PASSWORD BREACH

In Association with GOLDMAN SACHS , upcoming solution for password security and cyber security

Solution By: Soham Agarwal

Problem Statement : Conventional methods of salting and peppering Security are susceptible to brute-force attacks when confronted with substantial computational resources leading to Security breach

Proposed Solution : Applying Peppering to hash Obtained from the security key

Basis : The Pepper used is supposed to be a character and there are 52 combinations possible

1. **Increased Complexity:** By generating 51 additional pepper-modified hashes for each actual hash, we create a scenario where an attacker must process 52 potential hashes for every single password attempt.
2. **Uncertainty Factor:** The inclusion of the actual hash among the pepper-modified versions introduces an element of uncertainty, further complicating the attacker's task.
3. **Resource Intensive:** Storing 52 hashes per password entry significantly increases the storage requirements, making large-scale attacks more resource-intensive and less feasible.

Strategy required for hacking by hashcat

1. Accessing the leaked database containing the hashes without pepper used.
2. Create 51 additional hashes by applying unique pepper values to the original hash.
3. Store all 52 hashes (51 peppered + 1 original) in the database.
4. Apply reduction function and then store all that 52 plaintexts

Security Benefits

This approach substantially increases the computational workload and storage requirements for potential attackers. The necessity to process multiple hashes for each password attempt, combined with the uncertainty of which hash is genuine, creates a formidable barrier against unauthorized access.

By implementing this enhanced peppering technique, we can significantly improve password security, making it exceptionally challenging for attackers to compromise user credentials, even in scenarios involving high computational power.

While when the user inputs the password : the server hashes it first and then check the (hash+52 possible peppers) and then if it verifies then user gets the entry.

In that way it makes it faster also than usual peppering , because it only calculates hash once.

Rather than 52 hash calculations