Analyzing OpenAPI Specifications for Security Design Issues

Carmen Cheh and Binbin Chen

#IEEESecDev  https://secdev.ieee.org/2021
Motivation

T-Mobile Alerts 2.3 Million Customers of Data Breach Tied to Leaky API

(Threatpost, 2018)

Don’t Put It on the Internet: Tesla Backup Gateway Edition

Nov 17, 2020  |  13 min read | Derek Abdine
(Rapid7, 2020)

I Scraped Millions of Venmo Payments. Your Data Is at Risk

Opinion: Venmo makes sending and receiving money a social affair. But those emoji-laden payment descriptions leave you exposed to cyberattacks.

(Wired, 2019)
Related Work

State-of-the-Practice Tools

- APIsecurity.io
- ZAP
- appspider

State-of-the-Art Research

- Focused on discovering design flaws
- Atlidakis et al. [1] (2020)
  - Four security rules concerning design and operations on data objects
  - Vulnerability can result in privilege escalation or sensitive data exposure
  - Stateful API fuzzer with property checker
  - Dynamic analyses check HTTP status code and response body
  - Static analyses finds patterns in specification (e.g., keywords)

Our Contribution: Security Analyses Approach

**OpenAPI Graph Model**

- **IDENTIFY SENSITIVE FIELDS**
  - Shortlist high-frequency fields: Here are fields familiar to you. Which are sensitive or public info?
  - Expand list of sensitive/non-sensitive fields
  - Identify insecure & high-risk API call: X, Y, and Z are sensitive.
  - Calculate exposure level: Here are the exposure levels for X, Y, and Z – and B, C, and D.

- **INFER SECURITY DESIGN ISSUES**
  - Identify insecure & high-risk API call: API calls B, C, and D are potentially insecure.
Case Study: Open Bank Project (OBP)

- Global standard and open source API solution for open banking

- Statistics:
  - 304 API calls
  - 142 resources
  - 345 schemas
  - 402 data fields

- Available documents: [Retrieved May 1, 2021]
  - OpenAPI specification
  - Glossary – Richer details about API calls

We use this as a source of ground truth
Sample API call Specification:
GET /obp/v4.0.0/banks/{BANK_ID}/accounts

'/obp/v4.0.0/banks/{BANK_ID}/accounts':
get:
security:
  - directLogin: []
description: 'Returns the list of accounts at BANK_ID that the user has access to. Authentication is Mandatory.'
parameters:
  - in: path
    name: BANK_ID
    required: true
type: string
responses:
  '200':
schema:
    $ref: '#/definitions/BasicAccountsJSON'
  '400':
schema:
    $ref: '#/definitions/ErrorBankNotFound'

BasicAccountsJSON:
required:
  - accounts
properties:
  accounts:
    type: array
    items:
      $ref: '#/definitions/BasicAccountJSON'

BasicAccountJSON:
required:
  - id
  - label
  - bank_id
  - views_available
properties:
  id:
    type: string
  label:
    type: string
  bank_id:
    type: string
  views_available:
    type: array
    items:
      $ref: '#/definitions/BasicViewJson'

Sample HTTP 200 Response

```json
{
  "accounts": [
    {
      "id": "8cde9f0",
      "label": "NoneLabel",
      "bank_id": "GENODEM1GLS",
      "views_available": [
        {
          "id": "1",
          "short_name": "HHH",
          "is_public": true
        }
      ]
    }
  ]
}
```
OBP: Full OpenAPI Graph Model
Identify Sensitive Fields: OBP

**Identify Sensitive Fields**

- Shortlist high-frequency fields
  - Here are fields familiar to you. Which are sensitive or public info?
  - X and Y are sensitive.
  - We think Z and A are also sensitive. Is this true?
  - We agree Z is sensitive.

- Expand list of sensitive/non-sensitive fields
  - X, Y, and Z are sensitive.

**Infer Security Design Issues**

- Identify insecure & high-risk API call
  - API calls B, C, and D are potentially insecure.
  - X, Y, and Z are sensitive.

- Calculate exposure level
  - Here are the exposure levels for X, Y, and Z – and B, C, and D.
Identify Sensitive Fields: Results

How many resources use this data field?

<table>
<thead>
<tr>
<th>Field</th>
<th>Num of Resources</th>
</tr>
</thead>
<tbody>
<tr>
<td>bank_id</td>
<td>66</td>
</tr>
<tr>
<td>jsonString</td>
<td>47</td>
</tr>
<tr>
<td>date</td>
<td>43</td>
</tr>
<tr>
<td>currency</td>
<td>39</td>
</tr>
<tr>
<td>amount</td>
<td>39</td>
</tr>
<tr>
<td>email</td>
<td>33</td>
</tr>
<tr>
<td>type</td>
<td>32</td>
</tr>
<tr>
<td>provider</td>
<td>31</td>
</tr>
<tr>
<td>customer_id</td>
<td>27</td>
</tr>
<tr>
<td>legal_name</td>
<td>26</td>
</tr>
<tr>
<td>date_of_birth</td>
<td>26</td>
</tr>
<tr>
<td>mobile_phone_number</td>
<td>26</td>
</tr>
</tbody>
</table>
Identify Sensitive Fields: Results

How close is this data field to the sensitive data field?

\[
distance(amount, dependants) = 1 - \frac{19}{42} = 0.55
\]

\[
distance(amount, bank\_code) = 1 - \frac{7}{42} = 0.83
\]

Threshold

25th percentile of all \( distance(amount, \cdot) \) 

0.805
Identify Sensitive Fields: Results

- PUT /banks/{BANK_ID}/customers/{CUSTOMER_ID}/data
- GET /banks/{BANK_ID}/accounts/{ACCOUNT_ID}/transaction-requests

How close is this data field to the sensitive data field?

<table>
<thead>
<tr>
<th>customer_id</th>
<th>legal_name</th>
<th>last_ok_date</th>
<th>rating</th>
<th>kyc_status</th>
</tr>
</thead>
<tbody>
<tr>
<td>account_id</td>
<td>mobile_phone_number</td>
<td>challenge_type</td>
<td>bank_id</td>
<td>username</td>
</tr>
<tr>
<td>amount</td>
<td>branch_id</td>
<td>customer_number</td>
<td>type</td>
<td>provider_id</td>
</tr>
<tr>
<td>currency</td>
<td>transaction_ids</td>
<td>user_id</td>
<td>dob_of_dependants</td>
<td>role_name</td>
</tr>
<tr>
<td>provider</td>
<td>created</td>
<td>relationship_status</td>
<td>highest_education_attained</td>
<td>entitlement_id</td>
</tr>
<tr>
<td>date_of_birth</td>
<td>date</td>
<td>dependants</td>
<td>employment_status</td>
<td>password</td>
</tr>
</tbody>
</table>

19 sensitive fields

- amount
- bank_code

42 high-frequency fields

- 7 shortlisted fields

List of sensitive and non-sensitive fields

Manually identified sensitive fields based on glossary

32 non-sensitive fields
Infer Security Issues: OBP

**IDENTIFY SENSITIVE FIELDS**

Shortlist high-frequency fields

Here are fields familiar to you. Which are sensitive or public info?

Expand list of sensitive/non-sensitive fields

X and Y are sensitive.

We think Z and A are also sensitive. Is this true?

We agree Z is sensitive.

**INFER SECURITY DESIGN ISSUES**

Identify insecure & high-risk API call

X, Y, and Z are sensitive.

API calls B, C, and D are potentially insecure.

Calculate exposure level

Here are the exposure levels for X, Y, and Z – and B, C, and D.
Infer Security Issues: Results

What unauthenticated API calls return sensitive data?

GET /obp/v4.0.0/customers/CUSTOMER_ID/kyc_documents
Returns a list of documents that affirm the identity of the customer (e.g., passport, driving licence).

GET /obp/v4.0.0/banks/{BANK_ID}/balances
Get the balances for the accounts of the current user at a bank.

What API calls return both sensitive data and non-sensitive data?

34 API calls found
Infer Security Issues: Results

How exposed is this data field and API call to the public?

*Iteration 0*

- PUT /data
  - bank_code
  - amount

Iteration 1

- PUT /data
  - bank_code
  - amount

Iteration 2

- PUT /data
  - bank_code
  - amount

INFER SECURITY DESIGN ISSUES

- Identify insecure & high-risk API call
- Calculate exposure level

Here are the exposure levels for X, Y, and Z – and B, C, and D.
Infer Security Issues: Results

How exposed is this data field to the public?

<table>
<thead>
<tr>
<th>Field</th>
<th>Exposure</th>
</tr>
</thead>
<tbody>
<tr>
<td>customer_id</td>
<td>37.3</td>
</tr>
<tr>
<td>account_id</td>
<td>32.3</td>
</tr>
<tr>
<td>amount</td>
<td>27.7</td>
</tr>
<tr>
<td>currency</td>
<td>27.7</td>
</tr>
<tr>
<td>provider</td>
<td>25.8</td>
</tr>
<tr>
<td>date_of_birth</td>
<td>13.2</td>
</tr>
<tr>
<td>legal_name</td>
<td>13.2</td>
</tr>
<tr>
<td>mobile_phone_number</td>
<td>13.2</td>
</tr>
<tr>
<td>transaction_ids</td>
<td>9.01</td>
</tr>
<tr>
<td>customer_number</td>
<td>4.64</td>
</tr>
<tr>
<td>user_id</td>
<td>4.33</td>
</tr>
<tr>
<td>relationship_status</td>
<td>4.25</td>
</tr>
<tr>
<td>dependants</td>
<td>4.25</td>
</tr>
<tr>
<td>rating</td>
<td>4.25</td>
</tr>
<tr>
<td>dob_of_dependants</td>
<td>4.25</td>
</tr>
<tr>
<td>highest_education_attained</td>
<td>4.25</td>
</tr>
<tr>
<td>employment_status</td>
<td>4.25</td>
</tr>
<tr>
<td>username</td>
<td>4.02</td>
</tr>
</tbody>
</table>

How exposed is this API call to the public?

- PUT /management/banks/BANK_ID/cards/CARD_ID
- POST /management/banks/BANK_ID/cards
- GET /management/banks/BANK_ID/cards
- GET /banks/BANK_ID/accounts/ACCOUNT_ID/VIEW_ID/account
- GET /banks/BANK_ID/accounts/ACCOUNT_ID/permissions
Conclusion and Future Work

• Analyzing OpenAPI specifications using the relationships between API calls and data fields allow us to:
  • Identify sensitive data
  • Highlight potentially insecure or high-risk API calls

• **Future research area:** Identifying additional security design issues
  • **Potential solutions:** Looking at other components in OpenAPI specifications (e.g., data field type)