

# **Software Security: Principles (Part Trois)**

***CS 161: Computer Security***

**Prof. David Wagner**

**February 7, 2014**







Multi Expresso







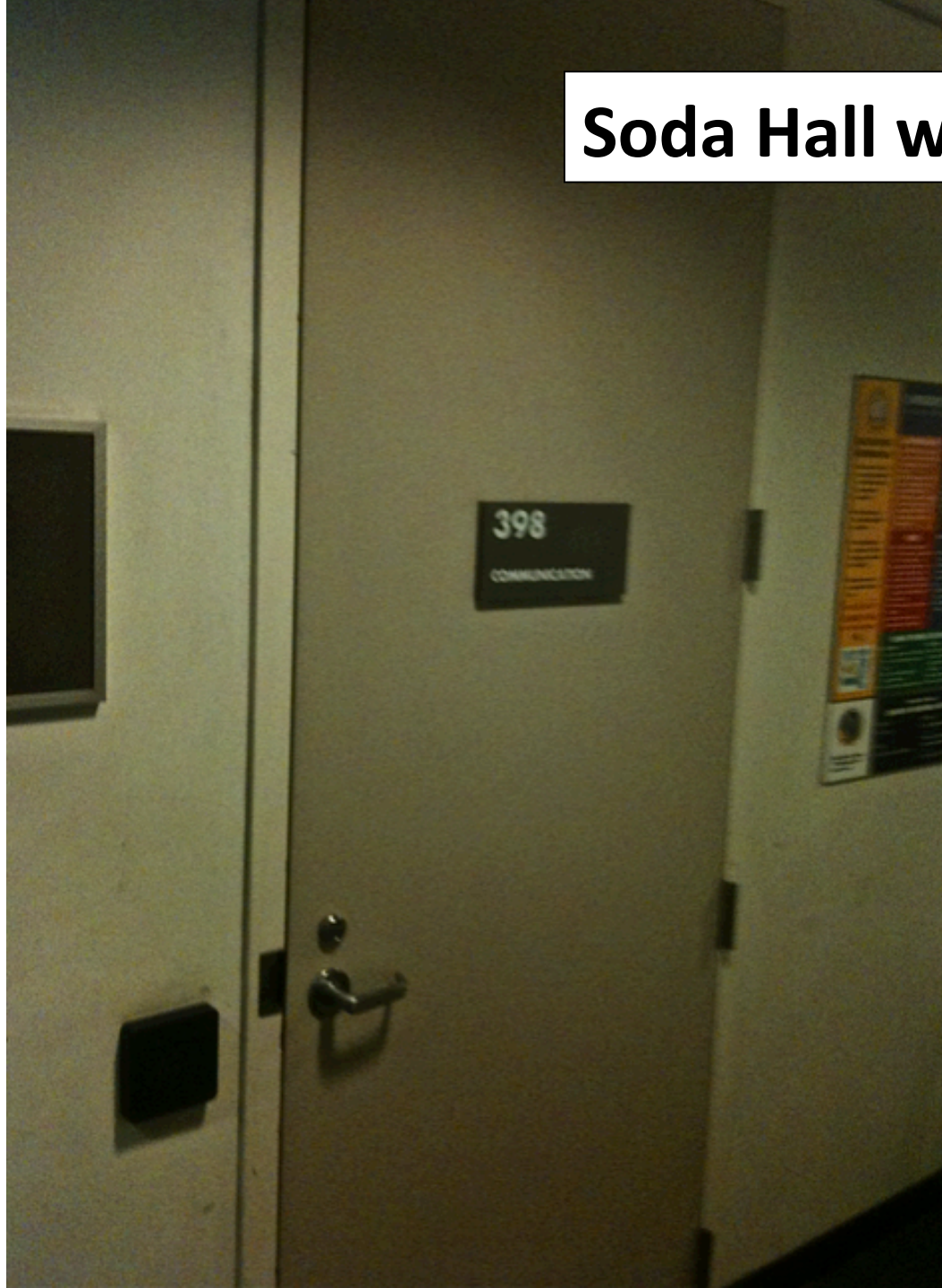


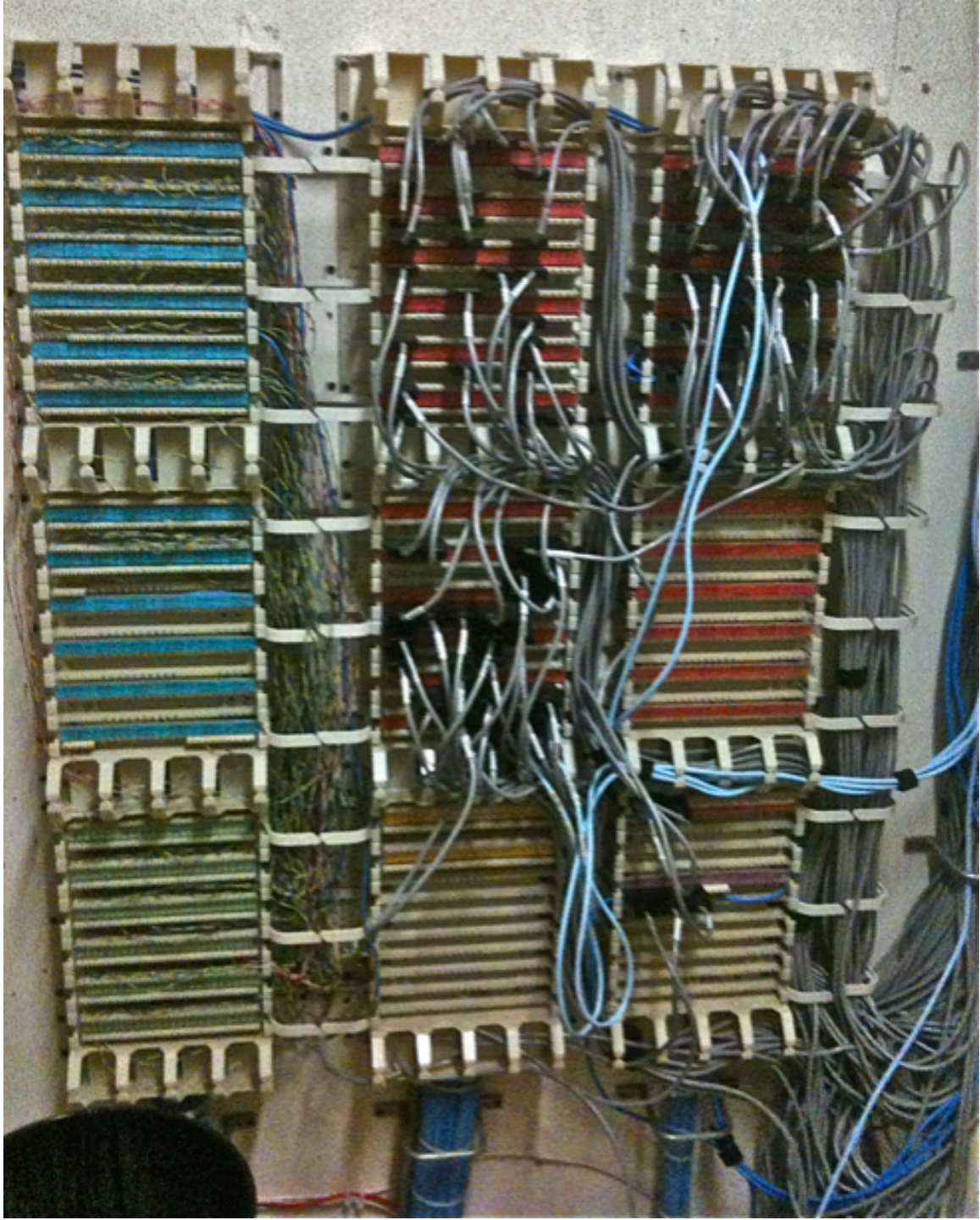


**“Trusted path.”**



# Soda Hall wiring closet









Protection?

398

COMMUNICATION



**“Use fail-safe defaults.”**



## **GSA Container Classes Defined:**

GSA: General Services Administration (US Government)

GSA Class 1:

a GSA approved container meeting Federal Specification AA-F-357(canceled) with entry protection consisting of 10 Man-Minutes forced entry, 30 Man-Minutes surreptitious entry and 1 hour fire rating

GSA Class 2:

a GSA approved container meeting Federal Specification AA-F-357(canceled) with entry protection consisting of 5 Man-Minutes forced entry, 20 Man-Minutes surreptitious entry and 1 hour fire rating

GSA Class 3:

a GSA approved uninsulated container meeting Federal Specification AA-F-358 with entry protection consisting of 20 Man-Minutes surreptitious entry

GSA Class 4:

a GSA approved uninsulated container meeting Federal Specification AA-F-358 with entry protection consisting of 5 Man-Minutes forced entry, 20 Man-Minutes surreptitious entry

GSA Class 5:

a GSA approved uninsulated container meeting Federal Specification AA-F-358 with entry protection consisting of 10 Man-Minutes forced entry, 20 Man-Hours surreptitious entry and 30 Man-Minutes Covert entry

GSA Class 6:

a GSA approved uninsulated container meeting Federal Specification AA-F-358 with entry protection consisting of 20 Man-Hours surreptitious entry and 30 Man-Minutes covert entry



**Class 5 Four Drawer**







**“Detect if you can’t prevent.”**

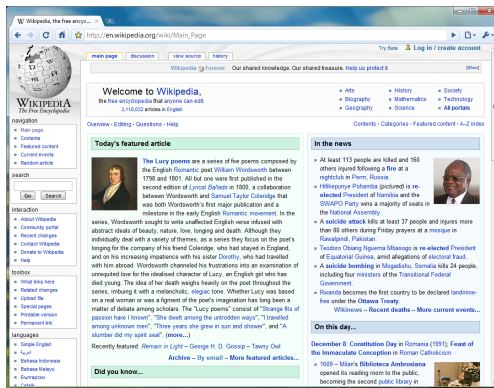
# Summary

- Use *security thinking* – think like an attacker, identify architectural defenses
- You can practice this in everyday life!

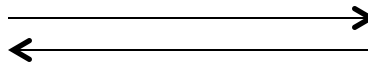
# **Web Security**

# What is the Web?

- A platform for deploying applications, *portably and securely*



client



server

# A historical perspective

- The web is an example of “bolt-on security”
- Originally, the web was invented to allow physicists to share their research papers
  - Only textual web pages + links to other pages; no security model to speak of
- Then we added embedded images
  - Crucial decision: a page can embed images loaded from another web server
- Then, Javascript, dynamic HTML, AJAX, CSS, frames, audio, video, ...
- Today, a web site is a distributed application



# Security on the web

- Integrity: malicious web sites should not be able to tamper with integrity of my computer or my information on other web sites
- Confidentiality: malicious web sites should not be able to learn confidential information from my computer or other web sites
- Privacy: malicious web sites should not be able to spy on me or my activities online

# Security on the web

- Risk #1: we don't want a malicious site to be able to trash my files/programs on my computer
  - Browsing to `awesomevids.com` (or `evil.com`) should not infect my computer with malware, read or write files on my computer, etc.

# Security on the web

- Risk #1: we don't want a malicious site to be able to trash my files/programs on my computer
  - Browsing to `awesomevids.com` (or `evil.com`) should not infect my computer with malware, read or write files on my computer, etc.
- Defense: Javascript is sandboxed; try to avoid security bugs in browser code; privilege separation; automatic updates; etc.

# Security on the web

- Risk #2: we don't want a malicious site to be able to spy on or tamper with my information or interactions with other websites
  - Browsing to evil.com should not let evil.com spy on my emails in Gmail or buy stuff with my Amazon account

# Security on the web

- Risk #2: we don't want a malicious site to be able to spy on or tamper with my information or interactions with other websites
  - Browsing to evil.com should not let evil.com spy on my emails in Gmail or buy stuff with my Amazon account
- Defense: the same-origin policy
  - A security policy grafted on after-the-fact, and enforced by web browsers
  - Intuition: each web site is isolated from all others

# Security on the web

- Risk #3: we want data stored on a web server to be protected from unauthorized access

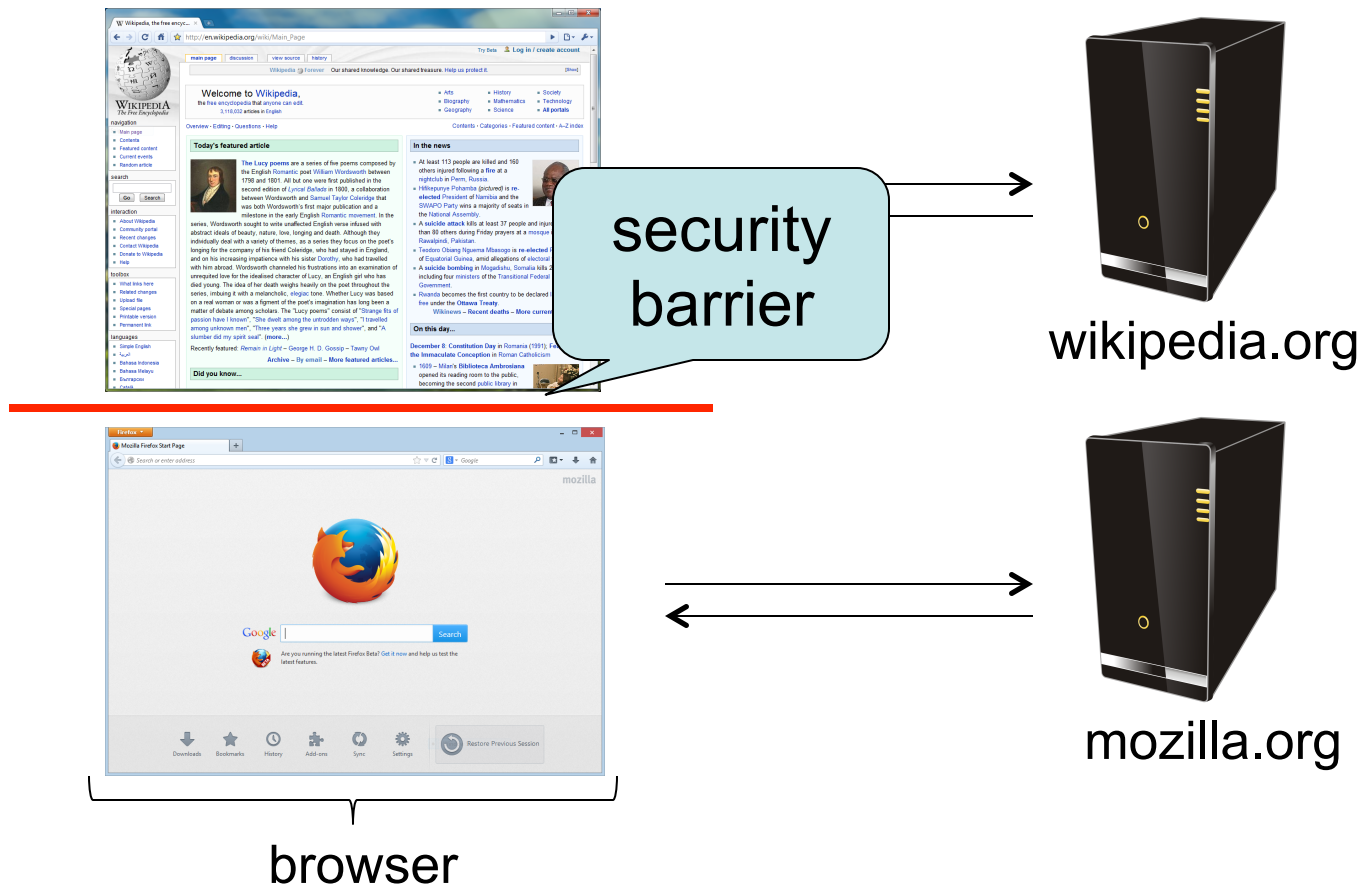


# Security on the web

- Risk #3: we want data stored on a web server to be protected from unauthorized access
- Defense: server-side security

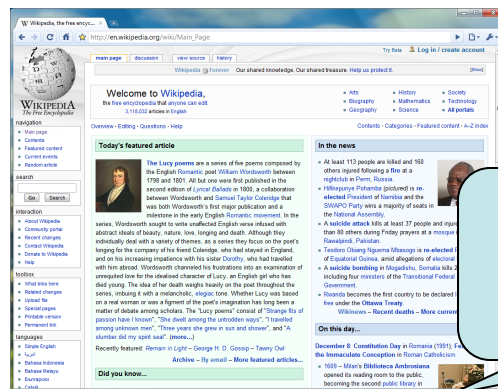
# Same-origin policy

- Each site is isolated from all others



# Same-origin policy

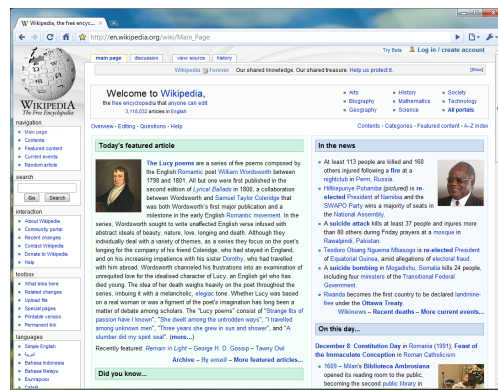
- Multiple races from same site aren't isolated



No security barrier



wikipedia.org

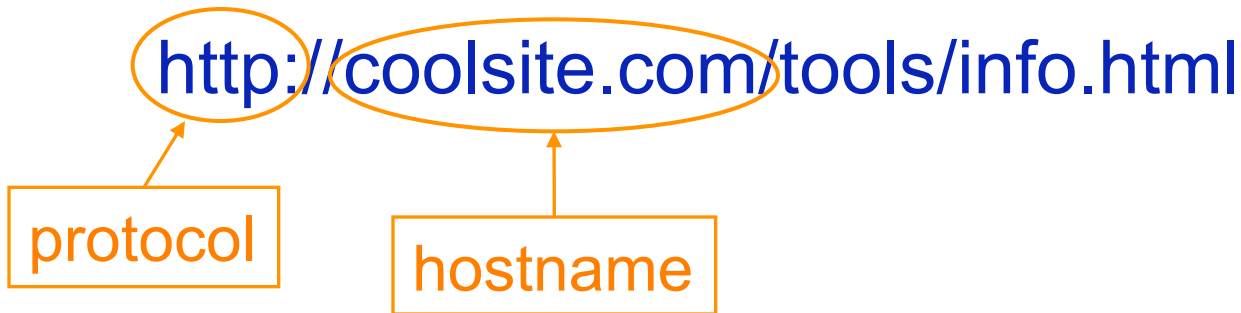


wikipedia.org

browser

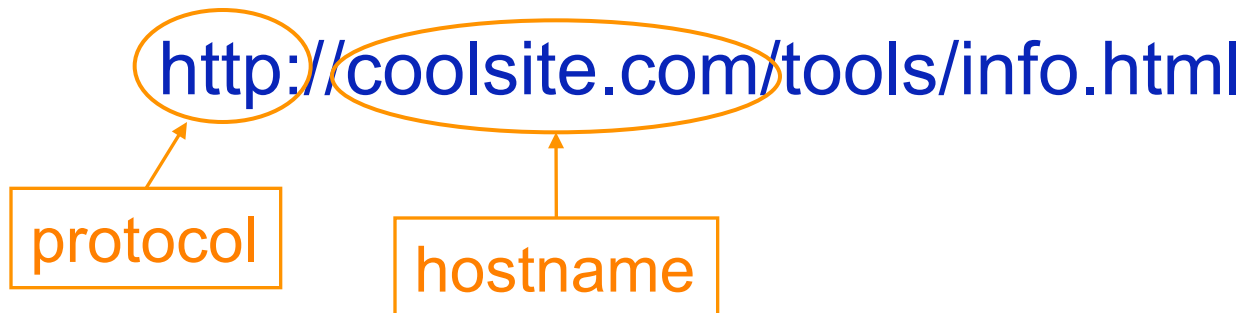
# Same-origin policy

- Granularity of protection: the *origin*
- Origin = protocol + hostname (+ port)



# Same-origin policy

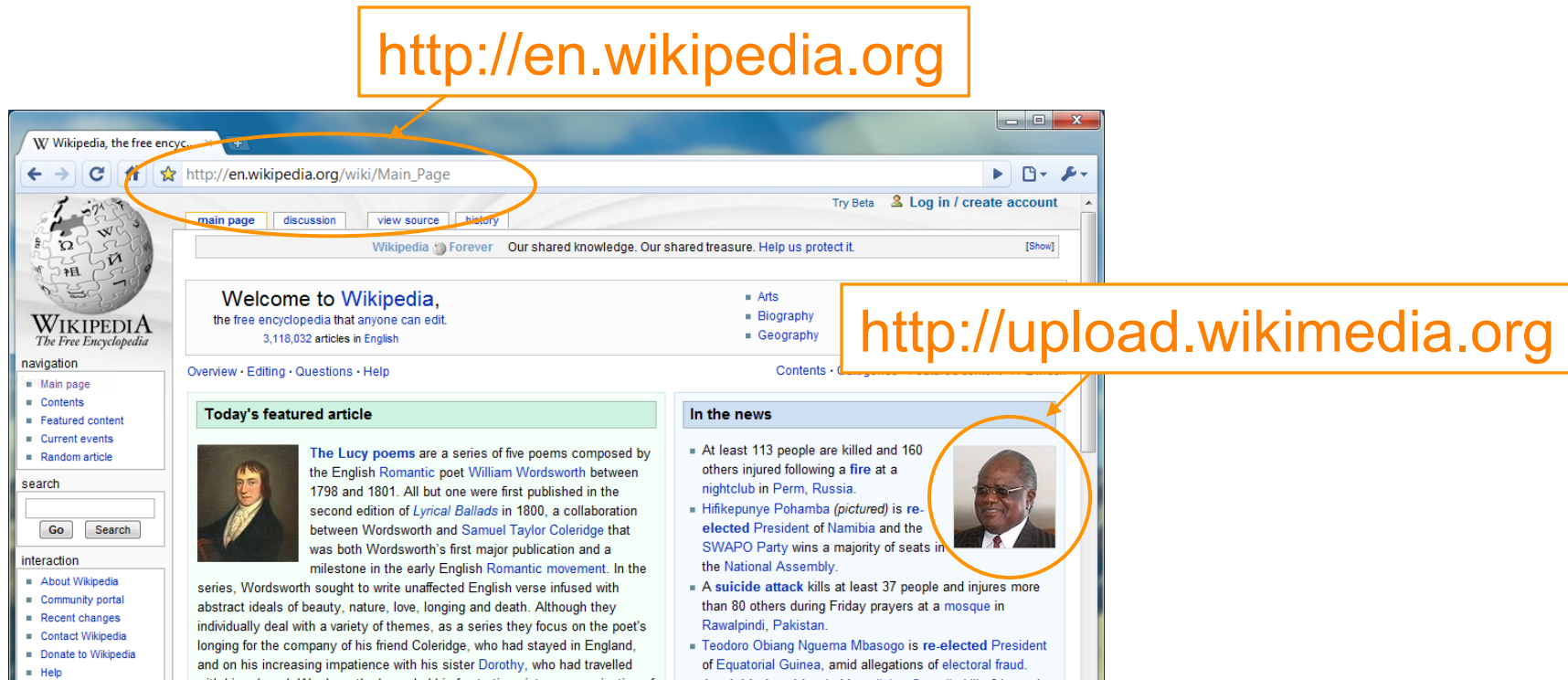
- Granularity of protection: the *origin*
- Origin = protocol + hostname (+ port)



- Javascript on one page can read, change, and interact freely with all other pages from the same origin

# Same-origin policy

- The origin of a page (frame, image, ...) is derived from the URL it was loaded from



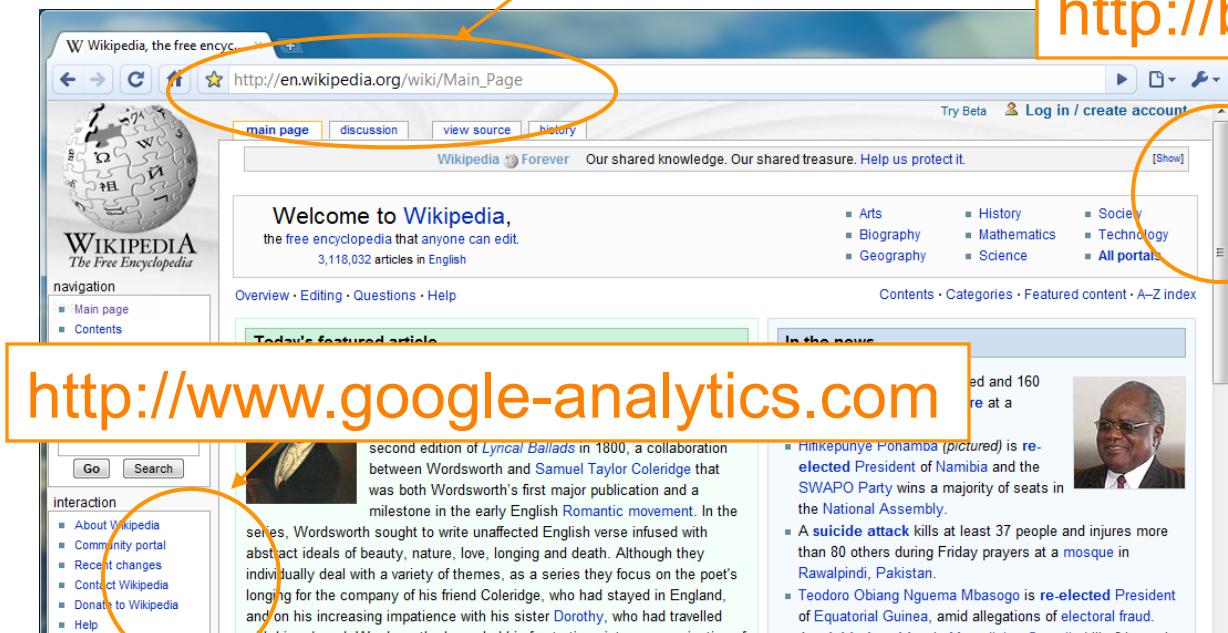


# Same-origin policy

- The origin of a page (frame, image, ...) is derived from the URL it was loaded from
- Special case: Javascript runs with the origin of the page that loaded it

<http://en.wikipedia.org>

<http://bits.wikimedia.org>



# Coming up

- Attacks on web servers