Strategies for Securing Web Content

Session 207

John Wilander, Safari and WebKit Engineer
Take the Swede’s Advice 🇸🇪
Yes, This Presentation Is for You
Agenda
Agenda

Secure transport
Agenda

Secure transport

Cross-origin lockdown
Agenda

Secure transport

Cross-origin lockdown

What you’re defending against
- Cross-origin attacks
- Speculative execution attacks
- Window control attacks
Agenda

Secure transport

Cross-origin lockdown

What you’re defending against
  • Cross-origin attacks
  • Speculative execution attacks
  • Window control attacks
Secure Transport

Your ToDo list

HTTPS and WSS

Your web content
Secure Transport

Your ToDo list

HTTPS and WSS

https:// Your web content
Secure Transport
Your ToDo list

HTTPS and WSS

Strict Transport Security (HSTS)
• Auto-upgrades your domain
Secure Transport
Your ToDo list

HTTPS and WSS

Strict Transport Security (HSTS)
- Auto-upgrades your domain

Upgrade Insecure Requests (UIR)
- Auto-upgrades cross-origin loads
Secure Transport
Your ToDo list

HTTPS and WSS

Strict Transport Security (HSTS)
• Auto-upgrades your domain

Upgrade Insecure Requests (UIR)
• Auto-upgrades cross-origin loads

Secure Cookies
• Are never sent in plaintext

https:// Your web content

https:// Cross-origin image
Secure Transport
Your ToDo list
Secure Transport

Your ToDo list

<table>
<thead>
<tr>
<th>Key</th>
<th>Type</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Information Property List</td>
<td>Dictionary</td>
<td></td>
</tr>
<tr>
<td>App Transport Security Settings</td>
<td>Dictionary</td>
<td></td>
</tr>
<tr>
<td>Allow Arbitrary Loads in Web Content</td>
<td>Boolean</td>
<td>NO</td>
</tr>
<tr>
<td>Exception Domains</td>
<td>Dictionary</td>
<td></td>
</tr>
</tbody>
</table>
Agenda

Secure transport

Cross-origin lockdown

What you’re defending against

• Cross-origin attacks
• Speculative execution attacks
• Window control attacks
Agenda

Secure transport

Cross-origin lockdown

What you’re defending against
  • Cross-origin attacks
  • Speculative execution attacks
  • Window control attacks
Cross-Origin Loads

Your web content
Cross-Origin Loads

Your web content

Cross-origin image
Cross-Origin Loads

Cross-origin image

Cross-origin script

Your web content

```javascript
function onload() {
  ...
}
```
Cross-Origin Loads

Cross-origin image

Cross-origin script

Cross-origin iframe

Your web content

function onload() {
  ...
}

Comment

Post
Cross-Origin Loads

- cloud.example/beach.png
- cdn.example/framework.js
- social.example/comment.html

Your web content
- Cross-origin image
- Cross-origin script
- Cross-origin iframe
Cross-Origin Loads

- cdn.example/beach.png
- cdn.example/framework.js
- social.example/comment.html

Your web content

- Cross-origin image
- Cross-origin script
- Cross-origin iframe
Cross-Origin Lockdown

Cross-origin loads

Your web content

Same-origin policy

Cross-origin iframe
Cross-Origin Lockdown

Cross-origin loads

Your web content
Cross-Origin Lockdown

Cross-origin loads

social.example’s content

Your web content
Cross-Origin Lockdown
Cross-Origin Lockdown

Subresource Integrity
Content Security Policy
HttpOnly cookies
SameSite cookies
Cross-Origin-Resource-Policy
Cross-Origin-Window-Policy
Cross-Origin Lockdown

**Subresource Integrity**
Simple markup change to script tags

Content Security Policy

HttpOnly cookies

SameSite cookies

Cross-Origin-Resource-Policy

Cross-Origin-Window-Policy
Cross-Origin Lockdown

Subresource Integrity

Content Security Policy

HttpOnly cookies

SameSite cookies

Cross-Origin-Resource-Policy

Cross-Origin-Window-Policy

Simple markup change to script tags

Server configuration + architectural changes
Cross-Origin Lockdown

Subresource Integrity: Simple markup change to script tags

Content Security Policy: Server configuration + architectural changes

HttpOnly cookies

SameSite cookies: Simple server configuration changes

Cross-Origin-Resource-Policy

Cross-Origin-Window-Policy
Cross-Origin Lockdown

Subresource Integrity
Content Security Policy
HttpOnly cookies
SameSite cookies
Cross-Origin-Resource-Policy
Cross-Origin-Window-Policy
Cross-Origin Lockdown
Subresource integrity

Your web content

<script src=...>
Cross-Origin Lockdown
Subresource integrity

<script src="https://cdn.example/framework.js"></script>
Cross-Origin Lockdown
Subresource integrity

```html
<script src="https://cdn.example/framework.js"
integrity="sha256-8WqyJLuWKRB...oZkCnxQbWwJVw=">
</script>
```
Cross-Origin Lockdown
Subresource integrity

```html
<script src="https://cdn.example/framework.js"
  integrity="sha256-8WqyJLuWKRB...oZkCnxQbWwJVw=">
</script>

window.framework || // reload from own domain
```
Cross-Origin Lockdown

Subresource Integrity
Content Security Policy
HttpOnly cookies
SameSite cookies
Cross-Origin-Resource-Policy
Cross-Origin-Window-Policy
Cross-Origin Lockdown
Content security policy

HTTP response:
:status: 200
Content-Security-Policy:
Cross-Origin Lockdown
Content security policy

HTTP response:
:status: 200
Content-Security-Policy:
  default-src 'self'; // No inline
Cross-Origin Lockdown
Content security policy

HTTP response:
:status: 200
Content-Security-Policy:
default-src 'self';
script-src cdn.example;

cdn.example/framework.js
Cross-Origin Lockdown

Content security policy

HTTP response:
:status: 200
Content-Security-Policy:
default-src 'self';
script-src cdn.example;
frame-src social.example;
Cross-Origin Lockdown
Content security policy

HTTP response:
:status: 200
Content-Security-Policy:
default-src 'self';
script-src cdn.example;
frame-src social.example;
frame-ancestors news.example;
Cross-Origin Lockdown
Content security policy

HTTP response:
:status: 200
Content-Security-Policy:
  default-src 'self';
  script-src cdn.example;
  frame-src social.example;
  frame-ancestors news.example;
Cross-Origin Lockdown
Content security policy

Content-Security-Policy:
default-src 'self';

Lock all resource loads to your own origin by default
Cross-Origin Lockdown
Content security policy

Content-Security-Policy:

script-src cdn.example;

Allow cdn.example for script loads on your page
Cross-Origin Lockdown
Content security policy

HTTP response:
status: 200
Content-Security-Policy:
default-src 'self';
script-src cdn.example;
frame-src social.example;
frame-ancestors news.example;

Allow social.example to be iframed on your page
Cross-Origin Lockdown

Content security policy

```
Content-Security-Policy:
frame-ancestors news.example;
```

Allow news.example to iframe your page
Cross-Origin Lockdown

Subresource Integrity
Content Security Policy
HttpOnly cookies
SameSite cookies
Cross-Origin-Resource-Policy
Cross-Origin-Window-Policy
Cross-Origin Lockdown

Subresource Integrity
Content Security Policy
HttpOnly cookies
SameSite cookies
Cross-Origin-Resource-Policy
Cross-Origin-Window-Policy
Cross-Origin Lockdown
HttpOnly cookies

<script>
document.cookie
</script>
Cross-Origin Lockdown
HttpOnly cookies

HTTP response:
:status: 200
Set-Cookie:
  auth=abc...123; HttpOnly

<script>
document.cookie
</script>
Cross-Origin Lockdown
HttpOnly cookies

HTTP response:
:status: 200
Set-Cookie: auth=abc...123; HttpOnly

<script>
document.cookie
</script>
Cross-Origin Lockdown

Content Security Policy
Subresource Integrity
HttpOnly cookies
SameSite cookies
Cross-Origin-Resource-Policy
Cross-Origin-Window-Policy
Cross-Origin Lockdown
SameSite cookies

HTTP response:

:status: 200
Set-Cookie: 
  auth=abc...123; HttpOnly;
  SameSite=strict
Cross-Origin Lockdown

SameSite cookies
Cross-Origin Lockdown
SameSite cookies

Your server

Someone else's content

Your web content
Cross-Origin Lockdown

Content Security Policy
Subresource Integrity
HttpOnly cookies
SameSite cookies
Cross-Origin-Resource-Policy
Cross-Origin-Window-Policy
Cross-Origin Lockdown
Cross-Origin-Resource-Policy

Your server

function onload() {
  ...
}

Your server

NEW

Your web content
Cross-Origin Lockdown
Cross-Origin-Resource-Policy

Your server

HTTP response:

status: 200
Cross-Origin-Resource-Policy: Same

Your web content

function onload() {
...
}

NEW
Cross-Origin Lockdown
Cross-Origin-Resource-Policy

Your server
HTTP response:
status: 200
Cross-Origin-Resource-Policy: Same

Your server
HTTP response:
status: 200
Cross-Origin-Resource-Policy: Same

Someone else’s content

function onload() {
    ...
}

NEW
Cross-Origin Lockdown
Cross-Origin-Resource-Policy

HTTP response:
:status: 200
Cross-Origin-Resource-Policy: Same

HTTP response:
:status: 200
Cross-Origin-Resource-Policy: Same

Someone else's content
Cross-Origin Lockdown

Content Security Policy
Subresource Integrity
HttpOnly cookies
SameSite cookies
Cross-Origin-Resource-Policy
Cross-Origin-Window-Policy
Cross-Origin Lockdown
Cross-Origin-Window-Policy

Someone else's content
Cross-Origin Lockdown
Cross-Origin-Window-Policy

New window with your content

Handle

Someone else’s content
Cross-Origin Lockdown
Cross-Origin-Window-Policy

HTTP response:
:status: 200
Cross-Origin-Window-Policy: Deny

New window with your content

Handle

Someone else’s content

NEW
Cross-Origin Lockdown
Cross-Origin-Window-Policy

HTTP response:
:status: 200
Cross-Origin-Window-Policy: Deny
Agenda

Secure transport

Cross-origin lockdown

What you’re defending against
  • Cross-origin attacks
  • Speculative execution attacks
  • Window control attacks
Agenda

Secure transport

Cross-origin lockdown

What you’re defending against

• Cross-origin attacks
• Speculative execution attacks
• Window control attacks
Cross-Origin Attacks

Cross-Site Scripting
Compromised CDN
Cross-Site Request Forgeries
Cross-Origin Attacks

Cross-Site Scripting
Compromised CDN
Cross-Site Request Forgeries
Cross-Origin Attacks

Cross-Site Scripting

Your web content

Message

Send
Cross-Origin Attacks
Cross-Site Scripting

Hello!
<script>
stead(
document.cookie
);
</script>
Cross-Origin Attacks
Cross-Site Scripting

HTTP response:
:status: 200
Set-Cookie:
  auth=abc...123; HttpOnly

Hello!
<script>
steal(
document.cookie
);
</script>
Cross-Origin Attacks
Cross-Site Scripting

Hello!
<script>
steal(
    document.cookie
);
</script>

HTTP response:
:status: 200
Set-Cookie:
    auth=abc...123; HttpOnly

🍪
Cross-Origin Attacks

Cross-Site Scripting

HTTP response:
:status: 200
Content-Security-Policy:
default-src 'self';

Hello!
<script>
steal(document.cookie);
</script>
Cross-Origin Attacks
Cross-Site Scripting

HTTP response:
:status: 200
Content-Security-Policy:
default-src 'self'; // No inline
Cross-Origin Attacks

Cross-Site Scripting

Compromised CDN

Cross-Site Request Forgeries
Cross-Origin Attacks

Cross-Site Scripting
Compromised CDN
Cross-Site Request Forgeries
Cross-Origin Attacks
Compromised CDN

cdn.example/framework.js

You wanted this

setUp(
  // Framework
);

Your web content
Cross-Origin Attacks
Compromised CDN

You got this instead

estail(
doocument.cookie
);
Cross-Origin Attacks
Compromised CDN

HTTP response:
:status: 200
Content-Security-Policy:
default-src 'self';
script-src cdn.example;

You got this instead

steal(
   document.cookie
);

Your web content
Cross-Origin Attacks
Compromised CDN

HTTP response:
:status: 200
Content-Security-Policy:
default-src 'self';
script-src cdn.example;
Cross-Origin Attacks
Compromised CDN

cdn.example/framework.js

```javascript
setUp(
    // Framework
);
```

Your web content
Cross-Origin Attacks

Compromised CDN

cdn.example/framework.js

Your web content

```javascript
steal(document.cookie);
```
Cross-Origin Attacks
Compromised CDN

<script src="https://cdn.example/framework.js" integrity="sha256-8WqyJLuWKRB...oZkCnxQbWwJVw=">
</script>

window.framework || // reload from own domain
Cross-Origin Attacks

Cross-Site Scripting
Compromised CDN
Cross-Site Request Forgeries
Cross-Origin Attacks

Cross-Site Scripting
Compromised CDN
Cross-Site Request Forgeries
Cross-Origin Attacks

Cross-Site Request Forgery

Your web content

Message

Send
Cross-Origin Attacks

Cross-Site Request Forgery
Cross-Origin Attacks

Cross-Site Request Forgery

evil.example’s content

Hidden resource load of your web content.
Cross-Origin Attacks
Cross-Site Request Forgery

Your server

evil.example’s content

Hidden resource load of your web content.

Send
Cross-Origin Attacks
Cross-Site Request Forgery

HTTP response:

:status: 200
Set-Cookie: auth=abc...123; SameSite=strict
Cross-Origin Attacks

Cross-Site Request Forgery

Hidden resource load of your web content.

Message

Send
Cross-Origin Attacks

Cross-Site Request Forgery

Your server

evil.example’s content

Message

Hidden resource load of your web content.
Agenda

Secure transport

Cross-origin lockdown

What you’re defending against

• Cross-origin attacks
• Speculative execution attacks
• Window control attacks
Agenda

Secure transport
Cross-origin lockdown
What you’re defending against
  • Cross-origin attacks
  • Speculative execution attacks
  • Window control attacks
Speculative Execution Attacks

Speculative Execution and the Attack (Spectre)

Defenses:
- WKWebView
- Content Security Policy
- HttpOnly cookies
- SameSite cookies
- Cross-Origin-Resource-Policy
Speculative Execution Defined

Is index x OK?
Speculative Execution Defined

Is index $x$ OK?

- Handle error
- Load array[$x$]
Speculative Execution Defined

Is index x OK?

Speculative

Handle error

Load array[x]
Speculative Execution Defined

- Is index x OK?
  - Handle error
  - Load array[x]
Speculative Execution Defined

- Is index x OK?
  - Handle error
Speculative Execution Attacks

Is index x OK?

- Handle error
- Load array[x]

Cache effects of the speculative execution can be measured and data leaked
Speculative Execution Attacks

Your web content

Same-Origin Policy

Cross-origin iframe
Speculative Execution Attacks

Your web content

Cross-origin iframe
Make sure your web content doesn’t end up in the same process as a frame from evil.example
Speculative Execution Attacks

Speculative Execution and the Attack (Spectre)

Defenses:

- WKWebView
- Content Security Policy
- HttpOnly cookies
- SameSite cookies
- Cross-Origin-Resource-Policy
WKWebView

- evil.example
- Your page
- WebKit
  - A fast, open source web browser engine.

Processes:
- evil.example content process
- Your web content process
- webkit.org content process
WebKit Architecture

Network process

- evil.example content process
- Your web content process
- webkit.org content process
WebKit Architecture

Network process

evil.example content process
UIWebView Versus WKWebView
UIWebView Versus WKWebView

UIWebView

evil.example content
UIWebView Versus WKWebView

UIWebView

evil.example content

Your web content
UIWebView Versus WKWebView

- Network
- evil.example content
- Your web content
UIWebView Versus WKWebView

Your app

UIWebView

Network

evil.example content

Your web content
UIWebView Versus WKWebView

Your app

UIWebView

Network

evil.example content

Your web content
UIWebView Versus WKWebView

Your app

Network

evil.example content

Your web content
UIWebView Versus WKWebView

WKWebView is out-of-process

Your app

Network

evil.example content

Your web content
Speculative Execution Attacks

Speculative Execution and the Attack (Spectre)

Defenses:
- WKWebView
- Content Security Policy
- HttpOnly cookies
- SameSite cookies
- Cross-Origin-Resource-Policy
Speculative Execution Attacks

Content security policy

Your web content

social.example
iframe
Speculative Execution Attacks
Content security policy

Injected iframe

Your web content

evil.example iframe

social.example iframe
Speculative Execution Attacks
Content security policy

Your web content

- ad.example
  iframe

- social.example
  iframe
Speculative Execution Attacks
Content security policy

Your web content

Redirected iframe

evil.example iframe

social.example iframe
Speculative Execution Attacks
Content security policy

HTTP response:
:status: 200
Content-Security-Policy:
default-src 'self';
Speculative Execution Attacks

Content security policy

HTTP response:
:status: 200
Content-Security-Policy:
  default-src 'self';
  frame-src ad.example
  social.example
Speculative Execution Attacks
Content security policy

evil.example’s content

Your web content
Speculative Execution Attacks
Content security policy

HTTP response:
:status: 200
Content-Security-Policy:
default-src 'self';
frame-ancestors 'none'

Your web content
Speculative Execution Attacks

Content security policy

HTTP response:
:status: 200
Content-Security-Policy:
  default-src 'self';
  frame-ancestors 'none'

Blocked in the Network Process

evil.example's content
Speculative Execution Attacks

Speculative Execution and the Attack (Spectre)

Defenses:
- WKWebView
- Content Security Policy
- HttpOnly cookies
- SameSite cookies
- Cross-Origin-Resource-Policy
Speculative Execution Attacks

HttpOnly cookies

evil.example’s content

Your web content
Speculative Execution Attacks
HttpOnly cookies

HttpOnly cookies stay in the network process
Speculative Execution and the Attack (Spectre)

Defenses:

- WKWebView
- Content Security Policy
- HttpOnly cookies
- SameSite cookies
- Cross-Origin-Resource-Policy
Speculative Execution Attacks
SameSite cookies

Your server

Your web content

evil.example’s content
Speculative Execution Attacks

SameSite cookies

Your server

Your web content
Speculative Execution Attacks

SameSite cookies

evil.example’s content
Speculative Execution Attacks

Speculative Execution and the Attack (Spectre)

Defenses:
- WKWebView
- Content Security Policy
- HttpOnly cookies
- SameSite cookies (new)
- Cross-Origin-Resource-Policy
Speculative Execution Attacks
Cross-Origin-Resource-Policy

HTTP response:
:status: 200
Cross-Origin-Resource-Policy: Same

function onload() {
  ...
}

Someone else's content
Speculative Execution Attacks
Cross-Origin-Resource-Policy

HTTP response:
:status: 200
Cross-Origin-Resource-Policy: Same

HTTP response:
:status: 200
Cross-Origin-Resource-Policy: Same

Someone else's content
Speculative Execution Attacks
Cross-Origin-Resource-Policy

Someone else’s content

Blocked in the Network Process

HTTP response:
:status: 200
Cross-Origin-Resource-Policy: Same

HTTP response:
:status: 200
Cross-Origin-Resource-Policy: Same
Agenda

Secure transport

Cross-origin lockdown

What you’re defending against

- Cross-Origin Attacks
- Speculative Execution Attacks
- Window Control Attacks
Agenda

Secure transport

Cross-origin lockdown

What you’re defending against

• Cross-Origin Attacks
• Speculative Execution Attacks
• Window Control Attacks
Window Control Attacks

Malicious window navigation

Defense:

• Cross-Origin-Window-Policy header
Window Control Attacks
Malicious window navigation

evil.example’s content

Your web content
Window Control Attacks
Malicious window navigation

Your server → Your web content → evil.example’s content
Handle
Window Control Attacks
Malicious window navigation

Your server -> Your web content <- evil.example’s content

Handle
Window Control Attacks
Malicious window navigation

Please reauthenticate

Username
Password
Send

evil.example’s content
Window Control Attacks

Malicious window navigation

Defense:

• Cross-Origin-Window-Policy header
Window Control Attacks

evil.example’s content

Your web content
Window Control Attacks

HTTP response:
:status: 200
Cross-Origin-Window-Policy: Deny

Your web content

evil.example’s content
Window Control Attacks

HTTP response:
:status: 200
Cross-Origin-Window-Policy: Deny

Your web content

Handle

evil.example’s content
Take Action
Take Action

Baseline

Use transport security (https, wss)
Mark cookies HttpOnly and Secure
Migrate from UIWebView to WKWebView
## Content Security Policy

<table>
<thead>
<tr>
<th>Issue</th>
<th>Status</th>
</tr>
</thead>
<tbody>
<tr>
<td>Cross-Site Scripting</td>
<td>✔️</td>
</tr>
<tr>
<td>Compromised CDN</td>
<td></td>
</tr>
<tr>
<td>Cross-Site Request Forgeries</td>
<td></td>
</tr>
<tr>
<td>Speculative Execution Attacks</td>
<td>✔️</td>
</tr>
<tr>
<td>Window Control Attacks</td>
<td></td>
</tr>
<tr>
<td>Issue</td>
<td>Content Security Policy</td>
</tr>
<tr>
<td>-----------------------------------</td>
<td>-------------------------</td>
</tr>
<tr>
<td>Cross-Site Scripting</td>
<td>✓</td>
</tr>
<tr>
<td>Compromised CDN</td>
<td></td>
</tr>
<tr>
<td>Cross-Site Request Forgeries</td>
<td></td>
</tr>
<tr>
<td>Speculative Execution Attacks</td>
<td>✓</td>
</tr>
<tr>
<td>Window Control Attacks</td>
<td></td>
</tr>
</tbody>
</table>
# Take Action

<table>
<thead>
<tr>
<th></th>
<th>Content Security Policy</th>
<th>HttpOnly Cookies</th>
<th>Subresource Integrity</th>
</tr>
</thead>
<tbody>
<tr>
<td>Cross-Site Scripting</td>
<td>✔️</td>
<td>✔️</td>
<td></td>
</tr>
<tr>
<td>Compromised CDN</td>
<td></td>
<td></td>
<td>✔️</td>
</tr>
<tr>
<td>Cross-Site Request Forgeries</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Speculative Execution Attacks</td>
<td>✔️</td>
<td>✔️</td>
<td></td>
</tr>
<tr>
<td>Window Control Attacks</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Take Action

<table>
<thead>
<tr>
<th>Attack Type</th>
<th>Content Security Policy</th>
<th>HttpOnly Cookies</th>
<th>Subresource Integrity</th>
<th>SameSite Cookies</th>
</tr>
</thead>
<tbody>
<tr>
<td>Cross-Site Scripting</td>
<td>✓</td>
<td>✓</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Compromised CDN</td>
<td></td>
<td></td>
<td></td>
<td>✓</td>
</tr>
<tr>
<td>Cross-Site Request Forgeries</td>
<td></td>
<td></td>
<td></td>
<td>✓</td>
</tr>
<tr>
<td>Speculative Execution Attacks</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td></td>
</tr>
<tr>
<td>Window Control Attacks</td>
<td></td>
<td></td>
<td></td>
<td>✓</td>
</tr>
</tbody>
</table>
## Take Action

<table>
<thead>
<tr>
<th>Attack Type</th>
<th>Content Security Policy</th>
<th>HttpOnly Cookies</th>
<th>Subresource Integrity</th>
<th>SameSite Cookies</th>
<th>Cross-Origin-Resource-Policy</th>
</tr>
</thead>
<tbody>
<tr>
<td>Cross-Site Scripting</td>
<td>✓</td>
<td>✓</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Compromised CDN</td>
<td></td>
<td></td>
<td></td>
<td>✓</td>
<td></td>
</tr>
<tr>
<td>Cross-Site Request Forgeries</td>
<td></td>
<td></td>
<td></td>
<td>✓</td>
<td></td>
</tr>
<tr>
<td>Speculative Execution Attacks</td>
<td>✓</td>
<td>✓</td>
<td></td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>Window Control Attacks</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
## Take Action

<table>
<thead>
<tr>
<th>Attack Type</th>
<th>Content Security Policy</th>
<th>HttpOnly Cookies</th>
<th>Subresource Integrity</th>
<th>SameSite Cookies</th>
<th>Cross-Origin-Resource-Policy</th>
<th>Cross-Origin-Window-Policy</th>
</tr>
</thead>
<tbody>
<tr>
<td>Cross-Site Scripting</td>
<td>✓</td>
<td>✓</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Compromised CDN</td>
<td></td>
<td></td>
<td></td>
<td>✓</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Cross-Site Request Forgeries</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>Speculative Execution Attacks</td>
<td>✓</td>
<td>✓</td>
<td></td>
<td></td>
<td></td>
<td>✓</td>
</tr>
<tr>
<td>Window Control Attacks</td>
<td></td>
<td></td>
<td></td>
<td>✓</td>
<td></td>
<td>✓</td>
</tr>
</tbody>
</table>
Summary

These security defenses are relatively easy to adopt.

Test to ensure the security is working and that your content still works.
https://webkit.org/blog/
<table>
<thead>
<tr>
<th>Event</th>
<th>Location</th>
<th>Date and Time</th>
</tr>
</thead>
<tbody>
<tr>
<td>Safari, WebKit, and Password AutoFill Lab</td>
<td>Technology Lab 3</td>
<td>Wednesday 2:00PM</td>
</tr>
<tr>
<td>What’s New in Safari and WebKit</td>
<td>Executive Ballroom</td>
<td>Friday 2:00PM</td>
</tr>
</tbody>
</table>

More Information
