Web Security Analysis for Java Using Conditional Weighted Pushdown System

HUA, Vy Le Thanh (1010225)

School of Information Science,
Japan Advanced Institute of Science and Technology

August 9, 2012

Keywords: stack-based access control, stack inspection, conditional weighted pushdown system, program analysis.

In modern Web platforms, such as Java or Microsoft .NET, applications comprise components from different origins with diverse levels of trust. A *stack-based access control* mechanism is employed in an attempt to prevent untrusted codes from accessing protected resources. Access control policies are expressed in terms of permissions, and a policy file is configured when an application is deployed. Developers can set checkpoints in their programs, such as system libraries, and access control is enforced dynamically at runtime by stack inspection. The call stack will be inspected at the checkpoints. If any caller in the current context does not possess the required permission, the program execution will be interrupted immediately.

Of a practical perspective, such runtime inspection may cause a high overhead cost. If access control at some checkpoints always succeed at runtime, the runtime overhead can be reduced by removing such redundant checkpoints. On the other hand, to our knowledge of practiced approaches, policy files are generated manually by developers based on domain-specific knowledge, and measured by testing as to whether the policy file allows the application to run properly. Testing cannot cover all program behaviors and the application could malfunction given the misconfigured policies. This thesis is dedicated to solving these problems by proposing a static permission analysis that detects whether stack inspections in concerned

Copyright © 2012 by HUA, Vy Le Thanh

domains always succeed given a policy file, and developing a research prototype for it.

As for program analysis, pushdown systems are well understood as natural models of (sequential) programs with recursive procedures. Recently conditional weighted pushdown systems (CWPDS) are proposed as an abstract model for programs with stack inspection. We implement model checking algorithms for CWPDSs, and look into the possibility of improving its practical efficiency. Particularly, to solve its forward reachability problem, we propose an on-the-fly algorithm which, by our preliminary empirical study, outperforms the original approach regarding both runtime and memory consumption. Furthermore, we specify the reachability problem of conditional pushdown systems in the declarative language Datalog. Such specification gives us the possibility of benefiting from a variety of scalable implementations of Datalog.

Finally, we thoroughly investigate the application of CWPDSs to the aforementioned permission analysis problems, i.e., given a policy file and protection domains of interest, our permission analysis systematically determines program points to be examined in the analysis, and detects whether involved access control always succeeds or may fail, with assuming permissions required for stack inspection over the analysis points. Since our analysis assumes non-trivial points-to analysis and string-analysis, the complete realization of our analysis framework is underway. We illustrate how our permission analysis framework works with small yet non-trivial examples, and sketch a blueprint to realize the entire framework. A sample usage of our package `cwpds` is provided in which the on-the-fly algorithm is implemented.