9/3/15 Privilege Separation

Scribe: Christie Dierk

September 16, 2015

1 Introduction

Privilege separation separate system into modules, each with their privilege

Least privilege give a component the smallest amount of privilege

Examples:

- browsers
- android permissions
- OS process isolation
- wireshark (network packet parser)
- SSH server
  - handshake component: password, keys
  - bytestream component: parses bytestreams (vulnerable to buffer overflows)
- Virtual Machines

How can you evaluate security improvement? (very roughly)

- evaluate how many known vulnerabilities are mitigated
- amount of code in trusted base (less is better)

2 Chromium browser's security architecture

Two modules

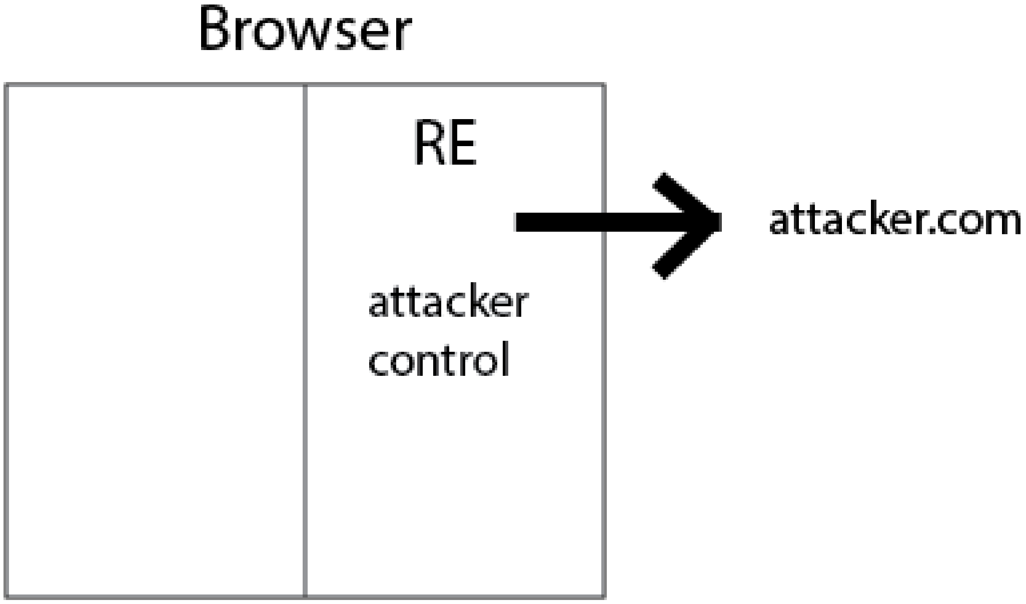
1. browser kernel (**BK**): user privilege
   - interacts with OS (FS - password, files, I/O, etc)
2. rendering engine (**RE**): (web) restricted privilege
   - no interaction with OS

Goal if **RE** gets compromised, **RE** cannot get access to OS and **BK** is not compromised

Challenge compatibility

2.1 Threat Model

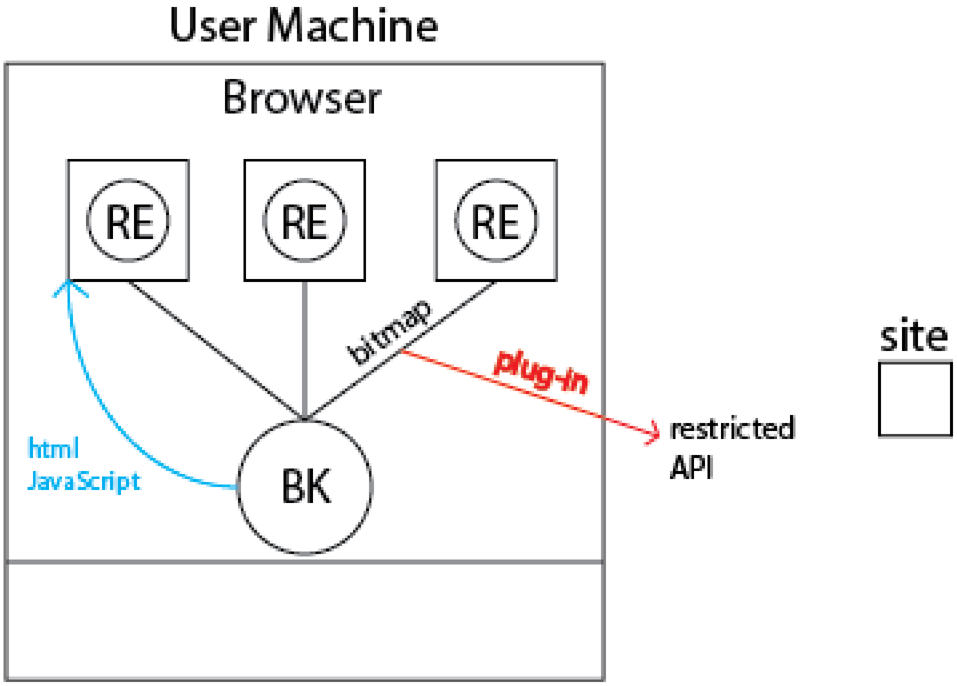
Chromium assumes the following:



Goals Should not compromise **BK** or gain access to OS

- attacker cannot install persistent malware
- attacker cannot monitor user keystrokes
- attacker cannot read from FS (file system)

2.2 Architecture



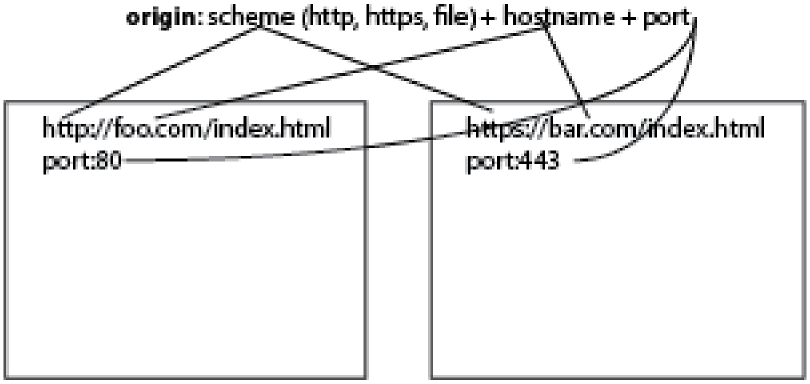
2.2.1 Rendering Engine

RE only allowed to interact with **BK** API

RE contains complexity of browser code

- parses HTML
- builds DOM
- runs JavaScript on DOM
- bitmap
- same-origin policy (isolates sites from each other)

  origin : scheme (http, https, file) + hostname + port


origin: scheme (http, https, file) + hostname + port
http://foo.com/index.html
port:80
https://bar.com/index.html
port:443

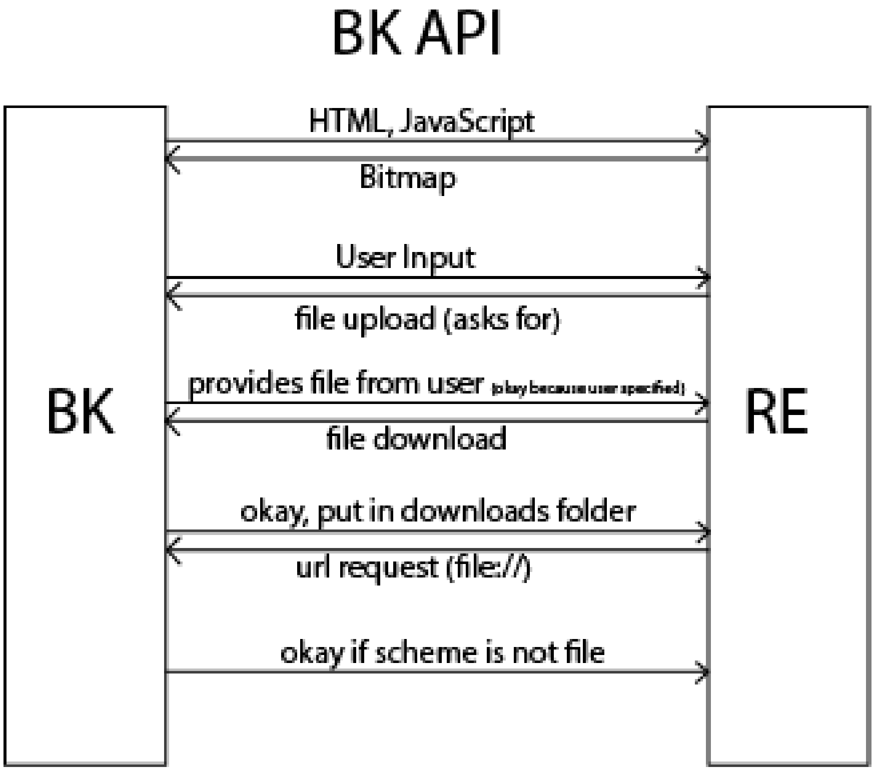
1. Each site is associated with resources
   - cookies, data, JavaScript
2. Each resource is associated with an origin (url)
3. Each script runs only on resources with same origin

Same-origin policy is a form of privilege separation!

RE implements same-origin policy

2.2.2 Browser Kernel

- cookie DB
- password DB
- window management
- network stack
- download manager


BK API
BK
RE
HTML, JavaScript
Bitmap
User Input
file upload (asks for)
provides file from user (okay because user specified)
file download
okay, put in downloads folder
url request (file://)
okay if scheme is not file

2.3 Attacks Not Prevented

- Phishing
- origin isolation → **RE** takes care of same origin
- web site vulnerabilities (XSS, CSRF)

2.4 Evaluation

- less code runs with user privilege compared to monolithic browser
- 67% of of vulnerabilities were in **RE**
  - 38 out of 87 (stopped by privilege separation)
    - 70% of **RE** vulnerabilities that gain full control