Handin 9: DISSY DA7 8

cbc-padding-oracle-task

Attachment description:

1. final_server.py

   Main program can recover secret successfully.

   The **default** target is the local host. If you want to change the target to web server, please comment the local part (in line 20) and comment out the website part (in line 25)

Approach：

Target: recover secret

Since the prompt has stated that the method is oracle padding, the server has related vulnerabilities. The server provides 4 types of responses (including padding error) in the following table, so the first part is completed with oracle padding to decrypt ciphertext

	Content	Meaning
1	PKCS#7 padding is incorrect. **Len=28**	Padding error
2	Padding is incorrect. **Len=21**	Padding error
3	'utf-8' codec can't decode byte 0xxx in position x: invalid continuation byte. **Len=77**	No Padding error
4	Nice try!. **Len=9**	No Padding error

CBC decryption：

P[i] = D(C[i]) XOR C[i-1]

C[0] = IV

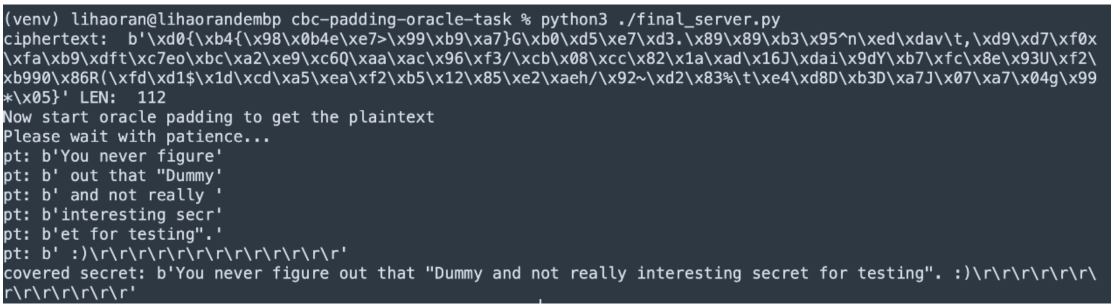
Take the ciphertext on the local server as an example (token size is 112 bytes, that is 7 blocks with IV). For example, assume the token has a total of 64 bytes. According to the 16byte BLOCK_SIZE, ciphertext occupies a total of 3 blocks, that is, token=IV+C[1]+C[2]+C[3]. I need to know D(C[1]), D(C[2]), D(C[3]) to get plaintext.

In short, **oracle padding is a way to get intermediate variable D(C[i]) by the padding error response from the server.** The method "single attack" is referred to the link: https://research.nccgroup.com/2021/02/17/cryptopals-exploiting-cbc-padding-oracles/

At present, I only know C[1], C[2], C[3]. I have to use oracle padding to know D(C[1]), D(C[2]), D(C[3]) and then XOR operations by the formula above to cover the secret.

According to the above table, it is found that there is no padding error in response 2 and 4, so when the length of content is less than 20 or greater than 50, it can be considered to find a value that meets the requirements.

Result:



```
(venv) lihaoran@lihaorandembp cbc-padding-oracle-task % python3 ./final_server.py
ciphertext:  b'\xd0{\xb4{\x98\x0b4e\xe7>\x99\xb9\xa7}G\xb0\xd5\xe7\xd3.\x89\x89\xb3\x95^n\xed\xdav\t,\xd9\xd7\xf0x
\xfa\xb9\xdft\xc7eo\xbc\xa2\xe9\xc6Q\xaa\xac\x96\xf3/\xcb\x08\xcc\x82\x1a\xad\x16J\xdai\x9dY\xb7\xfc\x8e\x93U\xf2\
xb990\x86R(\xfd\xd1$\x1d\xcd\xa5\xea\xf2\xb5\x12\x85\xe2\xaeh/\x92~\xd2\x83%\t\xe4\xd8D\xb3D\xa7J\x07\xa7\x04g\x99
*\x05}' LEN:  112
Now start oracle padding to get the plaintext
Please wait with patience...
pt: b'You never figure'
pt: b' out that "Dummy'
pt: b' and not really '
pt: b'interesting secr'
pt: b'et for testing".'
pt: b' :)\r\r\r\r\r\r\r\r\r\r\r\r\r'
covered secret: b'You never figure out that "Dummy and not really interesting secret for testing". :)\r\r\r\r\r\r\
r\r\r\r\r\r\r'
```

This is the result we got from local, we can still correct result from web server but it takes more time to execute.