

验 3: ATTACK 实验

一、实验目的

通过实现缓冲区溢出攻击，熟悉理解过程调用规则，包括控制的传递、参数的传递以及栈帧结构、缓冲区溢出问题等，了解缓冲区溢出保护，并进一步加深对理论课程中关于程序的机器级表示各方面知识点的理解，以及反汇编、跟踪/分析/调试等常用技能的掌握。

二、实验内容

作为实验目标的存在缓冲器溢出漏洞的可执行程序共有两个：`ctarget` 和 `rtarget`。对目标程序实施缓冲区溢出攻击（`buffer overflow attacks`）。通过造成缓冲区溢出来破坏目标程序的栈帧结构，继而不再返回原来的调用程序，而是转向执行指定的子程序，要求 `level1` 转向 `touch1` 子程序，`level2` 转向 `touch2` 子程序，`level3` 转向 `touch3` 子程序

(1) `level1`

构造攻击字符串作为目标程序输入，造成缓冲区溢出，使 `getbuf()` 返回时不返回到原来的调用函数，而是转向执行 `touch1`。

```
void touch1()
{
    vlevel = 1;          /* Part of validation protocol */
    printf("Touch1!: You called touch1()\n");
    validate(1);
    exit(0);
}
```

(2) `level2`

构造攻击字符串造成缓冲区溢出，使目标程序调用 `touch2` 函数，并将 `cookie` 值作为参数传递给 `touch2` 函数，使 `touch` 函数中的判断成功，需要在缓冲区中注入指令 `movq` 将 `cookie` 传送给 `rdi` 寄存器作为入口此参数。

```
void touch2(unsigned val)
{
    vlevel = 2;          /* Part of validation protocol */
    if (val == cookie) {
        printf("Touch2!: You called touch2(0x%.8x)\n", val);
        validate(2);
    } else {
```

```

    printf("Misfire: You called touch2(0x%.8x)\n", val);
    fail(2);
}
exit(0);
}

```

(3) level3

构造攻击字符串，使目标程序调用 touch3 函数，将数组指针作为入口参数，使相应判断成功。

```

void touch3(char *sval)
{
    vlevel = 3;          /* Part of validation protocol */
    if (hexmatch(cookie, sval)) {
        printf("Touch3!: You called touch3(\"%s\")\n", sval);
        validate(3);
    } else {
        printf("Misfire: You called touch3(\"%s\")\n", sval);
        fail(3);
    }
    exit(0);
}

```

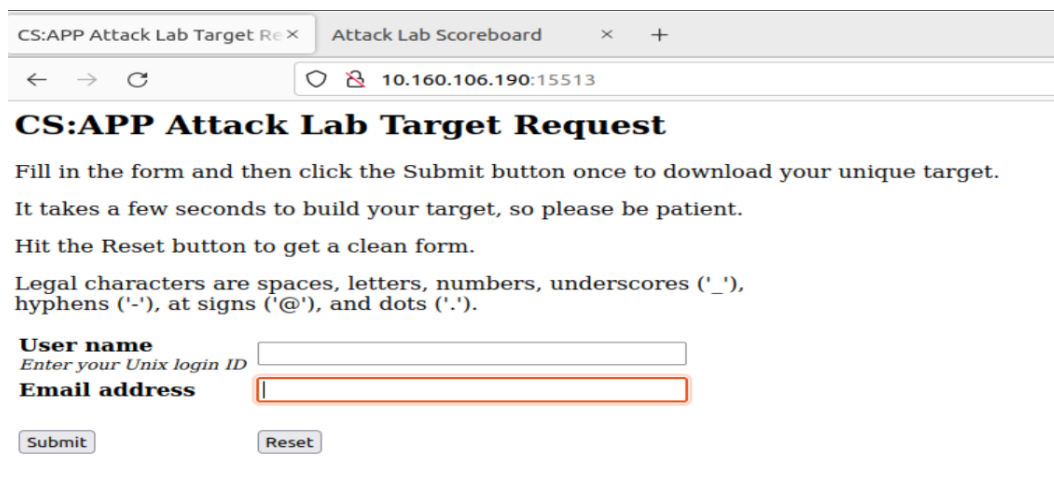
要求对 ctarget 实施代码注入攻击 (code injection)，包含 3 个阶段，分别对应三个 level: level1、level2 和 level3。对 rtarget 实施面向返回的编程攻击 (return-oriented programming)。rtarget 包含 2 个 level: level2 和 level3。

rtarget 程序开启了 栈随机化、栈不可以执行 这两个保护策略。所以需要通面向返回编程来完成第一部分 level2 和 level3 实验，也就是巧妙的利用 farm 程序中的 gadget 指令，gadget 指一段以 ret 指令结尾指令序列。通过调用这些代码，攻击者可以在拥有更简单攻击防范的程序内执行任意操作。(具体请查阅网上相应的资料)。

三、实验过程

第 1 步、获取炸弹程序

在 ubuntu 上通过浏览器输入 <http://10.160.106.190:15513> 后，会出现下面炸弹申请页面：



CS:APP Attack Lab Target Request

Fill in the form and then click the Submit button once to download your unique target.
It takes a few seconds to build your target, so please be patient.
Hit the Reset button to get a clean form.

Legal characters are spaces, letters, numbers, underscores ('_'),
hyphens ('-'), at signs ('@'), and dots ('.').

User name
Enter your Unix login ID

Email address

输入您的用户名（即学号）和电子邮箱地址，然后点击提交按钮。服务器会生成你的程序，并在浏览器返回一个名为 `targetk.tar` 的 `tar` 文件中，其中 `k` 是你程序唯一编号。`targetk.tar` 文件将存储在你的“下载”子目录中。

注意：等待时间比炸弹程序要长，请耐心等待，不要反复按提交按钮。

文件解压后共有以下这些文件，由于目前 CMU 提供的程序依赖较低版本的 `glibc`，`ctarget` 在现在高版本的 `ubuntu` 下无法正确运行，必须在运行时加载一个动态链接库 `printf.so`，因此请各位同学在 `ubuntu` 的浏览器输入 <http://10.160.106.190/printf.so>，下载该文件。

第 2 步：程序上传至服务器

（1）在本地虚拟机下载目录下，将缓冲区溢出攻击程序拷贝到服务器你的目录下：

```
scp targetk.tar username@10.160.106.190:/home/username
```

（2）登录您的 10.160.106.190 帐户

`username=学号`

`>>>>>` 在本地机器上 `>>>>>`

连接服务器：

```
ssh username@10.160.106.190
```

（3）运行解压命令：`tar -xvf targetk.tar`

解压后，它将创建一个目录 `targetk`，在此目录下会包含以下三个文件：

`cookie.txt`：基于学号产生 4 字节序列，如 `0x5f405c9a`，称为“cookie”

`ctarget`：可执行的二进制程序，易遭受目标代码攻击。

`rtarget`：可执行的二进制程序，易遭受面向返回的编程攻击。

`farm.c`：用于实施面向返回的编程攻击需要的 `gadget` 指令。

`hex2raw`：可执行程序，字符串格式转换程序。

（4）在本地虚拟机下载目录下，将 `printf.so` 拷贝到服务器 `targetk` 目录下：

```
scp printf.so username@10.160.106.190:/home/username/targetk
```

第 3 步：进行 ctargget 攻击实验

- (1) 首先进入服务器 `targetk` 目录，反汇编 `ctarget` 程序中的代码，重点阅读 `getbuf` 子程序和 `touch1`、`touch2`、`touch3` 子程序。

```
objdump -d ctargert
```

- (2) 构造攻击字符串的 ASCII 码, 使用文本编辑器保存在文本文件中(如 `ctarget.ll`) (十六进制格式), 如:

00 00 00 00 00 00 00 00

00 00 00 00 00 00 00 00

00 00 00 00 00 00 00 00

01 40 30 40 00 00 00 00

- (3) 使用 `hex2raw` 程序将 `ctarget.ll` 文件中数字对序列转化为对应的字节序列保存在另一个文本文件中如 `clevell.txt`:

```
./hex2raw < ctargct.ll > clevel1.txt
```

- #### (4) 进行攻击

情况 1: 不通知服务器

```
LD PRELOAD=./printf.so ./ctarget < clevel1.txt -q
```

攻击成功显示如下:

[illegible]

攻击失败可能显示如下:

```
teacher@ICS:~/CSAPP/attacklab/targets/target2$ LD_PRELOAD=./printf.so ./ctarget <clevel3.txt -q
Cookie: 0x47Db4e3a
Type string:ouch!: You caused a segmentation fault!
Better luck next time
FAIL: Would have posted the following:
    user id chenyanll
    course   15213-f15
    lab      attacklab
result     2: FAIL:0xffffffff:ctarget:0:48 C7 C7 88 21 65 55 C3 00 00 00 00 00 00 00 00 00 00 00 00 00
```

情况 2: 通知服务器, 计算成绩

```
LD PRELOAD=./printf.so ./ctarget < clevel1.txt
```

攻击成功显示如下:

```
teacher@ICS:~/CSAPP/attacklab/targets/target2$ LD_PRELOAD=./printf.so ./ctarget <clevel1.txt
Cookie: 0x47db4e3a
Type string:Touch1!: You called touch1()
Valid solution for level 1 with target ctarget
PASS: Sent exploit string to server to be validated.
NICE JOB!
```

攻击失败可能显示如下:

```
teacher@ICS:~/CSAPP/attacklab/targets/target2$ LD_PRELOAD=./printf.so ./ctarget <clevel3.txt
Cookie: 0x47db4e3a
Type string:Ouch!: You caused a segmentation fault!
Better luck next time
FAILED
teacher@ICS:~/CSAPP/attacklab/targets/target2$ █
```

注意：如果显示如下结果，请查看记分牌成绩，如果计分成功，则表明攻击成功。

```
teacher@ICS:~/CSAPP/attacklab/targets/target2$ LD_PRELOAD=./printf.so ./ctarget <clevel2.txt
Cookie: 0x47db4e3a
Type string:Touch2!: You called touch2(0x47db4e3a)
Valid solution for level 2 with target ctarget
Ouch!: You caused a segmentation fault!
Better luck next time
FAILED
```

通过浏览器输入 <http://10.160.106.190:15513/scoreboard>，查看记分牌来了解自己的实验成绩。

Phase 1: ctarget.l1,

Phase 2: ctarget.l2,

Phase 3: ctarget.l3,

Phase 4: rtarget.l2,

Phase 5: rtarget.l3,

注：也可如下使用管道操作符“|”连接 hex2raw 和 ctarget:

```
cat ctarget.l1 | ./hex2raw | LD_PRELOAD=./printf.so ./ctarget
```

(5) 跟踪调试程序

执行 gdb ctarget 进入 gdb 后，首先执行：

```
set environment LD_PRELOAD=./printf.so
```

设置断点后，运行 r < clevel1.txt 进行跟踪调试。

第 4 步：进行 rtarget 攻击实验(不需要 printf.so)

- (1) 首先进入服务器 targetk 目录，反汇编 rtarget 程序中的代码，重点阅读 getbuf 子程序和 touch2、touch3 子程序。

```
objdump -d ctarget
```

- (2) 构造攻击字符串的 ASCII 码，使用文本编辑器保存在文本文件中(如 ctarget.l1)（十六进制格式），如：

```
00 00 00 00 00 00 00 00
```

```
00 00 00 00 00 00 00 00
```

```
00 00 00 00 00 00 00 00
```

```
01 40 30 40 00 00 00 00
```

- (3) 使用 hex2raw 程序将 rtarget.l2 文件中数字对序列转化为对应的字节序列保存在另一个文本文件中如 rlevel2.txt:

```
./hex2raw < rtarget.l2 > rlevel2.txt
```

(4) 进行攻击

情况 1：不通知服务器

```
./rtarget < rlevel2.txt -q
```

情况 2：通知服务器，计算成绩

```
./rtarget < rlevel2.txt
```

注：也可如下使用管道操作符“|”连接 hex2raw 和 rtarget

```
cat rtarget.l2 | ./hex2raw | ./rtarget
```

(5) 跟踪调试程序

执行 `gdb ctarget` 进入 `gdb` 后，设置断点后，运行 `r < rlevel2.txt` 进行跟踪调试。

三、实验程序提交

1.在个人目录下创建子目录 lab3。

>>>>> 在服务器上 >>>>>

```
cd /home/username
```

```
mkdir lab3
```

4.将攻击字符串文件：ctarget.l1, ctartget.l2, ctargetl3, rtarget.l2, rtarget.l3 和实验报告复制到子目录 lab3 中。

实验报告命名格式：学号姓名 lab3.docx

四、通知

1.截止日期：2024.6.30

2.当实验室提交日期到期时，我们将从子目录中取走符合命名格式的作业，延迟提交或不符合命名格式的作业将不会被取走。

3.这个运行于 10.160.106.190 的实验室系统在我们的校园内。如果你在我们的外面校园，请使用南邮 VPN 客户端程序 ENWAgent。