SIFEX: Tool for Static Analysis of Browser Extensions for Security Vulnerabilities

Shikhar Agarwal, Indian Institute of Technology, Delhi

ADVISORS: Marianne Winslett, Sruthi Bandhakavi

INTERNSHIP GOAL: To build software environment to compare all 3 tools on Firefox Extensions.

Which Existing Tool for Identifying Potential Vulnerabilities is Best?

<table>
<thead>
<tr>
<th>Tool</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>GATEKEEPER [2]</td>
<td>Inclusion-based, context-insensitive pointer analysis. Relies on heap-based allocation. Developer inputs database of facts. Datalog facts are generated at each node of Call Graph. This is then analyzed by bddbddb tool.</td>
</tr>
</tbody>
</table>

Which tool correctly identifies the potential vulnerabilities the fastest, with the fewest false positives?

SIFEX: Tool for Static Analysis of Browser Extensions for Security Vulnerabilities

Generates explicit sources

Tool 1: This is an ml-lex/ml-yacc-based lexer/parser. It generates a list of sources for the confidentiality policy of SIF. The important features are:

- It takes care of simple aliasing
- All the fields of a tainted object are made tainted

For example, if document.content is a sensitive source, then for statements of the form dummy1 = document.content;
dummy2 = dummy1;
dummy1 and dummy2 would be tainted too.

For example, if document.content.field is tainted, then document.content.field is treated as tainted.

Tool 2: Parses the JavaScript file and makes minor replacements in the code to make it compatible with JSure, which is used by SIF.

The Residual Policy, obtained by passing the code through the SIF framework, is manually inspected for detecting desired flows. The potential vulnerabilities are further checked to determine whether they can be exploited.

Research Results

<table>
<thead>
<tr>
<th>FLOW PATTERN</th>
<th>GREP ALERTS</th>
<th>VEX ALERTS</th>
<th>SIFEX ALERTS</th>
</tr>
</thead>
<tbody>
<tr>
<td>Content Doc to innerHTML</td>
<td>534</td>
<td>46</td>
<td>142</td>
</tr>
<tr>
<td>RDF to innerHTML</td>
<td>60</td>
<td>4</td>
<td></td>
</tr>
<tr>
<td>Content Doc to eval()</td>
<td>430</td>
<td>13</td>
<td>38</td>
</tr>
</tbody>
</table>

SIFEX was able to detect the previously known exploitable vulnerabilities in the following extensions:

- Fizzle v-0.5.1, 0.5.2
- Beatnik v-1.2
- Feed-sidebar v-3.1
- Wikipedia Toolbar v-0.5.9

The following are lines from Feed-sidebar 3.1 where the vulnerability was detected:

```
88 1926
18911
```

Time Taken (in sec)

<table>
<thead>
<tr>
<th>Tool</th>
<th>Time Taken</th>
</tr>
</thead>
<tbody>
<tr>
<td>Tool 1</td>
<td>88</td>
</tr>
<tr>
<td>Tool 2</td>
<td>1926</td>
</tr>
<tr>
<td>SIFEX</td>
<td>18911</td>
</tr>
</tbody>
</table>

Future Work

- Study the potential vulnerabilities indicated by SIFEX and classify them as Exploitable or Non-Exploitable.
- Develop a similar tool with Gatekeeper as back-end and compare the results.

References