

Buffer Overflow

CSE127 Program 1
Due May 12th 4:30pm

Before you begin this assignment, we strongly recommend reading the following articles:

“Smashing The Stack For Fun And Profit”

<http://www.shmoo.com/phrack/Phrack49/p49-14>

“øSTACK OVERFLOW EXPLOiTS ON LiNux / BSDOS / FREEBSD / SUNOS / SOLARiS / HP-UXø”

<http://thc.org/papers/OVERFLOW.TXT>

You will be using the shell code provided in the first article to exploit a buffer overflow. Unlike the examples in the articles you will be performing this attack in a single process (i.e. you will be exploiting the buffer overflow directly through a function call). A number of files have been provided for you in the public folder on ieng6

(/home/linux/ieng6/cs127s/public):

attack.c: This is the only file you will be turning in. You must define the constant `STUDENT_ID` with the last 4 digits of your student ID number. This is to randomize the location of the target buffer for each account (see `target.h`). `attack.c` will not compile until you change the `#define` on line 9 and comment out line 8.

If you modify these files for debugging purposes, be aware that I will be testing `attack.c` against the originals.

Makefile: This contains the compiler options to allow execution of code on the stack.

shellcode.h: This contains the shell code from the phrack article. I will be changing the `"/bin/sh"` to some other program when grading so you should not hardcode any lengths when dealing with `shellcode[]`.

target.h: This contains the function you will be exploiting. A local array is created on the stack using the `STUDENT_ID` constant defined in `attack.c` for the size.

overflow1.c: This is an example from the phrack article. You should be sure this compiles and gives you a `/bin/sh` shell before writing the `attack.c` file.

What to do: Simply put, you are to execute the shell code by exploiting the vulnerability in foo(). This means you must prepare the attackstring in main and pass it to foo. Please read the above articles for information on how to prepare the attack string.

Turning in: When logged into ieng6 (or one of the workstations served by ieng6) use the following command:

```
turnin -c cs127s attack.c
```

You can turn in multiple times until the due date/time. Subsequent turnins overwrite the previous turnin.

Make sure your name, student ID number, and email address appear in the first 10 lines of the attack.c file.

Other info: This is an individual assignment. No cheating, copying, or otherwise. Automated comparison tools will be used

There is a class webboard at <http://webboard.ucsd.edu>. You are responsible for checking the webboard for any last minute updates prior to turning in your assignment (at least 24 hours prior to the due date).

Name: Your account is your UCSD network name
Password: Your student ID # (i.e. aXXXXXXXXX)