CS177: Computer Security
Prof. Stefano Tessaro

UC Santa Barbara
Fall 2018

Homework 4

Posted: Wednesday, November 7, 2018 – 11:59pm
Due: Friday, November 16, 2018 – 11:59pm – online submission via gradescope.

Instructions. You will need to login onto the class VM to solve this homework. Instructions have been sent to you via e-mail. **By logging in, you agree to use your account only to the extent needed to solve the tasks in this homework assignment.**

You will have to submit your solutions via gradescope. Include in particular:

- A README file explaining *all details* of how you solved the tasks, and how you produced the necessary exploits. This should include for example how malicious inputs have been generated, and why they look the way they do. If you need to compute particular offsets or lengths in memory, give all details of how you computed them.
- Exploits exploitX.py, where X $\in \{1, 2, 3, 4\}$, where the submission of the fourth exploit is not mandatory. Exploits can also be submitted in C if preferred.

Important. Some basic rules needed for you to obtain credit for your solution:

- An exploit approach for one task may work for a previous task. However, we are asking for specific approaches to be used, therefore solve the tasks *as requested*.
- You are not allowed to re-implement pieces of code provided in existing tutorials that e.g. automatize the attack. *Ideally, your exploit should just invoke the vulnerable process with a malicious input, and possibly set environment variables.*
- For Tasks 1,2, and 3, you can find C code of the executables in the respective directories. This is not really necessary to solve the tasks, but it may help you.

Task 1 – Control-Flow Hijacking (10 points)

The executable `/home/Mr177/auth` has syntax

```
/home/Mr177/auth password
```

and running it with the right password will result in a shell being opened with the privileges of Mr177's group.

Find a way to open this shell *without knowing* the password by appropriately re-directing the control flow of the execution. Here, *do not* attempt to inject your own shell code. Once you gained shell access, run the command `h4ck` to record your success.

Task 2 – Shellcode (10 points)

Another vulnerable executable is `/home/Shelly/auth`. Write an exploit for it resulting in opening up a shell with Shelly's group privileges.

Your shellcode should be injected into a buffer. Once you gained shell access, run the command `h4ck` to record your success.

Hint: While you may use other ones, it is recommended that you use the (right) shellcode from AlephOne's tutorial.

Task 3 – More Shellcode (10 points)

Write an exploit for `/home/Codey/auth` which results in opening up a shell with Codey's group privileges.

Once you gained shell access, run the command `h4ck` to record your success.

Task 4 – Some More Fun (10 bonus points)

Now have a look at `/home/MrCode`. Try to find a way to open up a shell with MrCode's group privileges.

Once you gained shell access, run the command `h4ck` to record your success.