MD5 长度扩展攻击 (未完成，存在问题)

2022 年 7 月 31 日

学　　校: 山东大学

学　　院: 网络空间安全学院 (研究院)

姓　　名: 张起萌 202000460118

目录

1 实验目的

实现 MD5 长度扩展攻击。

2 实验过程

MD5 的实现代码我使用的是参考文献中的代码 (有部分修改)，个人实现的部分是 MD5 长度扩展攻击的代码。

在该报告中所展示的仅为个人实现的代码，参考代码会上传到 github 仓库中 (MD5_main.py 文件)。

- 首先我利用在线 md5 的运算器计算出 meng 的 hash 值为 70110c42465beafda6c139ba93fe1eca，并将其作为 a 值输入。
- 根据标准公式 A=0X67452301，B=0XEFCDAB89，C=0X98BADCFE，D=0X10325476 计算出 ABCD 的值。
- 接下来填充：我们不知道 hash 值的原像，但我们知道其长度为 4，所以我们假设原像为 aaaa，将其补位为一个标准的 512bit 数据 (该 512bit 即为构造的 b 字符串，构造的字符串在代码的注释中展示)+a。
- 我们可以看出该数据长度长于 512bit，所以它会被填充为 1024bit。最后我们利用先前计算出的 ABCD 值计算后一部分 512bit 的 hash 值，即可实现 MD5 的长度扩展攻击。

  代码实现：

```
import md5_main
import sys
import six

a = '70110c42465beafda6c139ba93fe1eca'#meng的Hash值
length = 4

s1 = eval('0x' + a[:8].encode('utf-8')[::-1].decode('utf-8'))
s2 = eval('0x' + a[8:16].encode('utf-8')[::-1].decode('utf-8'))
s3 = eval('0x' + a[16:24].encode('utf-8')[::-1].decode('utf-8'))
s4 = eval('0x' + a[24:32].encode('utf-8')[::-1].decode('utf-8'))

print(s1,s2,s3,s4)

jiashe = "a" * length
test = jiashe + '\x80' + '\x00'* 51 + '\x20\x00\x00\x00\x00\x00\x00\x00' + 'meng'
#'aaaa'+'\x80\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00
#\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00
#\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x20\x00\x00\x00\x00\x00\x00\x00'+'meng'

s = md5_main.deal_rawInputMsg(b)
r = md5_main.deal_rawInputMsg(jiashe)
inp = s[len(r):]

print("填充后的md5为:" + md5_main.run_md5(s1,s2,s3,s4,inp))
```

3 实验总结

在实验过程中，我发现我参考的 MD5 的实现代码报错了，并且有很多处错误，而我并没有完全 debug 成功，所以最后并没有得到实验结果。

4 参考文献

MD5 的 Hash 长度扩展攻击