武汉大学计算机学院

2015-2016 学年第二学期

《 软件安全 》 考试试卷（A 卷，开卷）

（注：所有解答必须写在答题纸上，写在试卷上的无效）

一、 计算题（每题 5 分，共 20 分）

1、以下是某硬盘的分区表信息，计算所有分区大小。（按 1K=1000 计算，小数点后取 1 位，四舍五入，给出计算过程）

```
00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 01
C1 FF 07 FE FF FF 3F 00  00 00 FC 8E 33 02 00 FE
FF FF 05 FE FF FF 80 29  54 02 80 29 54 02 00 00
00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00
00 00 00 00 00 00 00 00  00 00 00 00 00 00 55 AA
```

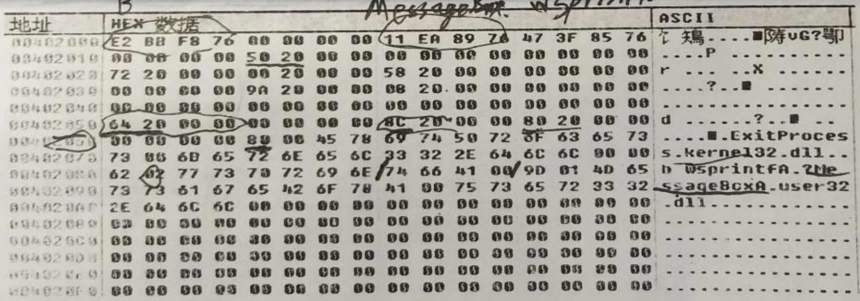
2. 下图为某程序的.rdata 节（开始位置 RVA：2000，文件偏移量：800H）在内存中的主要数据。试分析计算 MessageBox 函数的真实地址，wsprintfA 函数的真实地址，该文件 800H-803H 偏移处的值，文件 808H-80BH 偏移处的值。（给出计算思路和结果）

2-84

208C

5

MessageBox wsprintfA



```
地址      HEX 数据                                          ASCII
00402000  E2 BB F8 76 00 00 00 00 11 EA 89 76 47 3F 85 76  饣 鳺....■隋vG?鄍
00402010  00 00 00 00 50 20 00 00 00 00 00 00 00 00 00 00  ....P ..........
00402020  72 20 00 00 00 20 00 00 58 20 00 00 00 00 00 00  r ... ..X ......
00402030  00 00 00 00 9A 20 00 00 08 20 00 00 00 00 00 00  ....?..■ ......
00402040  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
00402050  64 20 00 00 00 00 00 00 8C 20 00 00 80 20 00 00  d ......?..■ ..
00402060  00 00 00 00 80 00 45 78 69 74 50 72 6F 63 65 73  ....■.ExitProces
00402070  73 00 6B 65 72 6E 65 6C 33 32 2E 64 6C 6C 00 00  s.kernel32.dll..
00402080  62 02 77 73 70 72 69 6E 74 66 41 00 9D 01 4D 65  b WsprintfA.?Me
00402090  73 73 61 67 65 42 6F 78 41 00 75 73 65 72 33 32  ssageBoxA.user32
004020A0  2E 64 6C 6C 00 00 00 00 00 00 00 00 00 00 00 00  .dll............
004020B0  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
004020C0  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
004020D0  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
004020E0  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
004020F0  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
```

3、画出调用 printf 函数的栈帧结构，并给出该函数的打印结果。

```
int main(void)
{  int i=1,j=2,k=3;
   char buf[]="test";
   printf("%s %d %d %d\n",buf,i,j);
return 0;}
```

12.

4、FAT32 中，一个 8KB 目录空间通常可以包含多少个目录项？当该目录空间占满时，如何扩充该目录空间？

二、 简单题（每题 5 分，共 35 分）

1、一般 USB 采用 FAT32 文件格式，其存储的一个 JPG 照被无意删除，试给出手动恢复的原理和过程。

2、木马的文件管理与资源管理器的文件管理有什么异同？

3、计算机病毒的感染与黑客攻击的 Shellcode 注入有什么异同？

4、PE 结构中重定位节的结构，以及重定位的作用。

5、在部署 DEP 时，会对哪些软件带来哪些兼容性问题？

6、Rootkit 和安全工具一般采用内联钩子（Inline Hooking），试给出对 MessageBox 函数的内联钩子的实现机理（以图表描述）。

7、Fuzzing 对哪些缺陷挖掘有效？该挖掘方法存在哪些局限性？

三、 缺陷分析题（每题 5 分，共 25 分）

指出下列代码中的安全缺陷、造成的危害以及简单的修补方式。

```
void handleConnection ( int socket ) {
  （1）  char user[100] ;char pass[200] ;char buff[400] ;
  （2）  int c = 0 ;
  （3）  strncpy (buff , "USER: " , 100 ) ;
  （4）  send ( socket , buff , 7 , 0 ) ;
  （5）  recv ( socket , buff , 400 , 0 ) ;
  （6）  strncpy ( user , buff , 100 ) ;
  （7）  snprintf ( buff , 400 , " Hello %s \ nPASS :
         " ,user ) ;
  （8）  c = strlen (buff ) + 1 ;
  （9）  send (socket , buff , c , 0 ) ;
  （10） recv ( socket , buff , 400 , 0 ) ;
  （11） strcpy ( pass , buff ) ;
  （12） strncpy ( buff , "Logged in " , 100 ) ;
  （13） send ( socket , buff , 23 , 0 ) ;
}
```

实例 1 字符串处理

```
Select * from article where
articleID=$id;
```

实例 2 SQL 查询

```
1.  nresp=packet_get_int();
2.  If(nresp>0){
3.  Response=xmalloc(nresp*sizeof(char*));
4.  For(i=0;i<nresp;i++)
5.  response[i]=packet_get_string(NULL));
6.  }
```

实例 3 动态内存分配

```
1.  Def      storepassword(username,
    password):
2.  Hasher=hashlib.new(“md5”)
3.  Hasher.update(password)
4.  hashedPassword=hasher.digest()
5.  return updateuserlogin(username,
    hashedPassword)
```

实例 4 口令更新

```
int main(int argc, char** argv)
{printf(argv[1]);return 0;}
```

实例 5 基本输出

四、　综合设计题（每题 10 分，共 20 分）

1、栈溢出攻击经常会覆盖栈中的返回地址，/gs 安全机制通过在函数入口往栈中压入一个随机值-cookie，函数出口检验该随机值，从而防御栈溢出。

1.1 试设计一种对栈中返回地址备份的安全防御机制。（8 分）

1.2 与/gs 相比，试分析该机制的优点与不足。（2 分）

2、如果下列代码存在缺陷，试设计一个 ROP 的 shellcode，该 Shellcode 弹出一个 shell（system(“/bin/sh”)）.

2.1 指出该代码存在的安全缺陷。（2 分）

2.2 设计该 Shellcode 在栈中的结构布局。（6 分）

2.3 如何检测这种 ROP 的 Shellcode？（2 分）

```
#define PASSWORD 1234567
int main(int argc, char ** argv)
{
      int authenticated;
      char buffer[8];
      authenticated=strcmp(argv[1],PASSWORD);
      if (authenticated)
          printf("ok!");
      Else
      {
         strcpy(buffer,~~password~~ argv[1]);
         printf("Fail!,%s",buffer);
      }
      return authenticated;
}
```