Blog

If you want to find vulnerabilities and make a money bug bounty hunting you may want to get a copy of my book. BUY NOW!

API Hacking GraphQL

Slack Group

Before we get started I have started a slack group dedicated to hacking. We welcome everyone from beginner to advanced to join. I will be on everyday answer questions, doing CTFs, and talking about cool hacks. If you enjoy hacking and are looking for like minded people join below:

NEW Hacking Group Slack Channel

Introduction

GraphQL is a data query language developed by Facebook and was released in 2015. GraphQL acts as an alternative to REST API. Rest APIs require the client to send multiple requests to different endpoints on the API to query data from the backend database. With graphQL you only need to send one request to query the backend. This is a lot simpler because you don't have to send multiple requests to the API, a single request can be used to gather all the necessary information.

GraphQL

As new technologies emerge so will new vulnerabilities. By default graphQL does not implement authentication, this is put on the developer to implement. This means by default graphQL allows anyone to query it, any sensitive information will be available to attackers unauthenticated.

When performing your directory brute force attacks make sure to add the following paths to check for graphQL instances.

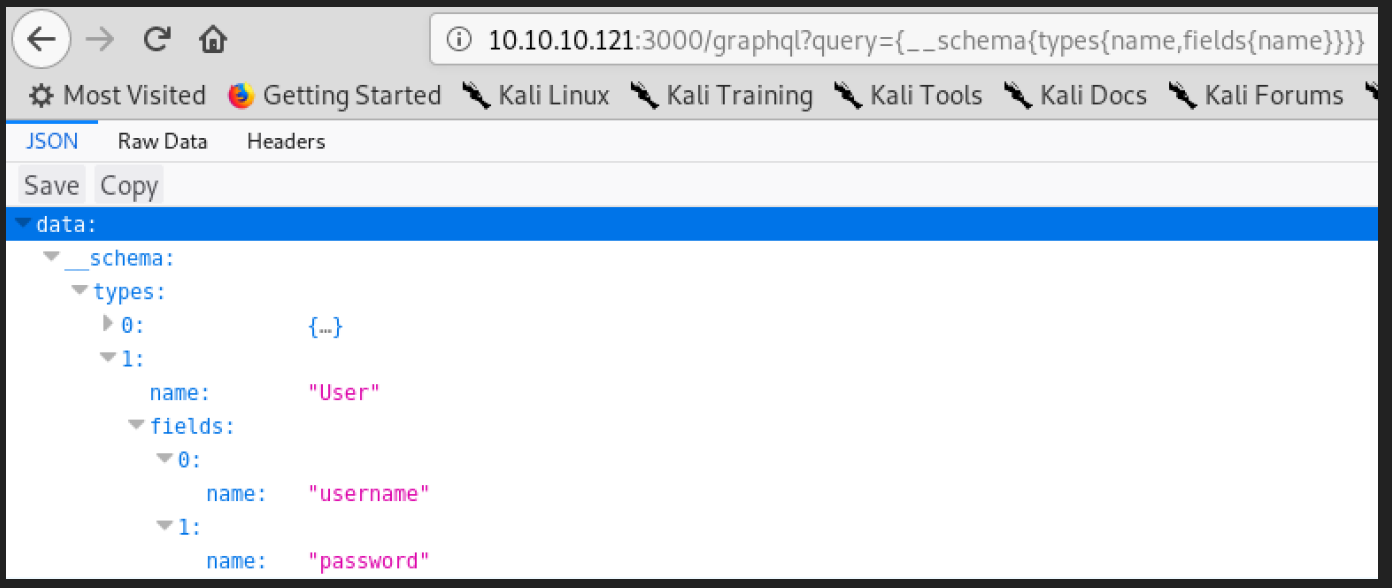
- */graphql*
- */graphiql*
- */graphql.php*
- */graphql/console*

Once you find an open graphQL instance you need to know what queries it supports. This can be done by using the introspection system, more details can be found here:**GraphQL: A query language for APIs.**

*It's often useful to ask a GraphQL schema for information about what queries it supports. GraphQL allows us to do so...*graphql.org

Issuing the following requests will show you all the queries that are available on the endpoint.

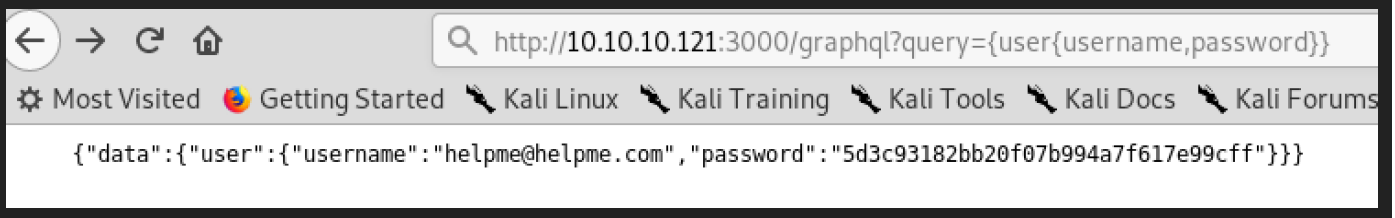
example.com/graphql?query={__schema{types{name,fields{name}}}}



As you can see there is a type called *"User"* and it have two fields called *"username"* and *"password"*. Types that start with a "__" can be ignored as those are part of the introspection system.

Once an interesting type is found you can query its field values by issuing the following query.

example.com/graphql?query={TYPE_1{FIELD_1,FIELD_2}}



Once the query is submitted it will pull the relevant information and return the results to you. In this case we get a set of credentials that can be used to login to the application.

GraphQL is a relatively new technology that is starting to gain some traction among startups and large corporations. Other than missing authentication by default graphQL endpoints can be vulnerable to other bugs such as IDOR.

Conclusion

GraphQL is still new and is not widely implemented but that does not mean you wont find it in the wild. Startups and fortune 500 companies tend to stay on the bleeding edge of technology so you will probably see those companies implement graphQL first. I personally think graphQL is an awesome alternative to REST APIs but you have to make sure to implement it correctly. If you find a graphQL endpoint make sure to test if they implemented authentication. If they didn't you just got an easy win!

← Linkedin OSINT

XML External Entity(XXE) -

Post Comment

TWITTER

SLACK CHANNEL

GITHUB