### **1. Mã hóa Mật khẩu**

Mật khẩu là một trong những dữ liệu nhạy cảm nhất trên một trang web. Việc bảo mật mật khẩu yêu cầu mã hóa chúng khi lưu trữ trên máy chủ để tránh việc bị rò rỉ thông tin.

**Phương thức mã hóa mật khẩu:**

**Hashing (Băm)**: Đây là phương thức phổ biến nhất để mã hóa mật khẩu. Hashing là một quá trình biến đổi dữ liệu đầu vào (mật khẩu) thành một chuỗi ký tự cố định, không thể đảo ngược.

**Thuật toán phổ biến**: bcrypt, Argon2, PBKDF2

**Ưu điểm**: Không thể lấy lại mật khẩu ban đầu từ giá trị đã băm, phù hợp cho việc lưu trữ mật khẩu an toàn.

**Nhược điểm**: Nếu sử dụng thuật toán yếu hoặc không thêm salt (muối) vào băm, mật khẩu có thể bị tấn công dễ dàng.

**Salted Hashing**: Để tăng cường bảo mật, có thể sử dụng một chuỗi ngẫu nhiên (salt) kết hợp với mật khẩu trước khi băm. Salt giúp bảo vệ hệ thống khỏi các cuộc tấn công như brute force hay rainbow table.

**Ưu điểm**: Tăng cường độ an toàn.

**Nhược điểm**: Phức tạp hơn trong việc triển khai và quản lý.

### **2. Mã hóa Dữ liệu Truyền Tải (SSL/TLS)**

Khi người dùng nhập thông tin như mật khẩu, địa chỉ email hoặc thông tin thanh toán, việc mã hóa dữ liệu khi truyền qua mạng là rất quan trọng. SSL/TLS là các giao thức bảo mật giúp mã hóa dữ liệu giữa máy khách và máy chủ.

**SSL/TLS**: Sử dụng HTTPS để mã hóa tất cả dữ liệu giữa máy chủ và trình duyệt. HTTPS sử dụng SSL/TLS để bảo vệ việc trao đổi thông tin.

**Ưu điểm**: Bảo mật thông tin truyền tải giữa người dùng và máy chủ, bảo vệ dữ liệu khỏi bị nghe trộm hoặc thay đổi.

**Nhược điểm**: Yêu cầu chứng chỉ SSL, và đôi khi có thể làm tăng độ trễ.

**Cách sử dụng SSL/TLS trên máy chủ:**

1. Cài đặt chứng chỉ SSL cho tên miền của bạn.
2. Cấu hình máy chủ web để sử dụng HTTPS thay vì HTTP.
3. Kiểm tra kết nối HTTPS bằng cách mở trang web và đảm bảo có biểu tượng khóa trong thanh địa chỉ.

### 

### 

### 

### 

### **3. Mã hóa Dữ liệu Lưu trữ (Encryption)**

Ngoài việc mã hóa mật khẩu, đôi khi bạn cần mã hóa thông tin nhạy cảm khác (ví dụ: thẻ tín dụng, địa chỉ). Điều này đảm bảo rằng dữ liệu không thể được đọc ngay cả khi cơ sở dữ liệu bị tấn công.

**AES (Advanced Encryption Standard)**: Một trong các thuật toán mã hóa đối xứng phổ biến. Nó có thể mã hóa và giải mã dữ liệu với khóa bí mật.

**Ưu điểm**: Mã hóa mạnh mẽ và nhanh chóng.

**Nhược điểm**: Cần quản lý khóa bảo mật một cách cẩn thận.

### **4. Xác Thực Hai Yếu Tố (2FA)**

Xác thực hai yếu tố giúp tăng cường bảo mật tài khoản người dùng bằng cách yêu cầu hai phương thức xác thực. Một trong những phương thức phổ biến là kết hợp mật khẩu với mã OTP (One-Time Password).

**Ưu điểm**: Ngay cả khi mật khẩu bị lộ, tài khoản vẫn được bảo vệ nhờ vào yếu tố thứ hai (ví dụ: mã OTP gửi qua SMS hoặc ứng dụng Authenticator).

**Nhược điểm**: Cần cấu hình thêm và có thể gây bất tiện cho người dùng.

### **5. Bảo mật API (Token-based Authentication)**

Khi trang web có API, cần bảo vệ các API khỏi việc truy cập trái phép. Một trong các phương thức phổ biến là sử dụng JSON Web Token (JWT).

**JWT**: Là một mã thông báo mã hóa, chứa thông tin về người dùng và có thể xác thực mà không cần lưu trữ thông tin trên máy chủ.

**Ưu điểm**: Không cần lưu trữ thông tin trên máy chủ, giúp giảm tải hệ thống.

**Nhược điểm**: Nếu token bị lộ, người tấn công có thể truy cập vào hệ thống.