Cracking di Password Hashate

Cos'è un hash?

- Funzione matematica unidirezionale
- Trasforma un input in una stringa fissa
- Non reversibile (non si può ottenere l’input originale)
- Usato per proteggere le password nei database e nei sistemi in genere

Esempi di hash comuni

- - MD5: veloce, ma debole (vulnerabile alle collisioni)
- - SHA-1: più sicuro di MD5, ma non più raccomandato
- - SHA-256: attualmente considerato sicuro
- - bcrypt, scrypt, Argon2: specifici per password, lenti e sicuri

Come si cracka un hash?

- - Dizionario: confronto hash con parole comuni
- - Forza bruta: prova tutte le combinazioni
- - Rainbow table: tabelle precalcolate di hash
- - Tool comuni: John the Ripper, Hashcat
- Wordlist:
  - Rockyou
  - https://github.com/danielmiessler/SecLists
  - https://github.com/brannondorsey/naive-hashcat/releases/download/data/rockyou.txt

Esempio Python

- - Si fornisce un hash MD5
- - Si usa un dizionario semplificato (mini_rockyou.txt)
- - Script Python confronta ogni parola con l'hash
- - Output: password trovata o non presente

Difese contro il cracking

- - Password complesse e uniche
- - Uso del salt: valore casuale aggiunto alla password
- - Hashing iterato: applicare la funzione hash più volte
- - Utilizzo di algoritmi specifici per password (es. bcrypt)

John the Ripper (esempio)

- 1. Preparare un file con gli hash
- 2. Usare il comando:
- john --wordlist=mini_rockyou.txt --format=Raw-MD5 hashes.txt
- 3. Verifica con:
- john --show hashes.txt

hashcat

- hashcat -m 0 -a 0 hashes.txt mini_rockyou.txt
- hashcat --show -m 0 hashes.txt