[pwn之路]patchelf之后，加载符号表！

前言

当你在进行二进制漏洞学习和利用时，经常需要使用调试工具来分析和理解程序的内部工作。在之前的交流中，我们提到了如何使用 `patchelf` 来修改二进制文件[Pwn之路]根据所给库，获得远程同环境——使用patchelf的正确姿势，以适应调试的需求，但没有详细介绍如何加载符号表。实际上，对于学习和利用二进制漏洞，符号表是非常重要的资源，因为它们提供了关键的函数和变量名称，使调试和分析更加轻松。

有些人可能会建议使用虚拟机来解决这个问题，但这通常会涉及到一些繁琐的设置和维护工作。幸运的是，你可以利用强大的工具如 `pwndbg` 以及 `glibc-all-in-one` 项目，来轻松加载符号表，无需复杂的虚拟机设置。

在下文中，我们将详细介绍如何使用pwndbg的加载符号表功能，以及如何结合glibc-all-in-one项目自动下载符号表来简化调试过程。这个方法不仅方便，还能帮助你更好地理解和利用二进制漏洞，提高你的漏洞研究和利用技能。让我们一起来探索这个强大的调试工具和资源吧！

脚本获取，配置

执行 `vim ~/.pwndbg/gdbinit.py`
然后复制下方代码到最底下，然后输入 `:wq` 回车，保存退出。

```python
import gdb
import os

# ANSI颜色转义码
COLOR_GREEN = "\033[32m"  # 绿色
COLOR_RED = "\033[31m"    # 红色
COLOR_RESET = "\033[0m"   # 重置颜色

# 递归加载符号文件的函数
def load_symbols_recursive(folder_path):
    # 获取文件夹下的所有内容，包括子文件夹
    items = os.listdir(folder_path)

    for item in items:
        item_path = os.path.join(folder_path, item)

        if os.path.isfile(item_path):
            try:
                gdb.execute("add-symbol-file {}".format(item_path))
                print(COLOR_GREEN + "[+] Loaded symbols " + COLOR_RESET + "from {}".format(item_path))
            except gdb.error as e:
                print(COLOR_RED + "[-] Failed to load" + COLOR_RESET + " symbols from {}: {}".format(item_path, e))
        elif os.path.isdir(item_path):
            load_symbols_recursive(item_path)  # 递归处理子文件夹

# 创建一个自定义的GDB命令，用于递归加载文件夹及其子文件夹中的所有符号文件
class LoadSymbolsRecursively(gdb.Command):
    def __init__(self):
        super(LoadSymbolsRecursively, self).__init__("loadfolder", gdb.COMMAND_USER)

    def invoke(self, arg, from_tty):
```

```
        # 解析参数为文件夹路径
        folder_path = arg.strip()

        if not folder_path:
            print("Usage: loadfolder <folder_path>")
            return

        # 检查文件夹是否存在
        if not os.path.exists(folder_path) or not os.path.isdir(folder_path):
            print(COLOR_RED + "[-] Folder does not exist: {}".format(folder_path) +
COLOR_RESET)
            return

        # 调用递归函数来加载符号文件
        load_symbols_recursive(folder_path)

LoadSymbolsRecursively()
```

使用

运行 `pwndbg` 时，输入

`loadfolder /path/to/glibcallinone/libs/your_glibc_version/.debug/.build-id/`

比如我最近在做一道题目，运行的是

`loadfolder /home/N1nE/ctf/tools/glibcallinone/libs/2.34-0ubuntu3_amd64/.debug/.build-id/`

。

重复加载问题

调试过程中，如果需要重新加载，直接 `ctrl+r` ，输入 `load` 就会出现刚才输入过的命令，重新加载即可，无需重新输入。

posted @ 2023-10-03 23:31 .N1nEmAn 阅读(687) 评论(2)