Overview

Web applications are used in several contexts of the smart grid, allowing for instance the creation of flexible user interfaces. The security of these applications continues to be a challenging problem, as programmers make mistakes and leave vulnerabilities in the source code.

The WAP tool aims to assist on the protection of web applications. It helps programmers find vulnerabilities in their source code and correct them, by automatically fixing the flaws that were found. For more accurate detection, the tool integrates machine learning to predict if the vulnerabilities that were identified are real or false alarms. Experts in secure coding can configure WAP for new vulnerability classes without programming.

Approach

Web Application Protection (WAP) aims to protect automatically smart grid applications by analyzing and fixing their source code.

Our approach is based on:

**TAINT ANALYSIS** to find vulnerabilities

WAP searches for input validation vulnerabilities in PHP source code, such as XSS and SQL injection. It does taint analysis, checking if inputs can reach sensitive functions (sensitive sinks) without proper sanitization or validation. The lists of input entry points, sensitive sinks, and sanitization/validation functions are produced by humans. However, our research has shown that taint analysis tends to produce false positives, i.e., to identify bugs that do not exist in practice.

**DATA MINING** to predict false positives

WAP uses a second form of analysis - data mining - to refine the results of taint analysis, classifying them as being false positives or real vulnerabilities. This analysis is based on learning about attacks automatically using a labeled data set of true and false positives.

**CODE FIXES** to remove vulnerabilities

WAP also removes identified bugs automatically using code fixes. These fixes do essentially proper validation or sanitization of user input before it is used in a sensitive sink. Fixes are PHP functions developed by us, as security problems were identified in several of sanitization functions available in PHP. They are inserted in sensitive sinks or close to them.

Experiments

**Research questions:**

1. Is WAP able to process a large set of PHP applications? (see Table 1)
2. Does the tool detect the vulnerabilities that it was programmed to detect? (see Table 1)
3. Is it more accurate and precise than other tools that do not combine taint analysis and data mining? (see Table 2)

We compare WAP with Pixy and PhpMinerII. Pixy does taint analysis such as WAP to discover vulnerabilities, but only detects SQL injection and XSS vulnerabilities. PhpMinerII does data mining of program slices that end at a sensitive sink, regardless of data being propagated through them starting at an entry point or not. PhpMinerII does this analysis to predict vulnerabilities, whereas WAP uses data mining to predict false positives in vulnerabilities detected by the taint analyzer.

Vulnerabilities

The WAP tool can detect eight classes of input validation vulnerabilities:

- SQL injection
- XSS (reflected and stored)
- Remote file inclusion (RFI)
- Local file inclusion (LFI)
- Directory/Path traversal (DT/PT)
- Source code disclosure (SCD)
- OS command injection (OSCI)
- PHP command injection (PHPCI)

**Table 1.** Two sets of experiments with WAP.

<table>
<thead>
<tr>
<th>Metric</th>
<th>Exper 1</th>
<th>Exper 2</th>
<th>Total</th>
</tr>
</thead>
<tbody>
<tr>
<td>Web applications</td>
<td>45</td>
<td>54</td>
<td>99</td>
</tr>
<tr>
<td>WP plugins</td>
<td>--</td>
<td>115</td>
<td>115</td>
</tr>
<tr>
<td>Files</td>
<td>6,700</td>
<td>8,300</td>
<td>15,000</td>
</tr>
<tr>
<td>MLoC</td>
<td>1.3</td>
<td>2.0</td>
<td>3.3</td>
</tr>
<tr>
<td>Vulnerabilities</td>
<td>388</td>
<td>582</td>
<td>970</td>
</tr>
<tr>
<td>Zero-day</td>
<td>1</td>
<td>519</td>
<td>520</td>
</tr>
</tbody>
</table>

**Table 2.** Evaluation between WAP, Pixy and PhpMinerII.

<table>
<thead>
<tr>
<th>Metric</th>
<th>WAP</th>
<th>Pixy</th>
<th>PhpMinerII</th>
</tr>
</thead>
<tbody>
<tr>
<td>accuracy</td>
<td>92.1%</td>
<td>44.0%</td>
<td>87.2%</td>
</tr>
<tr>
<td>precision</td>
<td>92.5%</td>
<td>50.0%</td>
<td>85.2%</td>
</tr>
</tbody>
</table>

References


SEGRID Security for smart Electricity GRIDs

www.segrid.eu  info@segrid.eu

This project has received funding from the European Union's Seventh Framework Programme for research, technological development and demonstration under grant agreement no. 607109