pwn零基础入门

原创

A0ko 于 2020-03-24 19:39:38 发布

pwn零基础入门

很多初学者在初学pwn的时候，对pwn可能存在许多疑问，学什么，怎么学，今天和大家分享一下刚学pwn的一些资料的学习历程。

首先分享一下pwn的一系列学习资料

一、首先语言基础得打牢，1、汇编语言，《汇编语言—王爽》2、编程语言（C语言和python）C语言必须要会，题目大部分是用C语言写的，而且很多那些知识都要读libc的源代码。会python，exp用python写的，CTF中pwn主要是用pwntools这个python库来写利用脚本,要熟悉这个库pwntools库文件 pwntools库常见用法

二、计算机组成与原理：pwn入门首先从栈溢出开始，首先了解栈的构造、函数的调用约定、函数参数的在栈上的分布，然后很多知识点要刷，就不断的刷题吧，，，，，好像也没什么捷径

学习平台：CTFwiki上面有许多教程和例题，wiki平台 、 i春秋、看雪

pwn入门视频可以关注一下B栈上的一个up主 (君莫笑hhhhhhhh)，里面讲了许多入门题

三、刷题网站

1、bugku入门网站

2、XCTF3、BUUCTF等等

四、软件插件安装

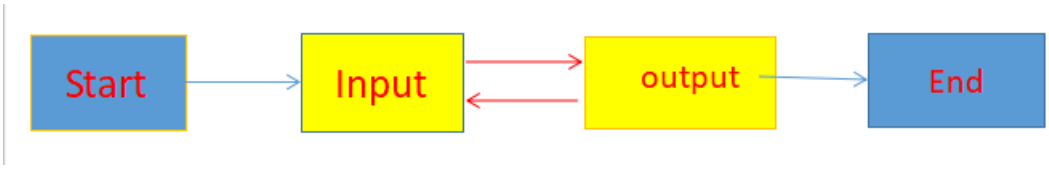
虚拟机以及VMtools、ubuntu（以及插件gdb、pwntools、peda、pwndbg等）IDA（32位和64位）这几个是必须要有的。其余的可以按自己的情况安装ubuntu的学习也可以看一下《鸟哥的私房菜》这本书，讲的还是比较好

what is pwn?

简单的来说就是控制程式流程，近而且达到攻击的效果。也就是一个挖掘漏洞、利用漏洞的过程

那么怎么来达到这个目的呢？

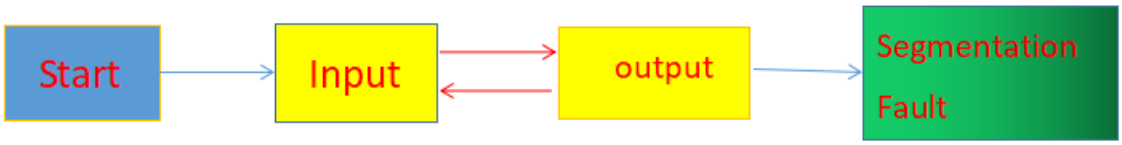
How to pwn?



input：使用者输入、操作。

output：包含运算和输出。

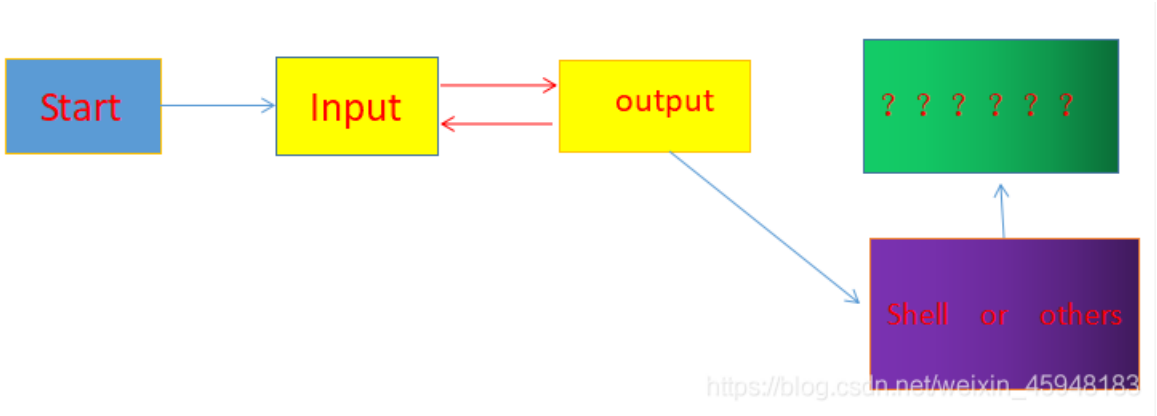
下一步寻找漏洞进一步利用：



Fuzz模糊测试：自主生成随意输入来寻找漏洞

看原始代码、用组合语言寻找漏洞

寻找漏洞，进一步利用



在end的地方，我们就可以做一些事情，一般情况下获取权限拿到shell
这么说可能缩小了它的范围，但是刚接触pwn知道大概就是这么个意思就行了，一般情况下，常见的又栈溢出漏洞，堆漏洞，格式化字符串漏洞。一些情况下还会存在整数溢出，首先了解计算机内各段

bss段：用来存放程序中未初始化的全局变量的一块内存区域。静态内存分配。

data段：数据段用来存放程序中已初始化的全局变量的一块内存区域。数据段静态内存分配。

text段：代码段用来存放程序执行代码的一块内存区域。内存区域通常属于只读(某些架构也允许代码段为可写，即允许修改程序)。

堆（heap）：低位往高位长，用于存放进程运行中被动态分配的内存段，大小不固定，可动态扩张或缩减。调用malloc、free等函数分配、释放内存

栈(stack)：高位往低位长，用户存放程序临时创建的局部变量，函数被调用时，其参数也会被压入发起调用的进程栈中，并且待到调用结束后，函数的返回值也会被存放回栈中。
一个程序本质上都是由 bss段、data段、text段三个组成的。

PLT和GOT表

外部函数的内存地址存储在 GOT，PLT 存储的入口点又指向 GOT 的对应条目，为什么选择 PLT 而非 GOT 作为调用的入口点？为了程序的运行效率。GOT 表的初始值都指向 PLT 表对应条目中的某个片段，这个片段的作用是调用一个函数地址解析函数。当程序需要调用某个外部函数时，首先到 PLT 表内寻找对应的入口点，跳转到 GOT 表中。如果这是第一次调用这个函数，程序会通过 GOT 表再次跳转回 PLT 表，运行地址解析程序来确定函数的确切地址，并用其覆盖掉 GOT 表的初始值，之后再执行函数调用。当再次调用这个函数时，程序仍然首先通过 PLT 表跳转到 GOT 表，此时 GOT 表已经存有获取函数的内存地址，所以会直接跳转到函数所在地址执行函数。第一次调用函数时解析函数地址并存入 GOT 表

linux pwn常见的保护机制

》RELRO
部分RELRO:
将.got段映射为只读(但.got.plt还是可以写)
重新排列各个段来减少全局变量溢出导致覆盖代码段的可能性.
完全RELRO：
执行部分RELRO的所有操作.
让链接器在链接期间(执行程序之前)解析所有的符号, 然后去除.got的写权限.
将.got.plt合并到.got段中, 所以.got.plt将不复存在.
》.stack canary
在函数返回值之前添加的一串随机数（不超过机器字长），末位为/x00（提供了覆盖最后一字节输出泄露canary的可能）
》.NX
no executable，标识页表是否可执行
》.PIE
elf文件加载基址随机化，Linux系统的aslr可通过

链接: link.