Generating Vulnerability Signatures for String Manipulating Programs
Using Automata-based Forward and Backward Symbolic Analyses

Fang Yu, Muath Alkhalaf, Tevfik Bultan
{yuf, muath, bultan}@cs.ucsb.edu

Overview

- We present automata-based symbolic string analyses for automatic verification of string manipulating programs
- We compute the pre- and post-conditions of common string functions using deterministic finite automata (DFAs)
- We compute DFAs that characterize all possible values that string expressions can take in any possible execution of a program using forward and backward symbolic analyses

Implementation

We use Pixy [3] as a front end and MONA [2] automata package for automata manipulation. The implementation consists of the following parts:

- **PHP Parser**: Parses the PHP program and constructs the control flow graph (CFG)
- **Taint Analyzer**: Identifies sinks (sensitive functions that may use tainted data) and generates their dependency graphs using alias and dependency analyses. If no sinks are found, the application is not vulnerable
- **String Analyzer**: Implements vulnerability (forward and backward) analysis on dependency graphs for all sinks that are found
- **String Manipulation Library (SML)**: Handles all core string and automata operations such as concatenation, prefix, suffix, replace, intersection, union, and widen

Why do we need string analysis?

The top three vulnerabilities in OWASP's top ten list (which lists the most serious web application vulnerabilities) are due to improper manipulation of strings:
1. Cross Site Scripting (XSS)
2. Injection Flaws (such as SQL injection)
3. Malicious File Execution (MFE)

Generating vulnerability signatures

- We generate vulnerability signatures using backward analysis
- A vulnerability signature is a characterization that includes all malicious inputs that can be used to generate attack strings
- Use backward analysis starting from the sink nodes and traverse the dependency graph backwards to find out malicious inputs (uses pre-condition computation)
- Both forward and backward analyses use an automata-based widening operator [1] to accelerate fixpoint computations

Vulnerability

- The purpose of the replace statement in line 4 is to remove any special characters from the input to prevent such attacks
- The echo statement in line 5 can contain a Cross Site Scripting (XSS) vulnerability

What does our string analysis achieve?

- Detects vulnerabilities in web applications that are due to string manipulation
- Proves the absence of vulnerabilities in web applications that use proper sanitization
- Generates a characterization of all malicious inputs that may compromise a vulnerable web application

Contributions

- Sound verification techniques for PHP web application vulnerability analysis and vulnerability signature generation, focusing on SQLCI, XSS and MFE attacks
- Combining forward and backward symbolic string analyses for vulnerability signature generation
- Implementation of forward and backward image computations for string operations (including complex operations such as preg_replace() using a symbolic automata representation (MBDDs))
- The first automata-based string analysis tool that can automatically generate vulnerability signatures of vulnerable PHP programs
- The implementation and benchmarks are available at: http://www.cs.ucsb.edu/~vlab/stranger

References