Project14: PoC impl of the scheme, or do implement analysis by Google

基于python的socket模块模拟客户端服务器的通信

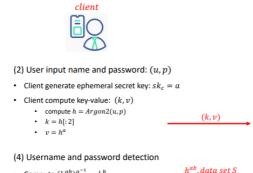
实现了如下图的功能:

3.7 Google Password Checkup

· Username and password detection

*Project: PoC impl of the scheme, or do implement analysis by Google

PART3 Application



- Compute $(h^{ab})^{a^{-1}} = h^b$
- Check whether h^b exists in S



(1) Process data info

- Data records: $(userName, password) \rightarrow (u_i, p_i)$
- Create key-value table (1TB): (k_i, v_i)
 - compute $h_i = Argon2(u_i, p_i)$
 - k_i is the first two bytes of h_i , namely $k_i \! = h_i [:2]$
 - $v_i = (h_i)^b$
- Divide the table into 2^{16} sets according to the key k_i (2 bytes)

- compute h^{ab}
- Find set S according to key k

其中选择了较小的a, b进行函数功能的验证

假设服务器端已经获得了泄露的账户密码如下:

```
username_passcode=[('zhangsan','12345'),('lisi','123456'),('wangwu','23415'),
                    ('zhaoliu','234567'),('zhengqi','345678'),('wuba','257890')]
```

客户端分别验证自己的两个账户密码是否泄露(结果应为第一个显示泄露第二个显示不泄露)

```
username_password=[('zhangsan','12345'),('zhangsan','zhangsan')]
```

在实现过程中 python的argon2库中的函数每次得到的值并不相同

单一拿argon2(username,password)值进行验证较为困难 改成了sm3求username||password的哈希 进行验证

代码在安装相应库的情况下可以直接运行

结果如下:

客户端运行结果:

```
password_checkup_client × password_c

D:\anaconda\python.exe E:/网安/大二上
('zhangsan', '12345') 账号有风险!
('zhangsan', 'zhangsan') 账号无风险!
```

服务器端运行结果:

```
password_checkup_client × ___ password_checkup_server ×
D:\anaconda\python.exe E:/网安/大二上课程ppt/pythor
启动监听,等待接入.....
成功连接: ('127.0.0.1', 50800)

开始验证新链接的合法性
链接合法,开始通信
Stop the server....

进程已结束,退出代码 0
```