内容与要求

内容

该作业主要考察同学们对二进制文件补丁修补方法及栈溢出漏洞的理解与掌握。所用的实验环境和工具为：

- ubuntu 18.04 LTS
- gcc 7.5.0
- IDA Pro 7.0
- Keypatch v2.2

程序的功能是实现一个非常简单的用户交互：要求同学们输入自己的学号，若输入的学号为10个字符，则在屏幕上打印一段感谢和表扬的话。源代码中共设计了一个逻辑缺陷和一个栈溢出漏洞，要求同学们在没有程序源代码的情况下对二进制文件进行修补。

程序的逻辑缺陷是当输入的学号为10个字符时，重复打印了对男孩和女孩进行感谢和表扬的话，正常情况下只需打印其中一句即可。修补的方法比较简单，直接用IDA中的Keypatch插件将其中一条打印语句改为空指令即可。

修改前的程序如下图所示：

```
.text:0000000000000708                call    _strlen
.text:000000000000070D                cmp     rax, 0Ah
.text:0000000000000711                jnz     short loc_72B
.text:0000000000000713                lea     rdi, aThankYouYouAre ; "Thank you! You are a good boy."
.text:000000000000071A                call    _puts
.text:000000000000071F                lea     rdi, aThankYouYouAre_0 ; "Thank you! You are a good girl."
.text:0000000000000726                call    _puts
```

修改后的程序如下图所示：

```
.text:0000000000000708                call    _strlen
.text:000000000000070D                cmp     rax, 0Ah
.text:0000000000000711                jnz     short loc_72B
.text:0000000000000713                lea     rdi, aThankYouYouAre ; "Thank you! You are a good boy."
.text:000000000000071A                nop                     ; Keypatch modified this from:
.text:000000000000071A                                        ;   call _puts
.text:000000000000071B                nop
.text:000000000000071C                nop
.text:000000000000071D                nop
.text:000000000000071E                nop
.text:000000000000071F                lea     rdi, aThankYouYouAre_0 ; "Thank you! You are a good girl."
.text:0000000000000726                call    _puts
```

gets()函数是一个非常典型的栈溢出函数，常用的修补方法是改用read()函数来实现gets()函数的功能。由于原始的程序中没有调用read()函数，因此只能利用.eh_frame段，通过3号syscall调用read()函数来实现补丁修补。

修改前的程序如下图所示：

```
.text:00000000000006F0                lea     rax, [rbp+s]
.text:00000000000006F4                mov     rdi, rax
.text:00000000000006F7                mov     eax, 0
.text:00000000000006FC                call    _gets
```

修改后，程序在原gets()函数处跳转到.eh_frame段执行，如下图所示：

```
.text:00000000000006F0                lea     rax, [rbp+s]
.text:00000000000006F4                mov     rdi, rax
.text:00000000000006F7                mov     eax, 0
.text:00000000000006FC                jmp     loc_880
```

.eh_frame段插入的漏洞修补代码如下图所示：

```
.eh_frame:0000000000000880 loc_880:
.eh_frame:0000000000000880                 mov     edx, 0Ah
.eh_frame:0000000000000885                 lea     rsi, [rbp+s]
.eh_frame:0000000000000889                 mov     rdi, 0
.eh_frame:0000000000000890                 syscall
.eh_frame:0000000000000892                 jmp     loc_701
```

修改后的程序执行结果如下图所示：

```
root@DESKTOP-C6VM2Q8:/home/project/stack_overflow# /mnt/f/stack_overflow/stack_overflow_2
Please input your student number:
a201900001
Thank you! You are a good girl.
root@DESKTOP-C6VM2Q8:/home/project/stack_overflow#
```

要求修补后的程序仅打印与自己性别相对应的话，且无论输入多长的字符串均不触发栈溢出漏洞。

要求

- 在超星平台提交，A运行前的截图，B修改过程截图，C运行后的截图；尽可能完整截下整个屏幕，带有自己电脑系统的时间等；
- 适当标注一下，自己对补丁的理解，鼓励可以多种方法冷补丁。

过程记录

输出语句限制

首先在WSL中运行该程序，如下：

```
lyg@LAPTOP-J204BNN5:/mnt/e/网络综合实践3/实验4/冷补丁$ ./stack_overflow_2
Please input your student number:
u201814851
Thank you! You are a good boy.
Thank you! You are a good girl.
lyg@LAPTOP-J204BNN5:/mnt/e/网络综合实践3/实验4/冷补丁$
```

观察到输入学号后会输出两条信息。为使得最终的输出进入自己的性别有关，使用64位IDA Pro打开该程序，并将字符串`Thank you! You are a good boy.`的输出部分指令更改为空指令，如下图：

```
00000000000006F0  48 8D 45 F5 48 89 C7 B8  00 00 00 00 E8 9F FE FF  H.E.............
0000000000000700  FF 48 8D 45 F5 48 89 C7  E8 83 FE FF FF 48 83 F8  .H.E.........H..
0000000000000710  0A 75 18 90 90 90 90 90  90 90 90 90 90 90 90 48  .u.............H
0000000000000720  8D 3D EA 00 00 00 E8 55  FE FF FF B8 00 00 00 00  .=.............
0000000000000730  C9 C3 66 2E 0F 1F 84 00  00 00 00 00 0F 1F 40 00  ..f...........@.
0000000000000740  41 57 41 56 49 89 D7 41  55 41 54 4C 8D 25 56 06  AWAVI...UATL.%V.
```

```
.text:000000000000070D                 cmp     rax, 0Ah
.text:0000000000000711                 jnz     short loc_72B
.text:0000000000000713                 nop
.text:0000000000000714                 nop
.text:0000000000000715                 nop
.text:0000000000000716                 nop
.text:0000000000000717                 nop
.text:0000000000000718                 nop
.text:0000000000000719                 nop
.text:000000000000071A                 nop
.text:000000000000071B                 nop
.text:000000000000071C                 nop
.text:000000000000071D                 nop
.text:000000000000071E                 nop
.text:000000000000071F                 lea     rdi, aThankYouYouAre_0 ; "Thank you! You are a good girl."
.text:0000000000000726                 call    _puts
```

此时再次运行程序，观察到仅输出与笔者性别相同的语句，如下图：

```
lyg@LAPTOP-J204BNN5:/mnt/e/网络综合实践3/实验4/冷补丁$ ./stack_overflow_2
Please input your student number:
u201814851
Thank you! You are a good girl.
lyg@LAPTOP-J204BNN5:/mnt/e/网络综合实践3/实验4/冷补丁$
```

输入不限长

由于`.eh_frame`段记录与调试、异常处理有关的信息，因此执行过程中实际并不会用到该段，故可以将目标指令写入该段中，方便跳转使用。为达到输入任意长不溢出的目的，在`.eh_frame`段添加`read()`函数的相关指令。同时，将`call _gets`更改为`jmp`指令跳转至`.en_frame`段执行相关代码，随后再跳转会原代码段的下一条指令执行。其添加的信息如下图所示：

```
00000000000006F0  48 8D 45 F5 48 89 C7 B8  00 00 00 00 E9 7F 01 00  H.E.............
0000000000000700  00 48 8D 45 F5 48 89 C7  E8 83 FE FF FF 48 83 F8  .H.E.........H..
0000000000000710  0A 75 18 90 90 90 90 90  90 90 90 90 90 90 90 48  .u.............H
0000000000000720  8D 3D EA 00 00 00 E8 55  FE FF FF B8 00 00 00 00  .=..............
0000000000000730  C9 C3 66 2E 0F 1F 84 00  00 00 00 00 0F 1F 40 00  ..f...........@.
0000000000000740  41 57 41 56 49 89 D7 41  55 41 54 4C 8D 25 56 06  AWAVI...UATL.%V.
0000000000000750  20 00 55 48 8D 2D 56 06  20 00 53 41 89 FD 49 89  ·.UH.-V.·.SA..I.
0000000000000760  F6 4C 29 E5 48 83 EC 08  48 C1 FD 03 E8 E7 FD FF  ....H...........
0000000000000770  FF 48 85 ED 74 20 31 DB  0F 1F 84 00 00 00 00 00  .H....1.........
0000000000000780  4C 89 FA 4C 89 F6 44 89  EF 41 FF 14 DC 48 83 C3  L..L.....A......
0000000000000790  01 48 39 DD 75 EA 48 83  C4 08 5B 5D 41 5C 41 5D  .H9.......[]A\A]
00000000000007A0  41 5E 41 5F C3 90 66 2E  0F 1F 84 00 00 00 00 00  A^A_Ð·f.........
00000000000007B0  F3 C3 00 00 48 83 EC 08  48 83 C4 08 C3 00 00 00  ....H...........
00000000000007C0  01 00 02 00 00 00 00 00  50 6C 65 61 73 65 20 69  ........Please·i
00000000000007D0  6E 70 75 74 20 79 6F 75  72 20 73 74 75 64 65 6E  nput·your·studen
00000000000007E0  74 20 6E 75 6D 62 65 72  3A 00 00 00 00 00 00 00  t·number:.......
00000000000007F0  54 68 61 6E 6B 20 79 6F  75 21 20 59 6F 75 20 61  Thank·you!·You·a
0000000000000800  72 65 20 61 20 67 6F 6F  64 20 62 6F 79 2E 00 00  re·a·good·boy...
0000000000000810  54 68 61 6E 6B 20 79 6F  75 21 20 59 6F 75 20 61  Thank·you!·You·a
0000000000000820  72 65 20 61 20 67 6F 6F  64 20 67 69 72 6C 2E 00  re·a·good·girl..
0000000000000830  01 1B 03 3B 3C 00 00 00  06 00 00 00 40 FD FF FF  ...;<.......@...
0000000000000840  88 00 00 00 80 FD FF FF  B0 00 00 00 90 FD FF FF  ................
0000000000000850  58 00 00 00 9A FE FF FF  C8 00 00 00 10 FF FF FF  X...............
0000000000000860  E8 00 00 00 80 FF FF FF  30 01 00 00 00 00 00 00  ........0.......
0000000000000870  14 00 00 00 00 00 00 00  01 7A 52 00 01 78 10 01  .........zR..x..
0000000000000880  BA 0A 00 00 00 48 8D 75  F5 48 C7 C7 00 00 00 00  .....H.u........
0000000000000890  0F 05 E9 6A FE FF FF 00  00 00 48 8D 75 F5 48 C7  ..........H.u...
00000000000008A0  C7 00 00 00 00 0F 05 E9  6A FE FF FF 01 78 10 01  .............x..
```

跳转到.eh_frame的指令，修改过程中为褐色

read()函数指令，随后跳回原处，修改过程中为褐色

```
.text:00000000000006DA                 mov     [rbp+var_3], 0
.text:00000000000006E0                 mov     [rbp+var_1], 0
.text:00000000000006E4                 lea     rdi, s          ; "Please input your student number:"
.text:00000000000006EB                 call    _puts
.text:00000000000006F0                 lea     rax, [rbp+s]
.text:00000000000006F4                 mov     rdi, rax
.text:00000000000006F7                 mov     eax, 0
.text:00000000000006FC                 jmp     loc_880
.text:0000000000000701 ; ---------------------------------------------------------------------------
.text:0000000000000701
.text:0000000000000701 loc_701:                                ; CODE XREF: .eh_frame:0000000000000892↓j
.text:0000000000000701                 lea     rax, [rbp+s]
.text:0000000000000705                 mov     rdi, rax        ; s
.text:0000000000000708                 call    _strlen
.text:000000000000070D                 cmp     rax, 0Ah
.text:0000000000000711                 jnz     short loc_72B
.text:0000000000000713                 nop
.text:0000000000000714                 nop
.text:0000000000000715                 nop
.text:0000000000000716                 nop
.text:0000000000000717                 nop
.text:0000000000000718                 nop
.text:0000000000000719                 nop
.text:000000000000071A                 nop
.text:000000000000071B                 nop
.text:000000000000071C                 nop
.text:000000000000071D                 nop
.text:000000000000071E                 nop
.text:000000000000071F                 lea     rdi, aThankYouYouAre ; "Thank you! You are a good girl."
.text:0000000000000726                 call    _puts
```

```
.eh_frame:0000000000000880 ; ---------------------------------------------------------------------------
.eh_frame:0000000000000880 ; START OF FUNCTION CHUNK FOR main
.eh_frame:0000000000000880
.eh_frame:0000000000000880 loc_880:                                ; CODE XREF: main+32↑j
.eh_frame:0000000000000880                 mov     edx, 0Ah
.eh_frame:0000000000000885                 lea     rsi, [rbp+s]
.eh_frame:0000000000000885 ; END OF FUNCTION CHUNK FOR main
.eh_frame:0000000000000889                 mov     rdi, 0
.eh_frame:0000000000000890                 syscall                 ; LINUX -
.eh_frame:0000000000000892                 jmp     loc_701
.eh_frame:0000000000000892 ; ---------------------------------------------------------------------------
```

read

修改后对其进行测试，如下：

```
lyg@LAPTOP-J204BNN5:/mnt/e/网络综合实践3/实验4/冷补丁$ ./stack_overflow_2
Please input your student number:
u201814851XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
Thank you! You are a good girl.
lyg@LAPTOP-J204BNN5:/mnt/e/网络综合实践3/实验4/冷补丁$ XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX: command not found
lyg@LAPTOP-J204BNN5:/mnt/e/网络综合实践3/实验4/冷补丁$
```

```
lyg@LAPTOP-J204BNN5:/mnt/e/网络综合实践3/实验4/冷补丁$ ./stack_overflow_2
Please input your student number:
u201814851
Thank you! You are a good girl.
lyg@LAPTOP-J204BNN5:/mnt/e/网络综合实践3/实验4/冷补丁$
```

观察到没有产生溢出，当得到10个字符后不再读入其他字符，而是直接输出对应的字符串。同时，多余的字符串作为下一条bash指令输入。这是由于read()函数严格限制了输入长度，对于超出长度的数据不做处理。而当 `stack_overflow_2` 程序运行结束后，超出部分被bash当作命令执行。

至此，成功完成了冷补丁。