A Facebook GraphQL crash course

@PHWD · FRIDAY, DECEMBER 16, 2016

Endpoint: https://graph.facebook.com/graphql

Who can use it?
Any Facebook User

Which apps use it?
Any Facebook native iOS/Android application
https://www.facebook.com/search/me/apps-used/str/Facebook/apps-named/intersect

How to get access_tokens to access it?

- Search Facebook for embedded tokens
- Follow the OAuth flow and simulate for mobile
- Forget the access_token and let Facebook do the work
- Disassemble mobile app and rebuild access_token generation request by hand.
- Add a proxy to your mobile device and listen to requests to graph.facebook.com via Fiddler, Burp Suite, etc.

1st Method: **Easiest**

Go to www.facebook.com/me using a desktop browser, view source and search for the string `access_token`. This will be the Android (App ID: 350685531728) access token.
(credit: @JosipFranjkovic)

2nd Method: **Easy**

Find valid redirect URIs for Facebook

Example: Instagram

https://www.facebook.com/v2.5/dialog/oauth?response_type=token&display=popup&client_id=124024574287414&redirect_uri=fb124024574287414%3A%2F%2Fauthorize

3rd Method: **Limited**

Look for calls in messenger.com and facebook.com that look like `/api/graphqlbatch/` and replay them changing the queries field.

4th Method: **Annoying**

Uses legacy method for OAuth api.facebook.com/restserver.php?method=auth.login.

Needs an API application secret from a Facebook native application. It's easier to just disassemble Android vs iOS when looking for this.

```
apktool d com.facebook.katana
```

```
vi
/Users/phwd/Facebook/APKS/com.facebook.katana/smali/com/facebook/katana/app/FacebookAppTypes.smali
```

```
15     move-result-object v2
16
17     const-string v3, "API_KEY"
18
19     const-string v4, "API_SECRET"
```

Watch for suspicious activity attempts, this might force you through the Facebook checkpoint roadblock and make you reset your password.

5th Method: **Hard (Maybe or maybe not)**

Facebook uses custom certificate pinning. Jailbreak device and bypass custom pinning.
iOS
https://github.com/nabla-c0d3/ssl-kill-switch2/issues/13,
https://github.com/phwd/OneForAllFacebook
Android
https://github.com/pouyadarabi/Facebook_SSL_Pinning/,
https://www.facebook.com/groups/349225725474262/permalink/416218442108323/

Deauthorization

There are cases where the token might not initially want to be revoked. Use wisely. I'm not responsible for any changes to your account
(Use test accounts, you should be using test accounts)

```
HTTP POST /auth/create_session_for_app
new_app_id=350685531728
access_token=FROM_METHOD_1
```

Take the new generated token and use that in the deauth flow
https://developers.facebook.com/docs/facebook-login/permissions/requesting-and-revoking#revokelogin

```
HTTP DELETE /me/permissions
access_token=TOKEN_FROM_ABOVE
```

How to use it?

Graph API Explorer already provides an easy interface for 3rd party Graph API requests, use that and an access_token from above.

developers.facebook.com/tools/explorer

Queries

User

```
graphql?q=node(13608786)
```

Page

```
graphql?q=node(113702895386410)
```

Multiple Objects

```
graphql?q=nodes(13608786, 113702895386410)
```

- Most IDs from Facebook can be used here.
- Some might need Base64 encoding `echo –n “yourstring” | base64`
- Most types of lists follow the format: `nodes(id)`
  e.g. `nodes(13608786){friends{nodes{id,name}}}`

Mutation

```
POST /graphql
```

```
q=Mutation MutationName : SomePayloadResponse {mutator_name(<input>){page{id},
client_mutation_id}}
```

```
query_params={"input":"
{\"actor_id\":\"13608786\",\"page_id\":\"113702895386410\",\"is_enabled\":\"1\",\"c
lient_mutation_id\":\"1\"}"}
```

- `<input>` takes parameters from `query_params`
- `mutator_name` is the name of the mutator call you are using e.g. `page_like`
- `query_params` , the variables for your call, take care of inner quotes `\"`
- every mutation *“should”* need a `client_mutation_id` and `actor_id`
- `actor_id` is the viewer (example, did you use a user access token or page access token) check by `graphql?q=viewer()` or `graphql?q=me()`
- other parameters are based on Mutation Type e.g. `"is_enabled\":\"1\"`
- `(<input>){page{id}, client_mutation_id}}` Use information here to see the result of your mutation. The fields returned are based on Mutation type. e.g. `page{id}`

How to find fields?

Rely on schema (http://graphql.org/learn/schema/) and hunt for calls through Facebook. Use Schema and Queries for GraphQL to make inferences about types, calls and fields.

- `strings /Users/phwd/Facebook/IPAS/Facebook\ 67.0/Payload/ Facebook.app/GraphQLQueries.data > ~/Desktop/GraphQLQueries.txt`
- Other areas within iOS, `mainbundle.js`, `modelMetaData.bin`, `schemaMetaData.bin`
- Areas in facebook.com, messenger.com, e.g. Look for facebook.com/api/graphqlbatch/ messenger.com/api/graphqlbatch/ calls