MoeCTF 2021

逆向工程入门指北， Copyright @ 2021 含树InnerSpace， XDSEC

逆向工程

当我们写代码是，首先写出来`test.c`，然后编译生成了`test.exe`(windows平台)。我们运行`test.exe`便可以做到代码中写的语句。

但是如果我只给你一个编译好的`test.exe`而不给你源代码，你怎么知道`test.exe`这个可执行文件中包含了什么代码语句。

这时，就需要逆向工程了。

逆向工程就是解读可执行文件（`test.exe`）中代码的逻辑，对其进行修改或者破解，来达到自己目的的一种手段。

说白了，破解软件大家应该都用过吧，比如我们在52破解上下载的破解软件，在3dm上下载的免费的单机游戏，甚至打游戏开挂（写外挂的人），这些都需要用到逆向工程。

Begin of CTF中的逆向

ctf中的逆向（`reverse`）涉及很多编程语言，最常见的是C、C++ 、python、java、C#，或者一些不太常见的编程语言golang、rust、lisp等等。

除了windows与linux系统下可执行文件的逆向，还经常遇到安卓的安装包（`apk`）逆向、系统驱动、固件逆向等等。

但是作为一个`beginner`，你不需要在刚入门的时候就去掌握这么多的编程语言

从C语言以及C语言逆向开始，是一个比较好的选择

因为C语言十分接近高级语言，又有充分的系统底层的支持

同时入门的逆向题很多是用C语言写的

并且当你学会了C语言，尽管你无法使用java、python等语言进行编程，但是你会发现，看懂这些高级语言写的代码还是比较容易的。

接下来引用rx大哥说过的一句话

> 如果上面说了这么多都无法打动你的话, 那么接下来的话你听好了: C语言是大一上学期必修课, 并且在你
> 整个大学过程中都阴魂不散, 这下你该去学了吧?

学了一点C语言之后

可以做一点简单的逆向题来练手了。这里介绍几个工具

查壳工具： exeinfope, die（这两个都挺好用的）

逆向工具：ida

入门的话，C语言逆向，用这三个就够了（其实exeinfope和die二选一就行，但是有的壳die查不出来，所以难题的话习惯两个都查一下）

壳的话一般是指加密壳&压缩壳，是一种保护措施，可以加大逆向难度

一般的做题步骤

1.先查壳

先用查壳工具看一下二进制文件是否加壳，同时还可以查看平台以及系统位数

2.拖进ida

查壳之后就可以拖进对应系统位数的ida了（32 \ 64），会弹出几个框框，一般来说都点OK就可以了

ida分析完之后，你第一眼看到的应该是解析出来的汇编代码，这是就要用到ida最强大的一个功能了，按下F5，可以把汇编代码转换成类似C语言的代码（伪C代码）

接下来你就可以照着伪C代码进行逆向了。

逆向题一般的套路

通常来说，你运行一道逆向题的附件，它会让你输入一个字符串（flag），然后程序会验证你的flag是否正确。并告诉你验证结果，这就写相当于密码验证了。

如果让你来写一个密码验证器，你会怎么写?

```
strcmp(input,"this_is_flag");
```

这样写的话，你的密码验证器拖进ida，按下F5，就可以很轻松的得到密码了，这样的验证器是十分不安全的，因此我们需要进行加密。如下:

```
char password_enc[] =
{176,184,178,190,169,187,166,148,130,177,180,182,184,130,187,188,169,184,242,
174,169,188,164,130,179,180,186,181,169,160};
for (int i = 0; i < 30; i++) {
    if (input[i] ^ 0xdd != password_enc[i]) {
        printf("Wrong!\n");
        exit(0);
    }
}
printf("Cong!\n");
```

这样的话，密码的明文就经过了一步异或加密，比之前的一句验证要好一点

所以当我们写逆向题的时候:

1. 找出被加密的数据
2. 识别加密算法
3. 写出对应的解密算法，得到flag

```
//因为a^b=c时，b^c=a，所以根据这种性质，我们可以写出解密算法：
char password_enc[] =
{176,184,178,190,169,187,166,148,130,177,180,182,184,130,187,188,169,184,242,
174,169,188,164,130,179,180,186,181,169,160};
for (int i = 0; i < 30; i++) {
    password_enc[i] ^= 0xdd;
}
puts(password_enc);
//meoctf{I_like_fate/stay_night}
```

但并非所有逆向题都是这样的

只能说上面提到的一种是很具有代表性的，但是逆向题目中的逻辑是五花八门的，随着你学习的深入，你会遇到各种各样的题目，比如更加复杂的加密算法（其中最基础的有base64、tea、RC4等等），或者一些反逆向手段，如加壳、花指令、指令虚拟化、虚拟机、反调试等等。

善于使用搜索引擎，这会是你的网安学习之路上的一个重要的技能

加油，祝你也能找的逆向工程的乐趣

本题flag：`moectf{Wooooooooo_YOu_kNow_the_key_Of_Reverse!!!}`