Spring Security Lab

1. What is the attack called?
2. How is the attack performed?
3. How does Spring Security defend against this attack?
4. Are there any disadvantages or configuration considerations when defending against this attack? For example, maybe the security measures would make your site less user-friendly, or cause compatibility issues with different browsers?

<https://docs.spring.io/spring-security/site/docs/5.3.4.RELEASE/reference/html5/#exploits>

1. Clickjacking attack

Clickjacking is an attack that tricks a user into clicking a webpage element which is invisible or disguised as another element. This can cause users to unwittingly download malware, visit malicious web pages, provide credentials or sensitive information, transfer money, or purchase products online.

Typically, clickjacking is performed by displaying an invisible page or HTML element, inside an iframe, on top of the page the user sees. The user believes they are clicking the visible page but in fact they are clicking an invisible element in the additional page transposed on top of it.

A more modern approach to address clickjacking is to use [X-Frame-Options](https://developer.mozilla.org/en-US/docs/HTTP/X-Frame-Options) header

X-Frame-Options:DENY

The X-Frame-Options response header instructs the browser to prevent any site with this header in the response from being rendered within a frame. As with the other response headers, this is automatically included when the <headers> element is specified with no child elements

1. Enumeration attacks

**Enumeration technically means complete and ordered listing of all the items in a collection.**

**Most often enumeration attacks is username enumeration attack.**

There can be several ways to prevent a username enumeration attack. **Many of them we can achieve through simple tweaks in the features like user messages on a web application.**

Moreover, Spring Security over time has matured enough to support handling many of these attack vectors. There are features out-of-the-box and extension points to create custom safeguards. We'll explore some of these techniques.

Tweaking Messages

First, we must rule out all possibilities of inadvertently giving out more information than what is required. This would be difficult in registration but fairly simple in login and reset password pages. For instance, we can easily make the message for login page abstract. We can do similar tweaks to the message for the password reset page.

While tweaking the messages works well on some pages, there are pages like registration where it's tricky to do so. In such cases, we can use another tool called CAPTCHA. Now, at this point, it's worthwhile to note that any enumeration attack most likely is robotic due to a vast number of possibilities to go through. Hence, **detecting a human or robotic presence can help us prevent an attack.** CAPTCHA serves as a popular way to achieve this.