

Forum:[Hack Forums](#)

Thread: [I Need Guidance Hacking Website Dom XSS Vulnerability](#)

Redshadow - May 11, 2016, 7:02 pm

Hello Web Hackers

I am new to Website Dom XSS hacking, please dont be angry by my question. A website that is vulnerable to Dom XSS. What are the procedure to hack such website?

Thank you for reading, please leave a comment

MaXoNe - May 12, 2016, 1:21 am

add `#"><img src=x onerror=alert(0)>!><img src=x onerror=alert(9)>` to the url where vulnerability is located.

Redshadow - May 12, 2016, 9:58 am

I added `#"><img src=x onerror=alert(0)>!><img src=x onerror=alert(9)>` to the url where vulnerability is located and i got access denied. The website has web application firewall

thepentesters - May 13, 2016, 10:05 am

Perhaps look for different vulnerability? DOM XSS is very difficult to exploit and unless you plan on using it for a Social Engineering attack, quite worthless.

Gee4rce - May 13, 2016, 8:57 pm

(May 12, 2016 01:21 AM)MaXoNe Wrote: [\[CITE\]](#) add `#"><img src=x onerror=alert(0)>!><img src=x onerror=alert(9)>` to the url where vulnerability is located.

wow, never expected that you are still around here.. (for anyone who don't know MaXoNe; hes an XSS Legend and made one of best XSS tuts on HF: /showthread.php?tid=2070398)

Nice to see 'old faces' still around here :)

Well to get back to op's post;

(May 11, 2016 07:02 PM)Redshadow Wrote: [\[CITE\]](#) A website that is vulnerable to Dom XSS. What are the procedure to hack such website?

first you have to find out where your DOM XSS is located.

For DOM XSS there are many places like the URL, input forms on the website, http headers and cookies, etc.

Consider the case we have a website with a simple backbutton (for example).

When you click it, it calls the value from 'document.refer' and uses this value to provide the last link you have came from.

If you tamper with http packets, you will see that there is a field called "Referer".

This tells the Browser where you came from- now we change this to 'java.script:alert('hi')'. (without the dot after java, but otherwise HF changes it to "jalert").

Now when you click the back button, the browser fetches the url that you last visited, but since we manipulated that, the Browser will fetch value 'java.script:alert('hi')' as last value, and therefore the link you will be sent to is 'java.script:alert('hi')'.. which produces an alert box.

There are many other places, it depends on where the so called "html sinks" are and where it gets reflected on the page again.

I hope this example helped you to understand DOM XSS a little better.

Cheers, Gee4rce.

Redshadow - May 16, 2016, 2:34 am

(May 13, 2016 10:05 AM)thepentesters Wrote: [\[CITE\]](#) Perhaps look for different vulnerability? DOM XSS is very difficult to exploit and unless you plan on using it for a Social Engineering attack, quite worthless.

I think you are right, I am still unable to exploit the DOM XSS vulns till now. Thank you for commenting

Thanks for your explanations. I now fully understand sinks. I was a little bit confused about sinks before. I still cant exploit the DOM XSS website vulns. The site has firewall preventing my exploit. I keep getting "Access Denied" when I inject my xss script!

(May 13, 2016 08:57 PM)Gee4rce3 Wrote: [\[CITE\]](#) wow, never expected that you are still around here.. (for anyone who don't know MaXoNe; hes an XSS Legend and made one of best XSS tuts on HF: /showthread.php?tid=2070398)

Nice to see 'old faces' still around here :)

Well to get back to op's post;

first you have to find out where your DOM XSS is located.

For DOM XSS there are many places like the URL, input forms on the website, http headers and cookies, etc.

Consider the case we have a website with a simple backbutton (for example).

When you click it, it calls the value from 'document.refer' and uses this value to provide the last link you have came from.

If you tamper with http packets, you will see that there is a field called "Referer".

This tells the Browser where you came from- now we change this to 'java.script:alert('hi')'. (without the dot after java, but otherwise HF changes it to "jalert").

Now when you click the back button, the browser fetches the url that you last visited, but since we manipulated that, the Browser will fetch value 'java.script:alert('hi')' as last value, and therefore the link you will be sent to is 'java.script:alert('hi')'.. which produces an alert box.

There are many other places, it depends on where the so called "html sinks" are and where it gets reflected on the page again.

I hope this example helped you to understand DOM XSS a little better.

Cheers, Gee4rce.

MaXoNe - May 16, 2016, 5:11 am

Thanks man, I reactivated my account after 2 years of being away. HF hasn't changed a lot since

2014.

(May 13, 2016 08:57 PM)Gee4rce3 Wrote: [\[CITE\]](#) wow, never expected that you are still around here.. (for anyone who don't know MaXoNe; hes an XSS Legend and made one of best XSS tuts on HF: /showthread.php?tid=2070398)

Nice to see 'old faces' still around here :)

Well to get back to op's post;

first you have to find out where your DOM XSS is located.

For DOM XSS there are many places like the URL, input forms on the website, http headers and cookies, etc.

Consider the case we have a website with a simple backbutton (for example).

When you click it, it calls the value from 'document.refer' and uses this value to provide the last link you have came from.

If you tamper with http packets, you will see that there is a field called "Referer".

This tells the Browser where you came from- now we change this to 'java.script:alert('hi')'. (without the dot after java, but otherwise HF changes it to "jalert").

Now when you click the back button, the browser fetches the url that you last visited, but since we manipulated that, the Browser will fetch value 'java.script:alert('hi')' as last value, and therefore the link you will be sent to is 'java.script:alert('hi')'.. which produces an alert box.

There are many other places, it depends on where the so called "html sinks" are and where it gets reflected on the page again.

I hope this example helped you to understand DOM XSS a little better.

Cheers, Gee4rce.