《计算机系统》
深入理解黑客攻击

湖南大学

《计算机系统》课程教学组

内容提要

01 SQL注入攻击

02 直接修改可执行文件

03 改变程序返回地址

04 缓冲区溢出攻击

05 键盘记录器

SQL注入攻击

http://hnulab.tech:28080/phplogin.html

```
Select *
From USER
Where 用户名='admin'
And 密码='123456'
```

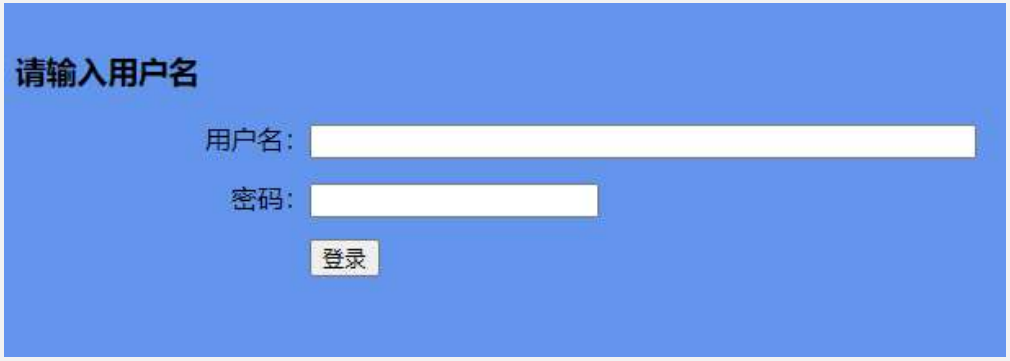
USER表

用户名	密码
admin	123456
ykh	54321
……	……

```
for (i=0;i<n;i++)
{
if (user[i].用户名== 'admin'
&& user[i].密码=='123456')
        return 1;
}
return 0;
```

利用此处单引号，编写绕过密码验证的字符串

SQL注入攻击




```
String SqlString="Select * From USER Where 用户名='" + Username.text()+"'
And 密码='" +Password.Text()+"'";
int i=SQLExecute(SqlString);
if(o==i)
{
        Alert("用户名或密码错误！"）:
        return;
}
else
        response.direct("index.aspx");
```

SQL注入攻击

http://hnulab.tech:28080/phplogin.html

USER表

用户名	密码
admin	123456
ykh	54321
……	……

```
Select *
From USER
Where 用户名='ad' or 1=1; --
And 密码='abcde'
```

-- 之后都为注释，无法执行

```
for (i=0;i<n;i++)
{
 if (user[i].用户名== 'ad'||'1'=='1'; -- )
 return 1;
}
return 0;
```

-- 之后都为注释，无法执行

内容提要

01 SQL注入攻击

02 修改可执行文件

03 改变程序返回地址

04 缓冲区溢出攻击

05 键盘记录器

修改可执行文件

1030.c

```c
#include <stdio.h>
static char* s1="Sorry,您可能是盗版用户的受害者!\n";
static char* s2="Welcome,欢迎进入系统!\n";
int main()
{
        int i;
        printf("请输入序列号：");
        scanf("%d",&i);
        if(i!=16)
        {
                printf("%s",s1);
                return 0;
        }
        printf("%s",s2);
        return 1;

}
```

```
0804846d <main>:
 804846d:       55                      push   %ebp
 804846e:       89 e5                   mov    %esp,%ebp
 8048470:       83 e4 f0                and    $0xfffffff0,%esp
 8048473:       83 ec 20                sub    $0x20,%esp
 8048476:       c7 04 24 ba 85 04 08    movl   $0x80485ba,(%esp)
 804847d:       e8 ae fe ff ff          call   8048330 <printf@plt>
 8048482:       8d 44 24 1c             lea    0x1c(%esp),%eax
 8048486:       89 44 24 04             mov    %eax,0x4(%esp)
 804848a:       c7 04 24 d0 85 04 08    movl   $0x80485d0,(%esp)
 8048491:       e8 ca fe ff ff          call   8048360 <__isoc99_scanf@plt>
 8048496:       8b 44 24 1c             mov    0x1c(%esp),%eax
 804849a:       83 f8 10                cmp    $0x10,%eax
 804849d:       74 1c                   je     80484bb <main+0x4e>
 804849f:       a1 24 a0 04 08          mov    0x804a024,%eax
 80484a4:       89 44 24 04             mov    %eax,0x4(%esp)
 80484a8:       c7 04 24 d3 85 04 08    movl   $0x80485d3,(%esp)
 80484af:       e8 7c fe ff ff          call   8048330 <printf@plt>
 80484b4:       b8 00 00 00 00          mov    $0x0,%eax
 80484b9:       eb 1a                   jmp    80484d5 <main+0x68>
 80484bb:       a1 28 a0 04 08          mov    0x804a028,%eax
```

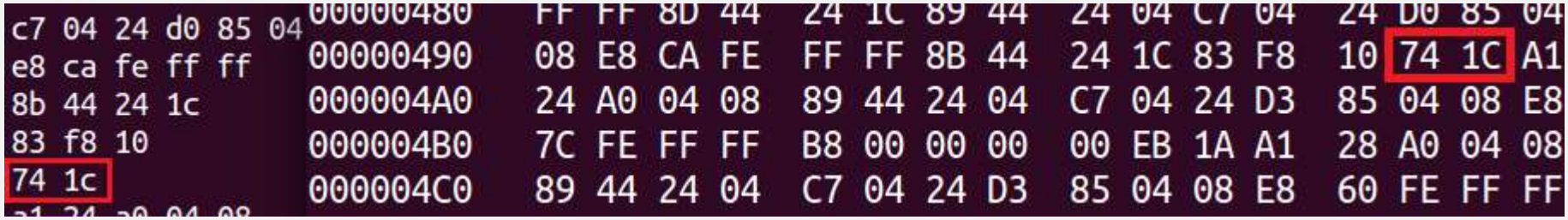
修改可执行文件

使用 hexedit 来修改可执行文件 1030.out:

ctrl+s 向下查找； ctrl+r 向上查找；ctrl+x退出。

je 指令 74 1C → jne指令 75 1C

颠倒判断逻辑，使得密码不符时反而允许通过



内容提要

01 SQL注入攻击

02 修改可执行文件

03 改变程序返回地址

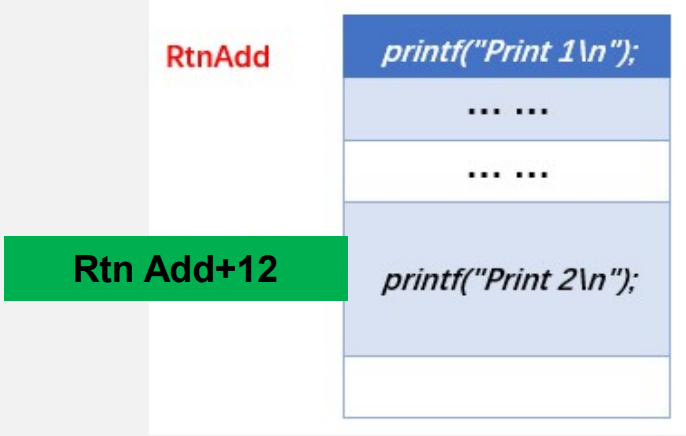
04 缓冲区溢出攻击

05 键盘记录器

修改返回地址

1031.c

```
void foo()
{
    int a,*p;
    p=(int*)((int)&a+12);
    *p+=12;
}
int main()
{
    foo();
    printf("Print 1\n");
    printf("Print 2\n");
    printf("Print 3\n");
    printf("Print 4\n");
    return 0;
}
```



修改返回地址

gamble.c

利用修改返回地址来破解博彩游戏

计算机犯罪

◆计算机犯罪行为会构成**破坏计算机信息系统功能罪**。它是指违反国家规定，对计算机信息系统功能进行**删除、修改、增加、干扰，造成计算机信息系统不能正常运行**，后果严重的行为。

◆根据《刑法》第286条第1款规定，犯本罪基本罪的，**处5年以下有期徒刑或者拘役**；犯重罪的，**处5年以上有期徒刑**。

内容提要

01 SQL注入攻击

02 修改可执行文件

03 改变程序返回地址

04 缓冲区溢出攻击

05 键盘记录器

缓冲区溢出攻击

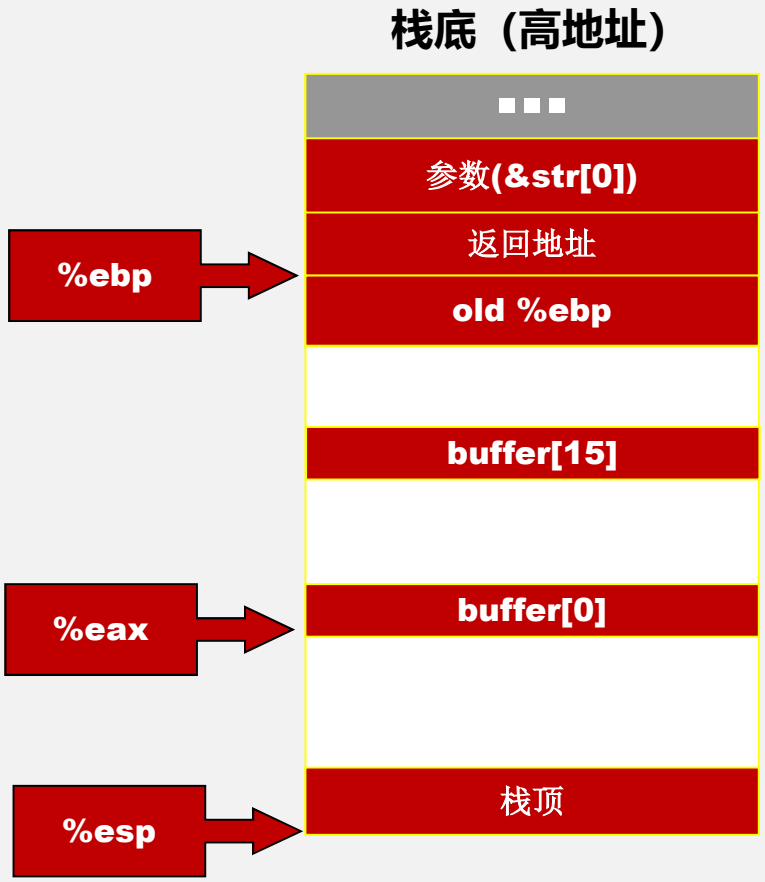
缓冲区溢出是一种非常普遍、非常危险的漏洞，在各种操作系统、应用软件中广泛存在。**缓冲区溢出攻击**是利用缓冲区溢出漏洞所进行的攻击行动。缓冲区溢出攻击可以导致程序运行失败、系统关机、重新启动等后果。如果有人恶意利用在栈中分配的缓冲区的写溢出，悄悄地将一个恶意代码段的首地址作为“返回地址”覆盖地写到原先正确的返回地址处，那么，程序就会在执行 ret 指令时悄悄地转到这个恶意代码段执行，从而可以轻易取得系统特权，进而进行各种非法操作。

造成缓冲区溢出的原因是程序没有对栈中作为缓冲区的数组进行越界检查。下面用一个简单的例子说明攻击者如何利用缓冲区溢出跳转到自己设定的程序 hacker 去执行。

以下是在文件 test.c 中的三个函数，假定编译、链接后的可执行代码为 test。

test.c

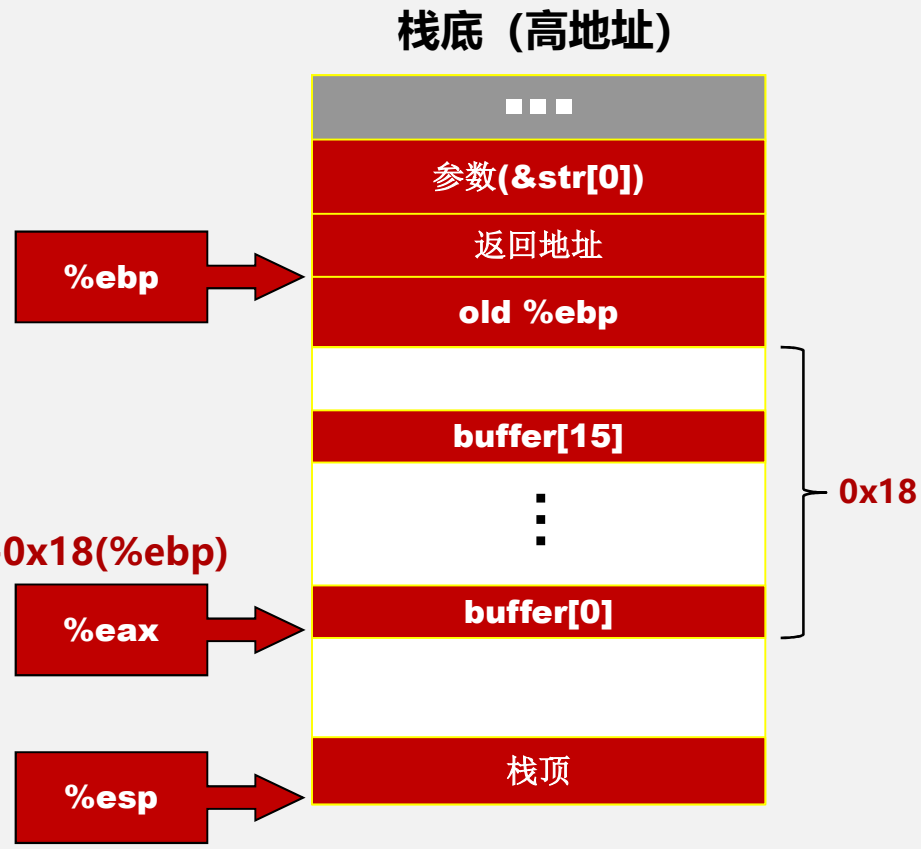
```c
#include <stdio.h>
#include "string.h"

void outputs(char *str)
{
        char buffer[16];
        strcpy(buffer,str);//str to buffer
        printf("%s \n",buffer);
}

void hacker(void)
{
        printf("being hacked\n");
}

int main(int argc,char *argv[])
{
        outputs("I love HNU");
        /*outputs("12345671234567123456712345671234567\xaa\x84\x04\x08");*/
        return 0;
}
```



test.c

```
push   %ebp
mov    %esp,%ebp
sub    $0x28,%esp
mov    0x8(%ebp),%eax
mov    %eax,0x4(%esp)
lea    -0x18(%ebp),%eax
mov    %eax,(%esp)
call   8048340 <strcpy@plt>
lea    -0x18(%ebp),%eax
mov    %eax,0x4(%esp)
movl   $0x8048570,(%esp)
call   8048330 <printf@plt>
leave
ret
```

test.c

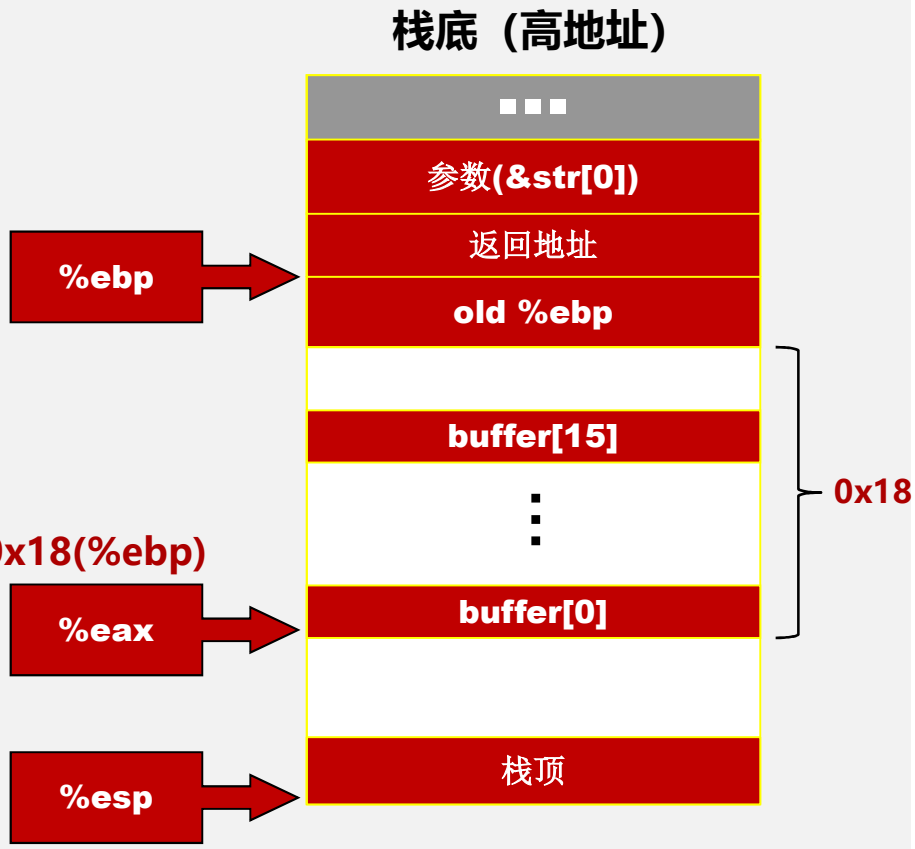


为防止GCC编译时打开缓冲区溢出保护使演示无法完成，使用如下命令编译:

gcc –fno-stack-protector –z execstack –o test.out test.c

test.c

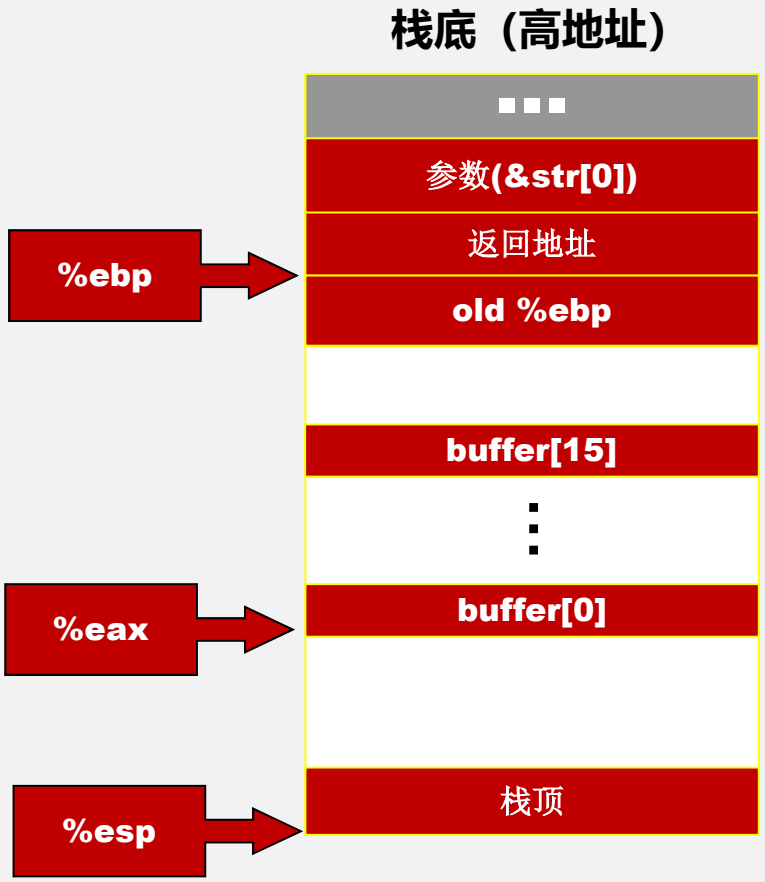
- buffer首地址到 old ebp，一共24字节（0x18），如果恶意增加buffer长度，buffer就会溢出。
- 如果恶意增加 buffer长度到28字节，则会覆盖掉old ebp；如果增加到32字节，则会覆盖掉返回地址，这些都是缓冲区溢出攻击的后果。
- 如果buffer增加到32字节，同时使得最后4字节（即覆盖掉返回地址的那4个字节）为某段恶意代码的首地址，则outputs函数执行完毕后就会跳转到恶意代码处执行，即黑客攻击。



test.c

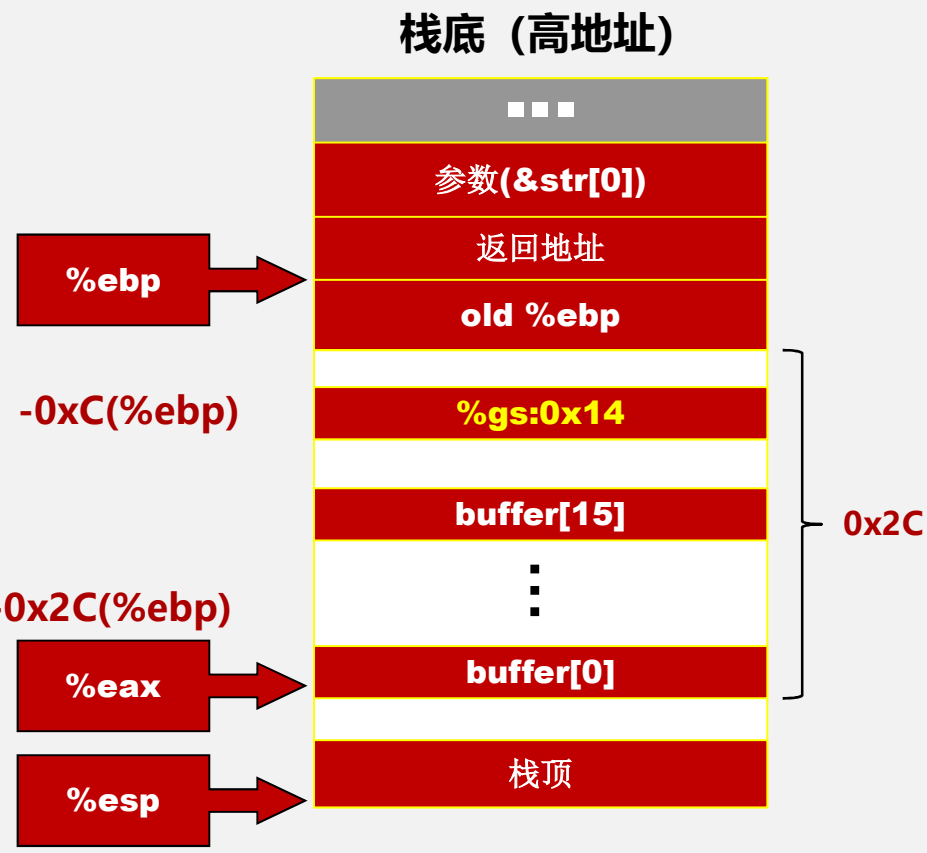
使用4组“1234567”以及最后的hacker首地址组成32字节的溢出攻击方式，实现从 outputs 到 hacker 的跳转。

```
#include <stdio.h>
#include "string.h"

void outputs(char *str)
{
        char buffer[16];
        strcpy(buffer,str);//str to buffer
        printf("%s \n",buffer);
}

void hacker(void)
{
        printf("being hacked\n");
}

int main(int argc,char *argv[])
{
        /*outputs( I love HNU );*/
        outputs("12345671234567123456712345671234567\xaa\x84\x04\x08");
        return 0;
}
```



test.c

```
push   %ebp
mov    %esp,%ebp
sub    $0x38,%esp
mov    0x8(%ebp),%eax
mov    %eax,-0x2c(%ebp)
mov    %gs:0x14,%eax
mov    %eax,-0xc(%ebp)
xor    %eax,%eax
mov    -0x2c(%ebp),%eax
mov    %eax,0x4(%esp)
lea    -0x1c(%ebp),%eax
mov    %eax,(%esp)
call   8048390 <strcpy@plt>
lea    -0x1c(%ebp),%eax
mov    %eax,0x4(%esp)
movl   $0x80485e0,(%esp)
call   8048370 <printf@plt>
mov    -0xc(%ebp),%eax
xor    %gs:0x14,%eax
je     804851a <outputs+0x4d>
call   8048380 <__stack_chk_fail@plt>
leave
ret
```



如果关闭栈保护，直接编译，缓冲区中插入了指令参数“%GS:0x14”（金丝雀—来自煤矿瓦斯探测，p181）

内容提要

01 SQL注入攻击

02 修改可执行文件

03 改变程序返回地址

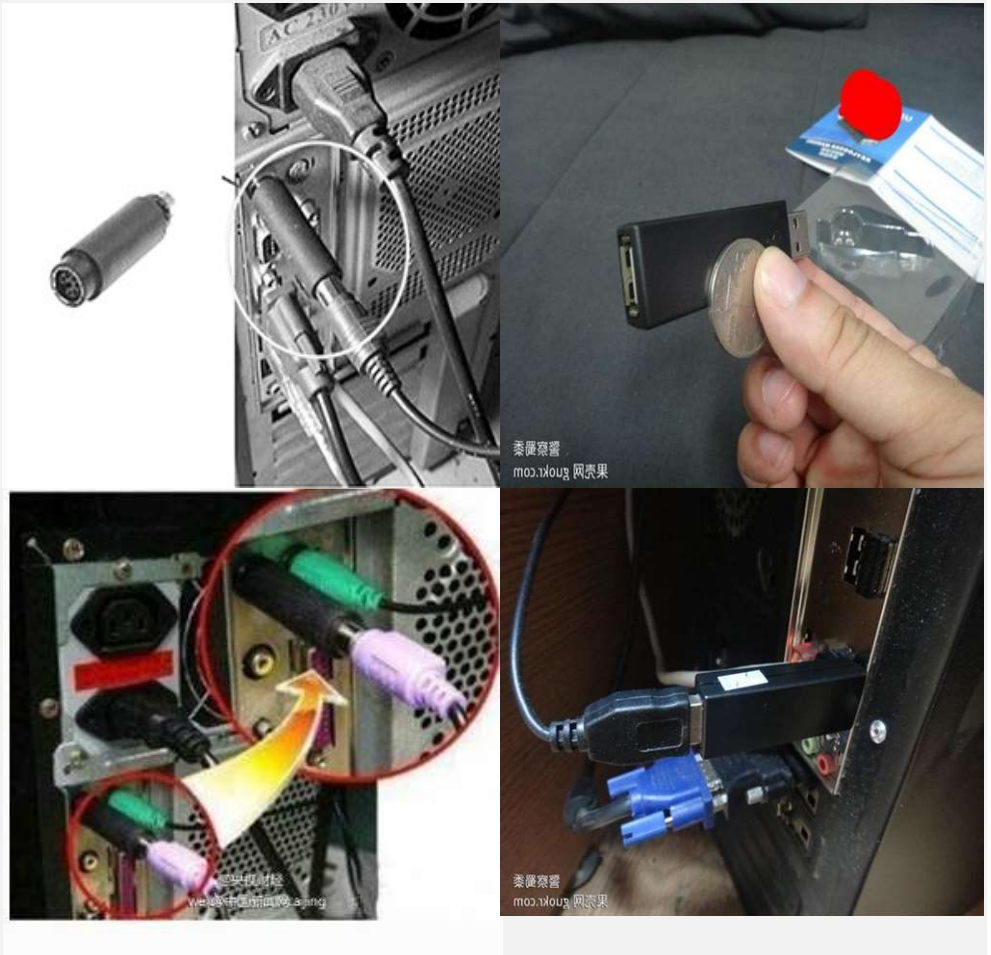
04 缓冲区溢出攻击

05 键盘记录器

键盘记录器

键盘记录器

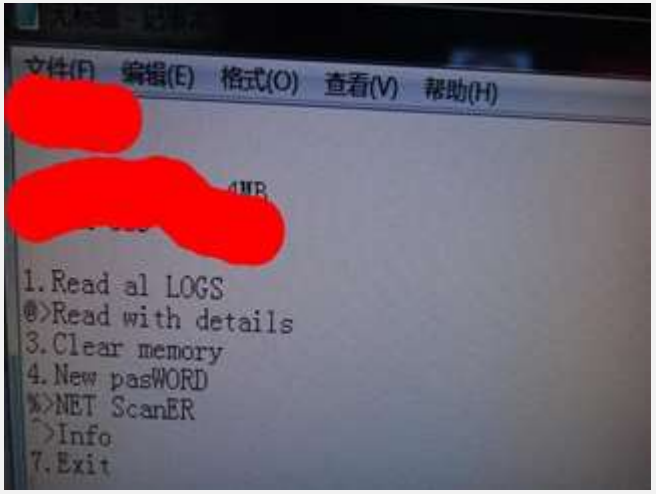
- 系统没有任何新硬件提示，安全软件也没用任何反应，也没用新u盘提示。
- 计算机内看不到任何新插入存储设备。



键盘记录器

只要打开记事本，然后输入一组说明书上的密码，回车，

然后就会出现使用菜单……



小结

- 黑客攻击的经典手法
  - SQL注入攻击
  - 修改可执行文件
  - 修改返回地址
  - 缓冲区溢出攻击
- 身为专业人士应当具备的职业道德与职业责任
- 掌握黑客防范手段，为国家建设保驾护航

中期学习建议

- 认真学习课堂讲授的原理
- 深刻理解课堂原理在汇编中的表达
- 熟练掌握使用汇编来实现所学原理的方法
- 对所学原理构建直观的模型和示意图来帮助记忆理解

下一节：数据通路概述

湖南大学

《计算机系统》课程教学组