Github 账号：Eyr3

实验题目：破解 SHA1-HASHED 密码

实验摘要

首先，参考键盘痕迹推断出 password 的长度以及可能组成字符，再利用 python 库函数高效地穷举出可能的不重复全排列，SHA1 生成摘要算法后与监测到的密码哈希值比较，若相等，则破解出密码。

题目描述

SHA1：

SHA1 在 1995 年被 NIST 定义为标准，并且在实践中，它是除 MD5 外使用最多的算法，其中一个例子是基于密文的认证。在这种情况下，服务器不用用户密码的明文，而是用其 SHA1 值存储。一旦用户输入他的密码并被服务器接收后，则会计算密码的哈希值并与已经存储在服务器上的值进行比较，以此验证密码的正确性。

情景预设：

一个脆弱的网页监视系统泄露了管理员账号密码的 SHA1 哈希值，密码的哈希值为：

67ae1a64661ac8b4494666f58c4822408dd0a3e4

此外，由于成功登录后的导航仅能通过方向键完成，因此登录时键盘的使用痕迹清楚地展示了已输入的密码。

所以管理员密码是什么？

实验过程

通过给出的键盘使用痕迹，可以看出密码为 8 位，且由键盘上 8 个字符组成。又因为一个键盘可能打出多种字符，则：

1. 遍历字母组合

例如：键盘 a 可打出：'5','%'；键盘 b 可打出：'8','('，则这两位的字符组合可由如下循环得。

```
for a in ['5','%']:
    for b in ['8','(']:
        base = a+b
```

八位的字符组合有 8! 种不重复的全排列，如下：

2. 生成 8 个字母的全排列

```
import itertools
base = '580qwin+'
for i in itertools.permutations(base, 8):
    data = ''.join(i)
```

注意：''.join(i) #i 由列表生成字符串

生成字符串后调用 python 的 hashlib 库，利用 sha1 函数，加密字符串，如下：

3. Python 调用 SHA1 摘要算法

```
import hashlib
data = '580qwin+'
hash_sha1 = hashlib.sha1(data).hexgigest()
```

注意：hexdigest()

计算程序运行时间，程序如下：

4. 计算程序运行时间

```
import time
start = time.clock()
elapesd = (time.clock() - start)
print("Time used:", elapesd)
```

最终所得结果为：

(Q=win*5

实验总结

一开始利用暴力穷举，代码很丑陋，不优雅，而且速度较慢，因此学习并利用 Python 库 itertools 中的 permutations 方法。方法嘛，就是 Google 找博客咯~不过最近十九大，被封的比较多。

参考文献

http://blog.chingzhu.com/?p=210

http://www.cnblogs.com/youxin/p/3157099.html

http://www.cnblogs.com/youxin/p/3157099.html