

Secure Web Browsing with the OP web browser

Represented by:
Han Zhong

Paper written by:
Chris Grier, Shuo Tang, and Samuel T.King
Department of Computer Science
University of Illinois at Urbana-Champaign

Summary of the paper

This paper describes a new browser **architecture** which aims on **security** of browser and **flexibility** of developing plug-in, and also the details of implementing the OP.

Background:

The current popular web browsers have similar architectures and security policies (same-origin). This caused many **vulnerabilities**.



93



29



74



9

Comments 1

Five subsystems form OP's Architecture (*appreciative*)

- * Main IDEA : **Isolation**
- * UI, Web page, Storage, Network, Browser kernel
- * Each subsystems run in its own OS-level process.
- * OS-level sandboxing on each subsystems.
- * Browser kernel is the main process which manages all other subsystems. All messages are passed through and checked by the kernel, deny any messages that violate the access control policy (sandboxing).

Comment 1 (continued)

- * Web page subsystem is broken into several (plug-in engines).
 - * Each new web page contains several process (HTML engine, JavaScript interpreter, plug-ins, etc.).
 - * HTML engine interact with JS interpreter and plug-in engines.
- * UI subsystem is distinct from web page subsystem
 - * A layer for isolating malicious content,
 - * and access **file-system** from other components, because this is the **only** component has unrestricted access to file-system.

Comment 2

Two new Plug-in security policies are enforced (*appreciative*).

- ❖ More efficient (compare with per-plug-in).
- ❖ **Provider domain policy:** separate hosts of **web page** and **plug-in content**
 - ❖ Included content is isolate from including page
- ❖ **Plug-in freedom policy:** give plug-ins unlimited network access but remove interactions with other components.
 - ❖ Increase flexibility for some P2P plug-ins.
 - ❖ Also prevent leaking client information.
 - ❖ Some correctly functioned plug-ins which must interact with page will not work.

Comment 3

- ❖ The KHTML HTML parsing and rendering engine is programmed in C++ which is **not safe**.
- ❖ *“To lessen the impact of this shortcoming we still allow KHTML to mark the source domain for JavaScript code and browser plug-ins, but we check them using our **Java-based** HTML parser.”*
- ❖ OP records events and visited objects to form a dependency **graph** (used to analyze attacks)

This is critical.

Question

In comment 2, how can we determine which policy should be used for particular plug-in?