Security Assurance for Web Device APIs

Maritza Johnson and Steven M. Bellovin
http://www.cs.columbia.edu/~{maritzaj,smb}
Columbia University

December 9, 2008

The Problem

- Web servers want access to very sensitive devices
- There is a history of trouble in this space
- We need a *high-assurance* guarantee that the implementation is correct
- We need a *high-assurance* guarantee that the user understands what is happening
- What are the design principles for any API spec, given that we cannot rely on bug-free code or bug-free users?

Usability Principles

The Problem
Usability Principles
Isolation Principles
Device Categories
High-Assurance Implementation
Failures. . .

1. *The user must explicitly authorize any and all accesses to devices*

Usability Principles

1. *The user must explicitly authorize any and all accesses to devices*
   Permission request cannot be generated implicitly; users ignore warnings and click through pop-up boxes

Usability Principles

1. *The user must explicitly authorize any and all accesses to devices*
   Permission request cannot be generated implicitly; users ignore warnings and click through pop-up boxes
2. *The user must understand the consequences of any change*

Usability Principles

1. *The user must explicitly authorize any and all accesses to devices*
   Permission request cannot be generated implicitly; users ignore warnings and click through pop-up boxes
2. *The user must understand the consequences of any change*
   Stream receive must be at same host as permission request; requests cannot come from IFRAMEs unless the URLs match

Usability Principles

1. *The user must explicitly authorize any and all accesses to devices*
   Permission request cannot be generated implicitly; users ignore warnings and click through pop-up boxes
2. *The user must understand the consequences of any change*
   Stream receive must be at same host as permission request; requests cannot come from IFRAMEs unless the URLs match
3. *The state of the system must be visible at all times*

Usability Principles

1. *The user must explicitly authorize any and all accesses to devices*
   Permission request cannot be generated implicitly; users ignore warnings and click through pop-up boxes
2. *The user must understand the consequences of any change*
   Stream receive must be at same host as permission request; requests cannot come from IFRAMEs unless the URLs match
3. *The state of the system must be visible at all times*
   User must see what access is authorized

Isolation Principles

- Must give implementors (and users) confidence that the system will behave properly
- Secure across software upgrades
- Secure against new, unforeseen devices
- System must "fail secure"

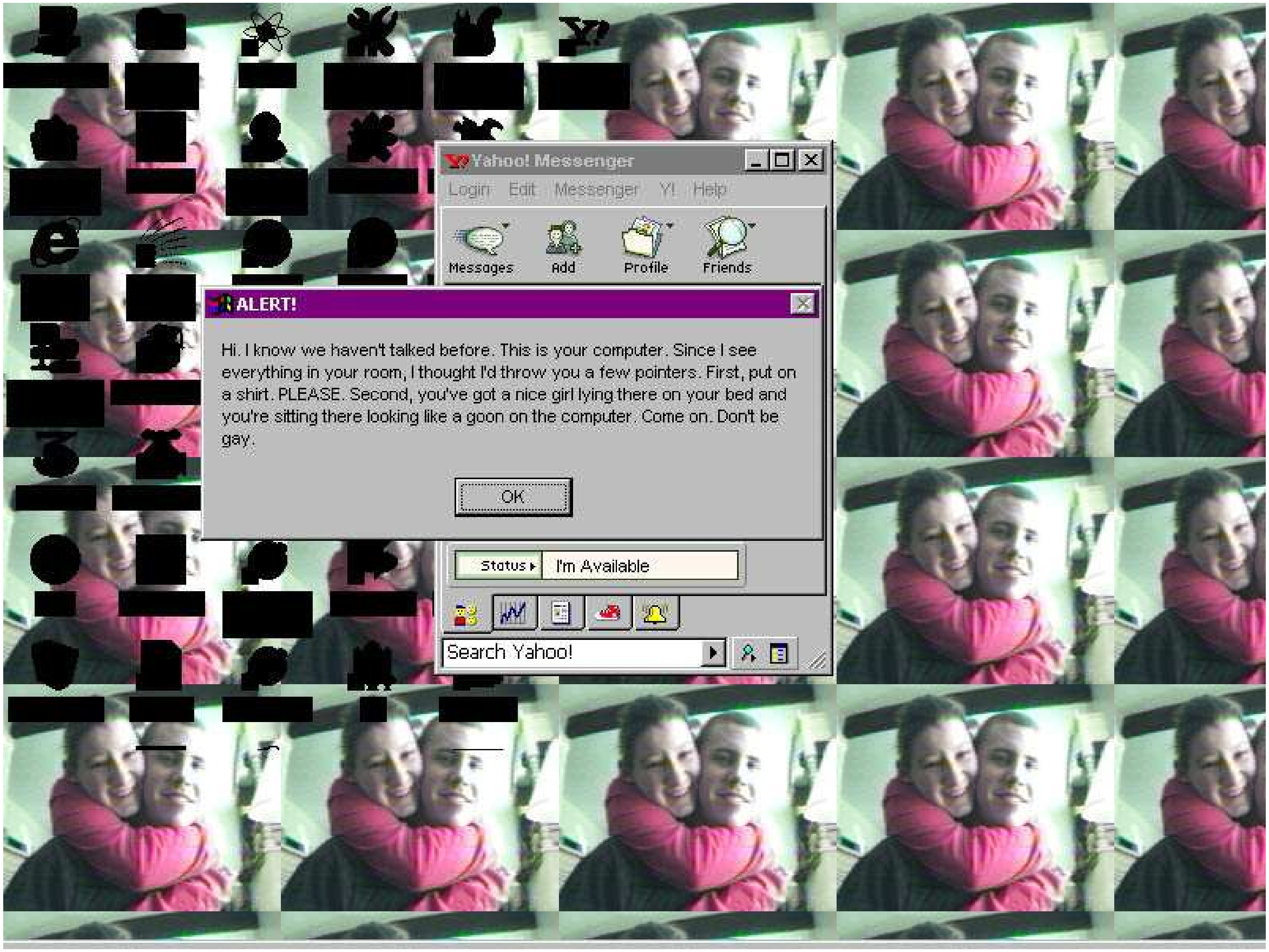
Device Categories

- Devise categories: PHYSICAL WORLD, PRIVACY, etc.
- Assign each device to a category
- New devices *must* be in a category to be used; forces a decision
- Grant or withhold permission based on at least category; simplifies user decision process

High-Assurance Implementation

- (Unix-style solution; Windows is similar)
- Create a group for each category; assign devices to the proper group with permission 060 (group read/write; no others)
- To enable a ⟨category,device⟩, create a setgid program executable by only that user but setgid to the category's group
- No page interpretation failure can access an unauthorized device (though erroneous web pages can)

Failures. . .



Hi. I know we haven't talked before. This is your computer. Since I see everything in your room, I thought I'd throw you a few pointers. First, put on a shirt. PLEASE. Second, you've got a nice girl lying there on your bed and you're sitting there looking like a goon on the computer. Come on. Don't be gay.