

DBTaint: Cross-Application Information Flow Tracking via Databases

Benjamin Davis, Hao Chen (University of California, Davis)

Motivation

As we become more reliant on Web services, these services become more attractive targets to attackers. Tracking the flow of "tainted" (untrusted) data through a system is a proven method of detecting vulnerabilities and preventing many types of attacks. Unfortunately, most existing information flow tracking systems are a poor fit for the multi-application architecture of modern Web services.

The Problem

Many existing information flow system can only track tainted data through a single application. However, most Web services include at least one Web application and one database application. Single-application systems lose taint information at the application boundaries, leaving users with imperfect options including:

- Consider all data from databases tainted
- Consider all data from databases untainted
- Manual annotation
- Application-specific decisions or specialized environments

Many system-wide information flow tracking systems are too coarsely grained (operating at the process or file level), while fine-grained systems that operate at the instruction level lack the ability to make use of high-level database semantics.

DBTaint provides

- End-to-end information flow tracking through Web services, across Web applications and databases
- Mechanisms for leveraging single-application information flow systems in multi-application Web services
- Information flow tracking in existing Web services, requiring no changes to Web applications
- Taint propagation through database functions

Use Cases

- Persistent cross-application information flow tracking
- Regression testing, bug detection
- Identification of incomplete sanitization policies via column analysis

Implementation

Supports:

- Perl applications
- Perl DataBase Interface (DBI) API
- Java applications
- Java Database Connectivity (JDBC) API
- PostgreSQL Database

Evaluation

- | | |
|---|--|
| RequestTracker (ticket tracking system) | JForum (discussion board system) |
| •60,000+ lines of Perl | •30,000+ lines of Java |
| •Perl DBI (DataBase Interface) API | •Java Database Connectivity (JDBC) API |
| •Perl taint mode | •Character-level taint engine [Chin '09] |

