INSTITUTO SUPERIOR TÉCNICO - ALAMEDA

SOFTWARE SECURITY PROJECT REPORT

Static Code Analysis Tool

Discovering vulnerabilities in PHP Web Applications

Group 10

Rui Ventura   Diogo Castilho
81045   78233

November 19, 2017

1 The Tool

The tool was conseived using the `Python` programming language, version
`3.6.3`. It consists of a main component, the analyser (`analyser.py`), and
a pattern module (`pattern.py`) that houses the `Pattern` class, used to in-
stantiate objects that represent vulnerable patterns.

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

```
Vulnerability
Entry₁,Entry₂,...,Entryᵢ
Sanitizer₁,Sanitizer₂,...,Sanitizerⱼ
Sink₁,Sink₂,...,Sinkₖ
```

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 (MySQL)
$_GET,$_POST,$_COOKIE,$_REQUEST,...
mysql_escape_string,mysql_real_escape_string
mysql_query,mysql_unbuffered_query,mysql_db_query
```

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].

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).