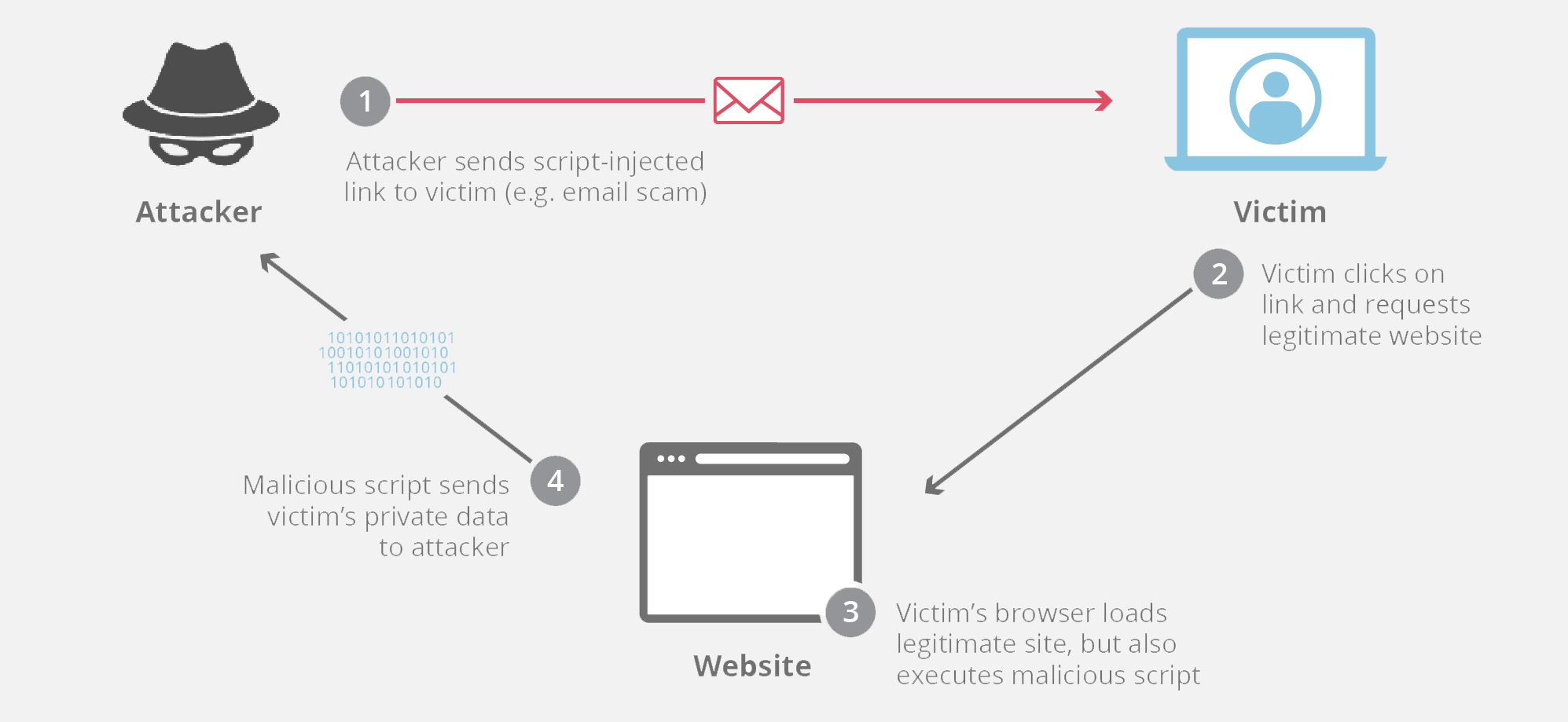
WHAT IS CROSS-SITE SCRIPTING?

Cross-site Scripting (XSS) refers to client-side code injection attack wherein an attacker can execute malicious scripts into a legitimate website or web application. By leveraging XSS, an attacker does not target a victim directly. Instead, an attacker would exploit a vulnerability within a website or web application that the victim would visit, essentially using the vulnerable website as a vehicle to deliver a malicious script to the victim’s browser.

HOW CROSS-SITE SCRIPTING WORKS?

In order to run the malicious application on the website which the victim visits he must find a way to inject payload on the webpage. He can go for social engineering techniques to convince a user to visit a vulnerable page with an injected JavaScript payload.

In order for an XSS attack to take place the vulnerable website needs to directly include user input in its pages. An attacker can then insert a string that will be used within the web page and treated as code by the victim’s browser.

Example of cross-site scripting:

The following server-side pseudo-code is used to display the most recent comment on a web page.

print "<html>"

print "<h1>Most recent comment</h1>"

print database.latestComment

print "</html>"

The above script is simply printing out the latest comment from a comments database and printing the contents out to an HTML page, assuming that the comment printed out only consists of text.

Attacker’s payload:

<html>

<h1>Most recent comment</h1>

<script>abc();</script>

</html>

Here abc() is some malicious function. The possible thing that can be done by this function is:

1. Cookies are often used to store session tokens, if an attacker can obtain a user’s session cookie, they can impersonate that user.
2. Make arbitrary modifications to the browser’s DOM
3. JavaScript can use XMLHttpRequest to send HTTP requests with arbitrary content to arbitrary destinations.