



#6 / Browser sandbox

To defend against attacks,
a developer needs to mitigate vulnerabilities and
add security features to an application.

공격으로부터 방어하기 위해,
개발자는 보안 취약점을 완화하고 어플리케이션에 보안 기능을 추가해야 한다.

Luckily, on the web,
the browser provides many security features.

다행히도, 웹 상에서,
브라우저는 많은 보안 기능을 제공하고 있다.

Some are available for developers to opt-in,
and some are turned on by default to protect users.

어떤 것들은 개발자들을 위한 것으로 opt-in 방식으로 사용 가능하며,
또 다른 것들은 사용자를 지키기 위해 기본적으로 켜져 있습니다.

The idea of a "sandbox"

"sandbox" 아이디어

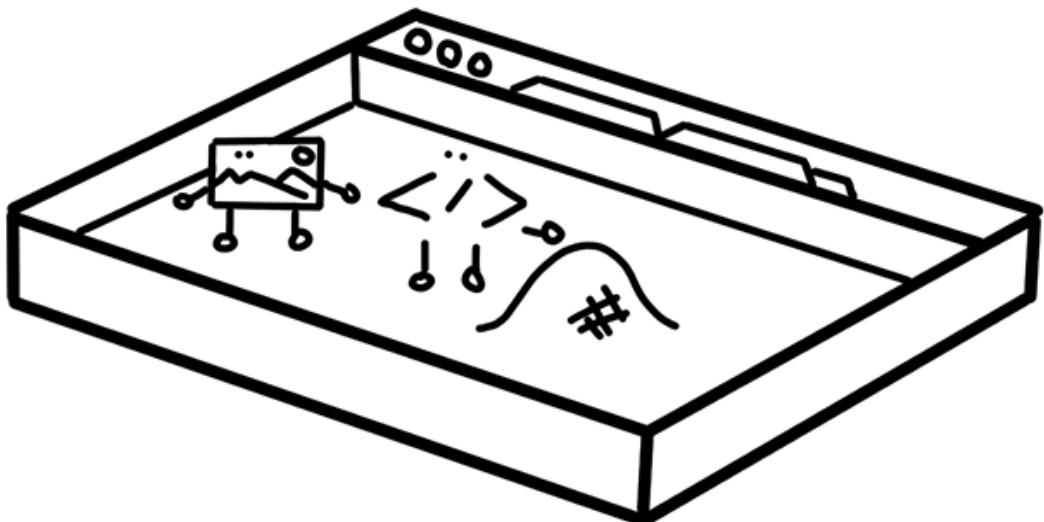


Figure: Browser as a sandbox
그림: sandbox 같은 브라우저

Modern web browsers are built on the idea of a "sandbox".

현대 웹 브라우저들은 "sandbox" 라는 아이디어를 바탕으로 만들어졌습니다.

A sandbox is a security mechanism used to run an application in a restricted environment.

sandbox 는 어플리케이션을 제한된 환경에서 실행하는데 사용되는 보안 메커니즘입니다.

Just like the physical sandbox at a playground where kids can create anything they want within the boundary without making a mess elsewhere, application code has the freedom to execute within a restricted environment.

아이들이 운동장의 모래사장에서 놀 때,
그곳 안에서만 그들이 원하는 모든 것을 만들지만,
밖의 다른 곳을 어지럽히지 않는 것처럼
어플리케이션 코드는 제한된 범위안에서만
자유롭게 실행될 수 있습니다.

For example,
JavaScript can add and modify elements on the page
but might be restricted from accessing an external JSON file.

예를 들어,
자바스크립트는 페이지의 요소들을 추가하거나 수정할 수 있지만,
외부의 JSON 파일에 접속하는 것은 제한될 것입니다.

This is because of a sandbox feature called same-origin.

이는 sandbox 특징이 same-origin 이라고 불리는 이유입니다.

Why is a sandbox necessary?

왜 sandbox 가 필요한가?

Every day, users of the web download arbitrary code and execute it on their computer or phone multiple times.

웹의 사용자들은 하루에도 몇 번씩 임의의 코드를 그들의 컴퓨터나 휴대폰에 다운받고 실행합니다.

If someone told you "Hey! Download and run this application!", you might pause to think if that application comes from a trusted source, read up on the application vendor, or check reviews carefully.

누군가 당신에게 "어이! 이 어플리케이션을 다운받고 실행해봐" 라고 말한다면, 당신은 아마 출처가 믿을만 한지, 어플리케이션 제공업체를 알아보거나, 리뷰를 주의깊게 읽기 위해 잠깐 멈추어 생각할 것입니다.

How about when someone sends you a URL saying "check out this blog post"?

누군가 URL 을 보내주며 "이 블로그 포스트를 확인해봐" 라고 말할 때는 어떨 것 같습니까?

You would probably click on it without asking questions like "What kind of JavaScript will this site download?"

당신은 아마 "이 사이트는 어떤 종류의 자바스크립트를 다운받을까?" 라는 의문 없이 그 사이트를 클릭할 것입니다.

The browser sandbox is the key feature that makes browsing on the web frictionless by making it safer to run arbitrary code.

브라우저 sandbox 는
임의의 코드 실행을 안전하게 만들어줌으로써,
웹 브라우징을 보다 부드럽게 만들어주는
핵심 기능입니다.

Make it secure by design

design 으로 안전하게 만들자

If the browser is sandboxing each web application,
should we even care about security?
Absolutely yes!

만약 브라우저가 각각의 웹 어플리케이션을 sandbox 화 하고 있더라도,
우리가 보안에 신경을 써야합니까?
당연합니다!

First of all, sandbox features are not the perfect shield.
무엇보다, sandbox 의 여러 특성들은 완벽한 방패가 아닙니다.

Even though browser engineers work hard,
browsers could have vulnerabilities and attackers are always trying to
bypass the sandbox
(such as with Spectre Attack).

아무리 브라우저 엔지니어들이 열심히 일을 해도,
브라우저들은 보안 취약점을 가질 수 있고 공격자들은 항상 sandbox 를
회피하려고 합니다 (Spectre 공격 처럼).

The sandbox could sometimes get in a way of creating a great web
experience.

sandbox 는 때때로 훌륭한 웹 경험을 만드는 데 있어
방해가 될 수 있습니다.

For example, a browser may block a fetch request to an image hosted on a
different domain.

예를 들어, 브라우저는 다른 도메인에 호스트된 이미지를 fetch 하는 요청을
차단할 수 있습니다.

You can share resources on different domains by turning on Cross-Origin Resource Sharing (CORS for short), but if it is not done carefully you can expose a resource to everyone else on the web, essentially undoing the sandbox.

당신은 Cross-Origin Resource Sharing (줄여서 CORS) 기능을 캔으로써, 다른 도메인의 리소스들을 공유할 수 있지만 당신은 필수적인 sandbox 를 하지 않음으로써, 웹의 모든 사람들에게 리소스를 노출시킬 수 있습니다.

Wrap up

마무리

A secure web experience can only be achieved if security is baked into the design of your application, and strong design starts with understanding existing features.

안전한 웹 경험은 오직 보안이 응용프로그램 설계에 적용될때 이루어질 수 있고, 보다 강력한 디자인은 기존에 존재하는 기능을 이해하면서 시작될 수 있습니다.

The next two guides dive into CORS and same-origin policy in depth.

다음의 두 가이드는 CORS 와 same-origin 정책을 심도있게 다룰 것입니다.