专业特色选修课《网络信息安全》

缓冲区溢出和渗透测试

第一集

Buffer Overflow and Penetration Testing

嵩　天

songtian@bit.edu.cn

北京理工大学计算机学院


Wana DecryptOr 2.0
Ooops, your files have been encrypted!
Chinese (simpli
我的电脑出了什么问题？
您的一些重要文件被我加密保存了。
照片、图片、文档、压缩包、音频、视频文件、exe文件等，几乎所有类型的文件都被加密了，因此不能正常打开。
这和一般文件损坏有本质上的区别。您大可在网上找找恢复文件的方法，我敢保证，没有我们的解密服务，就算老天爷来了也不能恢复这些文档。
有没有恢复这些文档的方法？
当然有可恢复的方法。只能通过我们的解密服务才能恢复。我以人格担保，能够提供安全有效的恢复服务。
但这是收费的，也不能无限期的推迟。
请点击 <Decrypt> 按钮，就可以免费恢复一些文档。请您放心，我是绝不会骗你的。
但想要恢复全部文档，需要付款点费用。
是否随时都可以固定金额付款，就会恢复的吗，当然不是，推迟付款时间越长对你不利。
最好3天之内付款费用，过了三天费用就会翻倍。
还有，一个礼拜之内未付款，将会永远恢复不了。
对了，忘了告诉你，对半年以上没钱付款的穷人，会有活动免费恢复，能否轮
Payment will be raised on
5/16/2017 02:26:59
Time Left
02:22:35:15
Your files will be lost on
5/20/2017 02:26:59
Time Left
06:22:35:15
About bitcoin
How to buy bitcoins?
Contact Us
bitcoin
ACCEPTED HERE
Send $300 worth of bitcoin to this address:
13AM4VW2dhxYgXeQepoHkHSQuy6NgaEb94
Copy
Check Payment
Decrypt

国家信息安全漏洞共享平台(CNVD)

信息安全漏洞周报

2020年04月06日-2020年04月12日

- 本周共收集、整理安全漏洞477 个
- 超危漏洞 104个、高危漏洞 176 个
- 涉及0day漏洞115个 (占48%)
- 本周接到漏洞总数1411个。

本节大纲

- 缓冲区溢出概述
- 缓冲区溢出原理

缓冲区溢出概述

- 缓冲区

  包含相同数据类型实例的一个连续的计算机内存块 或 程序运行期间在内存中分配的一个连续的区域

- 溢出

  所填充的数据超出了原有的缓冲区边界

- 缓冲区溢出

缓冲区溢出概述

- 缓冲区溢出的历史

1988年，Morris蠕虫使fingerd程序溢出

1989年，Spafford提交了一份分析报告，描述了fingerd缓冲区溢出程序的技术细节，引起了重视

1996年，Aleph One发表了一篇文章，详细解释原理

***Smashing the stack for fun and profit*文章**

缓冲区溢出概述

- 缓冲区溢出的历史

2001年“红色代码”蠕虫利用微软IIS Web Server中的缓冲区溢出漏洞使300 000多台计算机受到攻击；

2003年1月，Slammer蠕虫爆发，利用的是微软SQL Server 2000中的缺陷；

2004年5月爆发的“振荡波”利用了Windows系统的活动目录服务缓冲区溢出漏洞；

缓冲区溢出概述

- 缓冲区溢出的历史

2017年5月，WannaCry、永恒之蓝、魔窟、蠕虫、勒索病毒、比特币、445端口、150个国家、NSA、几十万台电脑、15个比特币

缓冲区溢出概述

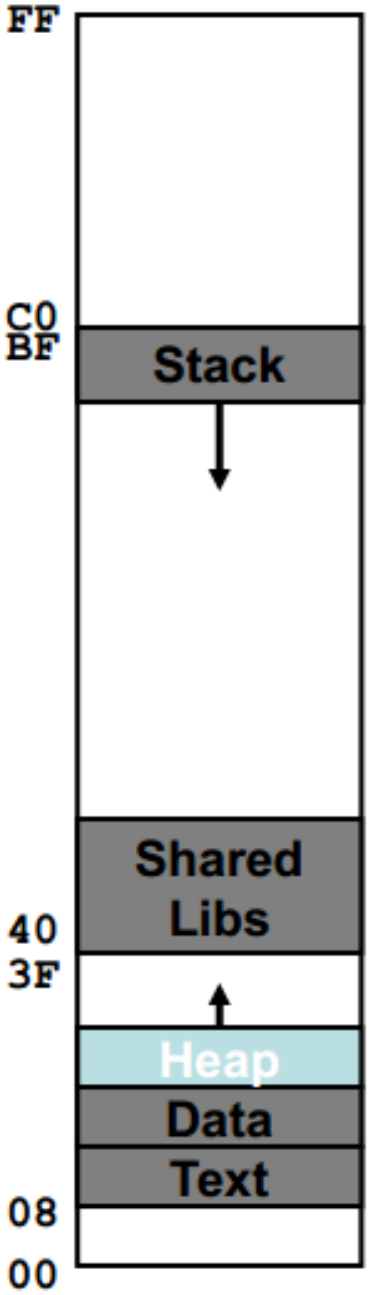
- 几个事实
- 缓冲区溢出已占所有系统攻击总数的80%以上
- 各种操作系统和应用软件上存在的缓冲区溢出问题数不胜数
- 可导致程序运行失败、系统崩溃、执行非授权指令、取得系统特权等后果

缓冲区溢出原理


内存高地址
命令行参数及环境变量
栈(动态)
堆(动态)
BSS段(静态)
数据段(静态)
代码段(只读)
堆栈区
数据区
代码区
低地址
程序在内存中的存放方式
栈
BSS: Block Started by Symbol
堆
未初始化全局或静态变量
初始化全局或静态变量
程序指令和只读数据

缓冲区溢出原理

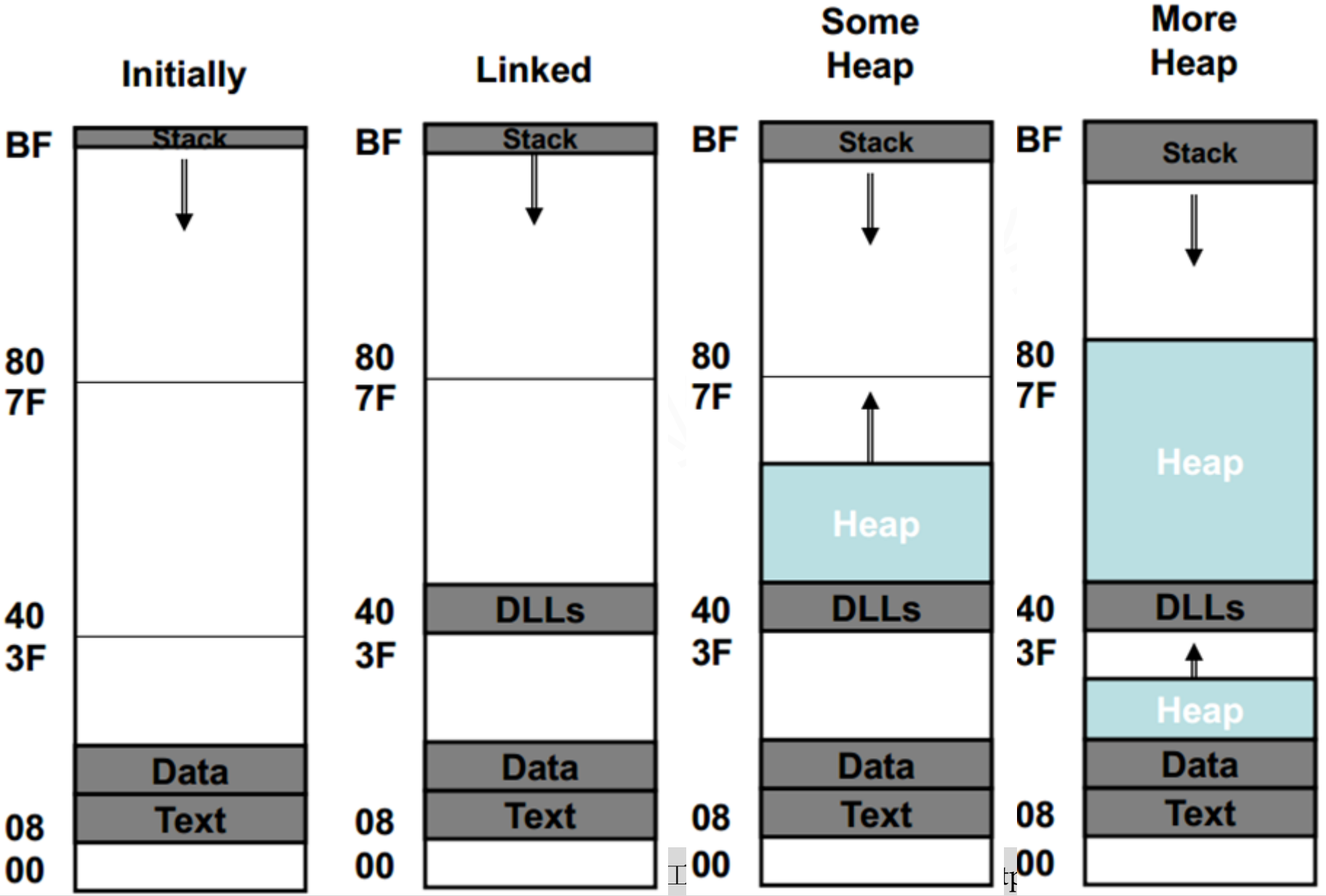
- 几个问题
  - 静态分配和动态分配的区别？
  - static 声明的变量放在那里？
  - BSS段和数据段中的变量在初始化上有何区别？
  - malloc()函数分配的内存空间在哪里？
  - 频繁malloc和free对内存空间造成什么影响？



Linux内存使用

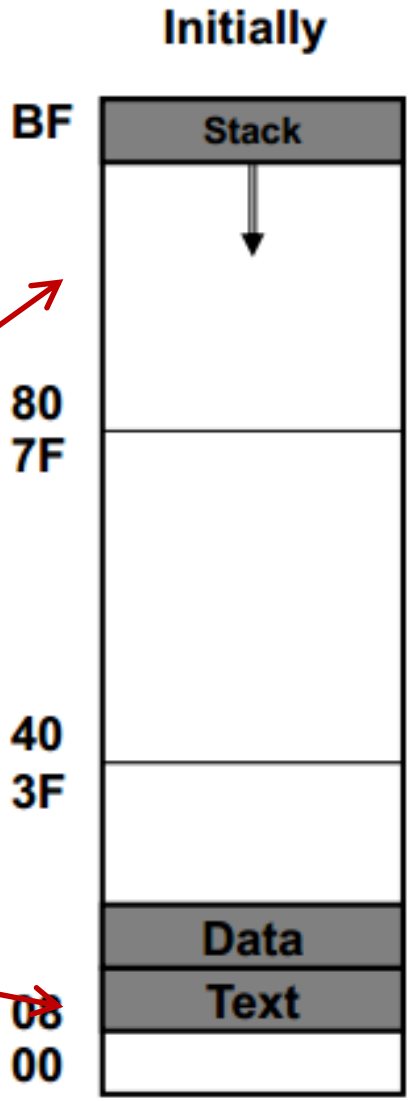
- 32bit系统
  - 4GB memory
  - 0-1GB: 用户空间 (text, code, malloc)
  - 1-3GB: 用户空间 (shared libs, stack)
  - 3-4GB: 内核空间

Linux内存使用


Initially
Linked
Some Heap
More Heap
BF
80
7F
40
3F
08
00
Stack
Heap
DLLs
Data
Text

基本程序实例

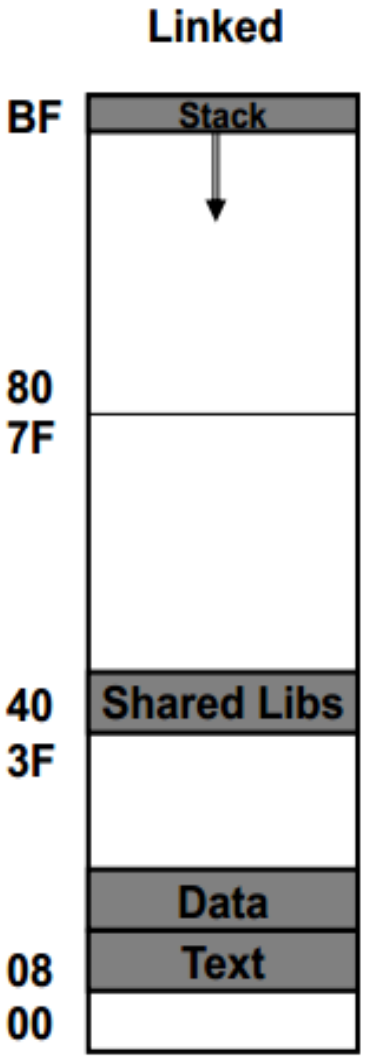
```
(gdb) break main
(gdb) run
  Breakpoint 1, 0x804856f in main ()
(gdb) print $esp
  $3 = (void *) 0xbffffc78
```



- 栈顶
  - 地址 0xbffffc78
- main()
  - 地址 0x804856f (0x0804856f)

动态链接实例

```
(gdb) print malloc
  $1 = {<text variable, no debug info>}
    0x8048454 <malloc>
(gdb) run
  Program exited normally.
(gdb) print malloc
  $2 = {void *(unsigned int)}
    0x40006240 <malloc>
```



- 程序运行前的malloc
  - 地址 0x8048454 (0x08048454)
- 程序运行后的malloc
  - 地址 0x40006240

内存分配实例

```
char big_array[1<<24];  /*  16 MB */
char huge_array[1<<28]; /* 256 MB */

int beyond;
char *p1, *p2, *p3, *p4;

int useless() {  return 0; }

int main()
{
 p1 = malloc(1 <<28);  /* 256 MB */
 p2 = malloc(1 << 8);  /* 256 B  */
 p3 = malloc(1 <<28);  /* 256 MB */
 p4 = malloc(1 << 8);  /* 256 B  */
 /* Some print statements ... */
}
```

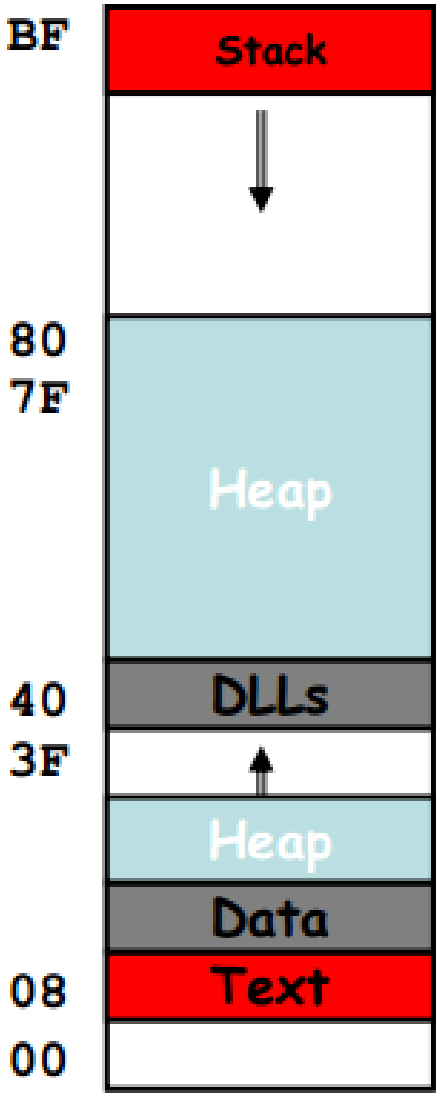
内存分配实例

```
char big_array[1<<24];  /*  16 MB */
char huge_array[1<<28]; /* 256 MB */

int beyond;
char *p1, *p2, *p3, *p4;

int useless() {  return 0; }
```

$esp	0xbffffc78
p3	0x500b5008
p1	0x400b4008
Final malloc	0x40006240
p4	0x1904a640
p2	0x1904a538
beyond	0x1904a524
big_array	0x1804a520
huge_array	0x0804a510
main()	0x0804856f
useless()	0x08048560
Initial malloc	0x08048454



缓冲区溢出原理

如果在堆栈中压入的数据超过预先给堆栈分配的容量时，就会出现堆栈溢出，从而使得程序运行失败；如果发生溢出的是大型程序还有可能会导致系统崩溃

系统中所有缓冲区都有溢出的可能性

缓冲区溢出原理

缓冲区溢出包括三种

栈溢出 （最常用）

堆溢出

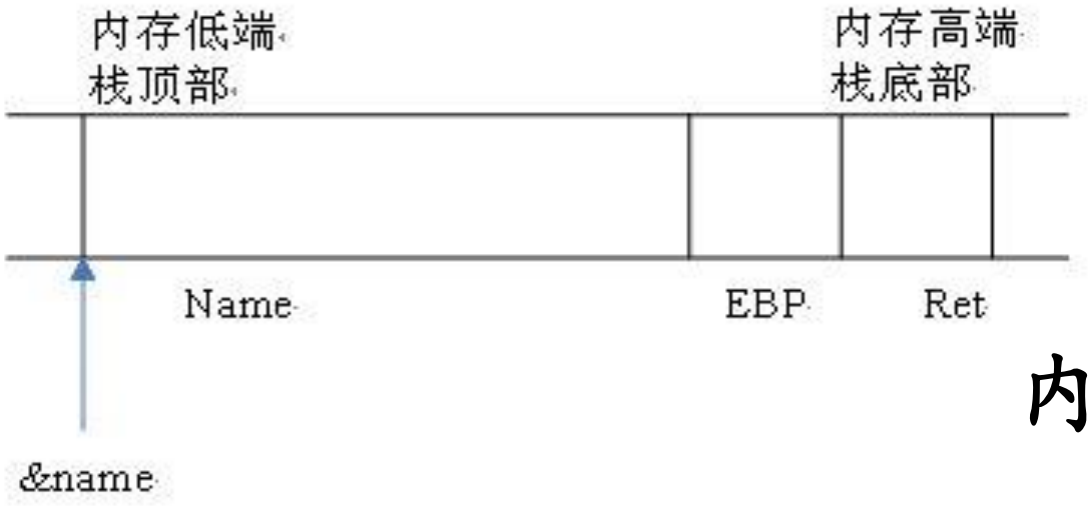
BSS溢出

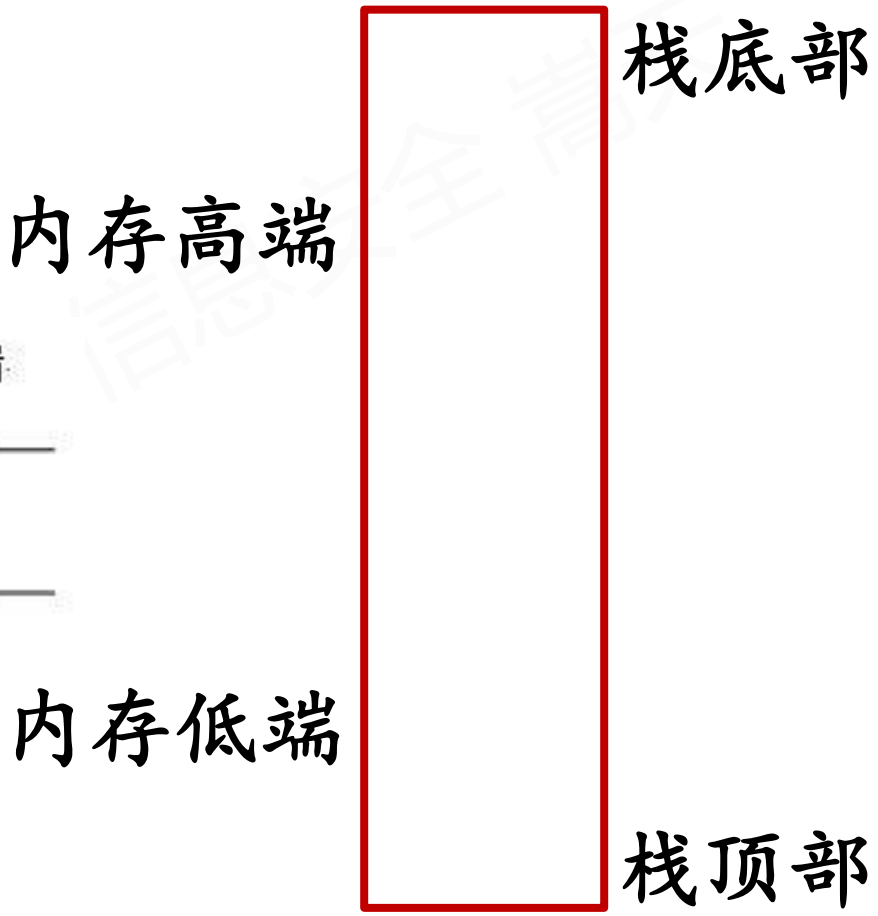
栈溢出实例

```
#include <stdio.h>
int main(){
      char name[16];
      gets(name);
      for(int i=0;i<16&&name[i];i++)
          printf("%c",name[i]);
}
```

栈溢出实例

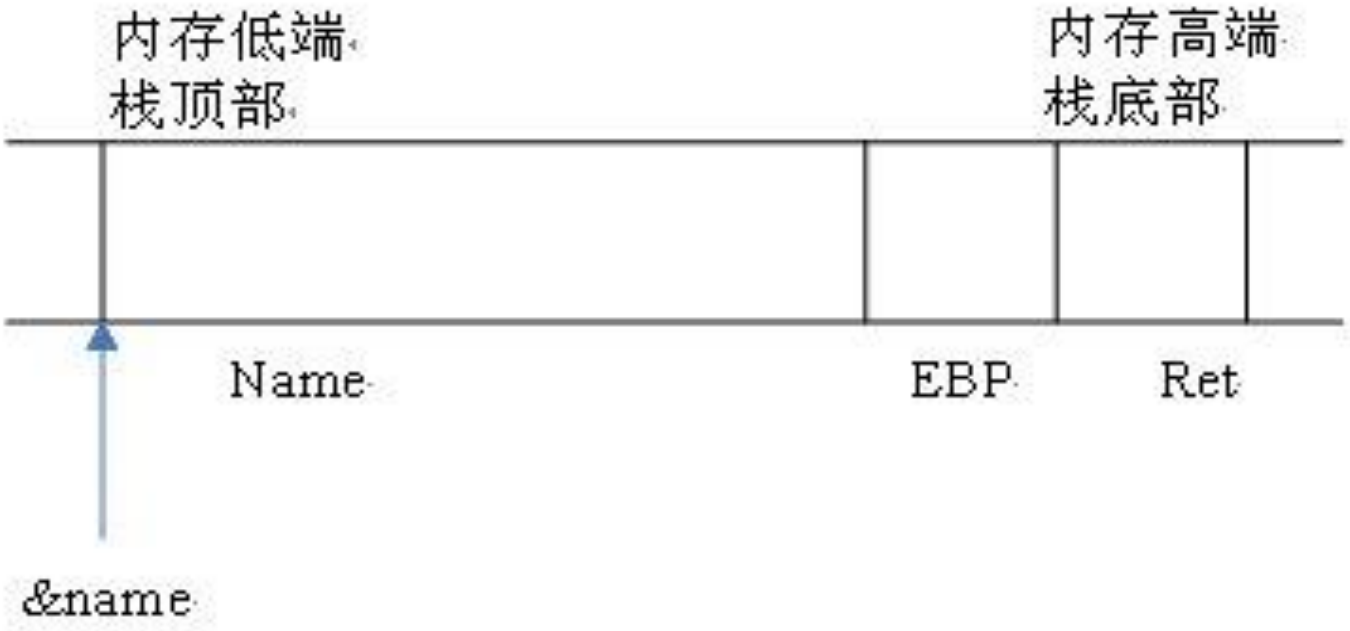
初始栈内存分布





栈溢出实例

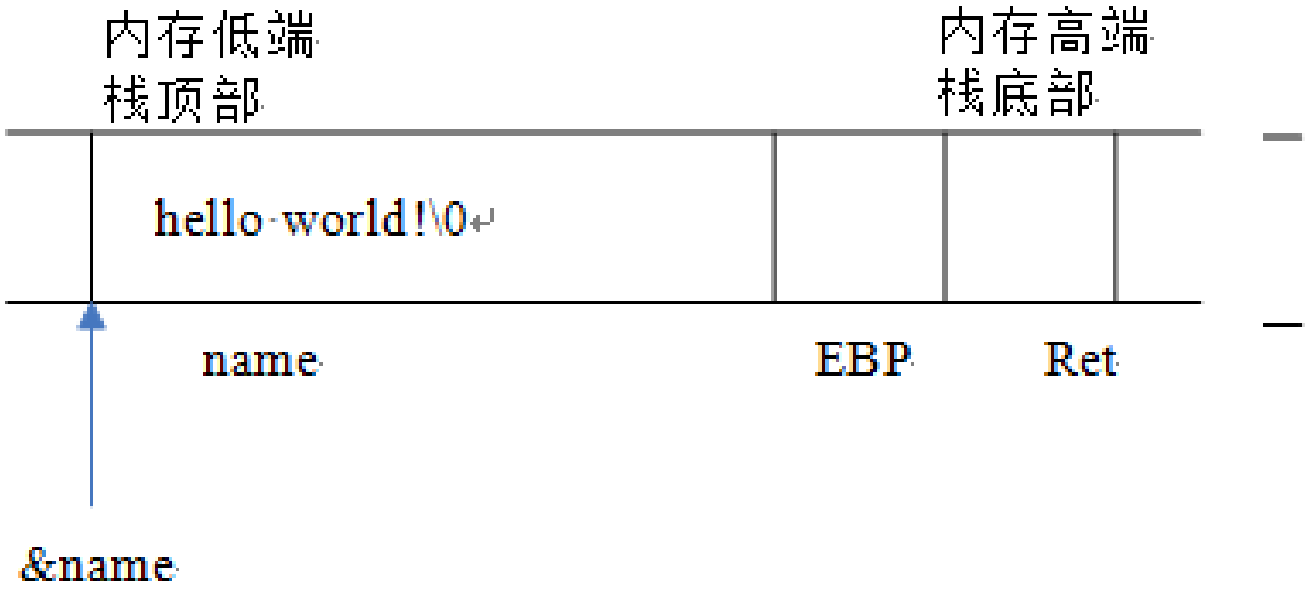
初始栈内存分布



栈溢出实例

执行gets(name)后的栈内存分布

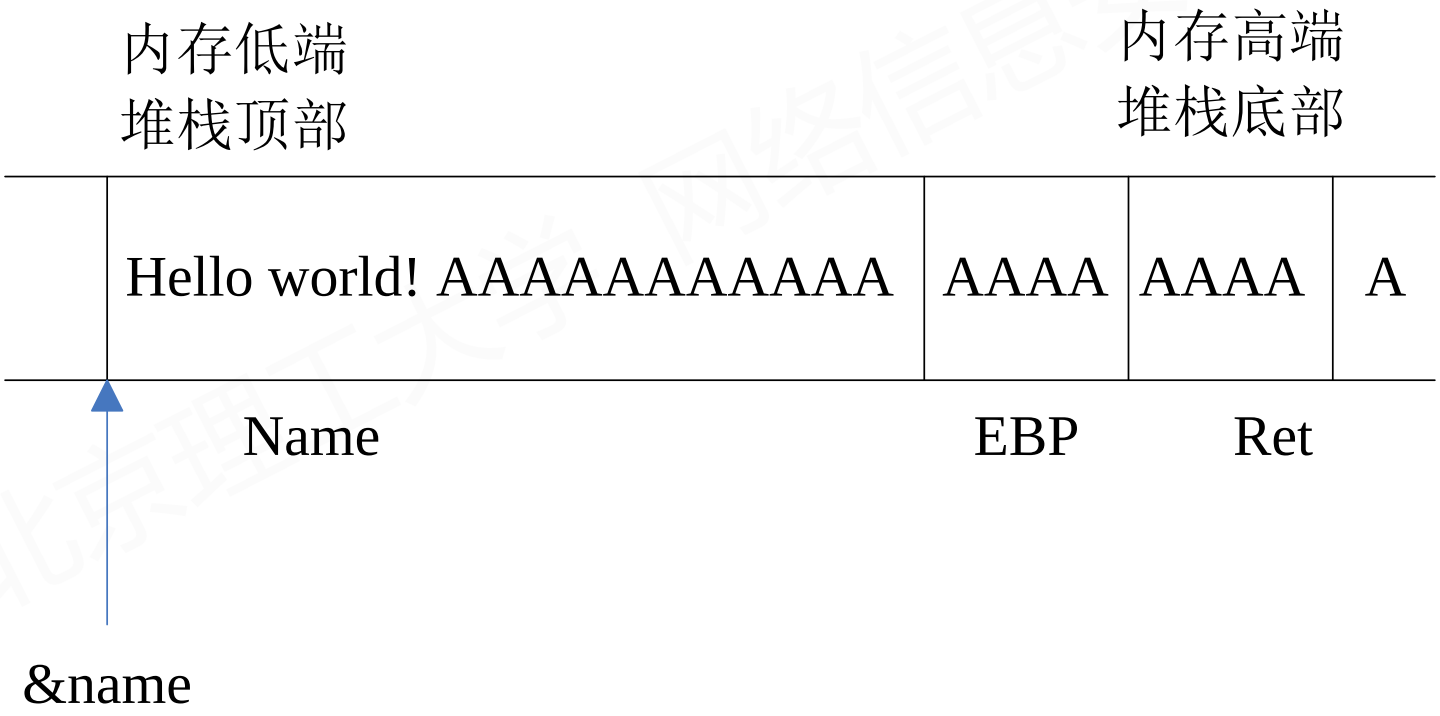
输入是“hello world!”



栈溢出实例

执行gets(name)后的栈内存分布

输入是“hello world! AAAAAA……”



Let's GO！！

课上实验

形式：完成实验报告

作业：完成一份报告

内容：结合Windows系统内存分配方式，对缓冲区溢出原理进行分析

要求：A4纸不超过2页，关注核心原理

提交：4月30日（周四）

作业需要标注：姓名、学号、email

发送到：zhran_bit@126.com

作业内容

内容：

1. 缓冲区溢出概念和基本原理描述

2. Windows系统逻辑内存结构简要说明

3. 调研并简述“exploit”的含义