Bug Report: Open Redirect

Victor.g.
me@gmail.com

June 18, 2025

Overview

Issue: Open redirection vulnerabilities arise when an application incorporates user-controllable data into the target of a redirection in an unsafe way. 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.
Severity: High
Tested on: Anonymized (https://victim.org)
Date of Discovery: June 19th, 2025

Vulnerability Description

0.1 Issue detail:

The value of the Referer HTTP header is used to perform an HTTP redirect. The payload https://example.com?https://victim.org was submitted in the Referer HTTP header. This caused a redirection to the following URL: https://example.com?https://victim.org

Steps to Reproduce

1. open the url `https://victim.org` in your browser.
2. prepare a list of testing domains `https://victim.org`).
3. send two diff requests to the endpoints with carying referers in the following formats:

   ```
   Referer: https://example.org?https://victim.org
   Referer: https://google.com?https://victim.org
   ```

4. The response from the target varies with the given altered domains

Expected vs Actual Behavior

- If it is considered unavoidable for the redirection function to receive user-controllable input and incorporate this into the redirection target, one of the following measures should be used to minimize the risk of redirection attacks:

- 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 your domain name
  The application should use absolute URLs for all of its redirects, and the redirection function should verify that the user-supplied URL begins with The application's domain name.

Impact

- 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.

Recommendations

If possible, applications should avoid incorporating user-controllable data into redirection targets. In many cases, this behavior can be avoided in two ways:

-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.

0.2 Resources

portswigger.net https://portswigger.net/web-security/access-control

Proof of Concept

A working HTML/JS PoC file is included in the link `https://imgur.com/a/SMoDxJG`.

Disclaimer

This proof-of-concept is created solely for educational and research purposes by an independent security researcher. No unauthorized access, disruption, or harm was caused to any systems or data. All testing was reported and bug patched. This report aims to help improve security by responsibly disclosing potential vulnerabilities. Please do not misuse this information