COMP3052 Computer Security

Session 09: Internet Security

Videoclip: Digital Certificates Explained - How digital certificates bind owners to their public key

https://www.youtube.com/watch?v=5rT6fZUwhG8

1. What are HTTP cookies?

Answer:

HTTP cookies are small data files stored on a user's computer by the web browser while browsing a website. They're designed to hold modest amounts of data specific to a client and server, enabling the server to deliver a page tailored to a particular user or carry information from one visit to another.

Reference: Top 25 HTTP Cookies Interview Questions and Answers https://interviewprep.org/http-cookies-interview-questions/#:~:

2. How cookies work?

Answer:

Cookies work through a process initiated when a user visits a site. The server sends a Set-Cookie header with the response containing a unique ID. This cookie is stored in the user's browser and sent back to the server every time the user revisits the site. Cookies can be used for various purposes such as maintaining sessions, remembering user preferences, tracking user behavior, and implementing shopping cart functionality.

Reference: Top 25 HTTP Cookies Interview Questions and Answers https://interviewprep.org/http-cookies-interview-questions/#:~:

3. How does the Secure attribute in a cookie affect its transmission?

Answer:

The Secure attribute in a cookie ensures that the cookie is only sent over HTTPS, not HTTP. This means it’s transmitted only through an encrypted connection, preventing potential interception by unauthorized parties. If a website tries to send a secure cookie via an unsecured HTTP connection, the browser will block its transmission, enhancing user data protection.

Reference: Top 25 HTTP Cookies Interview Questions and Answers https://interviewprep.org/http-cookies-interview-questions/#:~:

4. Describe a situation where you would use a session cookie instead of a persistent cookie?

Answer:

A session cookie would be used in an online shopping scenario. When a user adds items to their cart, the server creates a unique session ID for that user and stores it as a session cookie on the user's browser. This allows the server to keep track of the user's shopping cart contents while they browse the site. The session cookie is deleted once the user closes their browser, ensuring that their shopping cart doesn't persist beyond their current visit. A persistent cookie wouldn't be suitable here because we don't want the shopping cart data to remain after the user has left the site or closed their browser.

Reference: Top 25 HTTP Cookies Interview Questions and Answers

https://interviewprep.org/http-cookies-interview-questions/#:~:

5. Discuss the differences between first-party and third-party cookies.

Answer:

First-party cookies are created by the website a user is visiting. They enable site functionality, including remembering login details and products in shopping carts. These cookies are generally considered safe as they do not track users across multiple sites.

Third-party cookies, on the other hand, are created by domains different from the one visited by the user. They're often used for online advertising and tracking purposes, enabling advertisers to target ads based on browsing history. However, due to privacy concerns, their use is controversial and being phased out by many browsers.

Reference: Top 25 HTTP Cookies Interview Questions and Answers

https://interviewprep.org/http-cookies-interview-questions/#:~:

6. How would you implement HTTP-only cookies in a web application?

Answer:

HTTP-only cookies can be implemented in a web application by setting the HttpOnly attribute in the Set-Cookie HTTP response header. This is done server-side, typically using a back-end language like PHP or Node.js.

In PHP, you would use the setcookie() function and set the seventh parameter to true. For example: setcookie('name', 'value', time()+3600, "/", "", false, true).

In Node.js with Express, you'd use res.cookie(), passing an options object with httpOnly set to true. Example: res.cookie('name', 'value', {httpOnly: true}).

This makes the cookie inaccessible via client-side scripting languages such as JavaScript, enhancing protection against cross-site scripting (XSS) attacks.

Reference: Top 25 HTTP Cookies Interview Questions and Answers

https://interviewprep.org/http-cookies-interview-questions/#:~:

7. What are some security risks associated with HTTP cookies? How would you mitigate these risks?

Answer:

HTTP cookies pose several security risks. The primary risk is unauthorized access to sensitive data, often through cross-site scripting (XSS) or cross-site request forgery (CSRF). XSS attacks exploit vulnerabilities in web applications to inject malicious scripts, while CSRF tricks the victim into submitting a malicious request.

To mitigate these risks, one should implement secure and HttpOnly flags. Secure flag ensures that cookies are sent over HTTPS, preventing interception during transmission. HttpOnly flag prevents client-side scripts from accessing cookies, mitigating XSS attacks.

Another mitigation strategy is implementing same-site attribute which restricts cookies to first-party contexts, reducing the risk of CSRF attacks. Additionally, setting short expiration times for cookies can limit the window of opportunity for an attack.

Lastly, proper validation and sanitization of inputs can prevent injection of malicious scripts. Regularly updating and patching systems also helps in keeping up with new threats.

Reference: Top 25 HTTP Cookies Interview Questions and Answers

https://interviewprep.org/http-cookies-interview-questions/#:~: