Medical Device Security and the Curse of The Internet of Things

Thomas Engel, M.D.
Loma Linda University School of Medicine

The Internet of Things

• Almost everything has a computer in it.

The Curse of the Internet of Things

• Almost everything with a computer in it is broken.
Internet cameras have hard-coded password that can’t be changed

Cameras with vulnerable firmware are wide open to remote hacking.

1/10/18

Crypto weakness in smart LED lightbulbs exposes Wi-Fi passwords

More evidence the Internet of things treats security as an afterthought.

1/10/18

Internet-paralyzing Mirai botnet comes roaring back with new strain

Mirai devices infected in 60 hours by strain that targeted ZyXEL devices.

1/10/18
Researchers hack Philips Hue lights via a drone; IoT worm could cause city blackout

Researchers hack Philips Hue lights with a drone to show how IoT worm could take over smart lights in a city.

Android Version Adoption

<table>
<thead>
<tr>
<th>Version</th>
<th>Version</th>
<th>Android</th>
<th>Total</th>
</tr>
</thead>
<tbody>
<tr>
<td>3.5.7</td>
<td>4.6.3</td>
<td>5.0.4</td>
<td>6.0.4</td>
</tr>
<tr>
<td>4.6</td>
<td>5.0</td>
<td>6.0.4</td>
<td>6.0.4</td>
</tr>
<tr>
<td>4.4</td>
<td>5.0</td>
<td>6.0.4</td>
<td>6.0.4</td>
</tr>
<tr>
<td>5.0</td>
<td>5.0</td>
<td>6.0.4</td>
<td>6.0.4</td>
</tr>
<tr>
<td>6.0.4</td>
<td>6.0.4</td>
<td>6.0.4</td>
<td>6.0.4</td>
</tr>
</tbody>
</table>

Fiat Chrysler recalls 1.4 million cars over remote hack vulnerability

Fiat Chrysler will recall 1.4 million vehicles after hackers were able to turn off their engines and take over steering.
Alert (ICS-ALERT-13-164-01)

Medical Devices Hard-Coded Passwords

- The affected devices are manufactured by a broad range of vendors and fall into a broad range of categories including but not limited to:
  - Surgical and anesthesia devices.
  - Ventilators.
  - Drug infusion pumps.
  - External defibrillators.
  - Patient monitors.
  - Laboratory and analysis equipment.
Flaws

• Software bugs.
• Hard coded passwords.
• Lack of updates.
• Unsigned updates.
• Unnecessary services.
• Unpatched vulnerabilities in necessary services.

Attacks

• Botnets.
• Privacy and identity theft.
• Ads.
• Ransomware.
• Foothold for attacking other devices in the network.

Causes

• Low profit margins on cheap devices.
• Investor expectations on expensive devices.
• Lack of knowledge by manufacturers.
• Lack of knowledge by consumers.
The cost of making secure devices is borne by the manufacturer.

The cost of using insecure devices is borne by the consumer.

Medical Device Integration

• Place collected data into the patient’s medical record.
• Generate billing information.
• Five R’s
  • The right patient.
  • The right drug (procedure).
  • The right dose (equipment).
  • The right route (location on the patient).
  • The right time.

Device Security

• Depends upon software reliability.
Software Reliability
• Reduced bugs.
• Maintainable.
• Solved problem.
• Agile Methods.

Agile Methods
• Clarity.
• Continuous delivery.
• Test Driven Development (TDD).
• Robust design.

Clarity
• Code should be written to be easily understood by people.
  • Avoid cleveress.
  • Avoid terseness.
  • Use consistent style.
• Writing is rewriting.
• Rewriting software is refactoring.
• Can only refactor when the code is clear and covered by tests.
• Each time a programmer touches code, he/she should improve it.
Continuous Delivery

• Each feature is completed before the next feature is started.
• The opposite of Waterfall.
• Requires robust design.
• Can ship at any time!

Test Driven Development

• Each routine and module is individually tested.
• Very short cycle.
• Test can be written first.
• Entire suite of tests is run with each compilation.
• Additional tools are also used.
  • Static analyzers.
  • Memory sanitizers.
  • Thread sanitizers.
  • Performance profiling tools.

Robust Design

• Is about dependency management.
• SOLID principles.
  • Single-Responsibility principle.
  • Open-Closed principle.
  • Liskov substitution principle.
  • Interface segregation principle.
  • Dependency inversion principle.
• Can be implemented in any programming language.
Secure Software Design
• Simplicity.
• Continuous updates.
• Use best practices for secure software development.

Best Practices for Secure Software Development
• Use well tested software libraries.
• Avoid other libraries.
• Avoid interpreted environments like HTML, JavaScript, Python and VisualBasic.
• Validate and sanitize all input.
• Eliminate unnecessary services.
• Use certificate pinning.
• Use code signing.
• Enable remote updates.

Fixing Existing Products
• Most important first step is completing an external security audit.
• Shut off all unused services.
Creating a New Product

- Use a secure hardware platform.
- Use a modern programming language and development tools.
- For embedded systems, use established secure system software.
  - Internet RFC on Firmware Update Architecture.
- For PC based systems, use specially hardened versions of the operating system.
  - No great choices here.
- Plan for security and update infrastructure from the start.

---

Clean Code: A Handbook of Agile Software Craftsmanship
Robert C. Martin, 2008

Clean Architecture: A Craftsman’s Guide to Software Structure and Design
Robert C. Martin, 2017

Secure Programming Cookbook for C and C++:
Recipes for Cryptography, Authentication, Input Validation & More
John Viega and Matt Messier, 2003

Securing the Internet of Things
Shancang Li and Li Da Xu, 2017
The Open Web Application Security Project (OWASP)
https://www.owasp.org/

A Firmware Update Architecture for Internet of Things Devices
Internet Engineering Task Force (IETF)
https://tools.ietf.org/html/draft-moran-sub-architecture-00