

# Open Redirection



Allows an attacker to redirect users to a website of the attacker's choosing. Commonly, this is used as a part of a phishing attack.

## Regular use cases:

- \* Get a user back to a page after logging in.
- \* Pass single-sign-on information to a third party.

`https://vulnsite.com?next=/home`

Once a user requests this URL, the server will take the value of the "next" parameter to determine where to redirect the user (in this case, to the home page).

## HTTP/1.1 302 Found

Date: Wed, 16 Jun 2011 22:10:48 GMT

Location: <https://vulnsite.com/home>

Connection: Close

Content-Type: ~~text~~/html

Content-Length: 90

<html> <body> You are being

<a href = 'https://vulnsite.com/home'>

REDIRECTED

</a> </body>

</html>

An HTTP response. Notice the "Location" header.

Now imagine that an attacker forms and submits this URL:

<https://vulnsite.com?next=www.attacker.com>

If the server does not validate the value of the "next" parameter, then the HTTP response will be the following:

## HTTP/1.1 Found

Date: Wed, 16 Jun 2011 22:10 GMT

Location: <https://www.attacker.com>

Connection: Close

Content-Type: text/html

Content Length: 85

```
<html><body> You are being  
<a href='https://www.attacker.com'>  
  REDIRECTED  
</a></body>  
</html>
```

An attacker-controlled redirection.

The user will be seamlessly redirected to the attacker's web site.

### Plan of Attack :

1. Attacker creates a malicious URL for the victim to click and sends it to them.

<https://vulnsite.com?next=https://www.attacker.com>

2. The user clicks the link and is redirected to a malicious website.

Dangers: Attacker can abuse the trust of one site to get users to visit an evil site.

Remediation: Ensure that user-submitted URLs match an allowed value before redirecting.

Warn users that they are being redirected off-site to reduce the chance of a successful phishing attack.