The security model of the Web

Peter Cosemans - Michiel Olijslagers
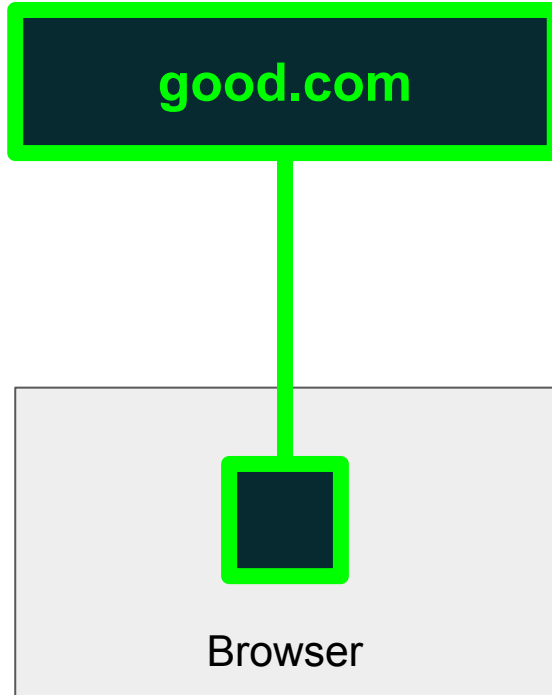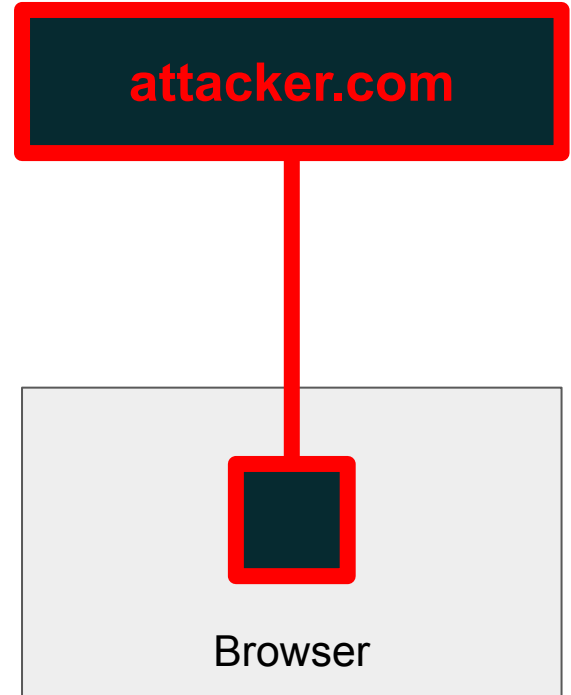
Attack models

Attack models

- Malicious server attacks the browser
- Server attacks other open web sites
- Malicious client attacks server
- Network attacks
- Web script injection attacks

# Attack models: Ideal situation

Attack models: Malicious server attacks the browser

- Example attack:
  - Drive by downloads
- Browser should protect the users local device from malicious web content
- Safe API design (no general-purpose only site-specific access)
  - File system API
 - Networking API
  - GUI API

attacker.com

Browser

Attack models: Server attacks other open web sites

- Example attacks
  - CSRF
- Countermeasures
  - Same-origin policy

**attacker.com**

good.com

Browser
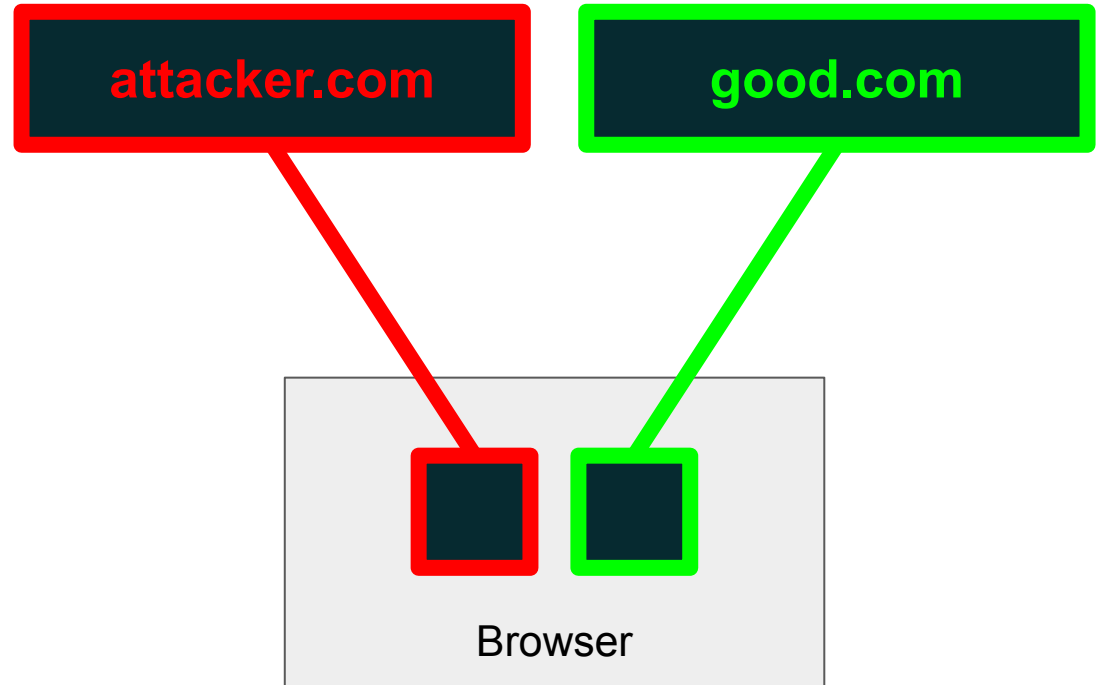
Attack models: Malicious client attacks server

- Example attacks:
    - SQL injection
    - Path injection
    - Command injections
- Countermeasures:
    - Access control
    - Defensive coding

good.com

Browser
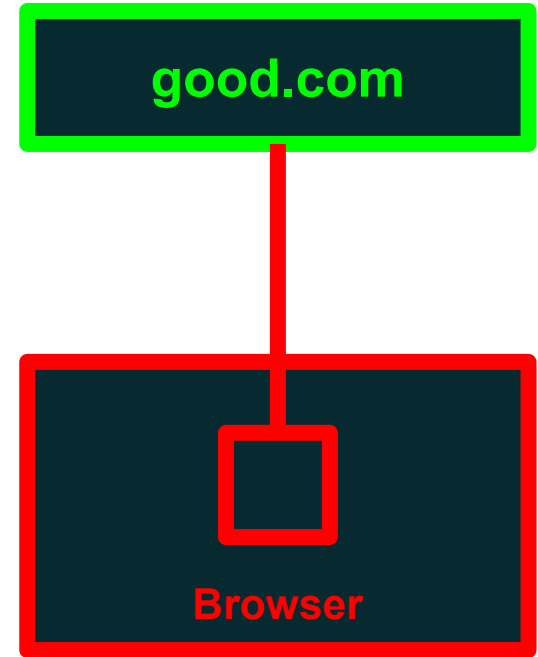
Attack models: Network attacks

- Example attacks:
    - SSL stripping
    - Attacks on public key infrastructure
- Countermeasures:
    - TLS/HTTPS

good.com

attacker.com
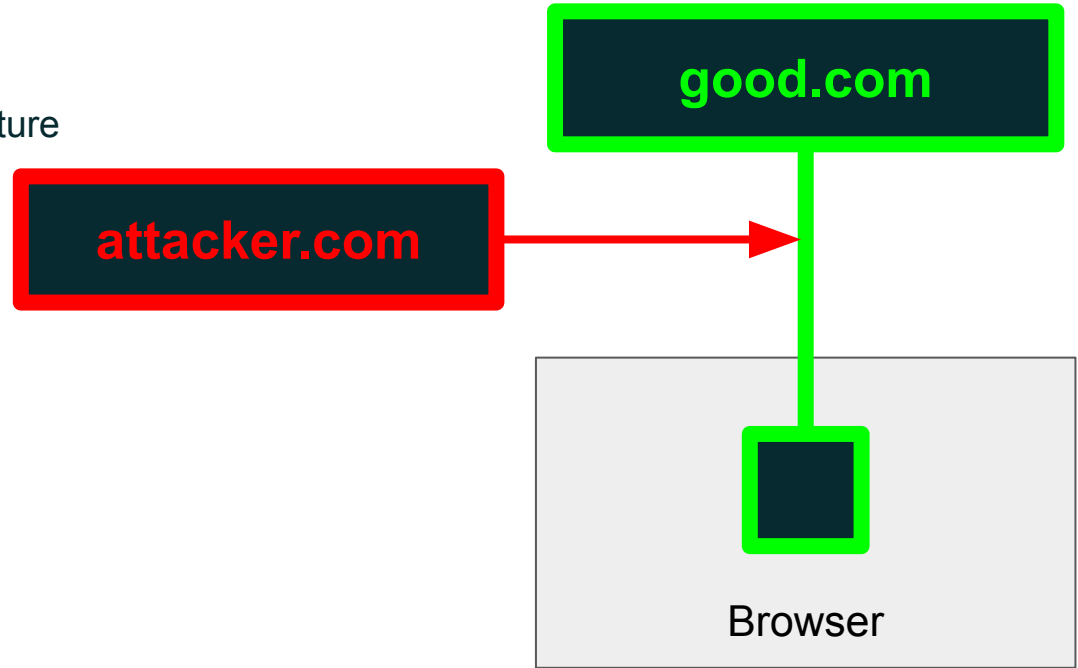
Browser

Attack models: Web script injection attacks

- Example attacks:
  - Inject a script: XSS
  - Distributing a malicious ad
  - Hacking a website that hosts a widely used script
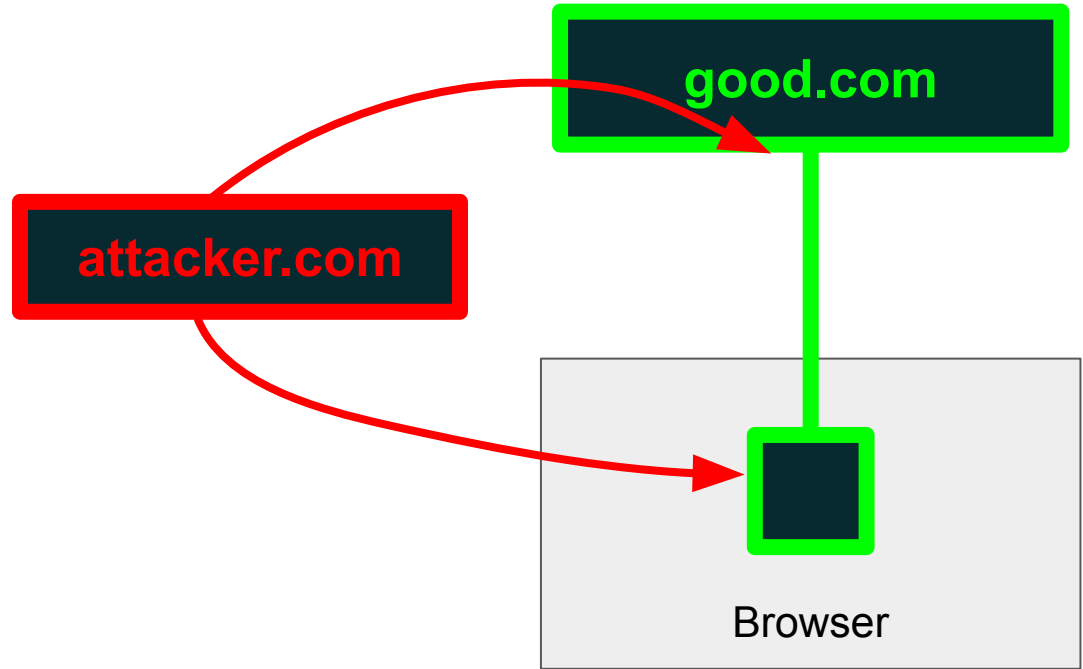
Browser

Major browsers

Chrome   Edge   Opera   Brave   Arc   Opera   Firefox   Tor
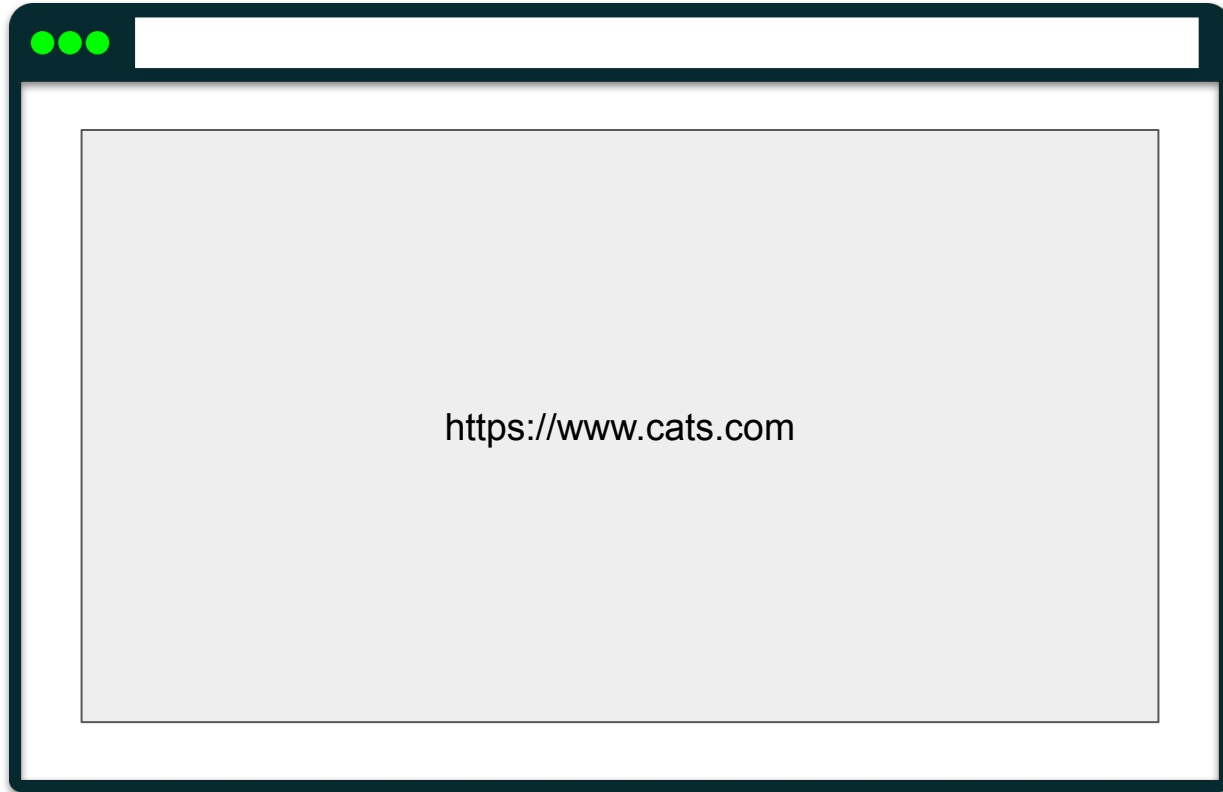
Chromium   Webkit   Gecko

https://dev.to/caffiendkitten/how-do-browser-make-websites-3709

https://developer.chrome.com/blog/inside-browser-part1/

https://www.lambdatest.com/blog/browser-engines-the-crux-of-cross-browser-compatibility/

The browser

The browser

https://www.cats.com

The browser

https://www.cats.com

The browser

https://www.cats.com
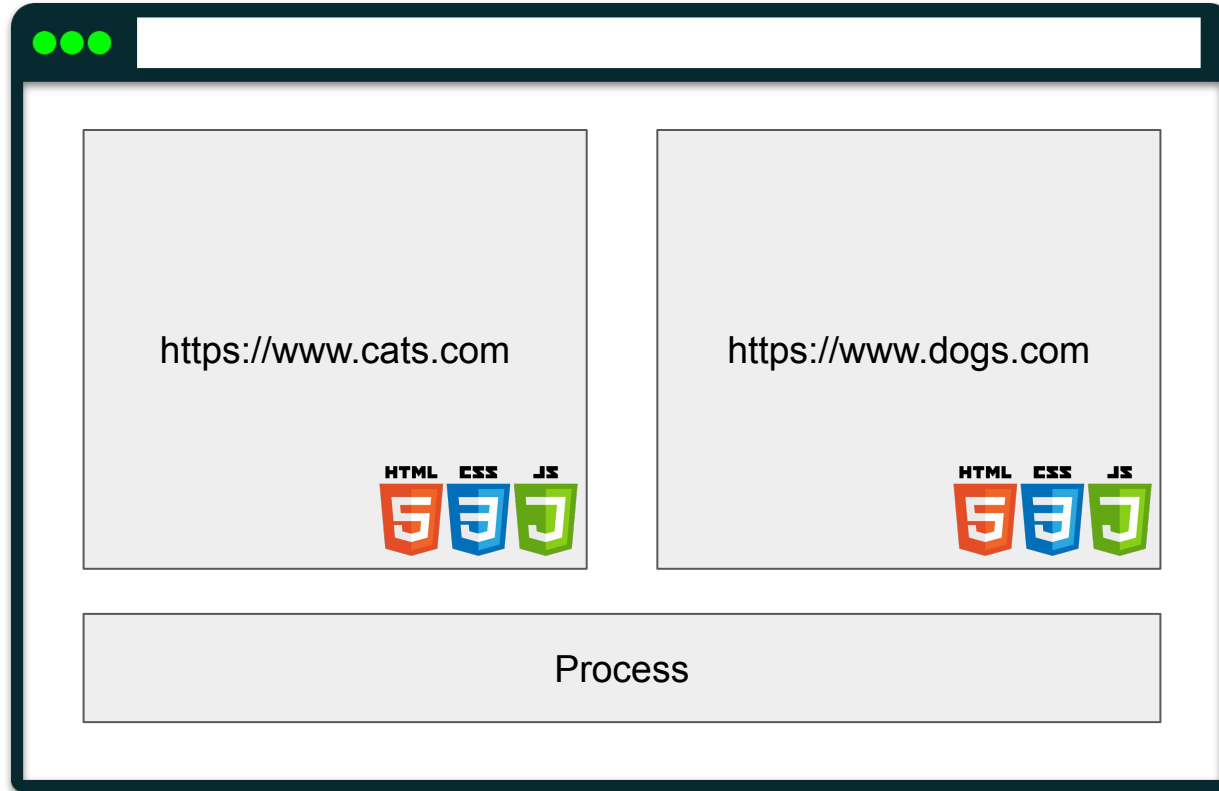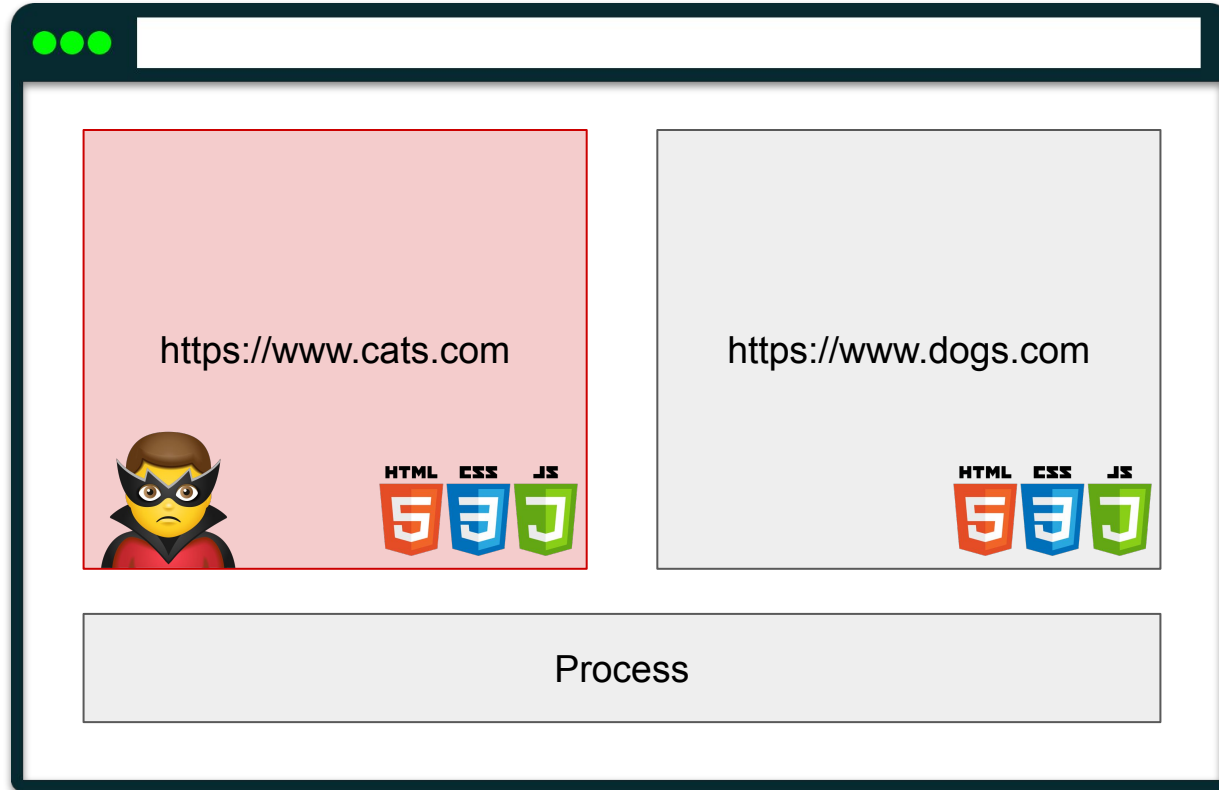
https://www.dogs.com

The browser

https://www.cats.com
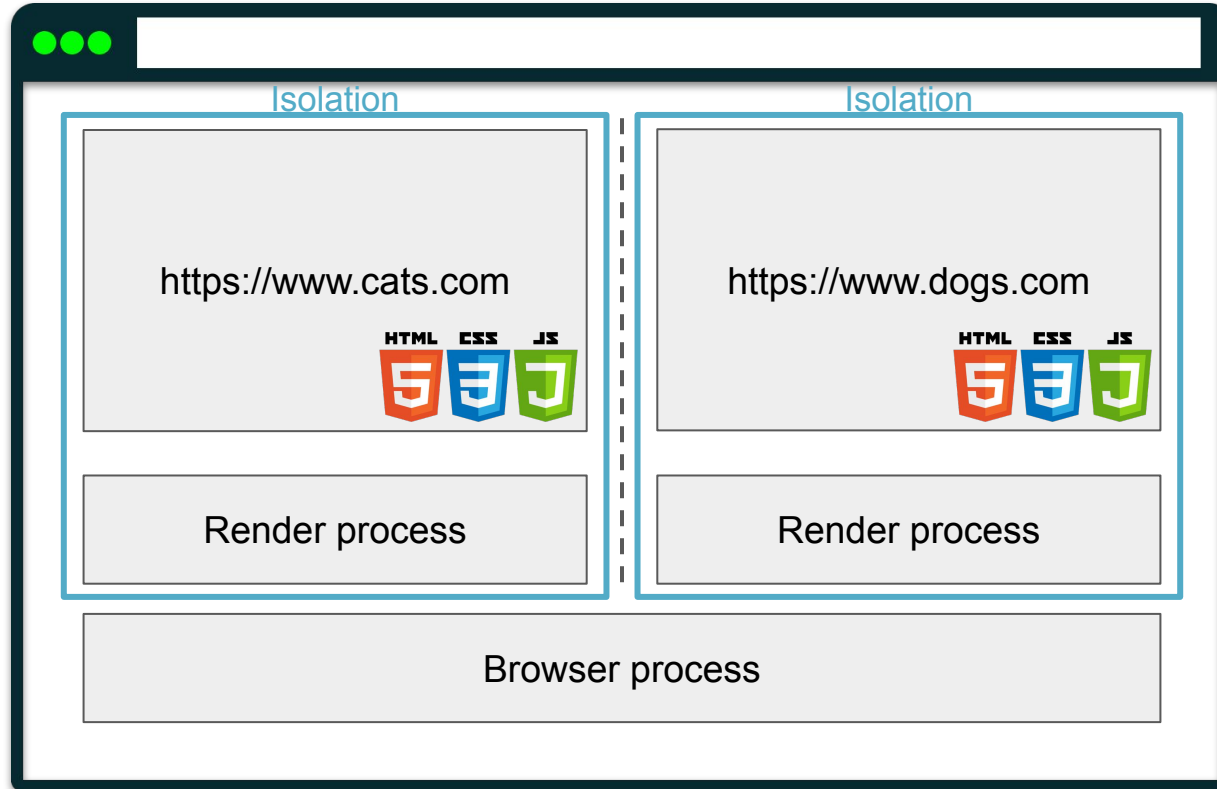
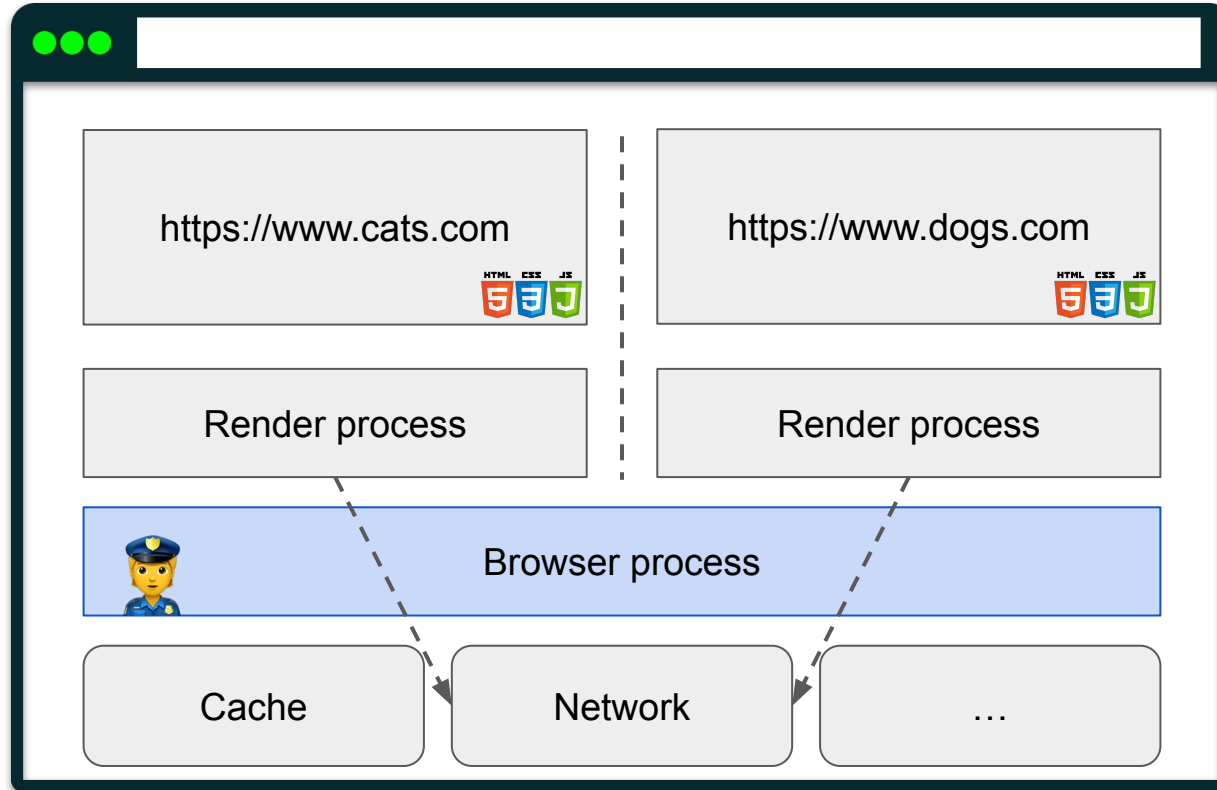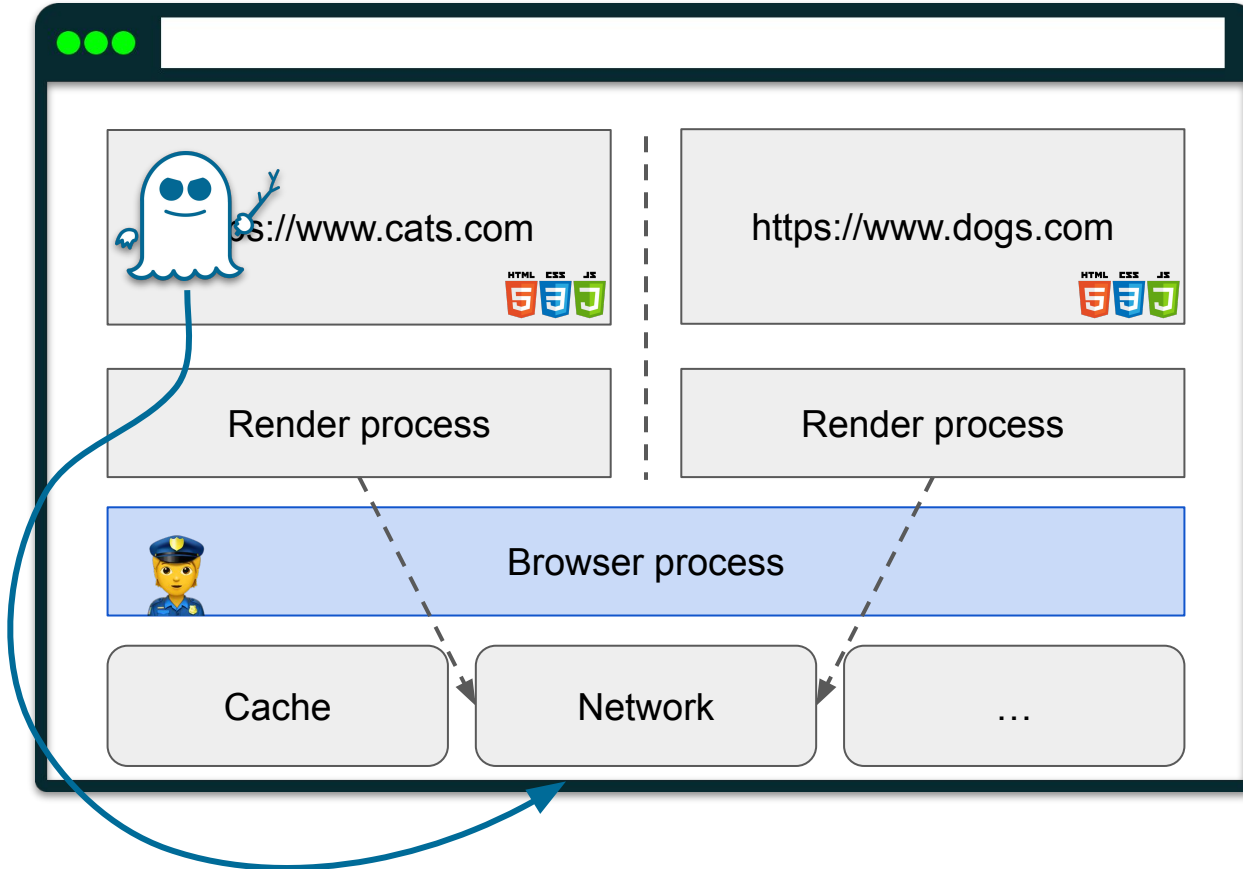https://www.dogs.com

Process

The browser

The browser

The browser - site isolation
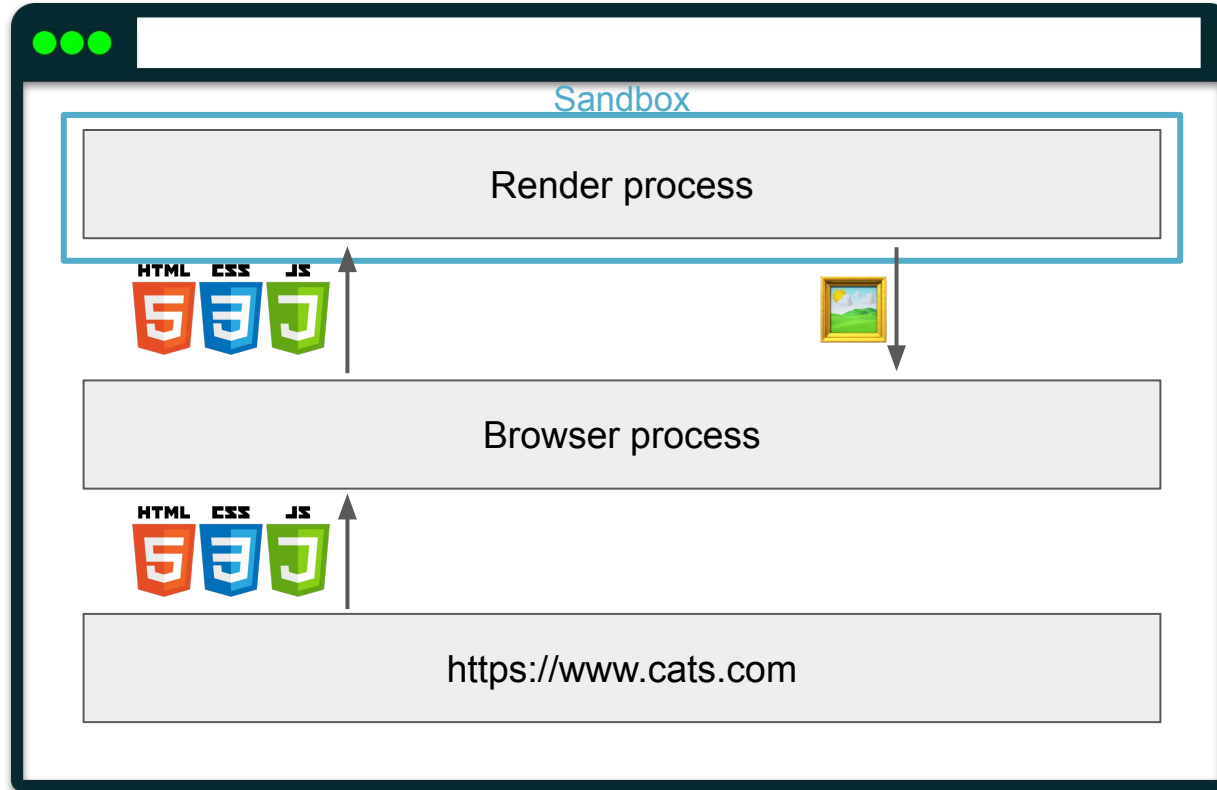
The browser - site isolation

The browser - site isolation

The browser - iFrame
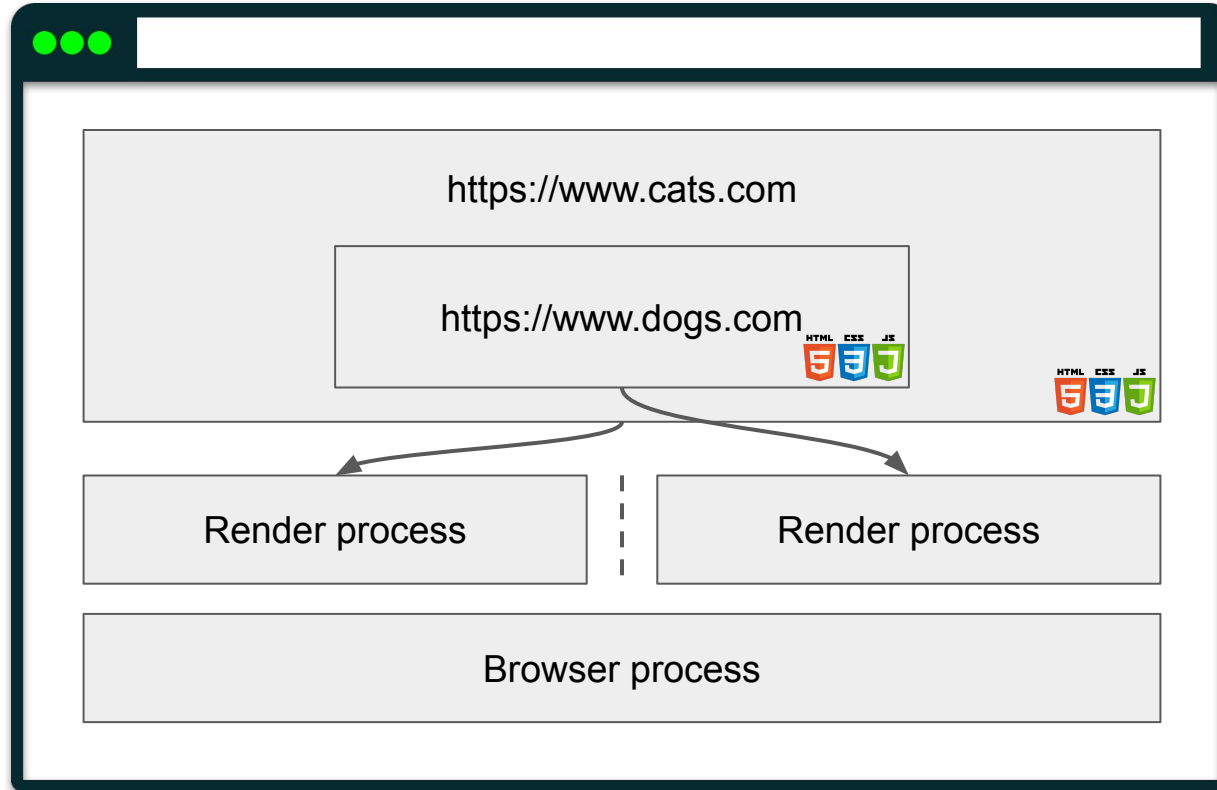
Origin vs site

Definitions

https://www.euri.com:433/bootcamp?language=js#contact

| scheme | host | port | path | query | fragment |

Origin

https://www.euri.com:433/bootcamp?language=js#contact

scheme   host   port   path   query   fragment

origin

Same-origin policy

https://euri.com

https://euri.com ✅

https://deskreservation.euri.com

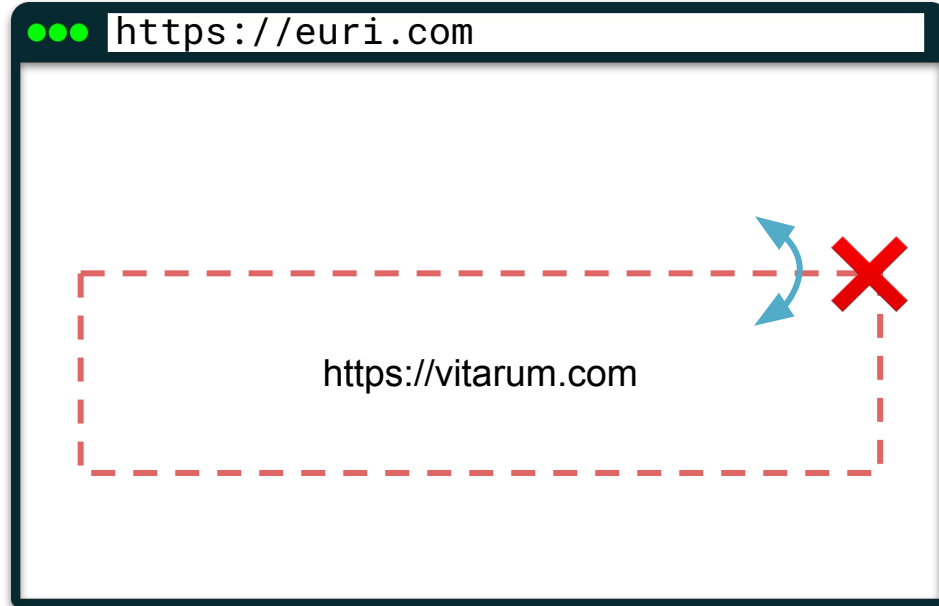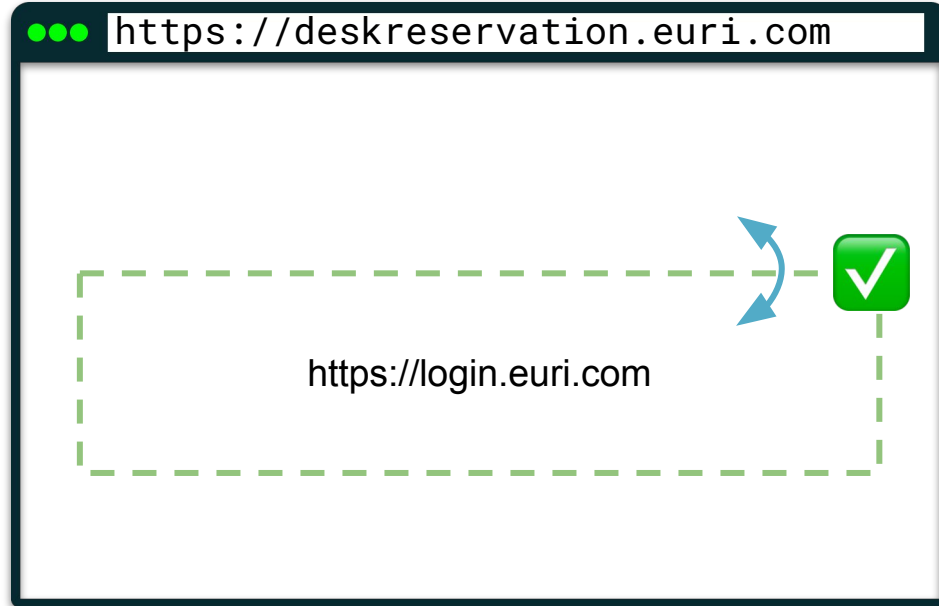https://login.euri.com ❌
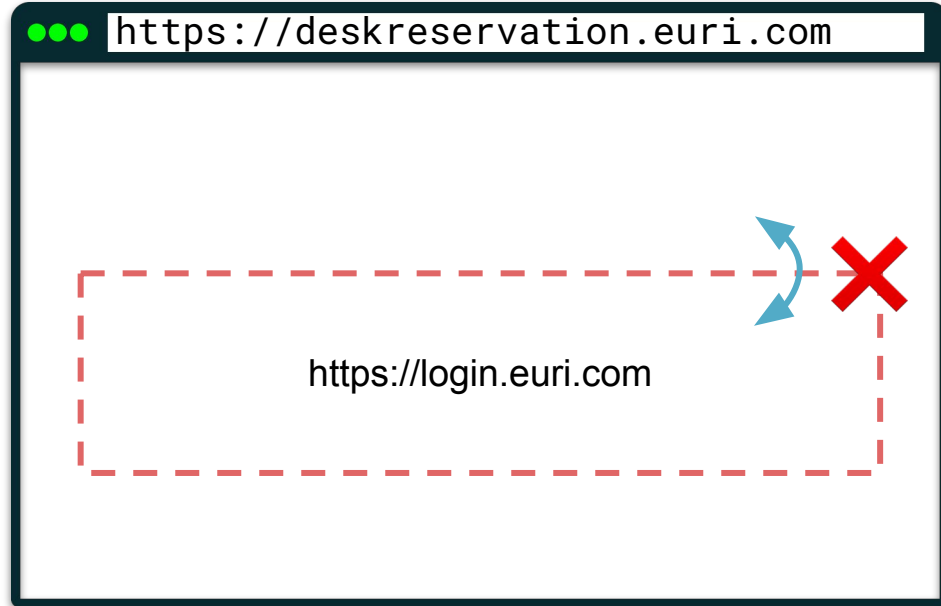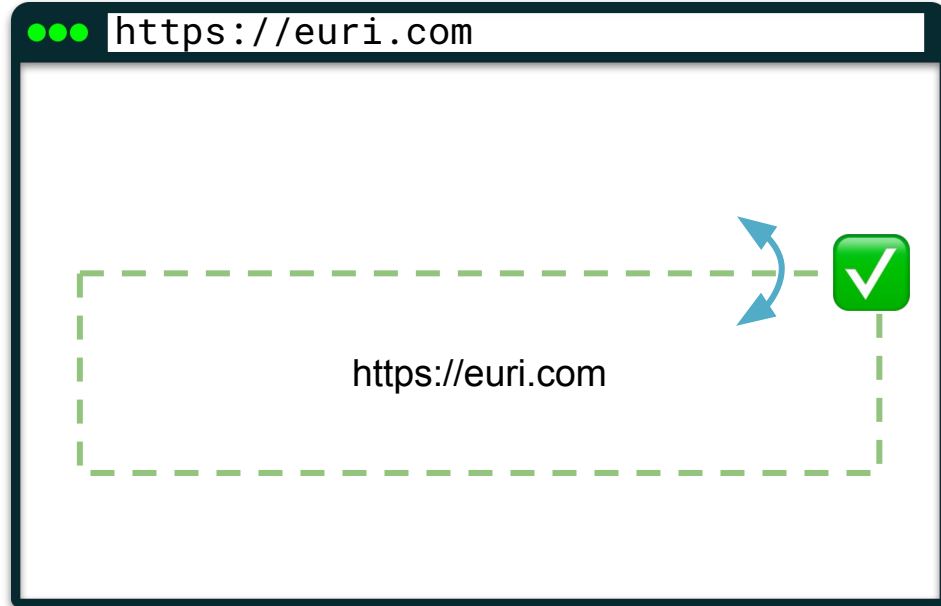
Site

- eTLD + 1
    - Example: euri.com, euri.co.uk
- Subdomains: cross origin but not cross site

site

https://www.euri.com:433/bootcamp?language=js#contact

scheme   host   port   path   query   fragment

origin

Same-site

https://deskreservation.euri.com

https://login.euri.com ✅

https://euri.com

https://vitarum.com ❌

Which are cross origin?

https://sec.euri.com/jobs

1. https://sec.euri.com/about
2. http://sec.euri.com/jobs
3. https://sec.euri.com:2800/jobs
4. https://app.euri.com/jobs

Which are cross site?

https://sec.euri.com/jobs

1. https://www.euri.com/jobs
2. https://app.euri.com
3. https://vitarum.com
4. https://app.sec.euri.com/jobs

Solutions

	Origin	Site
https://euri.com/jobs	(https, euri.com, 443)	euri.com
https://euri.com/about	(https, euri.com, 443)	euri.com
https://sec.euri.com	(https, sec.euri.com, 443)	euri.com
https://vitarum.com/jobs	(https, vitarum.com, 443)	viatrum.com

Key takeaways

Key takeaways

Browser isolation is important

Origins and sites are 2 different animals

Further reading

Chromium, **Site Isolation Design Document**
(https://www.chromium.org/developers/design-documents/site-isolation/)

Google, **Google chrome the comic**
(https://www.google.com/googlebooks/chrome/big_00.html)

Charles Reis, Alexander Moshchuk, and Nasko Oskov 2019, **Site Isolation: Process Separation for Web Sites within the Browser**
(https://www.usenix.org/conference/usenixsecurity19/presentation/reis)

Eric Rescorla, **Understanding The Web Security model**   *Very interesting read*
(https://educatedguesswork.org/posts/web-security-model-intro1/)