Computer Security HW9 Write-Up

Student ID: B07902143
Account: soyccan
Name: 陳正康

Cathub Party

FLAG{EE0DF17A410C90F86E88471346B6DA77E8C878200B37E60C53E9A56913211465}

Cookie

登入後看一下 cookie，看到有個 flag
把它 url unquote，再 base64 decode 後
發現都是 96 bytes
先猜 block size 是 16 byte

Padding Oracle Attack

任意改最後一個字元，發現會出現
`What the flag?! CHEATER!!! get out of here.`
但改第一個字元，會出現
`Your flag seems strange @@... okay....`

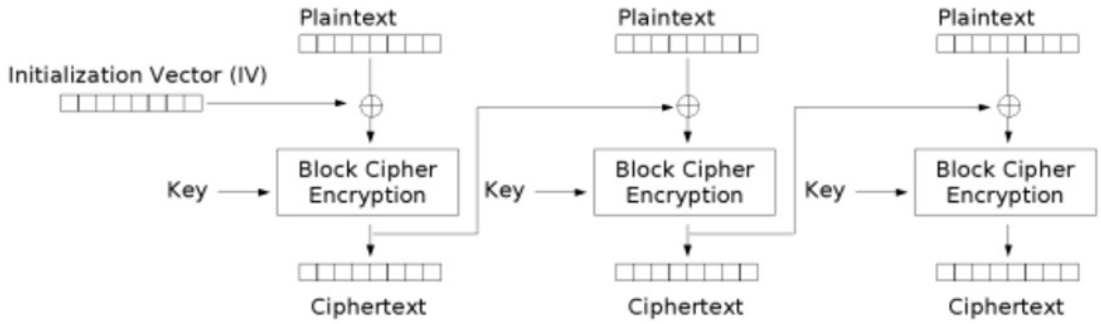
可知猜到是 padding 的問題

就當成 CBC + PKCS#7 的 padding oracle attack 去解
就可以解出除了前 16 bytes 的明文
flag 就在裡面

Details

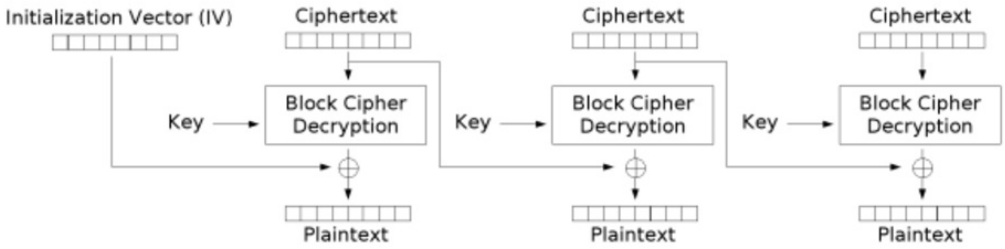
Reference: https://skysec.top/2017/12/13/padding-oracle和cbc翻转攻击/

CBC加密模式：



Cipher Block Chaining (CBC) mode encryption

CBC解密模式：



Cipher Block Chaining (CBC) mode decryption

Padding Oracle Attack攻擊過程

這裏主要關註一下解密過程
密文cipher首先進行一系列處理，如圖中的Block Cipher Decryption
我們將處理後的值稱為middle中間值
然後middle與我們輸入的iv進行異或操作
得到的即為明文
但這裏有一個規則叫做Padding填充：
因為加密是按照16位一組分組進行的
而如果不足16位，就需要進行填充

	BLOCK #1								BLOCK #2							
	1	2	3	4	5	6	7	8	1	2	3	4	5	6	7	8
Ex 1	F	I	G													
Ex 1 (Padded)	F	I	G	0x05	0x05	0x05	0x05	0x05								
Ex 2	B	A	N	A	N	A										
Ex 2 (Padded)	B	A	N	A	N	A	0x02	0x02								
Ex 3	A	V	O	C	A	D	O									
Ex 3 (Padded)	A	V	O	C	A	D	O	0x01								
Ex 4	P	L	A	N	T	A	I	N								
Ex 4 (Padded)	P	L	A	N	T	A	I	N	0x08	0x08	0x08	0x08	0x08	0x08	0x08	0x08
Ex 5	P	A	S	S	I	O	N	F	R	U	I	T				
Ex 5 (Padded)	P	A	S	S	I	O	N	F	R	U	I	T	0x04	0x04	0x04	0x04

比如我們的明文為admin

則需要被填充為 admin\x0b\x0b\x0b\x0b\x0b\x0b\x0b\x0b\x0b\x0b\x0b
一共11個\x0b
如果我們輸入一個錯誤的iv，依舊是可以解密的，但是middle和我們輸入的iv經過異或後得到的填充值可能出現錯誤
比如本來應該是admin\x0b\x0b\x0b\x0b\x0b\x0b\x0b\x0b\x0b\x0b\x0b
而我們錯誤的得到admin\x0b\x0b\x0b\x0b\x0b\x0b\x0b\x0b\x0b\x3b\x2c
這樣解密程序往往會拋出異常(Padding Error)
應用在web裏的時候，往往是302或是500報錯
而正常解密的時候是200
所以這時，我們可以根據服務器的反應來判斷我們輸入的iv
我們假設middle中間值為(為了方便，這裏按8位分組來闡述)

1

0x39 0x73 0x23 0x22 0x07 0x6a 0x26 0x3d

正確的解密iv應該為

1

0x6d 0x36 0x70 0x76 0x03 0x6e 0x22 0x39

解密後正確的明文為：

1

T E S T 0x04 0x04 0x04 0x04

但是關鍵點在於，我們可以知道iv的值，卻不能得到中間值和解密後明文的值
而我們的目標是只根據我們輸入的iv值和服務器的狀態去判斷出解密後明文的值
這裏的攻擊即叫做Padding Oracle Attack攻擊
這時候我們選擇進行爆破攻擊
首先輸入iv

1

0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00

這時候和中間值middle進行異或得到：

1

0x39 0x73 0x23 0x22 0x07 0x6a 0x26 0x3d

而此時程序會校驗最後一位padding字節是否正確
我們知道正確的padding的值應該只有0x01~0x08，這裏是0x3d，顯然是錯誤的
所以程序會拋出500
知道這一點後，我們可以通過遍歷最後一位iv，從而使這個iv和middle值異或後的最後一位是我們需要0x01
這時候有256種可能，不難遍歷出
Iv:

1

0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x3c

Middle:

1

0x39 0x73 0x23 0x22 0x07 0x6a 0x26 0x3d

兩者異或後得到的是：

1

0x39 0x73 0x23 0x22 0x07 0x6a 0x26 0x01

這時候程序校驗最後一位，發現是0x01，即可通過校驗，服務器返回200
我們根據這個200就可以判斷出，這個iv正確了
然後我們有公式：

1
2

Middle[8]^原來的iv[8] = plain[8]
Middle[8]^現在的iv[8] = 0x01

故此，我們可以算出middle[8] = 0x01^現在的iv[8]
然後帶入式1：

1

Plain[8] = 0x01$^{現在的iv[8]}$原來的iv

即可獲取明文plain[8]= 0x01^{0x3c}0x39=0x04
和我們之前解密成功的明文一致（最後4位為填充）
下面我們需要獲取plain[7]
方法還是如出一轍
但是這裏需要將iv更新，因為這次我們需要的是2個0x02，而非之前的一個0x01
所以我們需要將現在的iv[8] = middle[8]^0x02
（
為什麼是現在iv[8] = middle[8]^0x02？
因為現在的iv[8]^middle[8]=服務器校驗的值
而我們遍歷倒數第二位，應該是2個0x02，所以服務器希望得到的是0x02，所以

1
2

現在的iv[8]^middle[8]=0x02
故此iv[8] = middle[8]^0x02

）
然後再繼續遍歷現在的iv[7]
方法還是和上面一樣，遍歷後可以得到
Iv:

1

0x00 0x00 0x00 0x00 0x00 0x00 0x24 0x3f

Middle:

1

0x39 0x73 0x23 0x22 0x07 0x6a 0x26 0x3d

兩者異或後得到的是：

1

0x39 0x73 0x23 0x22 0x07 0x6a 0x02 0x02

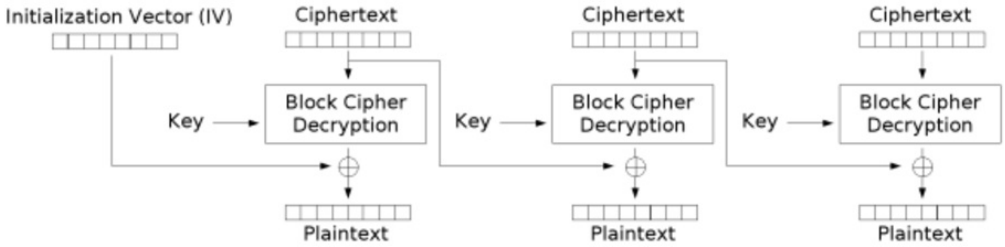
然後此時的明文值：

1

Plain[7]=現在的iv[7]$^{原來的iv[7]}$0x02

所以Plain[7] = 0x02^{0x24}0x22=0x04
和我們之前解密成功的明文一致（最後4位為填充）
最後遍歷循環，即可得到完整的plain

CBC翻轉攻擊過程



Cipher Block Chaining (CBC) mode decryption

這個實際上和padding oracle攻擊差不多

還是關註這個解密過程
但這時，我們是已知明文，想利用iv去改變解密後的明文
比如我們知道明文解密後是1dmin
我們想構造一個iv，讓他解密後變成admin
還是原來的思路

1

原來的Iv[1]^middle[1]=plain[1]

而此時
我們想要

1

構造的iv[1]^mddle[1]=’a’

所以我們可以得到

1

構造的iv[1] = middle[1]^’a’

而

1

middle[1]=原來的iv[1]^plain[1]

所以最後可以得到公式

1

構造的iv[1]= 原來的iv[1] ^ plain[1] ^ 'a'

所以即可造成數據的偽造
我們可以用這個式子，遍歷明文，構造出iv，讓程序解密出我們想要的明文
題目題解：
有了上面的知識基礎，我們就可以很快速的破解這道題
首先是我們未知Plain,即這裏的global $id
所以可以利用padding oracle攻擊去得到這個值plain
然後得到這個值後，再利用cbc翻轉攻擊，將這個plain偽造成我們需要的admin