4th-长度扩展攻击学习＆Plaid CTF 2014 Crypto 250 Parlor

原创

ThAnO3 于 2020-05-27 08:47:22 发布 101 收藏

分类专栏： 密码学 入门 文章标签： python 密码学

版权声明：本文为博主原创文章，遵循 CC 4.0 BY-SA 版权协议，转载请附上原文出处链接和本声明。

本文链接：https://blog.csdn.net/weixin_44222346/article/details/106275666

版权

密码学 同时被 2 个专栏收录

7 篇文章 0 订阅

订阅专栏

入门

9 篇文章 0 订阅

订阅专栏

内容：学习掌握长度扩展攻击，明白攻击原理，知道攻击的条件，实践长度扩展攻击，题目：Plaid CTF 2014 Crypto 250 Parlor，完成WP

掌握长度扩展攻击

学习来源

博主

博客地址

长度扩展攻击（length extension attack），是指针对某些允许包含额外信息的加密散列函数的攻击手段。对于满足以下条件的散列函数，都可以作为攻击对象：

```
① 加密前将待加密的明文按一定规则填充到固定长度（例如512或1024比特）的倍数；

② 按照该固定长度，将明文分块加密，并用前一个块的加密结果，作为下一块加密的初始向量（Initial Vector）。

满足上述要求的散列函数称为Merkle-Damgård散列函数（Merkle-Damgård hash function），下列散列函数都属于Merkle-Damgård散列函数：
```

MD4
MD5
RIPEMD-160
SHA-0
SHA-1
SHA-256
SHA-512
WHIRLPOOL

对于H(salt+data)形式的加密，在以下条件满足的情况下，攻击者可以通过该方法获取H(salt+一定规则构造的data)：

① 知道密文的加密算法且该算法满足Merkle–Damgård散列函数特征；

```
② 不知道salt，但知道salt的长度，并可控制data的值；

③ 可以得到一个H(salt+data)的值。
```

攻击方法详解

下面以MD5算法为例，讲述该攻击方式如何进行攻击。

百度百科中详细阐述了MD5算法的实现过程https://baike.baidu.com/item/MD5/212708?fr=aladdin，我们并不需要知道MD5具体的算法是怎么回事，只需要知道它的实现是满足上面所说的Merkle-Damgård散列函数的两个条件的，具体过程是这样的：

① 填充

拿到明文后，MD5现将明文转为二进制文件，然后将二进制文件的长度除以512比特（即64字节），如果余数等于448比特（即64-8字节），那么直接在后面加上八个字节的长度标识，使之成为512比特的倍数。否则则在明文后填一个1，再填充0直至其长度除以512等于448，再加上8位的长度标识。长度是使用大端序（big Endian）来存储，即低字节放在高地址位上。

比如加密的明文是admin，其二进制文件以16进制表示是0x61646d696e，长度是40比特（5字节），那么需要补充408比特（51字节）的填充符，填充内容第一位是1，其余全部是0。16进制表示是0x8000000000000000000000000000000000000000000000000000。然后再添加8个字节的长度标识，admin长度为40比特，16进制是0x28，这个28要放在高位，就是0x280000000000000。结果如下所示：

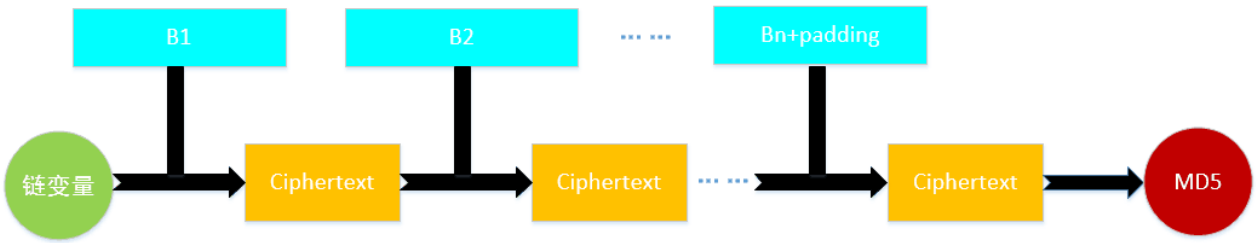
61	64	6d	69	6e	80	00	00	00	00	00	00	00	00	00	00	admin□
00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	
00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	
00	00	00	00	00	00	00	00	28	00	00	00	00	00	00	00	(

② 分块运算

填充完毕后，函数就将填充后的明文以512比特的长度分块，进行运算。在运算中会用到四个初始向量（MD5中称作链变量，Chaining Variable），分别是A=0x67452301，B=0xefcdab89，C=0x98badcfe，D=0x10325476。经过一系列复杂的数学运算，函数会得到第一块的MD5值，然后将该MD5值分成四块，以大端序形成新的链变量，投入到第二块的运算，形成新的MD5值……以此类推，直到算出最后一块的MD5值，就是整个数据块的MD5值。

例如加密的明文是adminadminadminadminadminadminadminadminadminadminadminadminadminadminadminadminadminadminadminadmin，首先根据①的描述的算法填充后，分成两块进行运算，第一块是adminadminadminadminadminadminadminadminadminadminadminadmi，其MD5码是e7d6ca05773d038378f5e2674850be25，分成四块并以大端序存储，则A=0x05cad6e7，B=0x83033d77，C=0x76e2f578，D=0x25be5048。将这四个变量作为链变量投入运算，再将第二部分加密，得到最终的MD5值是9ea2d490481dbcdadf61e7e404b99585。

流程如下图所示：

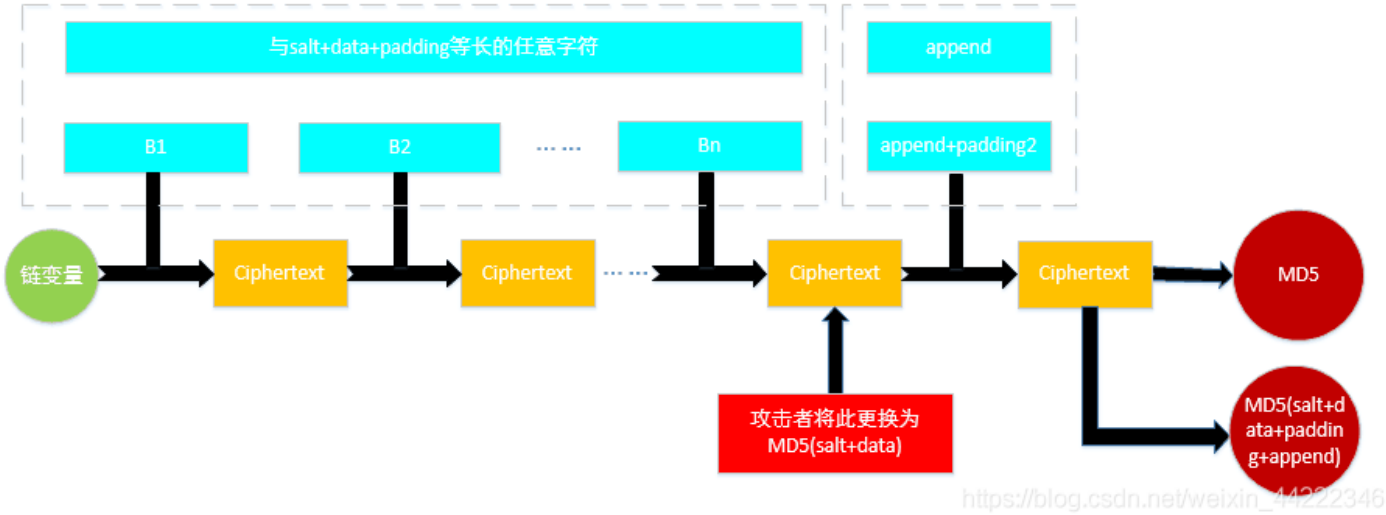


https://blog.csdn.net/weixin_44222346

如果攻击者知道MD5(salt+data)的值并可控制data的值，攻击者可以设定data为与data+padding+append等长的任意字符串，然后计算MD5(str+append)。我们知道，MD5需要先填充再运算，攻击者可以在程序计算append所在块之前，将MD5(salt+data)的值直接替换掉初始的链变量，就能够算出MD5(salt+data+padding+append)的值了。设MD5(CV,data)表示以链变量CV计算data的MD5值，那么（为简便起见，这里设append的长度不超过448比特，超过的原理也类似）

MD5(IV, salt+data+padding+append) = MD5(MD5(IV, salt+data), append)。

攻击原理如下：



之所以要知道salt的长度，是为了确保salt+data+padding+append和攻击者输入的data拥有相同的填充（即padding2相等），以确保最后一步的运算得到相同的结果。

Plaid CTF 2014 Crypto 250 Parlor

找到了2个关于这个题目的网站（仅有的2个）

1

2

解法看了很多遍，代码也试运行，终究是冒红。

```
nonce = "760c4a0f8ec61bec304ed4d8d8abeb98".decode('hex')
num = 'a\n'
md5(nonce + num) = '5b356daa0313063af25f8da01922128d'
a,b,c,d = md5tonum(md5(nonce + num))
# nonce = 16, a\n = 2, 所以填充\x80+\x00*37 + len 8, total = 64
padding = "\x80"+"\x00"*37 + "\x90"+"\x00"*7
print md5('a\n'+padding+'b\n')
block = [256511094 3961243278 3637792304 2565581784 8391265 0 0 0 0 0 0 0 0 0 144 0 8391266 0 0 0 0 0 0 0 0 0 0
0 0 0 528 0]
md5: 12b74d8200ff1c84500b1e55ada2ce7e
print guess('b\n',a,b,c,d,66) # 新的长度是 66 byte
block = [8391266 0 0 0 0 0 0 0 0 0 0 0 0 0 16 0]
md5: 12b74d8200ff1c84500b1e55ada2ce7e
```

最初md5冒红，想着是我的python是没有md5的包。查阅后发现python3的md5已被删除。然后自己找了MD5的代码，加入进去，代码中的md5倒是没有冒红，但是下面block的第二部分又冒红了，没有跑通。

那个block没有看懂，也是冒红的地方，在此记录，日后研究请教，待到功力长进，一定吃透。（PS：最后一天了没时间了，得交任务了。不过以我现在的水平，时间再多一点也不能学懂）

在学习三周理论后开始在攻防世界做题了。

base64、摩斯密码、凯撒位移密码。RSA也用到了

把之前学过的东西运用了起来，感觉真不一样。

one more thing：这次就没有什么想记的了。

创作打卡挑战赛
赢取流量/现金/CSDN周边激励大奖