CSE 564: Threat Modeling and Browser Security

January 26, 2024

Franziska (Franzi) Roesner
franzi@cs

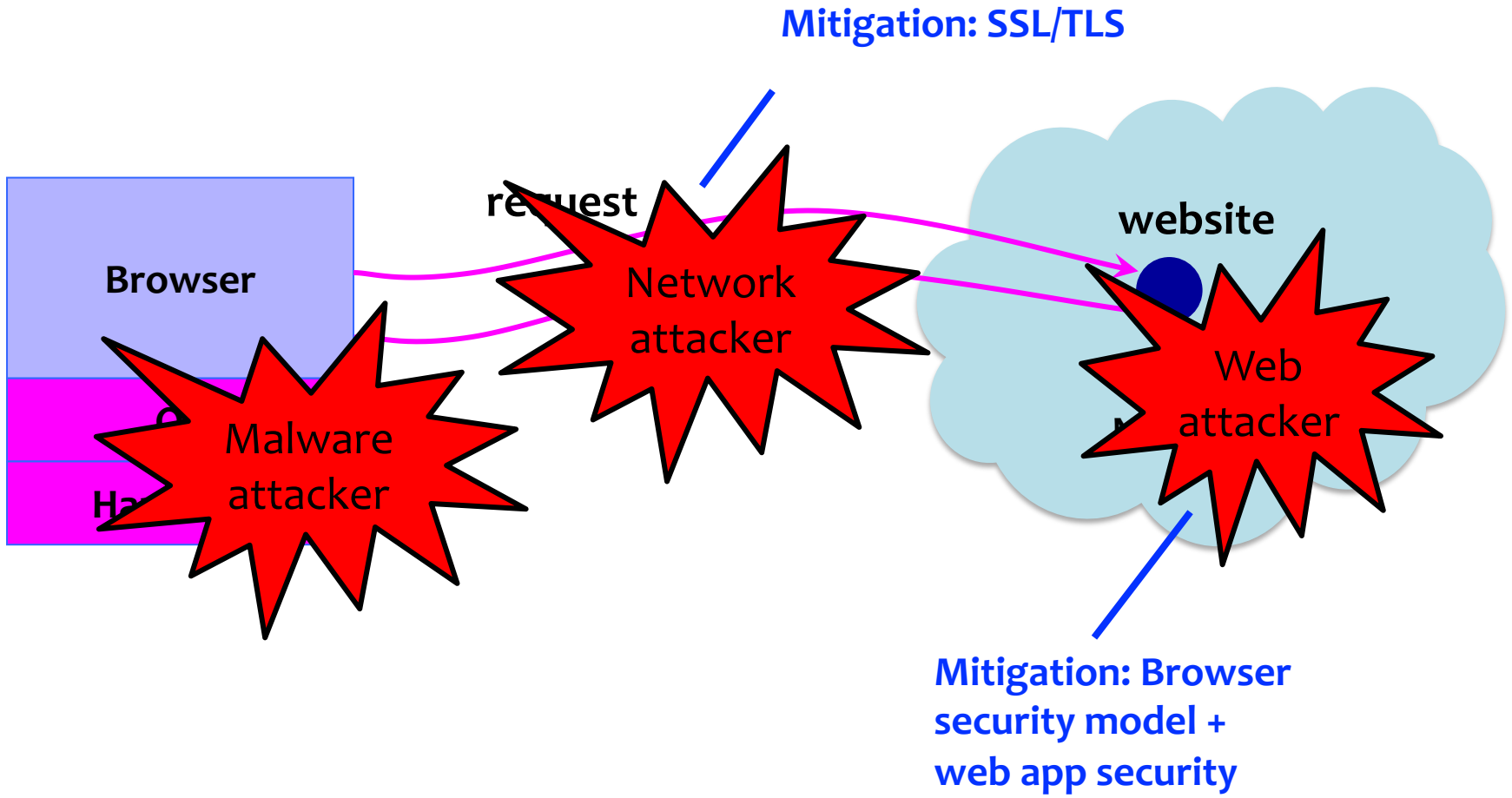
Upcoming Assignments

- 2 security reviews due on Wednesday, 1/31
  - Exercising your threat model muscle
  - Bullet points are great
  - Be thoughtful, discuss with others, but no need to spend more than an hour each (or less)
- For 1/31 readings, you may watch the video for Reis et al. paper

Threat Modeling (Security Reviews)

- Assets: What are we trying to protect? How valuable are those assets?
- Adversaries: Who might try to attack, and why?
- Vulnerabilities: How might the system be weak?
- Threats: What actions might an adversary take to exploit vulnerabilities?
- Risk: How important are assets? How likely is exploit?
- Possible Defenses
- Not "traditional" threat modeling, but important:
  - Benefits: Who might the system benefit, and how?
  - Harms: Who might the system harm, and how?

Web Security: Big Picture



Two Sides of Web Security

(1) Web browser

- Responsible for securely confining content presented by visited websites

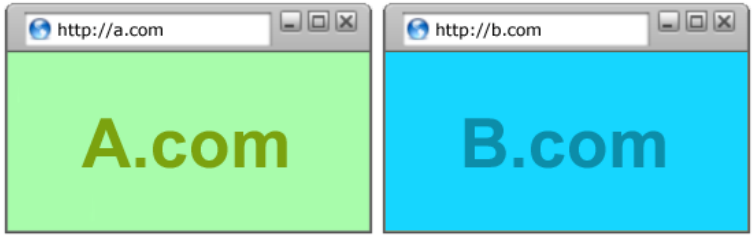
(2) Web applications

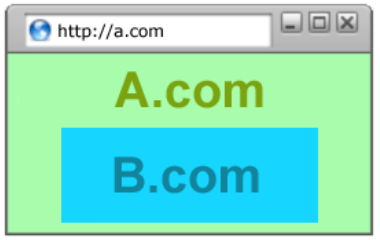
- Online merchants, banks, blogs, Google Apps …
- Mix of server-side and client-side code
  - Server-side code written in PHP, JavaScript, C++ etc.
  - Client-side code written in JavaScript (… sort of)
- Many potential bugs: XSS, XSRF, SQL injection

Browser: All of These Should Be Safe

- Safe to visit an evil website

- Safe to visit two pages
  - Simultaneously
  - Sequentially





- Safe delegation

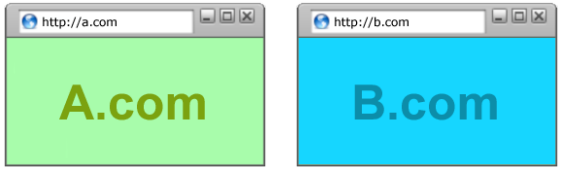
Browser Security Model

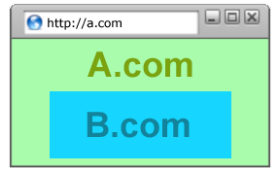
Goal 1: Protect local system from web attacker

→ Browser Sandbox

Goal 2: Protect/isolate web content from other web content

→ Same Origin Policy





Browser Sandbox

Goals: (1) Protect local system from web attacker; (2) Protect websites from each other

- E.g., safely execute JavaScript provided by a website
- No direct file access, limited access to OS, network, browser data, content from other websites
- Tabs (**newer: also iframes!**) in their own processes
- Implementation is browser and OS specific*

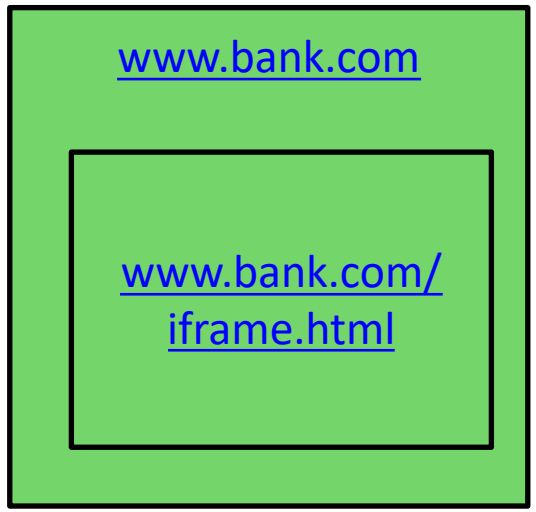
*For example, see: https://chromium.googlesource.com/chromium/src/+/master/docs/design/sandbox.md

	High-quality report with functional exploit
Sandbox escape / Memory corruption in a non-sandboxed process	$30,000

From Chrome Bug Bounty Program

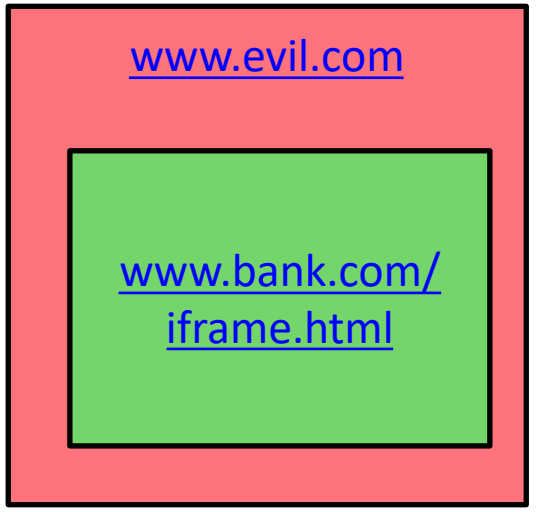
Same-Origin Policy Example

Only code from same origin can access HTML elements on another site (or in an iframe).



www.bank.com (the parent) **can** access HTML elements in the iframe (and vice versa).

```
<html> <body>
<iframe
  src="http://www.bank.com/iframe.html">
</iframe>
</body> </html>
```



www.evil.com (the parent) **cannot** access HTML elements in the iframe (and vice versa).