Feedback — week 3 quiz

Help

You submitted this quiz on **Thu 13 Nov 2014 11:21 AM PST**. You got a score of **34.00** out of **42.00**. You can attempt again, if you'd like.

Question 1

What is one difference between an HTTP GET and an HTTP POST request?

Your Answer		Score	Explanation
Only GET requests are subject to the same-origin policy			
Only POST requests may include parameter data in the request body	✔	2.00	POST requests are often issued for web forms whose content is included in the request data
Only GET requests use the REFERER header			
Only POST requests can encode parameters in the URL			

Total	2.00 / 2.00

Question 2

Which of the following is true about static and dynamic web content?

Your Answer		Score	Explanation
The server often produces dynamic content based on the contents of the database	✔	1.00	Dynamic content is regenerated with each request, often including database-resident content
Static content may be re-generated with each request			
Javascript programs are run server-side to produce dynamic content			
Static pages may include PHP programs, which execute at the browser			
Total		1.00 / 1.00	

Question 3

SQL injection exploits a bug in what interaction of a web application?

Your Answer		Score	Explanation
Client to server	✖	0.00	The client inputs data that the server passes on to the database, but SQL injection is not exploiting a bug involved in the client-server interaction
Server to client			
Server to database			
Network to server			
Total		0.00 / 2.00	

Question 4

SQL injection often allows an attacker to do which of the following?

Your Answer		Score	Explanation
Access information he shouldn't			
Overrun a buffer to smash the stack			
Cause memory to be used after it's freed			
All of the above	✖	0.00	Stack smashing and use-after-free bugs are not related to SQL injection, which is a bug in how a SQL command is constructed
Total		0.00 / 2.00	

Question 5

If you had to summarize the key programming failure with SQL injection, it would be:

Your Answer	Score	Explanation

Bypassing authentication			
Confusing data with code	✔	3.00	Data entered by an untrusted user is formatted so as to be interpreted as SQL code, which can be used to work around the application's purpose
Trusting without verifying			
Circumventing the same origin policy			
Total		3.00 / 3.00	

Question 6

What is *escaping* an example of?

Your Answer	Score	Explanation
Blacklisting		

Validation			
Whitelisting			
Sanitization	✔	2.00	Sanitization is a transformation of text that removes potentially harmful elements, and escaping does this when content could contain HTML markup
Total		2.00 / 2.00	

Question 7

Suppose a web application implements authentication by constructing an SQL query from HTML form data using PHP's *prepared statements*. What would happen if an attacker entered `FRANK' OR 1=1; --` in the web form's user field?

Your Answer		Score	Explanation
The text will modify the structure of the SQL query and possibly bypass authentication			
The application will try to authenticate a user whose name is `FRANK' OR 1=1; --`	✔	3.00	The text that is entered will be treated as data, and not confused as code

The text will corrupt the query structure and the database will view it as a syntax error

The text will be confused as the password and authentication will probably fail

Total	3.00 / 3.00

Question 8

Why is it undesirable to implement session identifiers using (only) hidden form fields?

Your Answer		Score	Explanation
Such fields cannot contain binary data			
Such fields cannot include timeout information			
The session ID is forgotten when the browser window is closed	✔	2.00	This adds inconvenience to the user, since closing the window necessitates logging in again, and complicates the construction of the site to always pass around the hidden field

These fields are easily modified by the user

Total	2.00 / 2.00

Question 9

Suppose a browser submits a GET request to URL `http://www.mybank.com/accountinfo` on 20 February 2015. Which of the following cookies, if already stored at the browser, would be sent with the request?

Your Answer		Score	Explanation
`lang=us-english; expires=Sat, 1-Aug-2015; path=/accountinfo/; domain=.fidelity.com`	✔	1.00	This cookie is not sent because the domain name does not match
`prefs=small:blue:refresh; expires=Sat, 1-Aug-2015; path=/specialoffers/; domain=.mybank.com`	✔	1.00	This cookie is not sent because the path is not a prefix of the path given in the URL
`edition=us; expires=Wed, 18-Feb-2015; path=/; domain=.mybank.com`	✖	0.00	This cookie is not sent because it has expired

☐ `sessid=ABCDEFG; expires=Sat, 21-Feb-2015; path=/; domain=.mybank.com`	✖	0.00	This cookie has not timed out, has a path that is a prefix of the given path, and references the proper domain suffix
Total		2.00 / 4.00	

Question 10

Which of the following are ways that session cookies could be stolen or forged?

Your Answer		Score	Explanation
☑ Compromising the browser or server	✔	1.00	Injected code could exfiltrate cookies used for all users/sites
☑ Predicting the cookie's structure and reconstructing it	✔	1.00	Knowing how cookies are constructed, and knowing features of the user, site, etc. permits creation of the cookie
☐ Copying them by keylogging	✔	1.00	Session IDs are not entered by the user, but determined by the site
☑ Reading it from an unencrypted web request	✔	1.00	An adversary that can see web requests (e.g., in an Internet cafe) can steal cookies from those requests
Total		4.00 /	

4.00

Question 11

Which of the following are ways to reduce the impact of a stolen cookies?

Your Answer		Score	Explanation
☑ Changing a user's cookie from session to session	✔	1.00	Per-session cookies, if stolen, cannot affect future sessions
☐ Associate the cookie with the server's IP address	✔	1.00	Associating a cookie with the client's IP address can help, despite false positives and false negatives, but associating it with the server's address would not
☐ Prevent cookies from entering the DNS cache	✔	1.00	Cookies don't go in the DNS cache, DNS addresses do
☑ Giving each cookie a timeout	✔	1.00	Timing out a cookie means it can only be misused for a limited period
Total		4.00 / 4.00	

Question 12

How can the referer field be used to defend against CSRF attacks?

Your Answer		Score	Explanation
It enforces that sensitive requests are (only) initiated by interaction with a site's own pages			
It ensures that requests only come from authenticated users	✖	0.00	HTTP has nothing to do with user-level authentication (HTTPS authenticates a client with a server, but that has no impact on CSRF)
It can't be used reliably because it only works for dynamic content			
It can be used to check that a Javascript program is from the proper origin			
Total		0.00 / 2.00	

Question 13

`<script></script>` tags in HTML pages most often identify programs written in what language?

Your Answer		Score	Explanation
Java			
C			
Javascript	✔	2.00	
PHP			
Total		2.00 / 2.00	

Question 14

The browser implements security for Javascript programs for what reason?

Your Answer	Score	Explanation
It doesn't -- these programs run at the server so the browser can ignore them		

Such programs may access browser-controlled resources, which include potentially sensitive data in HTML documents and cookies	✔	3.00
It doesn't -- these programs are only used to render dynamic content but are otherwise not security-relevant		
Such programs could deny service by running forever		
Total		3.00 / 3.00

Question 15

XSS subverts what policy?

Your Answer		Score	Explanation
Secure defaults			
Availability			
Whitelisting			
Same Origin	✔	3.00	XSS uploads a script from host A to site B, which serves the script with its privileges, thus violating the same origin policy

Total	3.00 / 3.00

Question 16

What is the difference between stored (or persistent) XSS and reflected XSS?

Your Answer		Score	Explanation
Stored XSS works by injecting code in a site's served content, while reflected XSS injects code in a URL	✔	3.00	
Stored XSS is amenable to blacklisting but reflected XSS is not			
Stored XSS embeds Javascript in an a URL, while reflected XSS embeds it in a mirrored site			
Stored XSS works on database queries while reflected XSS works on cookies, which are received from and reflected back to the server			
Total		3.00 / 3.00	