华中科技大学

HUAZHONG UNIVERSITY OF SCIENCE AND TECHNOLOGY

网络空间安全学院

其他漏洞类型

网络空间安全学院 慕冬亮

Email : dzm91@hust.edu.cn

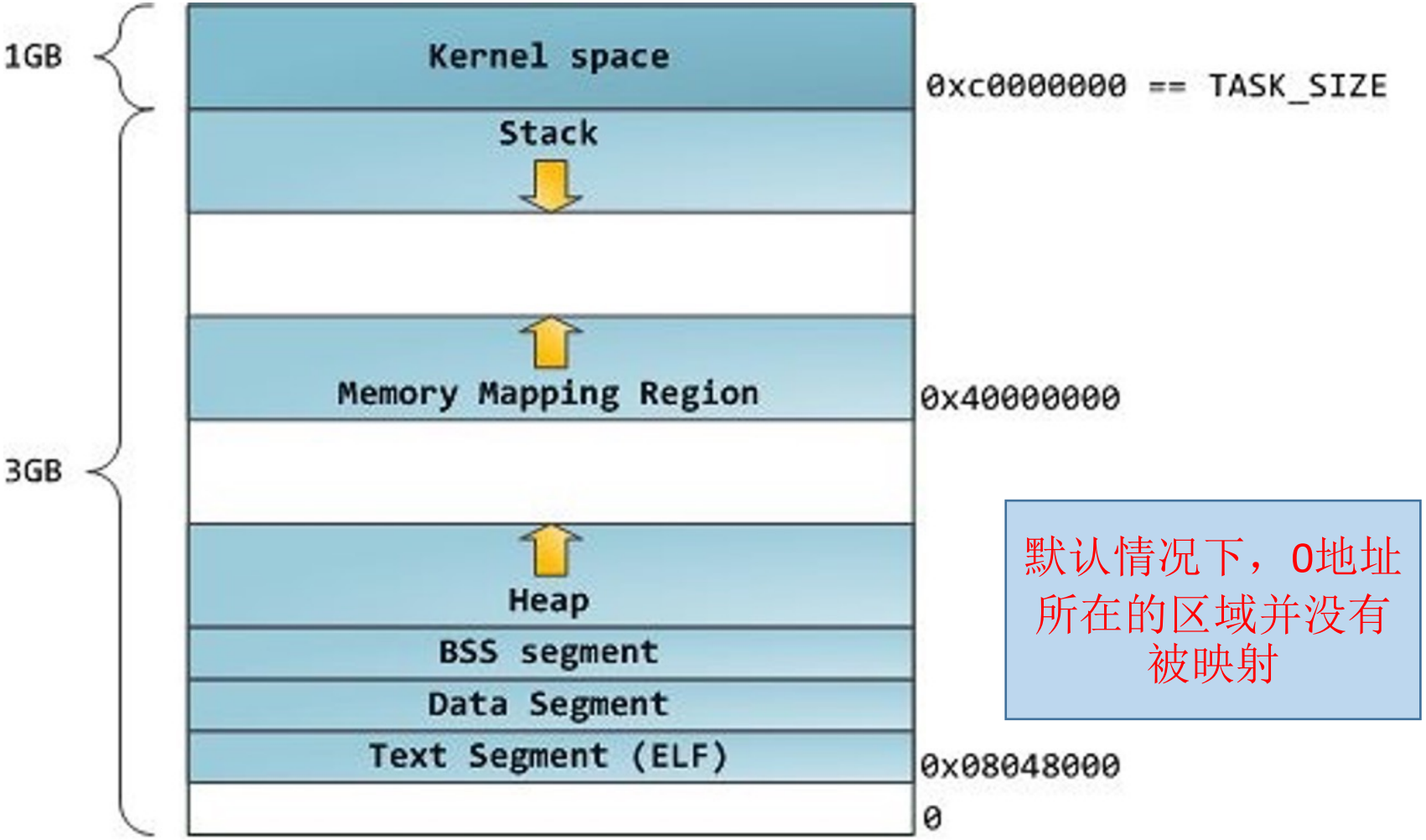
空指针引用漏洞

- 英文： Null Pointer Dereference

- char *p = NULL; // 数据指针
- *p

- void (*p)(int) = NULL; // 函数指针
- p(0);

空指针引用漏洞



默认情况下，0地址所在的区域并没有被映射

格式化字符串漏洞

- 可以进行信息泄露的格式化
  - “%s %d %x …”
- 产生原因
  - *printf()是不定参数输入
  - *printf()不会检查输入参数的个数

*可以是空，s,f,sn,v,vf,vs,w等等。

printf 栈结构图

int printf(const char *format, ...)

格式化字符串漏洞

- 关键受攻击的格式化
  - “%n”
  - 简单来说，%n 是将当前 printf 已打印的字符数写入一个 指定的变量
- 产生原因
  - *printf() 是不定参数输入
  - *printf() 不会检查输入参数的个数

*可以是空，s,f,sn,v,vf,vs,w等等。

格式化字符串漏洞

- 查看任意内存地址内容

```
int main()
{
    char text[1024];
    printf("input your string:");
    scanf("%s",text);
    printf(text);
    getchar();
    return 1;
}
```

```
int main()
{
    char text[1024];
    printf("input your string:");
    scanf("%s",text);
    printf(text);

    system("PAUSE");
    return EXIT_SUCCESS;
}
```

输入字符串“AAAA%08x. %08x. %08x. %08x. %08x. %08x. %08x. %08x. %08x. %08x. %08x. %08x. %08x. %08x.”

格式化字符串漏洞

- 修改返回地址

```
void functionown(int c)
{
    unsigned char Buf[2];
    printf("%74d%n",c,(int*)(Buf+6));
    printf("%19d%n",c,(int*)(Buf+7));
    printf("%64d%n",c,(int*)(Buf+8));
    printf("%n\r\n",c,(int*)(Buf+9));

}
int main(int argc, char* argv[])
{
    int a=1;
    printf("in a=%d\r\n",a);
    functionown(2);
    a=0;
    printf("out a=%d\r\n",a);
    return 0;
```

返回地址修改为 00 64 19 74所代表的地址!

防御措施

- 格式化串溢出通过静态扫描较容易发现

- 部分编译器已经可以限制部分格式化字符串问题。
  - -Wformat-security [1]

[1] https://clang.llvm.org/docs/DiagnosticsReference.html#wformat-security