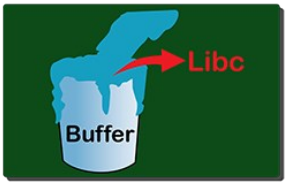
Return-to-libc Attack Lab

Overview



The learning objective of this lab is for students to gain the first-hand experience on an interesting attack on buffer-overflow vulnerability; this attack can bypass an existing protection scheme currently implemented in Linux operating systems. A common way to exploit a buffer-overflow vulnerability is to overflow the buffer with a malicious shellcode, and then cause the vulnerable program to jump to the shellcode that is stored in the stack. To prevent this kind of attacks, some operating systems, such as Fedora Linux, allow system administrators to make stacks non-executable; therefore, jumping to the shellcode will cause the program to fail.

Unfortunately, the above protection scheme is not fool-proof; there exists another type of attacks, the `return-to-libc` attack, which does not need an executable stack; it does not even use shell code. Instead, it causes the vulnerable program to jump to some existing code, such as the `system()` function in the `libc` library, which is already loaded into the memory.

In this lab, students are given a program with a buffer-overflow vulnerability; their task is to develop a `return-to-libc` attack to exploit the vulnerability and finally to gain the root privilege. In addition to the attacks, students will be guided to walk through several protection schemes that have been implemented in Linux to counter against the buffer-overflow attacks. Students need to evaluate whether the schemes work or not and explain why.

Tasks (English) (Spanish)

- **VM version:** This lab has been tested on our `SEED Ubuntu-20.04` VM
- **Lab setup files:**: Labsetup.zip

Time (Suggested)

- Supervised (closely-guided lab session): **2 hours**
- Unsupervised (take-home project): **1 week**

SEED Videos

- Udemy: Computer Security: A Hands-on Approach (§ 5)

SEED Books (English) (Chinese)

- Computer & Internet Security: A Hands-on Approach, 3rd edition (§ 5)
- Computer Security: A Hands-on Approach, 3rd edition (§ 5)

Feedback and Help

Please give us your feedback on this lab using this feedback form.

The SEED Labs project is open source. If you are interested in contributing to this project, please check out our Github page: https://github.com/seed-labs/seed-labs.