Asfiya $ha!kh

Follow 434 Followers About

Open Redirects(Unvalidated Redirects and Forwards)

Asfiya $ha!kh Apr 27, 2019 · 4 min read

Let's understand how Open redirect vulnerability can be found, attacked and re-mediated.

'Open redirect' is an OWASP Top 10 2013 vulnerability which can occur when a website sends a visitor to another page either immediately or after a specified amount of time.

Ya… The bookish definition,with which we always start .

Lets understand it in a layman terminology, consider an outdated page that you believe your visitors have bookmarked. You don't want to lose the visitors/customers, so you just automatically redirect them to another page. These redirects and forwards if not setup properly, could pose risk to your online presence or business credibility.

Still not understood? … Okay … Lets take a practical example…

Consider below example in which aboutUs page is redirecting the user to its home page.

http://wwww.vulnerablesite.com/aboutUs.php?redirect=http://www.vulnerablesite.com

In this URL , an attacker could simply pass his own malicious URL as the destination URL and make it look like the same as of the real site asking user to re-enter their credentials.

Innocent user unknowingly submits the credentials, this is from where an attacker gets the innocent user's credentials.

Got afraid? …Hehehe… don't worry this won't happen with your banks website…They surely follow secure coding practices.

But wait… You should fear … there is another thing called phishing which can make you believe it is a legitimate website of your bank and attacker can steal your credentials.

Lets have some fun …by learning how to steal a Facebook user credential …

Hey! Make sure you are only watching this for educational purpose and not to attack your friends.

Check out this phishing attack video of stealing user's credentials by sending him a malicious link , If that link is made as the destination URL of the redirect and if link is made to look like real Facebook link using some technique such as IDN Homographic, then victim could make visit to that URL thinking its legitimate one and submit their credentials to attacker.

Video demonstrates a phishing attack using social engineering toolkit(setoolkit). setoolkit is an open-source Python-based tool aimed at penetration testing around Social-Engineering.

Wanna replicate the same? … Follow below steps to reproduce the attack -

1. *Open the terminal in Kali Linux and type* ***setoolkit***
2. *Once SET loads, type 1 to select Social-Engineering Attacks*
3. *Now it will show another list , select 2 Website Attack Vectors*
4. *Again it will show another list , select 3 Credential Harvester Attack Method*
5. *Again another list will pop-up, be patient and select 2 Site Cloner*
6. *To run attack on internal network, set your internal ip address or else to perform attack on WAN choose your external IP address*
7. *Enter the URL to clone — in this blog we have used https://www.facebook.comand press enter*
8. *Site will be hosted on the provided IP address, open the IP in browser, you will see a phished facebook page similar to real facebook page*
9. *Share your hosted IP in internal network and get people's credentials.*

This easy it was…!!

> ***How to check if any website is vulnerable to Open redirect?***

You'll need to check if any of websites pages redirect to a different destination. If they do, you'll need to determine if the addresses are included in the address bar, and if they can simply be changed as described above. If so, then website users could be subject to phishing attempts.

One way to check is through Burp suite, Spider the site to see if it generates any redirects (HTTP response codes 300–307, typically 302).

Check if any URL is taking the user input to redirect the user to destination URL, If so then mark that parameter for open redirect vulnerability tests

- *Check if basic redirection to malicious site works*

http://wwww.vulnerablesite.com/aboutUs.php?redirect=http://www.malicioussite.com

- *Check if URL encoding works*

http://wwww.vulnerablesite.com/aboutUs.php?redirect=http%3A%2F%2Fwww.malicioussite.com

- *Check if hex encoding works*

http://wwww.vulnerablesite.com/aboutUs.php?redirect=%68%74%74%70%3a%2f%2f%77%77%77%2e%6d%61%6c%69%63%69%6f%75%73%73%69%74%65%2e%63%6f%6d

- *Check if Double hex encoding works*

http://wwww.vulnerablesite.com/aboutUs.php?redirect=%25%36%38%25%37%34%25%37%34%25%37%30%25%33%61%25%32%66%25%32%66%25%37%37%25%37%37%25%37%37%25%32%65%25%36%64%25%36%31%25%36%63%25%36%39%25%36%33%25%36%39%25%36%66%25%37%35%25%37%33%25%37%33%25%36%39%25%37%34%25%36%35%25%32%65%25%36%33%25%36%66%25%36%64

- *Check if Base64 encoding works*

http://wwww.vulnerablesite.com/aboutUs.php?redirect=aHR0cDovL3d3dy5tYWxpY2lvdXNzaXRlLmNvbQ==

- *Sometimes redirection happens with destination URL as 1 parameter along with 1 more parameter called* ***hash****(MD5,SHA1,SHA256,SHA512 …), If both of these 2 parameters accounts to the same value then server allows redirection. For example — 33e042b4710653790ffc3403cd460394 is MD5 hashed valued of http://www.malicioussite.com*

http://wwww.vulnerablesite.com/aboutUs.php?redirect=http://www.malicioussite.com&hash=33e042b4710653790ffc3403cd460394

Check for all the possible hashes of http://www.malicioussite.com such as MD2, MD4, MD5, SHA1, SHA256, Whirlpool etc.

- *Redirection could happen with destination URL as 1 parameter along with 2 more parameters called **hash**(MD5,SHA1,SHA256,SHA512 …) and **salt**(Salt value could be anything, it could be a combination of characters, digits , alphanumeric , special character or anything we want). If all 3 of these parameters accounts to server white listed destination URL then only redirection happens. For example — Suppose 33e042b4710653790ffc3403cd460394 is MD5 hashed valued of http://www.malicioussite.com and salt is Hacker*

http://wwww.vulnerablesite.com/aboutUs.php?redirect=http://www.malicioussite.com&hash=33e042b4710653790ffc3403cd460394&salt=Hacker

For security reasons, Parameters such as hash , salt are not included in URL , but if developer forgets to hide them , It could be a good find if we could be able to redirect the user to malicious site by understanding developer's logic of redirection.

Its too much of testing right? Don't worry you have automation and burp suite fuzzing as your friends.

How to Prevent Open Redirect?

The easiest way is to simply not use redirects and forwards, If a redirect is necessary, do not trust user input for its destination.

- *Use relative URLs instead of absolute URLs*
- *Implement "meta refresh" in your page, which uses HTML to automatically redirect visitors to another page. To implement this, you just need to add the following code in the <head> section of your page's HTML:*

<META httpequiv="refresh" content="5;URL=http://vulnerablesite.com">

In this, the page will be redirected to "vulnerablesite.com" after 5 seconds.

- *Another option is to implement redirection to the destination URL within the source code.*

Java

```
response.sendRedirect("http://www.mysite.com");
```

PHP

```
<?php
 /* Redirect browser */
 header("Location: http://www.mysite.com/");
 ?>
```

ASP.NET

```
Response.Redirect("~/folder/Login.aspx")
```

Rails

```
redirect_to login_path
```

In the examples above, the URL is being explicitly declared in the code and cannot be manipulated by an attacker.

If none of the above is possible, force all redirects to a page where the user will have to click a button to confirm they are leaving the trusted site.

If you are still reading this, Then let me tell you , Its the end. See you next time..!!

> ***References:***

https://www.owasp.org/index.php/Unvalidated_Redirects_and_Forwards_Cheat_Sheet

http://www.montana.edu/itcenter/security/web/unvalidated-redirects-and-forwards.html

http://stackoverflow.com/questions/20371220/what-is-the-difference-between-response-sendredirect-and-request-getrequestdis

https://blog.detectify.com/2016/08/15/owasp-top-10-unvalidated-redirects-and-forwards-10/

https://www.youtube.com/watch?v=bHTglpgC5Qg&list=PLpNYlUeSK_rkrrBox-xvSkm5lgaDqKa0X&index=10

Cybersecurity Pentesting Penetration Testing Security Owasp