Quark: Browser with Formally Verified Kernel

Dongseok Jang, Zachary Tatlock, and Sorin Lerner (UC San Diego)

Security Bugs in Browser Code

Lots of Browser Features
JIT for JavaScript, HTML5, WebGL, etc., …

 Lots of Security Policies
Same Origin Policies (JS, Cookie, XHR), Browser extensions

Lead to Security Bugs
938 Vulnerabilities in IE, Firefox, Chrome (2009–2011)

Kernel-based Browser
Chrome’s Architecture (Kernel / Renderer)
Kernel ← IPC → Sandboxed Renderer

Q: What if the kernel is buggy?

Quark Architecture

Input

Quark Kernel

Output

Network

Tab Process

Kernel: verified

Sandboxed Renderer: sandboxed

Kernel Ensures Security
Small kernel enables us to apply formal verification

Formal Verification

Security Policies
Tab isolation
Cookie same domain policy
Address-bar correctness

Security Policies (142 LOC)
Safe system call sequences

\[\text{Proved to satisfy}\]

Trace Spec (268 LOC)
All system call sequences that the code can generate

\[\text{Proved to satisfy}\]

Kernel Code (859 LOC)
Message handling / access control for resources
Internal book-keeping / Written in Coq, Ocaml

Verification Efforts
Staged proofs make it easier
Prototype first, prove later
Overall 6 man months to build / verify

Performance
20% overhead for page loading with optimizations for reducing IPC messages

Usability
Usable on major sites like Gmail, Maps, Amazon, …

Project Site: http://goto.ucsd.edu/quark/