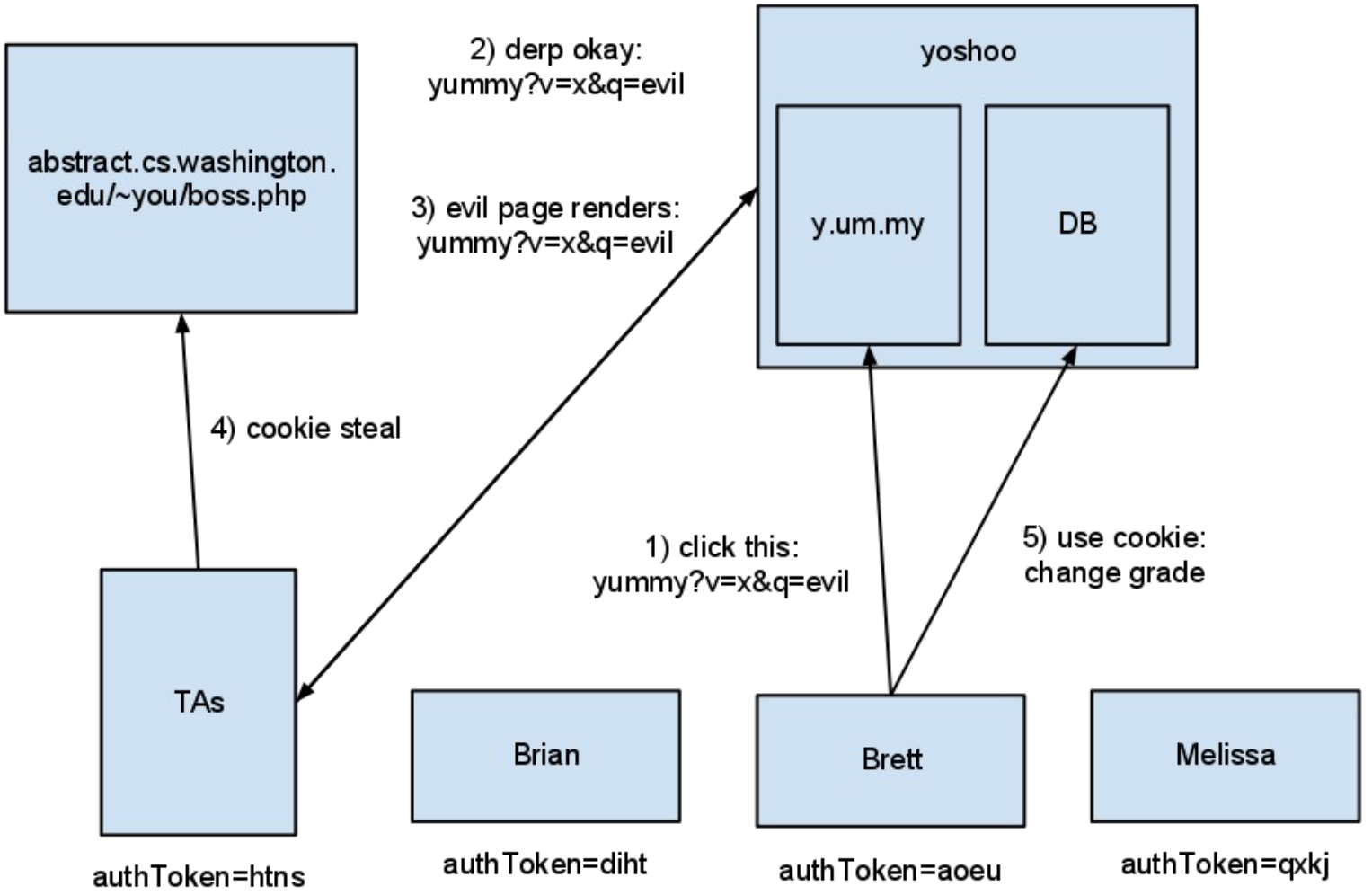
Web Security 2

Roy McElmurry

Lab 2 Explained



Legitimate Cross-Site Communication

- Let's say that company A wants to include a series of pages from company B in their website
- Company B does not want to provide their proprietary code to company A
- Now we have two domains that legitimately need to communicate

Illegitimate Cross-Site Communication

- I, Hacker McHackington load website A in an iframe that fills the screen
- Javascript on the page analyzes website A to learn sensitive information about the user

Same Origin Policy for DOM

When can one website's code access the html elements of another site? When is this even an issue?

```
def canAccess(w1, w2):
   if (w1.protocol, w1.host, w1.port) ==
         (w2.protocol, w2.host, w2.port):
      return True
   elif (w1.protocol, w1.port) == (w2.protocol, w2.port) and
         w1.document.domain == w2.document.domain and
         isSubDomain(w1.document.domain, w1.host) and
         isSubDomain(w2.document.domain, w2.host):
      return True
   else:
      return False
```

Changing Document.domain

- If two sites mutually set their domain to be proper subdomains that match, then they can pass the same origin policy
- When would this occur?
- Problems:
  - Once one domain does this it can be communicated with by any subdomain that also sets its document.domain

Unspecified Behavior

- What should happen in these situations
  - When the urls are ip addresses?
  - When the protocol is file://?
    - Can the file access anything on the harddrive?
    - Can it access anything on the web?

Same Origin Policy for AJAX

- We no longer allow document.domain to be taken into account
- This means that sites cannot collaborate even if they want to
- There are usually additional restrictions on the kinds of requests that can be made, you can send but not receive
- There are restrictions on the kinds of status codes that will be exposed to you and what headers you can send out
  - Why?

Header Restrictions

HTTP header	MSIE6	MSIE7	MSIE8	FF2	FF3	Safari	Opera	Chrome	Android
Accept	OK	OK	OK	OK	OK	OK	OK	OK	OK
Accept-Charset	OK	OK	OK	OK	BANNED	BANNED	BANNED	BANNED	BANNED
Accept-Encoding	BANNED	BANNED	BANNED	OK	BANNED	BANNED	BANNED	BANNED	BANNED
Accept-Language	OK	OK	OK	OK	OK	OK	OK	BANNED	BANNED
Cache-Control	OK	OK	OK	OK	OK	OK	BANNED	OK	OK
Cookie	BANNED	BANNED	BANNED	OK	OK	BANNED	BANNED	BANNED	OK
If-* family (If-Modified-Since, etc)	OK	OK	OK	OK	OK	OK	BANNED	OK	OK
Host	BANNED	BANNED	BANNED	BANNED	BANNED	BANNED	BANNED	BANNED	BANNED
Range	OK	OK	OK	OK	OK	OK	BANNED	OK	OK
Referer	BANNED	BANNED	BANNED	BANNED	BANNED	BANNED	BANNED	BANNED	BANNED
Transfer-Encoding	OK	OK	BANNED	BANNED	BANNED	BANNED	BANNED	BANNED	BANNED
User-Agent	OK	OK	OK	OK	OK	BANNED	OK	BANNED	BANNED
Via	OK	OK	OK	BANNED	BANNED	BANNED	BANNED	BANNED	BANNED

Cookies

- Provides persistent state that spans sessions
- Path option: can assign cookies on a smaller scope to specific paths
- Domain option: can assign cookies on wider scopes to broader domains
- Secure option: the cookie will be sent with requests only if it is over a secure connection
- Http-only option: the cookie cannot be inspected by javascript

Same Origin Policy for Cookies

- Code may only read or write cookies for the current domain
- Secure and Http-only limit the readability of cookies, but do not prevent overwriting
- Third party cookies will be sent with remote requests