Usability & Security by Design

A Case Study in R&D

Shamal Faily¹, John Lyle², Ivan Fléchais², Andy Simpson²
Bournemouth University¹, University of Oxford²
If we knew what we were doing, it wouldn’t be called Research.

– A. Einstein
Open-source project to get gadgets talking via the net

By Zoe Kleinman
Technology reporter, BBC News

More than 5,400 developers have downloaded a new open-source operating system designed to enable digital devices to communicate with each other.

They are now looking at ways in which Webinos could be used to connect a range of devices such as mobile phones, car stereos, heart monitors and TVs.

Webinos is a 15m euro ($18.4m; £11.8m) project supported by more than 30 organisations, including the EU.

BMW, W3C, Sony, Samsung and Telefonica are among its commercial partners.

While other operating systems that use the internet to connect devices to each other already exist, most are pre-installed and cannot be customised by individual users.

Free for all

Technical co-ordinator Nick Allott told the BBC Webinos was designed to provide an alternative to proprietary systems developed by Apple, Google and Microsoft.
Context of use

- **goals**
- **user**
- **task**
- **equipment**
- **environment**

intended outcome

**webinos context of use concepts**

Personas

- **Activity Scenarios**
- **Environment**
- **Asset**

**Characteristic**

Attacker

Persona

Attack Tree

- **1..***
- *****
Stolen from the WEB
Steal from storage on the device
Steal from the WEB environment on the device
Steal from the OS environment on the device

Client authentication and encryption, e.g. DRM scheme integrated in the WEB runtime
WEB application isolation within the WEB environment
OS protection of the WEB environment

Rooting style attacks
Rooted boot

Modify the WEB runtime

Leech on service backend

Service API access control

Extract service credential from the application
### Architectural Modelling & Attack Surface Measurement

#### Architectural Risk Analysis


<table>
<thead>
<tr>
<th>Obstacle</th>
<th>Requirement</th>
<th>Affected Component</th>
<th>Satisfied (Y/N)</th>
<th>Rationale</th>
</tr>
</thead>
<tbody>
<tr>
<td>Ambiguous request specification</td>
<td>Canonical request specification</td>
<td>Policy Manager</td>
<td>Y</td>
<td>Each request is enforced separately by definition. Impossible to grant access to more resources than...</td>
</tr>
<tr>
<td>Test API enabled</td>
<td>Test API disabled</td>
<td>Widget Manager</td>
<td>N</td>
<td>There are no supported means for distinguishing test APIs from those which are...</td>
</tr>
</tbody>
</table>

#### Architectural pattern

<table>
<thead>
<tr>
<th>Architectural pattern</th>
<th>DER_m</th>
<th>DER_c</th>
<th>DER_i</th>
</tr>
</thead>
<tbody>
<tr>
<td>Context Policy Management</td>
<td>1.2</td>
<td>6.1</td>
<td>28.6</td>
</tr>
<tr>
<td>NFC</td>
<td>1.4</td>
<td>4</td>
<td>39.5</td>
</tr>
<tr>
<td>PZH Authentication</td>
<td>9.6</td>
<td>6.1</td>
<td>29.7</td>
</tr>
<tr>
<td>PZP Enrolment</td>
<td>11.2</td>
<td>5</td>
<td>16.6</td>
</tr>
<tr>
<td>TV Service Discovery</td>
<td>3.7</td>
<td>4.2</td>
<td>29</td>
</tr>
<tr>
<td>Widget Processing</td>
<td>1.1</td>
<td>1</td>
<td>22.9</td>
</tr>
<tr>
<td>Widget Rendering</td>
<td>2.8</td>
<td>2</td>
<td>28.6</td>
</tr>
</tbody>
</table>

#### Attacker, Motives, Capabilities (Value)

<table>
<thead>
<tr>
<th>Attacker</th>
<th>Motives</th>
<th>Capabilities (Value)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Ethan</td>
<td>System resource theft</td>
<td>Methods (Med), Software (Med), Technology (Med)</td>
</tr>
</tbody>
</table>

#### Obstacle Origin

<table>
<thead>
<tr>
<th>Location</th>
<th>Exploit</th>
</tr>
</thead>
<tbody>
<tr>
<td>Locate and Exploit Test APIs</td>
<td>CAPEC-121</td>
</tr>
</tbody>
</table>

#### Target, Exploit

<table>
<thead>
<tr>
<th>Target</th>
<th>Exploit</th>
</tr>
</thead>
<tbody>
<tr>
<td>Application Data</td>
<td>Access Request</td>
</tr>
</tbody>
</table>
User research is not easy

Usability is not a priority
Peter is generally law-abiding and rule following. He is prudent with his purchases, and has an expectation of some security around financial transactions.
Sustaining adoption requires creativity

Lessons Learned

- Learn to work with imperfect data and expertise
- Security and usability design: better together
- Make time for designing security
- Designing for security is not just a process
Thank you for listening!

- Any questions?