Securing Software Ecosystem Architectures: Challenges and Opportunities

Walt Scacchi and Thomas A. Alspaugh
Overview

- Securing Open Architecture Command & Control Systems
- Improve security of OA software ecosystem processes
- Visually modeling (mapping) OA ecosystems
- Using architectural description language (ADL) tools and analysis techniques to model and identify OA ecosystem security vulnerabilities
<table>
<thead>
<tr>
<th>Software supply chain processes</th>
<th>Common ecosystem security problems for each process</th>
<th>Example threats to the supply chain(^1)</th>
<th>Software supply chain security defenses(^{1,12,13})</th>
<th>Further defenses enabled by explicit OA(^{1,2})</th>
<th>Challenges in progressing to the next software supply chain process(^{1,12,13})</th>
</tr>
</thead>
<tbody>
<tr>
<td>Component sourcing</td>
<td>Untrusted or corrupt software producers</td>
<td>Counterfeit repositories for sourced components(^7)</td>
<td>Independent validation of components and interconnections</td>
<td>Validation of provenance in component supply chains</td>
<td>Independent validation of components sourced from unknown providers</td>
</tr>
<tr>
<td>Continuous integration and release(^{1,2})</td>
<td>Infected or corrupt component producers</td>
<td>Counterfeit components in commercial products(^7)</td>
<td>Component provenance tracking and analysis</td>
<td>Re-creation of multiversion builds to validate software product integrity</td>
<td>Collecting and passing on provenances, taggants, and other information for evaluation</td>
</tr>
<tr>
<td>Delivery and deployment(^{1,2})</td>
<td>False flag download sites, bait-and-switch downloads</td>
<td>Counterfeit software with false certificates(^6)</td>
<td>Installation of components into security encapsulations</td>
<td>Maps of installed software configurations to guide defenses</td>
<td>Producing and delivering trustable, verifiable packages</td>
</tr>
<tr>
<td>Continuous evolution(^{11,12})</td>
<td>Outdated component versions with known vulnerabilities</td>
<td>Update mechanism hijacked to enable remote control and data exfiltration(^5)</td>
<td>All of the defenses listed</td>
<td>Repositories listed, deployment of multiversion releases</td>
<td>Maintaining security, trust, provenance, and other requirements as ecosystem and configurations evolve</td>
</tr>
</tbody>
</table>
Software Ecosystem Supply Network

Producers
- Mozilla Foundation
- Gnome Foundation
- AbiSource Community
- Corel Counterfeiter

Components (from repositories)
- Thunderbird Email Client
- Firefox Browser
- Gnome Evolution
- AbiWord WordProcessor
- WordPerfect WordProcessor (suspect)

Integrators
- Independent Software Vendors or Government Contractors
- Inhouse System Integrators or Consultants

OA Systems
- sys. rights, obligations

System Consumers
OA Ecosystem Product Line Supply Chains
An OA Ecosystem Product Supply Chain
OA Ecosystem Instance Configuration

Firefox

Gnome Evolution email, calendar

AbiWord

Red Hat / Fedora Linux
Alternative OA Ecosystem Product Supply Chain
Abstract OA Ecosystem Map

Concrete OA Configuration Map
ArchStudio 4 OA Modeling

ADL-based IP / Security License Analysis Capability

Conclusions

- Open Architecture and software ecosystem maps are useful to identify where and when security vulnerabilities occur.
- OA ecosystem maps serve as reference models.
- **OA maps**: abstract or concrete (detailed)
- OA maps reveal where and how system configurations are potentially modified during each supply chain process.
- Use Architectural Description Languages (ADLs) to specify, visually map, analyze, and update software architectures.
Acknowledgements

This research was supported by grants N00244-16-1-0004 and N00244-16-1-0053 from the Acquisition Research Program at the Naval Postgraduate School, Monterey, California. No endorsement implied.