#7 / Same-origin policy

The same-origin policy is a browser security feature
that restricts how documents and scripts on one origin
can interact with resources on another origin.

same-origin 정책은 출처를 가진 documents 나 script 가 다른 출처를 가진 리소스들과 어떻게 상호작용할지를 제한하는 브라우저 보안기능입니다.

A browser can load and display resources from multiple sites at once.

브라우저는 한 번에 여러 사이트의 리소스를 로드하고 표시할 수 있습니다.

You might have multiple tabs open at the same time,
or site could embed multiple iframes from different sites.

당신은 다양한 출처를 가진 iframes 를 embed 할 수 있는
여러 탭이나 사이트를 동시에 열었을 것입니다.

If there is no restriction on interactions between these resources, and a script is compromised by an attacker,
the script could expose everything in a user's browser.

만약 이러한 리소스들과의 상호작용을 제한하지 않고 공격자에 의해 script 가 손상된다면, script 는 사용자의 브라우저에 모든 것을 노출시킬 수 있습니다.

The same-origin policy prevents this from happening
by blocking read access to resources loaded from a different origin.

same-origin 정책은 서로 다른 출처로부터 로드된 리소스들의 읽기 접근을
차단함으로써 이러한 일이 일어나는 것을 예방합니다.

"But wait," you say, "I load images and scripts from other origins all the time."

아마 당신은 "잠시만요, 저는 매번 다른 출처로부터 이미지와 scripts 를
로드했는데요" 라고 말할지도 모릅니다.

Browsers allow a few tags to embed resources from different origin.

브라우저는 다른 출처로부터 온 리소스들을 몇개의 태그들을 허용함으로써
embed 할 수 있도록 합니다.

This policy is mostly a historical artifact
and can expose your site to vulnerabilities
such as clickjacking using iframes.

이 정책은 대체로 과거의 역사적 산물로
iframes 를 이용한 clickjacking 처럼
당신의 사이트를 보안 취약점에 노출시킬 수 있습니다.

You can restrict cross-origin reading of these tags using a Content Security Policy.

당신은 Content Security Policy 관련 태그들을 사용함으로써
cross-origin 의 reading 을 제한할 수 있습니다.

What's considered same-origin?

same-origin 은 무엇을 고려하는 것입니까?

An origin is defined by the scheme (also known as protocol, for example HTTP or HTTPS), port (if it is specified), and host.

origin 은 scheme (HTTP 나 HTTPS 같은 protocol), port (구체적으로 명시된 경우), 와 host 로 정의됩니다.

When all three are the same for two URLs,
they are considered same-origin.

두 개의 URL 에서 위의 3개가 모두 같다면,
그들은 same-origin 으로 간주됩니다.

For example,
`http://www.example.com/foo`
is the same origin as
`http://www.example.com/bar`
but not
`https://www.example.com/bar`
because the scheme is different.

예를 들어,

`http://www.example.com/foo` 과 `http://www.example.com/bar` 는
same-origin 이지만
`http://www.example.com/foo` 과 `http://www.example.com/bar` 는
cross-origin 입니다.

What is permitted and what is blocked?

무엇이 허용되고 또 차단됩니까?

Generally, embedding a cross-origin resource is permitted, while reading a cross-origin resource is blocked.

일반적으로, cross-origin 리소스를 embedding 하는 것은 허용하지만,
cross-origin 리소스를 reading 하는 것은 차단됩니다.

- iframes

  아이프레임

  - Cross-origin embedding is usually permitted
    (depending on the X-Frame-Options directive),
    but cross-origin reading (such as using JavaScript to access a document in an iframe) isn't.

    Cross-origin embedding 은 항상 허용하지만
    (X-Frame-Options directive 에 따라 달라질)
    cross-origin reading (JavaScript 를 사용해서
    iframe document 에 접근하는 것) 은 차단됩니다.

- CSS

  Cascading Style Sheet

  - Cross-origin CSS can be embedded using a <link> element or an @import in a CSS file.

    Cross-origin CSS 는 <link> 요소나 CSS file 의 @import 를 사용해서 embedded 될 수 있습니다

  - The correct Content-Type header may be required.

    적절한 Content-Type header 가 필요할 수 있습니다.

- forms

  폼

  - Cross-origin URLs can be used as the action attribute value of form elements.

    Cross-origin URL 들은 form 요소의 action 요소 값으로 사용될 수 있습니다.

  - A web application can write form data to a cross-origin destination.

    웹 어플리케이션은 cross-origin destination 으로 가는 form 데이터를 작성할 수 있습니다.

- images

  이미지

  - Embedding cross-origin images is permitted.

    cross-origin 이미지를 embedding 하는 것은 허용됩니다.

  - However, reading cross-origin images (such as loading a cross-origin image into a canvas element using JavaScript) is blocked.

    그러나, cross-origin 이미지를 읽는 것 (자바스크립트를 사용해서 canvas 요소에 cross-origin 이미지를 로드하는 것) 은 차단됩니다.

- multimedia

  멀티미디어

  - Cross-origin video and audio can be embedded using <video> and <audio> elements.

    Cross-origin 비디오와 오디오는 <video> 와 <audio> 요소를 사용해서 embedded 될 수 있습니다.

- script

  스크립트

  - Cross-origin scripts can be embedded;
    however, access to certain APIs (such as cross-origin fetch requests) might be blocked.

    Cross-origin scripts 는 embedded 될 수 있다;
    그러나 특정 API 들 (cross-origin fetch 요청같은) 은
    차단될 것입니다.

How to prevent Clickjacking

어떻게 clickjacking 을 막는가

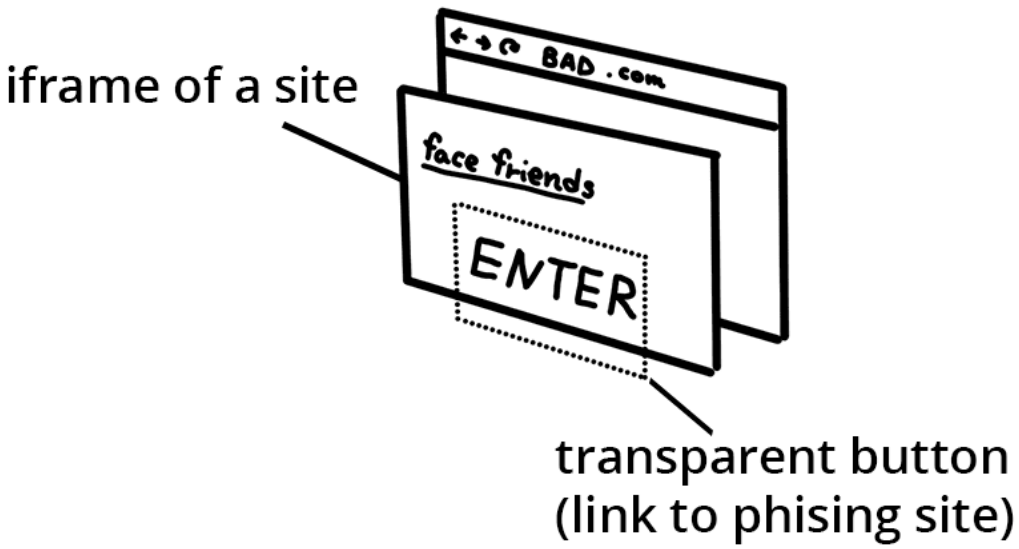

Figure: Clickjacking mechanism illustrated in 3 separate layers (base site, iframed site, transparent button)
Clickjacking 의 메커니즘이 3개의 분리된 층으로 그려져있다 (기본 사이트, iframed 사이트, 투명한 버튼)

An attack called "clickjacking" embeds a site in an `iframe` and overlays transparent buttons which link to a different destination.

"clickjacking" 이라고 불리우는 공격은 iframe 안에 한 사이트를 embed 하고, 다른 곳으로 링크를 거는 투명한 버튼을 그 위에 덧씌우는 것을 말합니다.

Users are tricked into thinking they are accessing your application while sending data to attackers.

사용자들은 공격자들에게 정보를 전송하면서 당신의 어플리케이션에 접속하고
있다고 속게 됩니다.

To block other sites from embedding your site in an iframe, add a content security policy with `frame-ancestors` directive to the HTTP headers.

당신의 사이트 iframe 에 다른 사이트가 embedding 되는 것을 막기 위해,
HTTP header 에 `frame-ancestors` 라는 directive 라는
컨텐츠 보안 정책을 추가합니다.

Alternatively, you can add `X-Frame-Options` to the HTTP headers see MDN for list of options.

또다른 방법으로는, X-Frame-Options 를 HTTP 헤더에 추가합니다.
(이 옵션의 설명은 MDN 을 참고하십시오.)

Wrap up

마무리

Hopefully you feel a little relieved that browsers work hard to be a gatekeeper of security on the web.

당신은 브라우저가 웹상의 보안을 지키는 든든한 문지기라는 점에
살짝 안도감을 느꼈을 것입니다.

Even though browsers try to be safe by blocking access to resources, sometimes you want to access cross-origin resources in your applications.

브라우저가 리소스에 대한 접근을 차단한다 할지라도,
때때로 당신은 당신의 어플리케이션에 cross-origin 리소스들을 접근하고 싶을
것입니다.

In the next guide, learn about Cross-Origin Resource Sharing (CORS) and how to tell the browser that loading of cross-origin resources is allowed from trusted sources.

다음 가이드에서는 Cross-Origin Resource Sharing (CORS) 에
대해서 배우며 브라우저가 어떻게 믿을 만한 출처에서 cross-origin
리소스들을 로드하는 것을 허용하는지 알려줄 것입니다.