Idor Checklist by Rhonny Sharma

IDOR: Attack vectors, exploitation, bypasses and chains

Tips: Don't blindly test for changing numbers till you get PII, tools can do this for you. Dive deep into applications, find hidden functionality and features, and know how your application works most of all to succeed in finding IDOR's.

*Finding IDOR Attack Vectors Ideas:**

1. What do they use for authorization?(JWT, API Keys, cookies, tokens) Tip: Find this out by replacing high privaledge authorization with lower privaledge authorization and seeing what the server responds with

2. Understand how they use ID's, hashes, and their API. Do this by looking at the API Documentations if they have one.

3.**Recon for IDOR's:**

- Try to search engine scrape for UUIDs, ex: google dork for IDOR URL parameters

- Use burp extension autorize + autorepeater

- Using tools like WaybackURLS or gau, and grep for UUID's, ids and common IDOR URL parameters

- Scraping JS files for API endpoints with UUID's, common IDOR parameters

4. **Note**: Recon for IDORs is very hard, many of them are found

manually using logic and they depend on each application highly. To add onto this, IDOR's are commonly found on API endpoints with JSON parameters, not URL. However, IDOR recon is still good to possibly find some low-hanging fruit.

5. **Every time you see a new API endpoint that receives an object ID from the client, ask yourself the following questions:**

- Does the ID belong to a private resource? (e.g /api/user/123/news vs /api/user/123/transaction)

- What are the IDs that belong to me?

- What are the different possible roles in the API?(For example — user, driver, supervisor, manager)

6. **Bypassing Object Level Authorization:**

- Add parameters onto the endpoints for example, if there was

```html

GET /api_v1/messages --> 401

vs

GET /api_v1/messages?user_id=victim_uuid --> 200

```

7. **Why Does this Work?**: Sometimes the applications authorization settings are not set to directory level, or the whole URL, making it so that adding URL parameters can bypass previous restrictions to certain endpoints.

8. **Note**: To find URL parameters for endpoints, use tools like

[Arjun](https://github.com/s0md3v/Arjun) to bruteforce common IDOR URL parameter names.

-9. HTTP Parameter pollution

```html
GET /api_v1/messages?user_id=VICTIM_ID --> 401 Unauthorized

GET /api_v1/messages?user_id=ATTACKER_ID&user_id=VICTIM_ID --> 200 OK

GET /api_v1/messages?user_id=YOUR_USER_ID[]&user_id=ANOTHER_USERS_ID[]
```

10. - Add .json to the endpoint, if it is built in Ruby!

```html
/user_data/2341 --> 401 Unauthorized

/user_data/2341.json --> 200 OK
```

11. ```- Test on outdated API Versions

```html
/v3/users_data/1234 --> 403 Forbidden

/v1/users_data/1234 --> 200 OK
```

12. Wrap the ID with an array.

```html
```

{“id”:111} --> 401 Unauthriozied

{“id”:[111]} --> 200 OK

```

13. Wrap the ID with a JSON object:

```html

{“id”:111} --> 401 Unauthriozied

{“id”:{“id”:111}} --> 200 OK

```

14. JSON Parameter Pollution:

```html

POST /api/get_profile

Content-Type: application/json

{“user_id”:<legit_id>,”user_id”:<victim’s_id>}

15. MFLAC:

```html

GET /admin/profile --> 401 Unauthorized

GET /ADMIN/profile --> 200 OK

```

16. Path Traversal:

```html
```

POST /users/delete/VICTIM_ID --> 403 Forbidden

POST /users/delete/MY_ID/../VICTIM_ID --> 200 OK

```

17. **Random Tips/Tricks:**

- Try to send a wildcard(*) instead of an ID. It's rare, but sometimes it works.

- Check through the same corresponding mobile API endpoints for the webapp, to find uuids, if you need them to complete the IDOR exploitation

- Many times there will be endpoints to translate emails into GUID's, check for those

- If it is a number id, be sure to test through a large amount of numbers, instead of just guessing Ex: Burp intruder from ID 100-1000

- If endpoint has a name like /api/users/myinfo, check for /api/admins/myinfo

- Replace request method with GET/POST/PUT

- If none of these work, get creative and ask around!

18. **Escalating/Chaining with IDOR's Ideas:**

1. Lets say you find a low impact IDOR, like changing someone elses name, chain that with XSS and you have stored XSS!

2. If you find IDOR on and endpoint, but it requires UUID, chain with info disclosure endpoints that leak UUID, and bypass this!

3. If none of these work, get creative and ask around!
```