CSC 472 Fall 2023 Lab 4

Dr. Si Chen

November 2, 2024

Multi-Stage Exploits

The primary objective of this lab is to understand and exploit vulnerabilities using techniques like Information Leakage, GOT Overwrite, and ROP (Return-oriented programming). Students are required to infiltrate a remote server using these methods and retrieve sensitive information.

The goals of this lab are:

- Understanding the intricacies of Multi-Stage Exploits.
- Gaining hands-on experience in exploiting vulnerabilities using Information Leakage, GOT Overwrite, and ROP.
- Retrieving sensitive data from a compromised server.

Server Details

- **Target IP:** 159.203.157.119
- **Target Port:** 6666
- **Vulnerable program:** lab4 (lab4.c)
- **Target File:** flag.txt
- **ASLR/NX:** On
- **StackGuard and PIE:** Off
- **Libc version:** libc6-i386_2.33-0ubuntu5_amd64

Exploit Objectives

Objective 1: Information Leakage

Understand the memory layout of the running application, leak addresses that will be useful for later stages of the exploit.

Objective 2: GOT Overwrite

Manipulate the Global Offset Table (GOT) entries to control the flow of execution.

Objective 3: Crafting ROP Chain

Using the knowledge from the previous stages, craft a ROP payload to gain control over the program's execution. Your payload should allow you to get an shell and be able to retrieve the contents of 'flag.txt'.

Steps:

1) Analyze the given vulnerable program 'lab4.c' for potential vulnerabilities.

2) Begin by leveraging Information Leakage to gather useful memory addresses and understand the memory layout.

3) Proceed to exploit the GOT to hijack the flow of the application.

4) Craft your ROP chain to gain a shell on the remote server.

5) Once you've obtained the shell, navigate to the target file 'flag.txt' and retrieve its contents.

6) Document each step with relevant screenshots and code snippets, detailing your findings, the challenges faced, and how you overcame them.

Deliverables: A comprehensive project report in PDF format detailing every step of the exploit. This should include all relevant screenshots, code snippets, and the content inside 'flag.txt'.

Hints

1. This Lab is closely related to the examples in the course "Multi-Stage Exploits". It is highly recommended to review the corresponding course slides and videos to ensure a complete understanding of the content before beginning this lab.
2. Use the 'lab4' binary ELF file downloaded from the website or found in the 'lab4' folder on BadgerCTF to find relevant information about PLT, GOT, and ROP Gadgets. Do

not compile the ‘lab4.c‘ code by yourself. Recompiling the ‘lab4.c‘ code might result in different memory addresses for PLT and GOT due to potential differences in the GCC compiler version from the remote server. This discrepancy could render your script unusable on the remote server.

3. You can download the libc file from the website and use Pwntools’ ELF function to discover offsets on your own:

```
libc = ELF("./libc.so.6")
offset_write = libc.symbols['write']
offset_system = libc.symbols['system']
```

   Note that these offsets only apply to the specified libc version on the remote server and not the local version on BadgerCTF.

4. If you opt to first hack locally, you need to modify the ‘lab4_exp.py‘ script, changing the line ‘p = remote(”159.203.157.119”, 6666)‘ to ‘p = process(”./lab4”)‘. After successfully exploiting the vulnerability locally, remember to update the offsets for all libc functions to those of the remote libc version (as mentioned in the previous hint) before attempting to exploit the remote server.

5. Once you successfully infiltrate the remote server, you can use commands to list the stored files on the server. Utilize the ‘cat‘ command to display the contents of the ‘flag.txt‘ file.

Submission

- The lab due date is available on our course website. Late submissions will not be accepted.
- The assignment should be submitted to D2L directly.
- Your submission should include: A **detailed project report in PDF format** describing your entire exploit process, including screenshots, code snippets, and the content from ‘flag.txt‘.
- **No copy or cheating is tolerated**. If your work is based on others’, please provide clear attribution. Otherwise, you **WILL FAIL** this course.