#WWDC18

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

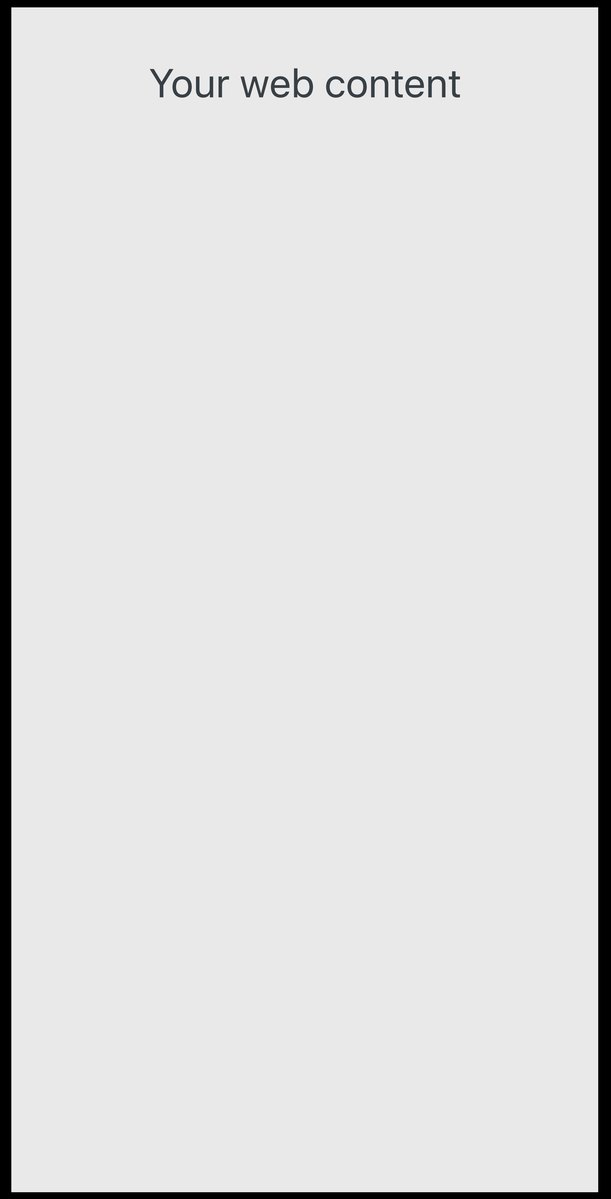
What you’re defending against

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

Secure Transport

Your ToDo list

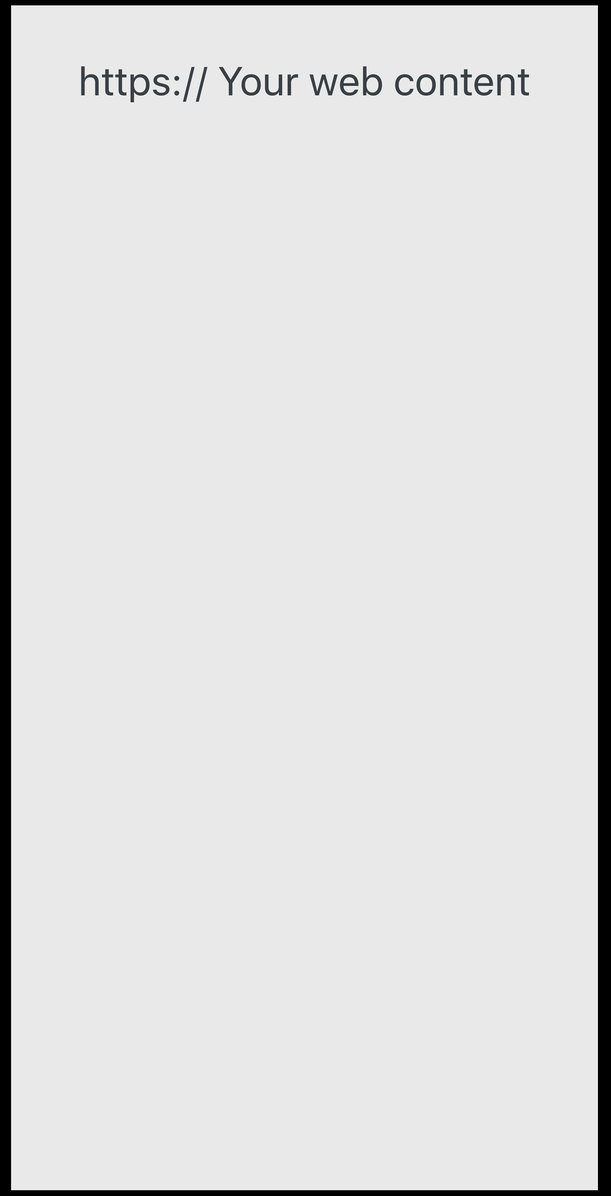
HTTPS and WSS



Secure Transport

Your ToDo list

HTTPS and WSS



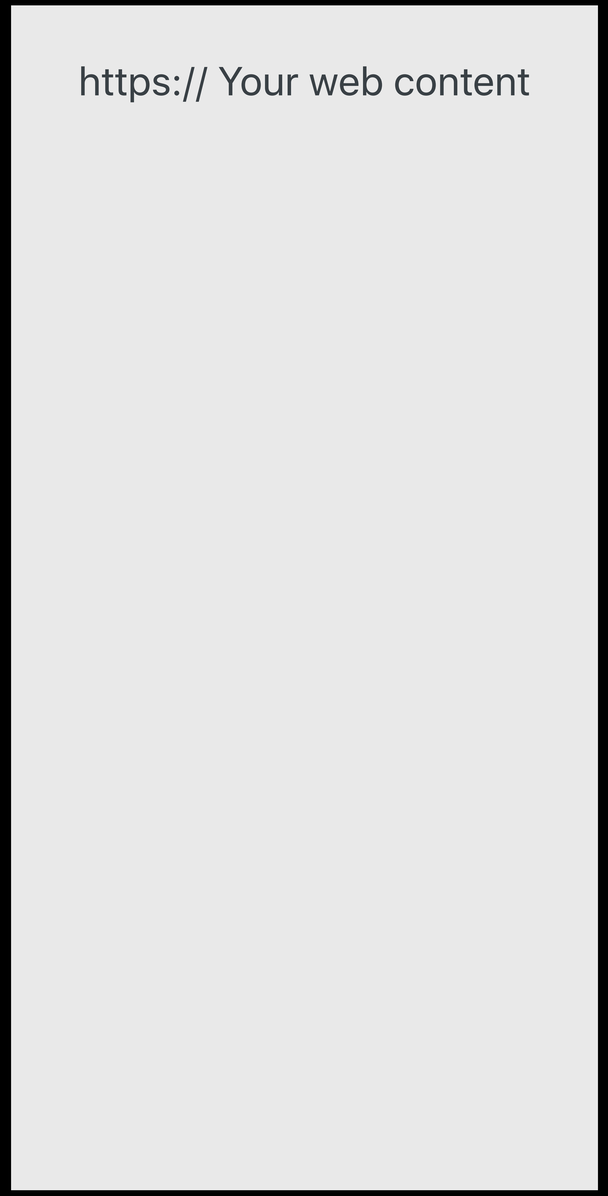
Secure Transport

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

https:// Cross-origin image

https:// Your web content

Secure Transport

Your ToDo list

HTTPS and WSS

Strict Transport Security (HSTS)

- Auto-upgrades your domain

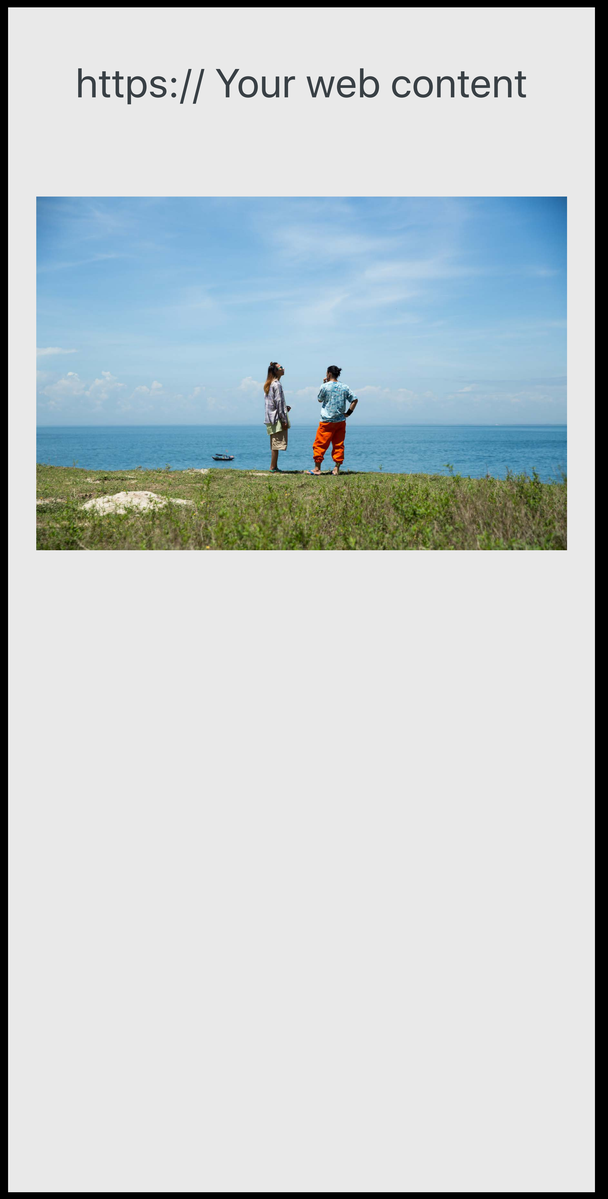
Upgrade Insecure Requests (UIR)

- Auto-upgrades cross-origin loads

Secure Cookies

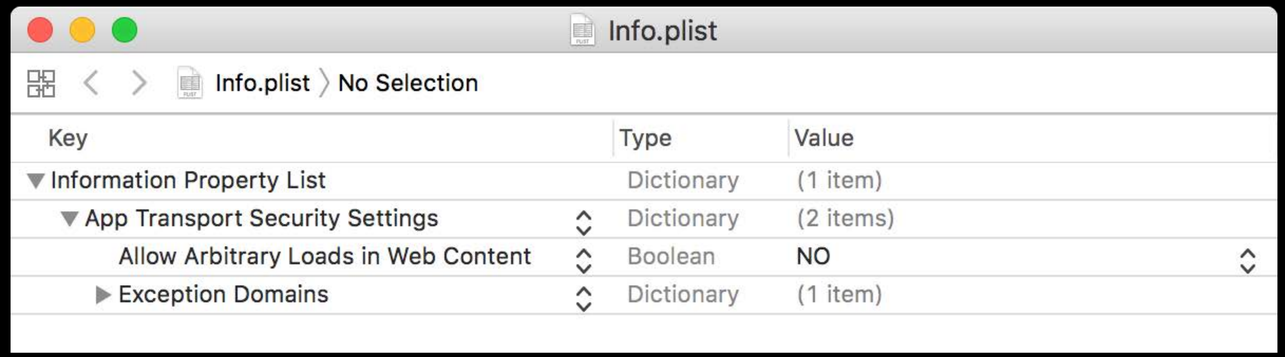
- Are never sent in plaintext

https:// Cross-origin image



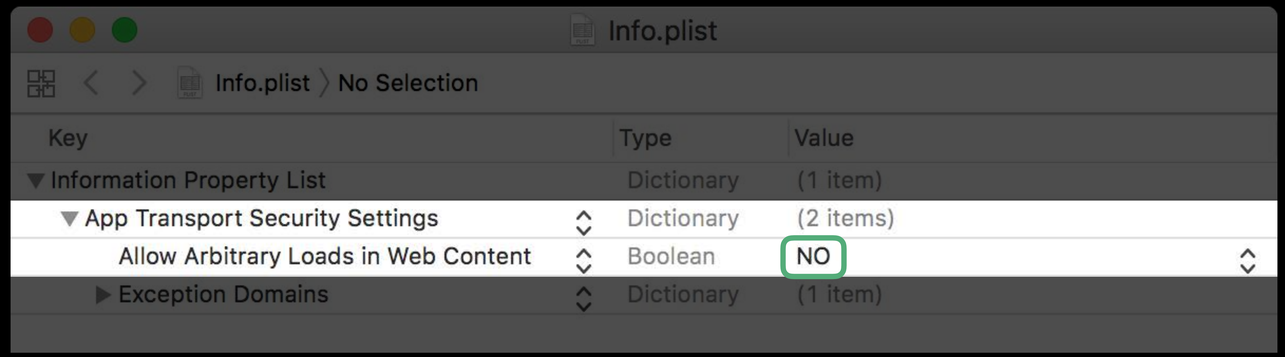
Secure Transport

Your ToDo list



Secure Transport

Your ToDo list



Agenda

Secure transport

Cross-origin lockdown

What you’re defending against

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

Agenda

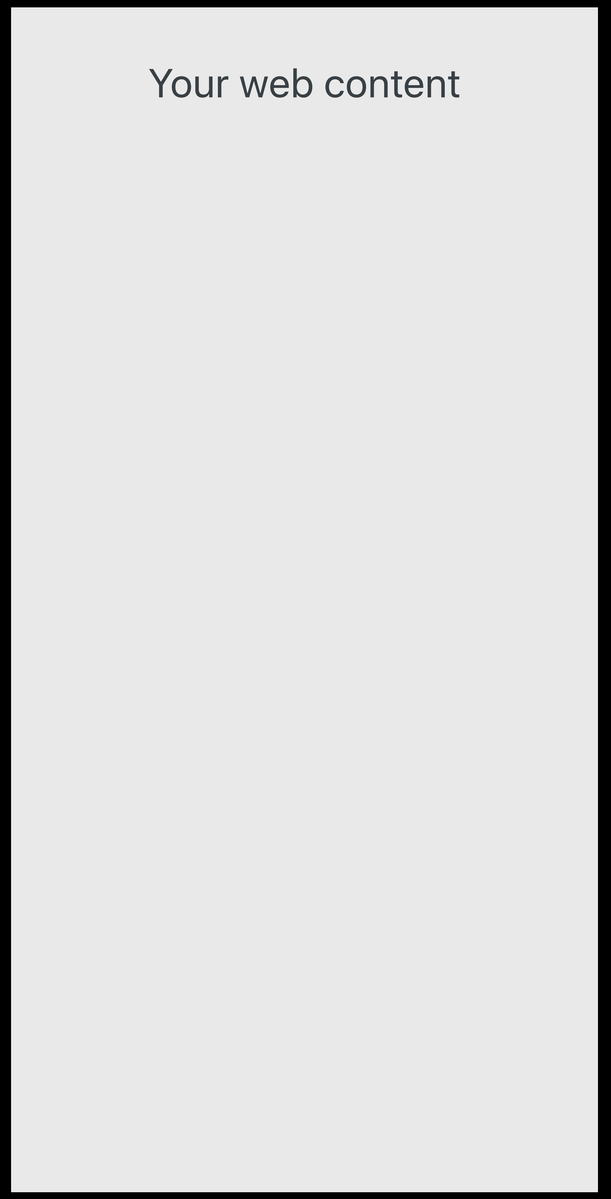
Secure transport

Cross-origin lockdown

What you’re defending against

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

Cross-Origin Loads



Cross-Origin Loads

Cross-origin image

Your web content

Cross-Origin Loads

Cross-origin image

Cross-origin script

Your web content

```
function onload() {
  ...
}
```

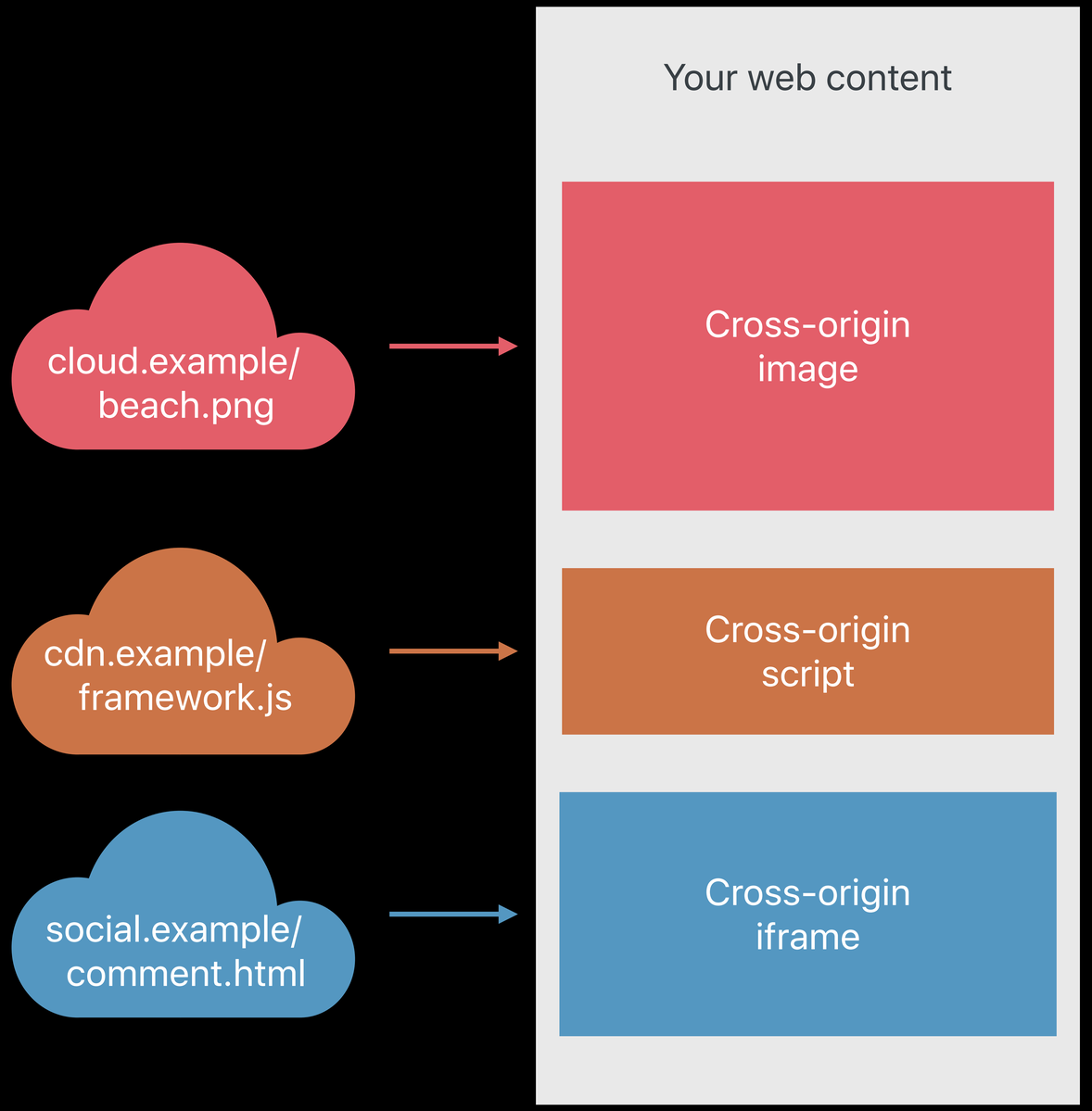
Cross-Origin Loads

Cross-origin image

Cross-origin script

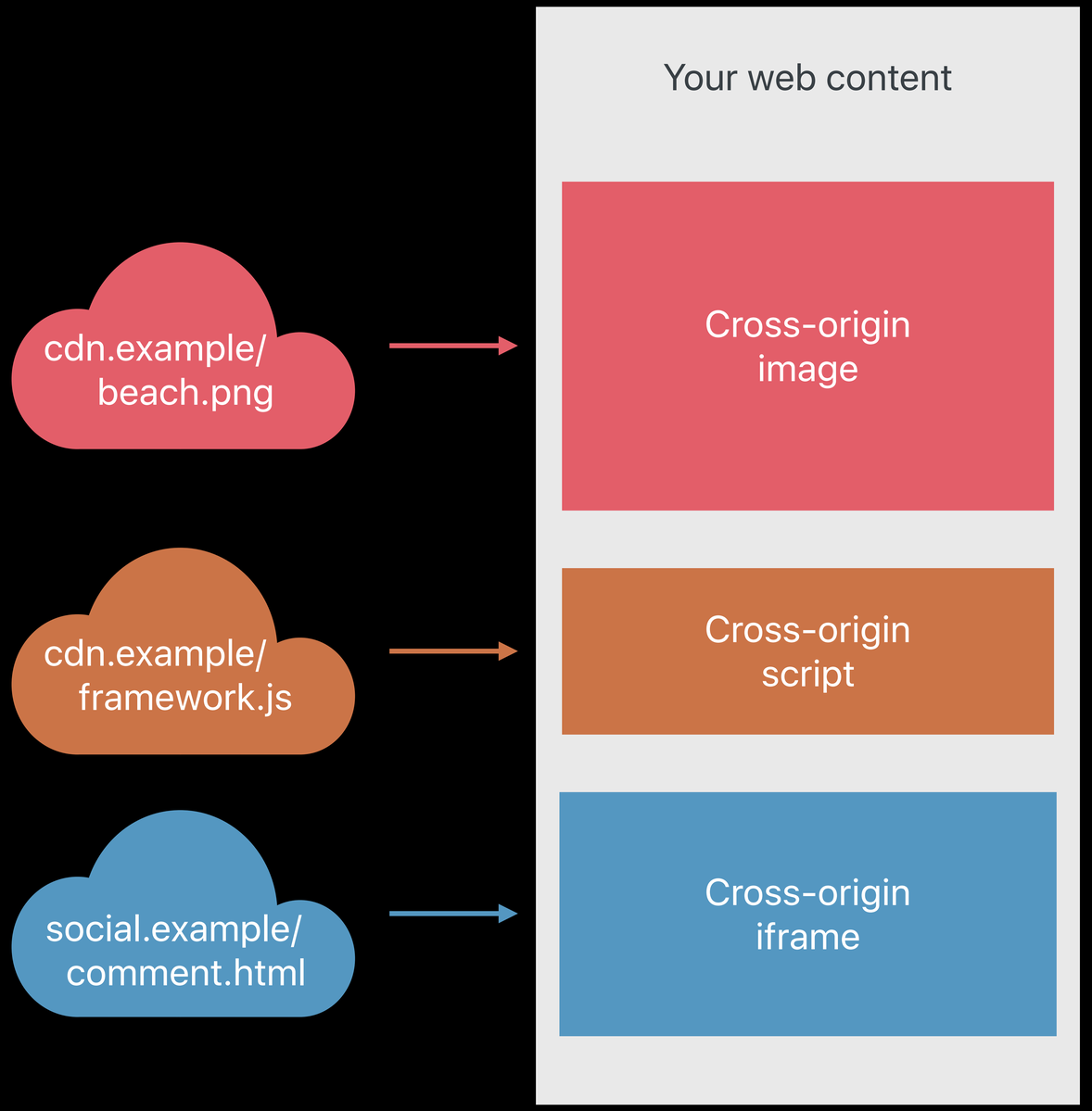
Cross-origin iframe

Cross-Origin Loads



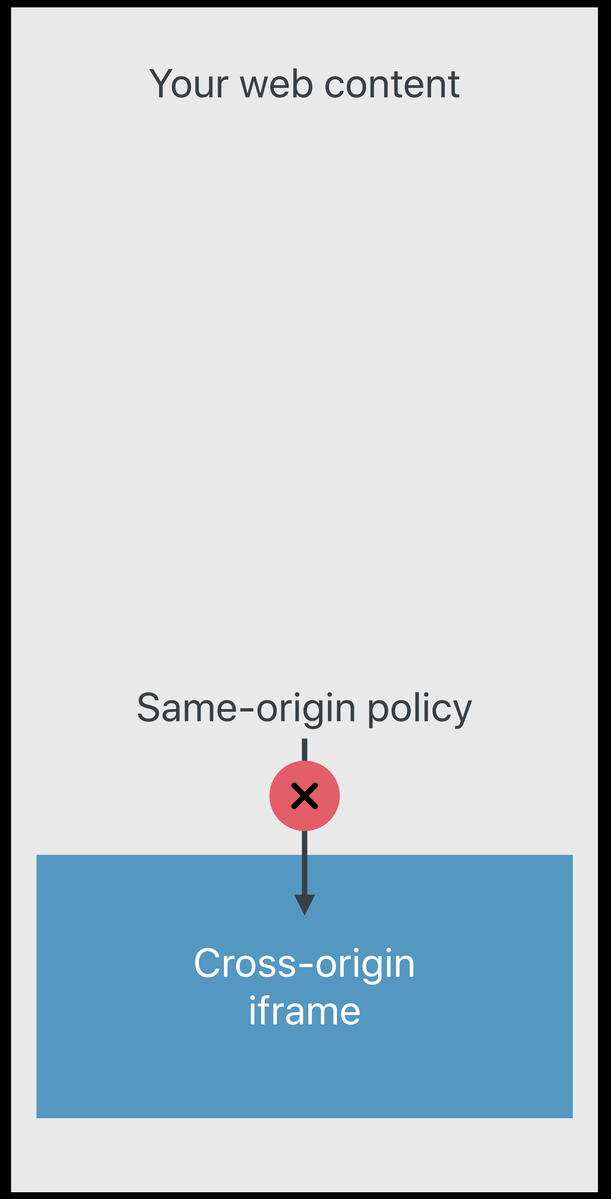
.example
.com
.org

Cross-Origin Loads



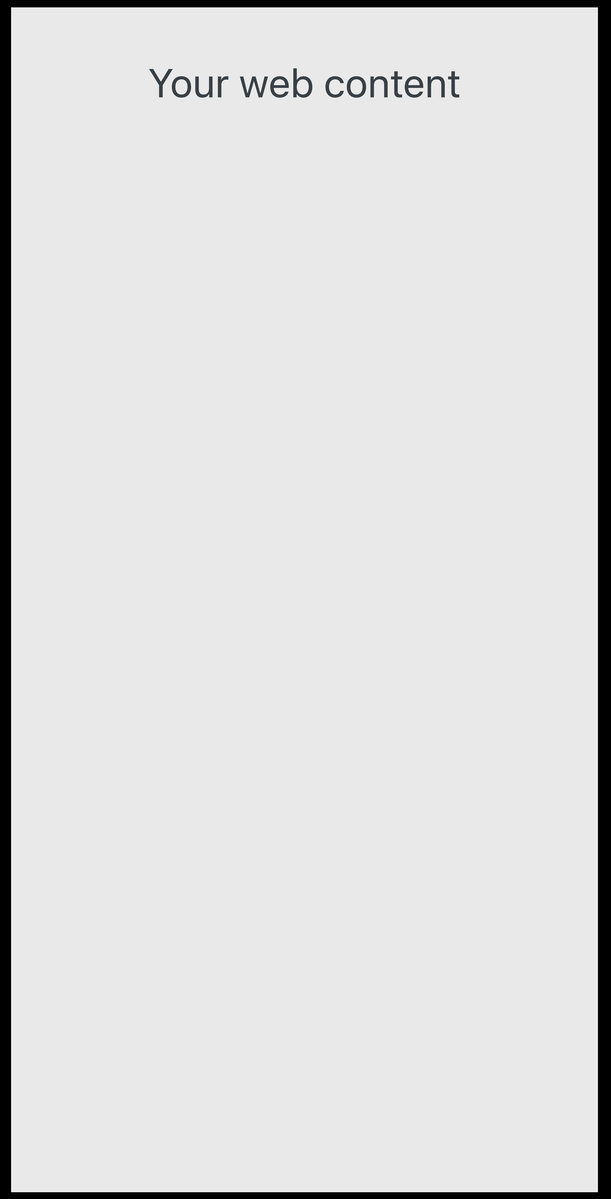
Cross-Origin Lockdown

Cross-origin loads



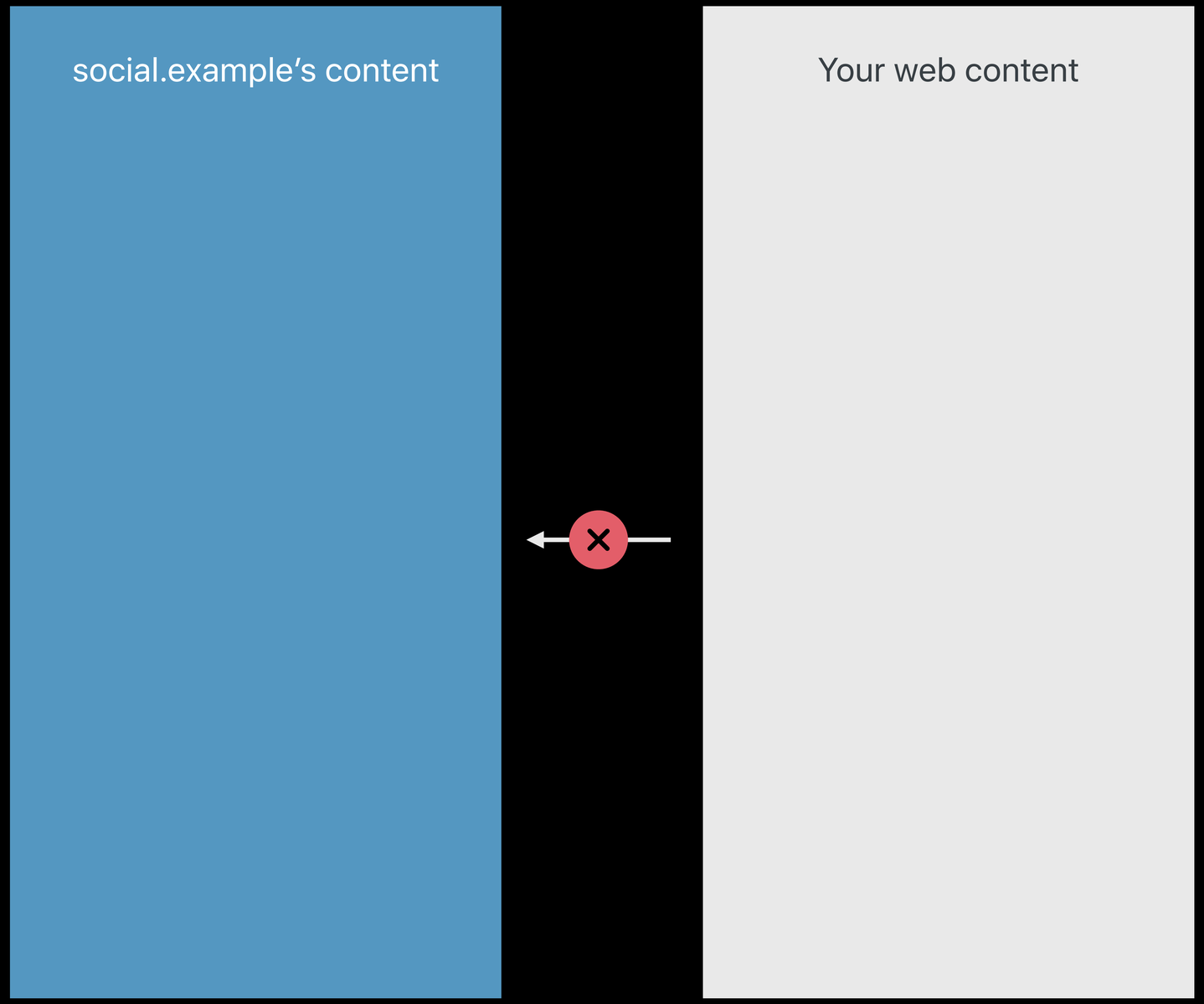
Cross-Origin Lockdown

Cross-origin loads



Cross-Origin Lockdown

Cross-origin loads



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	Simple markup change to script tags
Content Security Policy	Server configuration + architectural changes
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 — Server configuration + architectural changes

HttpOnly cookies

SameSite cookies

Cross-Origin-Resource-Policy

Cross-Origin-Window-Policy

Simple server configuration changes

Cross-Origin Lockdown

Subresource Integrity

Content Security Policy

HttpOnly cookies

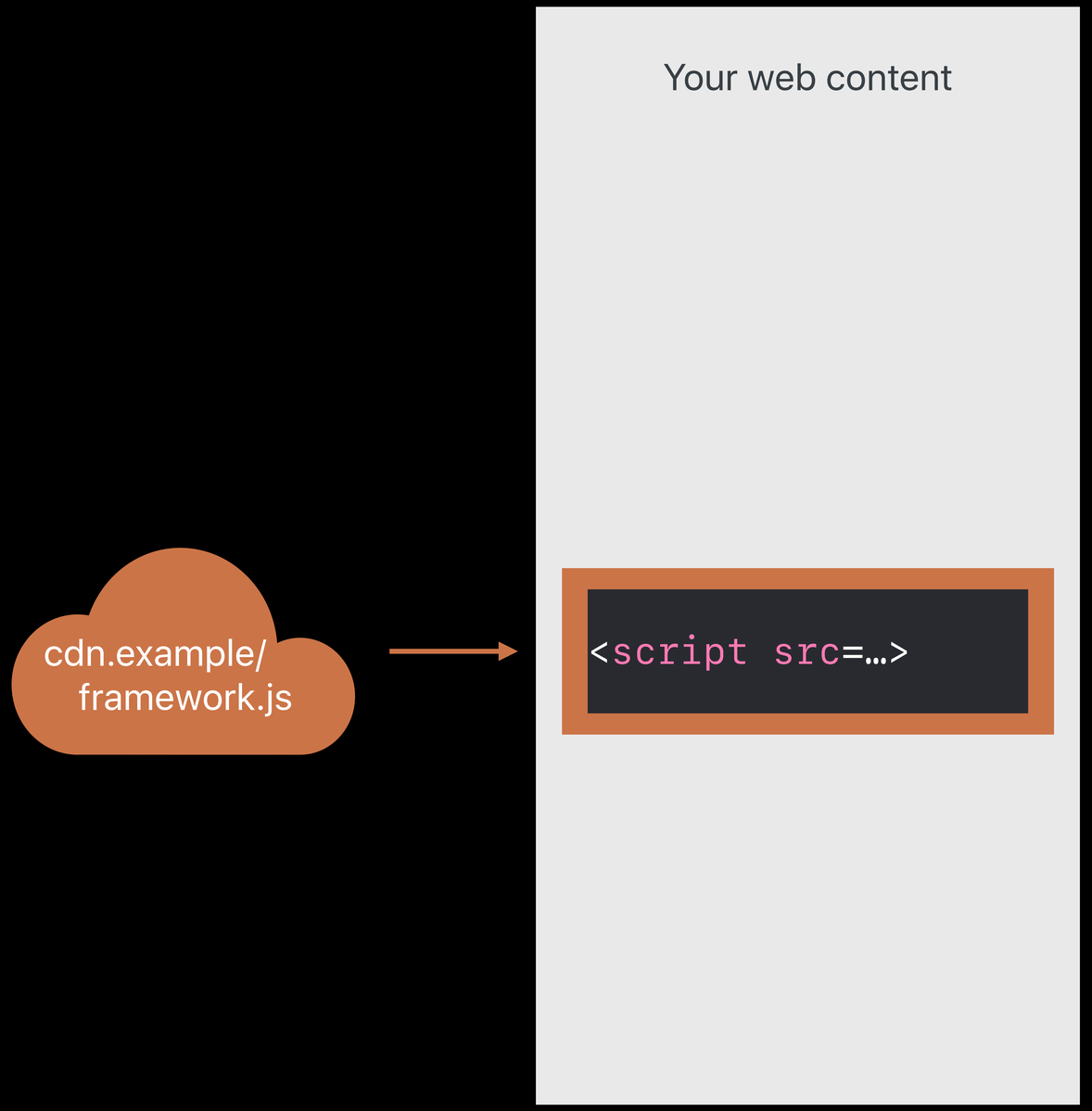
SameSite cookies

Cross-Origin-Resource-Policy

Cross-Origin-Window-Policy

Cross-Origin Lockdown

Subresource integrity



Cross-Origin Lockdown

Subresource integrity

```
<script src="https://cdn.example/framework.js">
</script>
```

Cross-Origin Lockdown

Subresource integrity

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

Cross-Origin Lockdown

Subresource integrity

```
<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

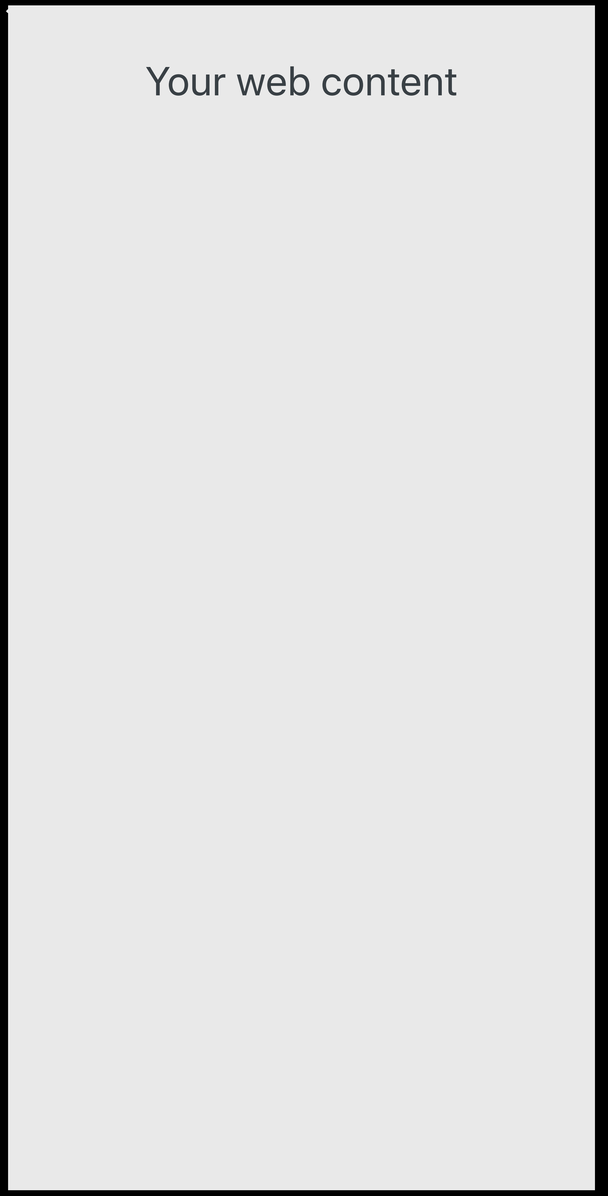
Cross-Origin-Window-Policy

Cross-Origin Lockdown

Content security policy

HTTP response:

```
:status: 200
Content-Security-Policy:
```

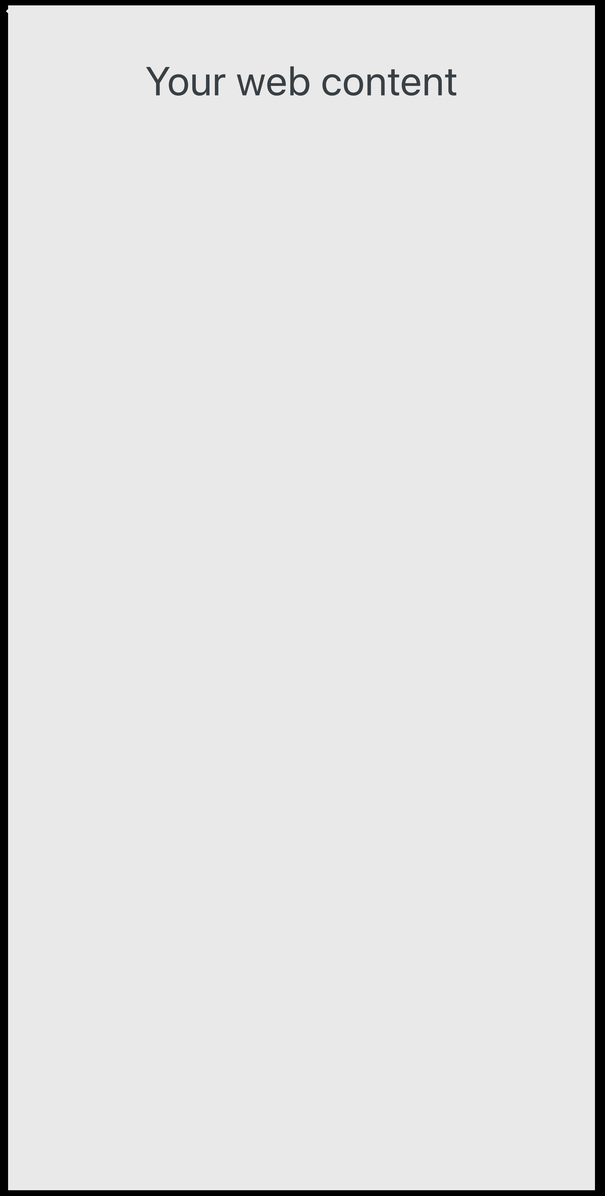


Cross-Origin Lockdown

Content security policy

HTTP response:

```
:status: 200
Content-Security-Policy:
 default-src 'self';  // No inline
```



Cross-Origin Lockdown

Content security policy

HTTP response:

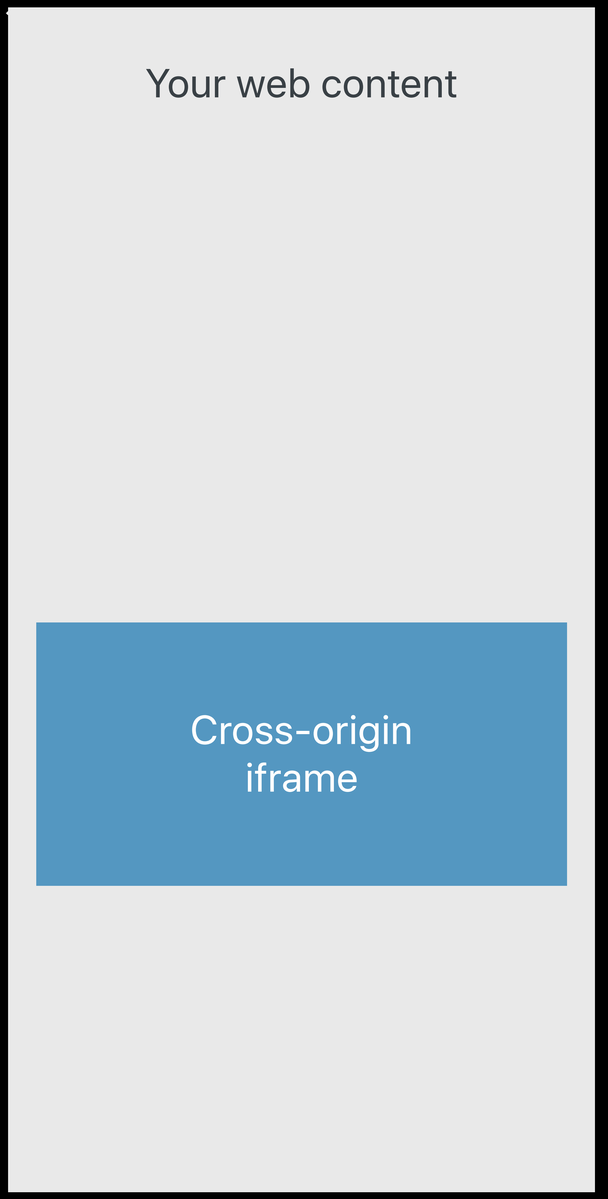
```
:status: 200
Content-Security-Policy:
 default-src 'self';
 script-src cdn.example;
```

Cross-Origin Lockdown

Content security policy

HTTP response:

```
:status: 200
Content-Security-Policy:
 default-src 'self';
 script-src cdn.example;
 frame-src social.example;
```

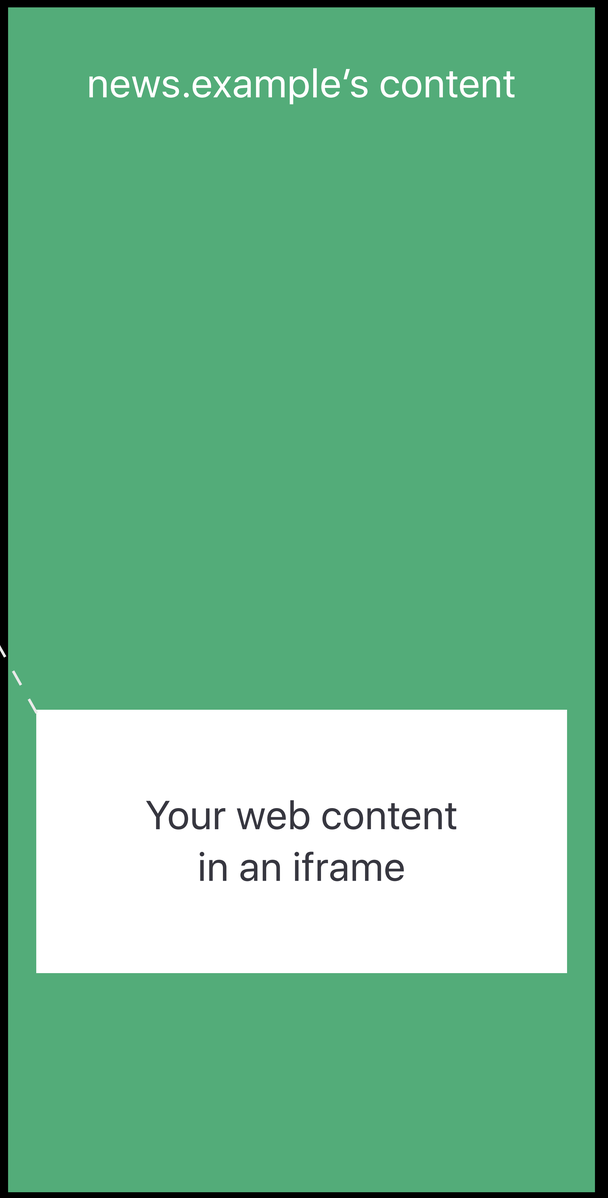


Cross-Origin Lockdown

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

```
Content-Security-Policy:

 frame-src social.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

Your web content

```
<script>
  document.cookie
</script>
```

Cross-Origin Lockdown

HttpOnly cookies

HTTP response:

```
:status: 200
Set-Cookie:
 auth=abc…123; HttpOnly
```

Your web content

```
<script>
  document.cookie
</script>
```

Cross-Origin Lockdown

HttpOnly cookies

HTTP response:

```
:status: 200
Set-Cookie:
 auth=abc…123; HttpOnly
```

Your web content

```
<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

NEW

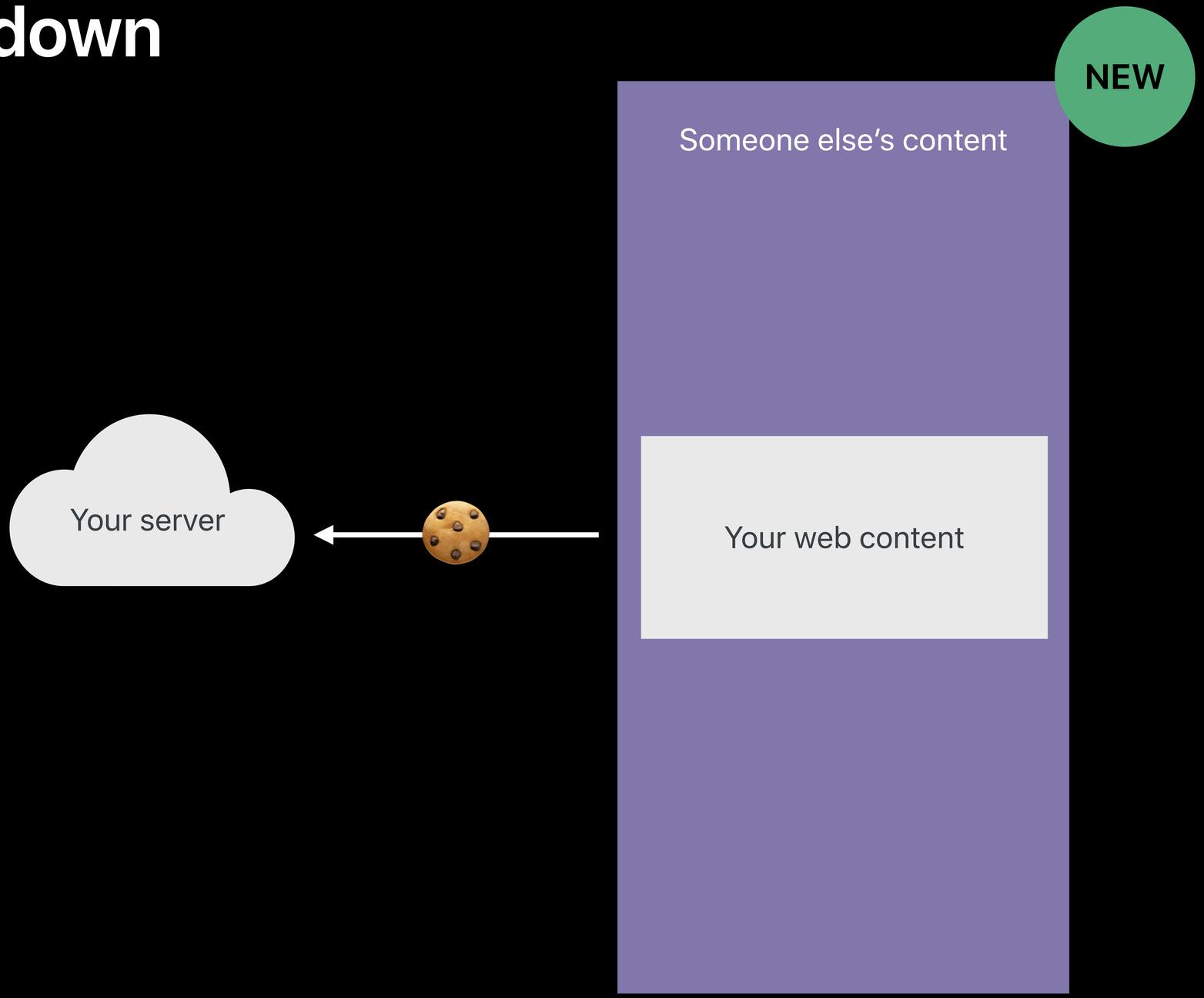
Your web content

HTTP response:

```
:status: 200
Set-Cookie:
 auth=abc…123; HttpOnly;
 SameSite=strict
```

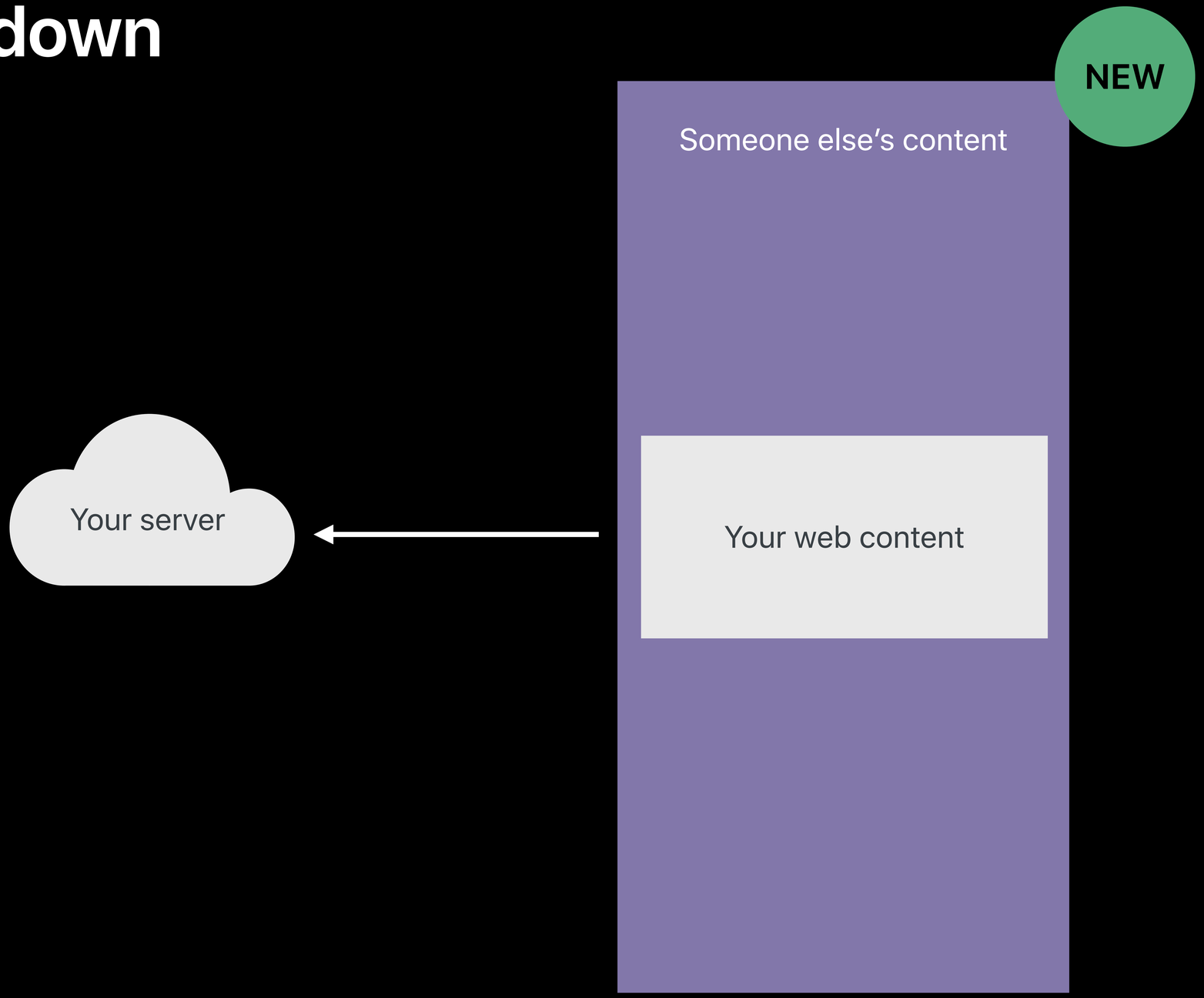
Cross-Origin Lockdown

SameSite cookies



Cross-Origin Lockdown

SameSite cookies



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

NEW

Your server

Your server

Your web content

```
function onload() {
  ...
}
```

Cross-Origin Lockdown

Cross-Origin-Resource-Policy

NEW

Your server

HTTP response:

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

Your server

HTTP response:

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

Your web content

```
function onload() {
  ...
}
```

Cross-Origin Lockdown

Cross-Origin-Resource-Policy

NEW

Your server

HTTP response:

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

Your server

HTTP response:

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

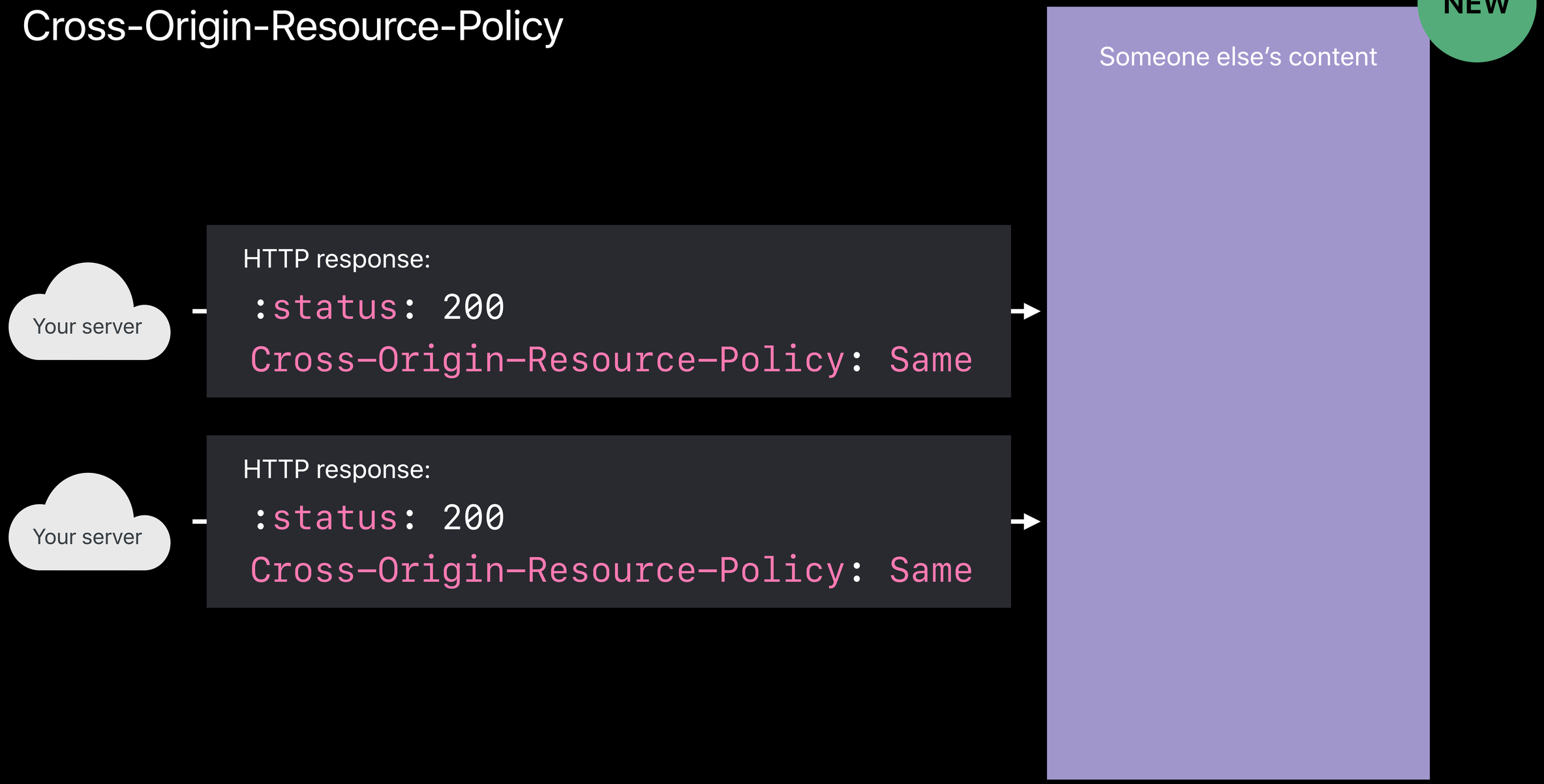
Someone else's content

```
function onload() {
  ...
}
```

Cross-Origin Lockdown

Cross-Origin-Resource-Policy

NEW



Cross-Origin Lockdown

Content Security Policy

Subresource Integrity

HttpOnly cookies

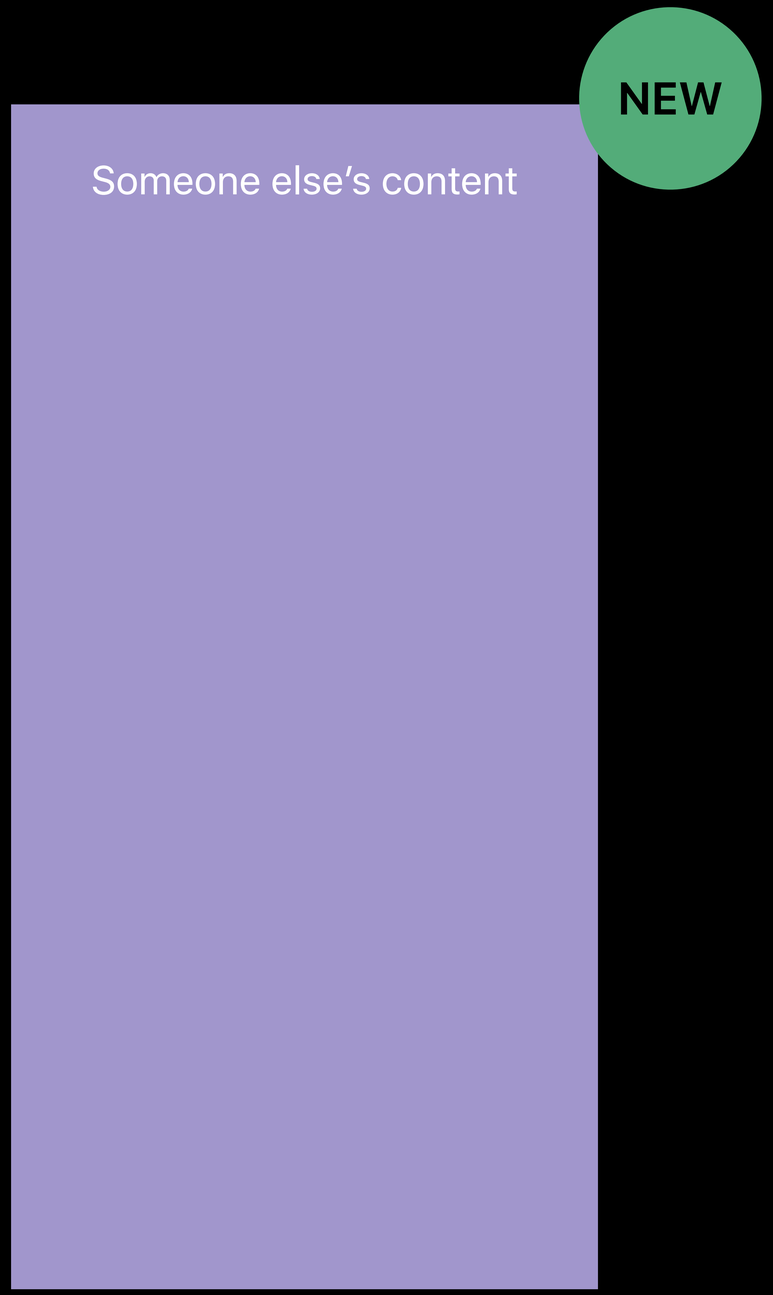
SameSite cookies

Cross-Origin-Resource-Policy

Cross-Origin-Window-Policy

Cross-Origin Lockdown

Cross-Origin-Window-Policy

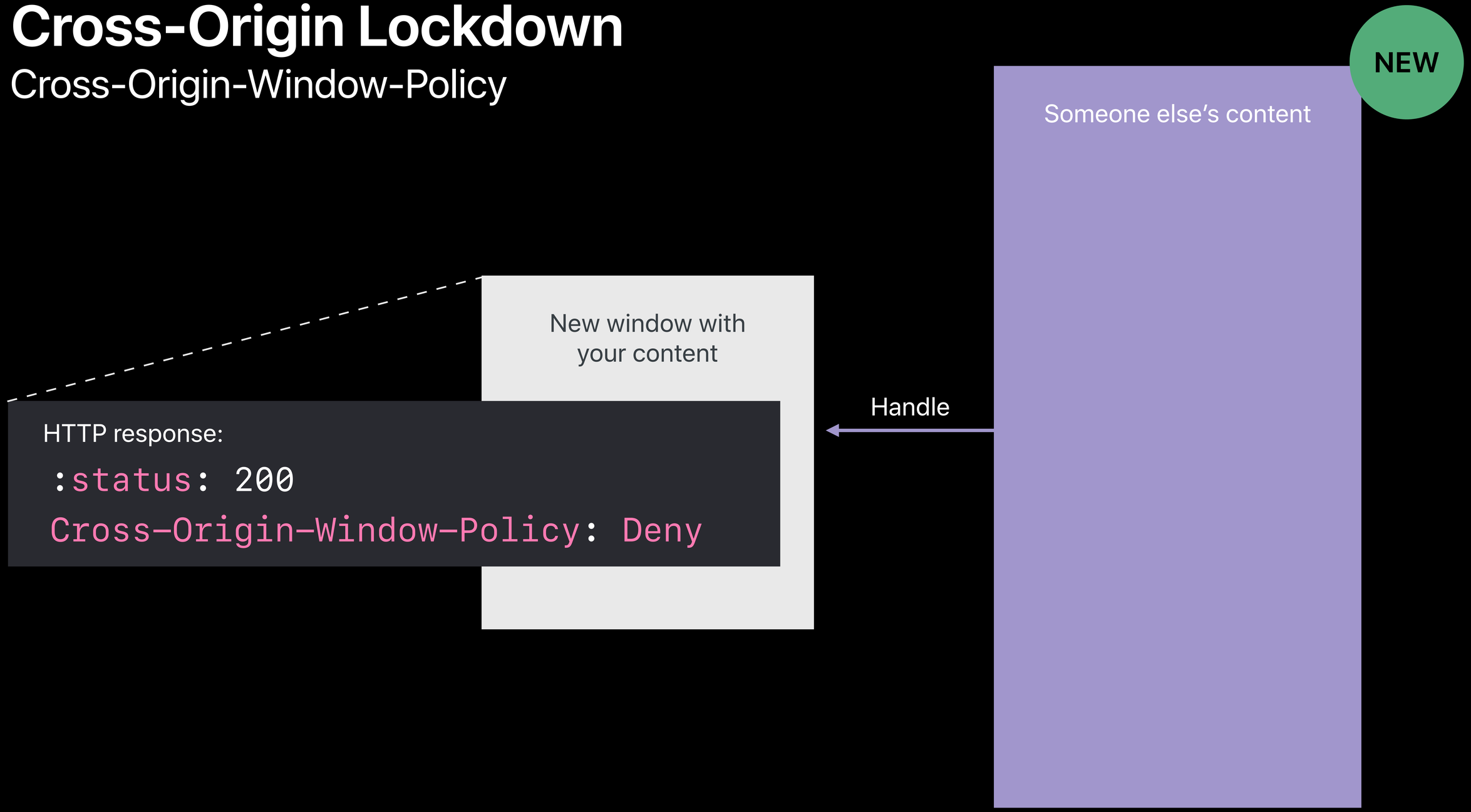


Cross-Origin Lockdown

Cross-Origin-Window-Policy

Cross-Origin Lockdown

Cross-Origin-Window-Policy



Cross-Origin Lockdown

Cross-Origin-Window-Policy

NEW

Someone else's content

New window with your content

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

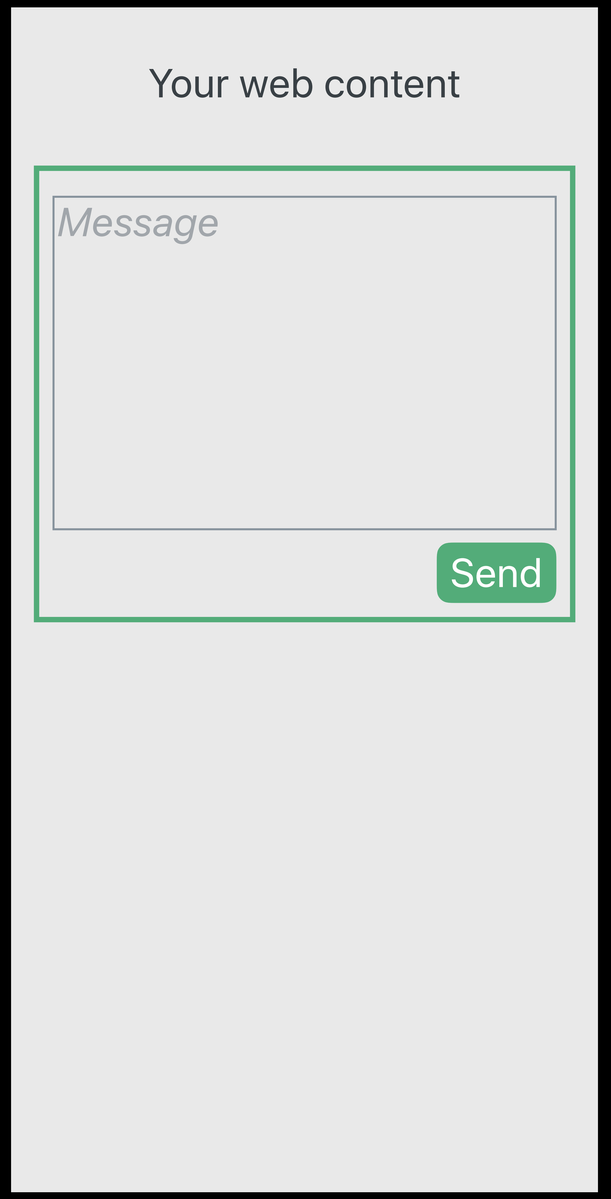
Cross-Site Scripting

Compromised CDN

Cross-Site Request Forgeries

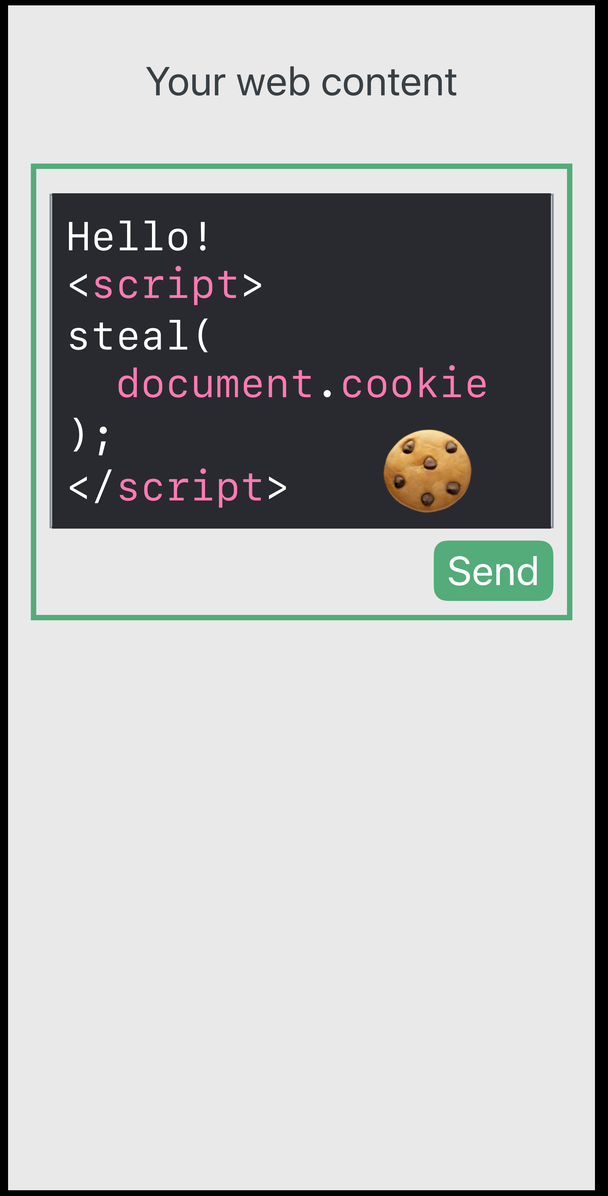
Cross-Origin Attacks

Cross-Site Scripting



Cross-Origin Attacks

Cross-Site Scripting



Cross-Origin Attacks

Cross-Site Scripting

HTTP response:

```
:status: 200
Set-Cookie:
 auth=abc…123; HttpOnly
```

Your web content

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

Send

Cross-Origin Attacks

Cross-Site Scripting

HTTP response:

```
:status: 200
Set-Cookie:
 auth=abc…123; HttpOnly
```

Your web content

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

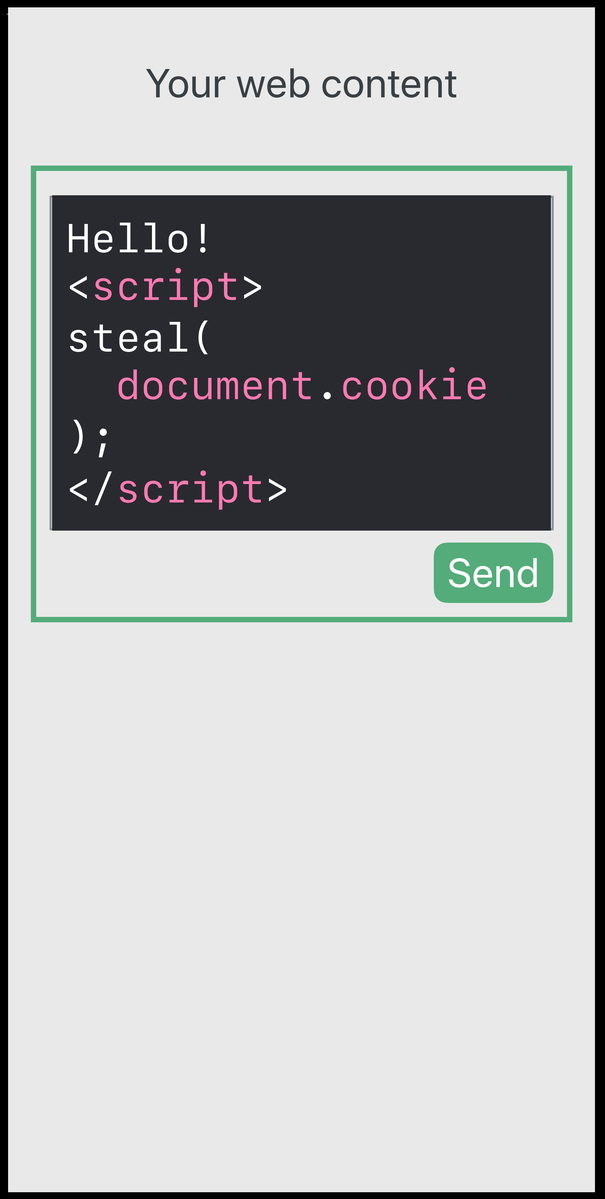
Send

Cross-Origin Attacks

Cross-Site Scripting

HTTP response:

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

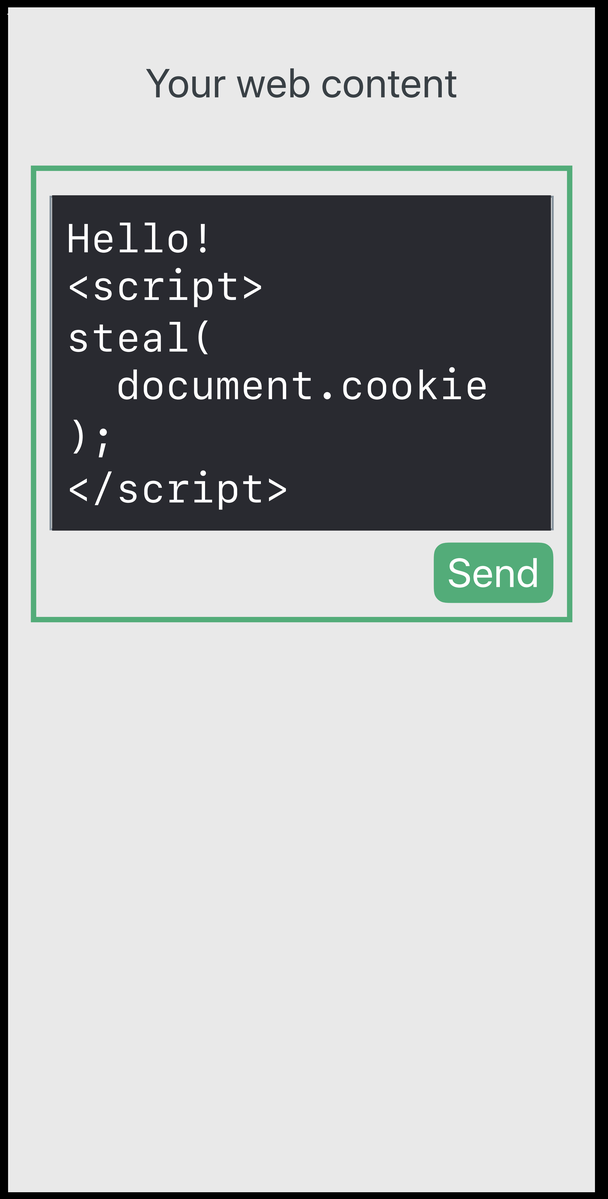


Cross-Origin Attacks

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

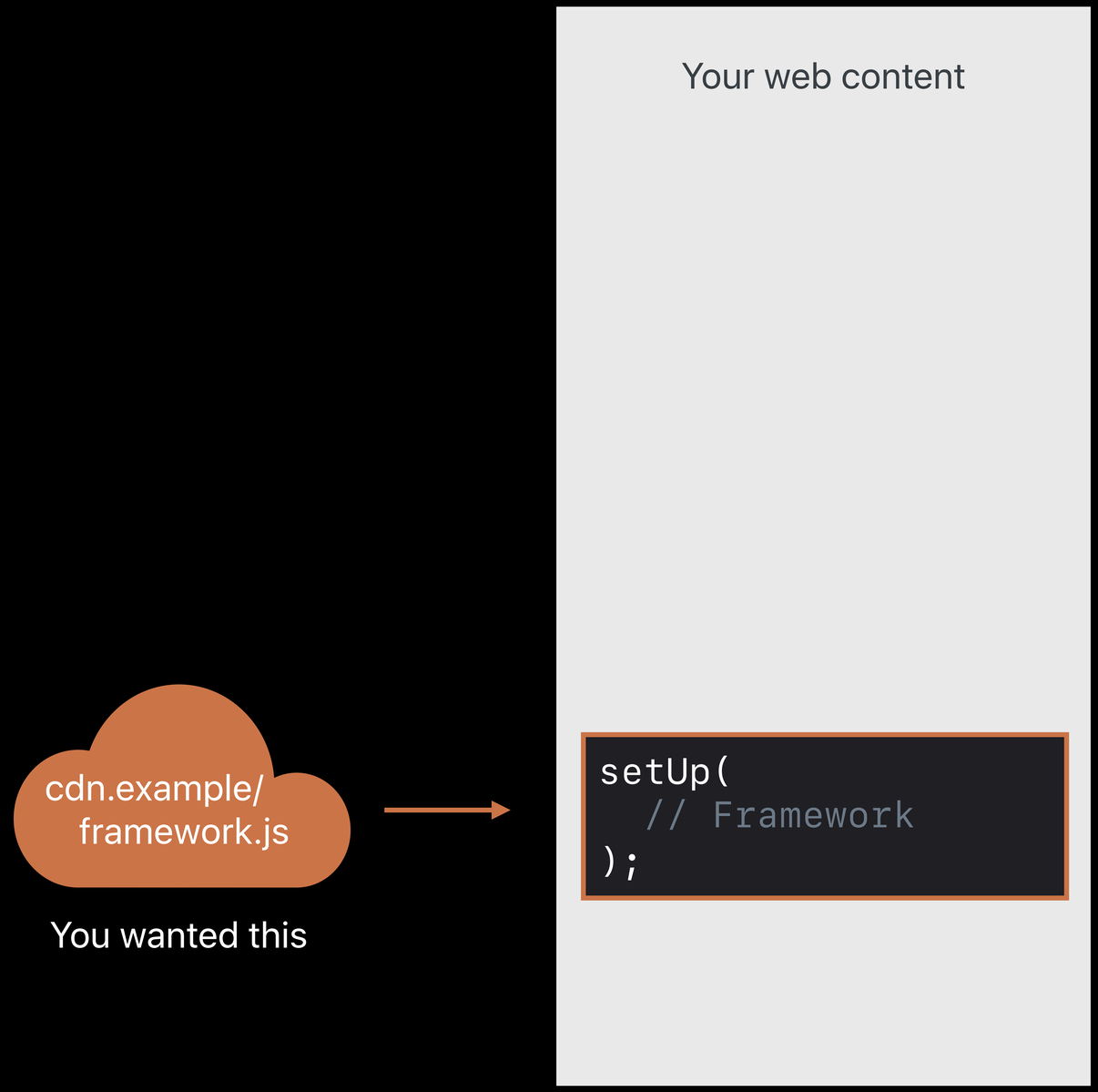
Cross-Site Scripting

Compromised CDN

Cross-Site Request Forgeries

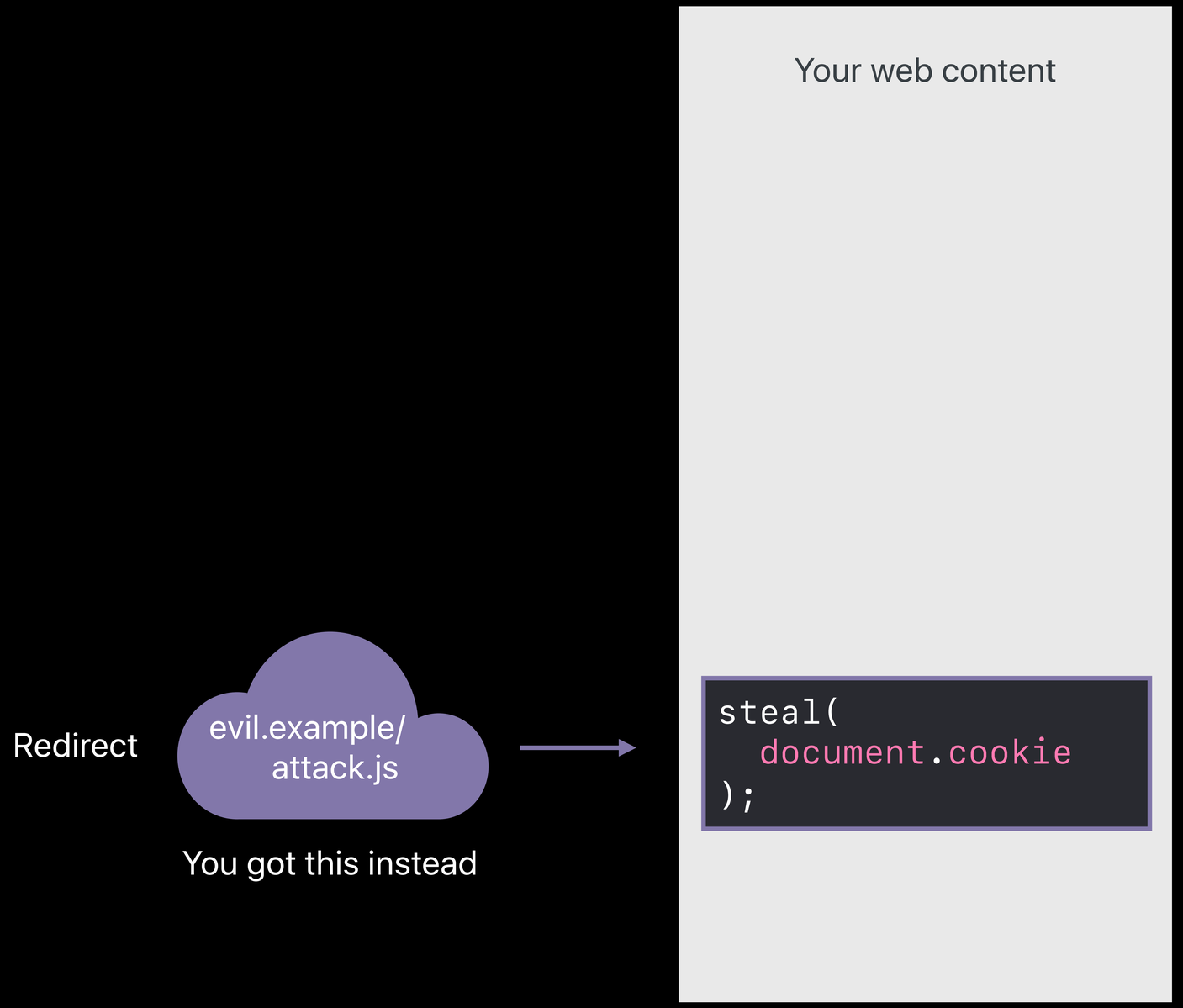
Cross-Origin Attacks

Compromised CDN



Cross-Origin Attacks

Compromised CDN

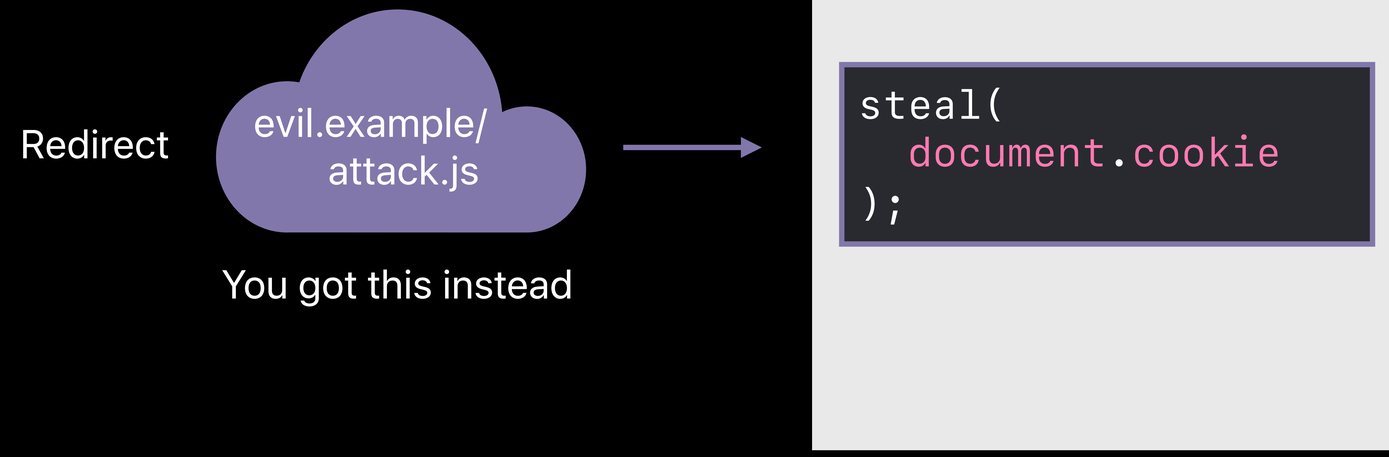


Cross-Origin Attacks

Compromised CDN

HTTP response:

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

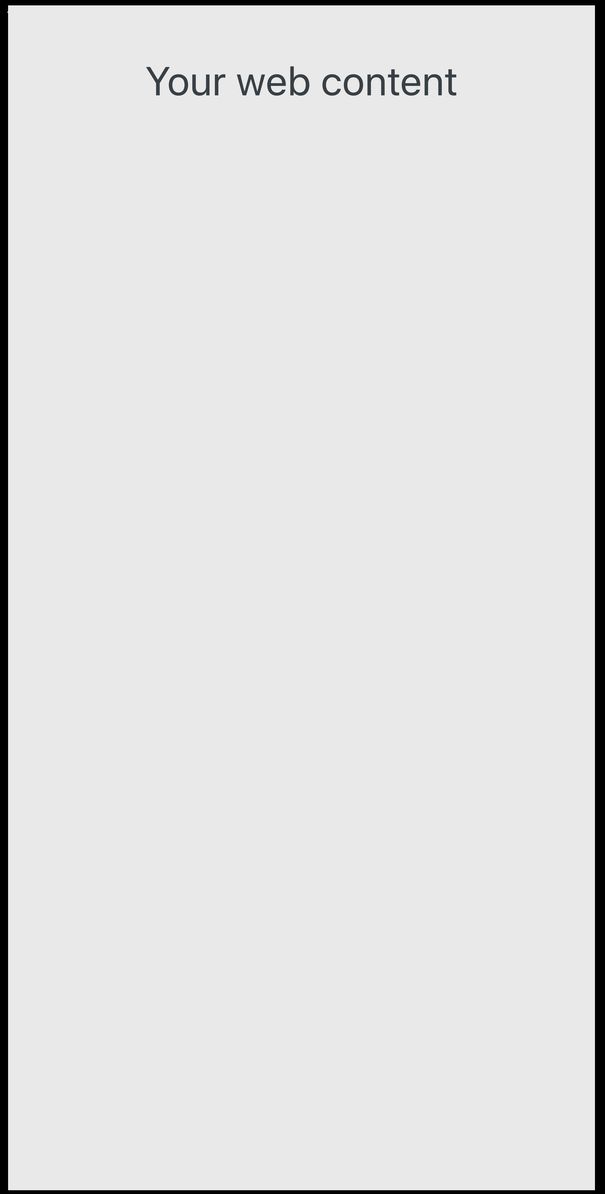


Cross-Origin Attacks

Compromised CDN

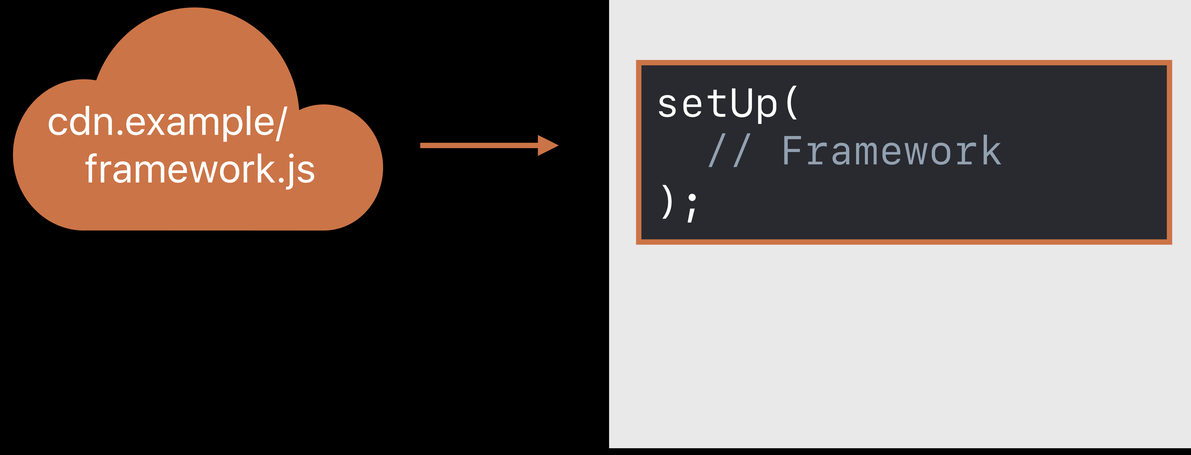
HTTP response:

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



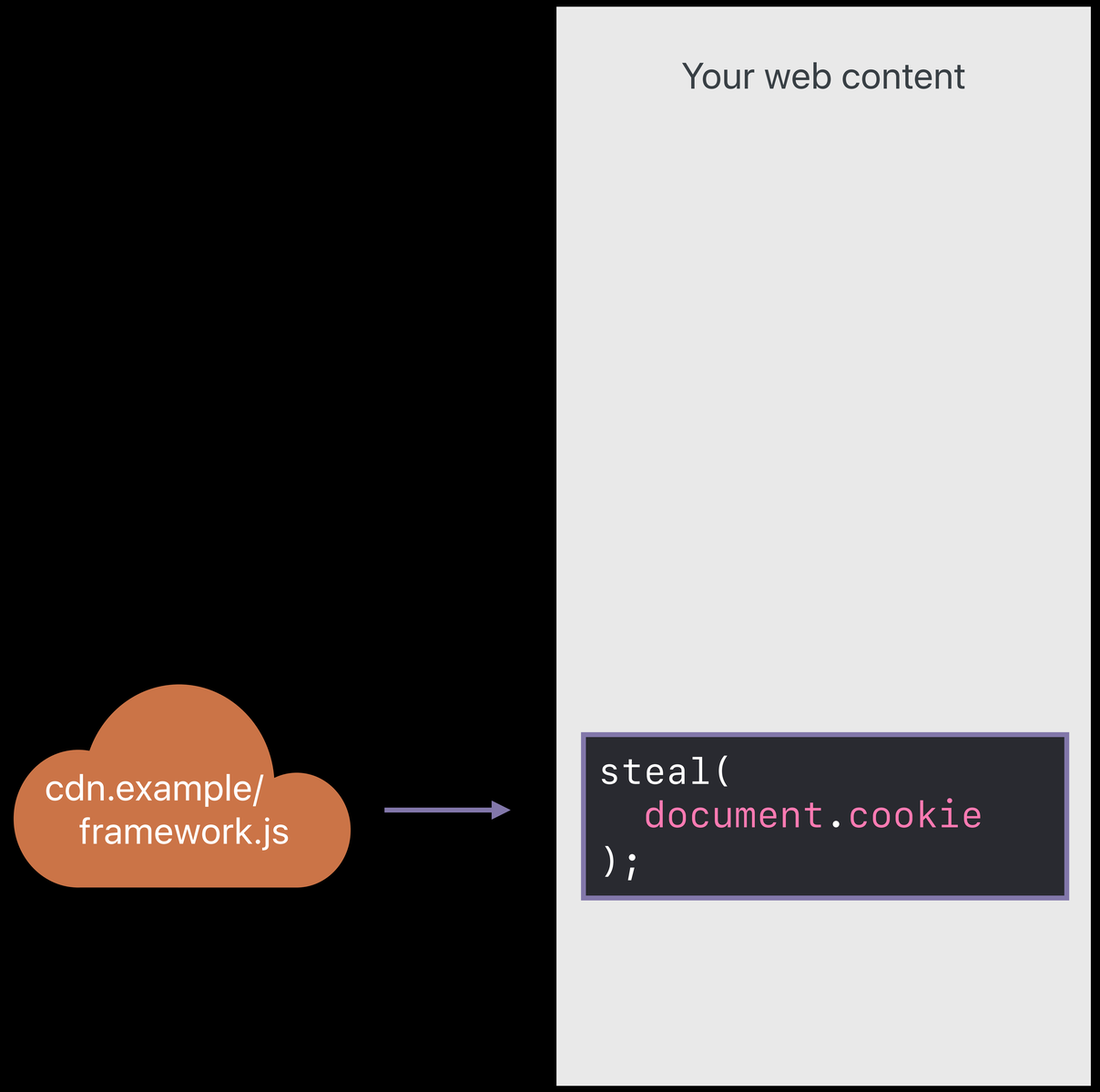
Cross-Origin Attacks

Compromised CDN



Cross-Origin Attacks

Compromised CDN



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

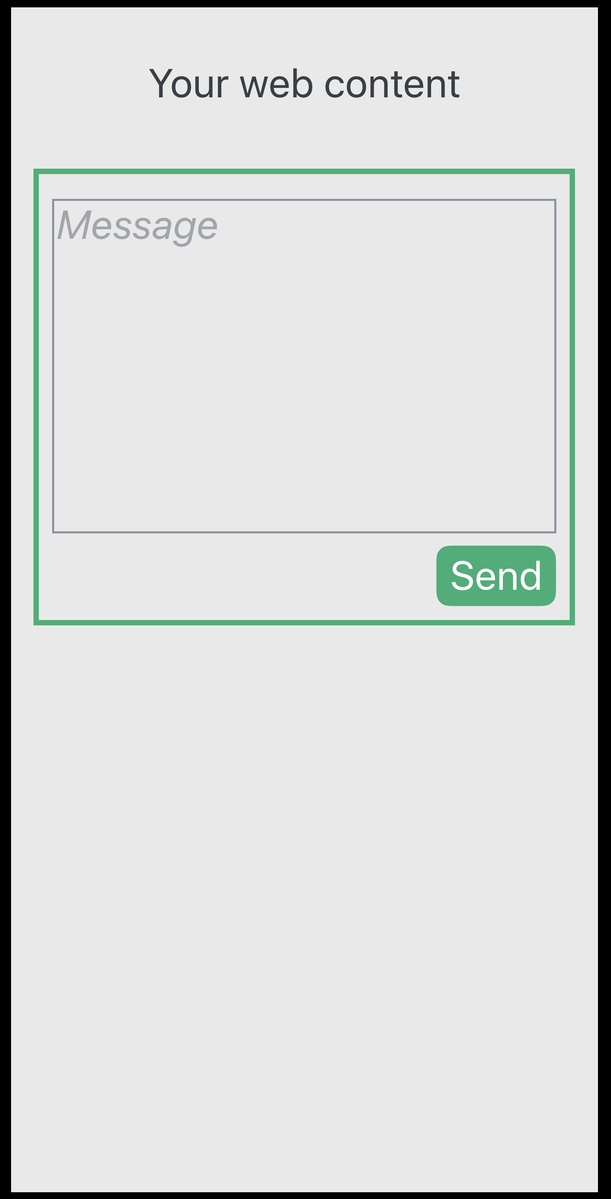
Cross-Site Scripting

Compromised CDN

Cross-Site Request Forgeries

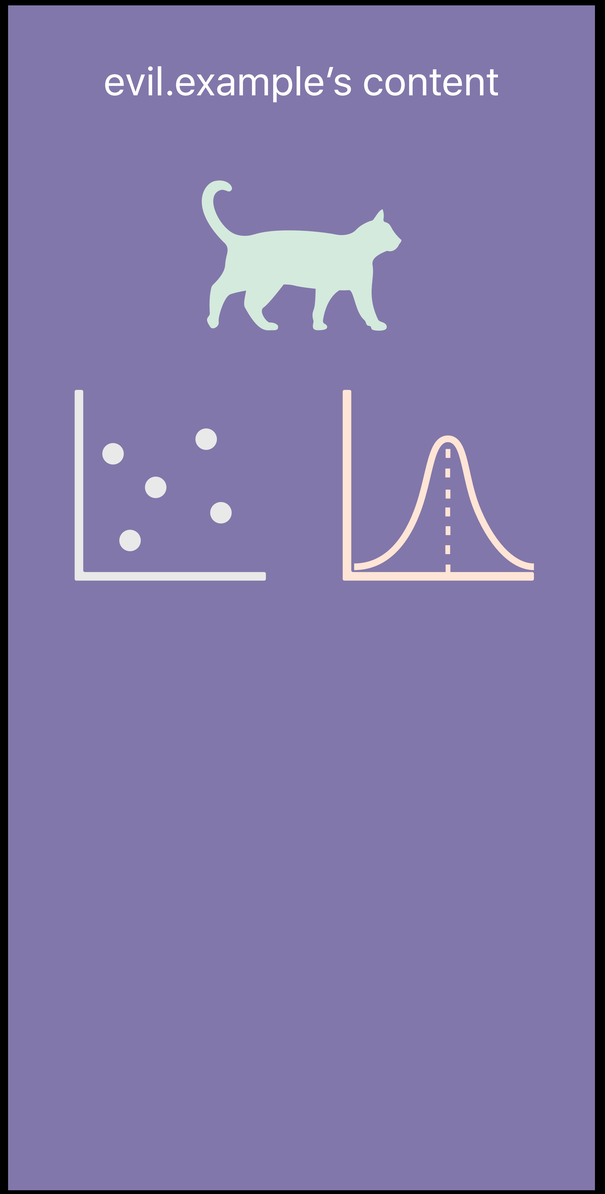
Cross-Origin Attacks

Cross-Site Request Forgery



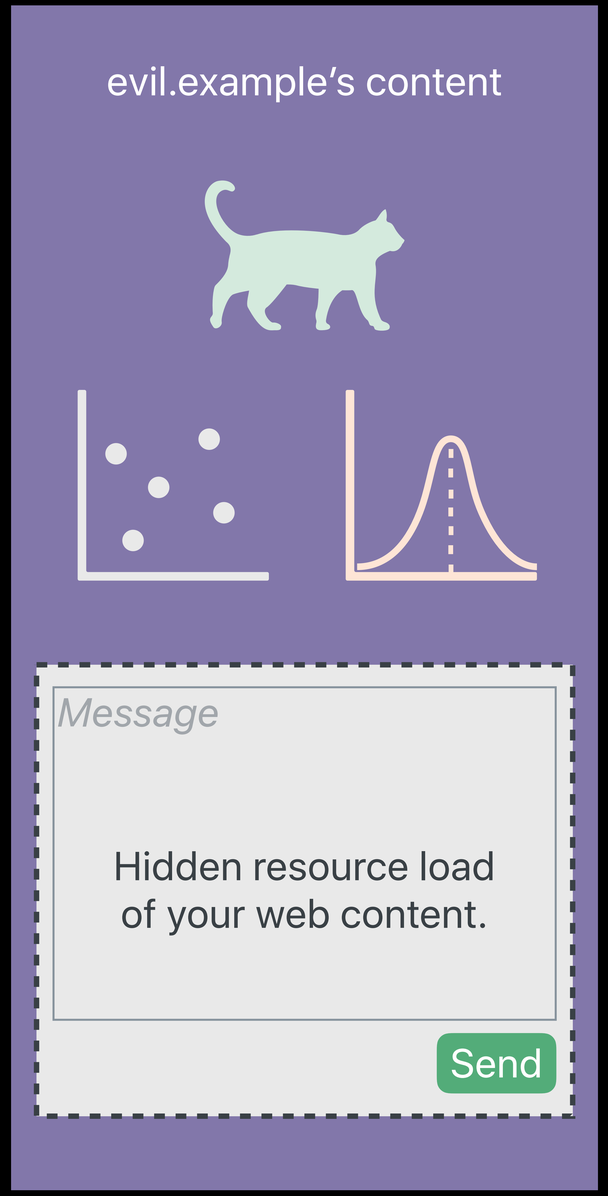
Cross-Origin Attacks

Cross-Site Request Forgery



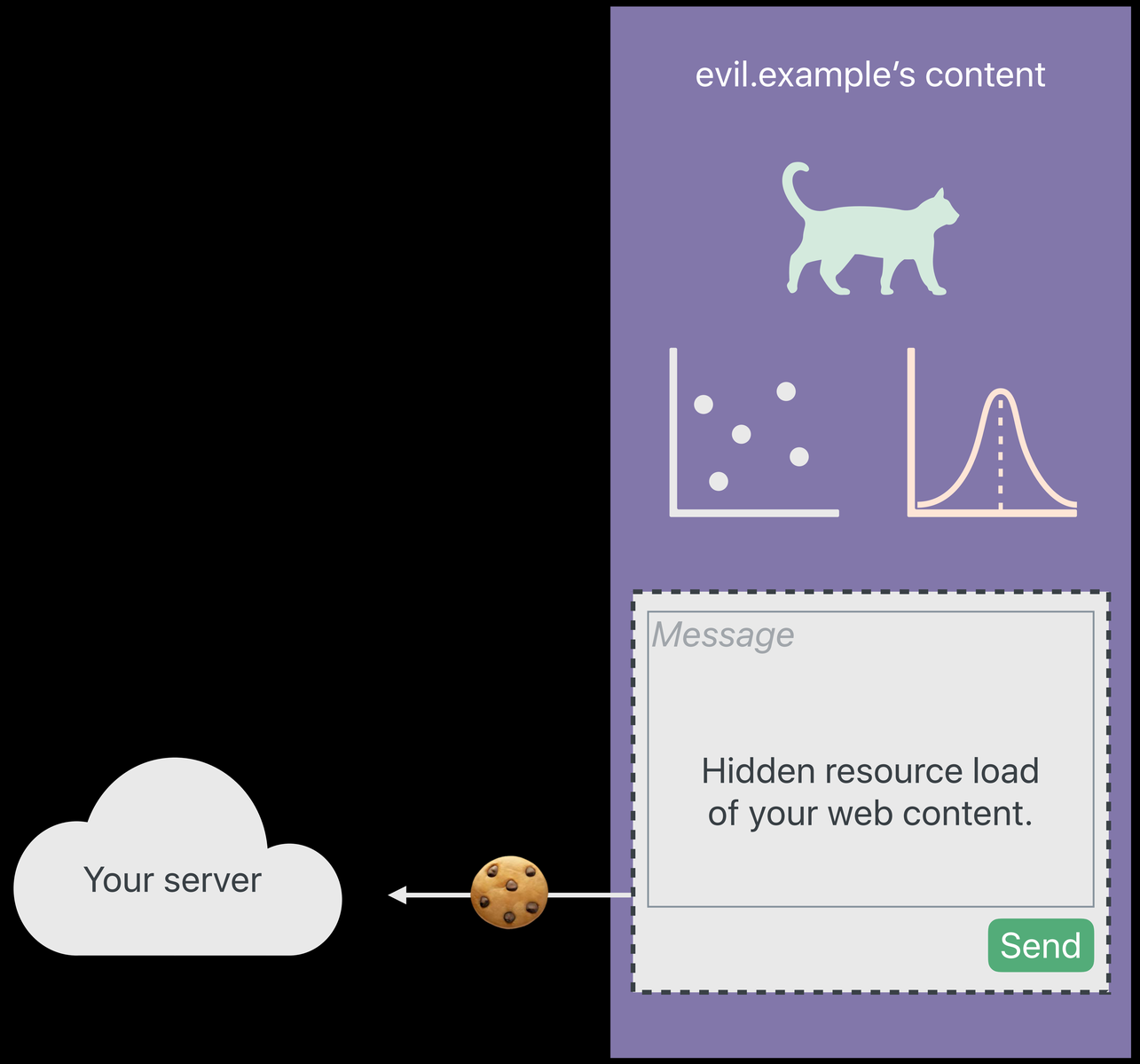
Cross-Origin Attacks

Cross-Site Request Forgery



Cross-Origin Attacks

Cross-Site Request Forgery

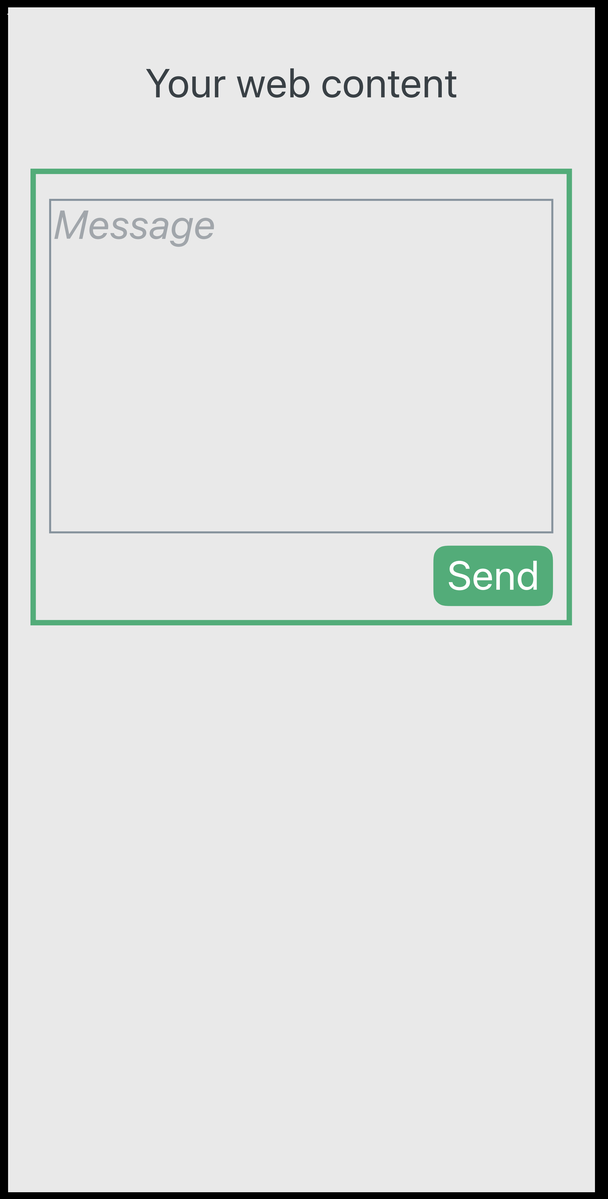


Cross-Origin Attacks

Cross-Site Request Forgery

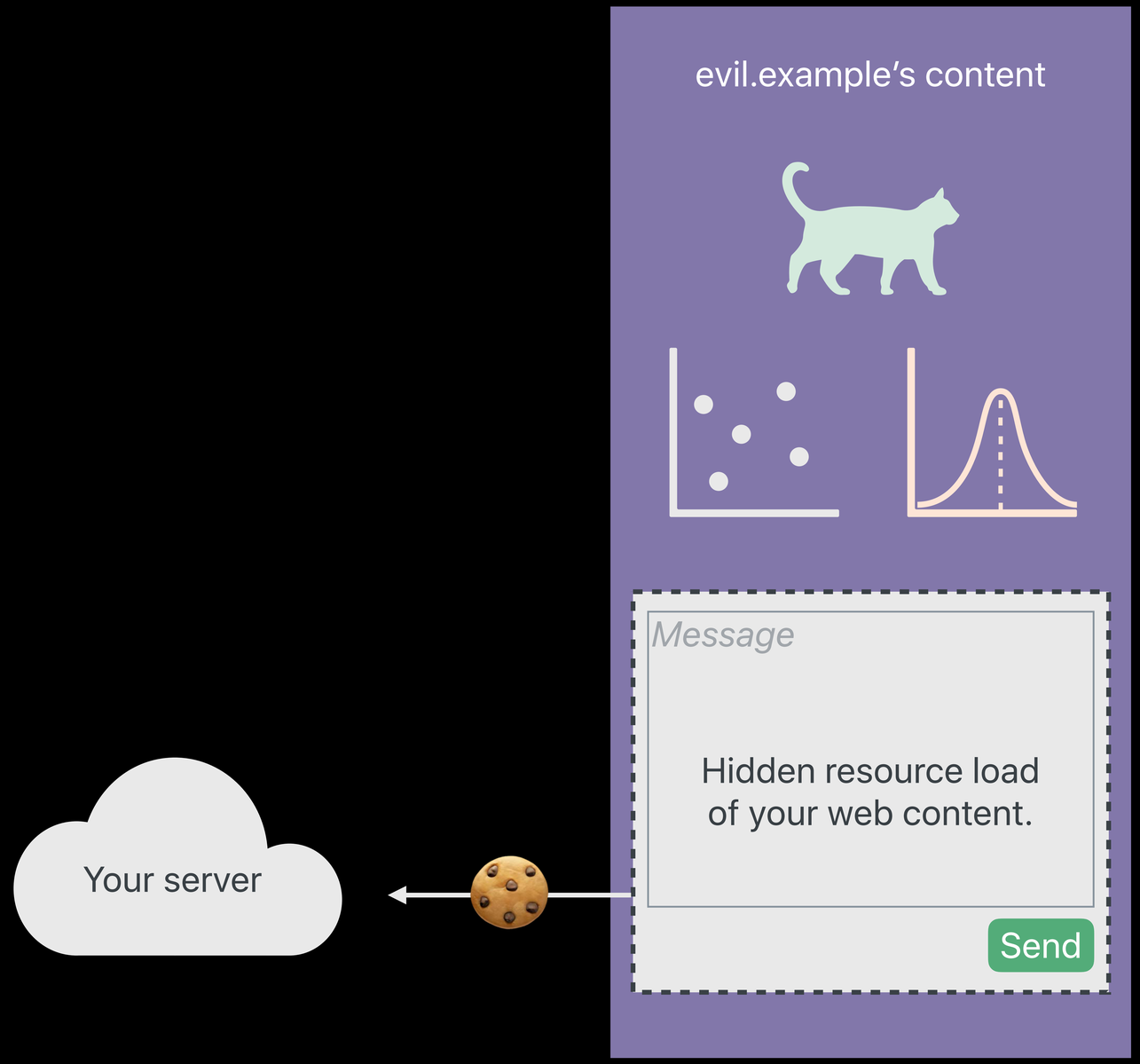
HTTP response:

```
:status: 200
Set-Cookie:
 auth=abc…123; SameSite=strict
```



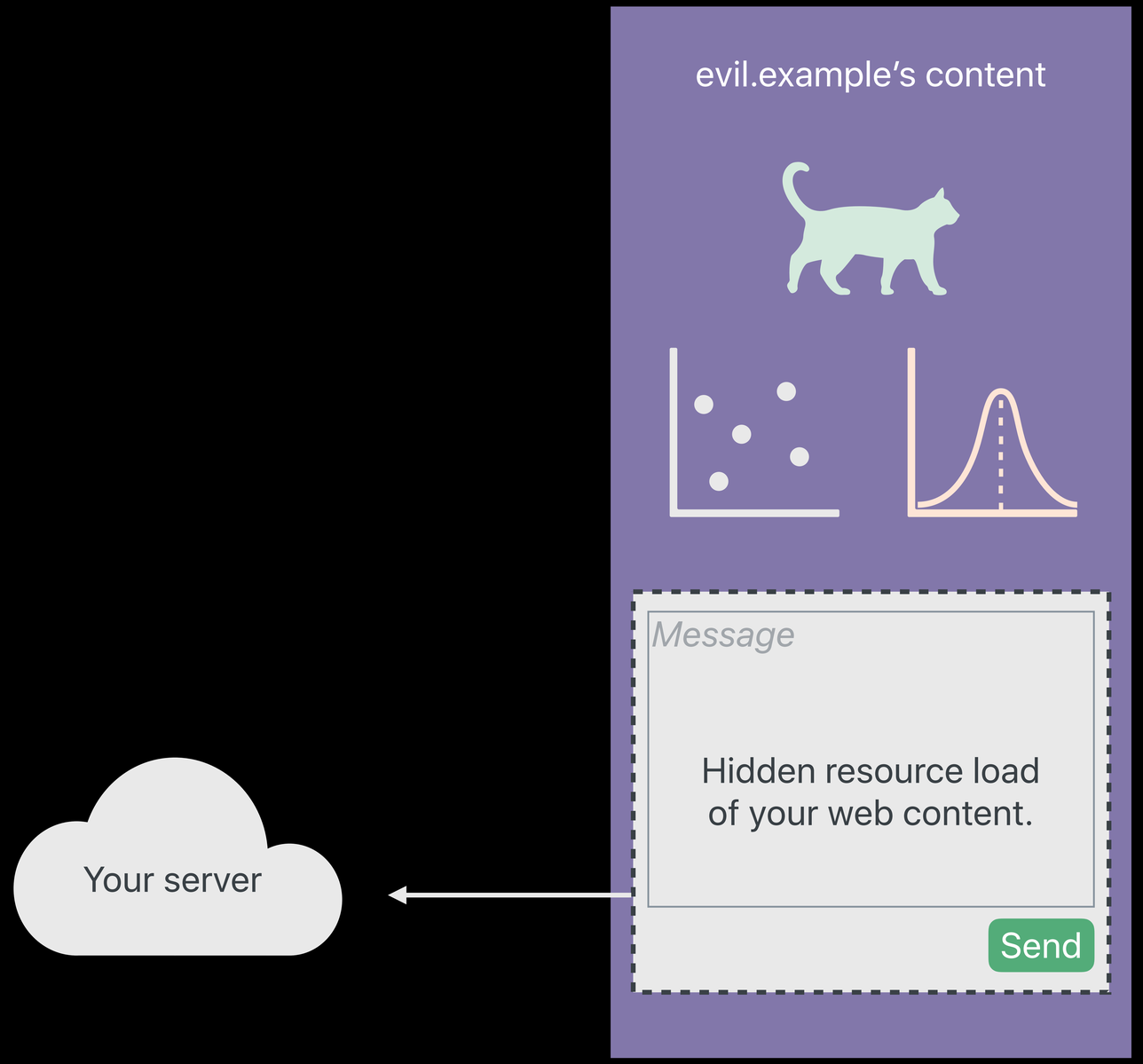
Cross-Origin Attacks

Cross-Site Request Forgery



Cross-Origin Attacks

Cross-Site Request Forgery



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

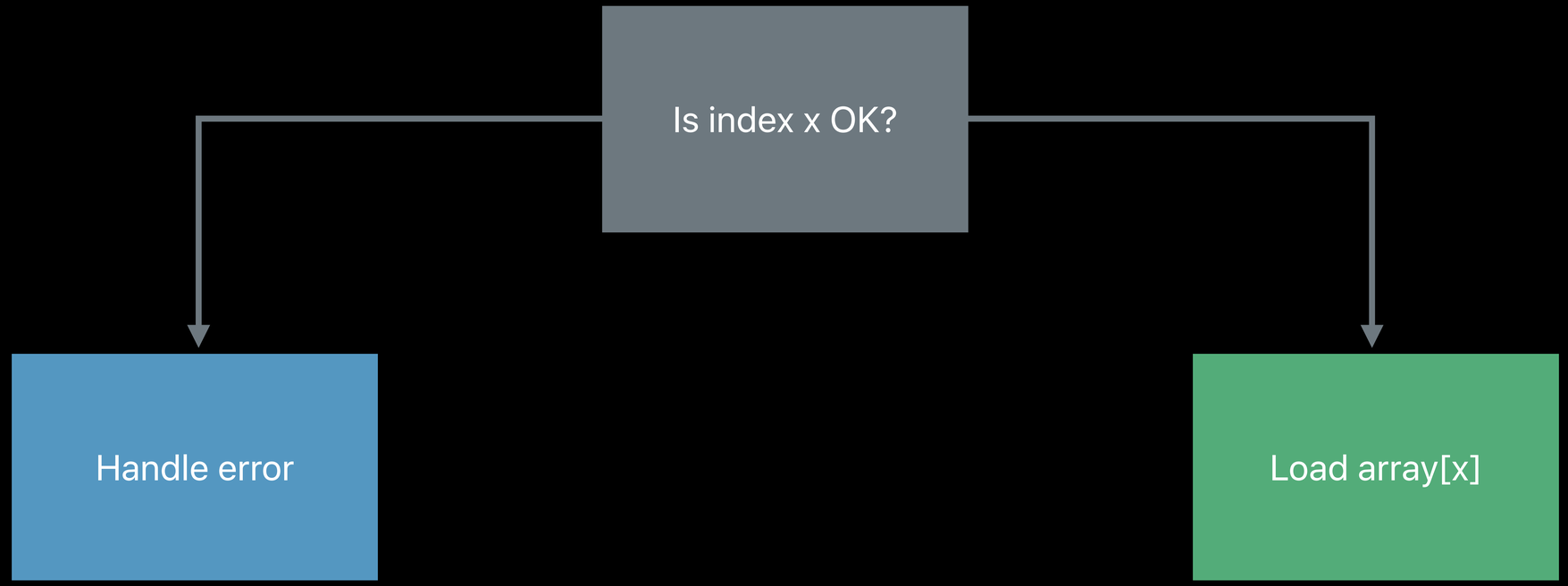
Speculative Execution and the Attack (Spectre)

Defenses:

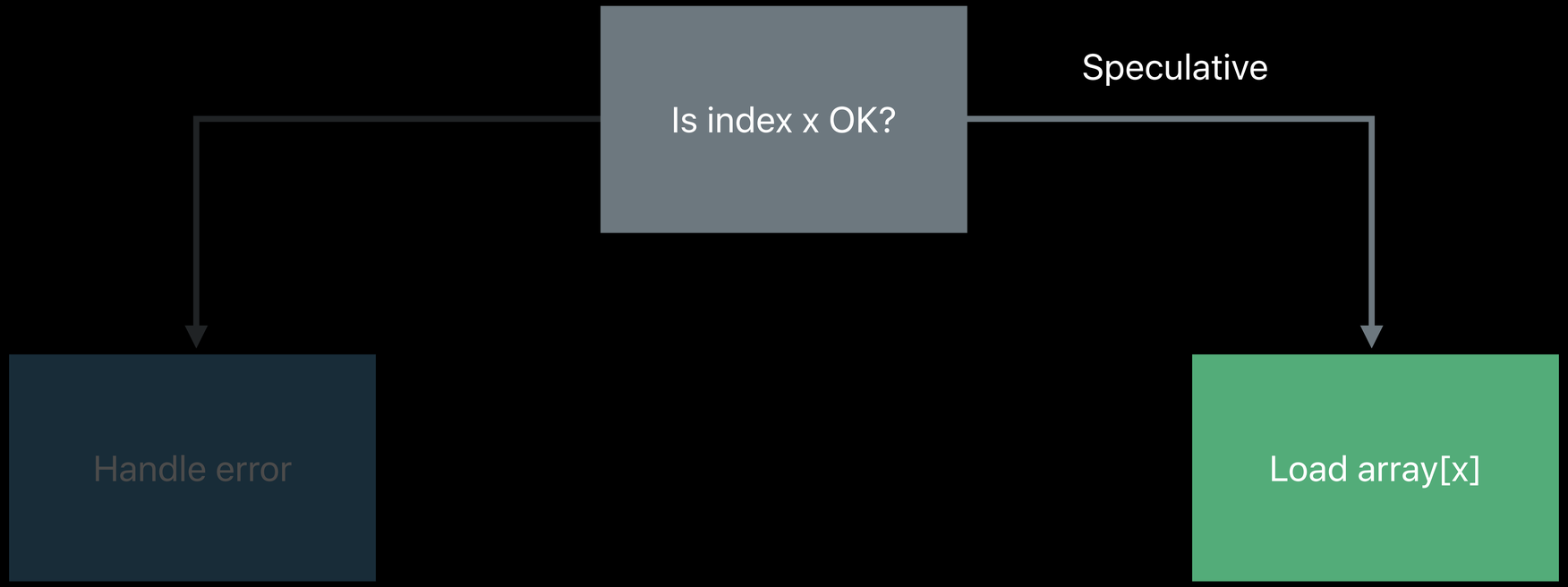
- WKWebView
- Content Security Policy
- HttpOnly cookies
- SameSite cookies
- Cross-Origin-Resource-Policy

Speculative Execution Defined

Speculative Execution Defined

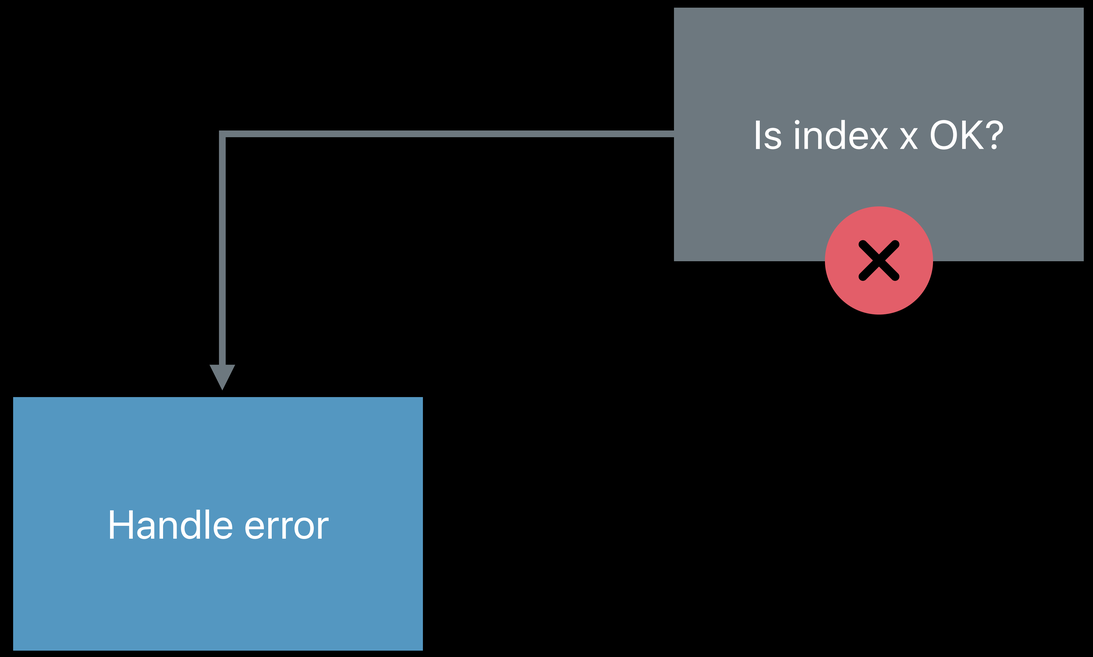


Speculative Execution Defined

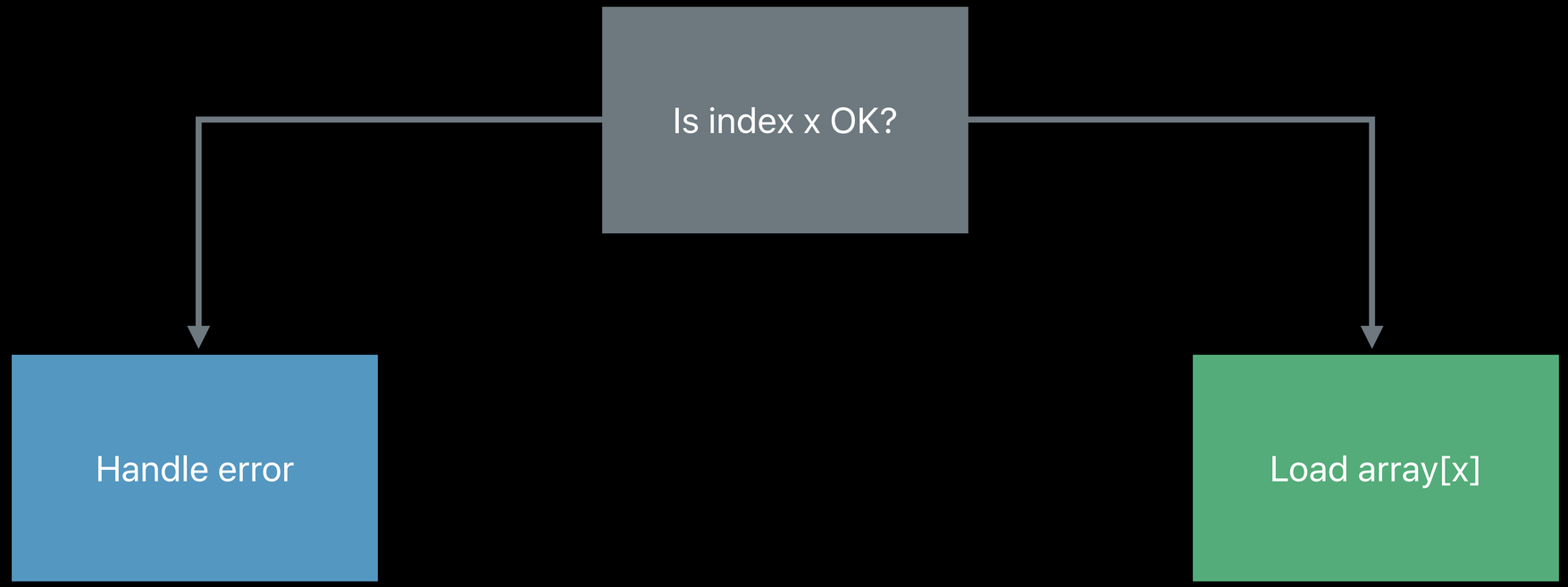


Speculative Execution Defined

Speculative Execution Defined

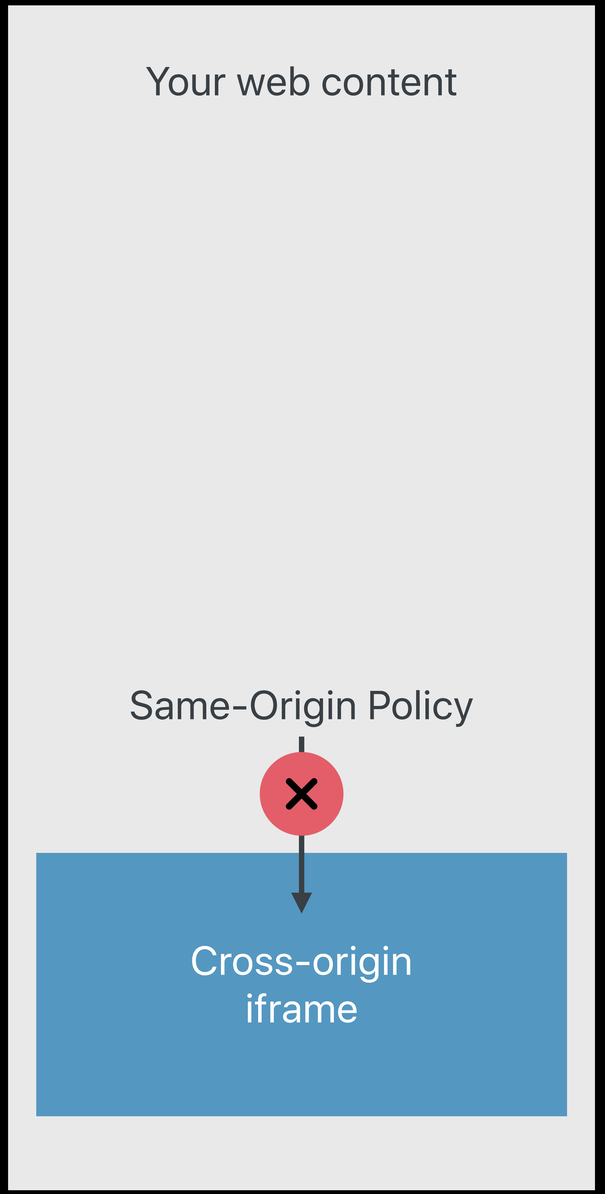


Speculative Execution Attacks

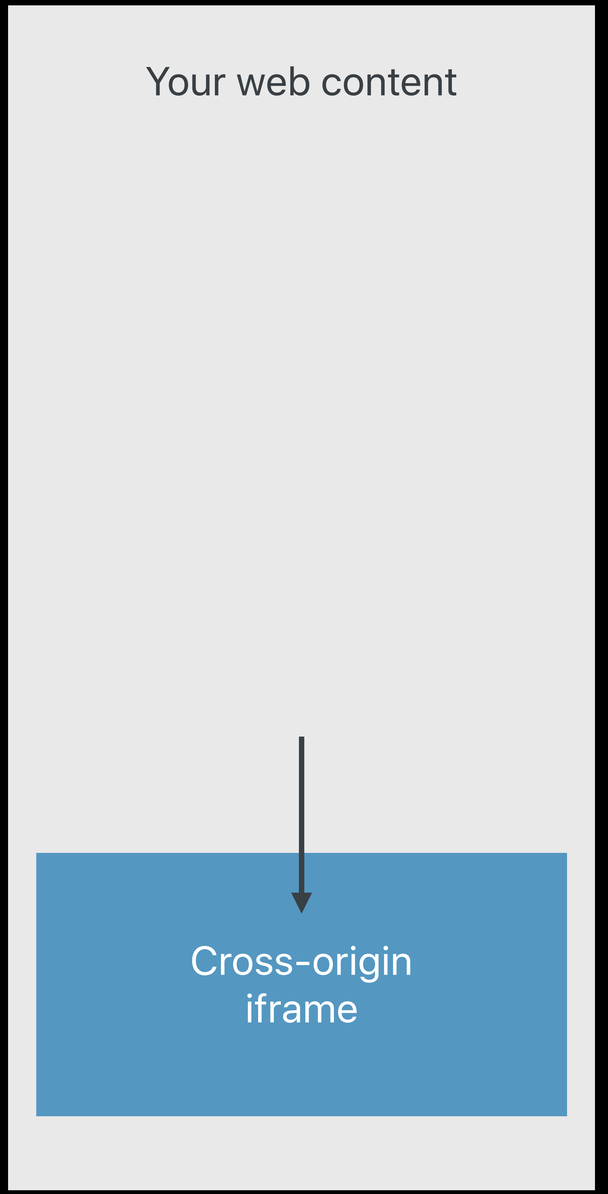


Cache effects of the speculative execution can be measured and data leaked

Speculative Execution Attacks



Speculative Execution Attacks



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


9:41
evil.example page
Evil
Your web content
Your page
WebKit
WebKit
A fast, open source web browser engine.
WebKit is the web browser engine used by Safari, Mail, App Store, and many other apps on macOS, iOS, and Linux. Get started contributing code, or reporting bugs.
Private
Done

WKWebView

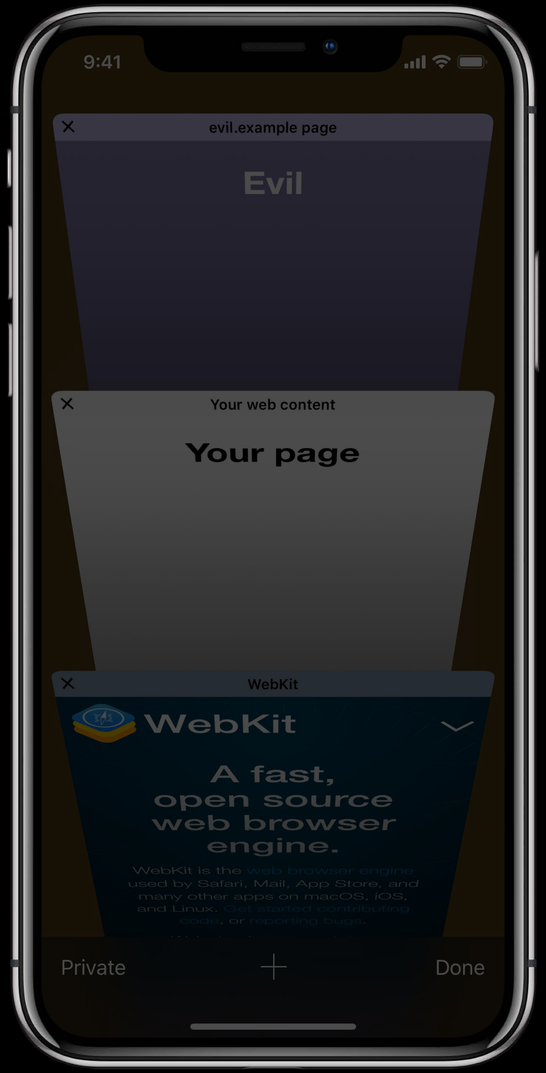
evil.example content process

Your web content process

webkit.org content process

WKWebView

Network process



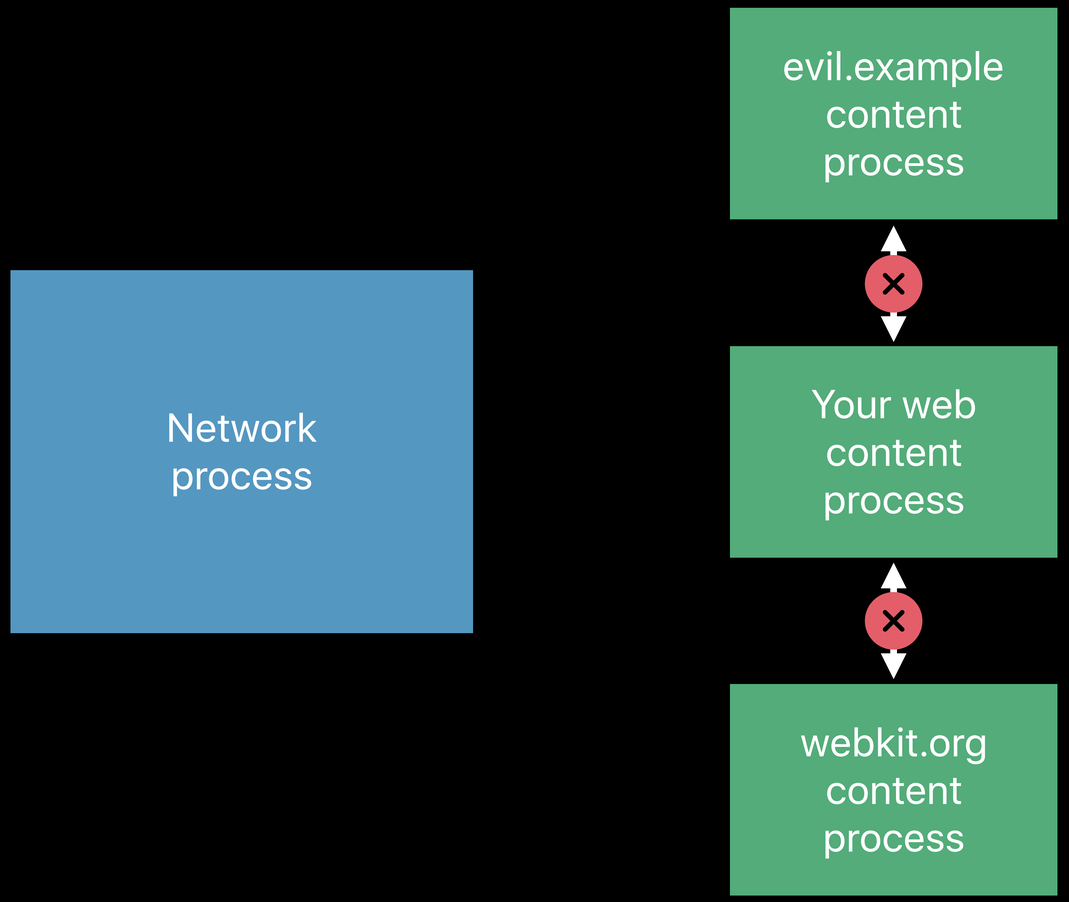
evil.example content process

Your web content process

webkit.org content process

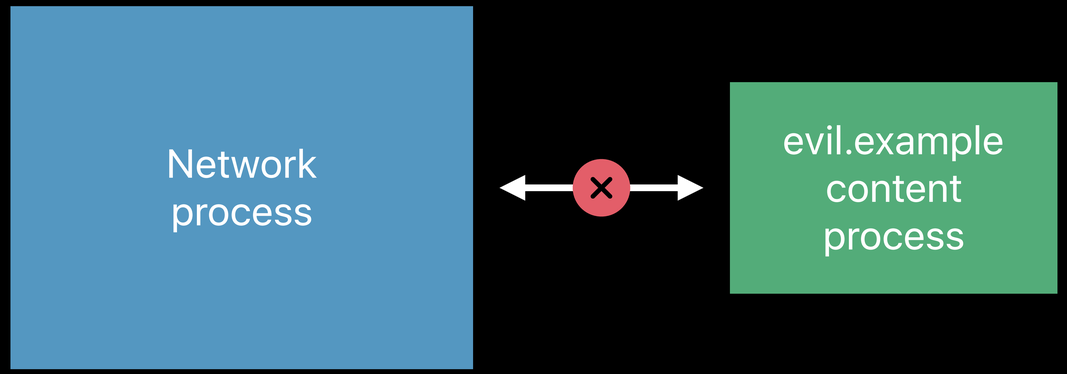
WebKit

Architecture



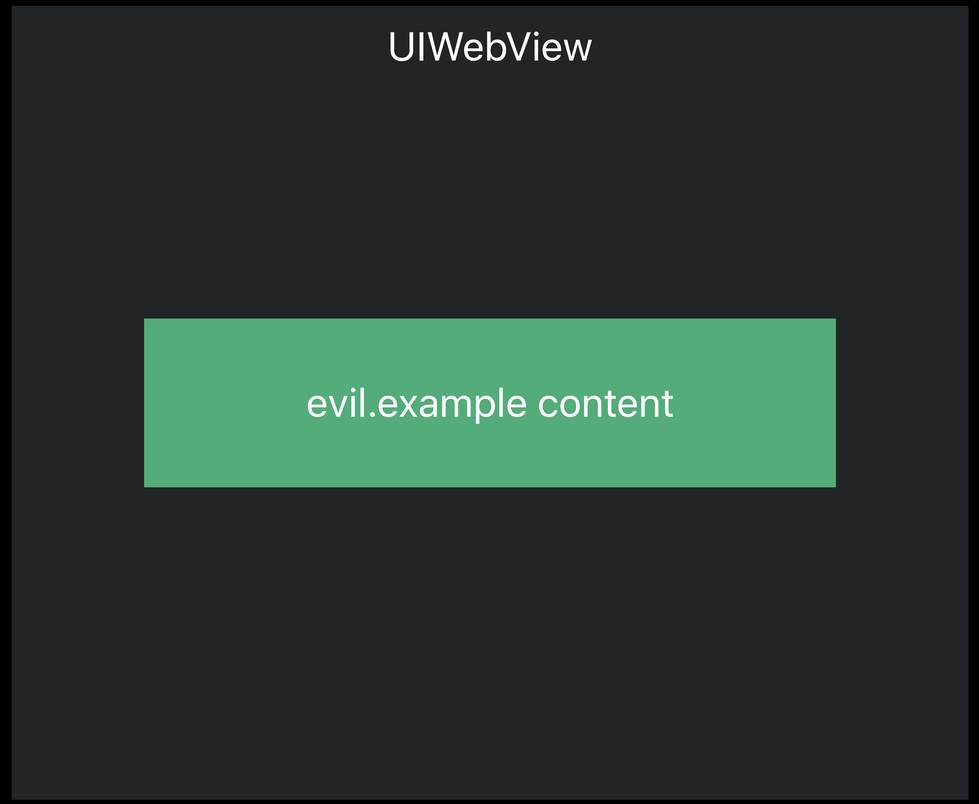
WebKit

Architecture

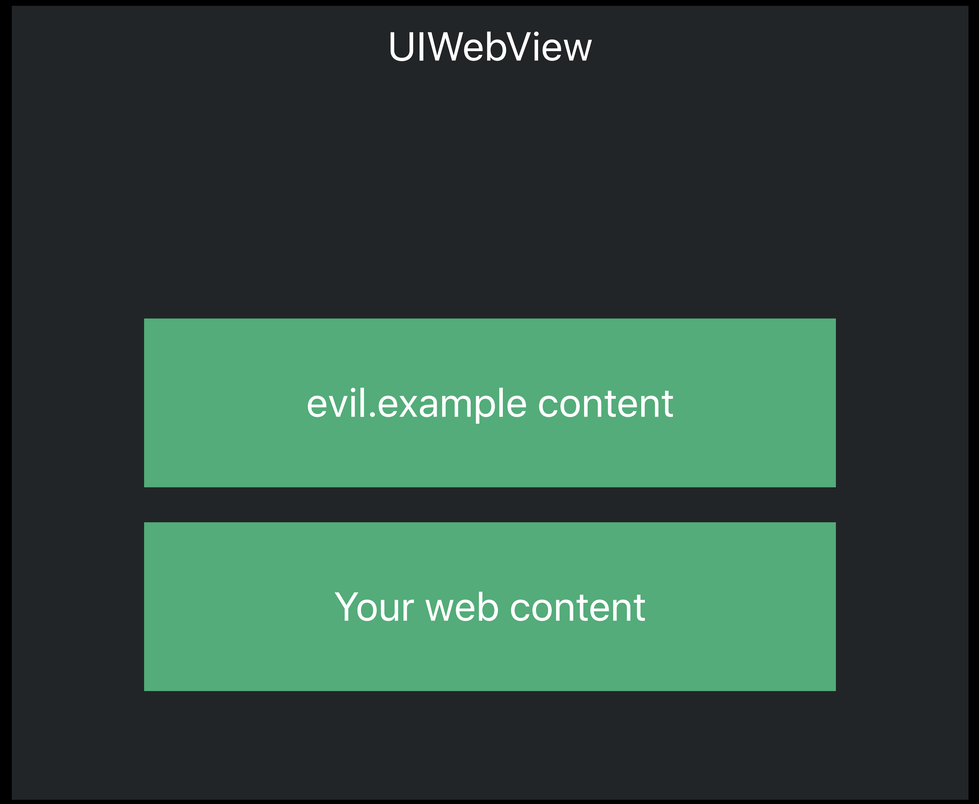


UIWebView Versus WKWebView

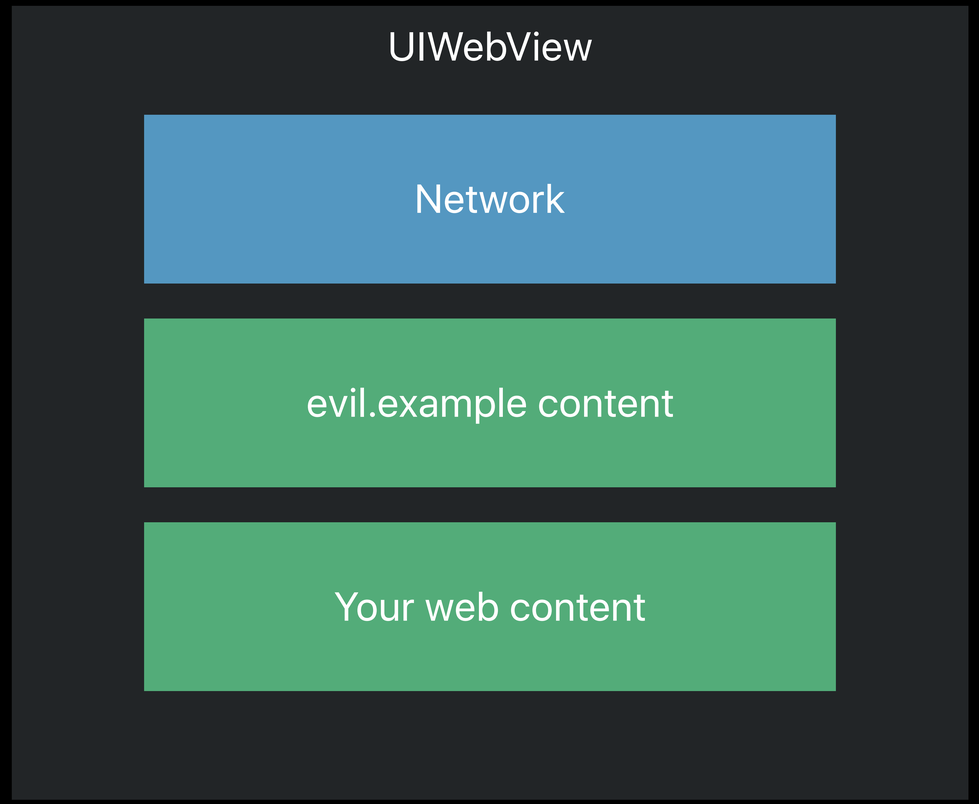
UIWebView Versus WKWebView



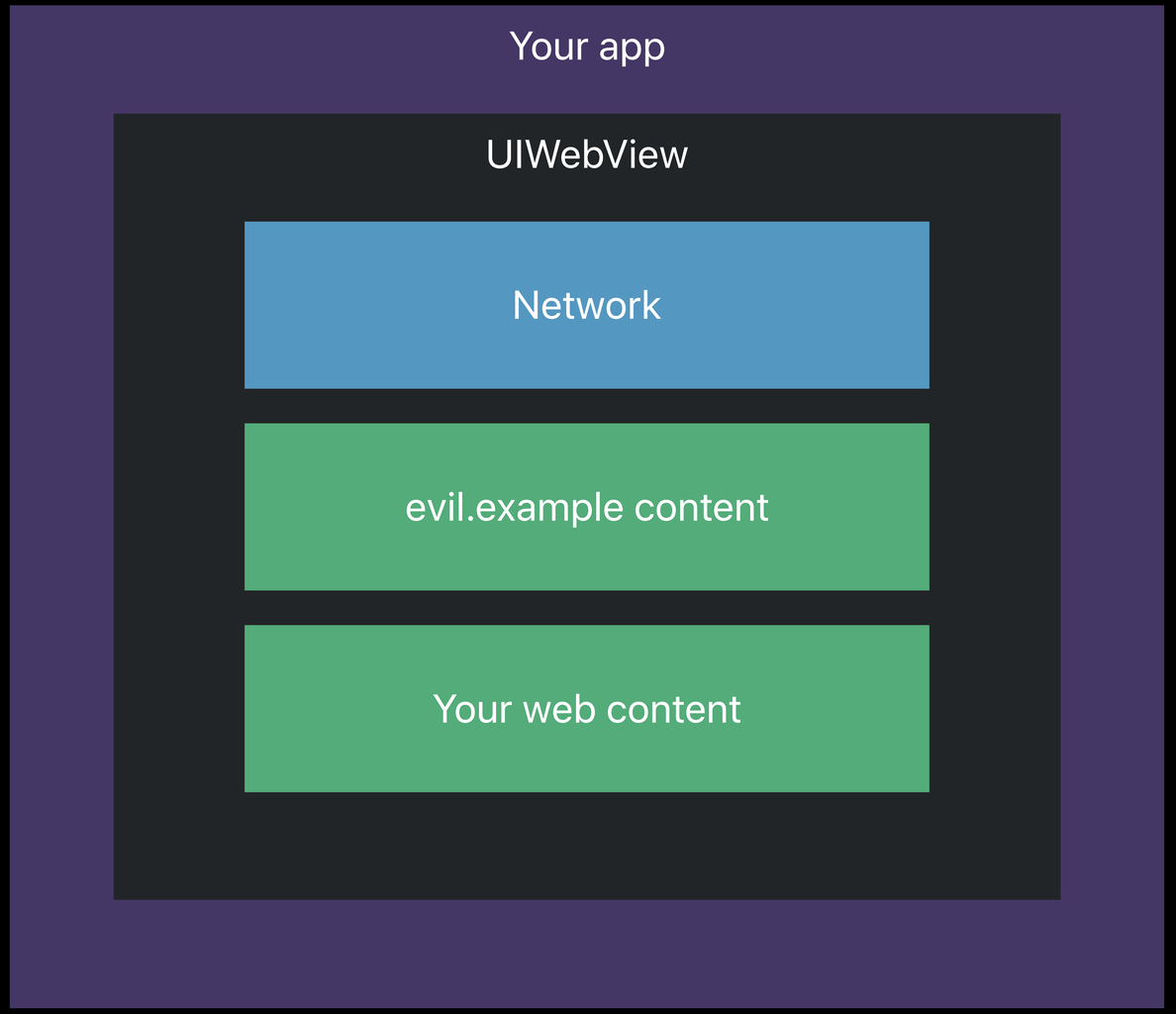
UIWebView Versus WKWebView



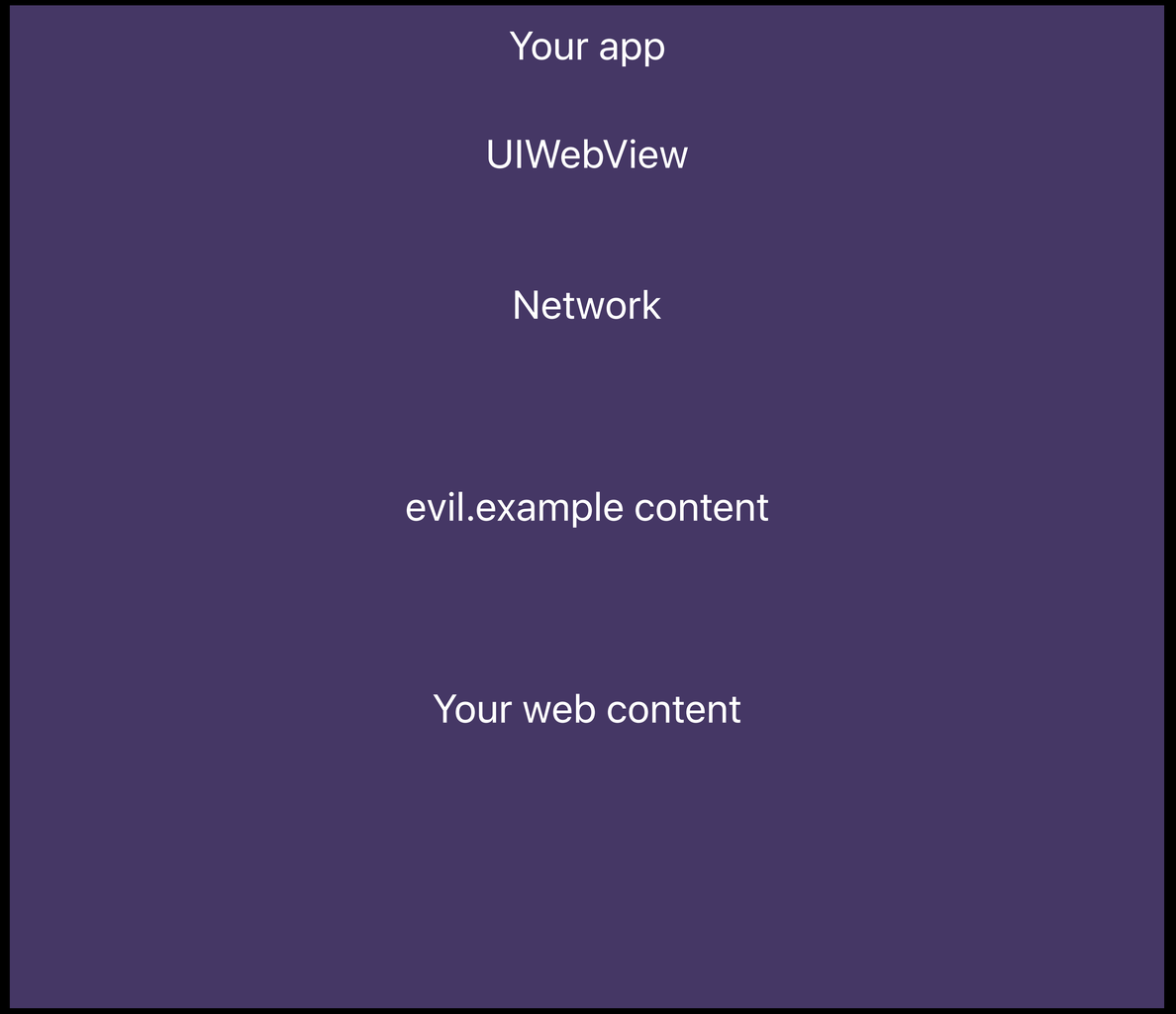
UIWebView Versus WKWebView



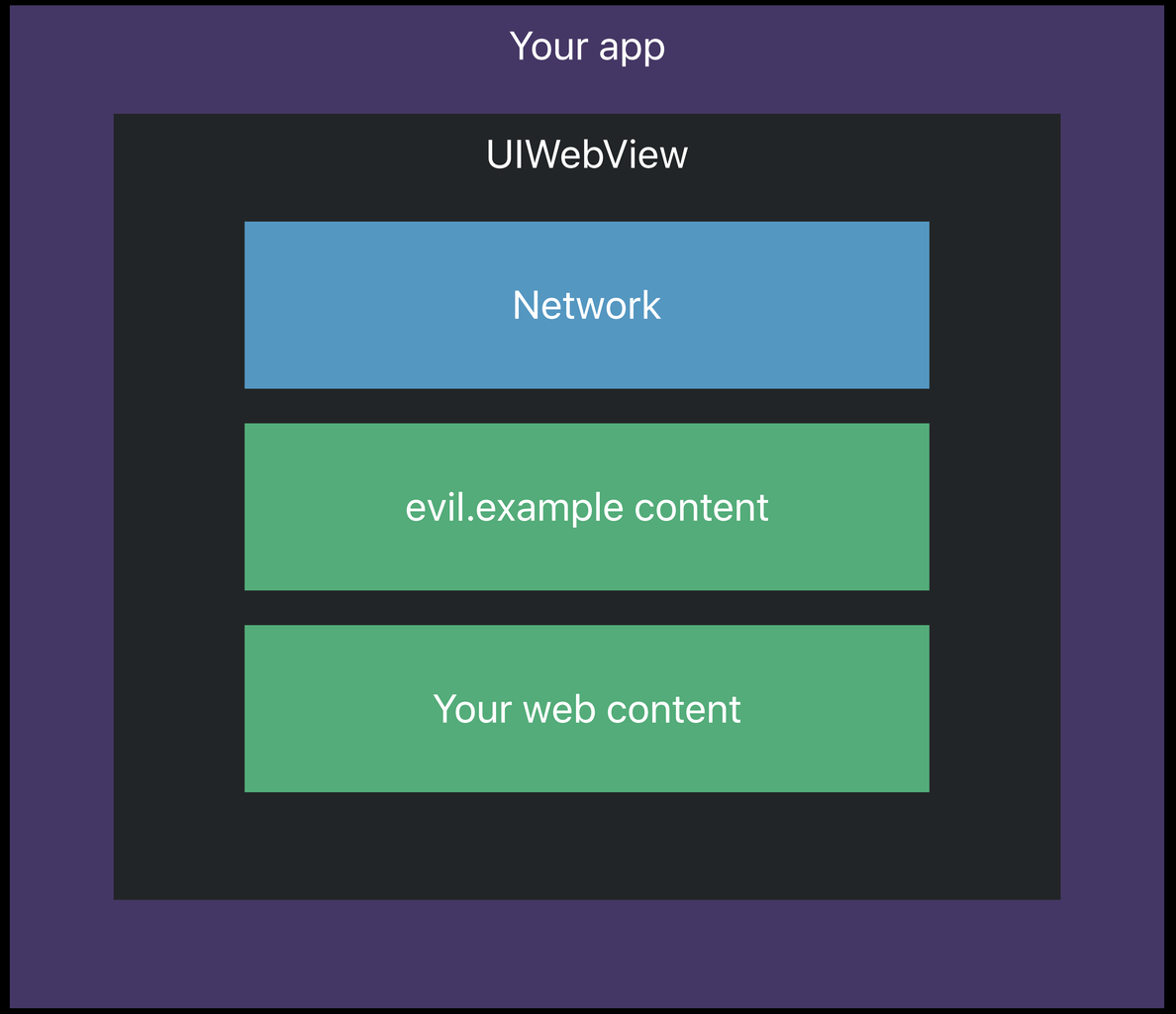
UIWebView Versus WKWebView



UIWebView Versus WKWebView

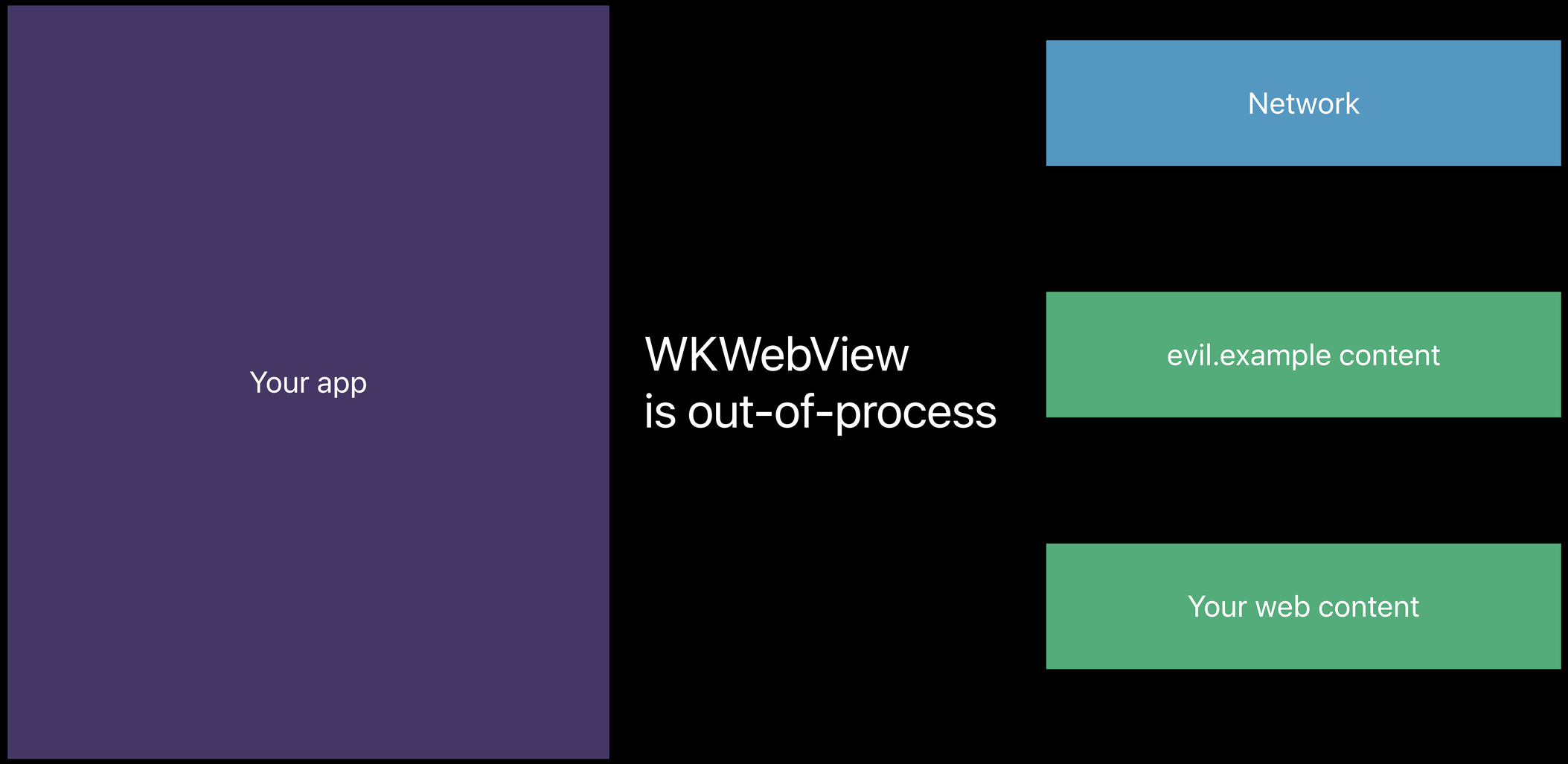


UIWebView Versus WKWebView



UIWebView Versus WKWebView

UIWebView Versus WKWebView



Speculative Execution Attacks

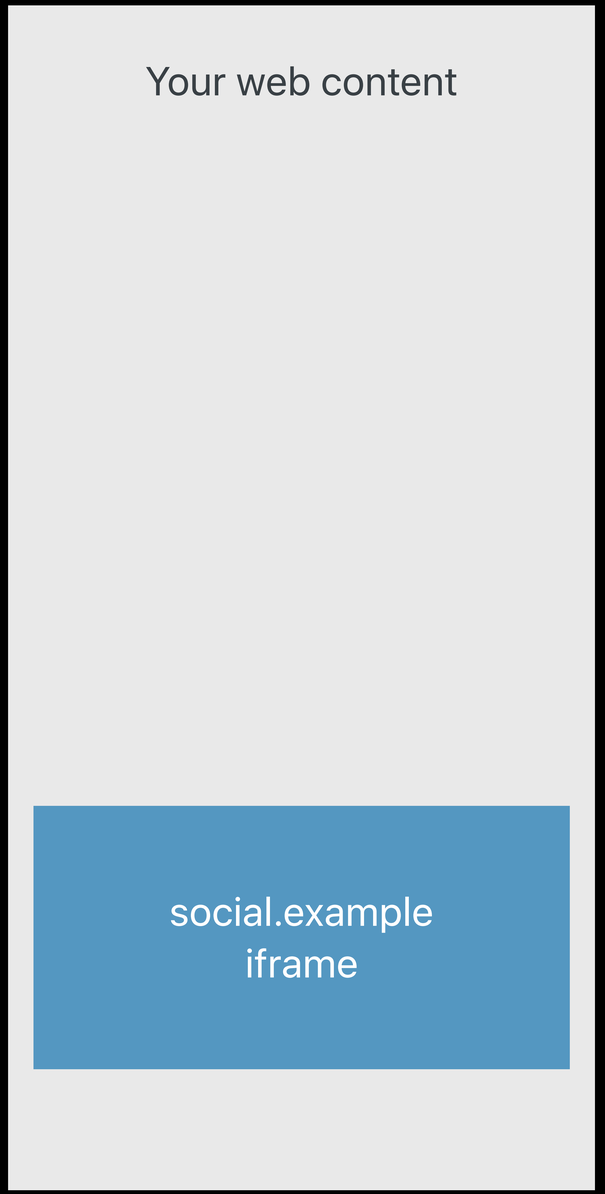
Speculative Execution and the Attack (Spectre)

Defenses:

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

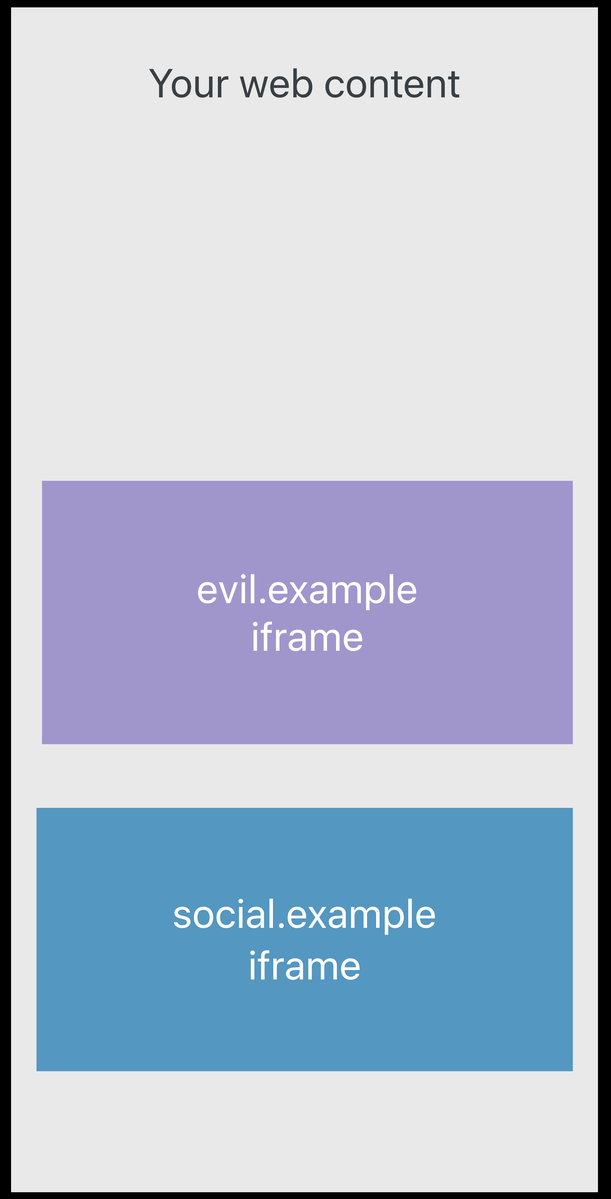
Speculative Execution Attacks

Content security policy



Speculative Execution Attacks

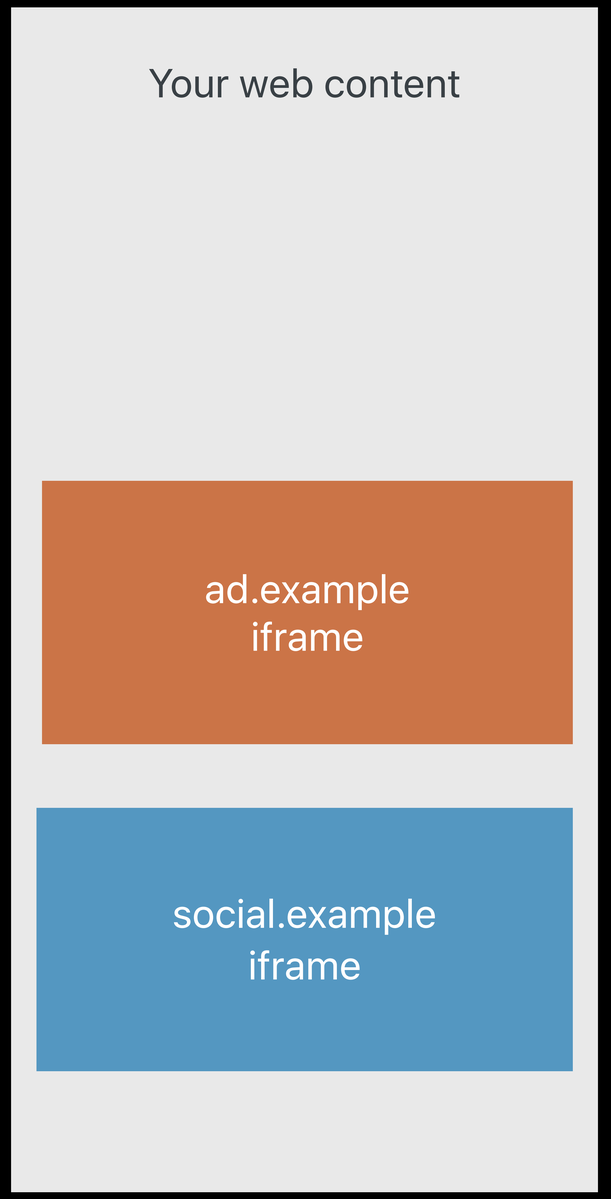
Content security policy



Injected iframe

Speculative Execution Attacks

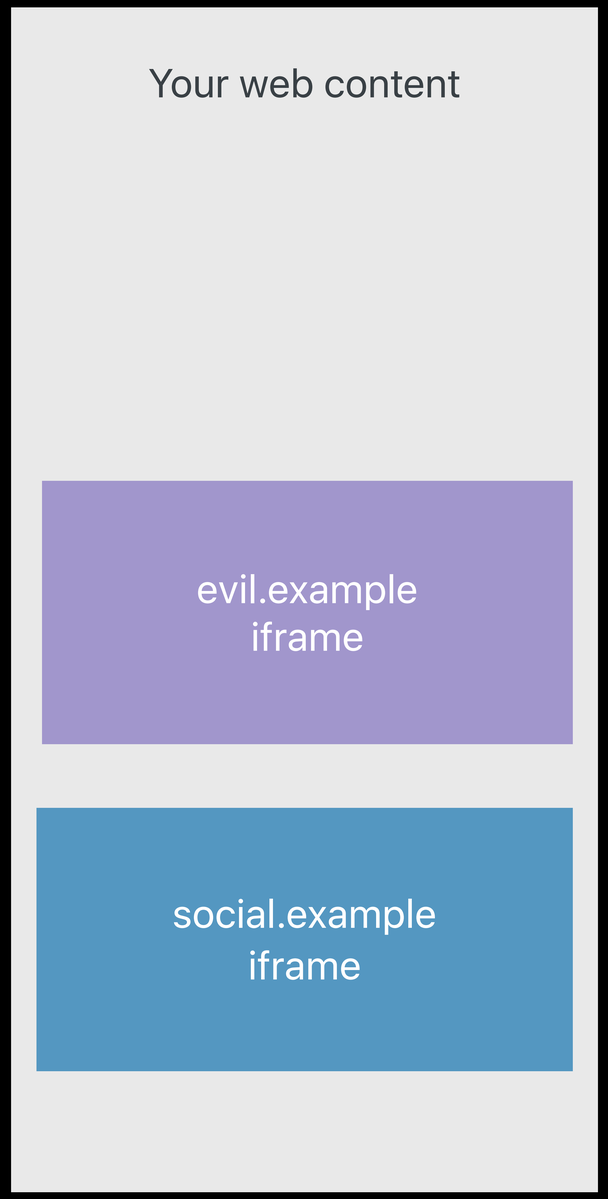
Content security policy



Speculative Execution Attacks

Content security policy

Redirected iframe

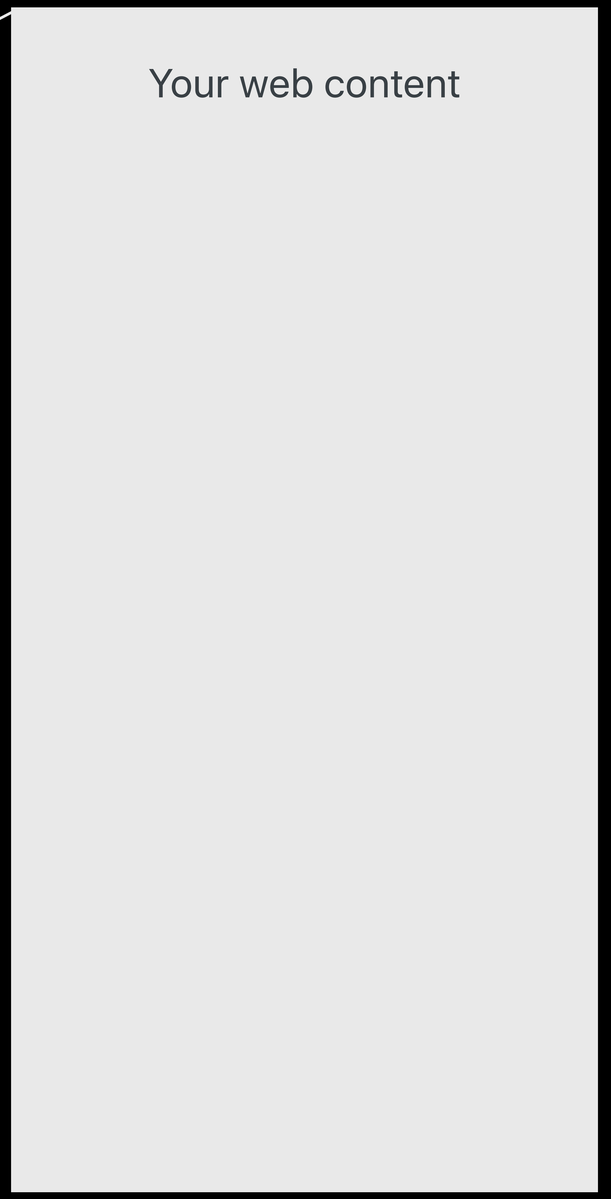


Speculative Execution Attacks

Content security policy

HTTP response:

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

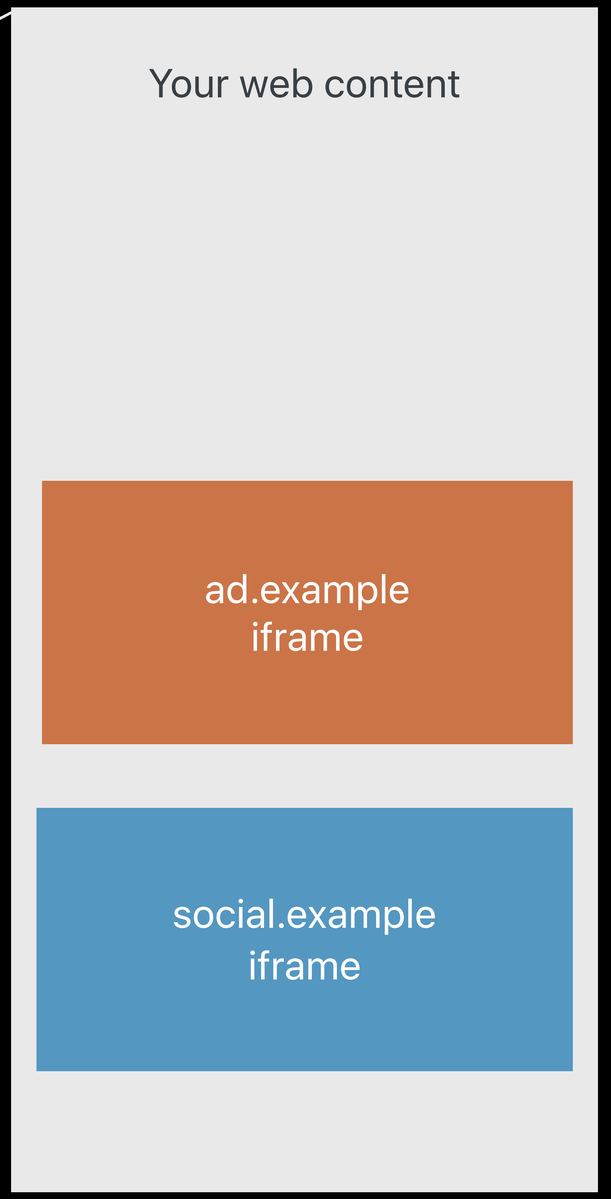


Speculative Execution Attacks

Content security policy

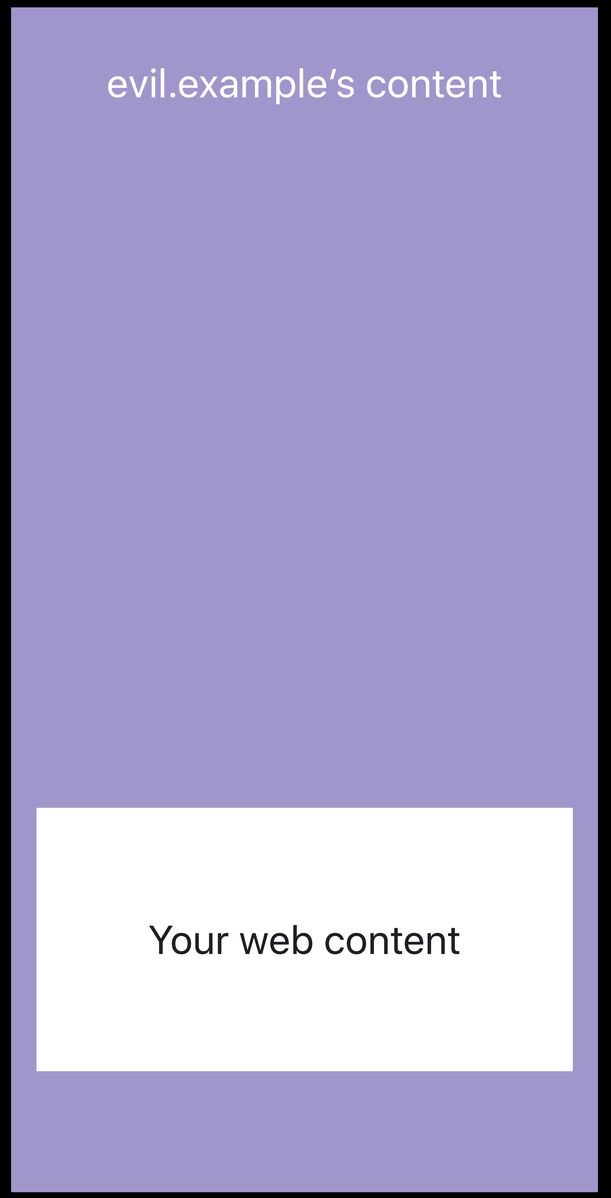
HTTP response:

```
:status: 200
Content-Security-Policy:
 default-src 'self';
 frame-src ad.example
           social.example
```



Speculative Execution Attacks

Content security policy

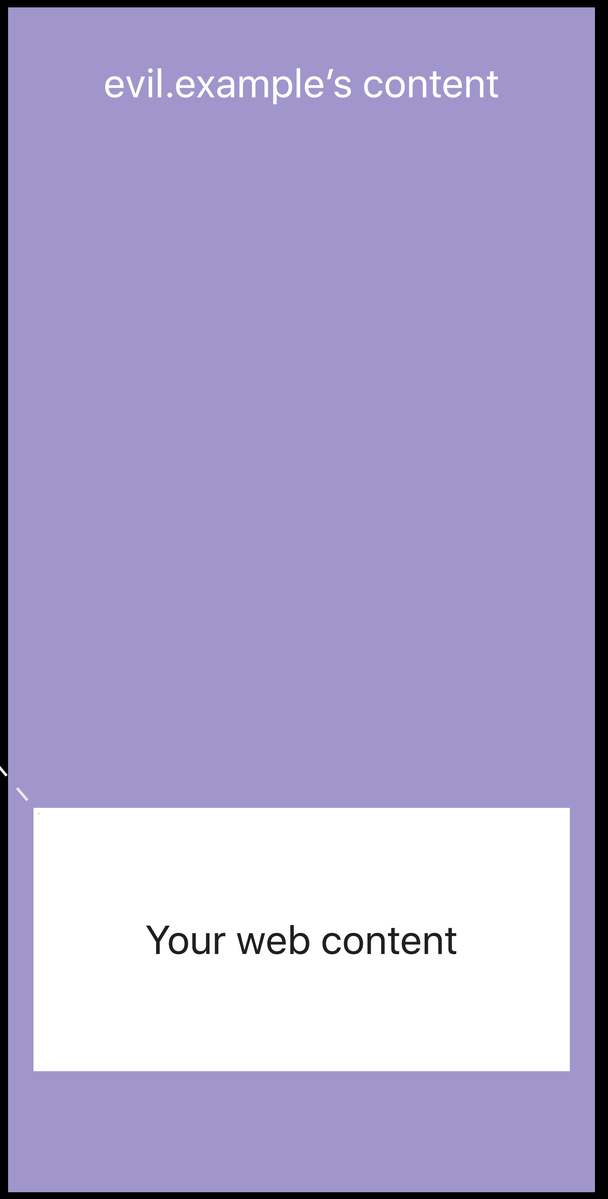


Speculative Execution Attacks

Content security policy

HTTP response:

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



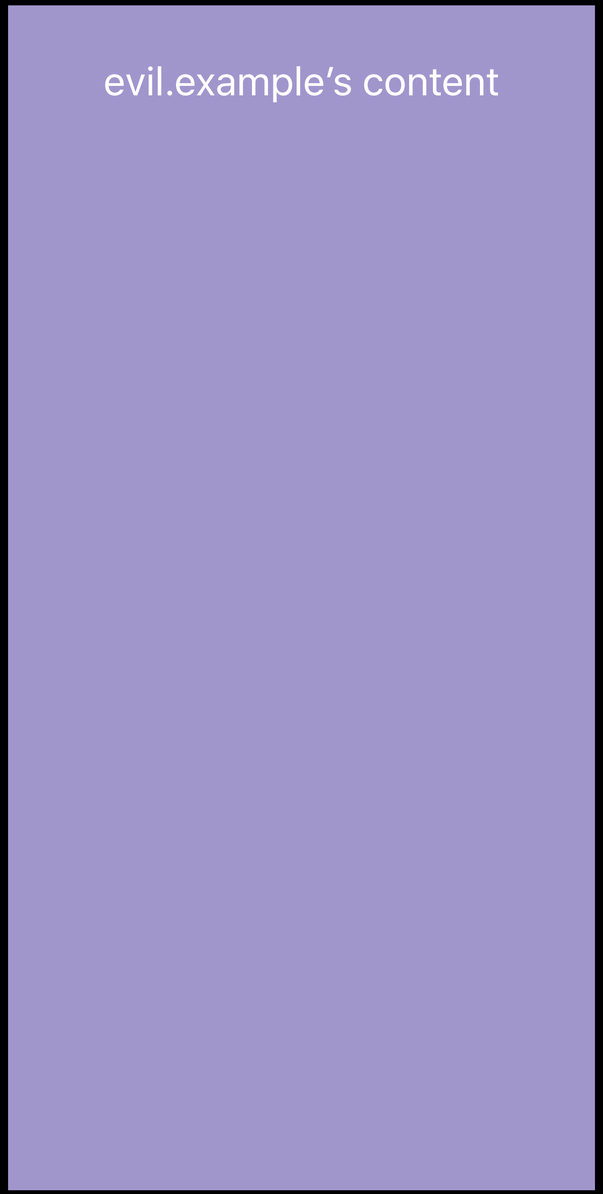
Speculative Execution Attacks

Content security policy

HTTP response:

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

Blocked in the Network Process



Speculative Execution Attacks

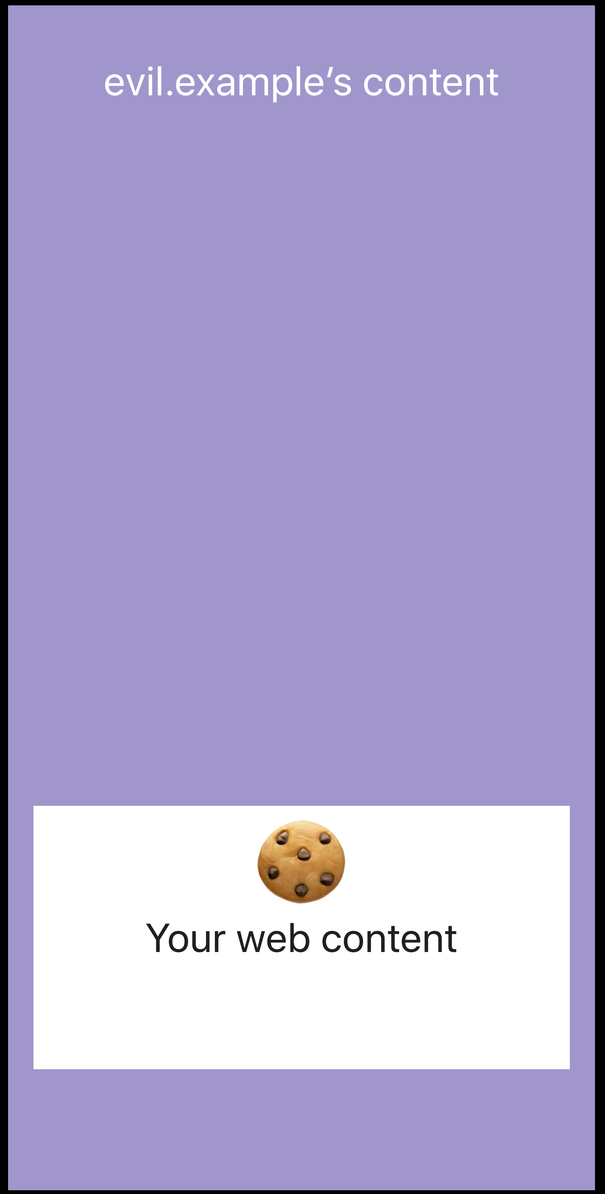
Speculative Execution and the Attack (Spectre)

Defenses:

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

Speculative Execution Attacks

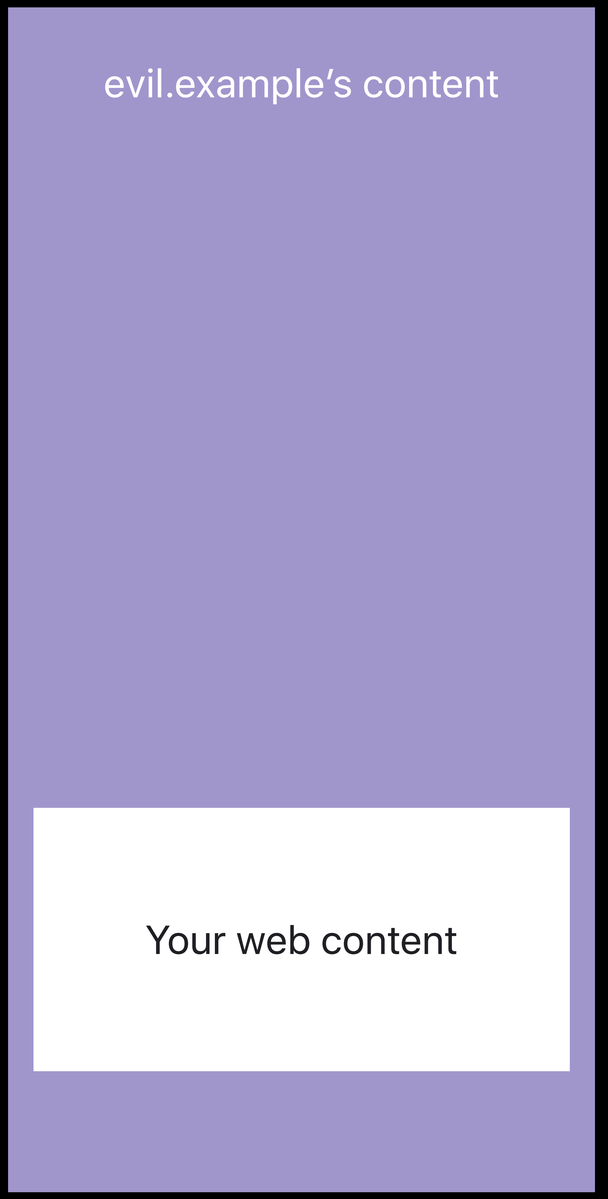
HttpOnly cookies



Speculative Execution Attacks

HttpOnly cookies

HttpOnly cookies stay
in the network process



Speculative Execution Attacks

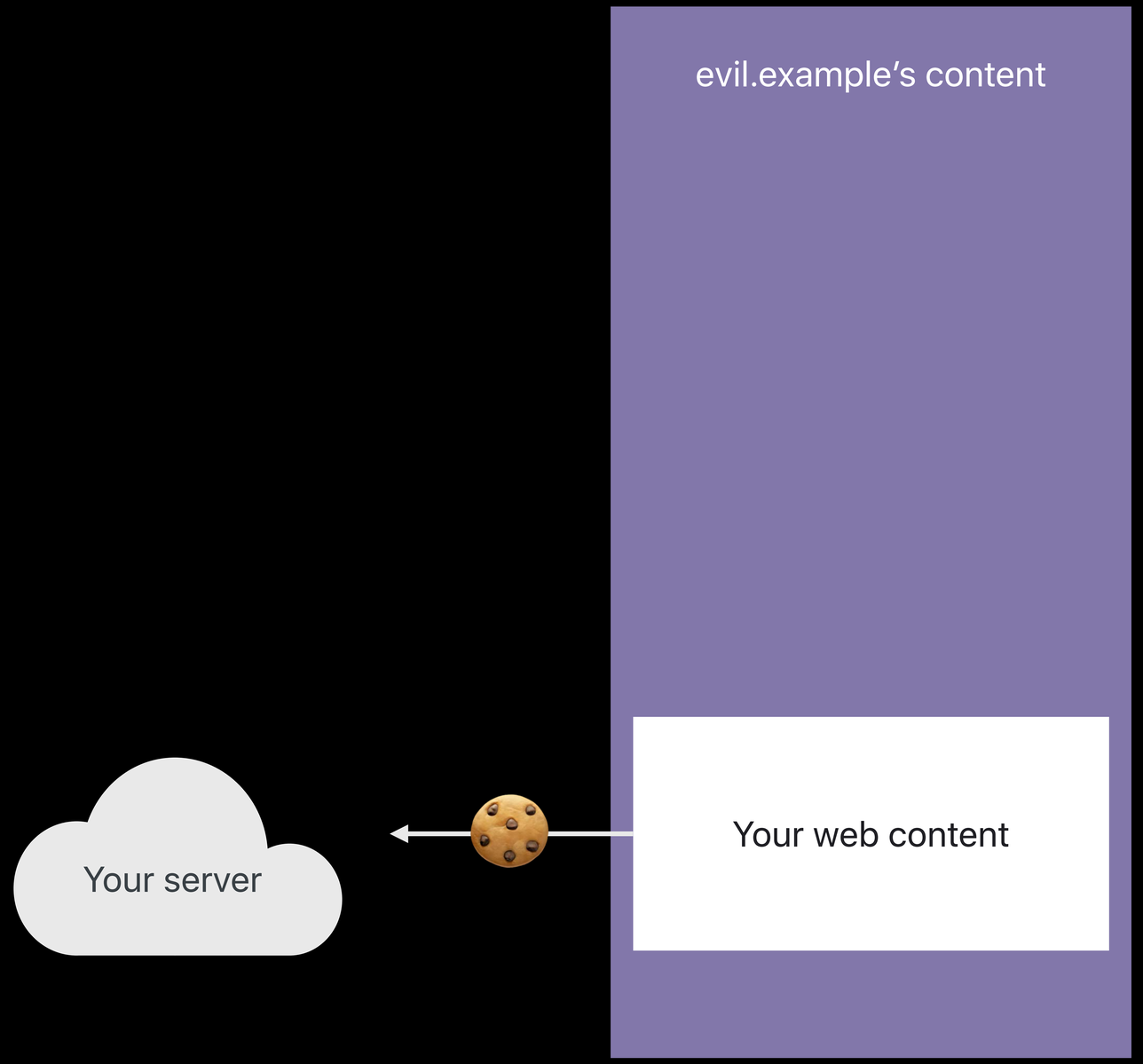
Speculative Execution and the Attack (Spectre)

Defenses:

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

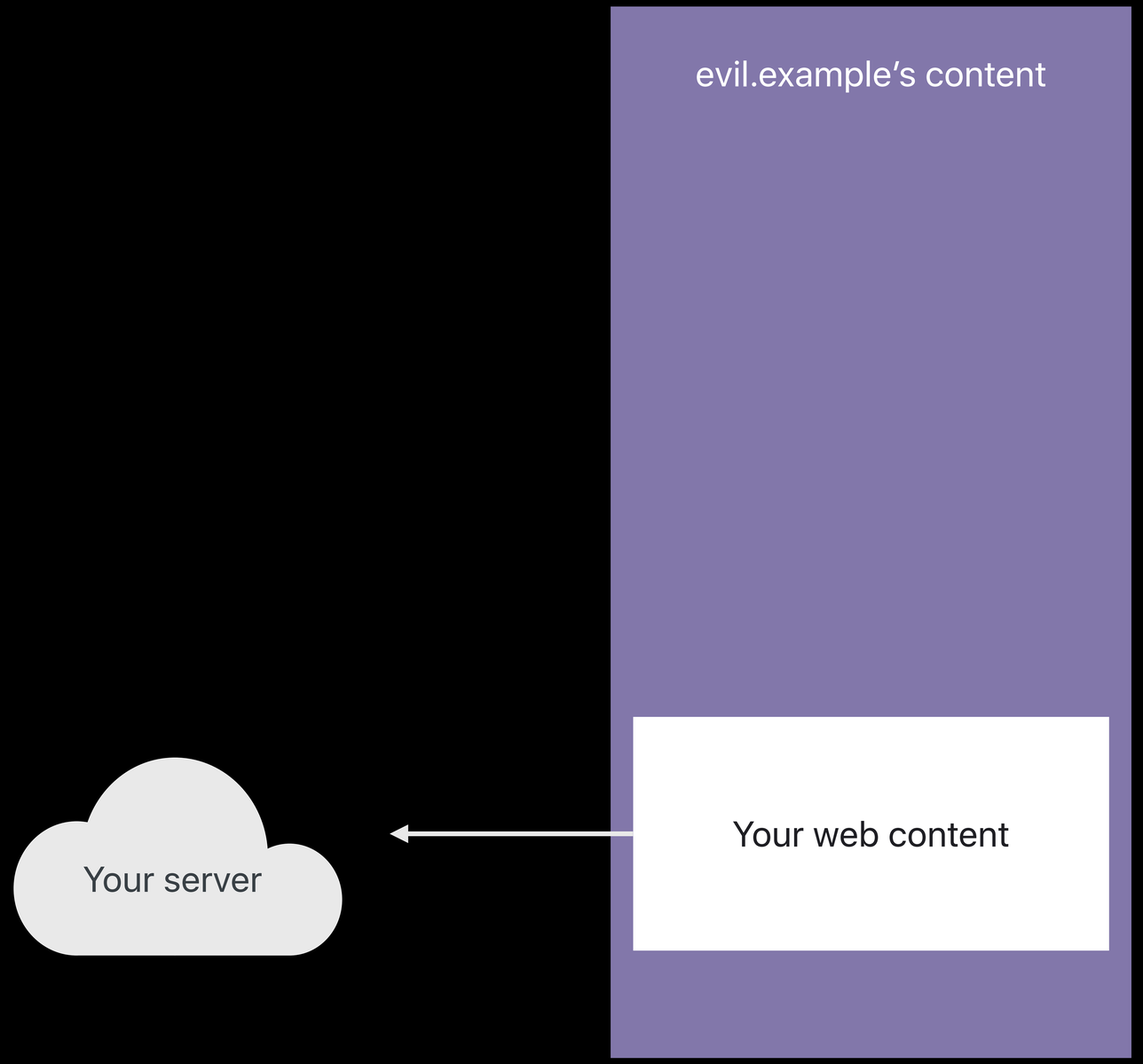
Speculative Execution Attacks

SameSite cookies



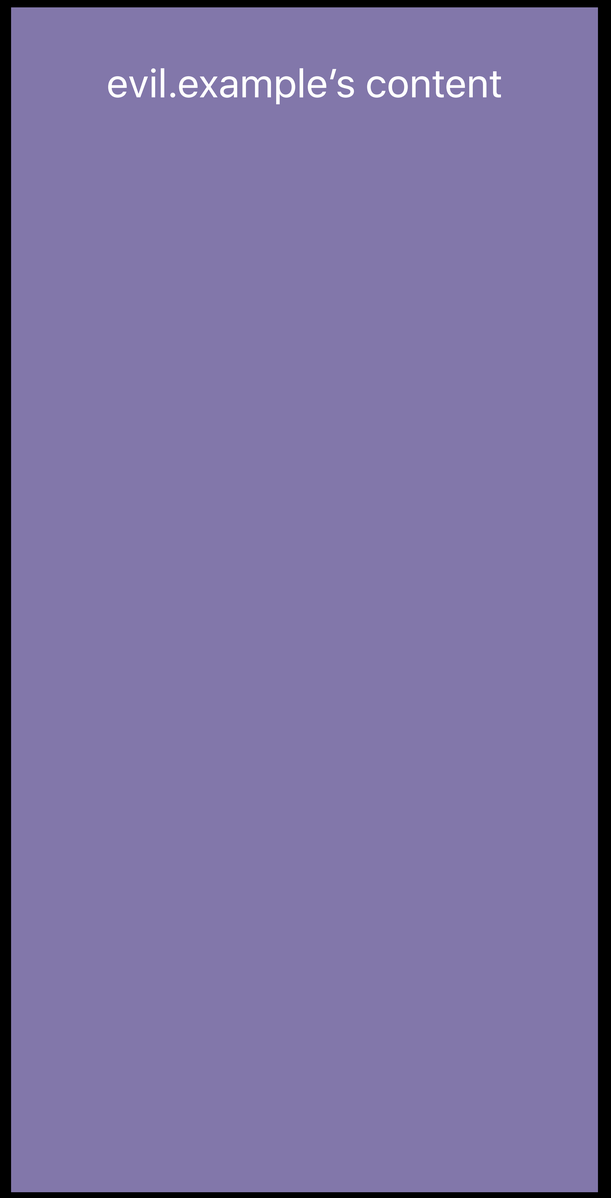
Speculative Execution Attacks

SameSite cookies



Speculative Execution Attacks

SameSite cookies



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

Someone else's content

Your server

HTTP response:

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

Your server

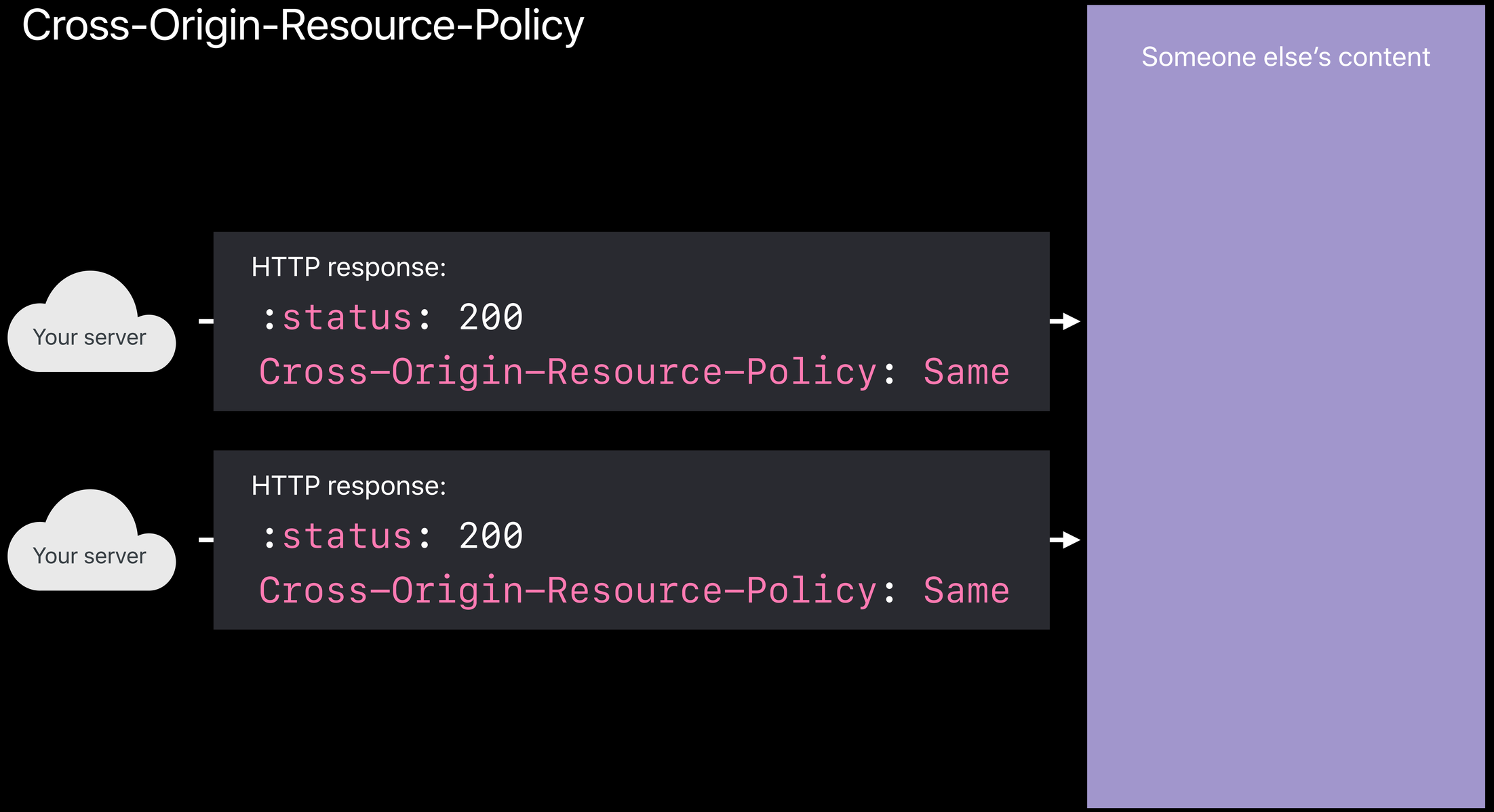
HTTP response:

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

```
function onload() {
  ...
}
```

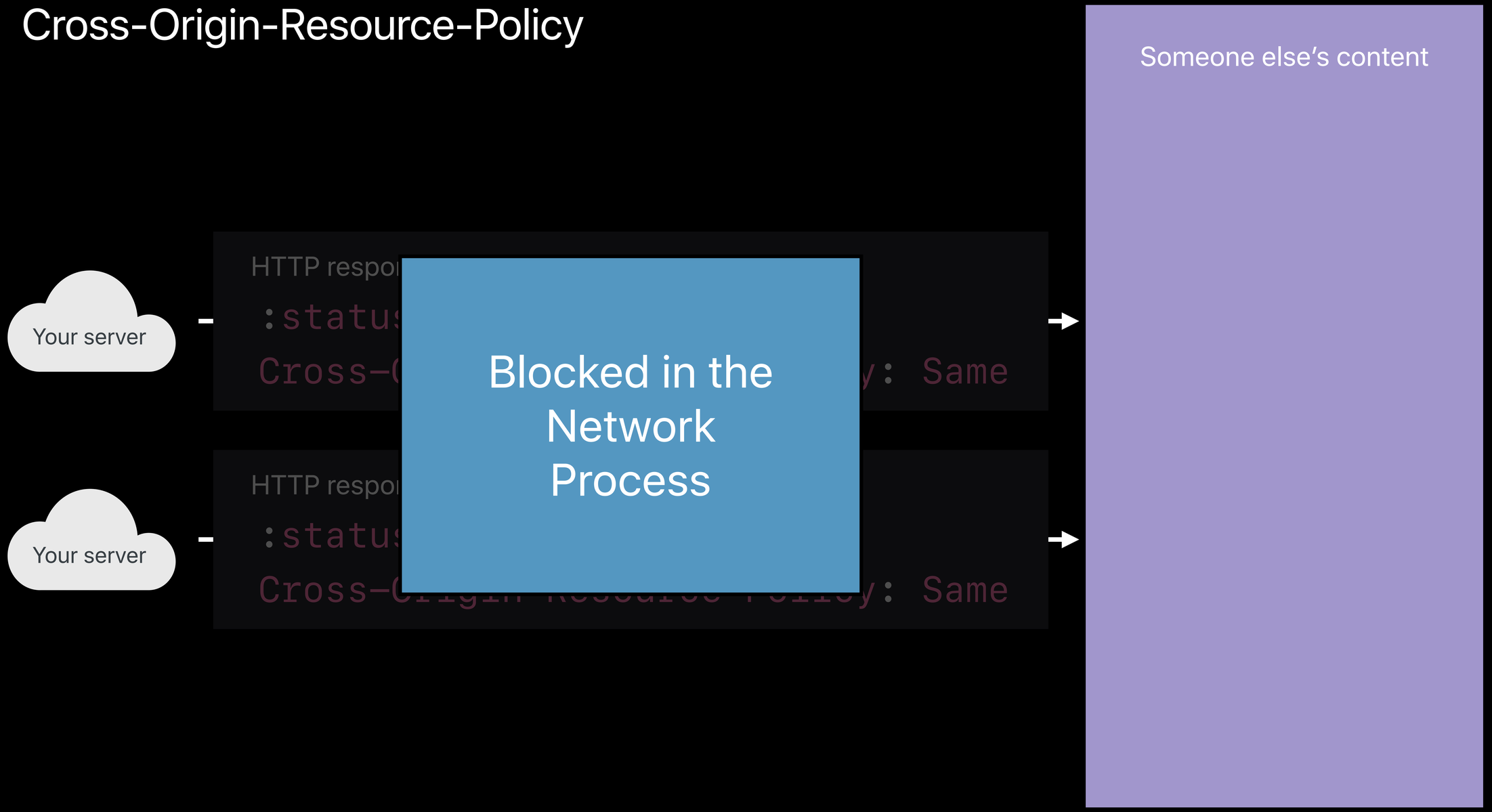
Speculative Execution Attacks

Cross-Origin-Resource-Policy



Speculative Execution Attacks

Cross-Origin-Resource-Policy



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

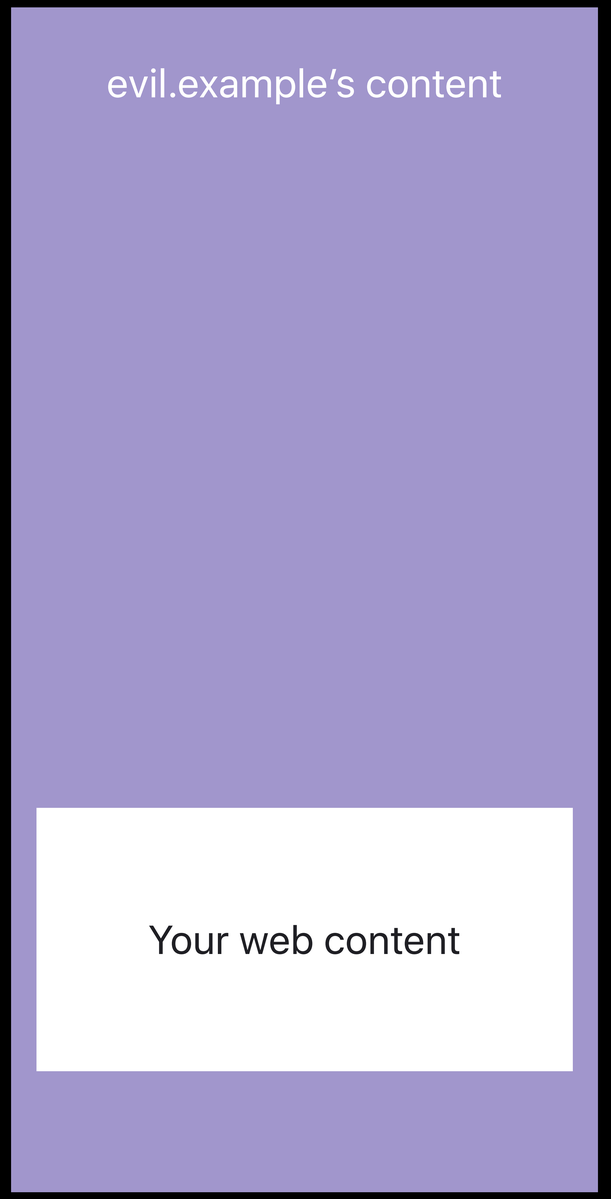
Malicious window navigation

Defense:

- Cross-Origin-Window-Policy header

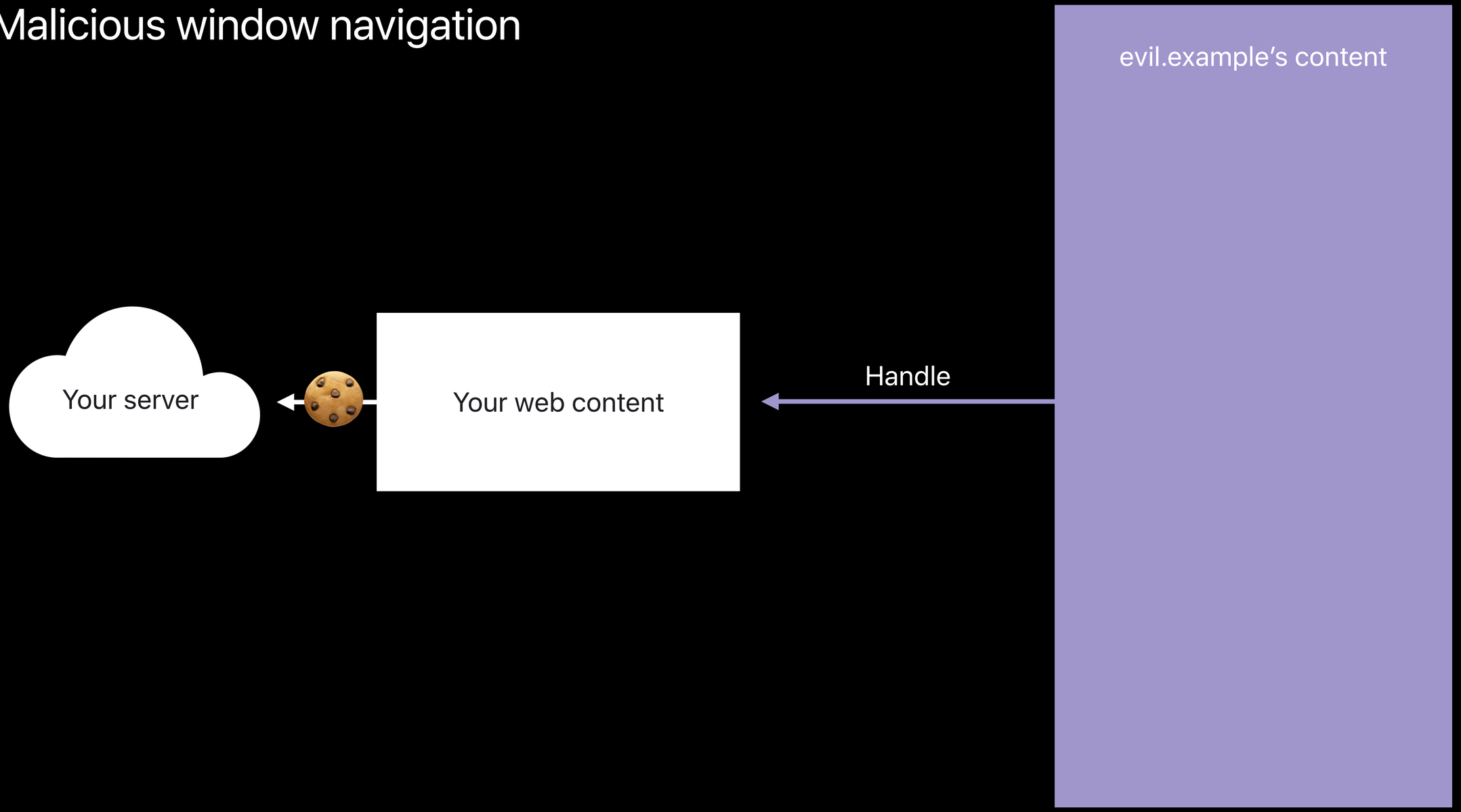
Window Control Attacks

Malicious window navigation



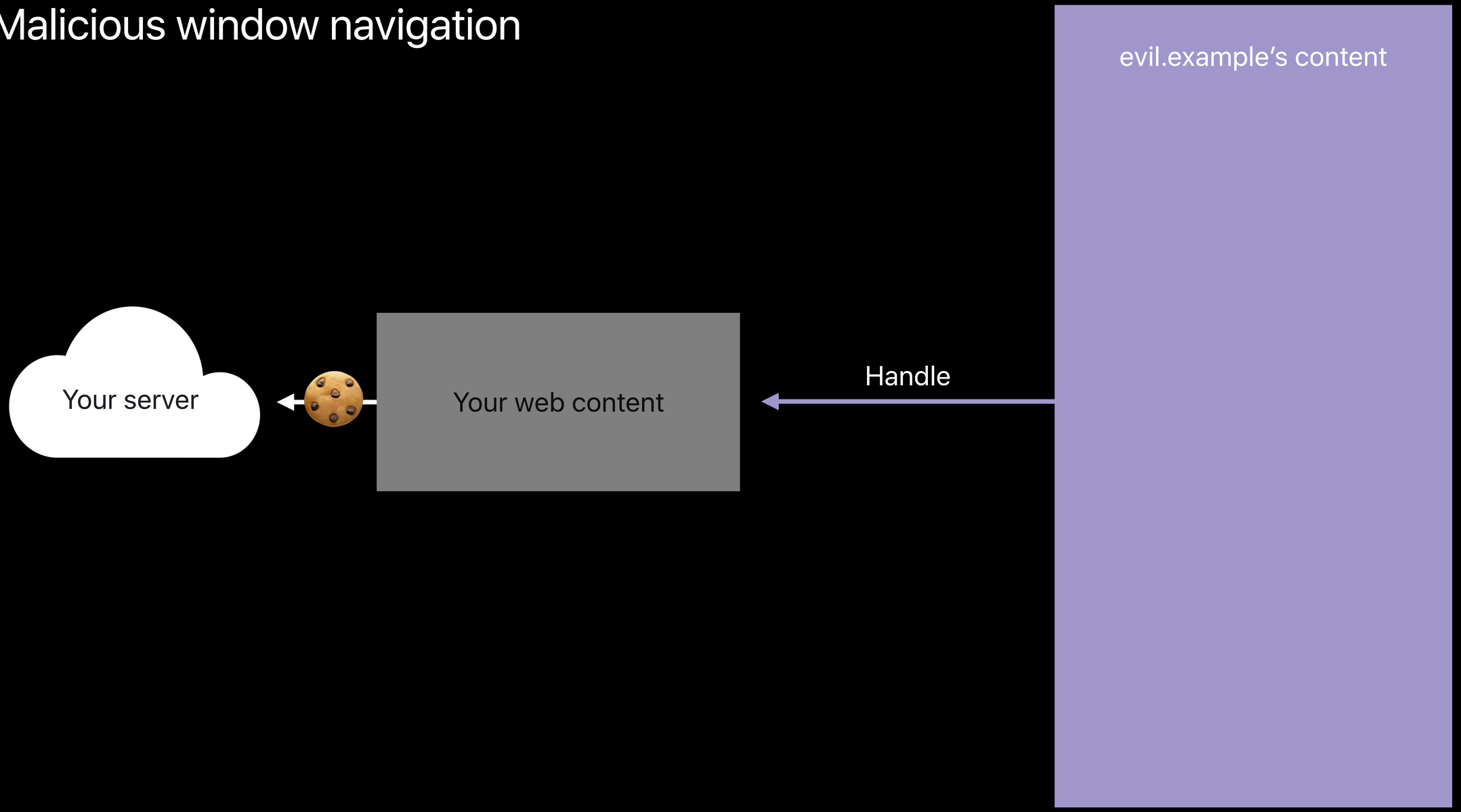
Window Control Attacks

Malicious window navigation



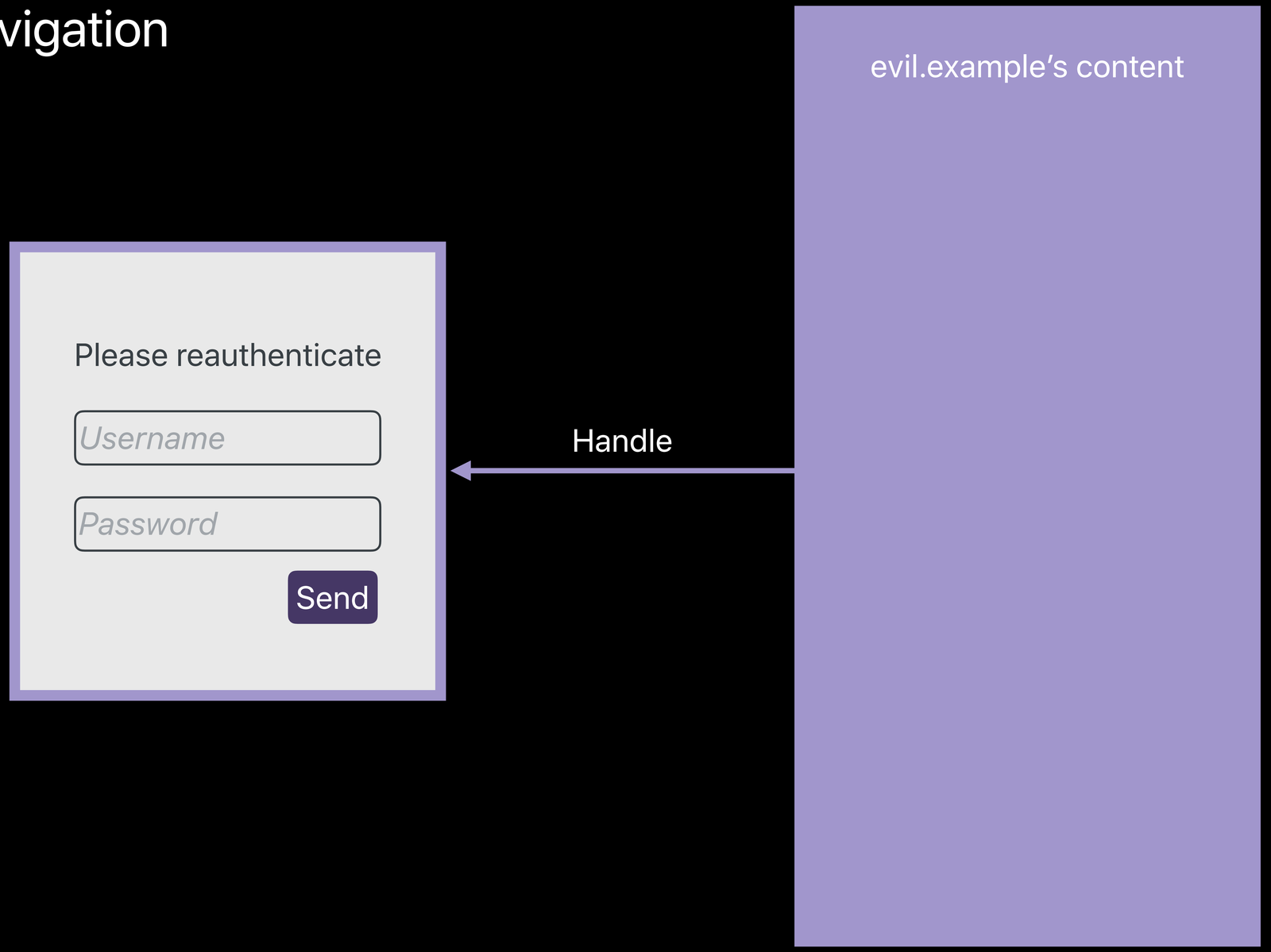
Window Control Attacks

Malicious window navigation



Window Control Attacks

Malicious window navigation



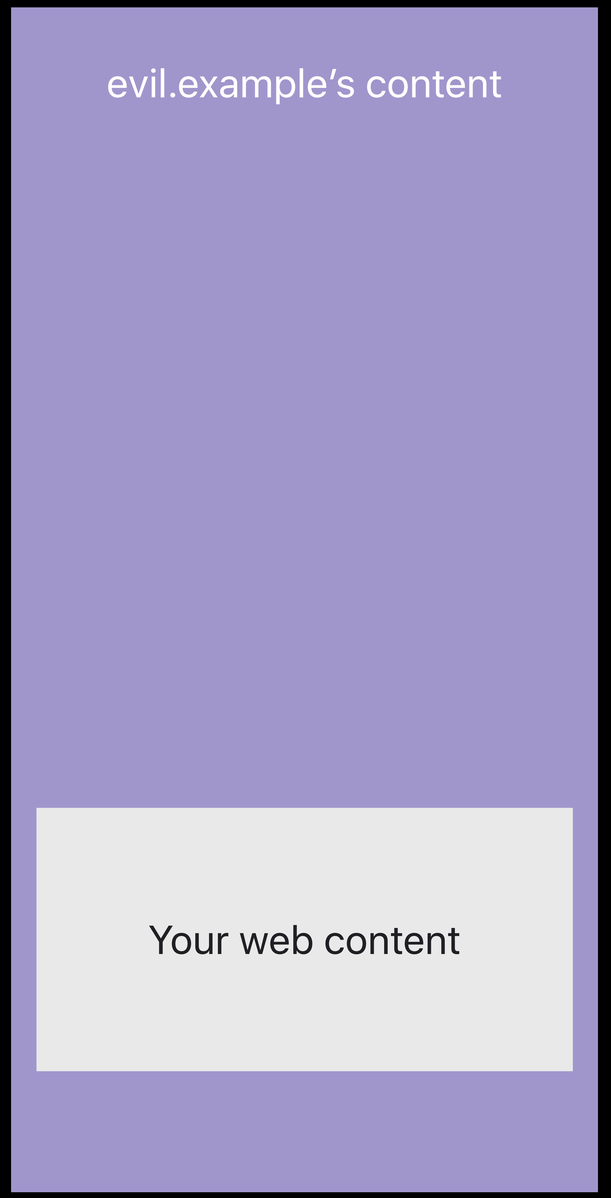
Window Control Attacks

Malicious window navigation

Defense:

- Cross-Origin-Window-Policy header

Window Control Attacks



Window Control Attacks

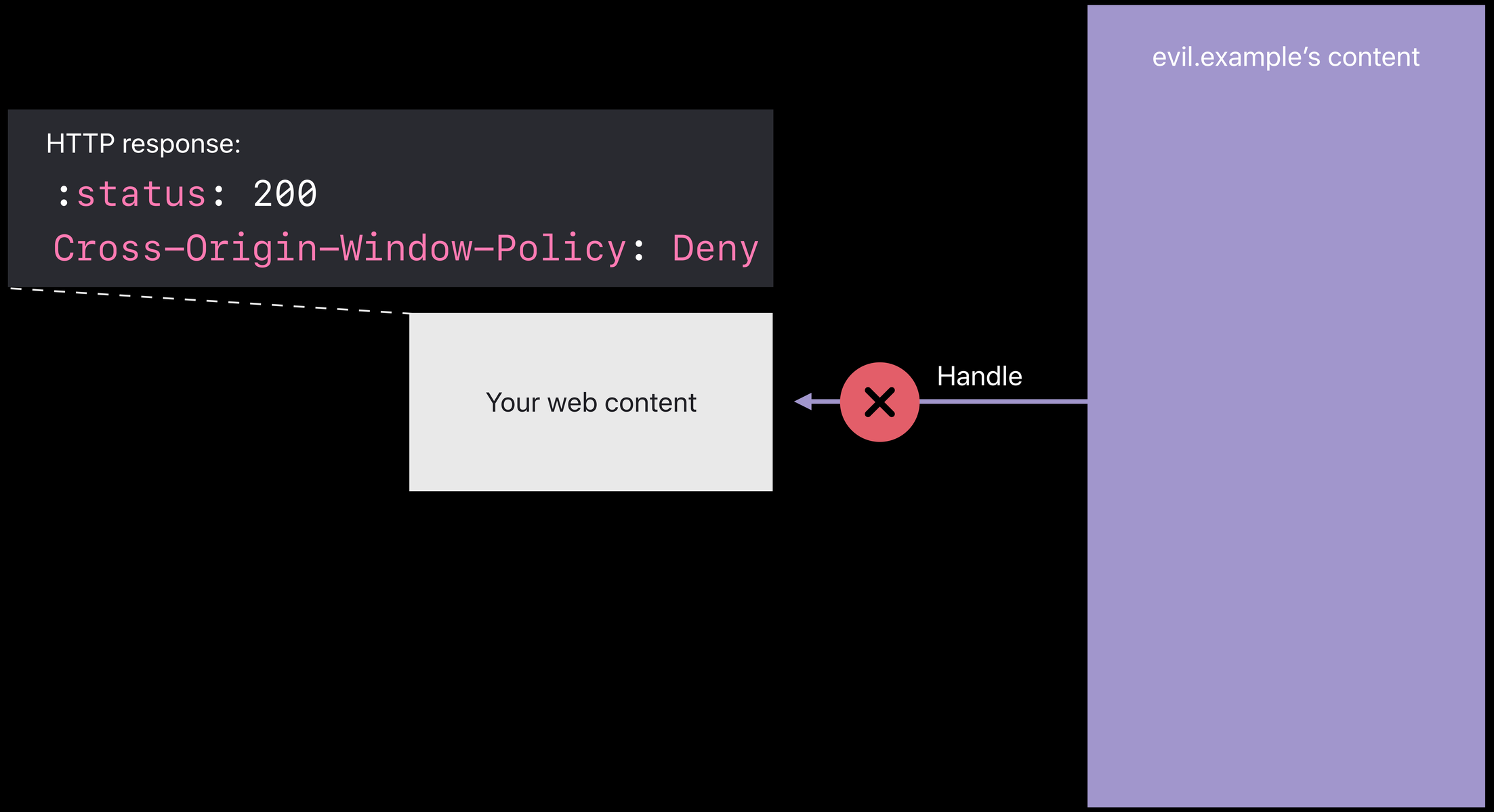
HTTP response:

```
:status: 200
Cross-Origin-Window-Policy: Deny
```

Your web content

evil.example's content

Window Control Attacks



Take Action

Take Action

Baseline

Use transport security (https, wss)

Mark cookies HttpOnly and Secure

Migrate from UIWebView to WKWebView

Take Action

	Content Security Policy
Cross-Site Scripting	✓
Compromised CDN	
Cross-Site Request Forgeries	
Speculative Execution Attacks	✓
Window Control Attacks	

Take Action

	Content Security Policy	HttpOnly Cookies
Cross-Site Scripting	✓	✓
Compromised CDN		
Cross-Site Request Forgeries		
Speculative Execution Attacks	✓	✓
Window Control Attacks		

Take Action

	Content Security Policy	HttpOnly Cookies	Subresource Integrity
Cross-Site Scripting	✓	✓	
Compromised CDN			✓
Cross-Site Request Forgeries			
Speculative Execution Attacks	✓	✓	
Window Control Attacks			

Take Action

	Content Security Policy	HttpOnly Cookies	Subresource Integrity	SameSite Cookies
Cross-Site Scripting	✓	✓		
Compromised CDN			✓	
Cross-Site Request Forgeries				✓
Speculative Execution Attacks	✓	✓		✓
Window Control Attacks				

Take Action

	Content Security Policy	HttpOnly Cookies	Subresource Integrity	SameSite Cookies	Cross-Origin-Resource-Policy
Cross-Site Scripting	✓	✓			
Compromised CDN			✓		
Cross-Site Request Forgeries				✓	
Speculative Execution Attacks	✓	✓		✓	✓
Window Control Attacks					

Take Action

	Content Security Policy	HttpOnly Cookies	Subresource Integrity	SameSite Cookies	Cross-Origin-Resource-Policy	Cross-Origin-Window-Policy
Cross-Site Scripting	✓	✓				
Compromised CDN			✓			
Cross-Site Request Forgeries				✓		
Speculative Execution Attacks	✓	✓		✓	✓	
Window Control Attacks						✓

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/

More Information

https://developer.apple.com/wwdc2018/207

Safari, WebKit, and Password AutoFill Lab	Technology Lab 3	Wednesday 2:00PM
What’s New in Safari and WebKit	Executive Ballroom	Friday 2:00PM


WWDC18