OAuth 2.0 Redirection Attacks

Rifaat Shekh-Yusef

IETF113, OAuth WG, Vienna, Austria

March 21st ,2022

Background

- Proofpoint published an article late last year that describes some implementation issues that lead to redirection attacks
  - https://www.proofpoint.com/us/blog/cloud-security/microsoft-and-github-oauth-implementation-vulnerabilities-lead-redirection

Attacker Setup

- Creates an account on the victim’s platform.
- Creates an application on that platform.
- Crafts an authorization request that meant to direct the user to the above application.
- Sends the request to the victim through SMS, email, etc.

Typical Authorization Request

GET /authorize?

response_type=code&

redirect_uri=webapp.com/callback&

scope=email profile&

client_id=<client_id>

Host: **login.as.com**

OAuth 2.0 Error Handling

RFC6749 - 4.1.2.1. Error Response

If the request fails due to a **missing**, **invalid**, or **mismatching redirection URI**, or if the **client identifier** is **missing** or **invalid**, the authorization server SHOULD **inform the resource owner** of the error and **MUST NOT** automatically **redirect** the user-agent to the invalid redirection URI.

If the resource owner **denies** the access request or if the request **fails** for reasons other than a missing or invalid redirection URI, the **authorization server informs the client** by adding the following parameters to the query component of the **redirection URI** using the "application/x-www-form-urlencoded" format, per Appendix B:

Issue 1 – Invalid Response Type or Scope

After the user is authenticated, if the request includes **invalid** values in the **response type** or **scope** parameters, the user will be **automatically redirected** to the **attacker's webapp**.

GET /authorize?

response_type=invalid value&

scope=invalid value&

client_id=attacker_client_id&

state=state&

redirect_uri=attacker.com/callback

Host: login.as.com

Issue 2 - Decline Consent

After the user is authenticated, and a consent page is displayed to the user, if the user **declines the consent**, the user will be **redirected** to the **attacker's application**.

GET /authorize?

response_type=code&

client_id=attacker_client_id&

redirect_uri=attacker.com/callback

Host: login.as.com

Issue 3 - Redirection before Authentication

The following example shows a request missing a number of parameters:

GET /authorize?

client_id=attacker_client_id

Host: login.as.com

As a result, the user is **automatically redirected** to the **attacker's client** and the user does not get a chance to do anything about it.

Issue 4 - Silent Authentication

OIDC support silent authentication with **prompt=none**, which could be used to check for existing authentication or consent.

In this case, the user will not be prompted to authenticate or consent and will be automatically redirected to the attacker's application.

GET /authorize?

response_type=code&

scope=openid profile email&

client_id=attacker_client_id&

state=<state>&

redirect_uri=attacker.com/callback&

prompt=none

Host: login.as.com

Rethinking Error Handling?

- Should the **Authorization Server** always be responsible for error handling?
- Should there be an explicit text about always authenticating the user before error handling?
- How to handle **silent authentication** (prompt=none)?
- How about when the user **declines** the consent?

- We should at least capture these issues
  - Security BCP?
  - A new dedicated document?