UiO-CTF

API Security

$ who

```
MaritIrenRognliTokle :0     2019-09-25 17:15 (:0)

      UiO-CTF Team Captain
      Leader TG:Hack
      Mobile Consultant Sopra Steria
      Mobile, web and pwn

NoraTomas :1                2019-09-25 17:15 (:1)

      UiO-CTF Team Captain
      n00b chief TG:Hack
      Consultant at Inmeta
```

$ agenda

> UiO-CTF intro

> What is an API?

> What is an HTTP request?

> Ways of sending HTTP requests

> API security

Practical information - What do we do?

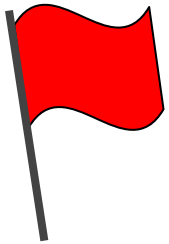
- **Practical workshops:** Going through tasks from hacking competitions
- **Presentations:** Security related issues
- **CTF nights:** Pizza and hacking.
- **CTF participation:** Online competitions

Practical information - Our goal

- Teach security
- **make people aware of the importance of security!**
- Get people interested in CTFs (hacking competitions) - Compete with us!

Practical information - Why join?

- Awesome and very useful knowledge about programming, hacking and security
- More attractive to employers(!)
- It’s super fun and addictive!



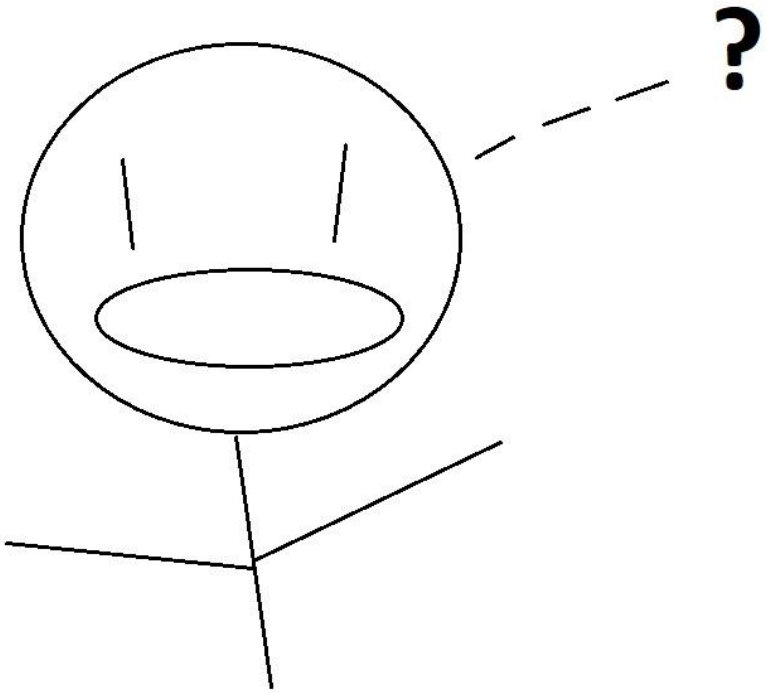
We would argue that most CTFers usually has a higher level of knowledge than the common developer/security person.

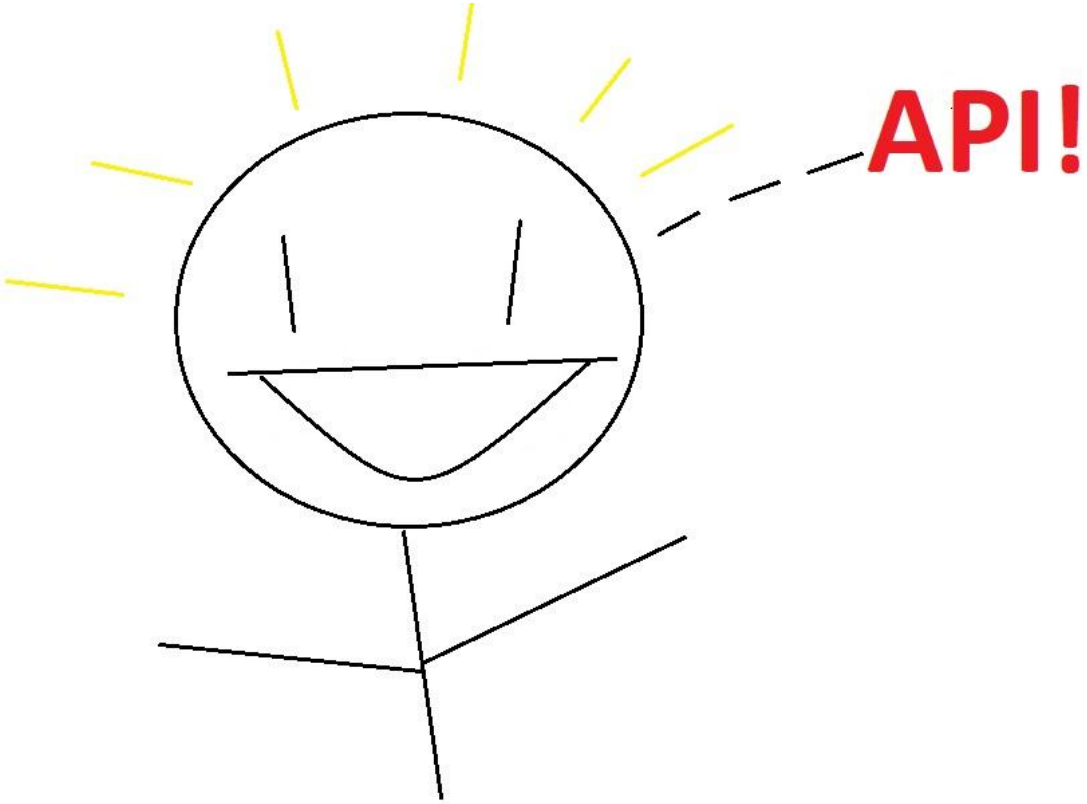
We need more UiO-CTFers!
Sounds fun? Join us!

What is an API?

API - A point of connection

- API (Application programming interface)
  - Part of web servers that receive requests
  - Functions that make calls to an operating system
  - Data that is created in a special format and that can be retrieved/changed by making requests

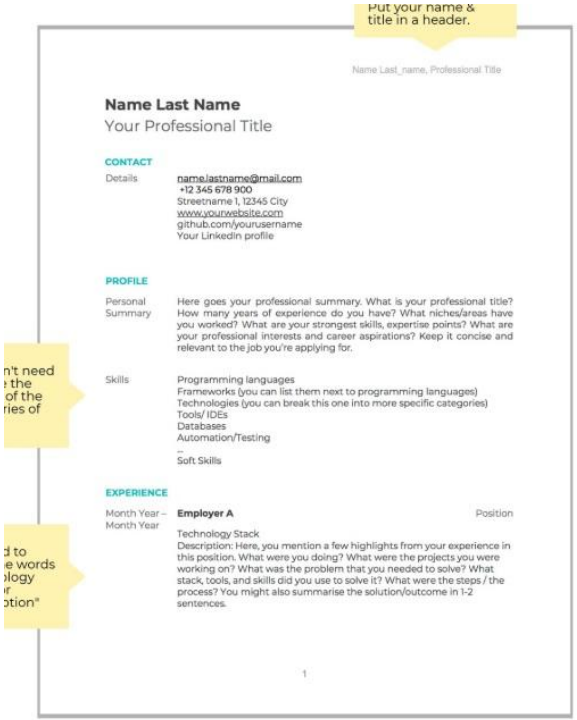



API!

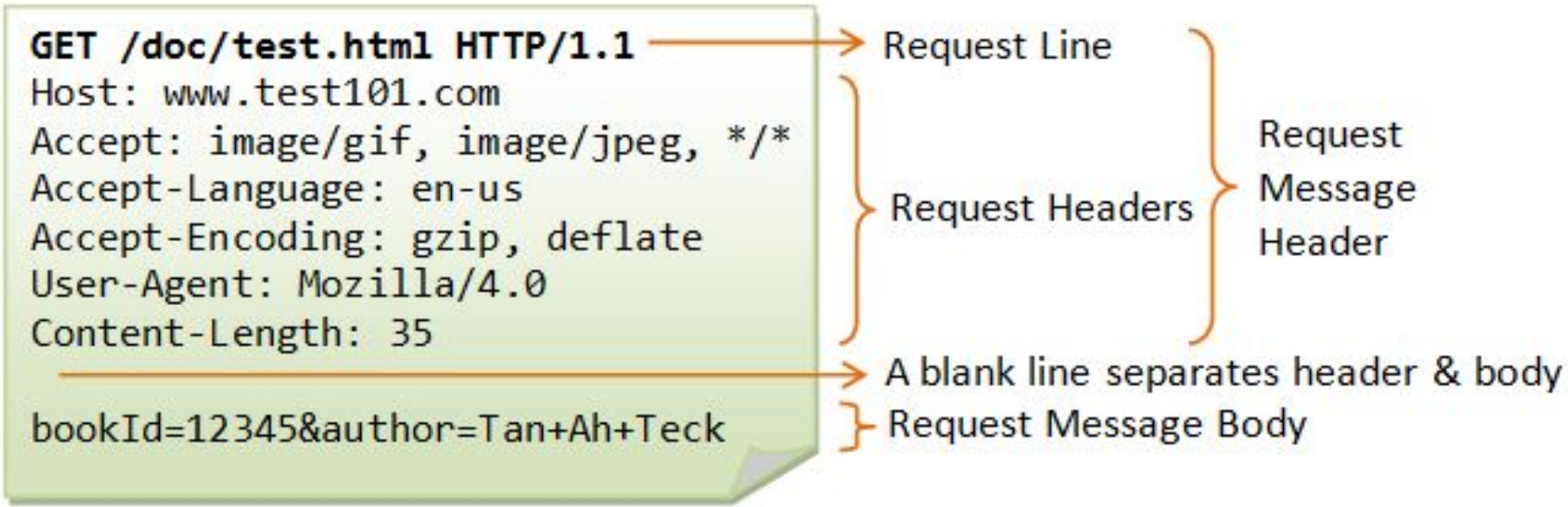
What is an HTTP request?

Purpose of HTTP

- HTTP is a standarized way to communicate with an application



HTTP Request



Response

`200 OK`

The request has succeeded. The meaning of the success depends on the HTTP method:

- `GET`: The resource has been fetched and is transmitted in the message body.
- `HEAD`: The entity headers are in the message body.
- `PUT` or `POST`: The resource describing the result of the action is transmitted in the message body.
- `TRACE`: The message body contains the request message as received by the server

Response

400 Bad Request

The server could not understand the request due to invalid syntax.

401 Unauthorized

Although the HTTP standard specifies "unauthorized", semantically this response means "unauthenticated". That is, the client must authenticate itself to get the requested response.

402 Payment Required

This response code is reserved for future use. Initial aim for creating this code was using it for digital payment systems, however this status code is used very rarely and no standard convention exists.

403 Forbidden

The client does not have access rights to the content, i.e. they are unauthorized, so server is rejecting to give proper response. Unlike 401, the client's identity is known to the server.

404 Not Found

The server can not find requested resource. In the browser, this means the URL is not recognized. In an API, this can also mean that the endpoint is valid but the resource itself does not exist. Servers may also send this response instead of 403 to hide the existence of a resource from an unauthorized client. This response code is probably the most famous one due to its frequent occurrence on the web.

HTTP response code resource

- https://developer.mozilla.org/en-US/docs/Web/HTTP/Status

Types of APIs

PokéAPI

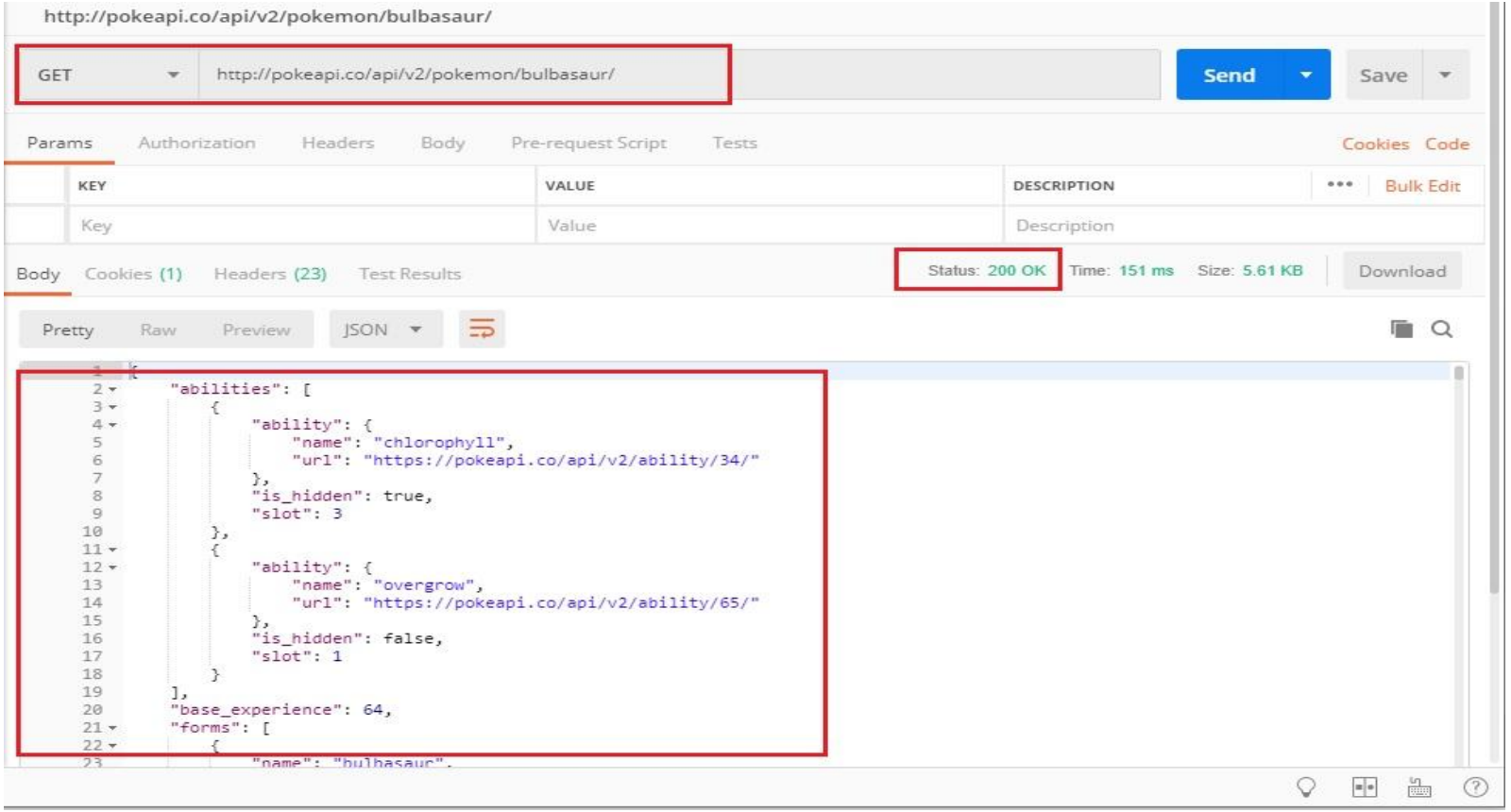
The RESTful Pokémon API

Serving over **17,000,000** API calls each month!

All the Pokémon data you'll ever need in one place,
easily accessible through a modern RESTful API.

Check out the docs!

REST API


http://pokeapi.co/api/v2/pokemon/bulbasaur/
GET
http://pokeapi.co/api/v2/pokemon/bulbasaur/
Send
Save
Params
Authorization
Headers
Body
Pre-request Script
Tests
Cookies
Code
KEY
VALUE
DESCRIPTION
Bulk Edit
Key
Value
Description
Body
Cookies (1)
Headers (23)
Test Results
Status: 200 OK
Time: 151 ms
Size: 5.61 KB
Download
Pretty
Raw
Preview
JSON
{
"abilities": [
{
"ability": {
"name": "chlorophyll",
"url": "https://pokeapi.co/api/v2/ability/34/"
},
"is_hidden": true,
"slot": 3
},
{
"ability": {
"name": "overgrow",
"url": "https://pokeapi.co/api/v2/ability/65/"
},
"is_hidden": false,
"slot": 1
}
],
"base_experience": 64,
"forms": [
{

REST fundamentals

- GET, PUT, POST and DELETE
- Architectural style that advocates use of HTTP as it was originally envisioned
- REST favours URLs but they don’t have to be pretty
  - “http://myserver.com/api/catalog/item/1729”
  - “http://myserver.com/api/addToCart?cart=314159&item=1729”

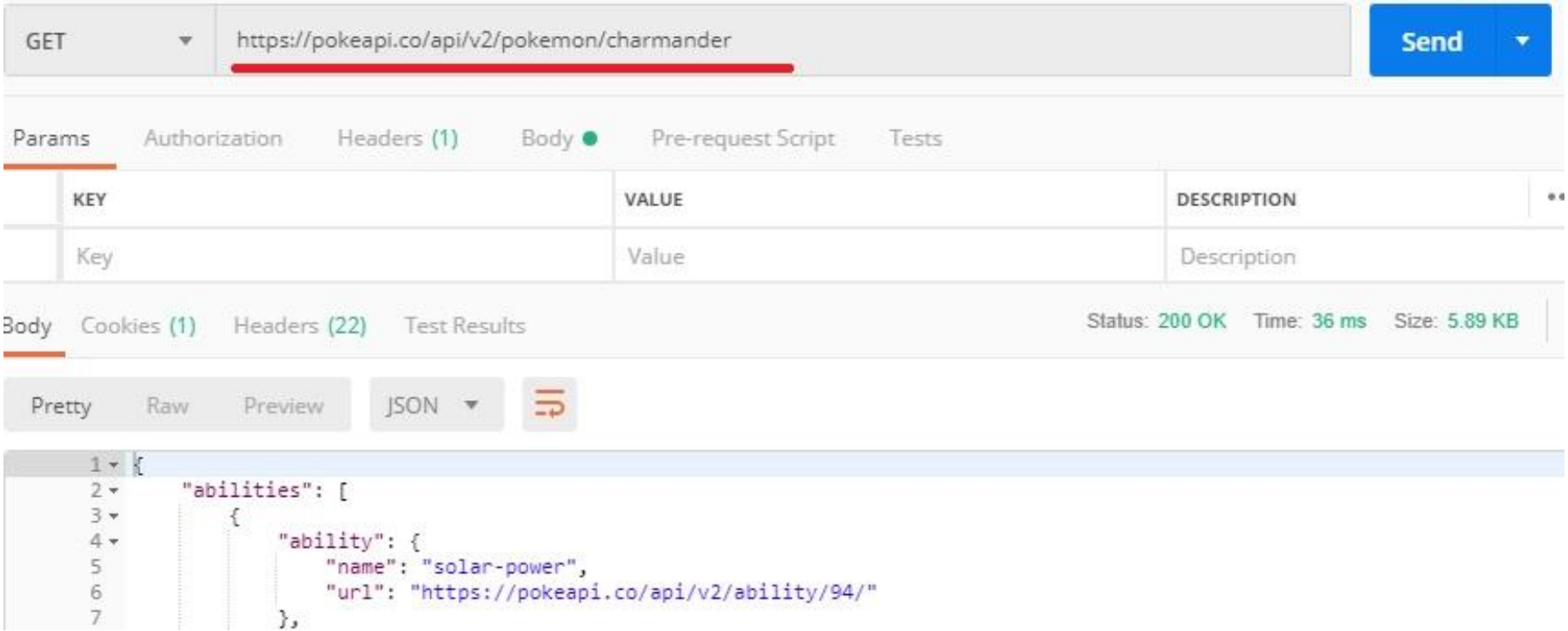
There are other types of APIs

- GraphQL, SOAP

```
<soapenv:Envelope
xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:hs="http://www.holidaywebservice.com/HolidayService_v2/">

<soapenv:Body>

<hs:GetHolidaysAvailable>

<hs:countryCode>UnitedStates</hs:countryCode>
```

Sending HTTP requests

Postman


GET
https://pokeapi.co/api/v2/pokemon/charmander
Send
Params
Authorization
Headers (1)
Body
Pre-request Script
Tests
KEY
VALUE
DESCRIPTION
Key
Value
Description
Body
Cookies (1)
Headers (22)
Test Results
Status: 200 OK
Time: 36 ms
Size: 5.89 KB
Pretty
Raw
Preview
JSON
{
"abilities": [
{
"ability": {
"name": "solar-power",
"url": "https://pokeapi.co/api/v2/ability/94/"
},

cURL

- curl -X GET https://pokeapi.co/api/v2/pokemon/charmander
- Install a tool like “jq” to see the JSON in a nicer format!

Let’s try do some tasks!

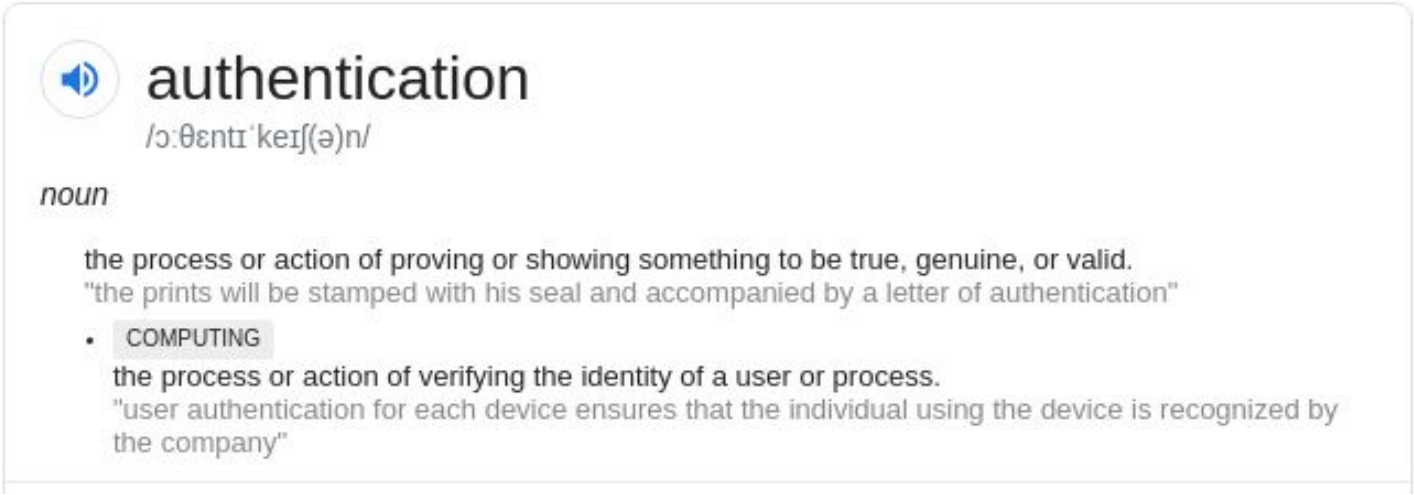
api-workshop.uioctf.no

API security

API security overview

- Authentication vs. authorization
- Scopes & roles
- Secure transfer
- Validation
- *Response messages*
- *Storage*

But first....
What is authentication and authorization?



authentication

/ɔːθɛntɪˈkeɪʃ(ə)n/

noun

the process or action of proving or showing something to be true, genuine, or valid.
"the prints will be stamped with his seal and accompanied by a letter of authentication"

- COMPUTING
  the process or action of verifying the identity of a user or process.
  "user authentication for each device ensures that the individual using the device is recognized by the company"

What is authentication and authorization?

“Authorization is the process to determine whether the authenticated user has access to a particular resource.”

https://medium.com/datadriveninvestor/authentication-vs-authorization-716fea914d55

Scopes and roles (Authorization)

- Scope example:
  - ‘create:pokemon’
  - ‘read:pokemon’
  - ‘delete:pokemon’
- Role example:
  - ‘blogger’
  - ‘admin’
  - ‘blogreader’

Transportation

- How is the authorization sent to the API?

  - Cookie? HTTP header?
  - Encrypted? Signed? Cleartext?

Attack vectors

- How to exploit authorization in APIs?

  - Scope or role manipulation
  - Not signed/encrypted
  - Bad validation

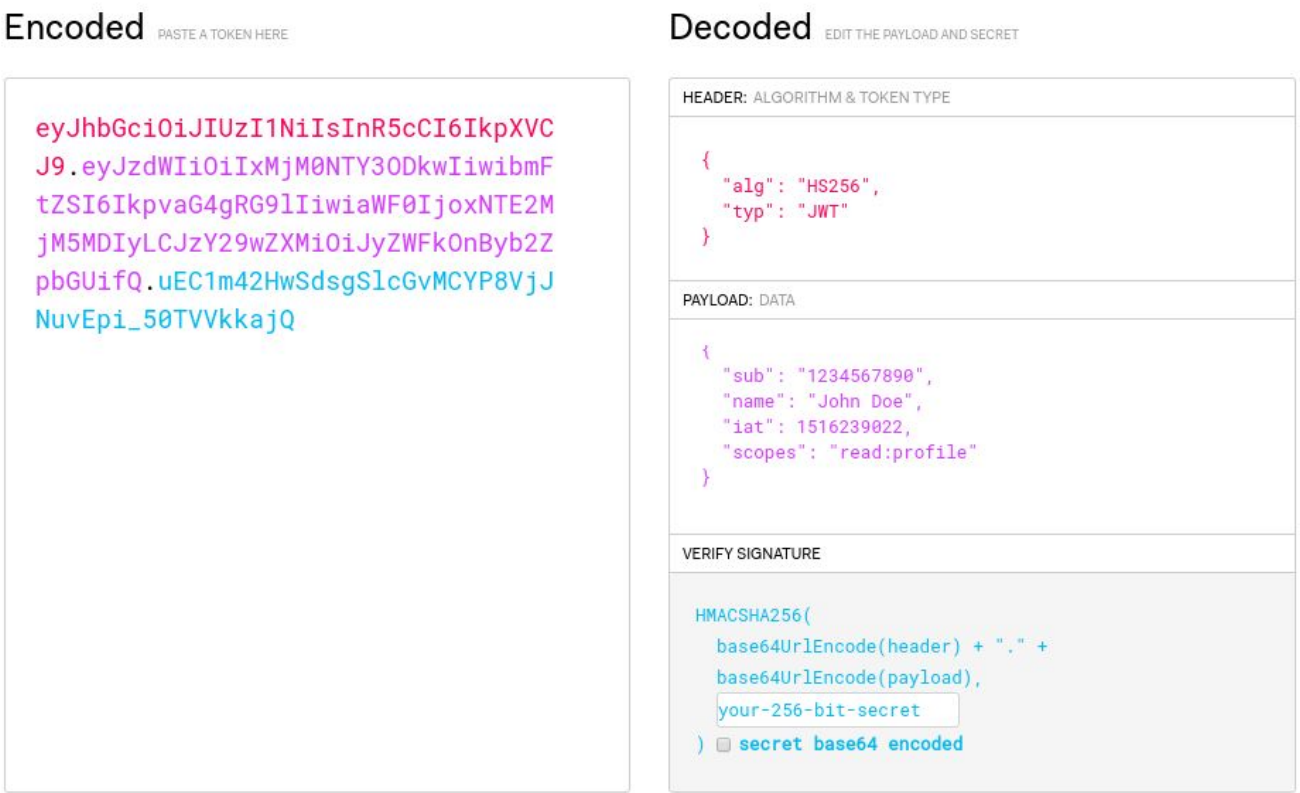
Example 1: Unencrypted cookie/token

- role=user -> ???
- c2NvcGU9cmVhZDpibG9ncG9zdA== -> ???

The newcomer: JSON Web Token!

- JWT
- Signed
- base64 encoded

Example 2: JSON Web Token (JWT)



Bad validation/implementation

- Not validating signature of JWT token, just decoding Base64
- Backdoors

Example 3: Bad validation/implementation

- Uhm, didn’t have enough time to do this. You will have to checkout the “Military grade encryption” task for example :))

CTF!? WTF?

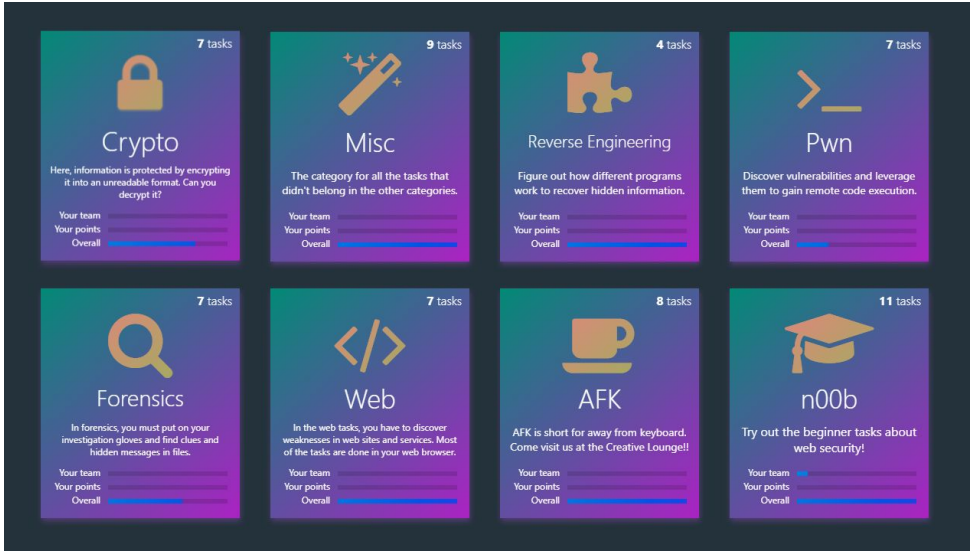
“CTF is a **hacking competition** with a wide spectre of security related **challenges** where the goal is to find a **flag**.”

Let’s try do the rest of the tasks!

api-workshop.uioctf.no

Like CTFs? Checkout TG:Hack!

https://tghack.no




I would tell you a
UDP joke, but you
might not get it
</NorCERT>
I would tell you a
UDP joke, but you
might not get it
</NorCERT>
I would tell you a
UDP joke, but you
might not get it
</NorCERT>

Sources

- https://www.howtogeek.com/343877/what-is-an-api/
- https://medium.com/datadriveninvestor/authentication-vs-authorization-716fea914d55
-