



## #2 / What are security attacks?

An insecure application could expose users and systems to various types of damage.

안전하지 않은 어플리케이션은  
사용자와 시스템을 다양한 유형의 위험에 노출시킬 수 있습니다.

When a malicious party uses vulnerabilities or lack of security features to their advantage to cause damage, it is called an **attack**.

악의적인 집단이 그들의 이익을 위해  
보안 취약점이나 보안 기능의 부족한 점을 이용해 피해를 주려는 것을  
**공격**이라고 부릅니다.

We'll take a look at different types of attacks in this guide so you know what to look for when securing your application.

이제 우리는 지금의 가이드를 통해 여러 유형의 공격을 살펴보면서,  
당신은 당신의 어플리케이션을 안전하게 하기 위해 무엇을 살펴봐야 하는지  
알게 될 것입니다.

---

### Active attacks vs passive attacks

적극적인 공격 vs 수동적인 공격

Attacks can be divided into two different types:  
active and passive

공격은 크게 두가지의 유형으로 나눌 수 있습니다:  
적극적인 것과 수동적인 것으로 말입니다

---

## Active attacks

### 적극적인 공격

With an **active attack**,  
the attacker tries to break into the application directly.

적극적인 공격에 있어,  
공격자는 어플리케이션에 직접적으로 침투하려고 합니다.

There are a variety of ways this could be done,  
from using a false identity to access sensitive data (masquerade attack)  
to flooding your server with massive amounts of traffic  
to make your application unresponsive  
(denial of service attack).

여기에는  
민감한 정보에 접근하기 위해 거짓 신원정보를 이용하거나  
(masquerade attack)  
당신의 어플리케이션이 반응하지 못하게 만들기 위해  
서버에 대량의 트래픽을 가득 채우는  
(denial of service attack)  
등의 다양한 방법이 존재합니다.

---

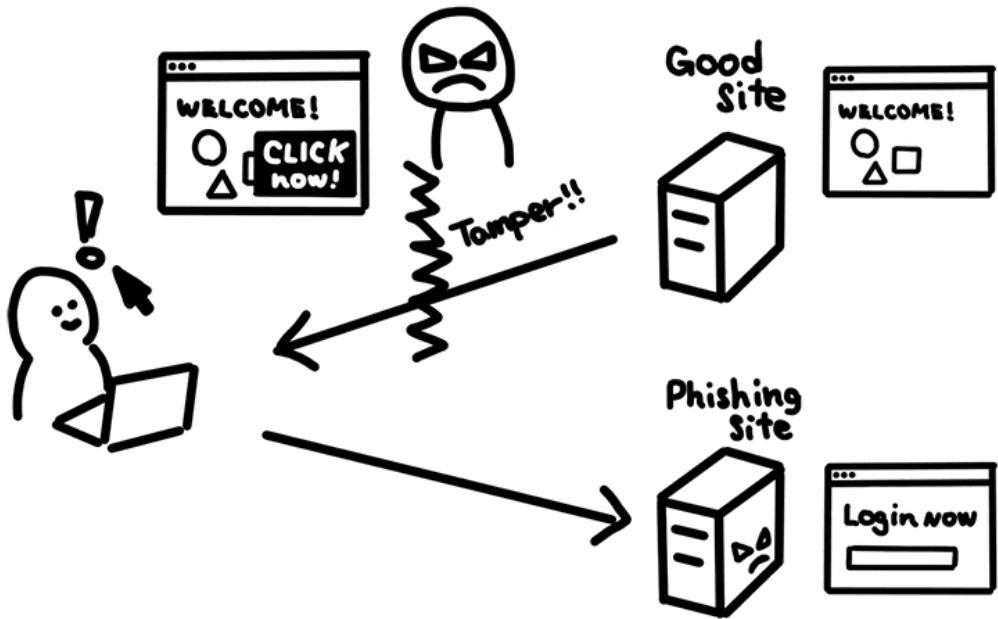
Active attacks can also be done to data in transit.  
적극적인 공격들은 전송중인 데이터에서도 일어날 수 있습니다.

An attacker could modify your application data  
before it gets to a user's browser,  
showing modified information on the site or direct the user to an  
unintended destination.

공격자는 사용자의 브라우저가 관련 정보를 얻기 전에,  
당신의 어플리케이션 정보를 수정하고  
그러한 수정된 정보를 사이트에 보여주거나 사용자를 의도하지 않은 목적지로  
유도할 수 있습니다.

This is sometimes called **modification of messages**.

이는 종종 **메세지 수정**이라고 불립니다.



A web site being tampered by attacker to guide user to a phishing site.  
사용자를 피싱 사이트로 유도하기 위해 공격자에 의해 함부로 손대진 사이트

Have you ever logged into free public wifi and  
seen ads wrapped around web pages you are accessing?

당신은 혹시 공공 와이파이에 로그인 하고 나서  
당신이 접근한 브라우저가 광고로 덮여있는 것을 보신 적이 있으십니까?

That's exactly what **modification of message** is!

이것이 바로 **메세지 수정**인 것입니다!

The wifi access point injected their advertising into a website  
before it got to your browser.

와이파이에 접속하는 포인트(?)가 당신의 브라우저가 정보를 얻기 전에  
그들의 광고를 사이트에 집어넣은 것입니다.

In many cases, you might dismiss it as "just ads for free wifi",  
but imagine if the same technique is used to replace some of the  
javascript or link to a phishing site.

대부분 당신은 "뭐 그냥 무료 와이파이 광고네" 하며 넘어갔겠지만,  
같은 기술이 자바스크립트나 링크를 피싱사이트로 가게끔 수정하는데  
사용된다고 상상해 보십시오.

Your site may be used by an attacker to misguide users without you noticing.

당신이 눈치채지 못한 사이 당신의 사이트는 사용자들을 그릇된 곳으로 이끄는데  
사용될 지 모릅니다.

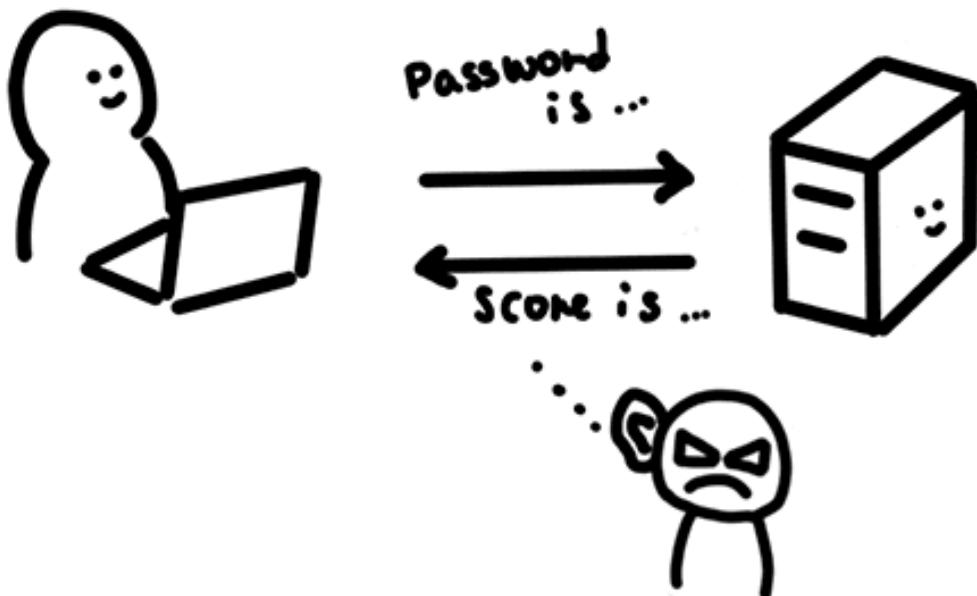
---

## Passive attack

### 수동적인 공격

With a **passive attack**, the attacker tries to collect or learn information from the application but does not affect the application itself.

**수동적인 공격**에 있어,  
공격자는 당신의 어플리케이션으로부터 정보를 수집하거나 배우지만  
어플리케이션 자체에 영향을 주지는 않습니다.



Attacker eavesdropping communication between a user and a server.  
사용자와 서버간의 통신을 도청하는 공격자

Imagine someone is eavesdropping on your conversation with friends and family,  
collecting information about your personal life,  
who your friends are, and where you hang out.

누군가가 당신이 가족과 친구들과 대화하는 것을 엿들으면서  
당신의 친구가 누구인지, 어디서 대화하고 있는지 등의 사생활 정보를  
수집하고 있다고 생각해보십시오.

The same thing could be done on your web traffic.

똑같은 일이 웹의 트래픽상에서 일어날 수 있습니다.

An attacker could capture data between the browser and the server  
collecting usernames & passwords, users' browsing history, and data  
exchanged.

공격자는 브라우저와 서버사이의 데이터에서  
사용자이름, 패스워드, 사이트 방문목록, 교환된 데이터를  
손에 넣을 수 있습니다.

---

## Defense against attacks

### 공격에 대항하는 방어

Attackers can directly harm your application or  
perform a malicious operation on your site  
without you or your users noticing it.

공격자는 당신 또는 당신의 사용자가 눈치채지 못한 사이  
당신의 어플리케이션을 직접적으로 손상시키거나  
당신의 사이트 위에서 악의적인 작업을 수행할 수 있습니다.

You need mechanisms to detect and protect against attacks.

당신은 이러한 공격을 탐지하고 보호할 수 있는 메커니즘이 필요합니다.

---

Unfortunately, there is no single solution  
to make your application 100% secure.

불행하게도, 당신의 어플리케이션을 100% 안전하게 만들어주는  
단 하나의 해결책은 존재하지 않습니다.

In practice,  
many security features and techniques are used in layers  
to prevent or further delay the attack.  
(this is called **defense in depth**.)

실제로 많은 보안 관련 기능과 기술은 여러 층위에서  
공격을 예방하거나 지연시키는게 쓰이고 있습니다.  
(이는 **다단계 방어**라고 불리웁니다.)

If your application contains a form,  
you might check inputs in the browser,  
then on the server, and finally at the database;  
you would also use HTTPS to secure the data in transit.

만약 당신의 어플리케이션이 form 을 포함하고 있다면,  
브라우저로부터의 input, 서버로의 input, 데이터베이스의 input 을  
모두 점검해야 할 것입니다;  
또한 전송중인 데이터를 안전하게 만들기 위해 HTTPS 를 사용해야 할 것입니다.

---

## Wrap up

### 마무리

Since many attacks can happen without ever hitting your server,  
it is sometimes hard to detect if attacks are happening or not.

많은 공격들이 당신의 서버를 건드리지 않고도 일어날 수 있기 때문에,  
때때로 공격이 일어나고 있는지 아닌지 알아내는 것도 매우 힘들 수 있습니다.

The good news is that web browsers have powerful security features already  
built in.

좋은 소식은 웹 브라우저가 이미 강력한 보안 기능들을 가지고 있다는 것입니다.

Follow the next topic "How browser mitigates against attacks" to learn  
more.

다음 주제인 "어떻게 브라우저가 공격을 다루는가" 에서  
이에대해 더욱더 자세히 알아보겠습니다.