Sony Ericsson Position Paper for the W3C Workshop “Security for Access to Device APIs from the Web”

10-11 December 2008, London

Marcus Liwell, Technology Area Group Leader – Web and Java Security

Claes Nilsson, Technology Area Group Leader – Web Browsing

Abstract

Security is a key issue when device APIs are opened up to web applications. We need to protect the user’s privacy, prohibit unwanted costs and we also need to protect the operator, content provider and device vendor from user’s abuse.

The security model should to the extent possible be build on existing standards such as a domain and signing concept at installation similar to MIDP for Widgets and TLS/SSL and XMLDsig for Web Applications. This should be combined with a mechanism to dynamically define device API permission policies for each domain.

In addition to a reliable security model we have to focus on usability. Irritating and un-logical pop-ups asking for user permissions when Widgets or Web Applications are running should be avoided. Instead it is generally better to ask for user permissions when Widgets are installed or Web Applications are loaded into the browser.

As new types of web applications constantly are evolving it is also essential that the security model allows for “over the air” dynamic extensions to the device APIs.

Security problems to solve

As the web is becoming an application execution environment script access to local device functionality is getting more and more important. There are numerous use cases involving APIs to the device features for location, call handling, messaging, camera etc.

However, security is an issue when and we can see two main groups of security problems to solve:

- We need to protect the user!
  This means that we have to protect the user's privacy, prohibit actions that cause costs and prevent technical/usability problems. Some examples of security threats are:
    - Unauthorized access to user's private data.
    - Generation of unwanted data traffic.
    - Viruses and other malicious attacks

- We need to protect the operator, device vendor, content provider etc from user's abuse. The control of the device needs to be kept!
  Some threats are:
    - Access to the SIM's operator sensitive data
    - Manipulation of the DRM protection for stored media
    - Viruses and other malicious attacks that drive customer service costs

Differences of the security approaches in a browser and in a widgets environment

The terms used by the industry for web based applications are not completely consistent but most people agree that there are two main types of web based applications:

- Widgets: Lightweight applications created with web technology that are downloaded and installed in the device. The device executes the Widgets by a Widget execution environment with a web rendering engine as base.
- Web Applications: Applications that are executed within the "normal web browser" context of a device.

When it comes to security there is a fundamental difference between Widgets and Web Applications as the security model for Widgets can be connected with the Widget package installation process. This can be compared with installation of Java MIDlets.

For Web Applications the security model is more complicated as Web Applications are dynamically invoked by the browser as any other web-pages by a URL and there is no installation process.

Even though the security models differ between Widgets and Web Application we should strive to achieve security solutions that are consistent between Widgets and Web

Applications from a usability point of view.

Differences of the security approaches in PCs and mobile devices

Currently the PC is perceived as a much more open environment than a mobile device and the average PC user is aware of and accepts security problems. Thereby firewalls and virus protection are generally used in PCs. Users of PCs are also used to install and uninstall programs and have a good overview of running applications.

For mobile devices the user's awareness of security issues are generally low. Mobile devices are considered "secure". However, as mobile devices are more and more based on open platforms we believe that user's awareness of security issues will increase. Our goal should be to create a mobile security framework that is both reliable and "user friendly" and we must strive to preserve the user's view that mobile devices are trusted and secure.

Potential solutions

Widgets

A protection domain and signing concept at installation similar to MIDP would be sufficient for Widgets. Signing authenticates the Widget creator and verifies the integrity of the Widget and is independent of the delivery solution, i.e. the server the Widget is fetched from.

We also need a mechanism to dynamically define API permission policies for the different domains, i.e. the APIs that are available for each domain and the level of user interaction required to permit access to the APIs.

From a usability point of view the number of pop ups should be limited. We see a problem with the current MIDP standard that pop ups are invoked in any situation. From a user point of view this may be confusing and irritating as the pop-ups often are difficult to relate to the current user context. When user dialogues are initiated the user must be aware of what's happening and be guided to allow or not allow access. This goal can for example be achieved by a declaration in the Widget manifest (metadata about the Widget) of the APIs that are used by the Widget. User dialogues asking for permissions to use device functionality can then be executed at installation time and the number of pop-ups during execution of the Widget can be limited.

Web Applications

For Web Applications we don't have an installation process and the application model is different compared with the Widget application model. For Widgets the executable content is contained and possible signed within a single package. The Widget content is persistently stored in the device. For Web Applications all the content resides behind several URLs and the content is typically dynamically generated. The conclusion is that the security model must be different.

We promote a security model based on existing technologies such as:

- Transport layer security (TLS/SSL): Authenticates the server from which the page was loaded and achieves integrity protection during the transport from server to client. However, this does not prevent the page from being manipulated before the page was stored in the server or after it is stored in the client.
- Digital signing of page or parts of the page by XMLDsig. This makes it possible to authenticate the content creator and perform integrity check on the whole or parts of the page.

A combination of transport layer security and digital signing gives us a flexible security solution which also can ensure end to end security. This means that cross server applications will not cause illegal use of sensitive APIs by a Web application hosted on a non trusted server when it is accessed through a trusted server.

All security levels can be combined with a method to define API policies. For example:

- No secure transport and signing: Only "harmless" APIs can be accessed (battery level, beep, vibration etc)
- Secure transport: Medium sensitive API can be accessed (Positioning, Camera, Call Handling, Messaging etc)
- Secure transport and signing: Highly sensitive APIs can be accessed (SIM, DRM etc)

Usability must be in focus also for Web Applications. Irritating pop-us should avoided and it should be considered if it is better to ask for user permissions when the page is loaded for the first time instead of using pop-ups when the web application is executed.

Dynamical extensions to device APIs

New types of web applications are constantly evolving and it is impossible to predict which device APIs that are needed in the future. It is therefore essential that the set of device APIs is extensible over the air and that the security model is capable of handling dynamic extensions of device APIs. This means that also permission policies must be dynamically extendable.

In addition we need a different set of protection domains for permitting extensions to device APIs than for access to the device APIs.

About Sony Ericsson

Having sold over 100 million phones in 2007, Sony Ericsson is currently one of the five largest mobile phone manufacturers in the world. An important industry player operating in over 80 countries, our phones, accessories and PC cards are synonymous with innovation and style. With R&D sites in Europe, Japan, China, India and North America, diversity is one of the core strengths of the company. Sony Ericsson was established as a 50:50 joint venture by Sony and Ericsson in October 2001, with global corporate functions located in London. For more information about Sony Ericsson please visit http://www.sonyericsson.com