# The Web of Confusion

Douglas Crockford

Yahoo! Inc.

http://crockford.com/codecamp/confusion.ppt

# Web 2.0 Security and Privacy

The problems started in 1995.

We have made no progress on the fundamental problems since then.

# Will the web ever reach the Threshold of Goodenoughness?

+ Discovery of vulnerabilities leads to corrections.
+ If the rate at which correcting vulnerabilities introduces new vulnerabilities, eventually goodenoughness should be achieved.
- Adding new features tends to introduce vulnerabilities at a higher rate: Unintended consequences.
- If the fundamental assumptions are faulty, incremental correction never converges onto goodenoughness.

We are compiling an evergrowing corpus of hazards.

# OAuth

An open protocol to allow secure API authentication in a simple and standard method from de
applications.

« TripIt announces API, secured with OAuth  An update on the OAuth session fixation vulnerability »

## ACKNOWLEDGEMENT OF THE OAUTH SECURITY ISSUE

I wanted to acknowledge that we are aware of a security threat first reported on by CNET that affects the OAuth protocol.

There have been no known exploits so far and for the past several days the OAuth community has been coordinating a response with as many known providers as possible to help them understand the threat and deploy whatever mitigating factors they can.

We'd like to publicly show our appreciation for Twitter's role in helping to minimize premature publicity of this threat, even at its own expense, taking the heat as if it was their own issue in order to allow other companies to address this threat.

# Perfection is not an option.

It is unreasonable to require developers to have an adequate understanding of the current model.

Is the web too big to fail?

The web came closer to getting it right than everything else.

But first: What goes wrong?

# The Standard Mistake

"We will add security in 2.0."

# The Itty Bitty -ity Committee

Quality Modularity Reliability
Maintainability Security

# Confusion of Cryptography and Security.

Digital Living Room

# Confusion of Identity and Authority.

# Blame the Victim

# Confusion of Interest

# Confusion of Interest

System Mode

Computer

# Confusion of Interest

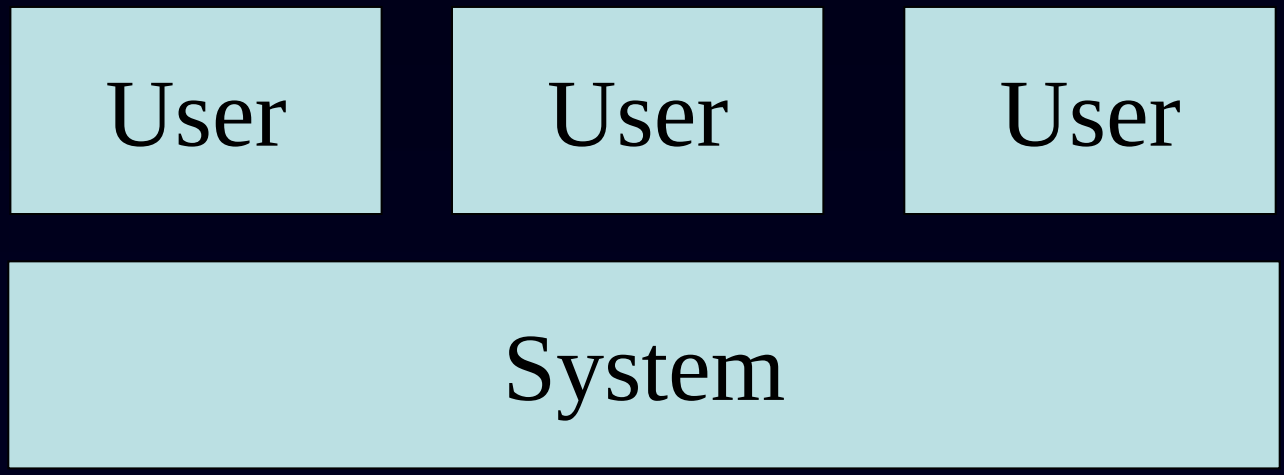
System Mode

User

System

# Confusion of Interest

System Mode



# Confusion of Interest

System Mode

CP/M MS-DOS MacOS Windows

The system cannot distinguish between the interests of the user and the interests of the program.

This mostly works when software is expensive and intentionally installed.

It is not unusual for the purpose or use or scope of software to change over its life. Rarely are the security properties of software systems reexamined in the context of new or evolving missions. This leads to insecure systems.

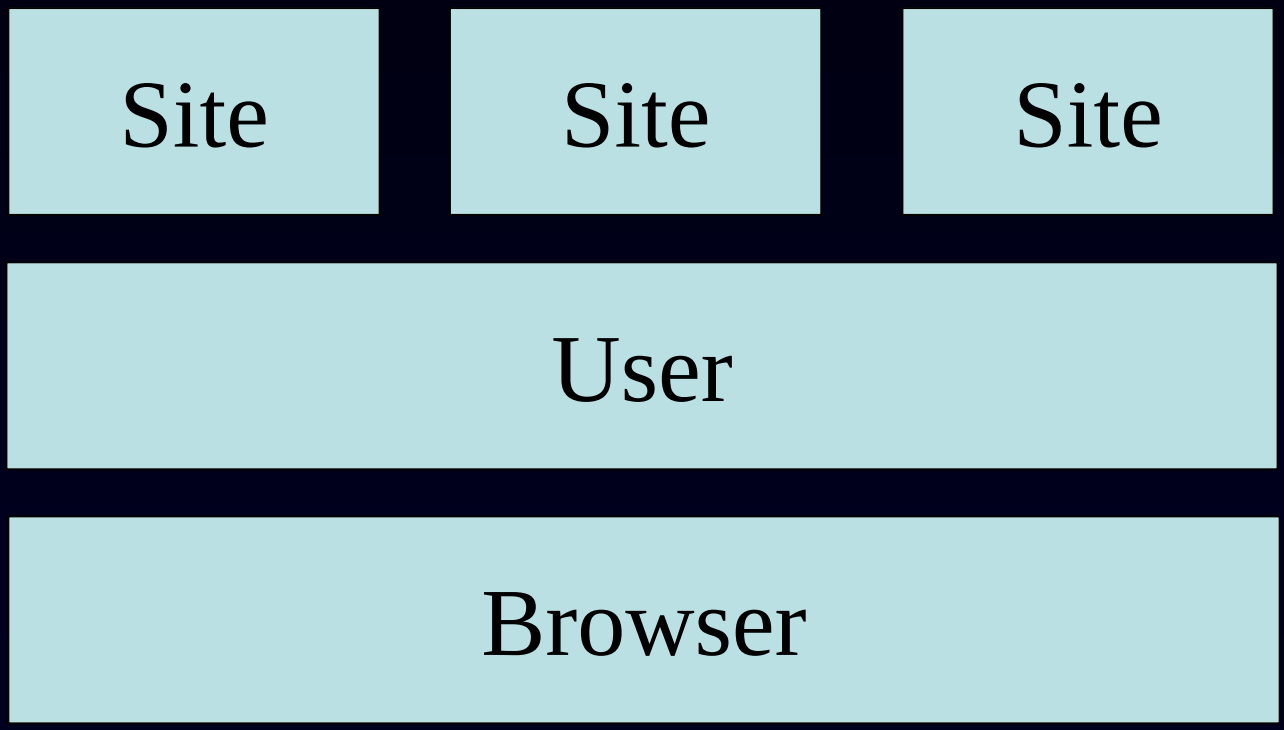
On the web we have casual, promiscuous, automatic, unintentional installation of programs.

The interests of the user and of the program must be distinguished.

The browser successfully distinguishes the interests of the user and the interests of the program.

# Confusion of Interest

The browser is a significant improvement, able to distinguish the interests of users and sites (usually).



But within a page,
interests are confused.

An ad or a widget or an Ajax library gets the same rights as the site's own scripts.

# Turducken

# This is not a Web 2.0 problem.

All of these problems came with
Netscape 2 in 1995.

# We are mashing things up.

There are many more interested parties represented in the page.

A mashup is a self-inflicted XSS attack.

(Advertising is a mashup.)

JavaScript got close
to getting it right.

A secure dialect is obtainable.
ADsafe and Caja leading the way.


A system for safe web advertising.

**http://www.ADsafe.org/**

# ADsafe

- ADsafe is a JavaScript subset that adds capability discipline by deleting features that cause capability leakage.
- No global variables or functions may be defined.
- No global variables or functions can be accessed except the **`ADSAFE`** object.
- These words cannot be used: **`apply arguments call callee caller constructor eval prototype unwatch valueOf watch`**
- Words starting with _ cannot be used.
- Use of the **`[]`** subscript operator is restricted.

# ADsafe DOM Interface

- Light weight.
- Query-oriented.
- Scope of queries is strictly limited to the contents of a widget's **`<div>`**.
- Guest code cannot get direct access to any DOM node.

# The DOM is much less close

- But the Ajax libraries are converging on a much better API.
- We need to replace the DOM with something that is more portable, more rational, more modular, and safer.
- We need to replace the DOM with something that is less complicated, less exceptional, less grotesque.

# W3C is moving in the opposite direction

HTML5 needs to be reset.

Or W3C needs to be abolished.

# We need a new security model: Object Capabilities.

*Robust Composition*, Mark Miller

http://erights.org/talks/thesis/

Cooperation under mutual suspicion.

We have gone as far as we can
go on luck and good intentions.

We need, at very long last,
to get it right.

Doing this will be very hard.

Not doing this will be even harder.