CSE 127: Computer Security

Web Intro

George Obaido

UCSD

Winter 2022

Some slides from Nadia Heninger, Deian Stefan, Zakir Durumeric, Dan Boneh, and Kirill Levchenko

Brief: Mitigating side channels

Next: Web Intro

https://www.microsoft.com/en-us/research/video/verifying-constant-time-implementations/

Mitigating Cache-based Side Channels

- There's never a completion solution to avoiding side-channel attacks. A few mitigations are:

- **Application-specific:** Disable resource sharing, or isolate applications. One example is page coloring.

- **Compiler-based:** One example is Biscuit, developed at Georgia Tech. Able to guess misses and alerts the CPU scheduler about abnormal behaviour.

- **Redesigning Hardware**: Hard due to large overheads involved.

- Other solutions are ASLR (although, easy to defeat by Spectre and Meltdown)

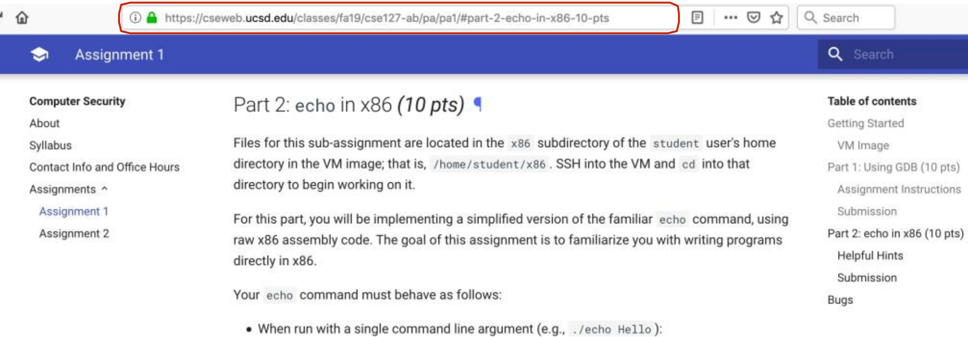
Overall, secure algorithms still need secure implementation.

Lecture objectives

- Basic understanding of how the web works
- Understand relevant attacker models
- Understand browser same-origin policy

HTTP protocol

- Protocol from 1989 that allows fetching of resources (e.g., HTML documents)
- Resources have a uniform resource location (URL):



https://cseweb.ucsd.edu/classes/fa19/cse127-ab/pa/pa1/#part-2-echo-in-x86-10-pts

Part 2: echo in x86 *(10 pts)*

Files for this sub-assignment are located in the `x86` subdirectory of the `student` user's home directory in the VM image; that is, `/home/student/x86`. SSH into the VM and `cd` into that directory to begin working on it.

For this part, you will be implementing a simplified version of the familiar `echo` command, using raw x86 assembly code. The goal of this assignment is to familiarize you with writing programs directly in x86.

Your `echo` command must behave as follows:

- When run with a single command line argument (e.g., `./echo Hello`):

HTTP protocol

- Protocol from 1989 that allows fetching of resources (e.g., HTML documents)
- Resources have a uniform resource location (URL):

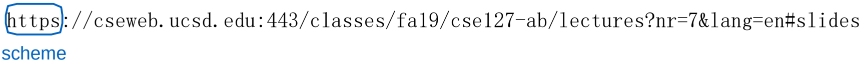
HTTP protocol

- Protocol from 1989 that allows fetching of resources (e.g., HTML documents)
- Resources have a uniform resource location (URL):

https://cseweb.ucsd.edu:443/classes/fa19/cse127-ab/lectures?nr=7&lang=en#slides

HTTP protocol

- Protocol from 1989 that allows fetching of resources (e.g., HTML documents)
- Resources have a uniform resource location (URL):

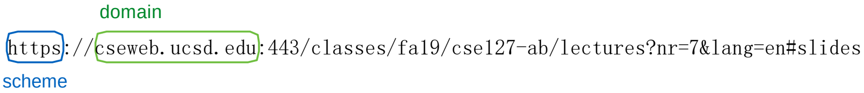


https://cseweb.ucsd.edu:443/classes/fa19/cse127-ab/lectures?nr=7&lang=en#slides

scheme

HTTP protocol

- Protocol from 1989 that allows fetching of resources (e.g., HTML documents)
- Resources have a uniform resource location (URL):



domain

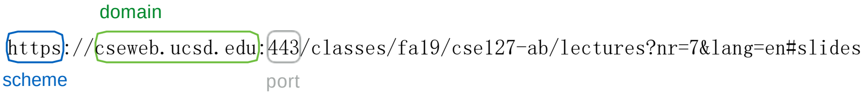
`https://cseweb.ucsd.edu:443/classes/fa19/cse127-ab/lectures?nr=7&lang=en#slides`

scheme

HTTP protocol

- Protocol from 1989 that allows fetching of resources (e.g., HTML documents)

- Resources have a uniform resource location (URL):



domain

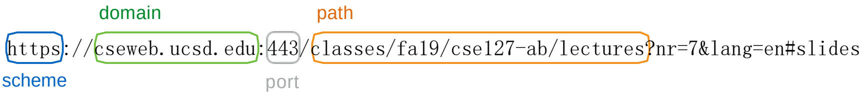
https://cseweb.ucsd.edu:443/classes/fa19/cse127-ab/lectures?nr=7&lang=en#slides

scheme

port

HTTP protocol

- Protocol from 1989 that allows fetching of resources (e.g., HTML documents)
- Resources have a uniform resource location (URL):

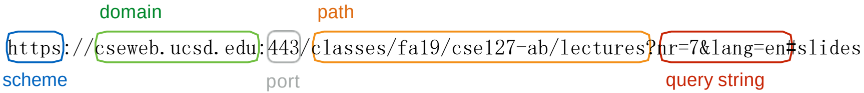


https://cseweb.ucsd.edu:443/classes/fa19/cse127-ab/lectures?nr=7&lang=en#slides

HTTP protocol

- Protocol from 1989 that allows fetching of resources (e.g., HTML documents)

- Resources have a uniform resource location (URL):



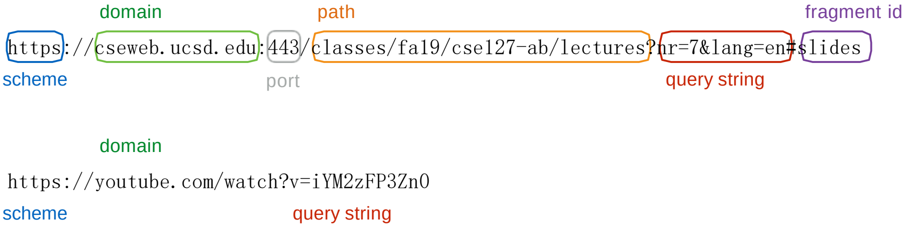
domain path

https://cseweb.ucsd.edu:443/classes/fa19/cse127-ab/lectures?nr=7&lang=en#slides

scheme port query string

HTTP protocol

- Protocol from 1989 that allows fetching of resources (e.g., HTML documents)
- Resources have a uniform resource location (URL):



domain

path

fragment id

https://cseweb.ucsd.edu:443/classes/fa19/cse127-ab/lectures?nr=7&lang=en#slides

scheme

port

query string

domain

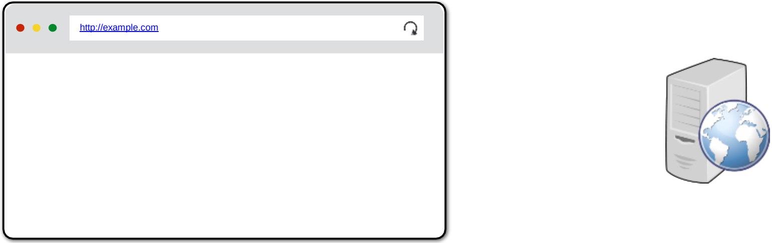
https://youtube.com/watch?v=iYM2zFP3Zn0

scheme

query string

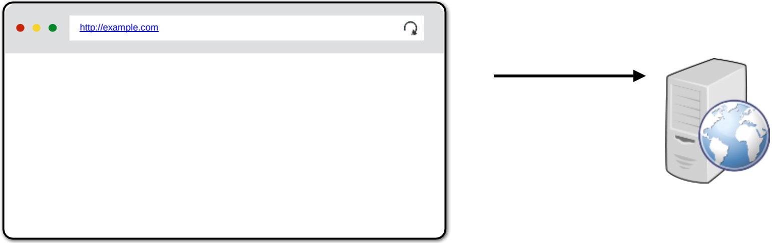
HTTP protocol

- Clients and servers communicate by exchanging individual messages (as opposed to a stream of data).



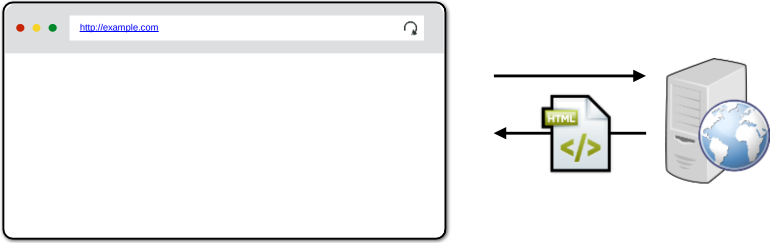
HTTP protocol

- Clients and servers communicate by exchanging individual messages (as opposed to a stream of data).



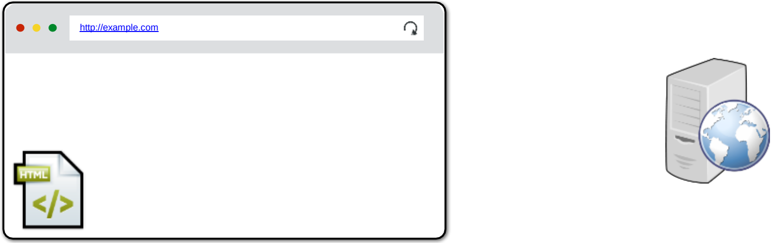
HTTP protocol

- Clients and servers communicate by exchanging individual messages (as opposed to a stream of data).



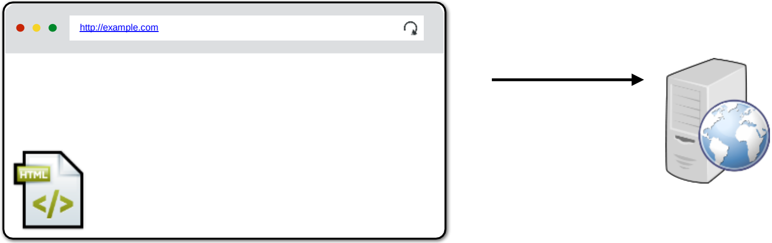
HTTP protocol

- Clients and servers communicate by exchanging individual messages (as opposed to a stream of data).



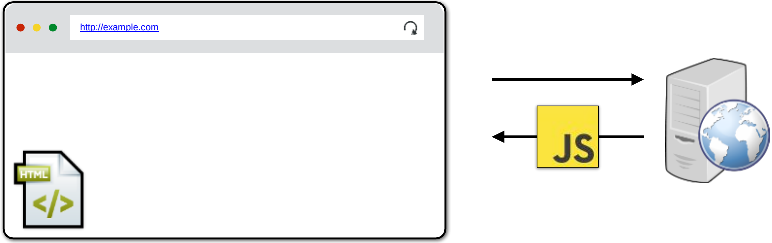
HTTP protocol

- Clients and servers communicate by exchanging individual messages (as opposed to a stream of data).



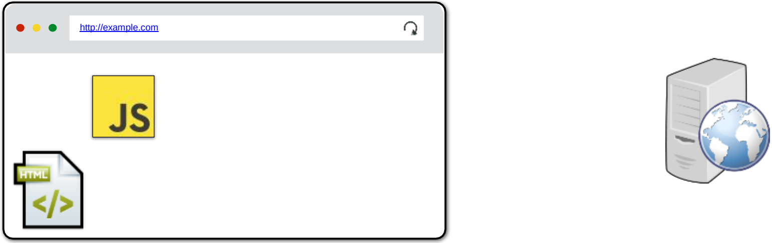
HTTP protocol

- Clients and servers communicate by exchanging individual messages (as opposed to a stream of data).



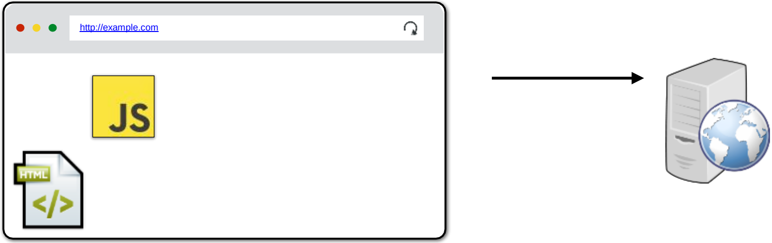
HTTP protocol

- Clients and servers communicate by exchanging individual messages (as opposed to a stream of data).



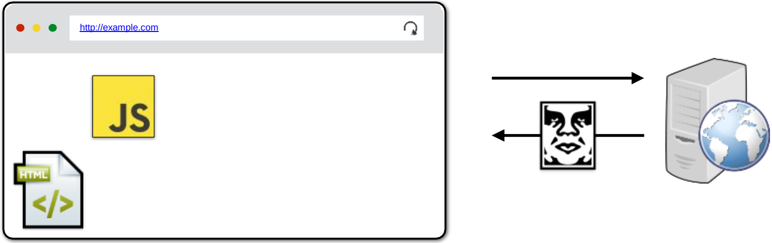
HTTP protocol

- Clients and servers communicate by exchanging individual messages (as opposed to a stream of data).



HTTP protocol

- Clients and servers communicate by exchanging individual messages (as opposed to a stream of data).



HTTP protocol

- Clients and servers communicate by exchanging individual messages (as opposed to a stream of data).

http://example.com
JS
HTML
</>

Anatomy of a request

```
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
```

Anatomy of a request

method

```
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
```

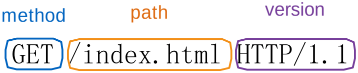
Anatomy of a request

method path

```
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
```

Anatomy of a request



method

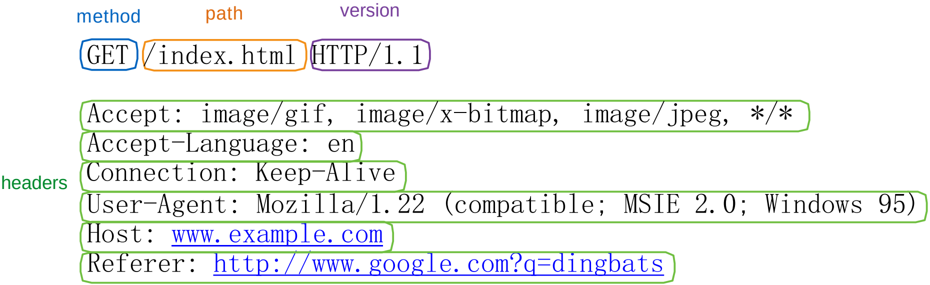
path

version

```
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
```

Anatomy of a request



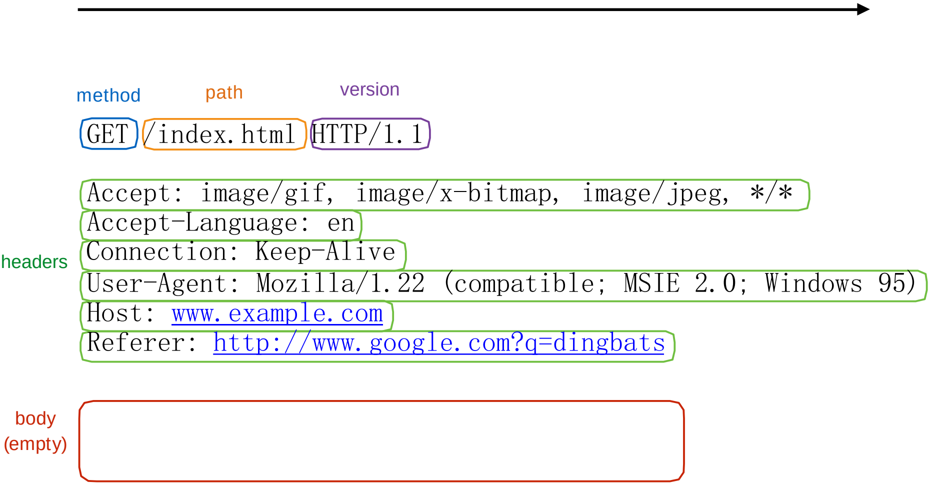
method path version

```
GET /index.html HTTP/1.1
```

headers

```
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
```

Anatomy of a request



```
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
```

Anatomy of a response

```
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... whatever ... </html>
```

Anatomy of a response

status code

```
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... whatever ... </html>
```

Anatomy of a response

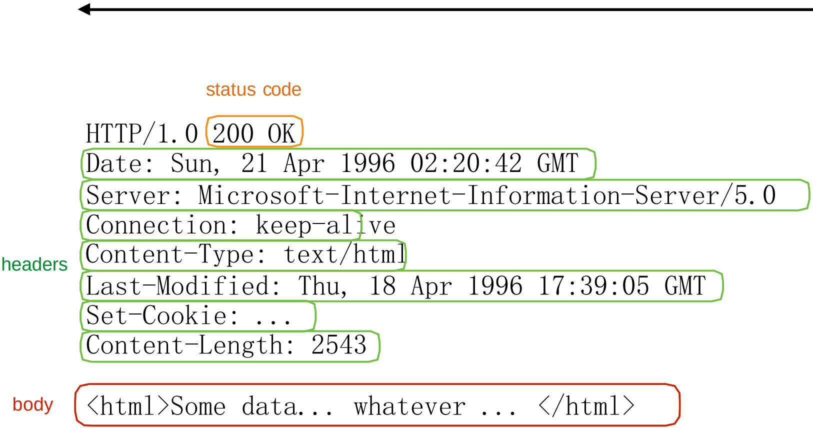
status code

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... whatever ... </html>
```

Anatomy of a response



status code

headers

body

```
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... whatever ... </html>
```

Many HTTP methods

- GET: Get the resource at the specified URL.
- POST: Create new resource at URL with payload.
- PUT: Replace current representation of the target resource with request payload.
- PATCH: Update part of the resource.
- DELETE: Delete the specified URL.

In practice: it’s a mess

- GETs should NOT change server state; in practice, they sometimes do
- Old browsers don’t send PUT, PATCH, and DELETE
  - So, almost all side-effecting requests are POSTs; real method hidden in a header or request body

In practice: we need state

In practice: we need state

- HTTP cookie: small piece of data that a server sends to the browser, who stores it and sends it back with subsequent requests
- What is this useful for?

In practice: we need state

- HTTP cookie: small piece of data that a server sends to the browser, who stores it and sends it back with subsequent requests
- What is this useful for?
  - Session management: logins, shopping carts, etc.
  - Personalization: user preferences, themes, etc.
  - Tracking: recording and analyzing user behavior

Setting cookies in response

```
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: trackingID=3272923427328234
Set-Cookie: userID=F3D947C2
Content-Length: 2543

<html>Some data... whatever ... </html>
```

Setting cookies in response

```
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: trackingID=3272923427328234
Set-Cookie: userID=F3D947C2
Content-Length: 2543

<html>Some data... whatever ... </html>
```

Sending cookie with each request

```
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)
Cookie: trackingID=3272923427328234
Cookie: userID=F3D947C2
Host: www.example.com
Referer: http://www.google.com?q=dingbats
```

Sending cookie with each request

```
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)
Cookie: trackingID=3272923427328234
Cookie: userID=F3D947C2
Host: www.example.com
Referer: http://www.google.com?q=dingbats
```

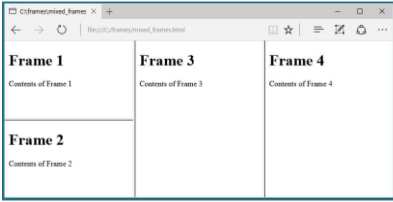
Going from HTTP response to code execution…

Basic browser execution model

- Each browser window….
  - Loads content
  - Parses HTML and runs Javascript
  - Fetches sub resources (e.g., images, CSS, JavaScript)
  - Respond to events like onClick, onMouseover, onLoad, setTimeout

Nested execution model

- Windows may contain frames from different sources
  - ➢ Frame: rigid visible division
  - ➢ iFrame: floating inline frame
- Why use frames?



Frame 1

Contents of Frame 1

Frame 2

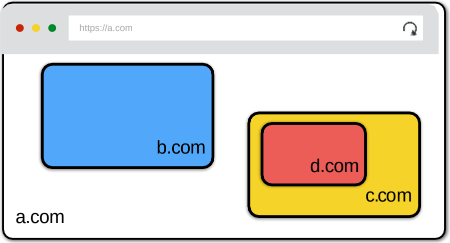
Contents of Frame 2

Frame 3

Contents of Frame 3

Frame 4

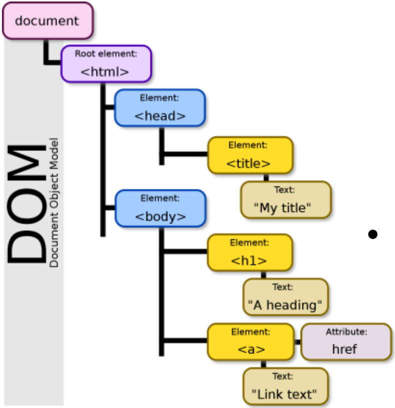
Contents of Frame 4



Nested execution model

- Windows may contain frames from diff sources
  - Frame: rigid visible division
  - iFrame: floating inline frame
- Why use frames?
  - Delegate screen area to content from another source
  - Browser provides isolation based on frames
  - Parent may work even if frame is broken

Document object model (DOM)

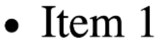


- Javascript can read and modify page by interacting with DOM
  - ➢ OO interface for reading and writing website content
- Includes browser object model
  - ➢ Access window, document, and other state like history, browser navigation, and cookies

https://en.wikipedia.org/wiki/Document_Object_Model

Modifying the DOM using JS

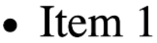
```
<html>
  <body>
    <ul id=“t1”>
      <li>Item 1</li>
    </ul>
...
  </body>
</html>
```



- Item 1

Modifying the DOM using JS

```
<html>
  <body>
    <ul id=“t1”>
      <li>Item 1</li>
    </ul>
...
  </body>
</html>
```

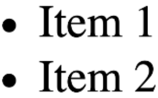


- Item 1

```
        <script>
          const list    = document.getElementById( ‘t1’);
          const newItem = document.createElement( ‘li’ );
          const newText = document.createTextNode( ‘Item 2’ );
          list.appendChild(newItem);
          newItem.appendChild(newText)
        </script>
```

Modifying the DOM using JS

```
<html>
  <body>
    <ul id= “t1” >
      <li>Item 1</li>
    </ul>
...
  </body>
</html>
```

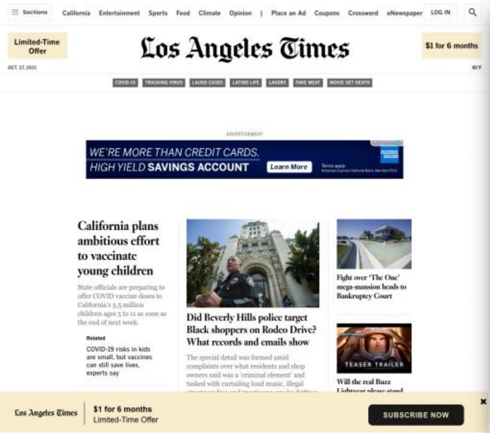


- Item 1
- Item 2

```
<script>
  const list    = document.getElementById( ‘t1’);
  const newItem = document.createElement( ‘li’ );
  const newText = document.createTextNode( ‘Item 2’ );
  list.appendChild(newItem);
  newItem.appendChild(newText)
</script>
```

Modern websites are complicated

Modern websites are complicated



Los Angeles Times

OCT. 27, 2021

81°F

California plans ambitious effort to vaccinate young children

State officials are preparing to offer COVID vaccine doses to California's 3.5 million children ages 5 to 11 as soon as the end of next week.

Related

COVID-19 risks in kids are small, but vaccines can still save lives, experts say

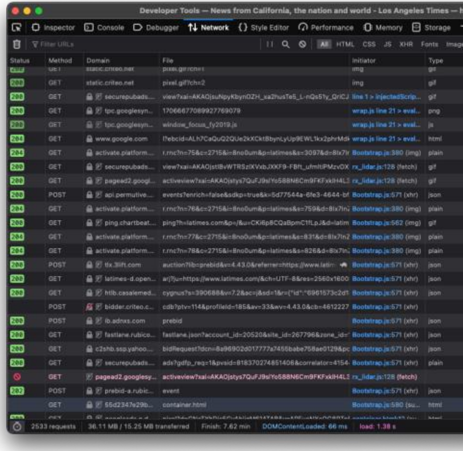
Did Beverly Hills police target Black shoppers on Rodeo Drive? What records and emails show

The special detail was formed amid complaints over what residents and shop owners said was a 'criminal element' and tasked with curtailing loud music, illegal

Fight over 'The One' mega-mansion heads to Bankruptcy Court

Will the real Buzz Lightyear please stand

Los Angeles Times | $1 for 6 months
Limited-Time Offer


Developer Tools — News from California, the nation and world - Los Angeles Times

Status	Method	Domain	File	Initiator	Type
200	GET	static.criteo.net	pixel.gif?ch=1	img	gif
200	GET	static.criteo.net	pixel.gif?ch=2	img	gif
200	GET	securepubads...	view?xai=AKAOjsuNpyKbynOZH_xa2husTe5_L-nQs51y_QrICJ	line 1 > injectedScrip...	gif
200	GET	tpc.googlesyn...	17066677089927769079	wrap.js line 21 > eval...	png
200	GET	tpc.googlesyn...	window_focus_fy2019.js	wrap.js line 21 > eval...	js
204	GET	www.google.com	l?ebcid=ALh7CaQuQ2QUe2kXCktBbynLyUp9EWL1kx2phrMdk	wrap.js line 21 > eval...	html
204	GET	activate.platform...	r.rnc?n=75&c=2715&i=8no0um&p=latimes&s=3097&d=8lx7lr	Bootstrap.js:380 (img)	plain
200	GET	securepubads...	view?xai=AKAOjstIBvWTRSzlXVxbJXKF9-FBft_ufmltPMzvOX	rx_lidar.js:128 (fetch)	gif
200	GET	pagead2.googl...	activeview?xai=AKAOjstys7QuFJ9slYo588N6Cm9FKFxklH4L:	rx_lidar.js:128 (fetch)	gif
200	POST	api.permutive...	events?enrich=false&sdkp=true&k=5d77544a-6fe3-4644-bf	Bootstrap.js:571 (xhr)	json
204	GET	activate.platform...	r.rnc?n=76&c=2715&i=8no0um&p=latimes&s=759&d=8lx7ln:	Bootstrap.js:380 (img)	plain
200	GET	ping.chartbeat....	ping?h=latimes.com&p=/&u=CKi6p8CQaBpmC1fLpJ&d=latim	Bootstrap.js:562 (img)	gif
204	GET	activate.platform...	r.rnc?n=77&c=2715&i=8no0um&p=latimes&s=831&d=8lx7ln2	Bootstrap.js:380 (img)	plain
204	GET	activate.platform...	r.rnc?n=78&c=2715&i=8no0um&p=latimes&s=826&d=8lx7ln:	Bootstrap.js:380 (img)	plain
200	POST	tlx.3lift.com	auction?lib=prebid&v=4.43.0&referrer=https://www.latim	Bootstrap.js:571 (xhr)	json
200	GET	latimes-d.open...	arj?ju=https://www.latimes.com/&ch=UTF-8&res=2560x1600	Bootstrap.js:571 (xhr)	json
200	GET	htlb.casalemed...	cygnus?s=390688&v=7.2&ac=j&sd=1&r={"id":"6961573c2d1	Bootstrap.js:571 (xhr)	json
	POST	bidder.criteo.c...	cdb?ptv=114&profileId=185&av=33&wv=4.43.0&cb=4612227	Bootstrap.js:571 (xhr)	
200	POST	ib.adnxs.com	prebid	Bootstrap.js:571 (xhr)	json
200	GET	fastlane.rubico...	fastlane.json?account_id=20520&site_id=267796&zone_id='	Bootstrap.js:571 (xhr)	json
200	GET	c2shb.ssp.yahoo...	bidRequest?dcn=8a96902d017777a7455babe758ae0129&pc	Bootstrap.js:571 (xhr)	json
200	GET	securepubads...	ads?gdfp_req=1&pvsid=818370274851408&correlator=4154	Bootstrap.js:571 (xhr)	plain
🚫	GET	pagead2.googlesy...	activeview?xai=AKAOjstys7QuFJ9slYo588N6Cm9FKFxklH4L:	rx_lidar.js:128 (fetch)	
202	POST	prebid-a.rubic...	event	Bootstrap.js:571 (xhr)	json
	GET	55d2347e29b...	container.html	Bootstrap.js:580 (su...	html

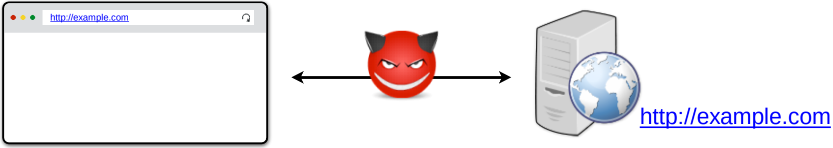
2533 requests | 36.11 MB / 15.25 MB transferred | Finish: 7.62 min | DOMContentLoaded: 66 ms | load: 1.38 s

Lecture objectives

- Basic understanding of how the web works
- Understand relevant attacker models
- Understand browser same-origin policy

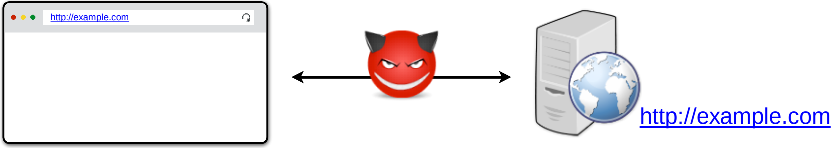
Relevant attacker models

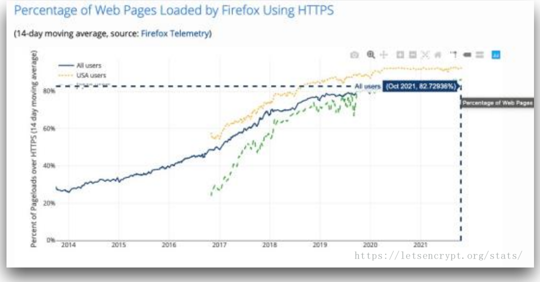
Network attacker



Relevant attacker models

Network attacker



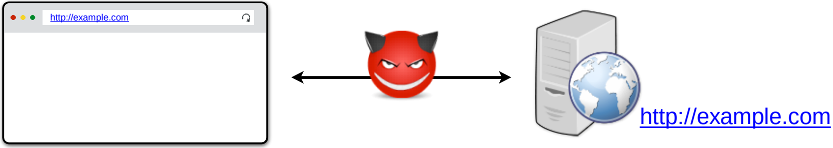

Percentage of Web Pages Loaded by Firefox Using HTTPS

(14-day moving average, source: Firefox Telemetry)

https://letsencrypt.org/stats/

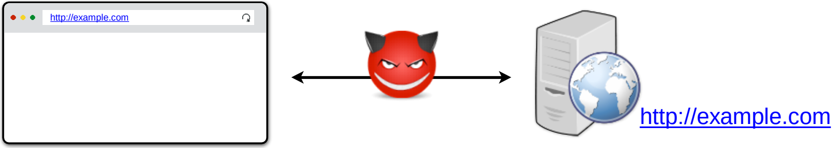
Relevant attacker models

Network attacker



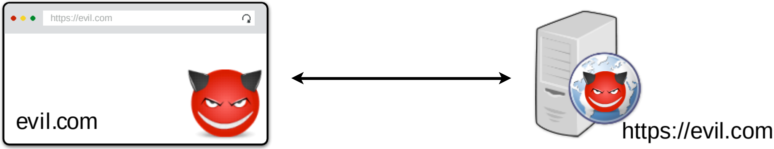
Relevant attacker models

Network attacker



http://example.com

Web attacker



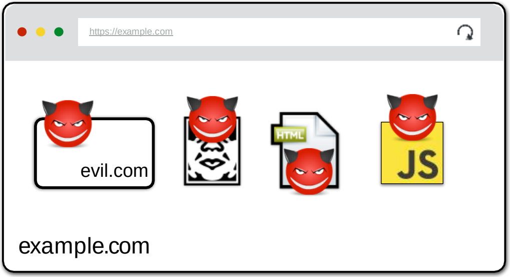
evil.com

https://evil.com

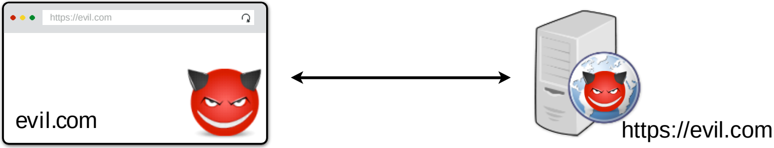
Relevant attacker models

Gadget attacker

Web attacker with capabilities to inject limited content into honest page



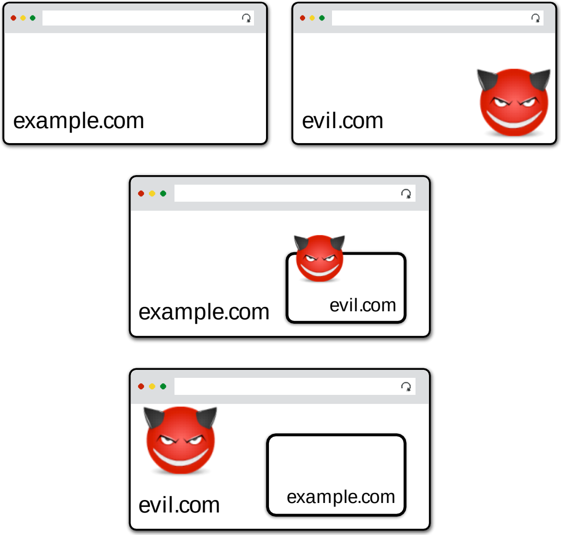
Most of our focus: web attacker



evil.com

https://evil.com

And variants of it



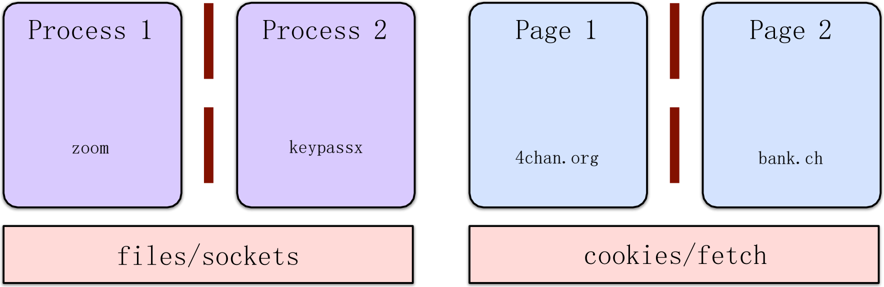
Lecture objectives

- Basic understanding of how the web works
- Understand relevant attacker models
- Understand browser same-origin policy

Web security model

Safely browse the web in the presence of attackers

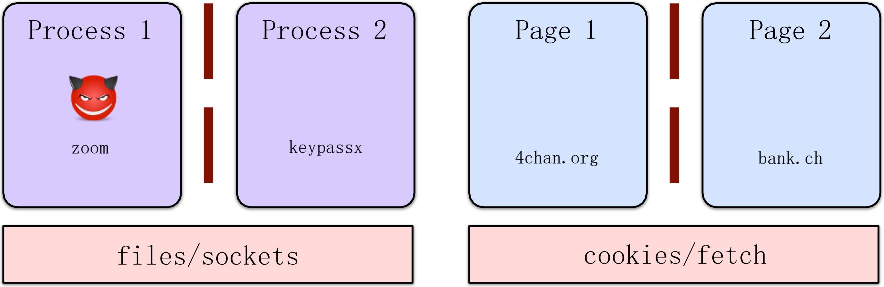
- The browser is the new OS analogy



Web security model

Safely browse the web in the presence of attackers

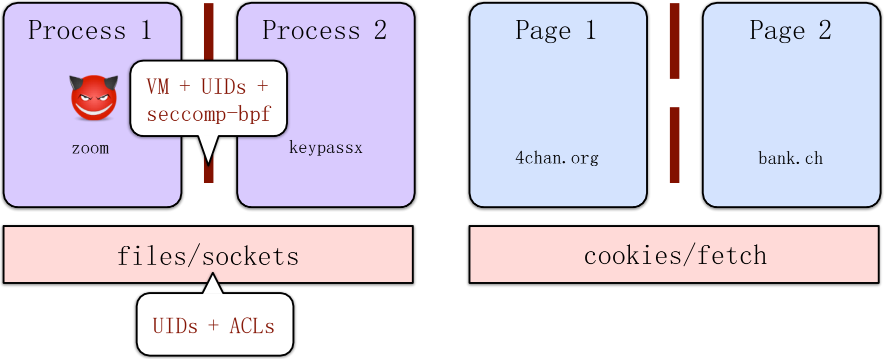
- The browser is the new OS analogy



Web security model

Safely browse the web in the presence of attackers

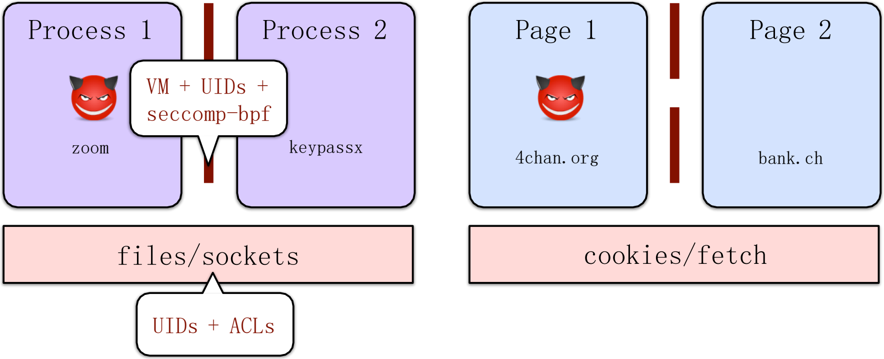
- The browser is the new OS analogy



Web security model

Safely browse the web in the presence of attackers

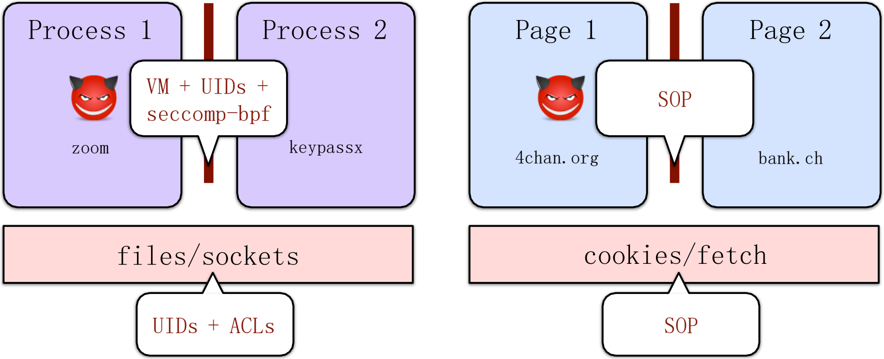
- The browser is the new OS analogy



Web security model

Safely browse the web in the presence of attackers

- The browser is the new OS analogy



Same origin policy (SOP)

- Origin: isolation unit/trust boundary on the web
  - (scheme, domain, port) triple derived from URL
- SOP goal: isolate content of different origins
  - **Confidentiality**: script contained in evil.com should not be able to read data in bank.ch page
  - **Integrity**: script from evil.com should not be able to modify the content of bank.ch page

There is no one SOP

- There is a same-origin policy for…
  - the DOM
  - message passing (via postMessage)
  - network access
  - CSS and fonts
  - cookies