

Write up soal\_CTF\_XOR\_Bytes

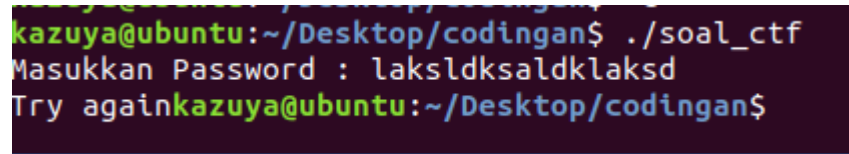
Disini yang pertama kita lakukan adalah melihat jenis filenya

file soal\_ctf

soal\_ctf: ELF 64-bit LSB shared object, x86-64, version 1 (SYSV), dynamically linked, interpreter /lib64/ld-linux-x86-64.so.2, for GNU/Linux 3.2.0, BuildID[sha1]=8cd294dc087d5532fd956aa23cf2f4504ee2caf3, not stripped

terlihat bahwa jenis file adalah elf 64 bit dan juga tidak di strip

kemudian kita tes dan coba jalankan program ini



```
kazuya@ubuntu:~/Desktop/codingan$ ./soal_ctf
Masukkan Password : laksldksaldklaksd
Try againkazuya@ubuntu:~/Desktop/codingan$
```

Terlihat bahwa program ini meminta inputan berupa password dan jika kita melihat di satu folder yang sama terlihat bahwa ada data txt yang bernama nano.txt kemudian setelah itu mari kita lihat isi dalam dari nano.txt

“ini adalah petunjuk kunci ini dienkripsi

c`afgdejhinolmrspqvwutuz{

ac!">{mw"qgg"og"kl"jgpqggg

JGNNM"UMPNF"

Disini ada beberapa karakter yang tidak jelas tapi menunjukkan bahwa ini adalah kuncinya

Langsung saja kita buka program ini untuk melihat apa yang dia lakukan

```

// local variable allocation has failed, the output may be wrong:
int __cdecl main(int argc, const char **argv, const char **envp)
{
    char s; // [rsp+10h] [rbp-30h]
    unsigned __int64 v5; // [rsp+38h] [rbp-8h]

    v5 = __readfsqword(0x28u);
    writefile();
    readfile((_QWORD *)&argc, argv);
    buka();
    strlen(&s);
    printf("Masukkan Password : ");
    __isoc99_scanf("%[^\n]", &s);
    if ( !strcmp(text3, &s) )
        printf("here is your flag = flag[%s]", text3);
    else
        printf("Try again", &s);
    getchar();
    return 0;
}

```

Dari pseudo code ini kita dapat melihat bahwa pertama program ini menjalankan function yang bernama write file, read file dan juga menjalankan function buka. Kemudian dia membandingkan antara text3 dan inputan yang kita masukkan

```

int writefile()
{
    FILE *s; // ST08_8

    xor();
    s = fopen("nano.txt", "w");
    fwrite("ini adalah petunjuk kunci ini dienkripsi\n", 1uLL, 0x29uLL, s);
    fprintf(s, "%s\n", text1);
    fprintf(s, "%s\n", text2);
    fputs(text3, s);
    return fclose(s);
}

```

Terlihat disini bahwa dia menuliskan global variable kedalam sebuah nano.txt tapi sebelum itu ada xor()

Function yang dapat kita lihat disini,

```
int64 xor()
{
    int64 result; // rax
    int i; // [rsp+Ch] [rbp-14h]
    unsigned int j; // [rsp+10h] [rbp-10h]
    int v3; // [rsp+18h] [rbp-8h]
    signed int v4; // [rsp+1Ch] [rbp-4h]

    v3 = strlen(text1);
    v4 = strlen(text3);
    for ( i = 0; i < v3; ++i )
    {
        text1[i] ^= 2u;
        text2[i] ^= 2u;
    }
    for ( j = 0; ; ++j )
    {
        result = j;
        if ( (signed int)j >= v4 )
            break;
        text3[j] ^= 2u;
    }
    return result;
}
```

Bahwa sebelum masuk kedalam nano.txt data tersebut di xor dengan angka senilai 2 namun dengan panjang string yang berbeda

Kemudian mari kita lihat function bukannya

```

size_t buka()
{
    size_t result; // rax
    int i; // [rsp+8h] [rbp-18h]
    int j; // [rsp+Ch] [rbp-14h]

    for ( i = 0; i < strlen(text1); ++i )
    {
        text1[i] ^= 2u;
        text2[i] ^= 2u;
    }
    for ( j = 0; ; ++j )
    {
        result = strlen(text3);
        if ( j >= result )
            break;
        text3[j] ^= 2u;
    }
    return result;
}

```

Disini terlihat jelas bahwa ketika process data sudah masuk kembali kedalam Main, program ini membuka encryption xornya sehingga dapat diketahui bahwa string yang dibandingkan didalam sini adalah string yang sudah dibuka encripsi XORnya kemudian untuk menyelesaikan problem ini maka saya memutuskan untuk membuat sebuah script dalam C untuk melihat apa isi dari text tersebut :

Ini scripnya

```

#include<stdio.h>
#include<string.h>

int main(){

//ini adalah petunjuk kunci ini dienkripsi

    char string1[100] = {"c`afgdejhinoImrspqvtuz{"};
    char string2[100] = {"acl\"{mw\"qgg\"og\"kl\"jgpggg"};
    char string3[100] = {"JGNNM\"UMPNF"};
    char string[100];
    for(int i = 0 ; i < strlen(string1) ; i++){
        string1[i] ^= 2;
    }
    for(int i = 0 ; i < strlen(string2) ; i++){
        string2[i] ^= 2;
    }
}

```

```
for(int i = 0 ; i < strlen(string3) ; i++){  
    string3[i] ^= 2;  
}  
printf("%s\n", string1);  
printf("%s\n", string2);  
printf("%s\n", string3);  
  
getchar();  
return 0;
```

```
}
```

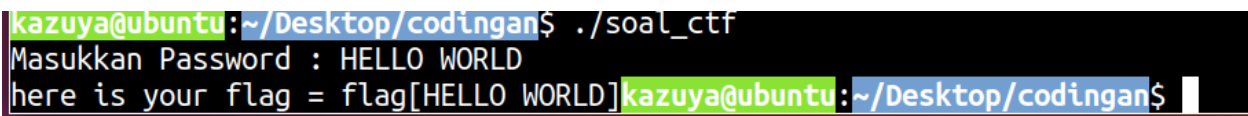
Kemudian kita akan mendapatkan output

abcdefghijklmnopqrstuvwxy

can you see me in hereee

HELLO WORLD

Mari kita coba satu-satu



```
kazuya@ubuntu:~/Desktop/codingan$ ./soal_ctf  
Masukkan Password : HELLO WORLD  
here is your flag = flag[HELLO WORLD]kazuya@ubuntu:~/Desktop/codingan$
```

Flag[HELLO WORLD]