Towards Easier Security Patch Porting

Luciano Bello
luciano@debian.org
luciano.bello@ibm.com

Joint work with Joey Schwarz and Jesús Pérez
Fix issue #658

Loading branch information

plusvic committed on GitHub May 14, 2017

Showing 8 changed files with 115 additions and 14 deletions.

libyara/arena.c

void yr_arena_destroy()

// VR_ARENA* arena - Pointer to the arena.
// Returns:

A pointer to the arena's data. NULL if no data has been written to

the arena yet.

libyara/exec.c

SOFTWARE EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

[jessie] - yara <no-dsa> (Minor issue)

https://github.com/VirusTotal/yara/issues/658
https://github.com/VirusTotal/yara/commit/053e67e3ec81cc9268ce30eaf9d6663d8639ed1e

debain
Fix issue #658

Loading branch information

plusvic committed on GitHub: May 14, 2017

Showing 8 changed files with 113 additions and 14 deletions

libyara/arena.c

@@ -2,7 +2,7 @@
 void yr_arena_de

    // VR_ARENA* arena - Pointer to arena, size of arena, size of object
    // Returns:

    // A pointer to the arena's data
+   // A pointer to the arena's data

    // the arena yet.

libyara/exec.c

@@ -44,7 +44,7 @@
 SOFTWARE EVEN IF

    // can't find file to patch at input line 237
 Perhaps you used the wrong -p or --strip option?

The text leading up to this was:

-------------
diff --git a/libyara/sizedstr.c b/libyara/sizedstr.c
index 767aa7e0..3bf058ac 100644
--- a/libyara/sizedstr.c
+++ b/libyara/sizedstr.c

Demo
Statistics

- **Sid**: 75
- **Buster**: 90
- **Stretch**: 50
- **Jessie**: 125
- **Wheezy**: 225

https://board.net/p/PatchPorting
Current(-ish) Features

- Building as a “test”
- Quilt-ready patch as a result
- Tries patches from different providers
- More flexible heuristics than patch
- Visualizes ambiguities
- Spots trivial cases easily
- Easy customizable re-run
Goal
To have a platform where security patches are shared and automatically ported to different versions, and multiple stakeholders can define together what is a good patch and consume solutions from others.
Roadmap and Plans

- Improve performance
- Integration with the tracker
  - If that's okey with you :)  
  - A Patch field would help
  - SSO for custom runs
Roadmap and Plans

- Better (and more) crawling for patches (e.g. from other distros)
- Add POC testing?
- Interface improvements
- Machine learning (?)
- Add new and fancy heuristics
try:
from django.http import HttpResponse
result_data = self.connection.getresult()
result = pickle.loads(result_data)

except:
    self.store.fail()

class FindRequest:
    response = self.connection.getresult()
    result_data = response.read()
    results = pickle.loads(result_data)
    except:
        self.store.fail()
**use of isomorphisms**

<table>
<thead>
<tr>
<th>Line</th>
<th>Code</th>
</tr>
</thead>
<tbody>
<tr>
<td>18</td>
<td><code>debug(&quot;source = &quot;+ p.source)</code></td>
</tr>
<tr>
<td>19</td>
<td><code>debug(&quot;target = &quot;+ p.target)</code></td>
</tr>
<tr>
<td>20</td>
<td><code>debug(&quot;source = &quot;+ p.source.decode(&quot;utf-8&quot;))</code></td>
</tr>
<tr>
<td>21</td>
<td><code>debug(&quot;target = &quot;+ p.target.decode(&quot;utf-8&quot;))</code></td>
</tr>
<tr>
<td>22</td>
<td><code>debug(&quot;target = %s&quot; % p.target)</code></td>
</tr>
<tr>
<td>23</td>
<td><code>debug(&quot;source = %s&quot; % p.source)</code></td>
</tr>
<tr>
<td>24</td>
<td><code>if(y != NULL)</code></td>
</tr>
<tr>
<td>25</td>
<td><code>if(NULL != y)</code></td>
</tr>
</tbody>
</table>

**same**

**debain**

https://board.net/p/PatchPorting
Call for comments and collaboration

- Everything is MIT licensed
  - Patch crawler: node.js
  - The patcher: python
- Add patches to the security tracker
- Clearly identify the CVE id in your security quilt patch header.