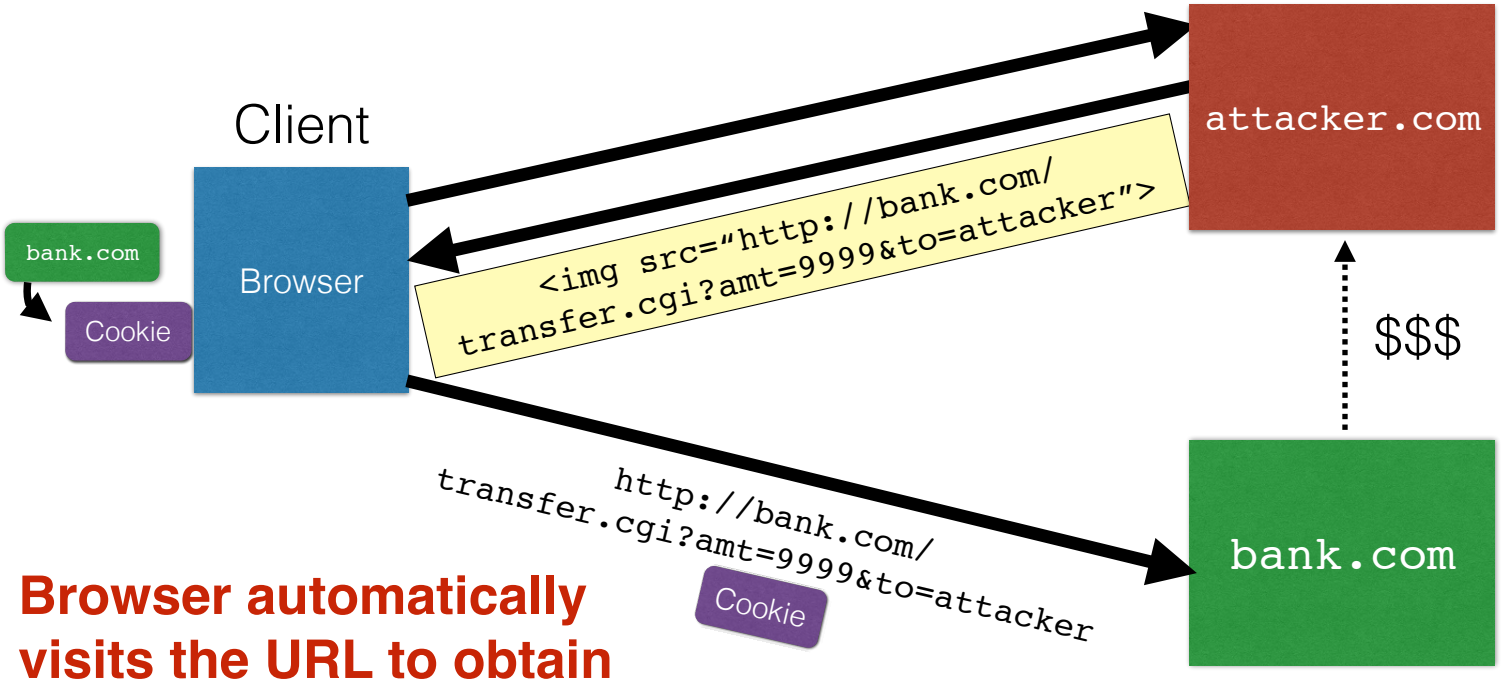
Cross-Site Request Forgery (CSRF)

URLs with side effects

```
http://bank.com/transfer.cgi?amt=9999&to=attacker
```

- GET requests often have **side effects on server state**
  - Even though they are not supposed to
- What happens if
  - the **user is logged in** with an active session cookie
  - a **request is issued for the above link?**
- How could you get a user to visit a link?

Exploiting URLs with side-effects



Browser automatically visits the URL to obtain what it believes will be an image

Cross-Site Request Forgery

- Target: User who has an account on a vulnerable server
- Attack goal: make requests to the server *via the user's browser* that look to the server like the user intended to make them
- Attacker tools: ability to get the user to "click a link" crafted by the attacker that goes to the vulnerable site
- Key tricks:
  - Requests to the web server have predictable structure
  - Use of something like `<img src=…>` to force the victim to send it

CSRF protections: REFERER

- The browser will set the **REFERER** field to the page that hosted a clicked link

HTTP Headers

http://www.zdnet.com/worst-ddos-attack-of-all-time-hits-french-site-7000026330/

GET /worst-ddos-attack-of-all-time-hits-french-site-7000026330/ HTTP/1.1
Host: www.zdnet.com
User-Agent: Mozilla/5.0 (X11; U; Linux i686; en-US; rv:1.9.2.11) Gecko/20101013 Ubuntu/9.04 (jaunty) Firefox/3.6.11
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Accept-Language: en-us,en;q=0.5
Accept-Encoding: gzip,deflate
Accept-Charset: ISO-8859-1,utf-8;q=0.7,*;q=0.7
Keep-Alive: 115
Connection: keep-alive
Referer: http://www.reddit.com/r/security

- **Trust requests from pages a user could legitimately reach**
  - From good users, if referrer header present, generally trusted
  - Defends against session hijacks too

Problem: Referrer optional

- Not included by all browsers
  - Sometimes other legitimate reasons not to have it
- Response: **lenient referrer checking**
  - Blocks requests with a bad referrer, but allows requests with no referrer
  - *Missing referrer always harmless?*
- **No:** attackers can **force the removal of referrer**
  - **Bounce** user off of `ftp:` page
  - **Exploit browser vulnerability** and remove it
  - **Man-in-the-middle** network attack

CSRF Protection: Secretized Links

- **Include a secret in every link/form**
  - Can use a hidden form field, custom HTTP header, or encode it directly in the URL
  - Must not be guessable value
  - Can be same as session id sent in cookie
- **Frameworks help**: Ruby on Rails embeds secret in every link automatically

http://website.com/doStuff.html?sid=81asf98as8eak