实验报告

课程名称：软件测试

实验名称：KLEE实验

专业班级：软件工程3班

学 号：1612880

姓 名：潘忠杰

2018年 12月 30日

实验一

|  |  |  |  |
| --- | --- | --- | --- |
| 实验名称 | KLEE实验 | | |
| 实验地点 | 5区机房 | 实验时间 | 2018.12 |
| 实验目的和要求 | | | |
| 使用KLEE测试软件对含有缺陷的C语言代码进行缺陷测试。 | | | |
| 实验环境 | | | |
| Ubuntu18.04+docker+klee | | | |
| 实验过程 | | | |
| 1.前往docker官方网站下载并安装docker；  2.在命令行中使用sudo apt-get install klee安装klee；  3.在docker创建一个永久性容器用于进行代码测试；  4.首先在容器中进入klee\_src文件夹，之后进入examples文件夹，找到tutorial1对应的文件进行测试，将生成的klee\_last文件夹目录截图 | | | |
| 5.其次在容器中找到regexp文件夹，将内部文件进行测试，将生成的klee\_last文件夹目录截图    6.之后在本地编写一个具有五种缺陷的C语言代码，使用docker cp指令将文件复制进入docker容器，对代码进行测试，得到五个错误，错误截图如下    7.找到klee\_last文件夹目录，并且将所有错误文件保存 | | | |
| 心得体会 | | | |
| 通过本次实验，我对klee软件有了初步的了解，并且对一些基本用法有所掌握，成功测试了待测程序，加深了对于程序缺陷的理解和认识。 | | | |

**附：源程序**

#include <klee/klee.h>

struct

{

int id[3];

char name[5];

int money[6];

char type[3];

}base[2];

void test1(int x) {

int \*ptr;

if (ptr == &x) {

x = 1;

}

}

void test2(int x) {

int a[x];

for(int i = 0; i <= x; i++) {

a[i] = i;

}

}

void test3(int x) {

int b = 1, c;

c = b / x;

}

void test4(int x) {

int b = 2;

x = b;

test5();

}

void test(int\* err) {

test1(err[0]);

test2(err[1]);

test3(err[2]);

test4(err[3]);

}

#define SIZE 4

int main() {

int err[SIZE];

klee\_make\_symbolic(err, sizeof err, "err");

test(err);

return 0;

}