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.

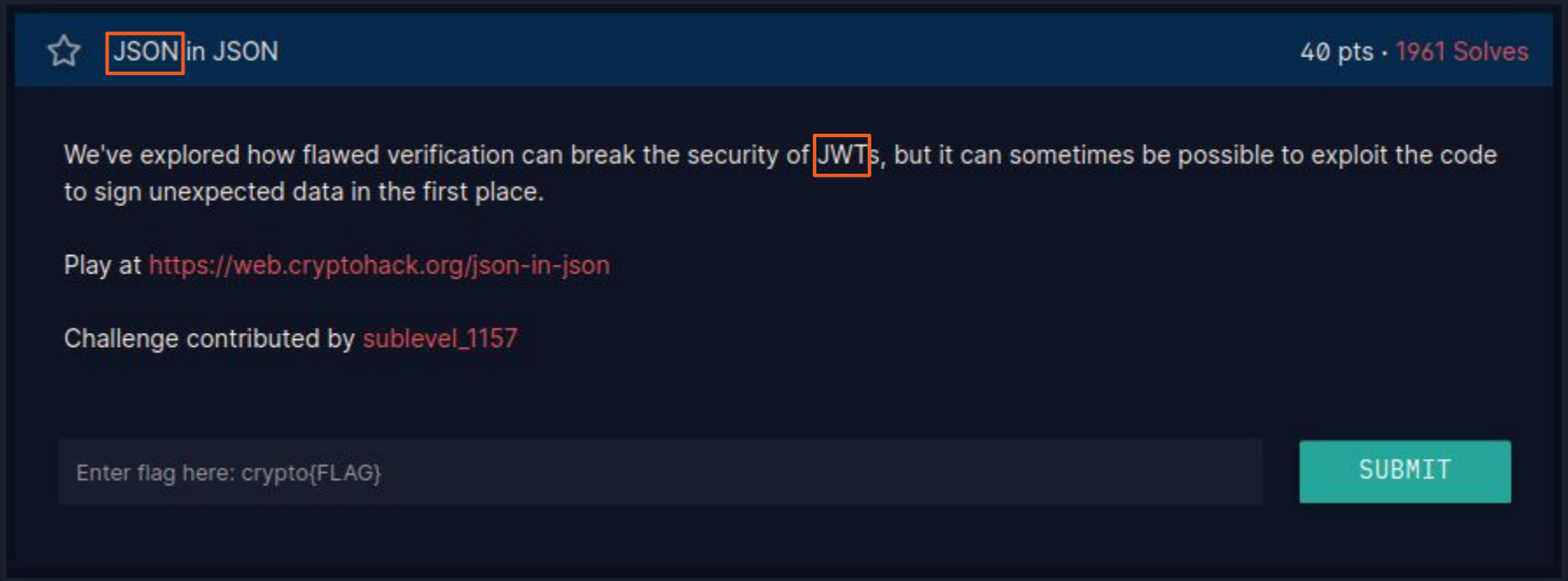
Play at https://web.cryptohack.org/json-in-json

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

une clé (str) une valeur (int, str, bool, None, list, dict)

```
{
    "menu": {
        "id": "file",
        "value": "File",
        "popup": {
            "menuitem": [
                { "value": "New", "onclick": "CreateNewDoc()" },
                { "value": "Open", "onclick": "OpenDoc()" },
                { "value": "Close", "onclick": "CloseDoc()" }
            ]
        }
    }
}
```

JSON - Des clés

```
{
    "menu": {
        "id": "file",
        "value": "File",
        "popup": {
            "menuitem": [
                { "value": "New", "onclick": "CreateNewDoc()" },
                { "value": "Open", "onclick": "OpenDoc()" },
                { "value": "Close", "onclick": "CloseDoc()" }
            ]
        }
    }
}
```

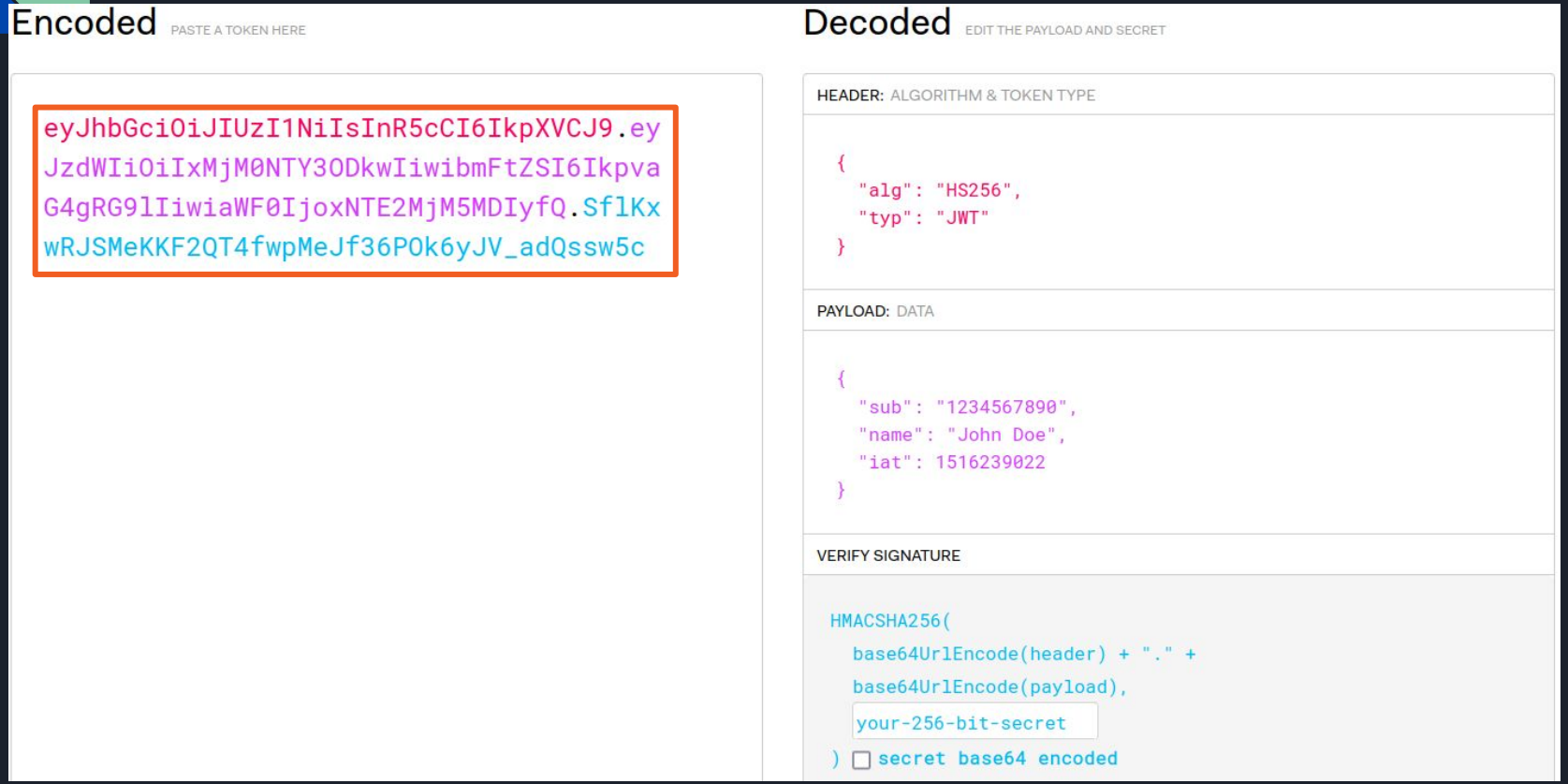
JSON - Des valeurs (part. 1)

```json
{
    "menu": {
        "id": "file",
        "value": "File",
        "popup": {
            "menuitem": [
                { "value": "New", "onclick": "CreateNewDoc()" },
                { "value": "Open", "onclick": "OpenDoc()" },
                { "value": "Close", "onclick": "CloseDoc()" }
            ]
        }
    }
}
```

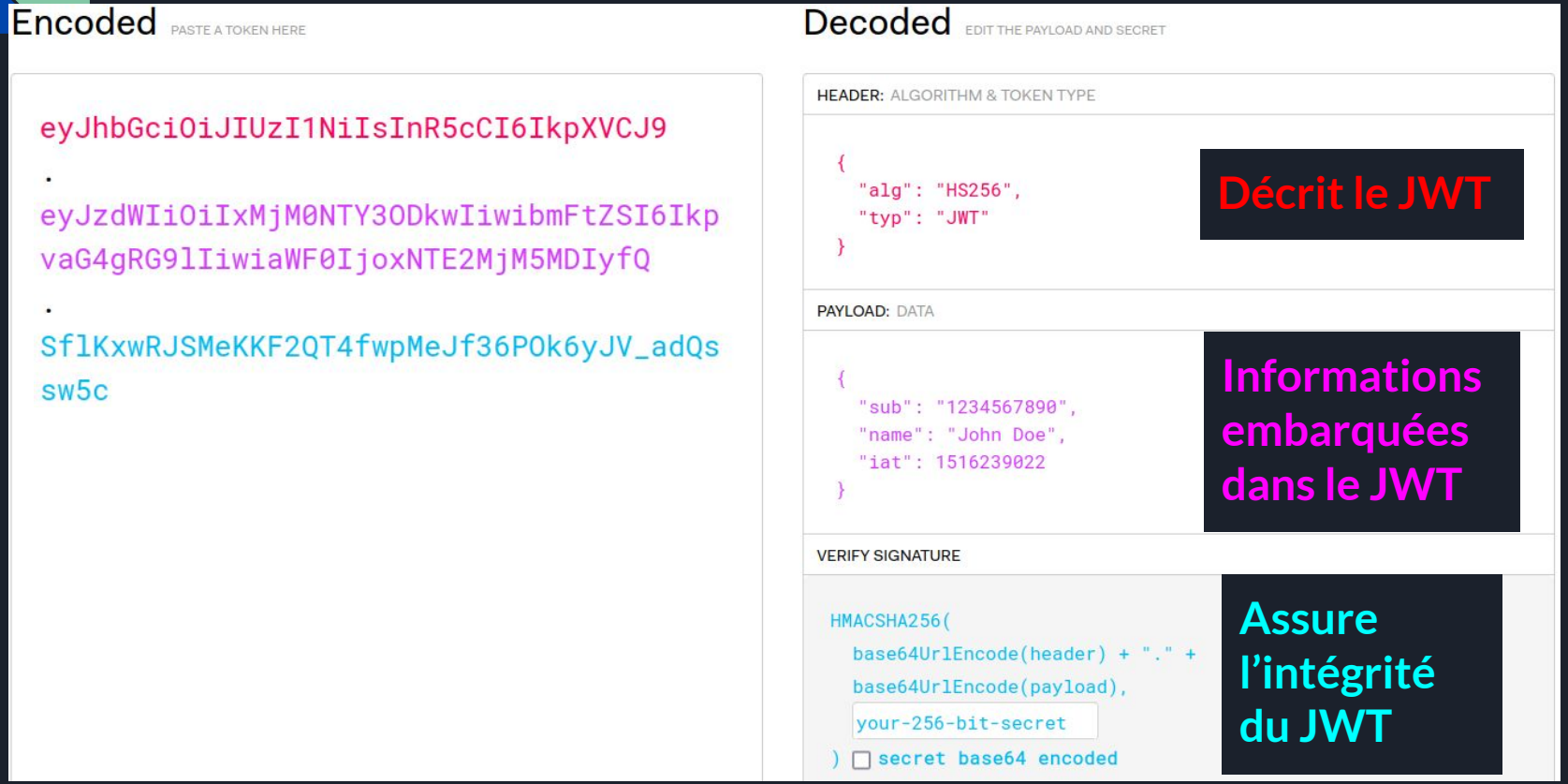
JSON - Des valeurs (part. 2)

```json
{
    "menu": {
        "id": "file",
        "value": "File",
        "popup": {
            "menuitem": [
                { "value": "New", "onclick": "CreateNewDoc()" },
                { "value": "Open", "onclick": "OpenDoc()" },
                { "value": "Close", "onclick": "CloseDoc()" }
            ]
        }
    }
}
```

JWT - JSON Web Token



JWT - JSON Web Token



Challenge - Analyse statique du code source

```python
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

```python
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"}
```

Code Source - Fonction *authorise*

```python
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"}
```

Code Source - Fonction *authorise*

```python
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"}
```

Code Source - Fonction *authorise*

JWT Objectif :

```
{
    "typ":"JWT",
    "alg":"HS256"
}
.
{
    "admin":"True"
}
.
SIGNATURE-NUMERIQUE
```

```
FLAG = ?
SECRET_KEY = ?

@chal.route('/json-in-jso
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"}
```

Code Source - Fonction *authorise*

```python
@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*

```python
@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 ⚠️

```python
@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*

Challenge - Analyse dynamique du code source

```python
def create_session(username):
    body = '{' \
                + '"admin": "' + "False" \
                + '", "username": "' + str(username) \
                + '"}'
    print(f"{body=}; {type(body)=}")
    print(f"{json.loads(body)=}; {type(json.loads(body))=}")
    encoded = jwt.encode(json.loads(body), SECRET_KEY, algorithm='HS256')
    print(f"{encoded=}; {type(encoded)=}")
    return {"session": encoded}

if __name__ == '__main__':
    username = input("Username: ")
    print(f"{username=}; {type(username)=}")
    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.eyJhZG1pbiI6IkZhbHNlIiwidXNlcm5h
bWUiOiJ5eGVuZSJ9.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.eyJhZG1pbiI6IkZhbHNlIiwidXNlcm5h
bWUiOiJ5eGVuZSJ9.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.eyJhZG1pbiI6IkZhbHNlIiwidXNlcm5h
bWUiOiJ5eGVuZSIsInJhbmRvbSI6InZhbHVlIn0.F-L2HbzpSjvHqECLO9RbGDrOmQYUHZRIUsSSUl
NRLiE'
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.eyJhZG1pbiI6IkZhbHNlIiwidXNlcm5h
bWUiOiJ5eGVuZSIsInJhbmRvbSI6InZhbHVlIn0.F-L2HbzpSjvHqECLO9RbGDrOmQYUHZRIUsSSUl
NRLiE'
type(encoded)=<class 'str'>
```

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

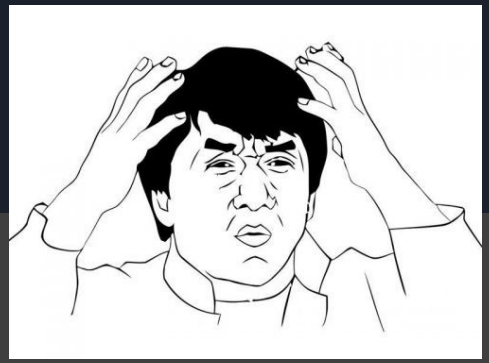
```
Username: yxene", "admin":"True
username='yxene", "admin":"True'
type(username)=<class 'str'>

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.eyJhZG1pbiI6IlRydWUiLCJ1c2VybmFt
ZSI6Inl4ZW5lIn0.VpxdwLy2RrEq2k0RyLgANkXpQgerIjW0yEBGfY_okBI'
type(encoded)=<class 'str'>
```

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



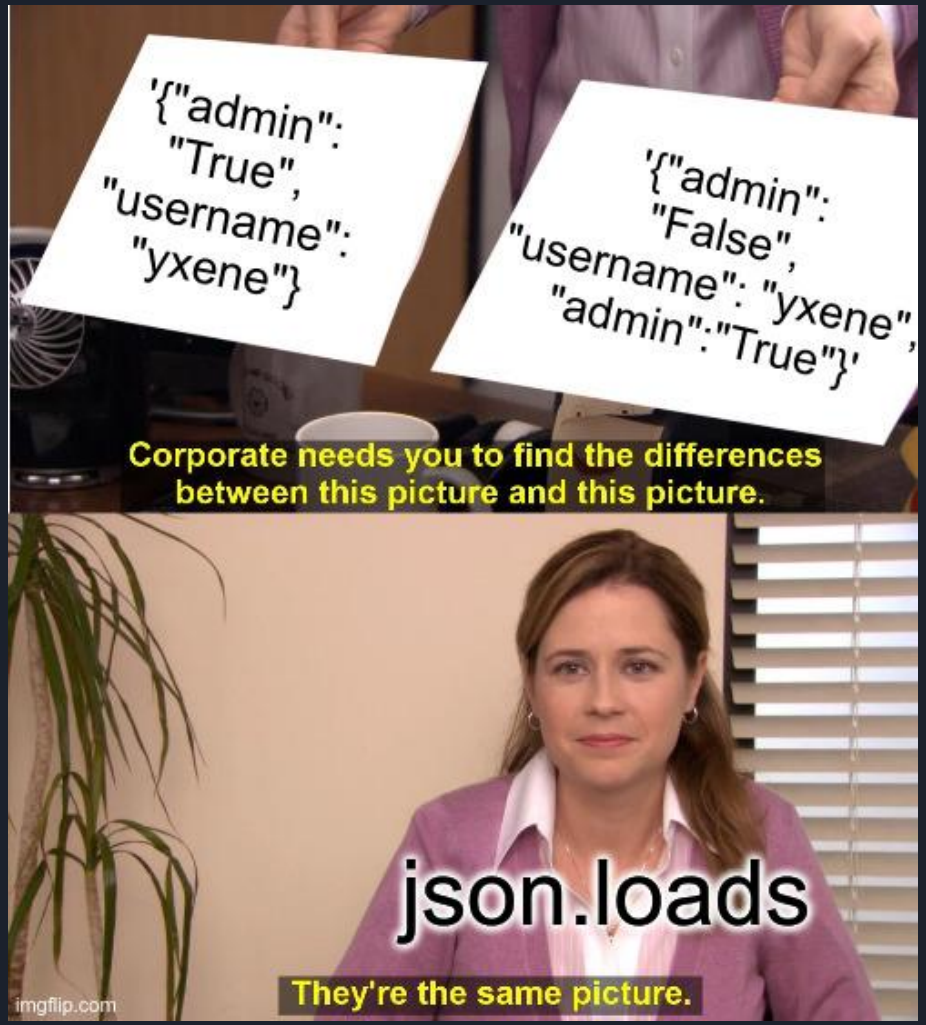
```
Username: yxene", "admin":"True
username='yxene", "admin":"True'
type(username)=<class 'str'>

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.eyJhZG1pbiI6IlRydWUiLCJ1c2VybmFt
ZSI6Inl4ZW5lIn0.VpxdwLy2RrEq2k0RyLgANkXpQgerIjW0yEBGfY_okBI'
type(encoded)=<class 'str'>
```

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


'{"admin": "True", "username": "yxene"}
'{"admin": "False", "username": "yxene", "admin":"True"}'
Corporate needs you to find the differences between this picture and this picture.
json.loads
They're the same picture.
imgflip.com

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.eyJhZG1pbiI6IlRydWUiLCJ1c2VybmFtZSI6Inl4ZW5lIn0.EVJIiRMnYFucUCz95L27YaASe5U6yAJMSAdjoF7Btz4"}

AUTHORISE(TOKEN)

token

:I1NiJ9.eyJhZG1pbiI6IlRydWUiLCJ1c2VybmFtZSI6Inl4ZW5lIn0.EVJliRMnYFucUCz95L27YaASe5U6yAJMSAdjoF7Btz4

String Input Only

SUBMIT

OUTPUT

{"response":"Welcome admin, here is your flag: [redacted]"}

Questions