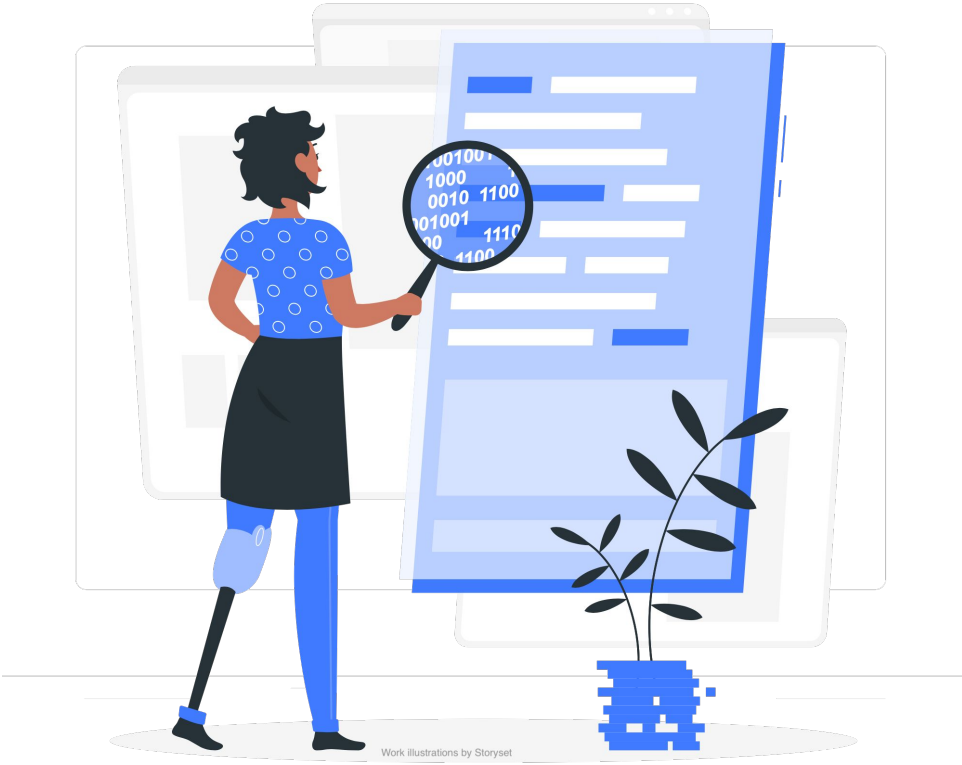
Secure Software Development

Mohammad Tahaei @Uni Edinburgh
Lead researcher
Mixed methods: Survey, experimental design

Overview

Problem Statement: Despite the **availability** of **developer tools** aimed at early-stage **security vulnerability identification**, the **efficacy** of these tools is in question.

Research Hypothesis: **Utilizing a tool improves developers' ability** in effectively remedying **security** vulnerabilities.



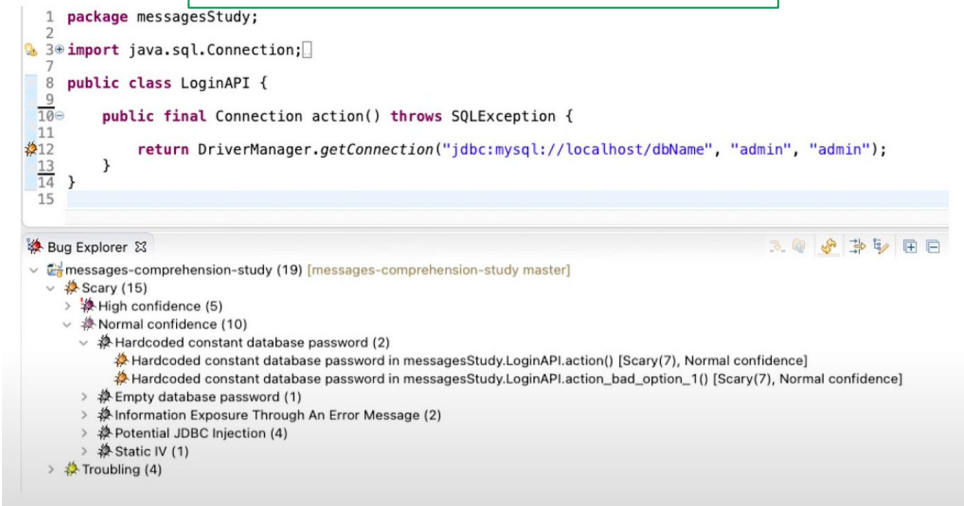
Context

- SonarQube and SpotBugs are **open-source** tools designed for analyzing code and **identifying issues**, including **security**.
- However, developers require **practical** action points to address issues.
- Research literature **lacked prior** coverage of this specific research area.

Program (Static) Analysis Tools

- **Safeguarding** software from **malicious attacks** is essential for success.
- Programming analysis **tools** drive **efficient** software development, fostering **security** in a fast-paced environment.

SpotBugs Output Example



Research Questions

- How **effective** are these tools, truly?
- Could they offer more than simply indicating a line number?
- What are developer **attitudes** toward these tools?
- Is there a **correlation** between attitudes and vulnerability-fixing ability?

Constraints

- Responsible for the **entire research** process: ideation, ethical approvals, recruitment, data collection, analysis, and composing the research paper.
- Recruitment challenges: **hard to reach developers**
- **Confirmatory** and **exploratory**

Method

- Conducted **online experiment** involving developers (N=132).
- Each participant saw **four examples of code with a security issue**:
  - SQL injection
  - Hard-coded credentials
  - Encryption
  - Logging sensitive data
- Participants were **divided into three groups**:
  - Control: No details, only line numbers provided
  - SonarQube: Tool-generated outputs
  - SpotBugs: Tool-generated outputs
- Attitudinal, behavioral, & demographics

- Analysis:
  - Qualitative for open-ended questions
  - Quantitative for closed questions

The following Java code establishes a database connection. Please answer the following questions based on the code and the provided output from the static analysis checker.

```
1  import java.sql.Connection;
2  import java.sql.DriverManager;
3  import java.sql.SQLException;
4  import java.util.Properties;
5
6  public final Connection action() throws SQLException {
7
8      return DriverManager.getConnection("jdbc:mysql://localhost/dbName", "admin", "admin");
9  }
```

Imagine that you finish writing the method above, and then run a static analysis tool on your code. After running, the tool generates the following notification on line 8:

Remove this hard-coded password.

You ask the tool for more details and see the information below:

Credentials should not be hard-coded

- Vulnerability
- Blocker
- Main sources
- cert, cwe, owasp-a2, sans-top25-porous
- Available SinceJan 27, 2020
- SonarAnalyzer (Java)
- Constant/issue: 30min

Because it is easy to extract strings from a compiled application, credentials should never be hard-coded. Do so, and they're almost guaranteed to end up in the hands of an attacker. This is particularly true for applications that are distributed.
Credentials should be stored outside of the code in a strongly-protected encrypted configuration file or database.
It's recommended to customize the configuration of this rule with additional credential words such as "oauthToken", "secret", ...

Noncompliant Code Example

```
Connection conn = null;
try {
  conn = DriverManager.getConnection("jdbc:mysql://localhost/test?" +
       "user=steve&password=blue"); // Noncompliant
  String uname = "steve";
  String password = "blue";
  conn = DriverManager.getConnection("jdbc:mysql://localhost/test?" +
       "user=" + uname + "&password=" + password); // Noncompliant


  java.net.PasswordAuthentication pa = new java.net.PasswordAuthentication("userName", "1234".toCharArray());  // Noncompliant
```

Compliant Solution

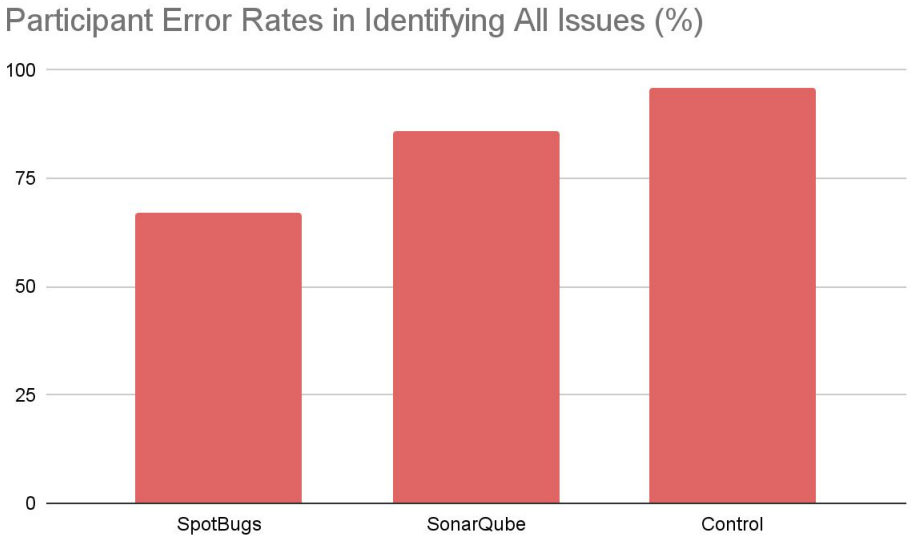
```
Connection conn = null;
try {
  String uname = getEncryptedUser();
  String password = getEncryptedPass();
  conn = DriverManager.getConnection("jdbc:mysql://localhost/test?" +
       "user=" + uname + "&password=" + password);
```

See

- OWASP Top 10 2017 Category A2 - Broken Authentication
- MITRE, CWE-798 - Use of Hard-coded Credentials
- MITRE, CWE-259 - Use of Hard-coded Password
- CERT, MSC03-J. - Never hard code sensitive information
- SANS Top 25 - Porous Defenses
- Derived from FindSecBugs rule Hard Coded Password

High-Level Findings

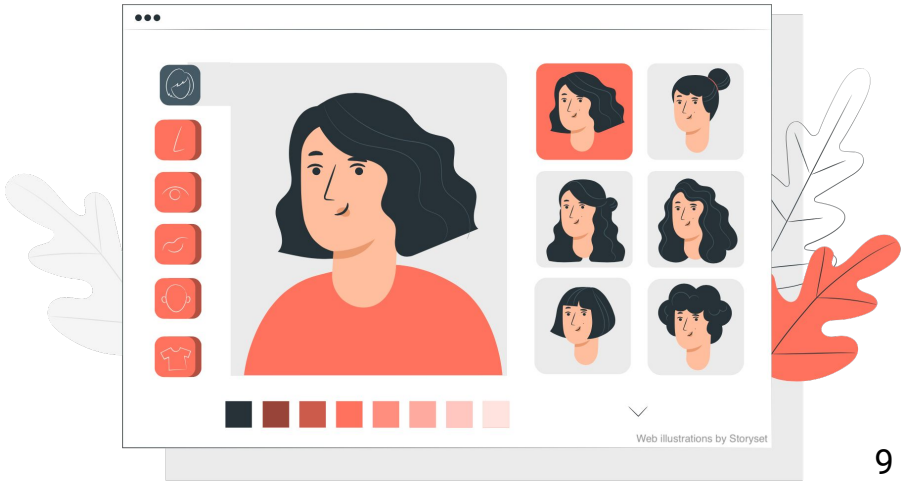
- Tool yielded **minor improvements** (insignificant statistically).
- Most tools & Control participants gave **at least one incorrect** code-correction response.
- Qualitative finding: Both vulnerable & valid code desired.



Product Implications

- **Customize** the tool for various years of experience!
  - **6+ years** of prior dev experience ~ **4X accuracy** boost.
  - **1X vulnerability importance** ~ **2X** boost.

- Tailoring the output to novice and experienced users is essential!



Broader Impact

- Presented a **peer-reviewed research paper** at the leading human-computer interaction conference.
- **26** subsequent research papers have drawn upon and expanded upon this work within **just 2 years** (as of August 2023).
- **Invited** to deliver a **talk** to **Snyk's** product and research team, a prominent player in the security domain.

Reflections - What I Learned?

- To address the challenge of **low participant return rates** through the exploration of innovative methods.
- To include a **skilled statistics professional** in the project to offer feedback and insights, augmenting the decision-making process.

What Do Former Teammates Have to Say?

[Quotes from LinkedIn recommendations]

"efficient in **time management**, allowing him to keep the research projects **on track** and **deliver** the results **on time**, without losing the quality." [Alisa Frik, Senior UXR]

"**highly professional and amiable** colleague . . . was involved in a number of projects, **worked with a colleagues at varying levels of seniority** and experience, and **acted as a mentor** for junior colleagues." [Louise Evans, Research Manager]

"easily one of my **most productive** students. He has an **excellent eye for interesting research problems** and the **attention to detail** needed to realize them."
[Kami Vaniea, Associate Professor]

Contact

https://mohammad.tahaei.com/

https://www.linkedin.com/in/tahaei/

mohammad@tahaei.com