Instituto Superior Técnico - Alameda

Software Security Project Report

Static Code Analysis Tool

Discovering vulnerabilities in PHP Web Applications

Group 10

Rui Ventura 81045

Diogo Castilho 78233

November 19, 2017

Experimental part

Our analysis tool was conceived using the `Python` programming language, version `3.6.3`. It consists of a main component, the analyser (`analyser.py`), containing the logic for traversing some elements of an AST, and a pattern module (`pattern.py`) that houses the `Pattern` class, used to instantiate objects that represent vulnerable patterns which can be found in the `patterns` file, another component of the tool.

Components

Analyser

The main component is run by invoking it and passing it a PHP program slice in JSON format as a command line argument.

```
analyser.py /path/to/slice.json
```

The slice is already in the form of an AST, according to the syntax of the AST's generated by Glayzzle's PHP Parser [1].

Patterns

The `patterns` file contains a set of vulnerable patterns with the following format:

```
Vulnerability
Entry_1,Entry_2,...,Entry_i
Sanitizer_1,Sanitizer_2,...,Sanitizer_j
Sink_1,Sink_2,...,Sink_k
```

Listing 1: Vulnerable pattern template

where `Vulnerability` is the name of the vulnerability, `Entry` is an entry point, `Sanitizer` is a sanitization/validation function, and `Sink` a sensitive sink. Example:

```
SQL injection (PostgreSQL)
$_GET,$_POST,$_COOKIE,$_REQUEST
pg_escape_string,pg_escape_bytea
pg_query,pg_send_query
```

Listing 2: SQL Injection pattern, specific to PostgreSQL

References

[1] Glayzzle and Various Contributors. PHP Parser. Available at `https://github.com/glayzzle/php-parser`. NodeJS PHP Parser - extract AST or tokens (PHP5 and PHP7).