**How do you write secure code? In your opinion, is it one of the developer's duties, or does it require a specialized role in the company? And why?**

*Secure code is such that has little or no vulnerabilities that can be exploited by malicious users.*

*Obfuscated code is code that is intentionally made hard to read.*

*Obscure code is code that is unintentionally hard to read and also probably insecure.*

*User followting* [*http://www.javascriptobfuscator.com/default.aspx*](http://www.javascriptobfuscator.com/default.aspx)

**Why is it said that cryptography is not something you should try to invent or design yourself?**

*You can roll your own, but you probably will make a major security mistake if you are not an expert in security/cryptography or have had your scheme analyzed by multiple experts. I'm more willing to bet on an open-source publicly known encryption scheme that's out there for all to see and analyze. More eyes means more likely that the current version doesn't have major vulnerabilities, as opposed to something developed in-house by non-experts.*

*From Phil Zimmermann's (PGP creator) Introduction to Cryptography (Page 54):*

*When I was in college in the early 70s, I devised what I believed was a brilliant encryption scheme. A simple pseudorandom number stream was added to the plaintext stream to create ciphertext. This would seemingly thwart any frequency analysis of the ciphertext, and would be uncrackable even to the most resourceful government intelligence agencies. I felt so smug about my achievement.*

*Years later, I discovered this same scheme in several introductory cryptography texts and tutorial papers. How nice. Other cryptographers had thought of the same scheme. Unfortunately, the scheme was presented as a simple homework assignment on how to use elementary cryptanalytic techniques to trivially crack it. So much for my brilliant scheme.*

*From this humbling experience I learned how easy it is to fall into a false sense of security when devising an encryption algorithm. Most people don’t realize how fiendishly difficult it is to devise an encryption algorithm that can withstand a prolonged and determined attack by a resourceful opponent.*

*Read Continue* [*https://security.stackexchange.com/questions/18197/why-shouldnt-we-roll-our-own*](https://security.stackexchange.com/questions/18197/why-shouldnt-we-roll-our-own)

**What is two factor authentication? How would you implement it in an existing web application?**

*Seems like a valid mechanism. But what if the SMS device is not in a service area? Or dead battery?*

*Talking about email or sms.*

**If not carefully handled, there is always a risk of logs containing sensitive information, such as passwords. How would you deal with this?**

*There are several best practices that you should consider when dealing with sensitive data.*

*First of all, minimize all of the sensitive data that you transfer at all. By hashing the password as early as possible in the architecture, you can ensure that any downstream components never even run the risk of exposing sensitive data.*

*Second, which is associated to the prior point, you should constrain/isolate the components that ever receive this sensitive information. For a password, only the authentication server would ever need this information - the rest of the components should really only deal with an authorization token (or something of the same ilk.) With regards to a credit card, this should likely only ever be associated with those services that would ever need it - for example the checkout/billing component of your system.*

*Third, for those systems that actually do need to get the actual value of the sensitive information for logic (e.g. credit card), you should consider creating/using a token-like system. This would basically be a place where you can store the sensitive information and provide an ID reference that replaces the original sensitive data element (e.g. exchange a credit card number for a credit card number id).*

*To close out the solution, exchange the sensitive data as close to the edge as possible (e.g. where you captured the credit card) in exchange for this token. Exchange this token for the sensitive information value (with the right authorization) to get the value back and then use it at the very latest possible moment. Finally, audit the logging that is done at those sensitive points to make sure that it's not logged in the clear.*

*This design pattern how some of the biggest e-commerce sites in the world do it. (Sorry, I am not allowed to disclose whom.) Hope it helps!*

[*https://stackoverflow.com/questions/33671027/logging-security-considerations-and-sensitive-data*](https://stackoverflow.com/questions/33671027/logging-security-considerations-and-sensitive-data)

**Write down a snippet of code affected by SQL Injection and fix it.**

*Write simple sql query for login without escaping data etc.*

**How would it be possible to detect SQL Injection via static code analysis? I don't expect you to write an algorithm capable of doing this, as it is probably a huge topic, but let's discuss a general approach.**

*Source Code Analysis Tools*

*The OWASP Foundation came online on December 1st, 2001 it was established as a not-for-profit charitable organization in the United States on April 21, 2004, to ensure the ongoing availability and support for our work at OWASP. OWASP is an international organization and the OWASP Foundation supports OWASP efforts around the world. OWASP is an open community dedicated to enabling organizations to conceive, develop, acquire, operate, and maintain applications that can be trusted. All of the OWASP tools, documents, forums, and chapters are free and open to anyone interested in improving application security. We advocate approaching application security as a people, process, and technology problem because the most effective approaches to application security include improvements in all of these areas. We can be found at* [*www.owasp.org*](http://www.owasp.org/)*.*

[*https://www.owasp.org/index.php/About\_The\_Open\_Web\_Application\_Security\_Project*](https://www.owasp.org/index.php/About_The_Open_Web_Application_Security_Project)

**What do you know about Cross-Site Scripting? If you don't remember it, let's review online its definition and let's discuss about it.**

*Cross-Site Scripting (XSS) attacks are a type of injection, in which malicious scripts are injected into otherwise benign and trusted websites. XSS attacks occur when an attacker uses a web application to send malicious code, generally in the form of a browser side script, to a different end user. Flaws that allow these attacks to succeed are quite widespread and occur anywhere a web application uses input from a user within the output it generates without validating or encoding it.*

*An attacker can use XSS to send a malicious script to an unsuspecting user. The end user’s browser has no way to know that the script should not be trusted, and will execute the script. Because it thinks the script came from a trusted source, the malicious script can access any cookies, session tokens, or other sensitive information retained by the browser and used with that site. These scripts can even rewrite the content of the HTML page. For more details on the different types of XSS flaws, see: Types of Cross-Site Scripting.*

[*https://www.owasp.org/index.php/Cross-site\_Scripting\_(XSS*](https://www.owasp.org/index.php/Cross-site_Scripting_(XSS)*)*

**What do you know about Cross-Site Forgery Attack? If you don't remember it, let's review online its definition and let's discuss about it.**

**How does HTTPS work?**

*Cross-Site Request Forgery (CSRF) is an attack that forces an end user to execute unwanted actions on a web application in which they're currently authenticated. CSRF attacks specifically target state-changing requests, not theft of data, since the attacker has no way to see the response to the forged request.*

*https://www.owasp.org/index.php/Cross-Site\_Request\_Forgery\_(CSRF)*

**What's a Man-in-the-middle Attack, and why does HTTPS help protect against it?**

**How can you prevent the user's session from being stolen? Chances are you remember what Session or Cookie Hijacking is, otherwise let's read its Wikipedia page together.**

*In cryptography and computer security, a man-in-the-middle attack is an attack where the attacker secretly relays and possibly alters the communication between two parties who believe they are directly communicating with each other. Wikipedia*

*https://us.norton.com/internetsecurity-wifi-what-is-a-man-in-the-middle-attack.html*