1. What is OWASP and what is its primary mission as described in the article?

   Open Web Application Security Project (OWASP) är en internationell organisation som jobbar med webb och applikations säkerhet. En av de principer är att allt dess material är fritt tillgängligt på dess webbplats, vilket gör det möjligt för alla att förbättra säkerheten för sin egen webbsidan eller applikation.

2. Explain the concept of "Injection". Provide an example of how an injection attack could compromise a web application's security.

   Injektionsattacker uppstår när en angripare skickar osäkra data till en kodtolkare via ett formulär eller annan användarinmatning i en webbplats. Ett exempel är när en angripare injicerar SQL-kod i ett formulär som förväntar sig ett användarnamn i vanlig text. Om formulär inte är ordentligt skyddad, kan SQL-koden köras, vilket kallas SQL-injektionsattack.Angriparen kan logga in utan att ha de rätta inloggningsuppgifterna och kan läsa, ändra eller radera känslig information i databasen.

3. Explain two strategies to prevent Broken Authentication vulnerabilities.

- Tvåfaktorsautetisering: det krävs både användarnamn/lösenord och en verifiering via SMS. När användare skapar ett konto måste han eller hon ange ett unikt användarnamn och ett lösenord. Det här ger den första autentiseringsfaktorn. Efteråt skickar tjänsten ett automatisk textmeddelande med en numerisk kod som SMS. Användares ombeds sedan att ange koden och om det är korret har användarens tillhandahållit en andra antentiseringsfaktor.

- Avancerad hastighetsbegränsning är ett system som styr och begränsar mängden begåvningar eller förfrågningar som en användare, IP-adres eller applikation kan göra till en server eller webbtjänst under en viss tidsperiod.

4. Briefly define Cross-Site Scripting (XSS) as outlined in the article and list two methods suggested in the article to prevent XSS attacks in web applications.

   XSS är en attack där en angripare injicerar skadlig kod på en webbplats eller webb applikation. När användaren besöker sidan kors koden i deras webbläsare och kan stjäla känslig information som cookies eller sessionstokens. Detta kan leda till obehörig åtkomst till användarkonton eller omdirigering till skadliga sidor.

   Två metoder för att förebygga kan vara:

   1. Validering att inmatning: utvecklare ka sätta upp reglerna som begränsar vilken typ av data användare kan skicka in. Till exempel kan ett formulär endast tillåta siffror eller vissa tecken för att undvika skadlig kod. De kan också blockera användningen av HTML i kommentarer eller inmatningar för att förhindra injektion av skadliga skript. Om det behövs stöd för rikt innehåll kan säkra alternativ som Markdown eller WYSIWYG-redigerare användas.

   2. Sanering av data:

   Innan data visas för andra användare kan utvecklare granska och filtrera bort skadlig kod, även om HTML tillåts. Utdata-kodning och escapning är också effektiva metoder där farligt innehåll omvandlas till ofarlig text, så att webbläsaren inte tolkar det som kod att köra.