SECURE WEB BROWSING WITH THE OP WEB BROWSER

Chris Grier, Shuo Tang, and Samuel T. King

IEEE Symposium on Security and Privacy (SP 2008), pp. 402-416, 2008

Presented by: Sharvin Ragavan

Summary

- This article discusses the design and implementation of the OP web browser, which is built based on operating system design principles and flexible security policies.

Appreciation 1

- Use of sandboxing technique
  - Partition the browser into 5 subsystems, each running in an OS-level process.
    - Browser kernel, web page, UI, storage, network.
  - Web page subsystem further broken into several components, running in separate OS-level processes.
    - HTML engine, JavaScript interpreter, plugins, etc.
  - Communication between subsystems can be made simple and explicit using OS-level pipes.

Appreciation 1 (cont.)

- Fine-grained level of isolation enables browser kernel to monitor message flows between subsystems and interactions with the underlying OS.
- Kernel checks messages and denies those that violate the access control policy.
- Kernel logs all messages, which helps for post-mortem analysis in case of an attack.

Appreciation 2

- Clarity of threat model
  - Clearly explained attributes of an attack
    - Origin, ability, and target of attack
  - Understand the nature of attack before designing the strategies to mitigate these threats
  - Helps reader understand the goals of the web browser and not overrate it

Criticism

- Use of unsafe external library in the implementation is contrary to the title of the article
  - KHTML HTML parsing and rendering engine
  - "*Relying on an unsafe programming language* [C++] *for our HTML engine is problematic because we rely on the HTML engine to tag JS code and browser plugins with the proper source domain.*"
  - Authors' failed first attempt to build a HTML engine might have lead to diverging from original goals.

Question

- What is the performance level of a browser that you would accept for secure web browsing?