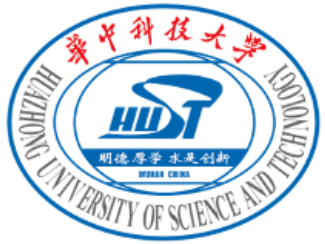


华中科技大学
HUAZHONG UNIVERSITY OF SCIENCE AND TECHNOLOGY

网络空间安全学院

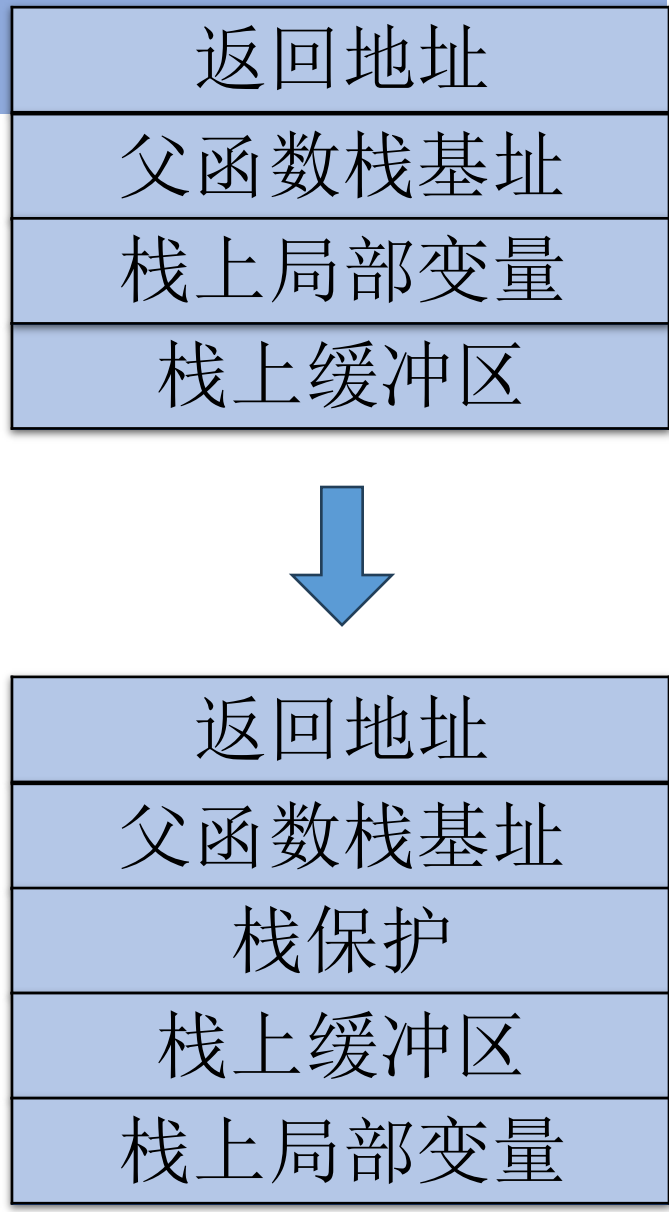
栈保护机制

网络空间安全学院 慕冬亮

Email : dzm91@hust.edu.cn

栈保护机制

- 栈溢出结果
  - 栈溢出会覆盖栈中的值，特别是返回地址
- 栈保护机制
  - Stack Protector / Stack Guard / Stack Canary
  - 在返回地址和父函数栈基址之前加入一个随机值
  - 栈帧变量的重排序
- 具体工作机制
  - 在函数开始时往栈中压入一个可以检验的随机数
  - 在函数结束时验证栈中的随机数是否一致

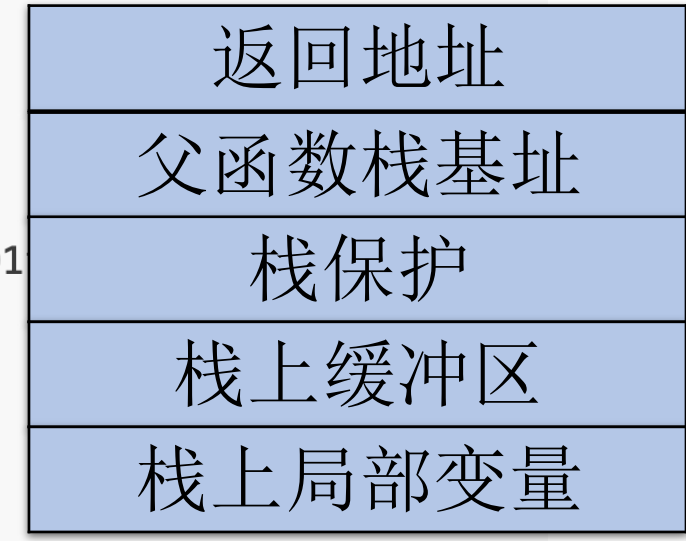


栈保护机制原理

```
0000000000401159 <vuln_func>:
  401159:       55                      push   rbp
  40115a:       48 89 e5                mov    rbp,rsp
  40115d:       48 83 ec 40             sub    rsp,0x40
  401161:       64 48 8b 04 25 28 00    mov    rax,QWORD PTR fs:0x28
  401168:       00 00
  40116a:       48 89 45 f8             mov    QWORD PTR [rbp-0x8],rax
  40116e:       31 c0                   xor    eax,eax
  401170:       c7 45 cc 03 00 00 00    mov    DWORD PTR [rbp-0x34],0x3
  401177:       48 8d 45 d0             lea    rax,[rbp-0x30]
  40117b:       48 89 c6                mov    rsi,rax
  40117e:       48 8d 3d 9a 0e 00 00    lea    rdi,[rip+0xe9a]        # 40201
  401185:       b8 00 00 00 00          mov    eax,0x0
  40118a:       e8 c1 fe ff ff          call   401050 <__isoc99_scanf@plt>
  40118f:       b8 00 00 00 00          mov    eax,0x0
  401194:       48 8b 55 f8             mov    rdx,QWORD PTR [rbp-0x8]
  401198:       64 48 33 14 25 28 00    xor    rdx,QWORD PTR fs:0x28
  40119f:       00 00
  4011a1:       74 05                   je     4011a8 <vuln_func+0x4f>
  4011a3:       e8 98 fe ff ff          call   401040 <__stack_chk_fail@plt>
  4011a8:       c9                      leave
  4011a9:       c3                      ret
```

插入随机数

vuln_func 栈帧结构

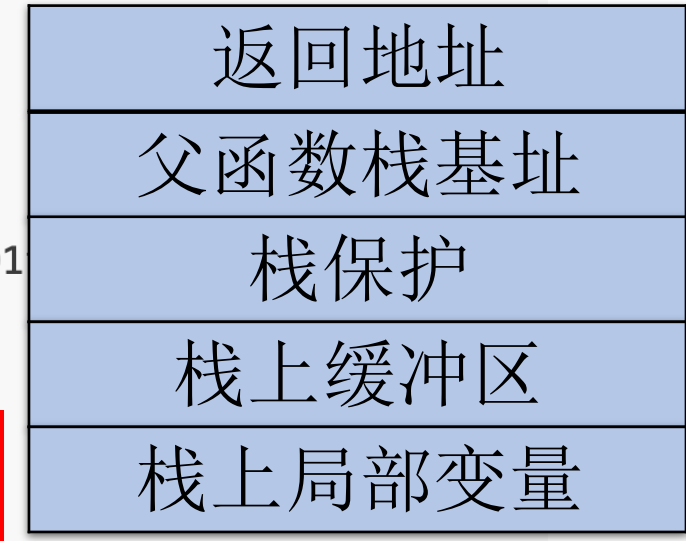


看 test_pwn_with_sp 实例

栈保护机制原理

```
0000000000401159 <vuln_func>:
  401159:       55                      push   rbp
  40115a:       48 89 e5                mov    rbp,rsp
  40115d:       48 83 ec 40             sub    rsp,0x40
  401161:       64 48 8b 04 25 28 00    mov    rax,QWORD PTR fs:0x28
  401168:       00 00
  40116a:       48 89 45 f8             mov    QWORD PTR [rbp-0x8],rax
  40116e:       31 c0                   xor    eax,eax
  401170:       c7 45 cc 03 00 00 00    mov    DWORD PTR [rbp-0x34],0x3
  401177:       48 8d 45 d0             lea    rax,[rbp-0x30]
  40117b:       48 89 c6                mov    rsi,rax
  40117e:       48 8d 3d 9a 0e 00 00    lea    rdi,[rip+0xe9a]        # 40201
  401185:       b8 00 00 00 00          mov    eax,0x0
  40118a:       e8 c1 fe ff ff          call   401050 <__isoc99_scanf@plt>
  40118f:       b8 00 00 00 00          mov    eax,0x0
  401194:       48 8b 55 f8             mov    rdx,QWORD PTR [rbp-0x8]
  401198:       64 48 33 14 25 28 00    xor    rdx,QWORD PTR fs:0x28
  40119f:       00 00
  4011a1:       74 05                   je     4011a8 <vuln_func+0x4f>
  4011a3:       e8 98 fe ff ff          call   401040 <__stack_chk_fail@plt>
  4011a8:       c9                      leave
  4011a9:       c3                      ret
```

vuln_func 栈帧结构



检测Stack Canary是否改变

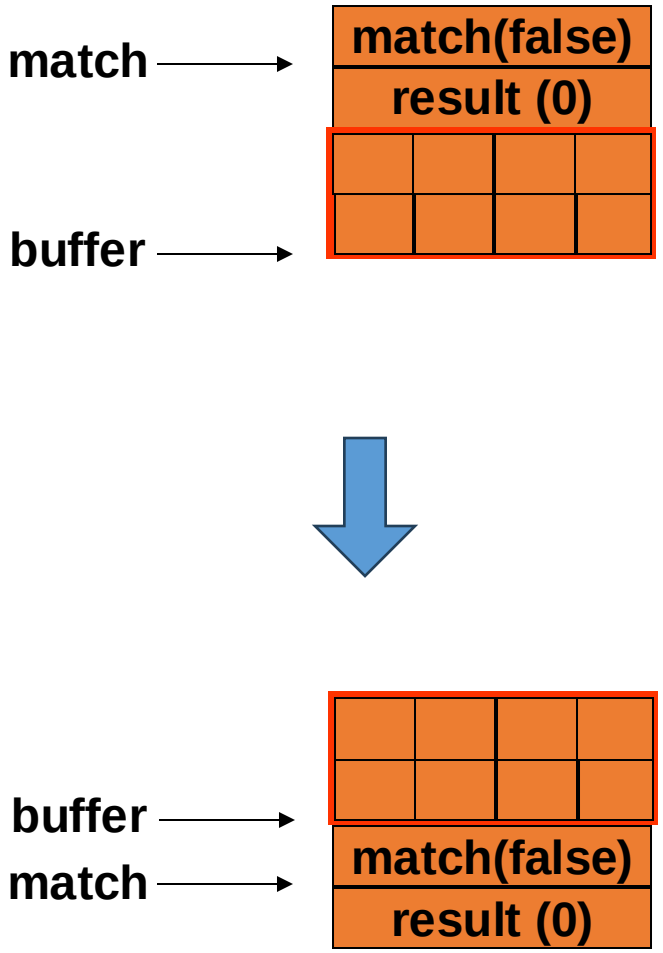
看 test_pwn_with_sp 实例

栈保护机制之变量重排序

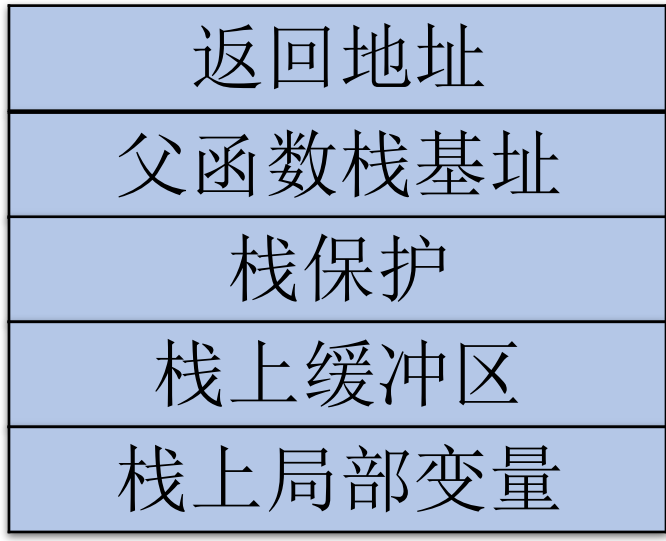
```
bool match = false; int result;
char buffer[8];

printf("3 + 5 = ?\n");
scanf("%s", buffer); // vulnerable scanf
result = atoi(buffer);
if (result == 3+5) match = true;

if (match) {
   printf("The answer is correct\n");
} else {
   printf("The answer is incorrect\n");
}
```



vuln_func 栈帧结构



栈保护机制应用

- 现有编译器基本支持该安全防御
  - GCC，Clang，Visual Studio，VC ……

- 配置选项
  - -fno-stack-protector
  - -fstack-protector
    - 仅适用于两类函数：1. 调用alloca的函数；2. 包含超过8字节的缓冲区的函数
  - -fstack-protector-all
    - 保护所有函数
  - -fstack-protector-strong
    - 包含一些其他函数，如局部数组定义，以及其他指向函数栈帧地址的指针
  - -fstack-protector-explicit
    - 仅保护带有`stack_protect` 属性的函数

[1] https://mudongliang.github.io/2016/05/24/stack-protector.html

栈保护机制绕过

- 利用未被保护的函数
- 同时替换Stack Canary和线程 TLS 数据
- 先通过信息泄露漏洞获取Stack Canary，然后将其加入到Payload中
- Byte-by-byte 栈保护绕过

[1] Preventing brute force attacks against stack canary protection on networking servers