Never_abort 解题思路

Authorized by binaryfang.

Medium diffculty but smart exploit technique required!

题目存在一个非常明显的栈溢出，但是防护非常严密：

```
Arch:     amd64-64-little
RELRO:    Partial RELRO
Stack:    Canary found
NX:       NX enabled
PIE:      PIE disabled
```

由于存在canary， main函数中的栈溢出是无法利用的。

但是在触发 `Sig abort` 流程后，程序接管了信号处理例程，在例程中有一个栈上变量extended 长度的高4个byte未初始化漏洞！

```
    int ans = 2;
    char content[4];
    char area[0x100];
    ret_addr = __builtin_return_address(0);
    printf("[*]you want to abort or what?Y/N: ");
    readin(content, &ans);
```

所以，在main函数中触发一个abort流程，跳到sig handler例程中，利用栈上布置的参数，可以在handler里触发一个栈溢出。
但是需要过掉sig handler中的检查和execve，所以需要在main函数中进行泄露。

如何进行泄露?

我们可以直接借助stk_check_failed进行泄露key！同时再覆盖多一点，就可以破坏envp，从而过掉handle中栈溢出之前的execve.

这里的泄露借助的是环境变量 `LIBC_FATAL_ERROR` 置位时，可以开启stderror.

最后，我们会得到一个abort例程的执行上下文，那么刚刚好是sigreturn，所以SROP解之。