COMP 4384: Assignment #4

Revision 1

Due on December 10, 2020 at 1:00 PM

30 Points (15% Overall)

Problem 1

(10 points)

For this problem, you will need the LiveHacking virtual machine you used in Assignment 3. The code for this problem (shown below) has a format string vulnerability that will allow you to get a reward. Are you brave enough to claim the reward? Your answer must include:

1. Enough details about the reasoning you followed to craft a malicious format string that allowed you to claim the reward.
2. Screenshot(s) showing the program run with the malicious input and the successful claim of the reward.

Hint: You can use `$(perl -e 'print "AAA"')` script to pass command line arguments to your program.

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

void vuln(char *string) {
    volatile int target;
    char buffer[16];
    target = 0;

    sprintf(buffer, string);

    if(target == 0xdeadbeef) {
        printf("You've got a reward :)\n");
    } else {
        printf("Target = %p", target);
    }
}

int main(int argc, char **argv) {
    vuln(argv[1]);
}
```

Problem 2

(10 points)

The code at (`https://gist.github.com/atamrawi/084e6ebb3b2085760141549024938 02f`) suffers from a Time-of-check to time-of-use (TOCTOU) vulnerability. Your task is take advantage of the vulnerability to read the contents of the `secret.txt` file. Your answer must include:

1. Enough details about the reasoning you followed to exploit the TOCTOU vulnerability.
2. Screenshot(s) showing the program run along with the successful reading of `secret.txt` file's content.

Problem 2 continued on next page...

For this problem, you can use any virtual Linux machine. To start working on your exploit, you need to setup an environment for our user `hacker` with proper permissions and files as follows:

1. In a Terminal window, add a new user `hacker` to our system:

   `sudo adduser hacker`

2. Enter a password of `hacker` twice and press `Enter` to leave all the other information blank.

3. Create a new file `race.c` and paste the vulnerable code for this problem to it.

4. Compile the code using `root` privileges:

   `sudo gcc race.c -o race`

5. Change permissions to `race` to set the SUID bit:

   `sudo chmod 4755 race`

6. Copy our vulnerable program to `hacker`'s home directory:

   `sudo cp -p race /home/hacker`

7. Create a new file `secret.txt` with the content `SECRET INFORMATION` under `hacker`'s home directory:

   `sudo vi /home/hacker/secret.txt`

8. Change permissions for `secret.txt` to limit access to file except for `root` user:

   `sudo chmod 600 /home/hacker/secret.txt`

9. Open a new terminal window, and change user to `hacker` and enter the password you set for `hacker`:

   `su hacker`

10. Go to `hacker`'s home directory:

    `cd`

11. Try to view the content's of `secret.txt` via (`cat secret.txt`), you should see a "Permission denied" message.

12. Create a new file `public.txt` with the content `PUBLIC INFORMATION` under `hacker`'s home directory:

    `vi /home/hacker/public.txt`

13. Try to view the content's of `public.txt` via (`cat public.txt`), you should see a "PUBLIC INFORMATION" message.

14. Now, let us test out `race` program.

15. Execute (`./race public.txt`), and enter `y` when prompted, you should see that you are able to access the contents of `public.txt` file.

16. Execute (`./race secret.txt`), and enter `y` when prompted, you should see "*You don't have access to secret.txt*" message.

17. Now, you are your own, try to exploit the TOCTOU vulnerability in `race` to view the contents of `secret.txt` without any `sudo` operations, changing permissions, or switching from `hacker` user.

Problem 3

(10 points)

The code at (`https://gist.github.com/atamrawi/38af7f0f75de6cd7d941e494ead791f1`) is a modified version of the example we discussed in class. The code has a use-after-free vulnerability that can be exploited to log in any user without a password. Can you take exploit the vulnerability to log in any user without a password?
Your answer must include:

1. Enough details about the reasoning you followed to exploit the vulnerability to log in any user without a password.
2. Screenshot(s) showing the program run with the different commands you tried for successfully logging in any user without a password.