CTF[Pwn] i春秋学习笔记

转载

weixin_30505751 于 2018-03-17 22:43:00 发布 205 收藏
文章标签： 数据结构与算法
原文链接：http://www.cnblogs.com/rookieDanny/p/8593266.html

buffer overflow

> 堆溢出、栈溢出、bss溢出、data溢出

> 例题：wellpwn\AliCtf 2016 vss\Hitcon 2015 readable\stkof\zerostorage

整数溢出

- 无符号数和有符号的转换(MIMA 2016 shadow)
- 整数加减乘除法，malloc(size*2)(pwnhbu.cn calc)
- 整数溢出通常会进一步转换为缓冲区溢出、逻辑漏洞等其他漏洞

格式化字符串

- printf sprintf fprintf 任意地址读写
- 用来leak

释放后使用(Use-After-Free)

- 释放掉的内存可能会被重新分配，释放后使用会导致重新分配的内存被旧的使用所改写
- Double free就是一种特殊的UAF
- 例题: Defcon 2014 Qual shitsco,AliCTF 2016 router,0CTF2016 freenote(double free),HCTF 2016 fheap(double free)

逻辑漏洞

- 访问控制
- 协议漏洞
- 多线程竞态条件等（fakefuzz）

还原结构体、接口、类

理清程序的执行逻辑

熟悉常见的数据结构 链表、树、堆、图等各种加密算法

了解各个寄存器的作用

rsp/esp pc rbp/ebp rax/eax rdi rsi rdx rcx

栈用于保存函数的调用信息和局部变量

函数调用:call,rer

调用约定：

__stdcall,__cdecl,__fastcall,__thiscall,__nakecall,__pasccal

参数传递:取决于调用约定，默认如下

- X86从右向左入栈
- X64优先寄存器，参数过多才入栈

函数调用的过程

call func： push pc,jmp func

Leave : mov esp,ebp ,pop ebp

Ret ：pop pc

栈溢出的保护机制

栈上的数据无法被当成指令来执行

- 数据执行保护(NX/DEP)
- 绕过方法:ROP

让攻击者难以找到shellcode地址 #不知道返回地址

- 地址空间布局随机化(ASLR)
- 绕过方法: infolleak,ret2dlresolve,ROP

检测Stack Overflow

- Stack Canary/Cookie(Cookie泄露)
- 绕过方法 infoleak

栈溢出的利用方法

现代栈溢出的技术基础:ROP(绕过NX和ASLR)

利用signal机制的ROP技术：SROP

没有binary怎么办: BROP(不是很常见)

劫持栈指针 :stack pivot #将栈指针劫持到其他区域

利用动态链接绕过ASLR:ret2dlresolve、fake linkmap

利用地址低12bit绕过ASLR：Partial Overwrite（低12位不会被随机化）

绕过Stack canary：改写指针与局部白能量、leak canary、overwrite canary

溢出位数不够怎么办：覆盖ebp,Partial Overwrite

现代栈溢出的利用技术ROP 代码复用技术

google关键字 Ret2libc,ROP

CTF中ROP常规套路

- 第一次触发漏洞，通过ROP泄露libc的address(如puts_got)，计算system地址 然后返回到一个可以重现触发漏洞的地方（main）再次触发漏洞。通过ROP调用system(‘/bin/sh’)
- 直接execve(“/bin/sh”,[“/bin/sh”],NULL) 这个通常在静态链接比较常用

例题：Defcon 2015 Qual:R0pbaby AliCTF 2016:vss PlaidCTF 2013:ropasaurusrex

转载于:https://www.cnblogs.com/rookieDanny/p/8593266.html