NSPJ24 User-Level System Call Hook

mystery
08 Mar 2024, 09:17 AM

Overview

Zpoline: a system call hook mechanism based on binary rewriting

Zpoline employs binary rewriting techniques to redirect the originally intended `syscall` instructions to a user-defined hook function, enabling users to implement user-level system call hooking.

In this project, you need to use *zpoline* to achieve monitoring and modification of system calls. The recommended operating environment is Ubuntu 22.04 x86_64. (If you are using an M1/M2 Mac, you will require a virtual machine that supports x86 emulation.)

Task I

Descriptions

In this section, you are required to build *zpoline* and try the default system call hook feature.

Requirement

1. Build *zpoline* from source code : GitHub - yasukata/zpoline: system call hook for Linux
2. Execute the default system call hook by *zpoline*.
   `LIBZPHOOK=./apps/basic/libzphook_basic.so LD_PRELOAD=./libzpoline.so /bin/ls`
3. Check the system call table to determine which system calls have been invoked by the ls program.

Task II

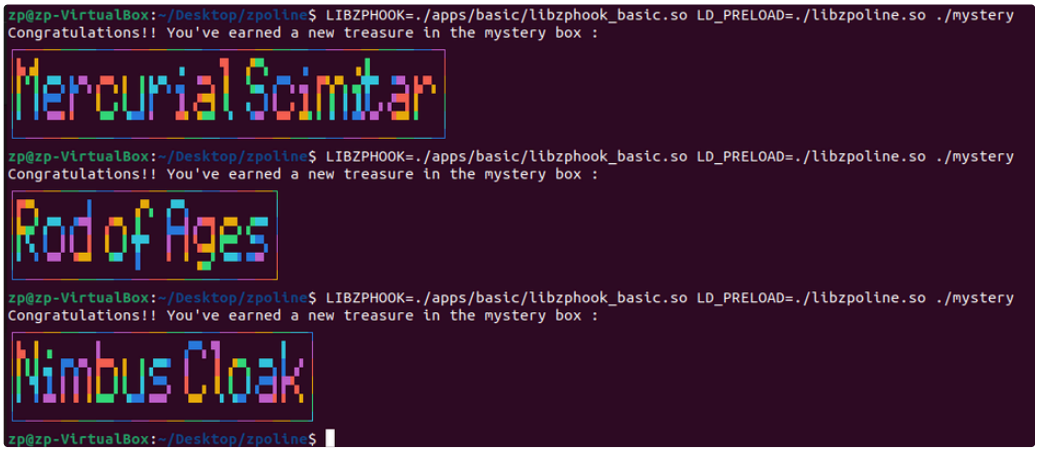
Descriptions

In this section, you need to implement a hook function to modify the behavior of the system call.

Requirement

1. Install *toilet*, a command-line tool in Linux that enables users to create colorful and stylized text art.
   `sudo apt install toilet`
2. Modify `hook_function` in `apps/basic/main.c`.
3. Compile and generate your shared object library.
   `make -C apps/basic`
4. Run `./mystery` several times. Just to ensure that you understand its behavior.
5. Try to hook system calls for `./mystery`.
   `LIBZPHOOK=./apps/basic/libzphook_basic.so LD_PRELOAD=./libzpoline.so ./mystery`

Goal : Bring colors to your treasures!



Note

1. You must ensure that your treasures are obtained randomly.
2. The text color should vary, while the rest of the text styling is flexible.

Report

You need to write a report answering the following questions :

- Task I
  a. A screenshot of your output.
  b. Describe which system call is used by /bin/ls to retrieve file and directory names.
- Task II
  a. A screenshot of your output.
  b. Describe how you design `hook_function` to modify the appearance of your treasures.
- In the paper, they mentioned that *zpoline* cannot hook vDSO-based system calls. Do you think performing a binary patch on the vDSO memory mapping inside the hooked process is a feasible solution? Provide your reasoning.
- Describe difficulties you encountered in the implementations and how you have addressed them.

Submission

Please submit your report and the program source, i.e., `apps/basic/main.c`, to E3.

- Make sure your code can be compiled and run on Ubuntu 22.04 LTS x86_64.
- Make sure your output is correct, as mentioned.

- Your report should be submitted in PDF format.

For questions, please contact TA Mr. Chang <m2955121314.11@nycu.edu.tw>