**Interview Practice Project**

**What is the most influential book or blog post you’ve read regarding web development?**

One of the most influential book I read was ‘Head First Design Patterns’. As I started reading and generating interest in this book, I took a course Programming Design Patterns as a part of my masters. I learned and implemented various patterns. Some of them are Model View Controller, Singleton, Factory Pattern, Builder pattern, Decorator Pattern, Observer Pattern, Command Pattern, Proxy Pattern, State Pattern, etcetera.

The most useful pattern in the web development I found was MVC since most of the frameworks today like AngularJS, KnockoutJS, Django, et cetera are based on this pattern. I am currently and in future will use these patterns for implementation of scalable and maintainable software applications.

**Tell me about a web application you have built. Why did you choose to build it? What did you learn? What challenges did you face and how did you overcome them?**

I built a RESTful web application for an online store that provides a list of items within a variety of categories and provides registered users the ability to post, edit and delete their own items from the catalog. I chose to develop this full stack application to learn all the features of the current web development needs.

The application’s backend is built using Python’s Flask framework. While for database, I have used SQLAlchemy. The front-end uses HTML, CSS, and JavaScript using KnockoutJS framework, jQuery, and Bootstrap. I created the Linux web server using Oracle’s VirtualBox and vagrant. I have also created the JSON endpoints that serves information from the catalog.

I learned to implement the user registration and authentication system using Google oAuth. The important pages are secured and need user authentication before using them. Implementing Google oAuth was a bit complicated process but I referred to the Google’s documentation and Stackoverflow posts to implement it. I also made sure that user can only update and delete the data he/she has created. This is achieved by implementing local user authorization.

Finally, I configured a Linux server from Amazon Web Service into a secure and efficient web application host to deploy this application as a WSGI application. Securing the web server was one of the challenging task. I ensured security by configuring the Uncomplicated Firewall to allow only SSH, HTTP and NTP ports. I hosted SSH on a non-default port and enforced key-based SSH authentication to make the server more secure.

**Write a function in Python that takes a list of strings and returns a single string that is an HTML unordered list (<ul>...</ul>) of those strings. You should include a brief explanation of your code. Then, what would you have to consider if the original list was provided by user input?**

Code:

def createHTMLList(listOfStrings):

 html = "<ul>"

 for x in listOfStrings:

   html += "<li>" + x + "</li>"

 html += "</ul>"

 return html

def main():

 listOfStrings = ["Bread", "Milk", "Cereals", "Honey", "Coffee"]

 print createHTMLList(listOfStrings)

main()

Here, the function createHTMLList() is generating a HTML list by attaching the required HTML tags (<ul> <li>...</li> </ul>).

The above program considers the list provided by a trusted source to the function is a valid list of strings. But, if the list is to be provided as the user input, that is from an untrusted source, it can attack and ruin the HTML to be generated. Thus to secure the code, we need to sanitize the text from the untrusted sources. To be precise, we need to strip the markups and attributes.

There is a library - Bleach, which intends to sanitize the text.

The above code can be modified as below.

import bleach

...

html += "<li>" + bleach.clean(x) + "</li>"

**List 2-3 attacks that web applications are vulnerable to. How do these attacks work? How can we prevent those attacks?**

*SQL Injection Attack:*

The attacker sends SQL queries from the html input elements instead of plain text. For example, the attacker can send this to delete all the posts from the posts table.

‘); delete from posts; --

Prevention

Never use string concatenation (+) and string parameter interpolation (%) to pass variables to SQL queries.

In Python, make sure to use query parameters instead of string substitutions

*Script Injection Attack:*

The attacker can send javascript code from a html form input field instead of plain text which is stored into the database. When the application requests this data from the database, it seems to be a JavaScript code to the browser and thus it executes it and causes problems.

For example, attacker can submit the text as below.

<script>  
setTimeout(function () {  
    var tt = document.getElementById('content');  
    tt.value = "<h2 style='color: #FF6699; font-family: Comic Sans MS'>Spam, spam, spam, spam,<br>Wonderful spam, glorious spam!</h2>";  
    tt.form.submit();  
}, 2500);  
</script>

Prevention

The user input must be sanitized before storing it to the database. In Python, there is a library - Bleach, which intends to sanitize the text by escaping or stripping the markups and attributes.

*CSRF attack:*

CSRF is an attack that tricks the victim into submitting a malicious request. It inherits the identity and privileges of the victim to perform an undesired function on the victim's behalf. CSRF attacks target functionality that causes a state change on the server, such as changing the victim's email address or password, or purchasing something.

Prevention

Anti-forgery state tokens protect the security of the users by preventing anti-forgery request attacks. The unique session token is created and sent with the authorization code and later it is verified with the server. This can ensure that the valid user is making the request.

**Here is some starter code for a Flask Web Application. Expand on that and include a route that simulates rolling two dice and returns the result in JSON. You should include a brief explanation of your code.**

Python file is attached.

The program generates random numbers simulating two dice rolls. The response sent is a JSON object containing the dice roll values. Some important parts of the program are as follows.

str(randint(1,6))

The function randint() takes a range of numbers to generate a random value while the function srt() converts the number to string data type.

jsonify()

This function transforms the object to a JSON object.

**If you were to start your full-stack developer position today, what would be your goals a year from now?**

Well I’m excited by this position at Google because in few years, I’d like to work hard and be seen as someone with deep expertise in the web development sector, and I know that’s something that I’ll have an opportunity to do here. I’m also really excited to take on more managerial responsibilities in the next few years and potentially even take the lead on some projects. I’ve been lucky enough to work with some amazing managers, and so developing into a great manager myself is something I’m really excited about.