Vulnerable Beneath the Source Code: Increasing Vulnerability by Decreasing Perceived (and Increasing Actual) Complexity - Shawan Ware


Keep it Simple, Keep it Safe – Cybersecurity Research on Impacts of Increasing Complexity of Modern Enterprise Solutions

Sensible Data Availability in High-Level Language Applications

Methods & Assumptions

- It is assumed that the enterprise application’s source code is available for analysis. This assumption allows the team to provide detailed feedback on the insecure-handling practices most likely to result in sensitive data persistence.
- It is assumed that the application does not implement stringent authentication within the application developed in Java for sensitive information.
- Sensitive information is frequently handled using potentially insecure methods.
- Java’s automated and abstracted memory management process large cybersecurity exposures. As developers continue to learn and practice high-level language application development, they are less likely to be responsible for the mismanagement of sensitive data persisting within memory boundaries.

Research Timeline

- A large amount of research remains to be completed. As the Cybersecurity Capture course proceeds, work will continue at a steady pace. Further, we have identified research directions and practices that will likely be important for the ongoing effort.
- Thus far, the research has led to the development of a web app which would allow a user to select the vendors they wish to support. This is type of research is extremely important to all institutions to help protect sensitive data and inform the research community.

References & Resources

2. Java Development Kit 8. Java 8 is the last major upgrade to Java 7. Java 8 adds several new features, including lambdas for functional interfaces, method references, and varargs.

Purpose

- The proposed layered solutions is the goal of an ability to understand the business implications of security vulnerabilities. This research is aimed at understanding the security impacts of Java SE 8 (J9) and its implications for enterprise Java applications.
- The purpose of this research is to study the vulnerability of sensitive data. If time permits, the second application will be deployed on the remote server and similarly tested, with data collection repeated.
- This is type of research is extremely important to all institutions to help protect sensitive data and inform the research community.

The timeline above is a rough example of what we hope to create by the end of our research. We plan to examine 10 enterprise VM instances and their vulnerabilities in order to create a web app which would allow a user to see which services they wish to support and which use Java SE 8 or other vulnerable Java versions.

The web app would then output a timeline that shows how long both solutions were vulnerable individually and together as a layered solution.

- The above graph is what an exploitation curve might look like. It shows the number of vulnerabilities disclosed and the number of exploitations happen and the vulnerability may be passed around the dark web. Once the vulnerability is disclosed, you see a large increase in the number of exploitations happen and the vulnerability may be passed around the dark web.
- Once the vulnerability is disclosed, you see a large increase in the number of exploitations happen and the vulnerability may be passed around the dark web.
- The above graph is what an exploitation curve might look like. It shows the number of vulnerabilities disclosed and the number of exploitations happen and the vulnerability may be passed around the dark web.