HardBlare: a Hardware-Assisted Approach for Dynamic Information Flow Tracking
Mounir Nasr Allah, Guillaume Hiet, Muhammad Abdul Wahab, Pascal Cotret, Guy Gogniat, Vianney Lapotre

To cite this version:

HAL Id: hal-01311032
https://hal-centralesupelec.archives-ouvertes.fr/hal-01311032
Submitted on 23 Jun 2016

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L’archive ouverte pluridisciplinaire HAL, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d’enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.
Example code | Tag initialization | Tag propagation | Tag check |
---|---|---|---|
\(p = 3;\) | \(p \leftarrow \text{public}\) | \(s = 42;\) | \(s \leftarrow p + s = 45\) |
\(a = 42;\) | \(a \leftarrow \text{secret}\) | \(x = p * a;\) | \(x \leftarrow p + a = 37\) | if \((x != \text{public})\) \(\Rightarrow\) raise interruption

During the compilation phase, a static analysis is done on the LLVM intermediate representation produced from the source code, and propagated to the ARM backend for the machine code generation. The result of static analysis gives a list of dependencies between information containers (e.g., registers, memory spaces...) for every basic blocks which are stored on a dedicated section in a ELF File. During run-time, the Program Trace Macrocell (PTM) generates a trace containing the address for each committed instruction modifying the PC value. Annotations related to the basic block identified by its address, given by the trace, are processed by the coprocessor to propagate tags.

State of the art

<table>
<thead>
<tr>
<th>Advantages</th>
<th>Disadvantages</th>
</tr>
</thead>
<tbody>
<tr>
<td>Flexible security policies</td>
<td>Overhead (from 300% to 3700%)</td>
</tr>
<tr>
<td>Multiple attacks detected</td>
<td>Fixed Security policies</td>
</tr>
<tr>
<td>Low overhead (&lt;10%)</td>
<td>Invasive modifications</td>
</tr>
<tr>
<td>Invasive modifications</td>
<td>Few security policies</td>
</tr>
<tr>
<td>Low overhead (&lt;10%)</td>
<td>Few modifications to CPU Energy consumption ((\times 2))</td>
</tr>
<tr>
<td>Flexible security policies</td>
<td>Communication between CPU and DIFT</td>
</tr>
<tr>
<td>Low overhead (&lt;10%)</td>
<td>CPU not modified</td>
</tr>
<tr>
<td>CPU not modified</td>
<td>Coprocessor</td>
</tr>
</tbody>
</table>

Main Contributions at a Glance

- Hardware-assisted DIFT system with limited time overheads.
- Approach based on a non-modified CPU with a standard Linux and generic binaries.
- Could be implemented by industrial partners in medium-term.
- HARDWARE ASSISTED DIFT system is HARDWARE ASSISTED DIFT system.
- HARDWARE ASSISTED DIFT system is HARDWARE ASSISTED DIFT system.
- HARDWARE ASSISTED DIFT system is HARDWARE ASSISTED DIFT system.

Some References