Web9: XSS Exploits - Part 3: Dom Based XSS

In this next section, TipsMake.com invites you to learn about Dom Based XSS, another method of XSS mining.

We have mentioned about 2 types of XSS exploits, reflected and stored, they both have the common feature that dangerous pieces of code after being inserted will be executed after the server's response, which means that the error is on the side. server. There is another type of XSS exploit that goes against this feature, malicious code that is executed immediately upon client-side processing without going through the server, known as DOM Based XSS aka Type 0 XSS.

First of all we need to know what is DOM?

DOM stands for Document Object Model is a standard form of W3C (http://www.w3.org/DOM/) provided to access and manipulate data of structured documents such as HTML, XML. This model represents documents as a hierarchical tree structure. All elements in HTML, XML are considered as a node.

DOM Based XSS is a technique to exploit XSS based on changing the DOM structure of the document, specifically HTML.

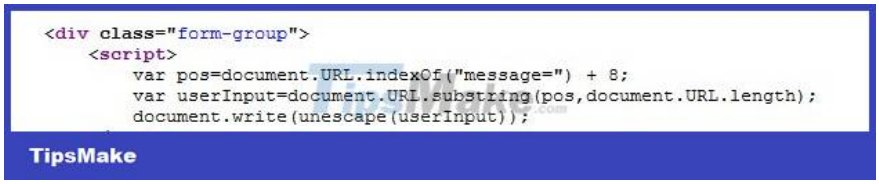
Let's take a look at a specific example:

A website has the following URL to the registration page:

```
example.com/register.php?message=Please fill in the form
```

When we access it, we see a very normal Form:

It is easy to deduce the message parameter passed to the message body on the form, look closely at the source code of this message:



The JavaScript snippet is responsible for taking the value from the message parameter and printing it out. From this lax input checking, it is possible to trick users into accessing dangerous URLs.

Instead of casting:

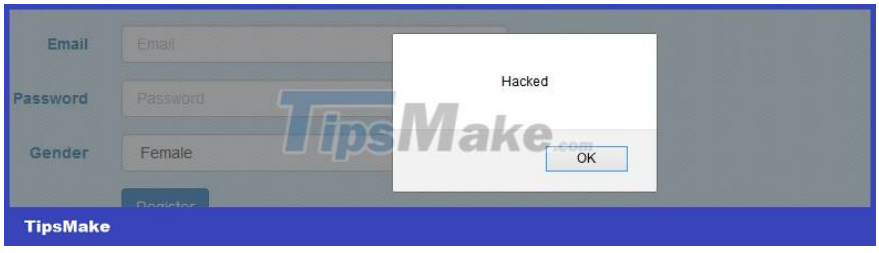
```
message=Please fill in the form
```

then transmit:

```
message=GenderMaleFemale function show(){alert();}
```

Then the registration form will become like this:

The user will have no doubts with a 'normal' form like this, and when selecting the gender, the Script will be executed:



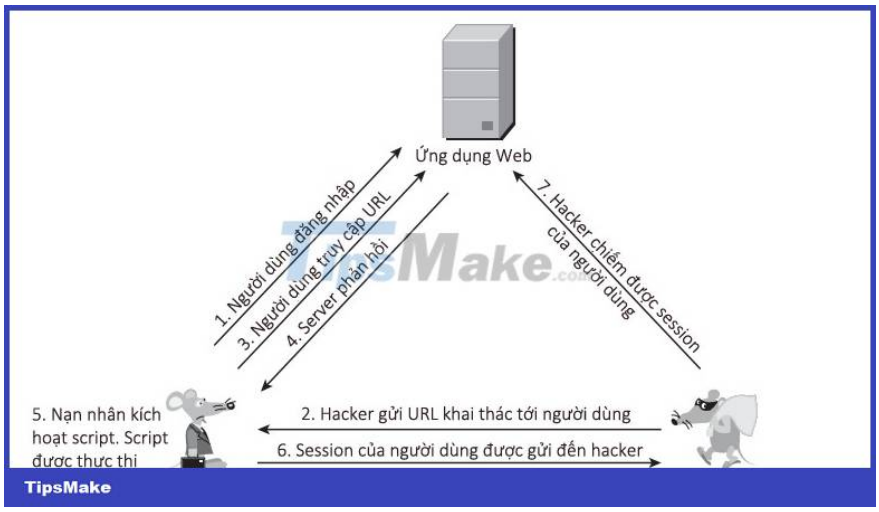
I'll explain a bit more about the value passed to the message parameter:

```
GenderMaleFemale function show(){alert();}
```

Its main purpose is to execute the show() function every time there is an onchage event on the select tag, the show() function here simply pops up a popup to show that the script has been executed. However, in practice, hackers will often use this show() function to execute a script that transfers the user cookie value to a predetermined server, readers can review the article Reflected XSS which mentions how hackers create How is this request?

This example gives us two important conclusions. First, the malicious code was executed as soon as the value in the select tag was clicked, ie executed on the client side without going through the server's response. Second, the HTML structure has been changed with the script passed in. And can also see the actual exploit scenario, DOM Based is somewhat similar to Reflected than Stored XSS when it has to trick users into accessing a URL that has embedded malicious code.

The following figure shows step-by-step implementation of the DOM Based XSS attack technique:



Good luck!

You finished reading the article "**Web9: XSS Exploits - Part 3: Dom Based XSS**" edited by the **TipsMake** team. We hope this article has provided you with many useful tech tips and tricks. You can search for similar articles on tips and guides. Thank you for reading and for following us regularly.