Title: Java Deserialization Vulnerabilities and Mitigations

Length: 90 Minutes

Description:

This tutorial provides developers with practical guidance for securely implementing Java Serialization. Java deserialization is a clear and present danger as its widely used both directly by applications and indirectly by Java subsystems such as RMI (Remote Method Invocation), JMX (Java Management Extension), JMS (Java Messaging System). Deserialization of untrusted streams can result in remote code execution (RCE), denial-of-service (DoS), and a range of other exploits. Applications can be vulnerable to these attacks even if they did nothing wrong.

This tutorial explains and demonstrate these attacks and show developers how to securely code their systems to support Java serialization. Material in this presentation was derived from the Addison-Wesley book The CERT Oracle Secure Coding Standard for Java and is supported by the Secure Coding Rules for Java LiveLessons videos.

Prerequisites

The tutorial is designed primarily for Java SE 8 developers but should also be useful to developers using older versions of the SE platform as well as Java EE and ME developers. The tutorial assumes basic Java programming skills but does not assume an in-depth knowledge of software security.

Students must bring a personal computer equipped with the following:

- Java SE Development Kit 8
- Eclipse IDE for Java Developers or other a Java 8 compatible IDE
- 100MB or greater of free hard disk space
- The latest version of Adobe Reader
- Students will receive instructions on obtaining the tutorial exercises, demos, and examples. Before class, students should ensure that these resources are available from their personal computers.

Participants should come away from the tutorial with a working knowledge of common programming errors that lead to software vulnerabilities, how these errors can be exploited, and effective mitigation strategies for preventing the introduction of these errors.

In particular, participants will learn how to:

- Explain the need for secure coding
- Validate and sanitize data
- Securely deserialize Java streams
- Securely implement exception handling
- Predict how the numerical types behave in Java
- Avoid pitfalls in the use of characters and strings
- Securely process input and output
Moreover, the tutorial encourages programmers to adopt security best practices and develop a security mindset that can help protect software from tomorrow’s attacks, not just today’s.

**Textbook**


**Outline:**

**Serialization**

1. Understand Java object serialization
2. Beware of hidden constructors
3. Mitigate security risks of deserializing unvalidated data
4. Whitelist valid deserialization objects
5. Apply appropriate security permissions to serialization and deserialization
6. Use serialization proxies instead of serialized instances

**Exercises and Demonstrations**

1. Remote command execution and denial-of-service attacks
2. Whitelisting using JEP 290

Demos and tutorials available at: [https://github.com/rcseacord/JavaSCR](https://github.com/rcseacord/JavaSCR)

**Author:**

Robert C. Seacord
Principal Security Consultant
NCC Group

**Bio**

Robert is a Principal Security Consultant with NCC Group where he works with software developers and software development organizations to eliminate vulnerabilities resulting from coding errors before they are deployed. Previously, Robert led the secure coding initiative in the CERT Division of Carnegie Mellon University’s Software Engineering Institute (SEI). Robert is also an adjunct professor in the School of Computer Science and the Information Networking Institute at Carnegie Mellon University. Robert is the author of six books, including “The CERT C Coding Standard, Second Edition” (Addison-Wesley, 2014), “Secure Coding in C and C++, Second Edition” (Addison-Wesley, 2013), and “Java Coding Guidelines: 75 Recommendations for Reliable and Secure Programs” (Addison-Wesley, 2014). Robert is on the Advisory Board for the Linux Foundation and an expert on the ISO/IEC JTC1/SC22/WG14 international standardization working group for the C programming language.

**Similar Tutorials**

Secure Coding in Java has been taught dozens of times to audiences ranging in size from 5 to 30 participants. Most recently, this material was taught as part of three day course at AppSecEU 2017 in Belfast.