CSE 484 / CSE M 584 (Autumn 2011)

Detour: Web Security

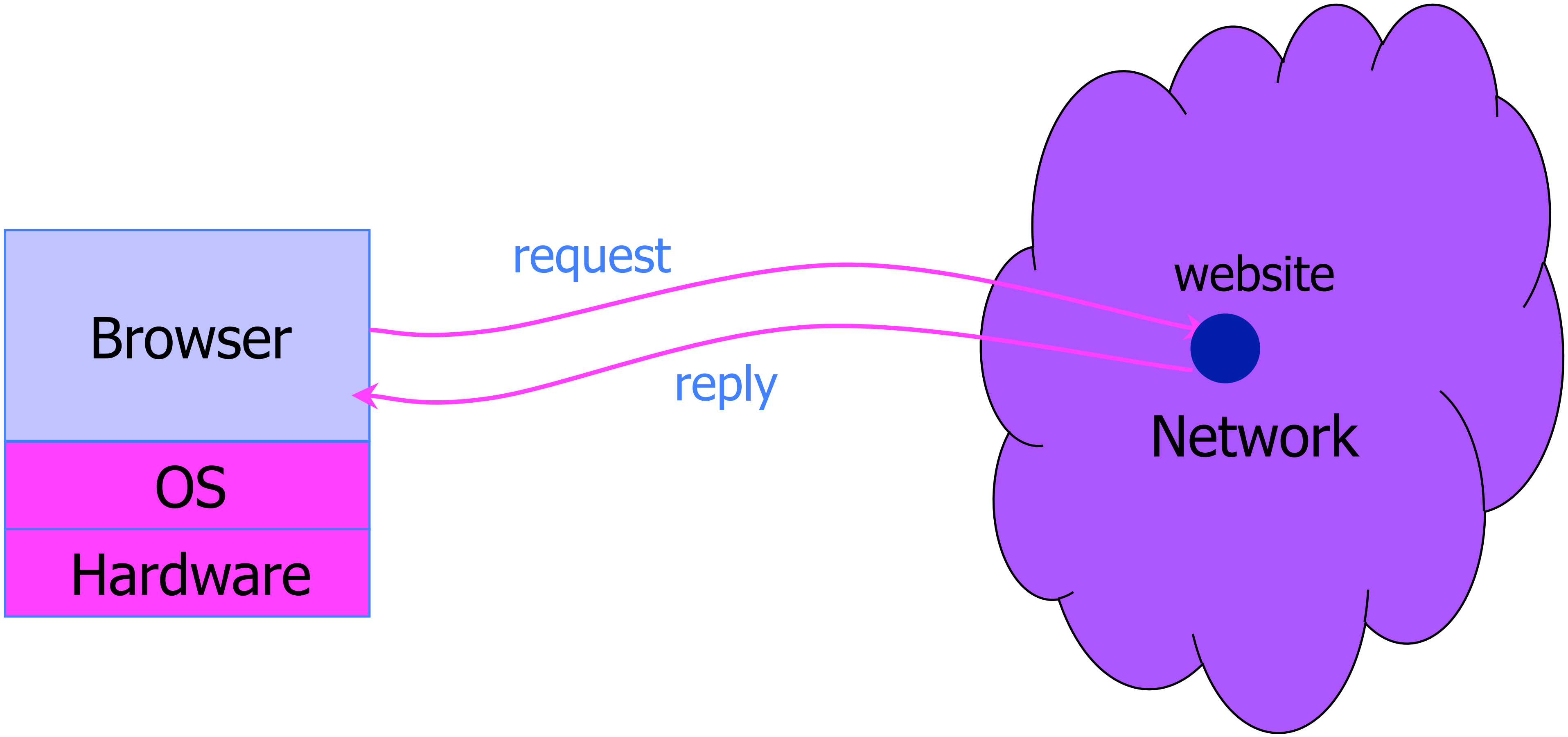
Daniel Halperin
Tadayoshi Kohno

Thanks to Dan Boneh, Dieter Gollmann, John Manferdelli, John Mitchell, Vitaly Shmatikov, Bennet Yee, and many others for sample slides and materials ...

Today, 10/24

- Web Security (intro to Lab 2)
  - Back to Asymmetric Cryptography in a bit
- CELT
- Office hours after class in CSE 210
- Homework 2 (Crypto) coming soon

Browser and Network



Types of problems

- Web browser problems (client side)
  - Exploit vulnerabilities in browsers
  - Install botnets, keyloggers
  - Exfiltrate data

- Web application code (server side)
  - Exploit vulnerabilities in code running on servers (and coming from servers)
  - Examples: XSS, XSRF, SQL injection, insecure parameters, security misconfigurations
  - Steal user credentials, data from databases, ...

Example Questions

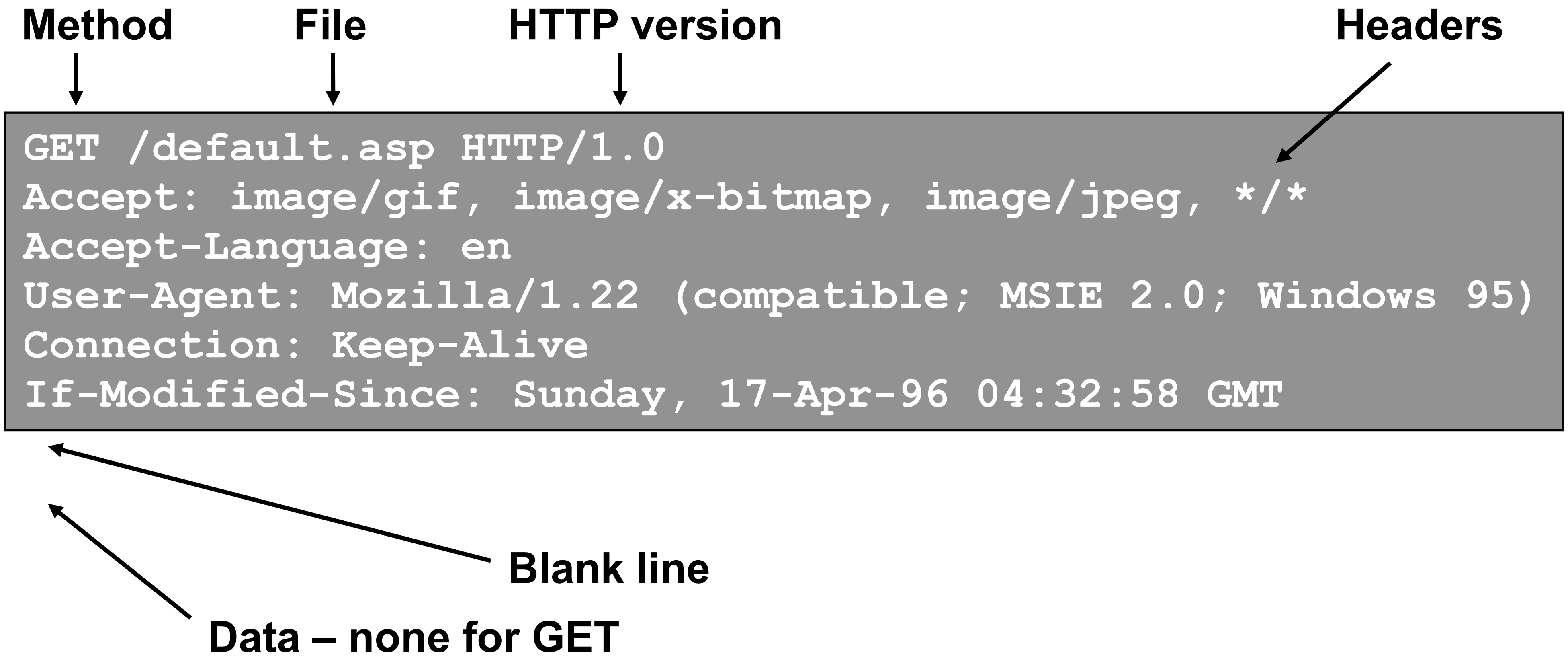
- How does website know who you are?
- How do you know who the website is?
- Can someone intercept traffic ?
- Related: How can you better control flow of information?

- Our focus: High-level principles (lab focuses on pragmatics)
- Focus on a bit of history: How we got here

HTTP: HyperText Transfer Protocol

- Used to request and return data
  - Methods: GET, POST, HEAD, …
- Stateless request/response protocol
  - Each request is independent of previous requests
  - Statelessness has a significant impact on design and implementation of applications
- Evolution
  - HTTP 1.0: simple
  - HTTP 1.1: more complex
  - ... Still evolving ...

HTTP Request



Method

File

HTTP version

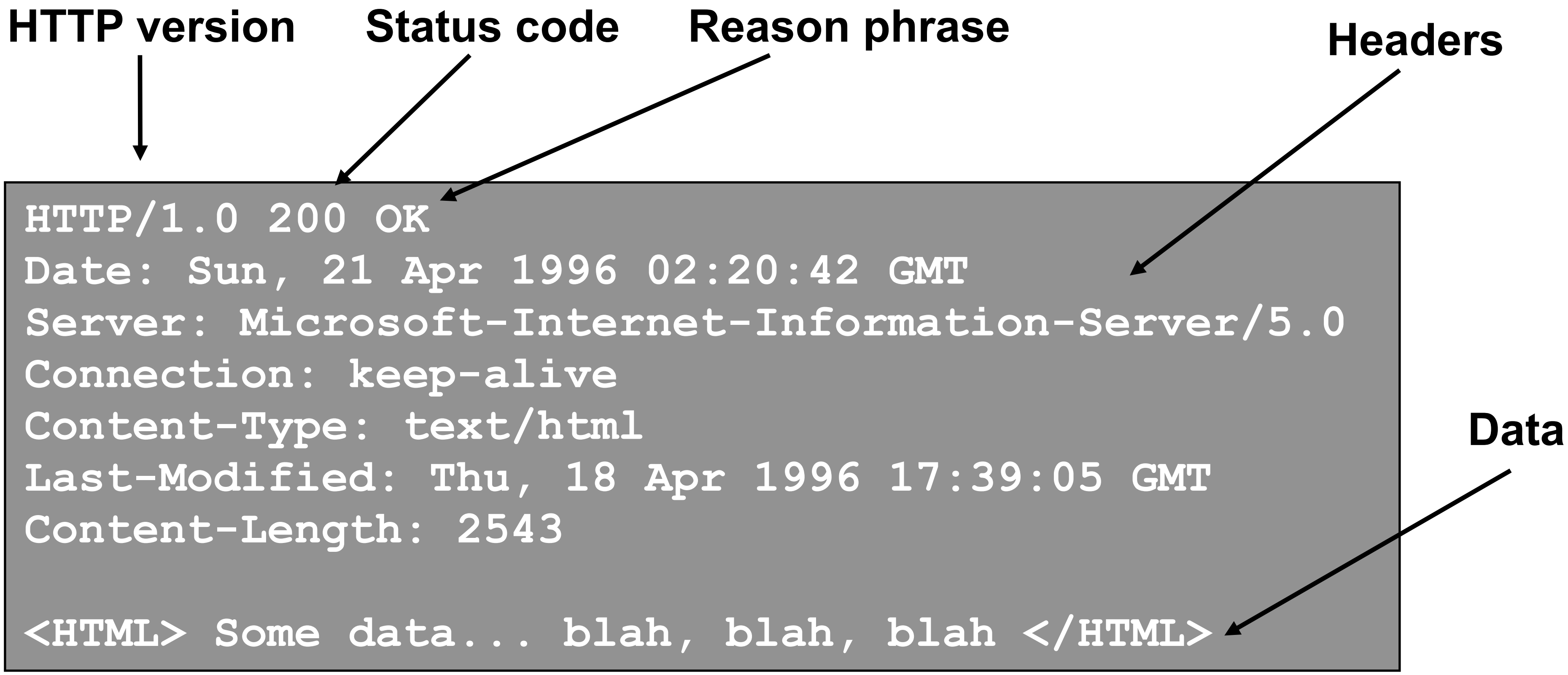
Headers

```
GET /default.asp HTTP/1.0
Accept: image/gif, image/x-bitmap, image/jpeg, */*
Accept-Language: en
User-Agent: Mozilla/1.22 (compatible; MSIE 2.0; Windows 95)
Connection: Keep-Alive
If-Modified-Since: Sunday, 17-Apr-96 04:32:58 GMT
```

Blank line

Data – none for GET

HTTP Response



HTTP version

Status code

Reason phrase

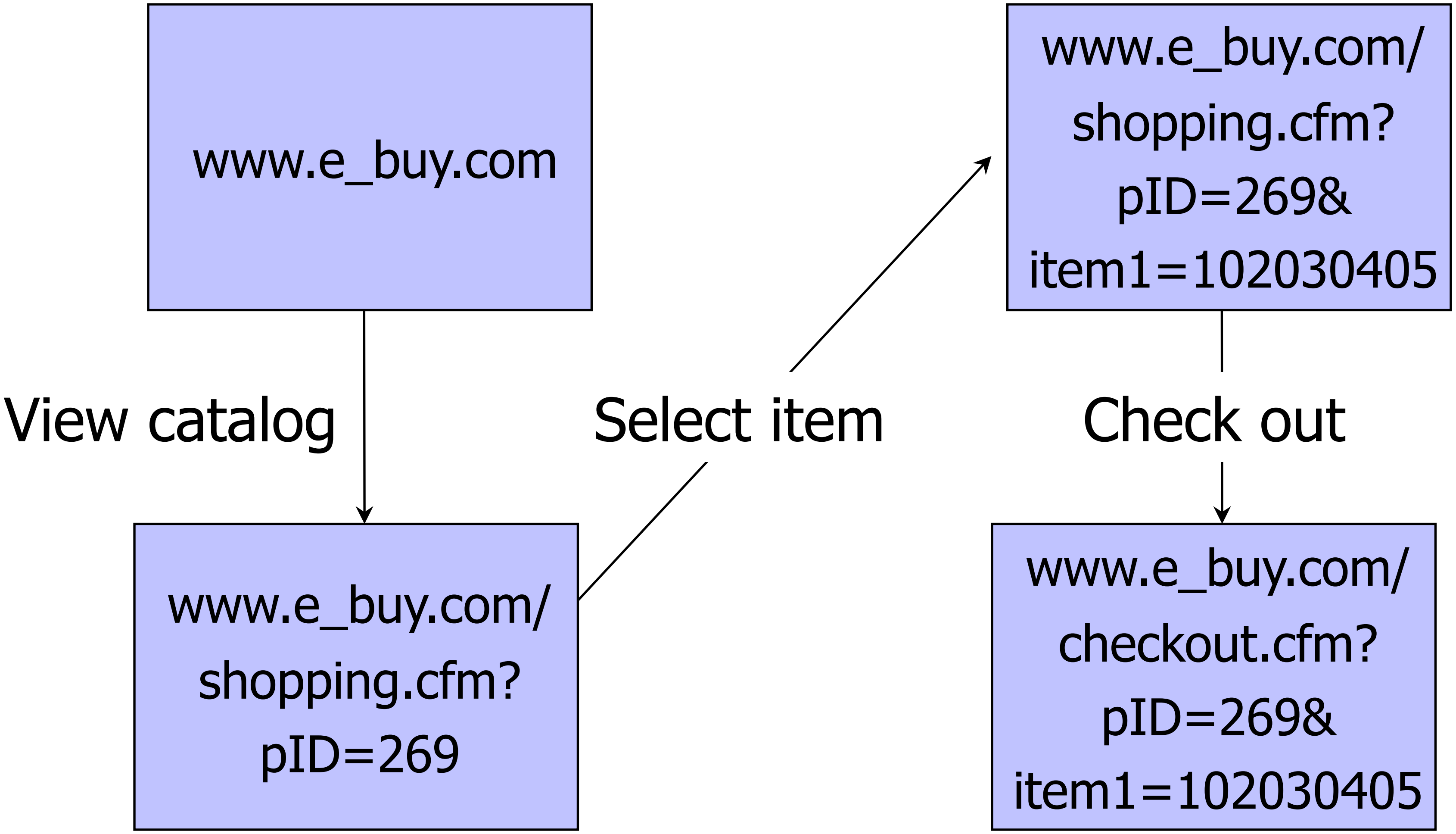
Headers

Data

```
HTTP/1.0 200 OK
Date: Sun, 21 Apr 1996 02:20:42 GMT
Server: Microsoft-Internet-Information-Server/5.0
Connection: keep-alive
Content-Type: text/html
Last-Modified: Thu, 18 Apr 1996 17:39:05 GMT
Content-Length: 2543

<HTML> Some data... blah, blah, blah </HTML>
```

Primitive Browser Session



www.e_buy.com
View catalog
www.e_buy.com/ shopping.cfm? pID=269
Select item
www.e_buy.com/ shopping.cfm? pID=269& item1=102030405
Check out
www.e_buy.com/ checkout.cfm? pID=269& item1=102030405

Store session information in URL; easily read on network

FatBrain.com circa 1999 [due to Fu et al.]

- User logs into website with his password, authenticator is generated, user is given special URL containing the authenticator

  https://www.fatbrain.com/HelpAccount.asp?t=0&p1=me@me.com&p2=540555758

  - With special URL, user doesn't need to re-authenticate
    - Reasoning: user could not have not known the special URL without authenticating first. That's true, BUT…
- Authenticators are global sequence numbers
  - It's easy to guess sequence number for another user

    https://www.fatbrain.com/HelpAccount.asp?t=0&p1=SomeoneElse&p2=540555752
  - Partial fix: use random authenticators

Bad Idea: Encoding State in URL

- Unstable, frequently changing URLs
- Vulnerable to eavesdropping
- There is no guarantee that URL is private
  - Early versions of Opera used to send entire browsing history, including all visited URLs, to Google
  - Modern “browser bars” do similar things
    (possibly somewhat cleaned up, but this is not easy!)