picoCTF2025-Writeup

kindun

2025 年 3 月 20 日

1 初めに

本記事の内容は筆者の理解に基づいており, 誤りが含まれる可能性があります.

2 Cookie Monster Secret Recipe[Web Exploitation]

2.1 問題分

Cookie Monster has hidden his top-secret cookie recipe somewhere on his website. As an aspiring cookie detective, your mission is to uncover this delectable secret. Can you outsmart Cookie Monster and find the hidden recipe? You can access the Cookie Monster here and good luck

2.2 解法

2.2.1 cookie に着目

1. **ctrl+shift+i で開発者ツール (呼び名色々) を開く**
2. **上タブの Storage をクリック**
3. **左タブの Cokkies をクリック → すぐ下の URL をクリック**
   右側になにか出ていたら次の節に移動
4. **開発者ツールは閉じずに適当に Username と Password を入力し Login する**

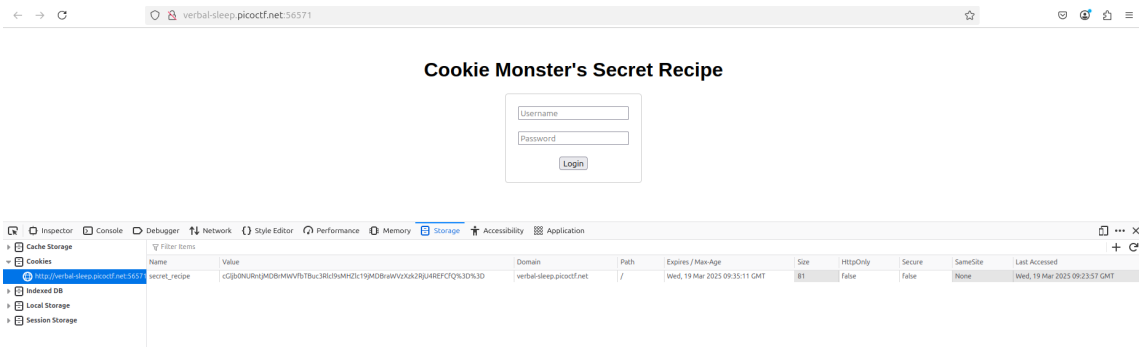

図 1 cookie

2.2.2 URL をデコード by CyberChef

1. **文字列を確認**

   Name が secret __ recipe とあることから flag だと分かる.

   問題文的にも cookie になにかあることが推測できる.

   しかし, このままではいけなさそう.picoCTF の形式ではない

2. **デコード**

   文字列の中に%があるので URL でエンコードされている可能性がある.

   とりあえず CyberChef の Magic を使ってみる.

   だめだったので次に URL Decode を使ってみる.

   デコードができた. やはり URL でエンコードされていた.

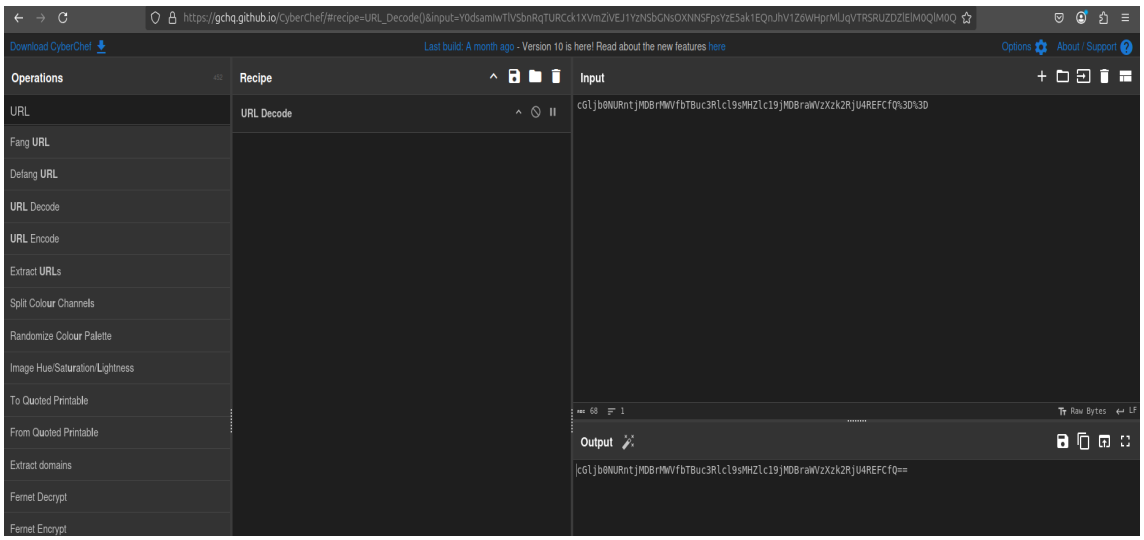

図 2　URL_decode

2.2.3 base64 をデコード by CyberChef

1. **とりあえず,Cyberchef の Magic を使う.**

   base64 でエンコードされていたことが分かる.

   デコードされた結果が出ていて, それが flag だと分かる.

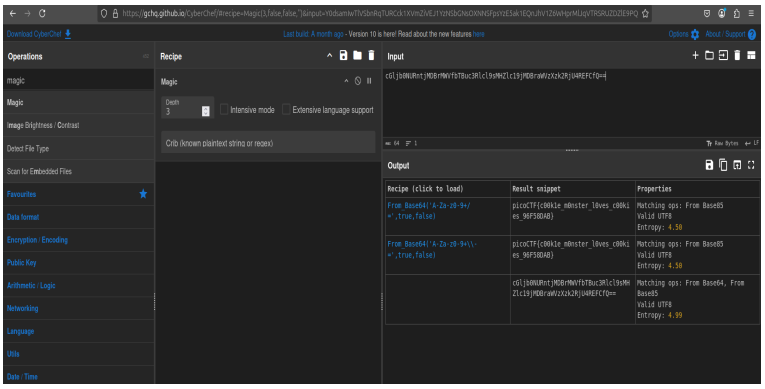

図 3　base64_decode

3　PIE TIME[Binary Exploitation]

3.1　問題文

Can you try to get the flag? Beware we have PIE! Additional details will be available after launching your challenge instance.

3.2　解法

3.2.1　問題の全体図を捉える

1. **Launch Instance をクリックし,nc コマンドで実行**

```
$nc rescued-float.picoctf.net ○○
Address of main: 0x55f9a559433d
Enter the address to jump to, ex => 0x12345:
```

main のアドレスが書いてあり, 入力するとアドレスがジャンプするらしい

2. **とりあえず, プログラムをダウンロードし, 見てみる (一応 binary ファイルも)** 中身はこんな感じ

```c
#include <stdio.h>
#include <stdlib.h>
#include <signal.h>
#include <unistd.h>

void segfault_handler() {
  printf("Segfault␣Occurred,␣incorrect␣address.\n");
  exit(0);
}

int win() {
  FILE *fptr;
  char c;

```

```
15    printf("You won!\n");
16    // Open file
17    fptr = fopen("flag.txt", "r");
18    if (fptr == NULL)
19    {
20        printf("Cannot open file.\n");
21        exit(0);
22    }
23
24    // Read contents from file
25    c = fgetc(fptr);
26    while (c != EOF)
27    {
28        printf ("%c", c);
29        c = fgetc(fptr);
30    }
31
32    printf("\n");
33    fclose(fptr);
34  }
35
36  int main() {
37    signal(SIGSEGV, segfault_handler);
38    setvbuf(stdout, NULL, _IONBF, 0); // _IONBF = Unbuffered
39
40    printf("Address of main: %p\n", &main);
41
42    unsigned long val;
43    printf("Enter the address to jump to, ex => 0x12345: ");
44    scanf("%lx", &val);
45    printf("Your input: %lx\n", val);
46
47    void (*foo)(void) = (void (*)())val;
48    foo();
49  }
```

見てみると.win 関数を実行できれば flag.txt が見れることが分かる.

3. **44,47,48 行のプログラムに注目**

```
1        void (*foo)(void) = (void (*)())val;
```

void 型 foo という関数ポインタに入力値 val を代入している.

```
1        foo();
```

よって, 入力したアドレスの関数が実行されるプログラムであるとわかった.

4. main 関数と win 関数のアドレスを知りたいダウンロードした ELF ファイル (vuln) にいろいろ書いてあるのでコマンドを使って見てみる.

```
$ vuln | grep -w -e "win" -w -e "main"
000000000000133d T main
00000000000012a7 T win
```

-w は完全一致か,-e は複数検索のときに使う.

5. `I want to jump`
   main 関数のアドレスが毎回変わってしまうが main 関数と win 関数のアドレスの差は変化していないので (何回も nc 接続したらわかる), アドレス差さえわかれば win 関数のアドレスがわかるということだ.
6. **差を求める**
   さまざまなツールがあるが, とりあえず,10 進数に直して計算すると差が 150(10),0x96(16) だと分かった.
7. **入力**
   nc 接続をして main のアドレスが表示されるので, そこから-0x96(16) のアドレスを求め, 入力すると flag が出てくる

4 hashcrack[Cryptography]

4.1 問題文

A company stored a secret message on a server which got breached due to the admin using weakly hashed passwords. Can you gain access to the secret stored within the server? Access the server using nc verbal-sleep.picoctf.net 57356

4.2 解法

1. `hash` **をとりあえず検索**

```
$ nc verbal-sleep.picoctf.net 57356
Welcome!! Looking For the Secret?

We have identified a hash: 482c811da5d5b4bc6d497ffa98491e38
Enter the password for identified hash:
```

hash をとりあえず検索してみると

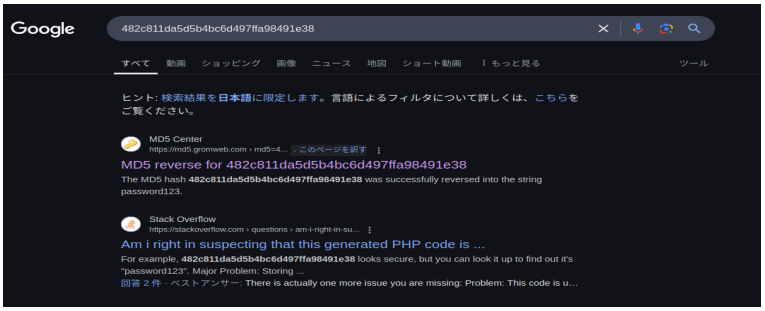

図 4　hash_serch

このように MD5 と書かれており, ついでに decode もできそうなのでやって, それを入力するとクリア

2. **次も同じように**
3. SHA256
   検索しても出ない人が多いかもしれません. しかし,SHA256 **デコード**と調べると変換ツールが出ると思うので, そちらで行いましょう

5 おわりに

ご覧いただきありがとうございました.