More Cross-Site Scripting

Recap:

Attacker: "hey, check this out! http://foo.com/?k=\langle\text{script}\rangle\ldots\""

Victim's Web Browser

Stolen cookies?

Webserver

Results: \langle\text{Script}\rangle\ldots\

Cookies:

Same-origin policy: cookies are associated with domain name

only pages within that domain, or scripts loaded by those pages can read cookies

Mitigations/Countermeasures

Within web browser:

Chrome has "XSS Auditor" security feature

"if same JS appears in HTTP request & HTML content, don't execute it!"

not just JS

\(=\) can cause false positives, though!
Within web app:
- escape all user data before including in output
- validate: whitelist vs blacklist

**Reflected vs. Stored XSS**

**Reflected**: victim must send HTTP request containing
the “malicious payload”
- must convince someone to click malicious link
- mostly mitigated by XSS auditor features

**Stored**: server stores data that is served up
to other users