Browser and network,

Browser send request to the website and then network reply back.

Web attacker (control a malicious website)& web attacker

**Web browser**: responsible for security confining web content

Goals for web security: safe to visit and evil website, safe to visit two pages at the same time, safe delegation( a.com include b.com)

Principals for web browser is ‘origins’: mandatory access control ( for origins: scheme(https/http)/ host(domain) (nyu.edu)/ port(80))

Vulnerabilities: cross-site scripting and universal scripting

Browser: basic execution model:

DOM: document object model html is a structured data, dom is objected-oriented representation of the html.

Browser sandbox: goal to safe the execute JS code provided by remoted.

Script access to DOM: protocol, domain, port

**Cookie**: path, protocol, domain. Secure, expires, http-only (no js, for security)

Used for authentication, personalization and tracking

To delete the cookie: setting expires, or change the domain,

**Web applications**:

Cross-site request forgery

Cross-site scripting !!

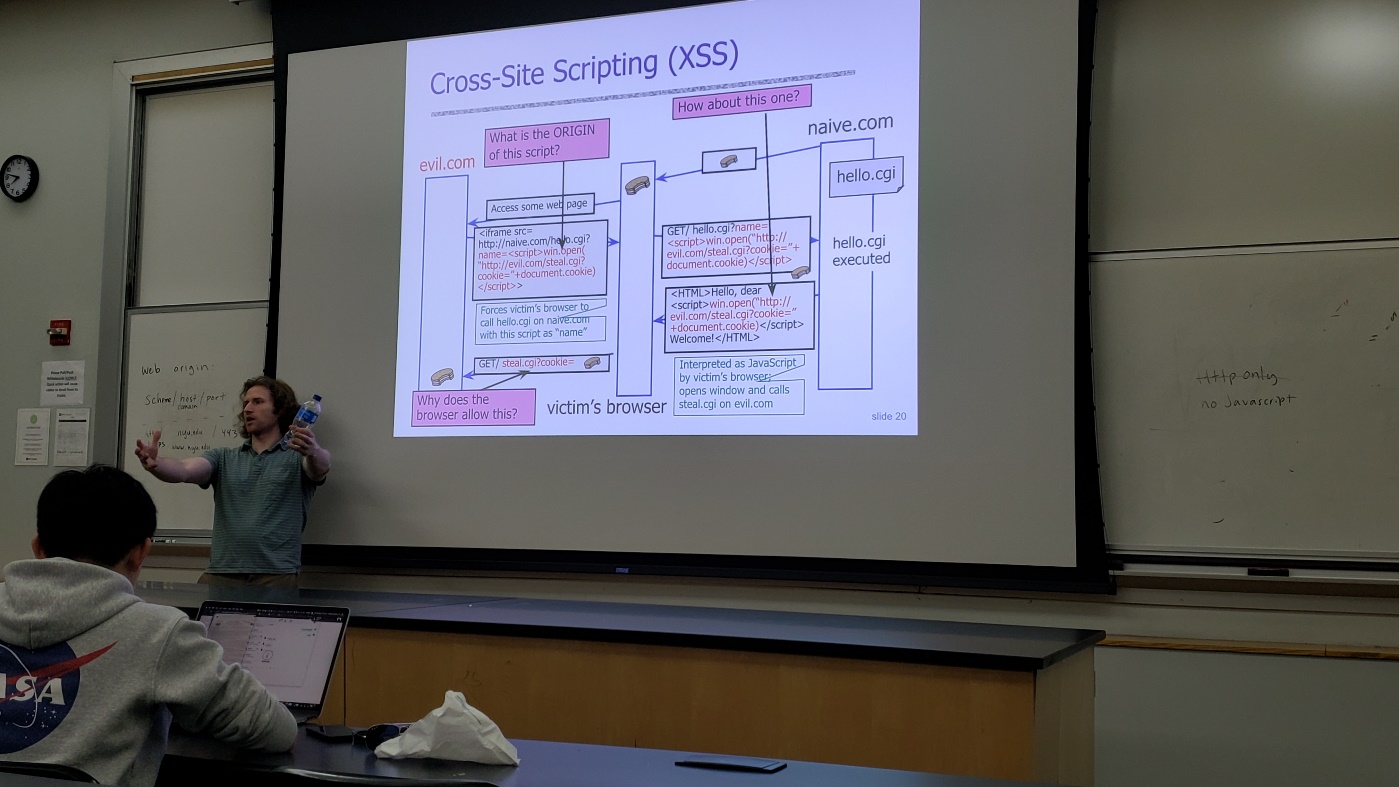
SQL injection

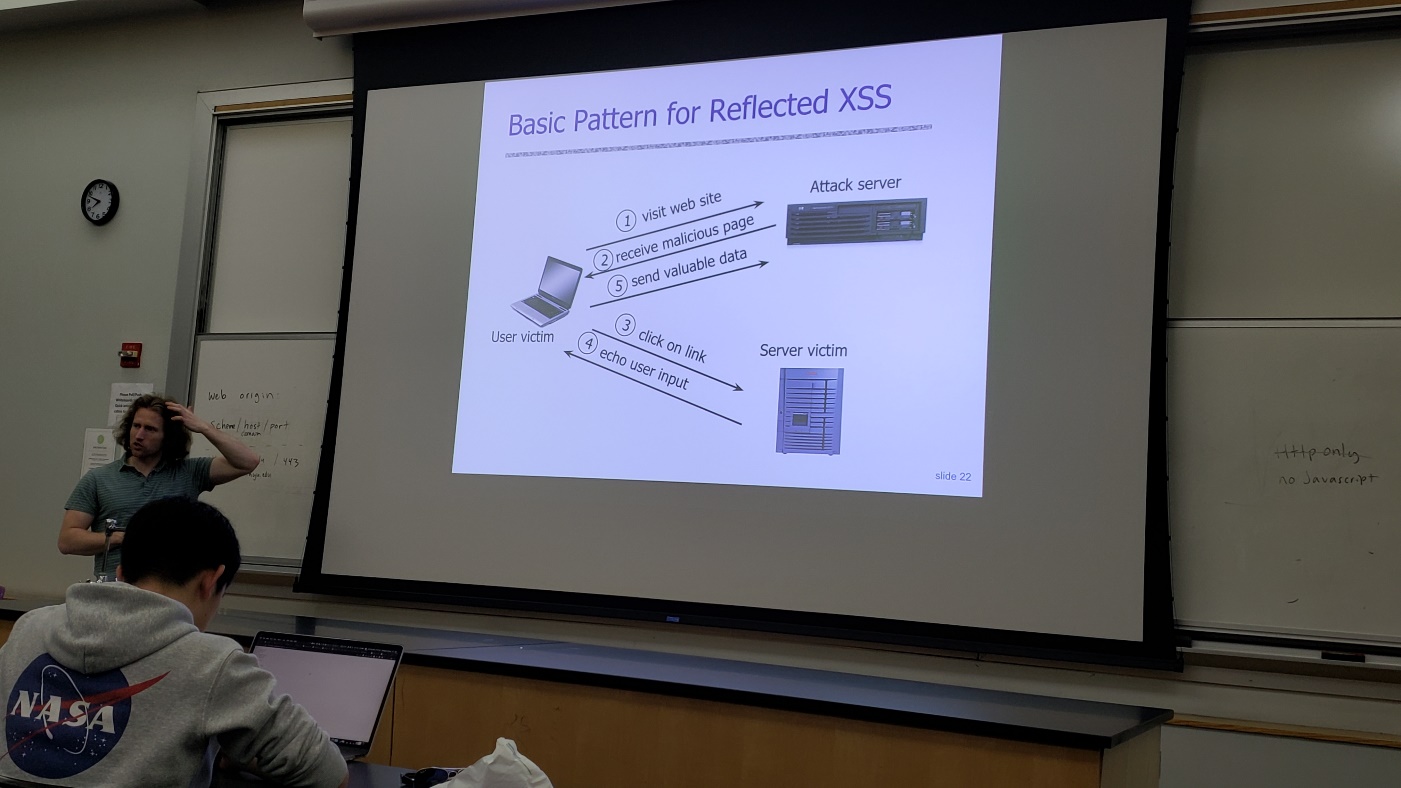
XSRF defense:

Secure validation token ( type = hidden) only works for post method. Get request never change the server.

Referrer validation

Custom HTTP header.





Prevent cross-sit scripting:

Must be preprocesses before it is used inside HTML,

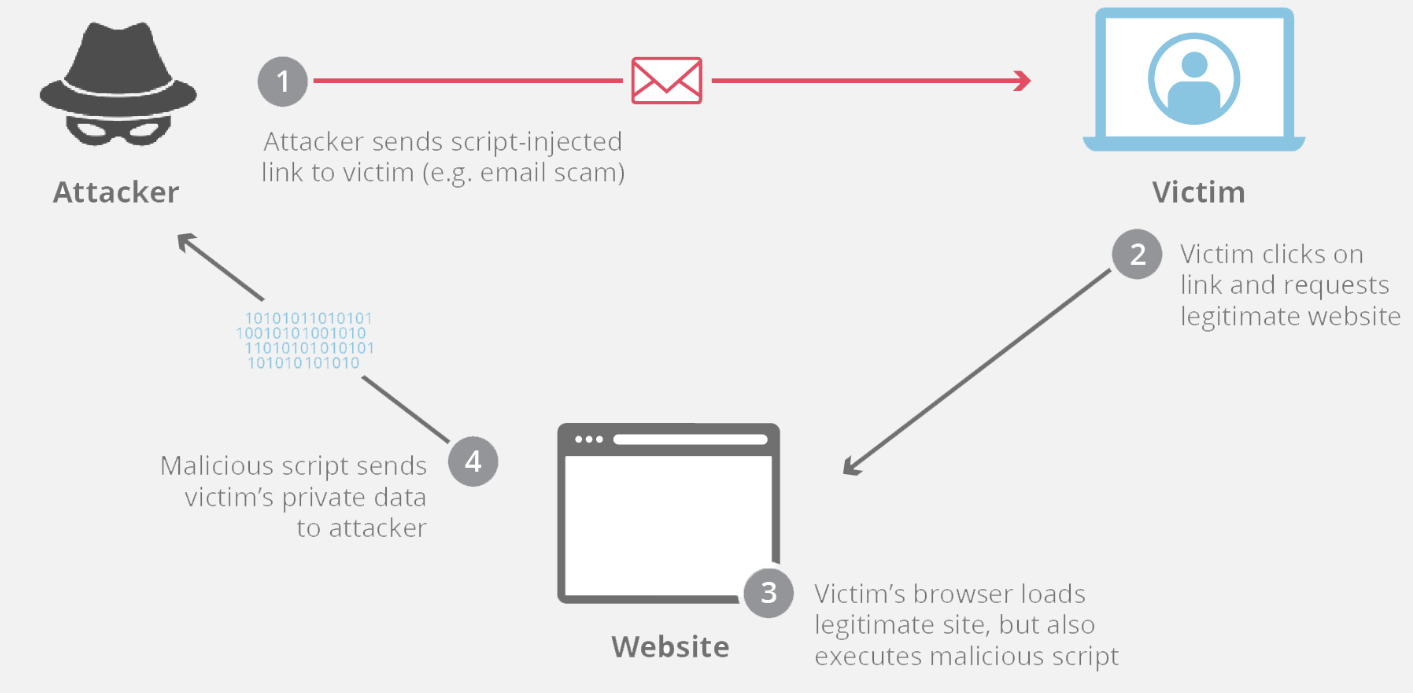
Use good escaping library

String will replace all special characters with their docs,

Evading XSS filters

Preventing injection of scripts into HTML is hard

Beware of filter evasion tricks

cross site scripting

1. Httponly
2. CSP, content security policy

Client side validation, checkout,

Prevent SQL, escape often. Validate the input, whitelist permitted characters.

Prepare statement, allow creation of static queris.