Facebook Private Photos Hack – How I Exposed Your Private Photos

Laxman Muthiyah

Security Researcher

May 27, 2021

Abstract- This paper is about how I found a vulnerability on Facebook that allowed me to view private photos of Facebook users that are uploaded using Sync feature. Facebook mitigated the issue and rewarded me $10,000 USD as a part of their bug bounty program.

Index Terms- Facebook vulnerability, Bug Bounty Program, Private Photos, Syncing feature

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. FACEBOOK MOBILE SYNCING

Facebook has a feature called photo syncing, that uploads all our mobile photos to Facebook server for backup purpose and later it can be published if the user intends to. Sync photos feature is turned on by default in some mobile phones. We can control it in the app settings. Most of us are unaware of this feature.

I was curious about the endpoint that is handling these private photos. To check the endpoint, we need to use MITM proxy like burp suite or Charles proxy to intercept the requests sent from Facebook mobile app to server.

After the MITM setup, I got to know that “vaultimages” endpoint of Facebook Graph API is handling these synced photos. I started exploring through the endpoint. Reading the synced photos through this endpoint got caught my eyes. After few minutes of testing, I realized that “vaultimages” endpoint is vulnerable.

Facebook mobile application makes a GET request to https://graph.facebook.com/me/vaultimages with a top-level access token to read the synced photos. Facebook server check the request for a proper access token and serve the synced photos of the respective user as the response.

Request :-

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

access_token=<top_level_access_token_here>
```

Response :-

```
{“photos”:{“photos_id_and_uri”}}
```

The vulnerable part is, it just checks the owner of the access token and not the application which is making the request. So, it allows any third party application with user_photos permission to read your mobile photos. There are large numbers of Facebook applications which use user_photos permission to read user's public photos.

A malicious app that you are using can hack all of your private photos in few seconds. I know that most of us won't see the list of permissions while using any application.

III. PROOF OF CONCEPT

So, to view the private photos using third party apps, the following HTTP GET request has to be sent to graph.facebook.com along with third party app's access token.

Request :-

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

access_token=<third_party_access_token >
```

Response :-

```
{"photos":{"photos_id_and_uri"}}
```

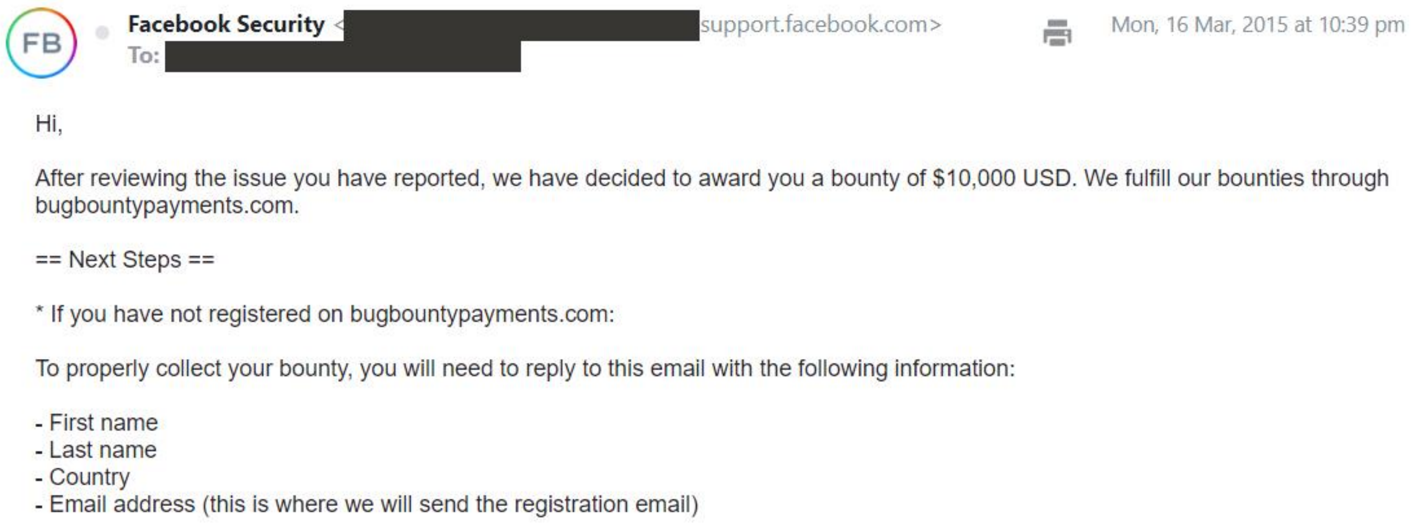
As you can see above, the response is successful and the private photos of the target user will be listed.

Reported this vulnerability to Facebook Security Team, as usual, they were very fast in addressing this issue. They pushed a fix in less than 30 minutes after the acknowledgement of report.

They just whitelisted their official mobile applications in that endpoint and no other applications can access your private photos anymore. This vulnerability is completely patched and vault images cannot be accessed by any application except the whitelisted applications.

At some point, Facebook removed the sync photos feature from their mobile apps. Therefore, Facebook no longer uploads our mobile photos to their server.

After the patch, Facebook rewarded me $10,000 for responsible disclosing this vulnerability as a part of their bug bounty program.



Facebook Security <[redacted]support.facebook.com>

Mon, 16 Mar, 2015 at 10:39 pm

To: [redacted]

Hi,

After reviewing the issue you have reported, we have decided to award you a bounty of $10,000 USD. We fulfill our bounties through bugbountypayments.com.

== Next Steps ==

* If you have not registered on bugbountypayments.com:

To properly collect your bounty, you will need to reply to this email with the following information:

- First name
- Last name
- Country
- Email address (this is where we will send the registration email)

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] Pocatilu, Paul, Catalin Boja, and Cristian Ciurea. "Syncing mobile applications with cloud storage services." Informatica Economica 17.2 (2013): 96.

[4] 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.