Crowdsourcing Software Security

Enhancing Security by Secure Code Collaboration amongst Developers

Edited by **MOHAN DAS VISWAM**

Writing secure code in the Software Development Life Cycle (SDLC) phase and adapting to security by design should be a top priority for good developers. The benefit of secure code is that many of the potential exploits and attacks can be simply prevented by writing better and more secure code.

What is Secure Coding?

Secure coding, or secure programming, involves writing code in a high-level language that follows strict principles to prevent the potential vulnerabilities that could expose data or cause harm to a system. It is more than just writing, compiling, and releasing code into applications.

Rajesh Mishra
Sr. Technical Director
mrajesh@nic.in

Anil Kumar Jha
Sr. Technical Director
aniljha@nic.in

Rohit K. Sharma
Scientist-C
rohit.kumar89@nic.in

Software vulnerabilities are always on the rise. They may exist at any layer in the software, including the operating system, application server, database server, etc. If not addressed, they may get exploited, and organisation's data may be breached, and in a worst case scenario, even a ransomware call may come up. As more and more government services go online, insecure web apps of important government services (G2B, B2G, G2C, C2G and G2G) can result in data theft, loss of confidentiality, financial losses, and service unavailability. The industry has come up with a plethora of tools and technologies to address these issues, but one of the cheapest and most economical ways is to develop a secure SDLC and write secure code.

To fully embrace the phenomenon of secure coding, one needs to create a secure development environment that is built on a reliable and secure IT infrastructure using secure hardware, software, services, and service providers.

However, building secure software by writing secure code is easier said than done because the developers building the software normally have less idea of vulnerabilities, exploits, and remediation.

Secure software engineering, including secure coding concepts, is also not taught in the college. Moreover, in several organisations, both development and security teams may also be working in silos.

To address this problem of writing secure code, NIC has designed and developed a secure code crowdsourcing platform for exchanging secure code among the NIC developer community.

Crowdsourcing is the practice of obtaining information or input into a task or project by enlisting the services of a large number of people, either paid or unpaid, typically via the internet.

The sole purpose of this platform is to allow for the voluntary contribution of secure code snippets by NIC developers that may help other NIC developers in patching security vulnerabilities with ease that are found during the app audit process. NIC developers may use this platform to gather more information and best practices to be followed during the development of web or mobile apps.

This Secure Code Crowdsourcing platform can be accessed by the NIC developers using their Parichay credentials. (Link to the platform: https://x-seccode.nic.in)

NIC Developers may use this platform for finding remediation code for security vulnerabilities as well as for contributing their audited secure code for security vulnerabilities that can be used by another software developer. In other words, we can say that this platform is for the developer and by the developer.

This platform broadly covers the following features:

Vulnerability Listing

After logging into the system, a developer may view the list of vulnerabilities based on of his / her preferred programming language (Java, .Net, PHP), for which secure code snippets are available. These vulnerabilities are sourced from web app security issues that were found and reported by the NIC Security Audit Team. These vulnerabilities are mainly based on the OWASP Top 10, a list of the top 10 most critical web application security risks published by the Open Web Application Security Project (OWASP).

Vulnerability Description

Here, each security vulnerability is briefly explained. The vulnerability may be related to a programming language, an application framework, or open-source libraries. The patching of these vulnerabilities requires an understanding of the vulnerability, platform, and application flow. If not, it might cause more damage than just protecting the apps.

Solution

Here, the portal displays the actual code snippets for resolving the security issue for a platform. The developer may use this solution and integrate it into the application. They may also add their fixes for resolving the issues, subject to being certified by the NIC Audit Team.

Feedback

If the solution given on the platform does not work or needs some fine-tuning, it can be shared under the comment / feedback option provided with each answer.

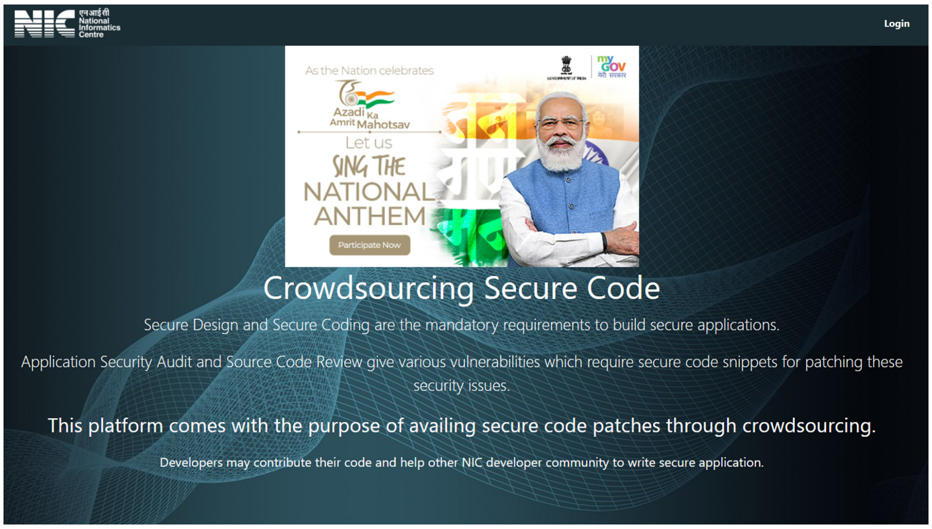
Conclusion

The platform can inculcate the culture of writing and sharing secure code among the developer community in NIC. This will certainly empower the new developers to write secure software and cut down the time required to get a security clearance certificate.

Other than writing secure code, there is also a need to enable security at other stages of the SDLC to propagate a secure development cycle across projects.

Contact for more details

Rohit Kumar Sharma
NIC Headquarters, CGO Complex
Lodhi Road, New Delhi - 110003
Email: rohit.kumar89@nic.in, Phone: 011-24305935



Crowdsourcing Secure Code

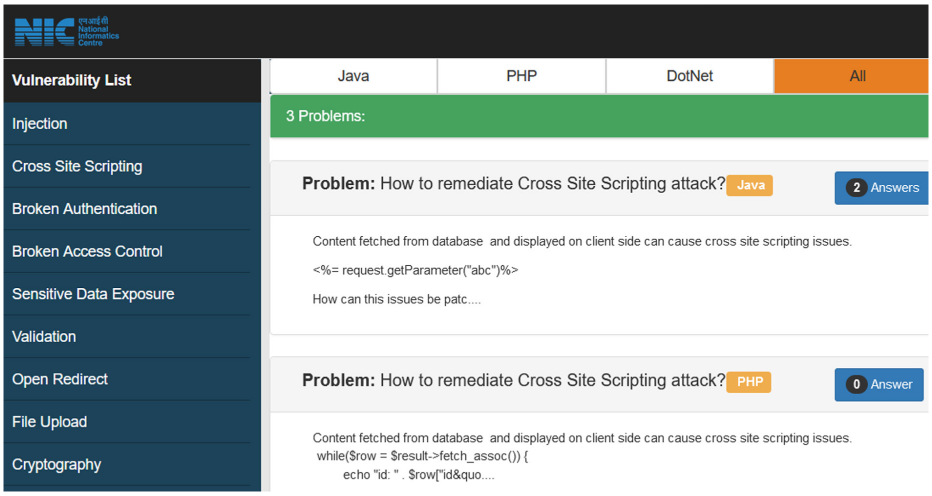
Secure Design and Secure Coding are the mandatory requirements to build secure applications.

Application Security Audit and Source Code Review give various vulnerabilities which require secure code snippets for patching these security issues.

This platform comes with the purpose of availing secure code patches through crowdsourcing.

Developers may contribute their code and help other NIC developer community to write secure application.

Fig. 13.1: NIC Source Code Crowdsourcing Portal Homepage



3 Problems:

Problem: How to remediate Cross Site Scripting attack? Java

Content fetched from database and displayed on client side can cause cross site scripting issues.

<%= request.getParameter("abc")%>

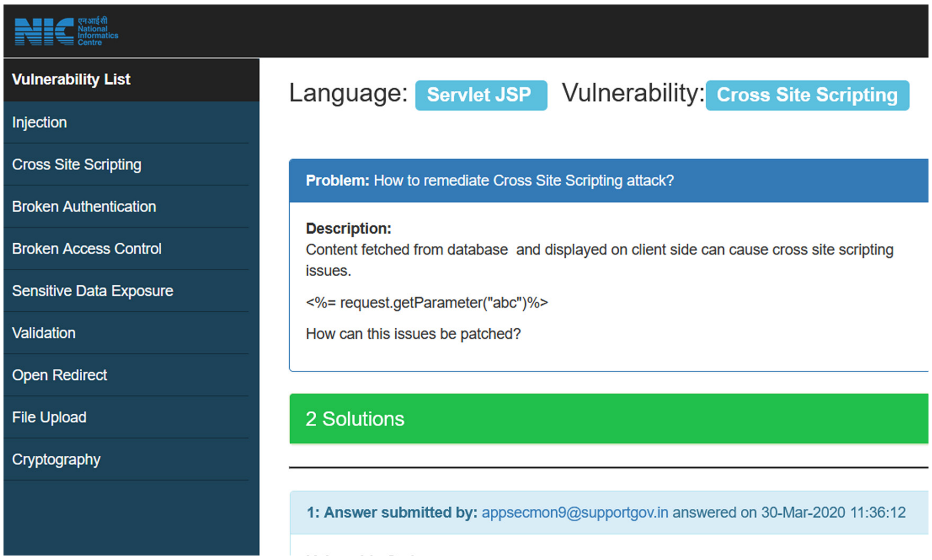
How can this issues be patc....

Problem: How to remediate Cross Site Scripting attack? PHP

Content fetched from database and displayed on client side can cause cross site scripting issues.

```
 while($row = $result->fetch_assoc()) {
       echo "id: " . $row["id&quo....
```

Fig. 13.2: Vulnerability Description Page



Language: Servlet JSP Vulnerability: Cross Site Scripting

Problem: How to remediate Cross Site Scripting attack?

Description:
Content fetched from database and displayed on client side can cause cross site scripting issues.

<%= request.getParameter("abc")%>

How can this issues be patched?

2 Solutions

1: Answer submitted by: appsecmon9@supportgov.in answered on 30-Mar-2020 11:36:12

Fig. 13.3: Vulnerability Solution Page