

Open Redirect

What is a Redirect?

A redirect happens when the website or web application changes the URL that is accessed in the client (usually external – internal redirects are usually called forwards). There are several ways to do this from the back-end. Usually, redirects are made by sending specific HTTP headers to the client but you can also create redirects, for example, using JavaScript code.

Status Codes

Status codes are issued by a server in response to a client's request made to the server.

The below list shows different kinds of status codes.

| 1XX Informational | | 4XX Client Error Continued | |
|-------------------|-------------------------------|----------------------------|------------------------------------|
| 100 | Continue | 409 | Conflict |
| 101 | Switching Protocols | 410 | Gone |
| 102 | Processing | 411 | Length Required |
| 2XX Success | | 412 | Precondition Failed |
| 200 | OK | 413 | Payload Too Large |
| 201 | Created | 414 | Request-URI Too Long |
| 202 | Accepted | 415 | Unsupported Media Type |
| 203 | Non-authoritative Information | 416 | Requested Range Not Satisfiable |
| 204 | No Content | 417 | Expectation Failed |
| 205 | Reset Content | 418 | I'm a teapot |
| 206 | Partial Content | 421 | Misdirected Request |
| 207 | Multi-Status | 422 | Unprocessable Entity |
| 208 | Already Reported | 423 | Locked |
| 226 | IM Used | 424 | Failed Dependency |
| 3XX Redirectional | | 426 | Upgrade Required |
| 300 | Multiple Choices | 428 | Precondition Required |
| 301 | Moved Permanently | 429 | Too Many Requests |
| 302 | Found | 431 | Request Header Fields Too Large |
| 303 | See Other | 444 | Connection Closed Without Response |
| 304 | Not Modified | 451 | Unavailable For Legal Reasons |
| 305 | Use Proxy | 499 | Client Closed Request |
| 307 | Temporary Redirect | 5XX Server Error | |
| 308 | Permanent Redirect | 500 | Internal Server Error |
| 4XX Client Error | | 501 | Not Implemented |
| 400 | Bad Request | 502 | Bad Gateway |
| 401 | Unauthorized | 503 | Service Unavailable |
| 402 | Payment Required | 504 | Gateway Timeout |
| 403 | Forbidden | 505 | HTTP Version Not Supported |
| 404 | Not Found | 506 | Variant Also Negotiates |
| 405 | Method Not Allowed | 507 | Insufficient Storage |
| 406 | Not Acceptable | 508 | Loop Detected |
| 407 | Proxy Authentication Required | 510 | Not Extended |
| 408 | Request Timeout | 511 | Network Authentication Required |
| | | 599 | Network Connect Timeout Error |

HTTP STATUS CODES

When a browser requests a service from a web server, an error may occur.
This is a list of HTTP status messages that might be returned.

The status code for Redirects are 301 Moved Permanently and 302 Found.

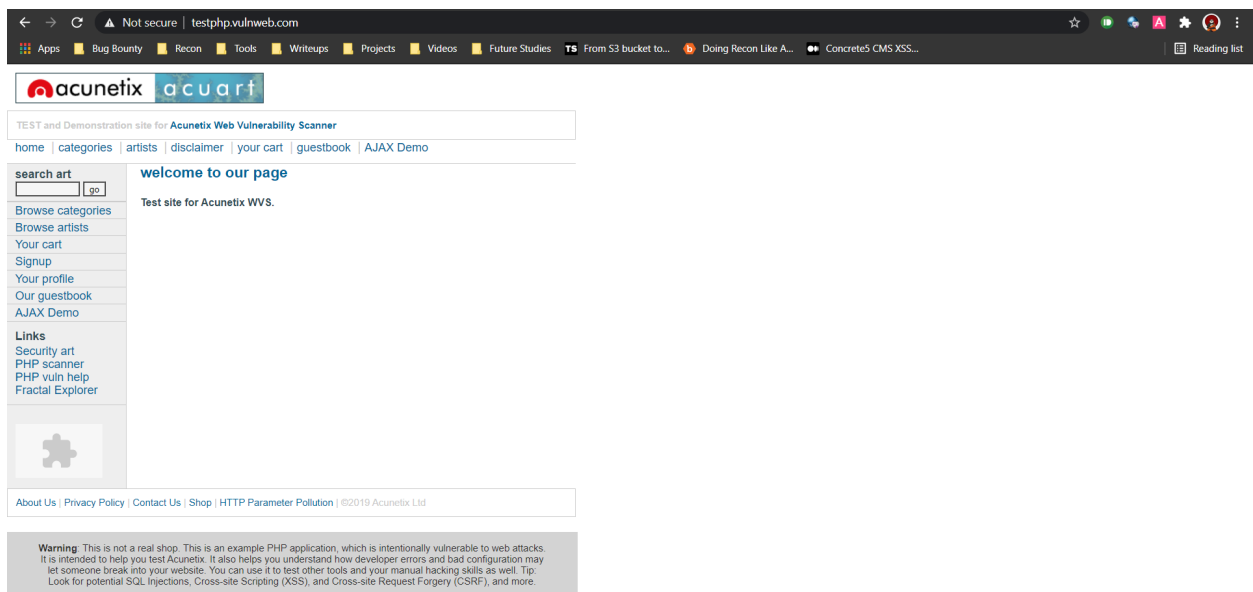
What is Open Redirect?

An open redirect vulnerability exists when the destination of the redirect is provided by the client and it is not filtered or validated. An attacker can construct a URL within the application that causes a redirection to an arbitrary external domain. This behavior can be leveraged to facilitate phishing attacks against users of the application. The ability to use an authentic application URL, targeting the correct domain and with a valid SSL certificate (if SSL is used), lends credibility to the phishing attack because many users, even if they verify these features, will not notice the subsequent redirection to a different domain.

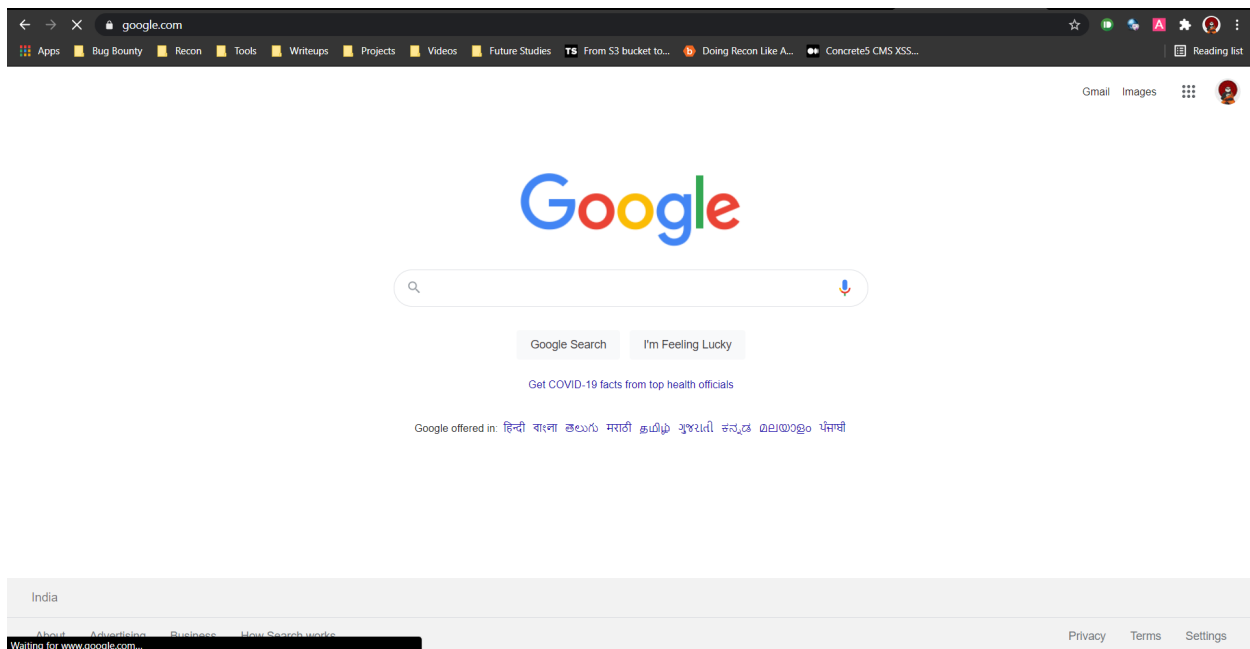
In short, if an attacker is able to redirect the user to a malicious website then it can be termed as Open Redirect.

Let's understand with an example

So over here I am onto a vulnerable website: <https://testphp.vulnweb.com>



The endpoint <https://testphp.vulnweb.com/redir.php?r=https://google.com> is vulnerable to Open Redirect. Notice the **parameter r**. This parameter does not sanitize the input and hence any attacker can use this to exploit the Open Redirect vulnerability and can redirect the attacker to any attacker controlled domain.



Exploiting Open Redirect Vulnerabilities

Open Redirect Vulnerabilities are simple to exploit. Always remember the Universal Truth of `parameters=values`. All you need to do is find vulnerable parameters. A list of different Open Redirect parameters can be found at:

<https://github.com/EdOverflow/bugbounty-cheatsheet/blob/master/cheatsheets/open-redirect.md>

Steps to exploit Open Redirect:

- Find parameters using Burp Suite Spider (or Crawl as called in newer version)
 - Different parameters include `?redirect=` | `?url=` | `?redirect_url=`
- Add the malicious website name to the vulnerable parameter and hit enter
- If redirected, BOOM! You found an Open Redirect vulnerability

Tips & Tricks

- Wayback Crawler: Wayback Crawler is a beautiful tool which finds out endpoints of the domain provided.

```
Installation: go get github.com/tomnomnom/waybackurls
Syntax to use: waybackurls <ip/domain>
```

```
To write in a file: waybackurls <ip/domain> | tee file.txt  
To grep target: cat file.txt | grep "param"
```

- gf patterns: GF is a wrapper around grep which help you grep for things you provide

```
Installation: go get -u github.com/tomnomnom/gf  
Usage with waybackurls: waybackurls <ip/domain> | grep redirect
```

- Always spider
- Collecting endpoints is very necessary
- Use those parameters which developers thinks that can be used for redirect
- Use grep "=" when grepping a target to find all `params = values`

Severity

The severity of Open Redirect Vulnerability can be categorized as P4 with a CVSS score of 3.3 which is Low.

Impact of Open Redirect Vulnerability

An attacker can use this vulnerability to redirect users to other malicious websites, which can be used for phishing and similar attacks

Prevention of Open Redirect Vulnerability

- Remove the redirection function from the application, and replace links to it with direct links to the relevant target URLs.
- Maintain a server-side list of all URLs that are permitted for redirection. Instead of passing the target URL as a parameter to the redirector, pass an index into this list.
- The application should use relative URLs in all of its redirects, and the redirection function should strictly validate that the URL received is a relative URL.
- The application should use URLs relative to the web root for all of its redirects, and the redirection function should validate that the URL received starts with a slash character. It should then prepend `http://yourdomainname.com` to the URL before issuing the redirect.
- The application should use absolute URLs for all of its redirects, and the redirection function should verify that the user-supplied URL begins with `http://yourdomainname.com/` before issuing the redirect.

References

- Open Redirect by PortSwigger : https://portswigger.net/kb/issues/00500100_open-redirection-reflected
- OWASP Open Redirect:
https://cheatsheetseries.owasp.org/cheatsheets/Unvalidated_Redirects_and_Forwards_Cheat_Sheet.html
- Open Redirect by Acunetix : <https://www.acunetix.com/blog/web-security-zone/what-are-open-redirects/>