



Security

Coursework 2 - Group 16



System Overview




- SoccerDB is a website for user to search soccer events and save events
- There are two types of user: normal user and administrator
- The search soccer event function is not restricted to any type of user
- Normal user can bookmark soccer event and view it in future
- Administrator can view and delete normal user
- Security is important to protect admin page from unauthorised access
- This is because admin page is possibly destructive
- Security is also important to maintain attribute-based access control
- This is because the bookmarked events of each normal user is different




Vulnerability 1 - Password hashing

- Password is directly hashed without adding salt
- This opens to rainbow table attack
- Attacker can pre-compute password hashes and find for matches
- This also opens to reversed lookup table attack
- Attacker can run attack on multiple password at one time
- Solution: Add salt to password before hashing
- Reason: Different salt for different password make the attacking multiple passwords at once not effective



Vulnerability 1.2 – Password hashing (After salted)

- ▶ Attacker still can compute hashes at high speed
- ▶ So, dictionary attack and brute force attack is still effective
- ▶ Solution: Key stretching, i.e. PBKDF2
- ▶ Reason: Computation is added to slow down the password hashing process



Vulnerability 2 – Password rule

- Current password rule is any password with minimum length 1
- User can use real dictionary word as password
- This opens to dictionary attack
- User can also use short password
- This makes brute force attack more effective
- Solution: Enforce password rules, i.e.
 - Minimum 10 characters
 - Must have:
 - digit
 - lowercase alphabet
 - upper case alphabet
 - special character

Vulnerability 2 – Password rule (continued)

- The solution was added with a password strength checker
- The checker:
 - updates the adherence of password rule as user enter password
 - depicts the password entropy on a meter
- The formula for password entropy (E): $E = \log_2 R^L$, R : number of possible characters, L : length of password
- The R in this project is determined based on the following scheme:

Class	All possible characters	Number of possible characters
Digit	0-9	10
Lower case alphabet	a-z	26
Upper case alphabet	A-Z	26
Special character	!@#\$%^&*()_-=+={}[] \ ' " : ; ? / < > , . ~ `	32

Calculation of Password Entropy

- ▶ The formula for password entropy (E): $E = \log_2 R^L$, R : number of possible characters, L : length of password
- ▶ If all requirements are merely fulfilled, the number of possible characters, $R = 10 + 26 + 26 + 32 = 92$
- ▶ The length of password, $L = 10$
- ▶ Then, $E = \log_2 92^{10}$
- ▶ And, $E = \frac{\log_{10} 92^{10}}{\log_{10} 2}$
- ▶ $E = \frac{19.6379}{0.3010}$
- ▶ $E = 65.2422$



Simple User Notes



- The main page is home page
- From home page, all types of user can search soccer event, then click bookmark button
- Then, user will be prompted for login, only normal user is allowed to use the bookmark button
- A new normal user can be registered at register page
- Normal user can also navigate to bookmark page to view past bookmarked soccer events (This requires login)
- Admin can go to admin page to view the list of normal user information and remove any normal user (This requires login)

A decorative graphic on the left side of the slide. It features a solid orange arrow pointing to the right, positioned horizontally. Behind the arrow and extending downwards and to the right are several thin, curved green lines that create a sense of movement or flow.

Demonstration



The End

Thank you



User Manual



User manual

- The git hub link: <https://github.com/ChanKhaiShen/SoccerDB.git>
- Step1: Download .zip file
- Step 2: Open in Visual Studio Code
- Step 3: In terminal, run “npm install”
- Step 4: In terminal, run “npm run dev”
- Step 5: In browser: go to “http://localhost:5000/home.html”