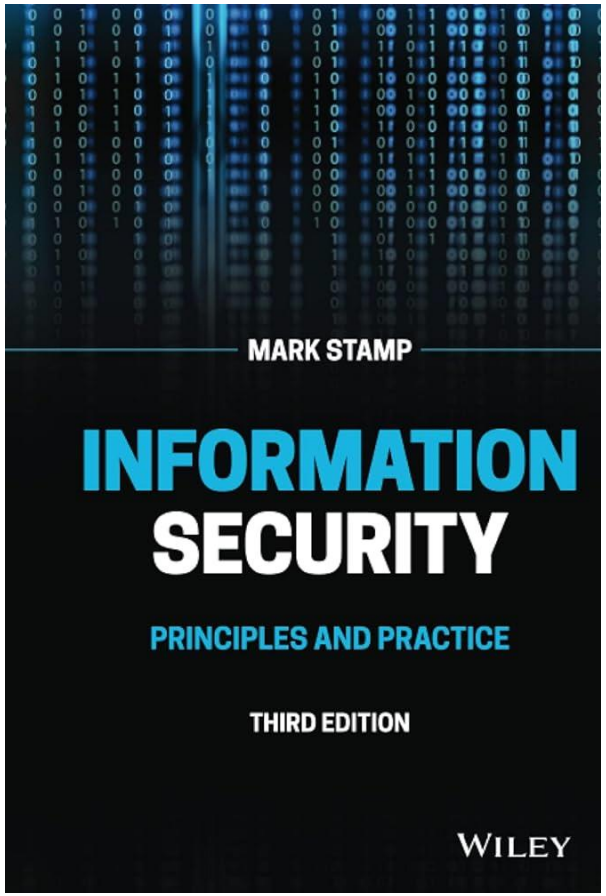
بسم الله الرحمن الرحیم

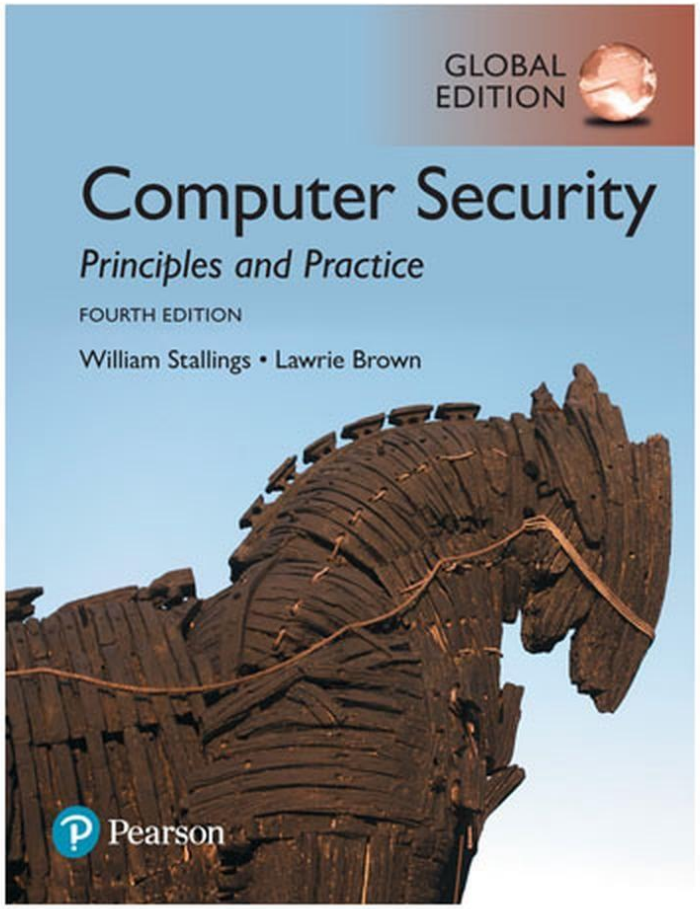
مبانی رایانش امن

جلسه ۱۸

مجتبی خلیلی
دانشکده برق و کامپیوتر
دانشگاه صنعتی اصفهان

- فصل ۳ استالینگ
- فصل ۷ استمپ





Access Control

Attacks on Passwords

- Attacker could…
  - Target one particular account
  - Target any account on system
  - Target any account on any system
  - Attempt denial of service (DoS) attack
- Common attack path
  - Outsider → normal user → administrator
  - May only require **one** weak password!

Password Retry

- Suppose system locks after 3 bad passwords. How long should it lock?
  - 5 seconds
  - 5 minutes
  - Until SA restores service
- What are +'s and -'s of each?

Password File?

- Bad idea to store passwords in a file
- But we need to verify passwords
- Solution? **Hash** passwords
  - Store y = h(password)
  - Can verify entered password by hashing
  - If Trudy obtains the password file, she does not (directly) obtain passwords
- But Trudy can try a *forward search*
  - Guess x and check whether y = h(x)

Dictionary Attack

- Trudy pre-computes h(x) for all x in a **dictionary** of common passwords
- Suppose Trudy gets access to password file containing hashed passwords
  - She only needs to compare hashes to her pre-computed dictionary
  - After one-time work of computing hashes in dictionary, actual attack is trivial
- Can we prevent this forward search attack? Or at least make it more difficult?

Salt

- Hash password with **salt**
- Choose random salt s and compute

  y = h(password, s)

  and store (s,y) in the password file
- Note that the salt s is not secret
  - Analogous to IV
- Still easy to verify salted password
- But lots more work for Trudy
  - Why?

Password Cracking: Do the Math

- Assumptions:
- Pwds are 8 chars, 128 choices per character
  - Then $128^8 = 2^{56}$ possible passwords
- There is a **password file** with 2^{10} pwds
- Attacker has **dictionary** of 2^{20} common pwds
- **Probability** 1/4 that password is in dictionary
- **Work** is measured by number of hashes

Password Cracking: Case I

- Attack 1 specific password ***without*** using a dictionary
  - E.g., administrator's password
  - Must try $2^{56}/2 = 2^{55}$ on average
  - Like exhaustive key search
- Does **salt** help in this case?

Password Cracking: Case II

- Attack 1 specific password ***with*** dictionary
- With **salt**
  - Expected work: $1/4\ (2^{19}) + 3/4\ (2^{55}) \approx 2^{54.6}$