Secure Coding Practices in Java: Challenges and Vulnerabilities

NA MENG
STEFAN NAGY
DANFENG (DAPHNE) YAO
WENJIE ZHUANG
GUSTAVO ARANGO ARGOTY
Problem Statement

- Security libraries facilitate secure coding
  - E.g., APIs for cryptography, SSL, and authentication
- Library misuses cost lots of debugging effort, and cause security software vulnerabilities


Related Work

- Cryptographic vulnerabilities and misuses [Lazar et al. 2014, Nadi et al. 2016]
- Vulnerabilities in Android code [Acar et al. 2016]
What are the biggest challenges and vulnerabilities in secure coding practice?
Methodology

- 22,195 StackOverflow (SO) posts containing keywords “Java” and “security”
- Mainly focus on 503 posts for manual inspection after filtering the posts
  - Initially classify posts based on the software libraries under discussion
  - Further refine the classification based on the security concerns, e.g., cryptography, access control
RQ1: What are the common concerns?

- All StackOverflow posts (503)
  - Implementation questions (478)
    - Java platform security (140)
    - Cryptography (64)
    - Access control (43)
  - Understanding questions (25)
    - Java EE security (58)
    - Spring Security (267)
    - Other (13)
  - Secure communication (31)
  - Authentication (225)

19% posts are about cryptography and SSL, indicating a lack of understand of the problem domain
Developers’ major security concern has shifted from Java platform security to enterprise application security over the years.
**RQ2: What are the common programming challenges?**

- **Authentication (for Spring Security)**
  - Challenge 1: The way to integrate Spring Security with different applications varies a lot
    - E.g., Spring Boot, JBoss
  - Challenge 2: The two ways of security configuration (XML-based and Java-based) are hard to implement correctly
  - Challenge 3: Converting from XML-based to Java-based configuration is challenging
RQ3: What are the common security vulnerabilities?

https://www.acunetix.com/websitesecurity/csrf-attacks/

//Create a trust manager that does not validate certificate chains
TrustManager[] trustAllCerts = new TrustManager[] {
    new X509TrustManager() {
        public Java.security.cert.X509Certificate[]
            getAcceptedIssuers() { return null; }
        public void checkServerTrusted(…) {}
    }
};

- Standard security technology for establishing an encrypted connection between a client browser and a webserver (HTTPS)
- TrustManager should be implemented to validate servers’ certificates on the client side
Man-in-the-Middle (MITM) Attack
9 of 11 SSL-relevant posts discussed insecure code

3 posts after 2012 discussed the dangerous solution

SO contains many obsolete and insecure practices
# Social Aspects of Insecure Code on SO

<table>
<thead>
<tr>
<th>Insecure Posts</th>
<th>Total Views</th>
<th>No. of Posts</th>
<th>Min Views</th>
<th>Max Views</th>
<th>Average</th>
</tr>
</thead>
<tbody>
<tr>
<td>Disabling CSRF Protection*</td>
<td>39,863</td>
<td>5</td>
<td>261</td>
<td>28,183</td>
<td>7,258</td>
</tr>
<tr>
<td>Trust All Certs</td>
<td>491,567</td>
<td>9</td>
<td>95</td>
<td>391,464</td>
<td>58,594</td>
</tr>
<tr>
<td>Obsolete Hash</td>
<td>91,492</td>
<td>3</td>
<td>1,897</td>
<td>86,070</td>
<td>30,497</td>
</tr>
<tr>
<td><strong>Total Views</strong></td>
<td><strong>622,922</strong></td>
<td><strong>17</strong></td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
</tbody>
</table>
Social Dynamics on SO

“Do NOT EVER trust all certificates. That is very dangerous.”

“the "accepted answer" is wrong and INDEED it is DANGEROUS. Others who blindly copy that code should know this.”

“once you have sufficient reputation you will be able to comment”

“If you don't have enough rep to comment, … then participate … until you have enough rep.”

https://stackoverflow.com/questions/10594000/when-i-try-to-convert-a-string-with-certificate-exception-is-raised
Conclusion

- A lot of developers do not appear to understand the security implications of coding options, showing a lack of cybersecurity training
- Spring Security usage is very popular, overly complicated, and poorly documented
- The social dynamics among askers and responders may impact people’s security choices
Our StackOverflow data set is available at
http://people.cs.vt.edu/nm8247/icse18.xlsx
September 30-October 2, 2018 At the Hyatt Regency, Cambridge, MA

IEEE Secure Development Conference

Sponsored by the IEEE Cybersecurity Initiative and the IEEE Computer Society Technical Committee on Security and Privacy

Submission open for 1-page abstracts
Practitioner’s abstract due: July 20

Daphne Yao (VT)
Stephen Chong (Harvard)

https://secdev.ieee.org/2018/home
References


Cryptography

- Challenge 1: The error message did not provide sufficient useful hints about fixes
- Challenge 2: It is difficult to implement security with multiple programming languages
  - E.g., Encryption in Python and decryption in Java
- Challenge 3: Implicit constraints on API usage cause confusion

```java
//privKey should be in PKCS#8 format
byte[] privKey = ...;
PKCS8EncodedKeySpec keySpec =
    new PKCS8EncodedKeySpec(privKey);
```
Research Questions

- What are the developers’ common concerns on Java secure coding?
- What are the common programming challenges?
- What are the common security vulnerabilities?
**SO Post Filtering**

- Filter less useful posts
  - Removing duplicated posts, posts without accepted answers, and posts whose questions received negative votes
  - Removing posts without code snippets with keyword-based search: “public” and “class”
  - Discarding irrelevant posts based on manual inspection