

## 3. 热补丁 repeat

### 3.1 实验要求

- 程序repeat会重复输出一句格言，将输出的内容修改为自己的姓名。

### 3.2 实验过程

#### 3.2.1 程序分析

- 首先使用IDA打开getshell程序，其main函数的反汇编结果如下：

```
; Attributes: noreturn bp-based frame

; int __cdecl main(int argc, const char **argv, const char **envp)
public main
main proc near

    argc= dword ptr 8
    argv= dword ptr 0Ch
    envp= dword ptr 10h

; __unwind {
    lea     ecx, [esp+4]
    and     esp, 0FFFFFF0h
    push    dword ptr [ecx-4]
    push    ebp
    mov     ebp, esp
    push    ebx
    push    ecx
    call    __x86_get_pc_thunk_bx
    add     ebx, 2E76h
```

```
loc_8049190:
sub     esp, 0Ch
lea     eax, (aPracticeMakesP - 804C000h)[ebx] ; "Practice makes perfect."
push    eax
call    _puts
add     esp, 10h
sub     esp, 0Ch
push    2
call    _sleep
add     esp, 10h
jmp     short loc_8049190
; } // starts at 8049176
main endp
```

- 将main函数反编译，结果如下：

```

1 int __cdecl __noreturn main(int argc, const char **argv, const char **envp)
2 {
3     while ( 1 )
4     {
5         puts("Practice makes perfect.");
6         sleep(2u);
7     }
8 }

```

### 3.2.2 使用 Preload Hook

- 编写自定义的 `puts()` 函数如下：

```

1 // preload_hook_puts.c
2 #define _GNU_SOURCE
3 #include <sys/stat.h>
4 #include <unistd.h>
5 #include <dlfcn.h>
6
7 int puts(char *s) {
8     int (*old_puts)(char *);
9     old_puts = (int (*)(char *))dlsym(RTLD_NEXT, "puts");
10    old_puts("My name is XuBiang@U201911803.");
11 }

```

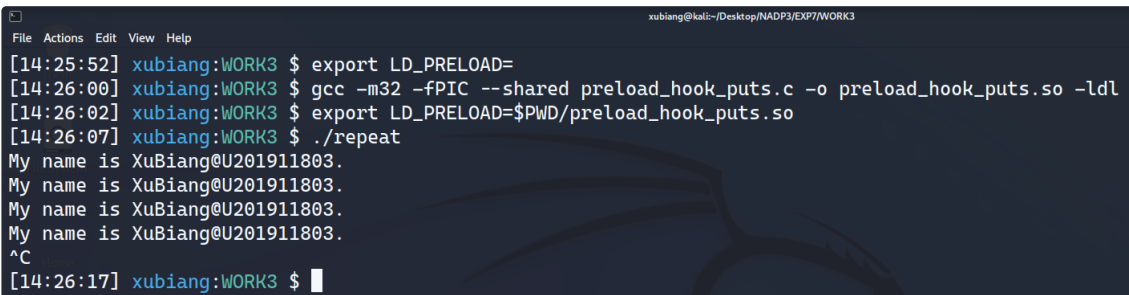
- 编译该文件生成动态链接库：

```

1 gcc -m32 -fPIC --shared preload_hook_puts.c -o
  preload_hook_puts.so -ldl

```

- 加载动态链接库 `export LD_PRELOAD=$PWD/preload_hook_puts.so`，并执行 `repeat` 程序，可以正常完成补丁功能：



```

xubiang@kali:~/Desktop/NADP3/EXP7/WORK3
[14:25:52] xubiang:WORK3 $ export LD_PRELOAD=
[14:26:00] xubiang:WORK3 $ gcc -m32 -fPIC --shared preload_hook_puts.c -o preload_hook_puts.so -ldl
[14:26:02] xubiang:WORK3 $ export LD_PRELOAD=$PWD/preload_hook_puts.so
[14:26:07] xubiang:WORK3 $ ./repeat
My name is XuBiang@U201911803.
My name is XuBiang@U201911803.
My name is XuBiang@U201911803.
My name is XuBiang@U201911803.
^C
[14:26:17] xubiang:WORK3 $

```

### 3.2.3 完整的热补丁

- 补丁代码如下：

```

1 // patch.c
2 #include <stdio.h>
3
4 int newputs() {
5     puts("My name is XuBiang@U201911803.");
6 }

```

- 编译补丁文件:

```

1 gcc -m32 -fPIC --shared patch.c -o patch.so

```

- 加载程序同 [例题/6.2/hotfix](#);
- 先启动 **repeat** 程序, 通过 `ps -aux | grep repeat` 查看其进程 ID, 将其作为参数运行 **hotfix**:

```

xubiang@kali:~/Desktop/NADP3/EXP7/WORK3
[14:54:04] xubiang:WORK3 $ ps -aux | grep repeat
xubiang 380201 0.0 0.0 2524 700 pts/4 S+ 14:54 0:00 ./repeat
xubiang 380211 0.0 0.0 6296 2200 pts/5 S+ 14:54 0:00 grep --color=auto --exclude-dir=.bzr --exclude-dir=CVS --exclude-dir=.git --exclude-dir=.hg --exclude-dir=.svn --exclude-dir=.idea --exclude-dir=.tox repeat
[14:54:10] xubiang:WORK3 $ ./hotfix 380201 ./patch.so puts newputs
main pid = 380201
main libpath : ./patch.so
main oldfunname : puts
main newfunname : newputs
phdr_addr 0x8048034
dyn_addr 0x804bf14
map_addr 0xf7fd9d0
get_sym_info exit
ptrace_read failed: Input/output error
get_sym_info exit
ptrace_read failed: Input/output error
get_sym_info exit
find_symbol_in_linkmap str = printf
find_symbol_in_linkmap sym->st_value = 53ed0
found printf at addr 0xf7e0fed0
status = b7f
get_sym_info exit
ptrace_read failed: Input/output error

```

- 热补丁效果如下:

```

xubiang@kali:~/Desktop/NADP3/EXP7/WORK3
[14:54:01] xubiang:WORK3 $ ./repeat
Practice makes perfect.
Practice makes perfect.
Practice makes perfect.
Practice makes perfect.
Practice makes perfect.
patch succeeded
My name is XuBiang@U201911803.
My name is XuBiang@U201911803.
My name is XuBiang@U201911803.
My name is XuBiang@U201911803.
^C
[14:54:28] xubiang:WORK3 $

```