# Type-Based Capability for Java

Xi Wu[1], Yi Lu[2], Ian J. Hayes[1], and Larissa A. Meinicke[1]

[1] School of ITEE, The University of Queensland, Brisbane 4072, Australia
Ian.Hayes@itee.uq.edu.au, {xi.wu,l.meinicke}@uq.edu.au
[2] Oracle Labs Australia yi.x.lu@oracle.com

The programming language Java is widely used in a variety of domains, ranging from embedded devices to e-commerce. Security issues raised by Java are drawing a growing interest. The most critical issue is security flaws within the Java Class Library (JCL), which may expose vulnerabilities for applications. Since Java-SE contains a huge number of lines of code, securing the JCL is still a challenge.

In order to provide more secure access to resources, capabilities for Java were recently proposed by Hayes et al. with the aim of preventing security flaws. Capabilities are interface-like in Java and code can only access resources if it is given explicit capabilities, allowing replacement of the use of doPrivileged blocks. Currently, capabilities are created and managed by the capability manager, which is a (meta)capability and responsible for checking whether the user or process has the necessary permissions to grant capabilities or not via the run-time permission check. However, no further control on the propagation of capabilities may cause them being potentially obtained by unauthorized code. Thus, a static guarantee for correct permission checks becomes a focus of our current ongoing work.

This presentation will start with reviewing some features of Java security and exploring some security issues raised by Java. A brief overview of capabilities for Java will also be given. We continue by highlighting our attempt to transform dynamic permission checks in Security Manager to static type checks and reporting our ongoing work on how to use types to track the information flow to restrict the access control between codes and capabilities.