Web8: XSS Exploits - Part 2: Stored XSS

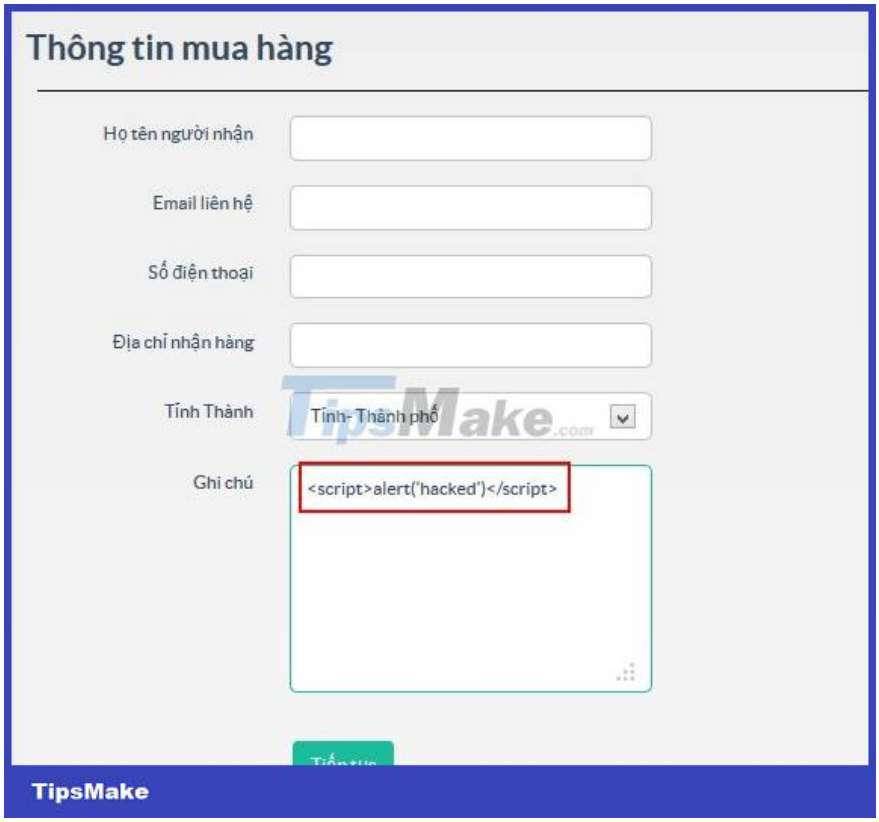
In this article, TipsMake.com invites you to learn about Stored XSS, another popular XSS mining method.

In the previous post, we learned about XSS (Cross Site Scripting) errors and the actual exploitation direction of XSS Reflected. Another type of XSS that is considered more dangerous is Stored XSS.

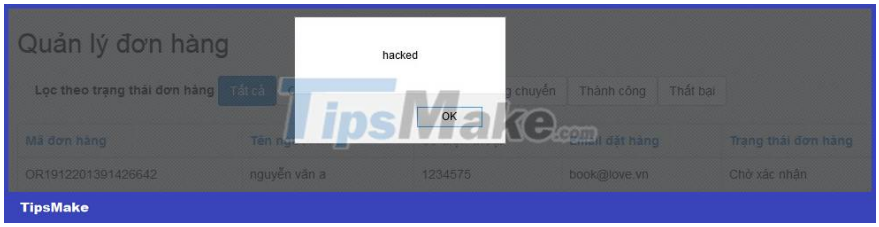
Unlike Reflected, which directly attacks some of the victims that hackers target, Stored XSS targets more victims. This error occurs when the web application does not thoroughly check the input data before saving it to the database (here I use this term to refer to the database, files or other areas where the application's data is stored). web).

With Stored XSS technique, hackers do not exploit directly, but must perform at least 2 steps.

First, hackers through the input points (form, input, text area.) do not filter carefully to insert dangerous code into the database.

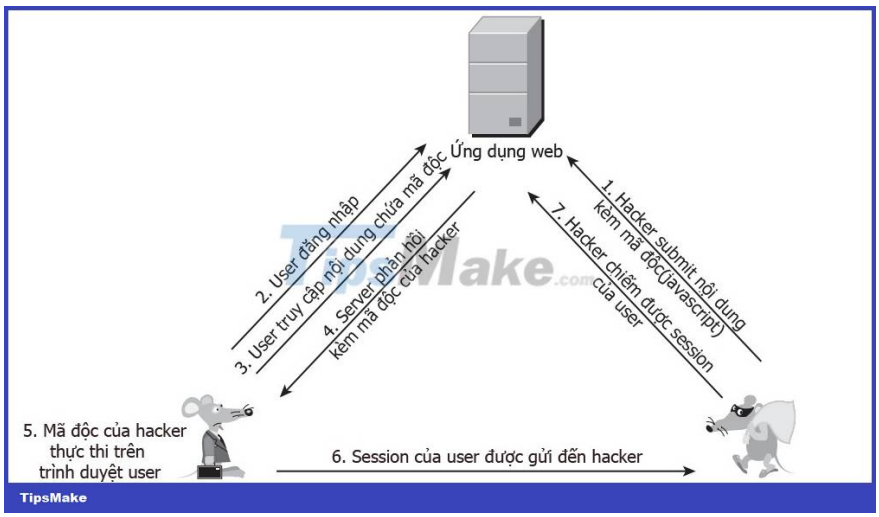


Next, When the user accesses the web application and performs operations related to this saved data, the hacker's code will be executed on the user's browser.



At this point, the hacker seems to have achieved his goal. For this reason, the Stored XSS technique is also known as second-order XSS.

The mining scenario is described as follows:



Reflected XSS and Stored XSS have two major differences in the attack process.

1. First, to exploit Reflected XSS, the hacker must trick the victim into accessing his URL. And Stored XSS does not need to do this, after inserting malicious code into the application's database, the hacker just waits for the victim to automatically access. For victims, this is completely normal because they do not know that the data they access has been infected.
2. Second, the hacker's goal is easier to achieve if the victim is still in the session of the web application at the time of the attack. With Reflected XSS, a hacker can convince or trick a victim into logging in and then accessing the URL he provides to execute malicious code. But Stored XSS is different, because the malicious code is stored in the Web database, whenever the user accesses the related functions, the malicious code will be executed, and most likely these functions require authentication. real(login) first so obviously the user is still in the session during this time.

From these things, it can be seen that Stored XSS is much more dangerous than Reflected XSS, the affected objects can be all but the users of that web application. And if the victim has an administrative role, there is also the risk of web hijacking.