ECE404 Introduction to Computer Security: Homework 10

Spring 2024
Due Date: 5:59pm, April 4, 2024

1 Introduction

In this assignment, we will explore one of the most prevalent vulnerabilities in software systems - buffer overflows. Buffer overflow attacks remain a significant threat to computer systems, and understanding how they work is crucial for any aspiring cybersecurity professional.
The objective of this assignment is to provide you with hands-on experience in identifying, exploiting, and mitigating buffer overflow vulnerabilities. Specifically, you will be tasked with mounting a buffer overflow attack on a socket program.
After, the buffer overflow assignment, you will also be tasked with setting up spam filter accounts for next week's homework assignment.
As always, please read the homework document in its entirety before coming to office hours with your questions. The teaching staff have spent a long time writing the assignment to cover many common questions you might have.

2 Buffer Overflow Attack

Included with this homework, you will find two socket programs written in C. One of them acts as a server and the other as a client. Your first task is to launch an attack that executes the "secret" function in the server side code by using the client program to send a carefully crafted string to the server. Following this, your second task is to show how you would fix `server.c` to prevent such an attack from happening. Listed below are some more details to consider when completing these two tasks.

- It is advised that you do this entire assignment on `eceprog.ecn.purdue.edu`
- When compiling the source code with `gcc`, make sure to include the `-fno-stack-protector` flag. Refer to pages 36-37 of Lecture 21 for more details.

- You will need two terminals to complete this assignment: one terminal for the server and one terminal for the client. Make sure to start the server program before starting the client program
- Use `gdb` to determine how you can develop a string to send (using the client program) to the server program and trigger the execution of `secretFunction()`. Refer to Lecture 21.6 for more details on how to do this. Some things to consider when crafting the string are:
    1. While you send the data with the client program, you will have to run the server program with gdb to determine the buffer overflow string to use.
    2. When sending the string to the server program, note that you can send both ASCII characters as well as hexadecimal bytes. You can send, for example, the hex byte 0xAD using the format \xAD
    3. As in the lecture notes, you will need to reverse the order of addresses sent to deal with the big-endian little-endian conversion problems.
- After you have completed crafting your buffer overflow string and triggered the execution of the secret function, modify `server.c` to remove the buffer overflow vulnerability. Your fix should allow the program to run without the threat of a buffer overflow attack.

3 Spam Filter Account Set-up

Next week, we will be doing an assignment involving spam filters. For that assignment, we are providing you temporary ECN accounts on the Shay server. To obtain your credentials for your account, first find ECE404 on Brightspace. Under the "Grades" section for this class, there are two items of interest, one is your account's username (titled "login") and the other is your account's password (titled "Password"). The "Grade" value for these items indicate the username and password you will use.

On Windows, you can use PuTTY to ssh into the account. Alternatively you can ssh into the account using the following syntax on MacOSX and Linux:

```
1    ssh yourUsername@shay.ecn.purdue.edu
```

Please make sure that you can log into this account. Should you face any trouble, please reach out to Joseph as soon as possible to get the issue resolved. Once you have successfully logged into the account, feel free to change your passworld using the passwd command. For future reference the email address associated with this account is `yourUsername@ecn.purdue.edu`. Do not try to log into this account with Microsoft Outlook or ECN webmail.

Once you are able to access your account, follow the instructions below to get your account up and fully operational.

1. Unzip the attached tar.gz file that contains a text file named `dot_procmailrc` and a Perl script named `GET_MESSAGE_INDEX`
2. Apply the `dos2unix` command to these files to remove the carriage return characters added after line feed. Do not skip this step as the `dot_procmailrc` file is sensitive to such characters, because it contains regular expressions.
3. Carefully read the comments in the `dot_procmailrc` file and make the necessary changes to Recipes 2 and 3.
   - Recipe 2 requires that you place the name of your special account in the last line of the recipe
   - Recipe 3 requires that the string user_name in the last line be repalced by your email account at Purdue
4. Rename `dot_procmailrc` to `.procmailrc` and store it your new account's home directory (e.g. `/home/shay/a/ece404q8`). You can invoke sftp or scp to move the file from your local machine to the new account.
5. Create a new directory called `Mail` at the top level of your new ECN account and place the `GET_MESSAGE_INDEX` script in that directory
6. Send a test email message to your new account. You can verify that you recived this test mail if a file titled `logfile` is created in your `Mail` directory.
7. The best command-line tool for processing the email recived by your new account is `mailx`. Do `man mailx` to see all the options that go with this command. You can even use this command to send messages to others. In my experience, mailx works best when you use SSH to access the account directly (as opposed to using ThinLinc or ECEGrid).

8. After your account has become operational, please subscribe to random newsletters, newsgroups, websites... etc. This process will cause spam to be directed to your email addresses in a fairly short time. Include a page worth of logfile contents in your pdf submission.

4 Submission Instructions

- For this homework you will be submitting a zip file titled `hw10_<last name>_<first name>.zip`, which consists of:
  - A pdf titled `hw10_<last name>_<first name>.pdf` containing:
    - the specially-crafted buffer overflow string
    - an explanation of how you determined the special string
    - an explanation of your fixes to `server.c` including screenshots of code snippets that reflect the necessary changes.
    - A page worth of logfile contents from the Mail directory of your spam accounts
  - the modified `server.c` with comments explaining the vulnerability and the fix.
  - The updated `.procmail` file

References