Checklist

API Security

▼ API Security

  ▼ 31 Days of API Security Testing

- □ To change versions: `api/v3/login` → `api/v1/login`
- □ Check other AuthN endpoints: `/api/mobile/login` → `/api/v3/login` `/api/magic_link`
- □ Verb Tampering: `GET /api/trips/1` → `POST /api/trips/1` `POST /api/trips` `DELETE /api/trips/1`
- □ Try Object IDs in HTTP headers and bodies, URLs tend to be less vulnerable.
- □ Try Numeric IDs when facing a GUID/UUID: `GET /api/users/6b95d962-df38` → `GET /api/users/1`
- □ Wrap ID with an array: `{"id":111}` → `{"id":[111]}`
- □ Wrap ID with a JSON object: `{"id":111}` → `{"id":{"id":111}}`
- □ HTTP Parameter Pollution: `/api/profile?user_id=legit&user_id=victim` `/api/profile?user_id=victim&user_id=legit`
- □ JSON Parameter Pollution: `{"user_id":legit,"user_id":victim}` `{"user_id":victim,"user_id":legit}`
- □ Wildcard instead of ID: `/api/users/1` → `/api/users/*` `/api/users/%` `/api/users/_` `/api/users/.`
- □ Ruby application HTTP parameter containing a URL → Pipe as the first character and then a shell command.
- □ Developer APIs differs with mobile and web APIs. Test them separately.
- □ Change Content-Type to `application/xml` and see if the API parse it.
- □ Non-Production environments tend to be less secure (staging/qa/etc.) Leverage this fact to bypass AuthZ, AuthN, rate limiting & input validation.
- □ Export Injection if you see `Convert to PDF` feature.
- □ Expand your attack surface and test old versions of APKs IPAs.

  ▼ Misc

- Google Dorks

```
site:target.tld inurl:api
site:target.tld intitle:"index of" "api.yaml"
site:target.tld inurl:/application.wadl
site:target.tld ext:wsdl inurl:/%24metadata
site:target.tld ext:wadl
site:target.tld ext:wsdl
```

```
user filetype:wadl
user filetype:wsdl
```

- Check different `Content-Types`

```
x-www-form-urlencoded --> user=test
application/json --> {"user": "test"}
application/xml --> <user>test</user>
```

- If it's regular POST data try sending arrays, dictionaries

```
username[]=John
username[$neq]=lalala
```

- If JSON is supported try to send unexpected data types

```
{"username": "John"}
{"username": true}
{"username": null}
{"username": 1}
{"username": [true]}
{"username": ["John", true]}
{"username": {"$neq": "lalala"}}
```

- If XML is supported, check for XXE

Gathered By: HolyBugx