lab05A

17376283 李智健

Q1

1. `%ebp = 0xbffff5b0`
2. `make_alias` 有 5 个参数，其中 3 个来自于 `save_file` 的参数，2 个局部变量。

   一一对应即可知

   ```
   char *file = 0xbffff780
   int len = 0x000feedb
   char *descriptor = 0xbffff7a0
   ```

   还剩下最后 2 个参数 `0x83045c30` 和 `0x00000045` 。由于 `result = 0` ，所以这两个参数应该来自于 `root` 的结构体成员，类型分别是 `struct l_node *` 和 `int`

   故答案为：

   ```
   void make_alias(char *, int, char *, int, struct l_node *)
   ```

   ```
   make_alias(descriptor, len, file, root->tag, root->next);
   ```

Q2

正确选项应为 A，D，E，F，G，可分为 3 类

第一类是直接把 bp 转换成整数运算。x86-64 使用了 48 位的地址位数，强制转换成 int 和 char 存在溢出风险，因此 B, C 不正确。

第二类是把 bp 转换成指针运算，对于 `long *` ,`int *`, `char *` 型的变量，`--` 对应于分别减去 8，4，1，因此 D，E，F 都对。

第三类是转换成指针的指针，在 x86-64 上，这种类型 `--` 等价于数值减去 8，因此 G 对

Q3

易得

4 < B <= 8

8 + 4 + 2 * B = 24

4 + 8 * A * B = 52

解之得，A = 1，B = 6

Q4

Stack at point B	note
pirintFunc	argument 2
tree	argument 1
return address	
old ebp	
callee-saved esi	
callee-saved ebx	
XXXX	no value
XXXX	no value
printFunc	argument 2
tree->left	argument 1

Q5

1. `buf = 0xffffffc0(%ebp) = %ebp - 0x10 = 0x55683a68`
2. `%ebp = 0x55683a78`
3. `0x08048bf9`
4. `0x55683a58`
5. 只需要保证最后 4 个字节是 `20 8b 04 08` 即可

```
0x01 0x02 0x03 0x04 0x05 0x06 0x07 0x08
0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00
0x00 0x00 0x00 0x00 0x20 0x8b 0x04 0x08
```