threatpost


# POWER GRID HONEYPOT PUTS FACE ON ATTACKS

by **Michael Mimoso** Follow @mike_mimoso February 9, 2016 , 5:36 am

TENERIFE, Spain –The rhetoric around hacking the power grid would have you believe it's a relatively mundane practice. Policymakers, intelligence agencies and vendors, for example, spread the word gleefully, leaning on scenarios such as state-sponsored hackers shutting off the lights in the dead of winter as a scare tactic to glean budget and influence.

The honeypot gave the APT groups access to transmission controls and protection relays that could disconnect a power generating plant from the bulk grid, Chowdhury said, leading to massive outages for hundreds of thousands of people.

The honeypot beaconed out to hackers with intentionally misconfigured operational technology systems, open Wi-Fi, and more.

# cs642

computer security

# *web security*

adam everspaugh
ace@cs.wisc.edu

# review

* memory protections
  / data execution prevention
  / address space layout randomization
  / stack protector

* Sandboxing
  / Limit damage if a process is compromised

# today

* web software stack
* HTTP basics, DOM
* Browser security model
* Cookies + session hijacking

As few permissions as possible,
essentially: no system calls

Browser
Manager

IPC

IPC

IPC

sandbox

Rendering Engine

HTML parsing, CSS parsing, JavaScript, DOM, rendering

sandbox

Rendering Engine

HTML parsing, CSS parsing, JavaScript, DOM, rendering

sandbox

chrome extension

sandbox

Adobe Flash

Each sandbox is a separate process

# chrome sandbox

# {Threat model}

assets — attackers — vulnerabilities

attack vectors

# {Security model}

subjects — trusted components

countermeasures — security goals

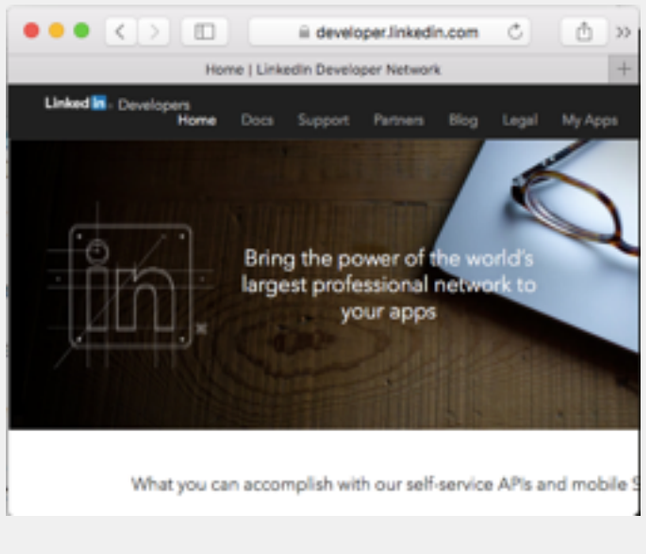
Why might web browsers be desirable targets?

What's involved in a typical web stack?

think-*pair*-share

# web browsers

http://linkedin.com




Browser

+ajax / websockets

html +css +javascript

http

[tls?]

tcp/ip

linkedin.com

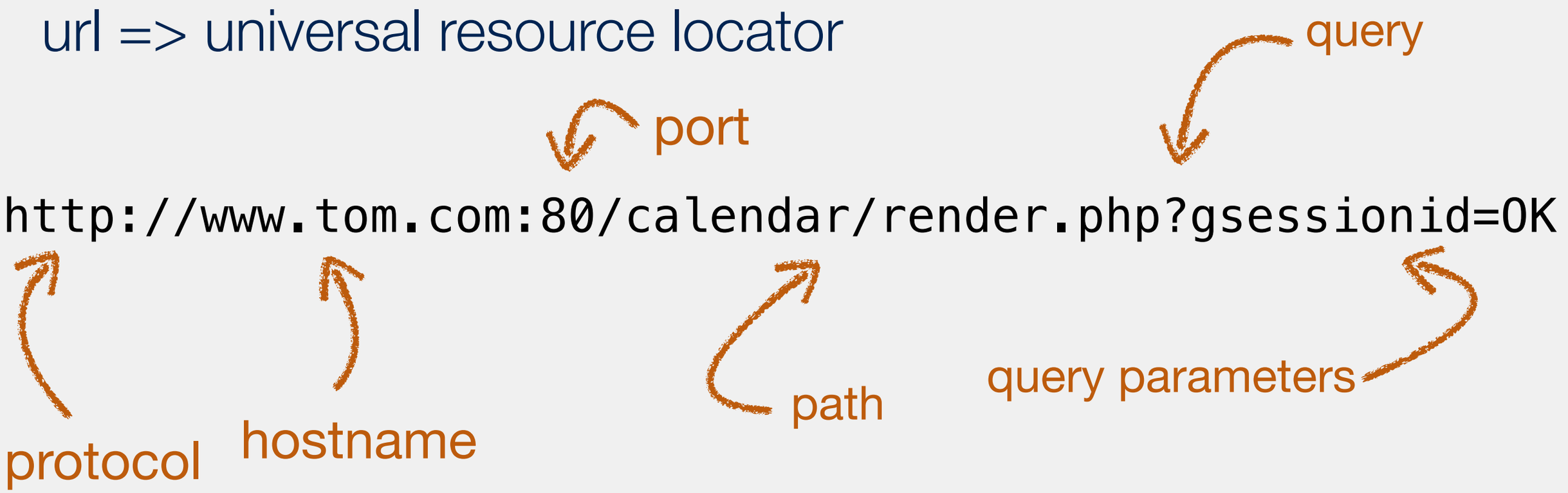
web server -- nginx, apache, ...

app server -- ruby, php, django, asp, node.js, ...

datastore -- sql, nosql, ...

# web ecosystem

url => universal resource locator




Special characters:
+ = space
? = separates URL from parameters
% = special characters
%0A = newline
%20 = space
/ = divides directories, subdirectories
# = bookmark
& = separate query parameters

# http basics


http

Method Path HTTP version Headers

```
GET /index.html HTTP/1.1
Accept: image/gif, image/x-bitmap, image/jpeg, */*
Accept-Language: en
Connection: Keep-Alive
User-Agent: Mozilla/1.22 (compatible; MSIE 2.0; Windows 95)
Host: www.example.com
Referer: http://www.google.com?q=dingbats

```

Blank line

Data – none for GET

| GET: no side effect | POST: possible side effect |
|---|---|

# http request

HTTP version

Status code

Reason phrase

Headers

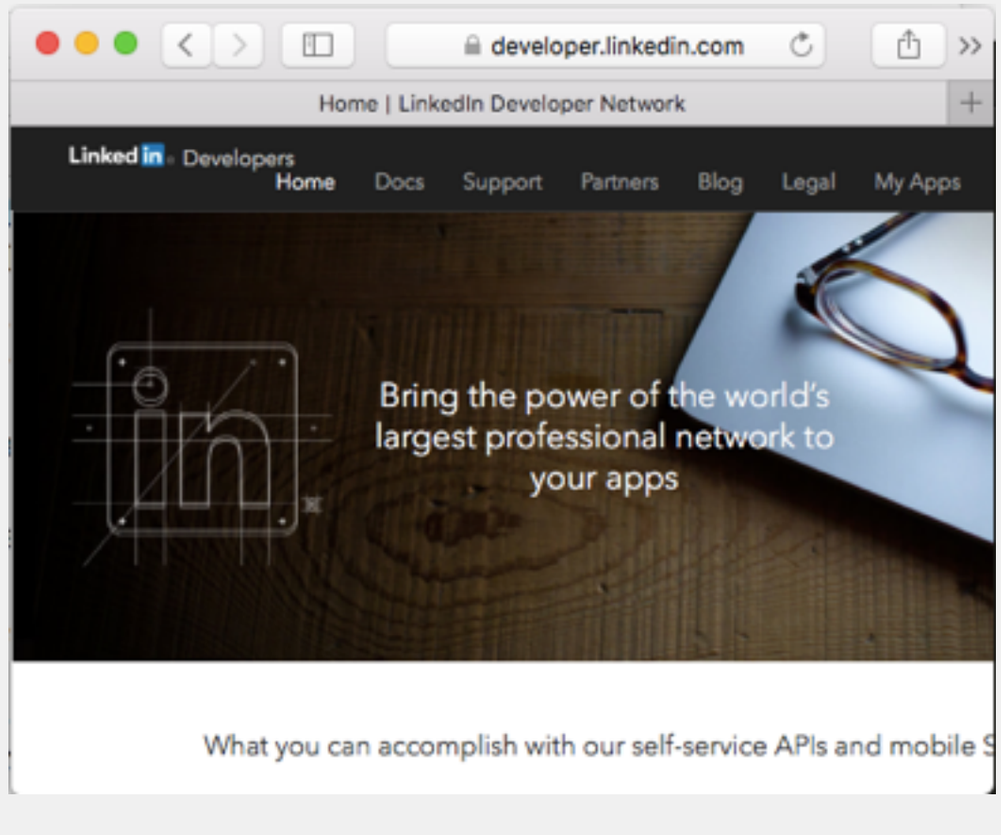
```
HTTP/1.0 200 OK
Date: Sun, 21 Apr 1996 02:20:42 GMT
Server: Microsoft-Internet-Information-Server/5.0
Connection: keep-alive
Content-Type: text/html
Last-Modified: Thu, 18 Apr 1996 17:39:05 GMT
Set-Cookie: …
Content-Length: 2543

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

Data

Cookies

# http response

* Each tab or window
  / Fetch, parse content
  / Render page
    // Process HTML, CSS
    // Might fetch more content
    // Run JavaScript




* Respond to events
  / User actions: OnClick, OnMouseover
  / Rendering: OnLoad, OnBeforeUnload
  / Timing: setTimeout(),  clearTimeout()

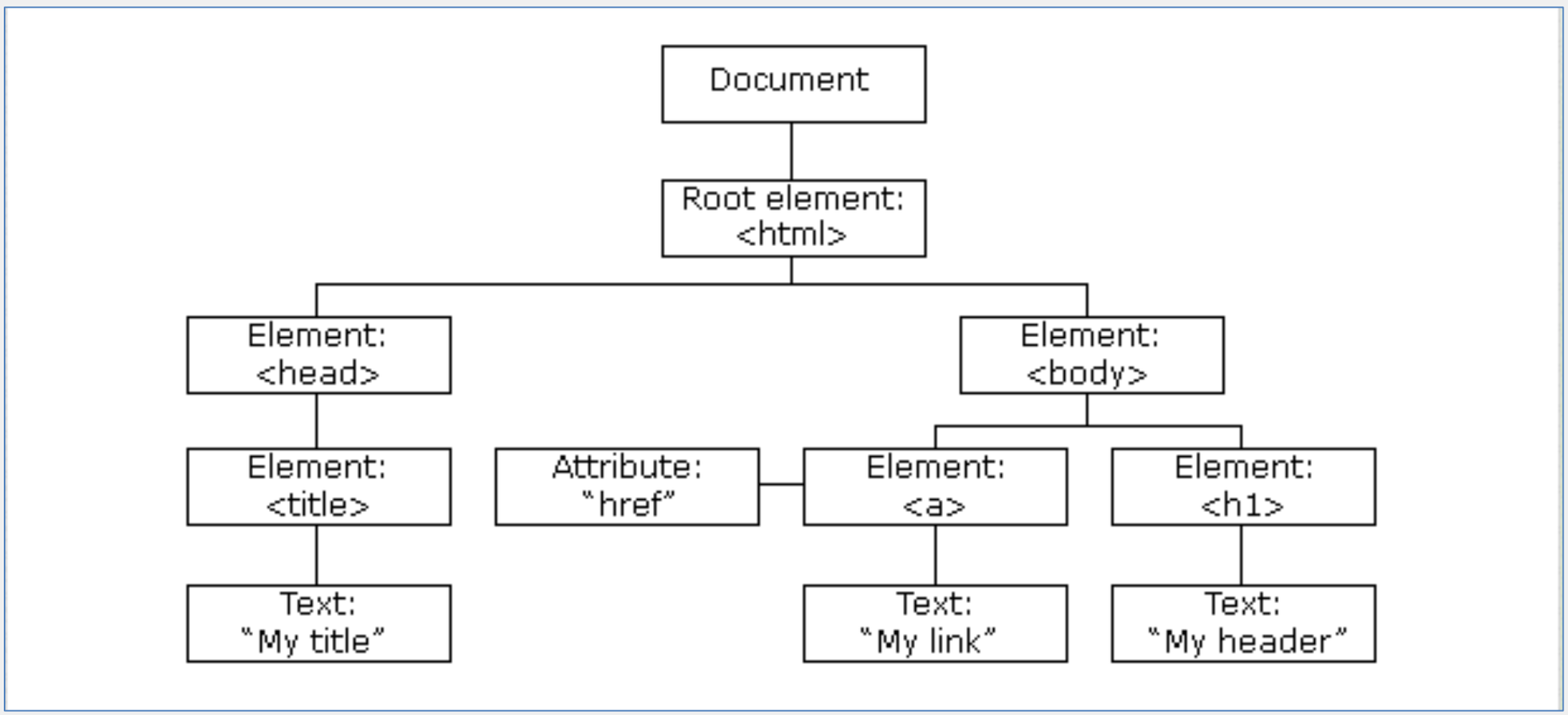
basic browser execution

# Document object model (DOM)

Object-oriented way to organize objects in a web page

Properties: document.alinkColor, document.URL, document.forms[ ], document.links[ ], document.anchors[ ]

Methods: document.write(document.referrer)



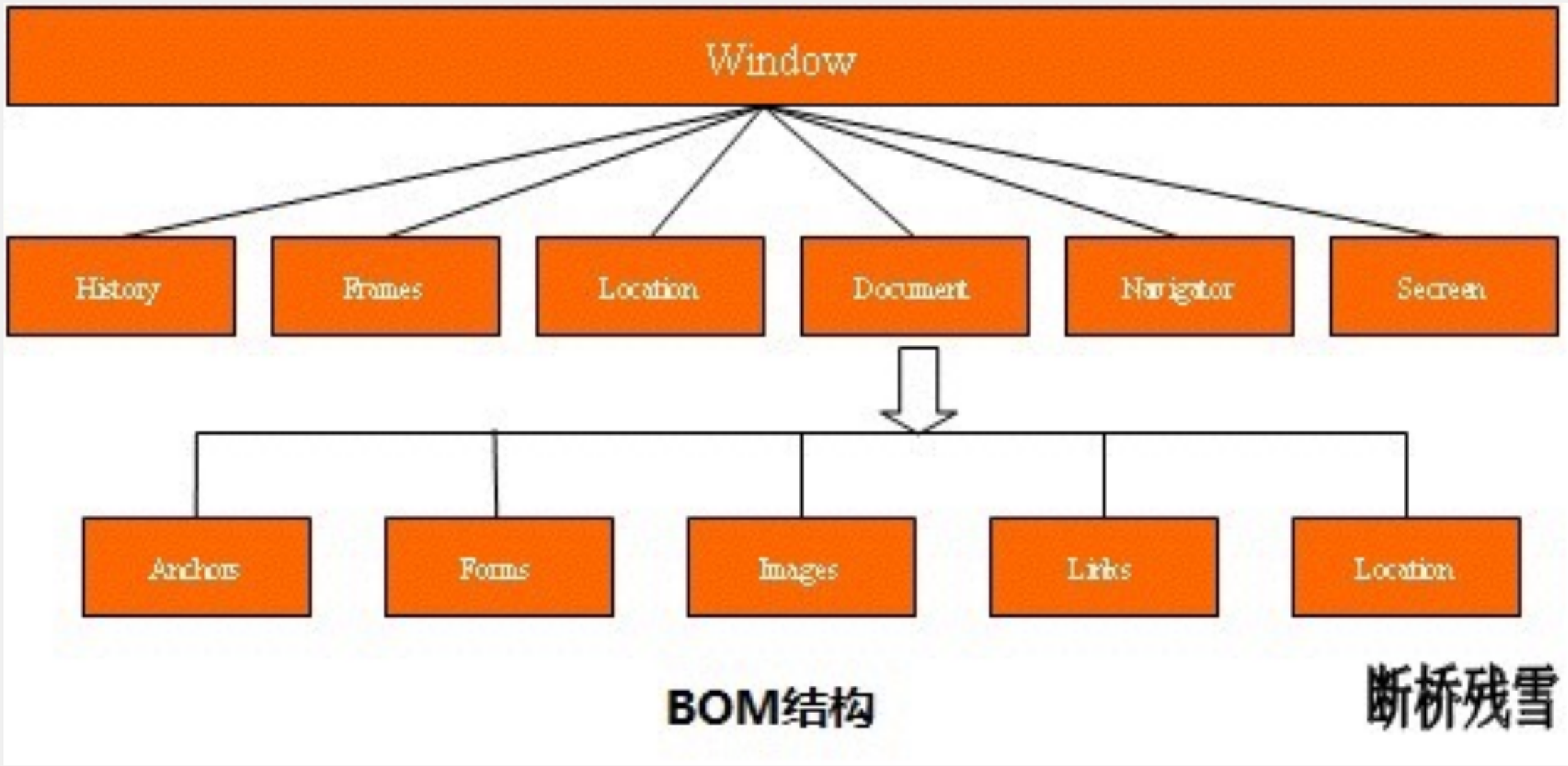

From http://w3schools.com/htmldom/default.asp

dom

# Browser object model (BOM)

Corresponding model for larger browser window,

document, frames[], history, location,
navigator (type and version of browser)




bom

```
<img src="bucky.jpg" height="50pt" width="50pt"/>
```

Displays an image

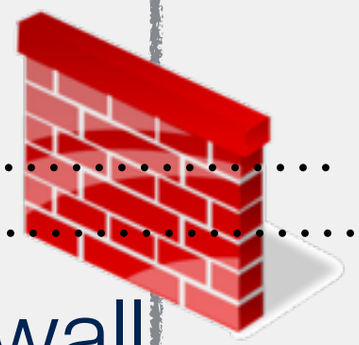
How can an attacker use this?

<img/>

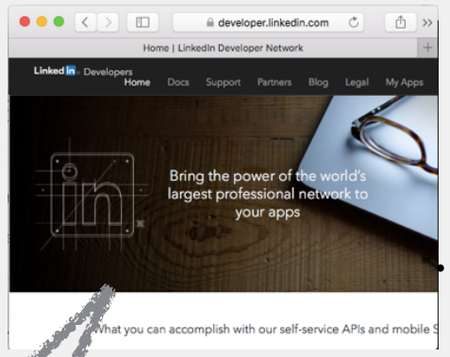
```
<html><body><img id="test" style="display: none">
<script>
  var test = document.getElementById('test');
  var start = new Date();
  test.onerror = function() {
    var end = new Date();
    alert("Total time: " + (end - start));
  }
  test.src = "http://www.example.com/page.html";
</script>
</body></html>
```

# javascript timing

webserver



firewall




Browser

JavaScript can:
Request images from an internal IP address
<img src="192.168.0.4:8080"/>
Use timeout/error to determine success
Fingerprint known applications
Report back to webserver

corporate network

# reconnaissance

Should be safe to visit a malicious website


Should be safe to visit multiple websites simultaneously


Should be safe to delegate content


# browser security model

# multi-origin pages

Most web pages are not single-origin --
content comes from many different places

Why? What other domains will be referenced on any given web page?

If you have a laptop: load a web page, view the source, try to count the number of URLs referenced

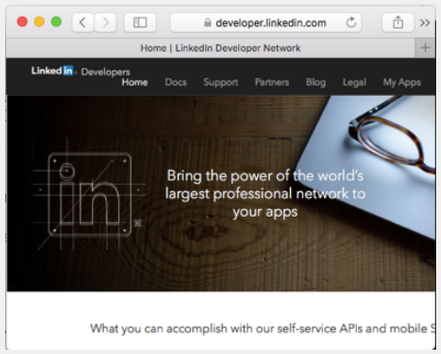
think-*pair*-share

# multi-origin pages

* iframes: `<iframe src="//site.com/frame.html"/>`
* scripts: `<script src="//site.com/script.js"/>`
* CSS: `<link rel="stylesheet" type="text/css” href="//site.com/theme.css"/>`
* Images
* Videos
* Content delivery network (CDN)
* Authentication (OAUTH, others)
* Payment
* Social sharing buttons
* Tracking beacons
* Analytics
* Ads
* Ads
* Ads
* Even more ads

# *url counting*

- Cookies permit browsers to store state associated with a website
- JavaScript can also set cookies




Browser

GET …

website.com

```
HTTP Header:
Set-cookie:  NAME=value;
domain = (when to send);
path = (when to send);
secure = (only send over SSL);
expires = (when expires);
```

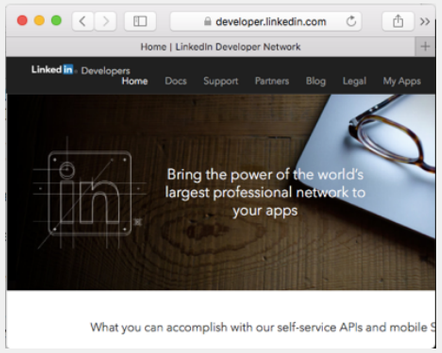
# cookies

# cookie scope rules

* Legal settings for domain and path

* Any (non-top-level domain) suffix is valid:
  / Navigate to www.cs.wisc.edu
  / Valid domains: www.cs.wisc.edu, cs.wisc.edu, wisc.edu
  / Invalid: eng.wisc.edu, umich.edu

* Path can be set to anything




Browser

```
GET website.com/some-path
Cooke:NAME=value;
```

website.com

* Browser automatically sends all cookies with:
  / domain scope is a suffix of url-domain
  / path is a prefix of url-path
  / protocol=HTTPS if cookie marked `secure`

# sending cookies

* Web ecosystem
* HTTP basics, DOM
* Multi-origin pages
* Browser security model
* Cookies

# recap