COMP 8005
Assignment 2
Report

Daryush Balsara
A01265967
Sept 23, 2025

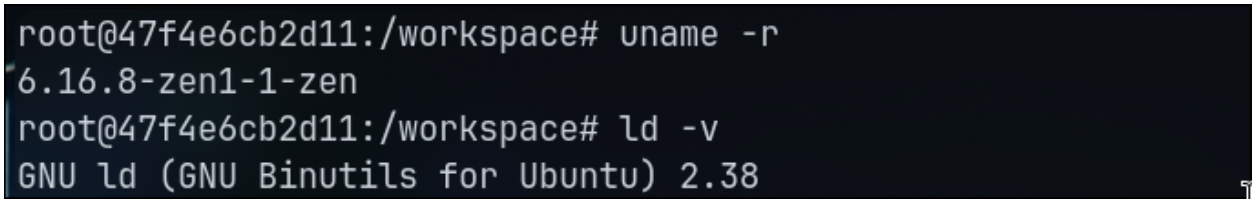
Executive Summary

LD_PRELOAD hides items only within a process's userland view and leaves kernel-level and offline artifacts intact. Correlating userland output with kernel /proc data and file-integrity evidence reliably reveals dynamic linker interposition. Defenses include kernel-side monitoring, integrity checks, and restricting untrusted LD_PRELOAD usage.

Introduction

This project implements a small LD_PRELOAD interposer that overrides a few libc symbols to hide files/processes from userland tools. You'll also design two independent detection techniques (for example, kernel /proc checks and offline file-integrity verification) and gather evidence from multiple vantage points to show the concealment is runtime linker interposition—not on-disk removal.

Environment



Setup

```
root@47f4e6cb2d11:/workspace# cd ldpreload_lab/
root@47f4e6cb2d11:/workspace/ldpreload_lab# mkdir labdata
root@47f4e6cb2d11:/workspace/ldpreload_lab# printf 'visible\n' > labdata/visible.txt
root@47f4e6cb2d11:/workspace/ldpreload_lab# printf 'secret\n' > labdata/secret.txt
root@47f4e6cb2d11:/workspace/ldpreload_lab# printf 'hide_me\n' > labdata/hide_me.txt
root@47f4e6cb2d11:/workspace/ldpreload_lab# touch labdata/.hidden_public
root@47f4e6cb2d11:/workspace/ldpreload_lab# ls -la labdata
total 12
drwxr-xr-x 1 root root 92 Sep 23 23:55 .
drwxr-xr-x 1 root root 14 Sep 23 23:54 ..
-rw-r--r-- 1 root root  0 Sep 23 23:55 .hidden_public
-rw-r--r-- 1 root root  8 Sep 23 23:55 hide_me.txt
-rw-r--r-- 1 root root  7 Sep 23 23:55 secret.txt
-rw-r--r-- 1 root root  8 Sep 23 23:55 visible.txt
root@47f4e6cb2d11:/workspace/ldpreload_lab#
```

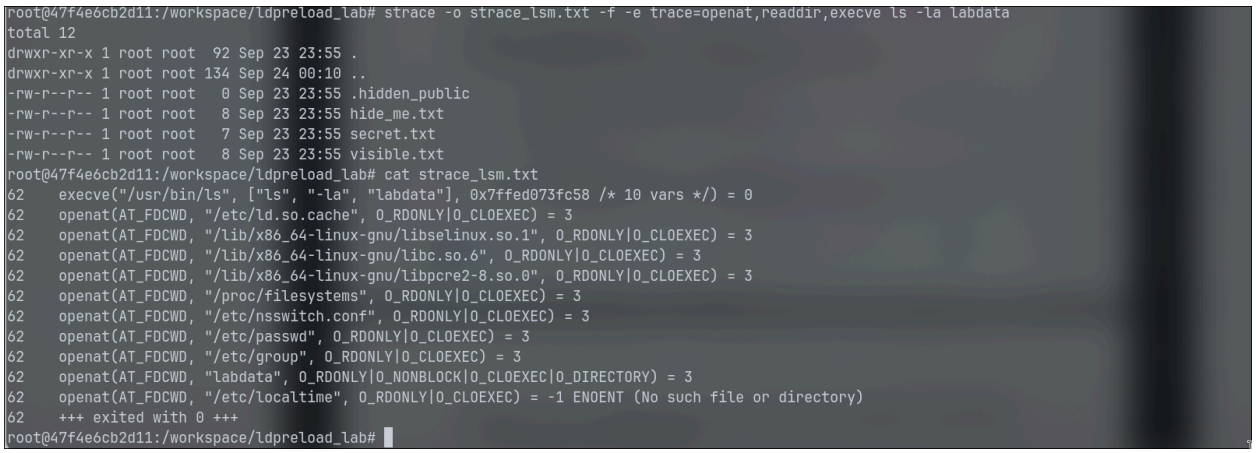
```
root@47f4e6cb2d11:/workspace/ldpreload_lab# vi libhide.c
root@47f4e6cb2d11:/workspace/ldpreload_lab# gcc -fPIC -shared -o libhide.so libhide.c -ldl
root@47f4e6cb2d11:/workspace/ldpreload_lab# readelf -Ws libhide.so | head

Symbol table '.dynsym' contains 12 entries:
   Num:    Value          Size Type    Bind   Vis      Ndx Name
     0: 0000000000000000     0 NOTYPE  LOCAL  DEFAULT  UND
     1: 0000000000000000     0 FUNC    GLOBAL DEFAULT  UND getenv@GLIBC_2.2.5 (2)
     2: 0000000000000000     0 NOTYPE  WEAK   DEFAULT  UND _ITM_deregisterTMCloneTable
     3: 0000000000000000     0 FUNC    GLOBAL DEFAULT  UND _exit@GLIBC_2.2.5 (2)
     4: 0000000000000000     0 NOTYPE  WEAK   DEFAULT  UND __gmon_start__
     5: 0000000000000000     0 FUNC    GLOBAL DEFAULT  UND dlsym@GLIBC_2.34 (3)
     6: 0000000000000000     0 FUNC    GLOBAL DEFAULT  UND fwrite@GLIBC_2.2.5 (2)
root@47f4e6cb2d11:/workspace/ldpreload_lab#
```

Method 1 strace

ls -al

```
root@47f4e6cb2d11:/workspace/ldpreload_lab# ls -al labdata/
total 12
drwxr-xr-x 1 root root 92 Sep 23 23:55 .
drwxr-xr-x 1 root root 80 Sep 24 00:01 ..
-rw-r--r-- 1 root root  0 Sep 23 23:55 .hidden_public
-rw-r--r-- 1 root root  8 Sep 23 23:55 hide_me.txt
-rw-r--r-- 1 root root  7 Sep 23 23:55 secret.txt
-rw-r--r-- 1 root root  8 Sep 23 23:55 visible.txt
root@47f4e6cb2d11:/workspace/ldpreload_lab#
```



```
root@47f4e6cb2d11:/workspace/ldpreload_lab# strace -o strace_lsm.txt -f -e trace=openat,readdir,execve ls -la labdata
total 12
drwxr-xr-x 1 root root  92 Sep 23 23:55 .
drwxr-xr-x 1 root root 134 Sep 24 00:10 ..
-rw-r--r-- 1 root root   0 Sep 23 23:55 .hidden_public
-rw-r--r-- 1 root root   8 Sep 23 23:55 hide_me.txt
-rw-r--r-- 1 root root   7 Sep 23 23:55 secret.txt
-rw-r--r-- 1 root root   8 Sep 23 23:55 visible.txt
root@47f4e6cb2d11:/workspace/ldpreload_lab# cat strace_lsm.txt
62    execve("/usr/bin/ls", ["ls", "-la", "labdata"], 0x7ffed073fc58 /* 10 vars */) = 0
62    openat(AT_FDCWD, "/etc/ld.so.cache", O_RDONLY|O_CLOEXEC) = 3
62    openat(AT_FDCWD, "/lib/x86_64-linux-gnu/libselinux.so.1", O_RDONLY|O_CLOEXEC) = 3
62    openat(AT_FDCWD, "/lib/x86_64-linux-gnu/libc.so.6", O_RDONLY|O_CLOEXEC) = 3
62    openat(AT_FDCWD, "/lib/x86_64-linux-gnu/libpcre2-8.so.0", O_RDONLY|O_CLOEXEC) = 3
62    openat(AT_FDCWD, "/proc/filesystems", O_RDONLY|O_CLOEXEC) = 3
62    openat(AT_FDCWD, "/etc/nsswitch.conf", O_RDONLY|O_CLOEXEC) = 3
62    openat(AT_FDCWD, "/etc/passwd", O_RDONLY|O_CLOEXEC) = 3
62    openat(AT_FDCWD, "/etc/group", O_RDONLY|O_CLOEXEC) = 3
62    openat(AT_FDCWD, "labdata", O_RDONLY|O_NONBLOCK|O_CLOEXEC|O_DIRECTORY) = 3
62    openat(AT_FDCWD, "/etc/localtime", O_RDONLY|O_CLOEXEC) = -1 ENOENT (No such file or directory)
62    +++ exited with 0 +++
root@47f4e6cb2d11:/workspace/ldpreload_lab#
```

When I run strace on the ls -al command it the shell executing the ls command located at the /usr/bin directory. It also shows the dynamically linked ld library which is at /lib/x86_64-linux-gnu/. Then it checks what filesystem is supported by the kernel. The following lines are for the usernames and groups that are associated with the ls command. The last line adds timestamps to the output of the ls command.

Modified ls -al

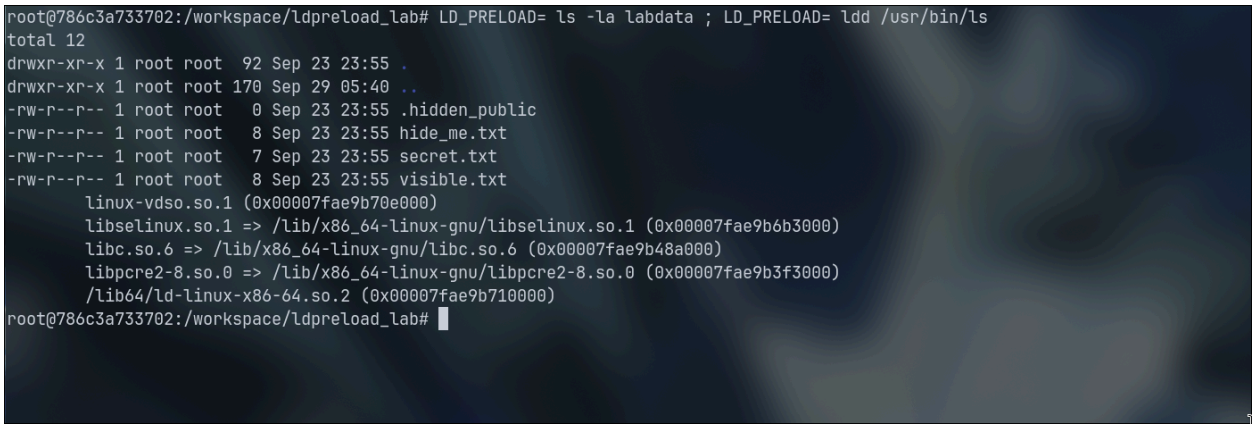
```
root@47f4e6cb2d11:/workspace/ldpreload_lab# LD_PRELOAD=./libhide.so ls -la labdata > dynamic_ls.txt 2>&1
root@47f4e6cb2d11:/workspace/ldpreload_lab# cat dynamic_ls.txt
total 8
drwxr-xr-x 1 root root 92 Sep 23 23:55 .
drwxr-xr-x 1 root root 80 Sep 24 00:01 ..
-rw-r--r-- 1 root root  0 Sep 23 23:55 .hidden_public
-rw-r--r-- 1 root root  7 Sep 23 23:55 secret.txt
-rw-r--r-- 1 root root  8 Sep 23 23:55 visible.txt
root@47f4e6cb2d11:/workspace/ldpreload_lab# cat labdata/hide_me.txt
hide_me
root@47f4e6cb2d11:/workspace/ldpreload_lab#
```

```
root@47f4e6cb2d11:/workspace/ldpreload_lab# strace -o strace_ls.txt -f -e trace=openat,readdir,execve env LD_PRELOAD=./libhide.so ls -la labdata
total 8
drwxr-xr-x 1 root root  92 Sep 23 23:55 .
drwxr-xr-x 1 root root 106 Sep 24 00:07 ..
-rw-r--r-- 1 root root   0 Sep 23 23:55 .hidden_public
-rw-r--r-- 1 root root   7 Sep 23 23:55 secret.txt
-rw-r--r-- 1 root root   8 Sep 23 23:55 visible.txt
root@47f4e6cb2d11:/workspace/ldpreload_lab# cat strace_ls.txt | less
bash: less: command not found
root@47f4e6cb2d11:/workspace/ldpreload_lab# cat strace_ls.txt
55    execve("/usr/bin/env", ["env", "LD_PRELOAD=./libhide.so", "ls", "-la", "labdata"], 0x7ffddd640958 /* 10 vars */) = 0
55    openat(AT_FDCWD, "/etc/ld.so.cache", O_RDONLY|O_CLOEXEC) = 3
55    openat(AT_FDCWD, "/lib/x86_64-linux-gnu/libc.so.6", O_RDONLY|O_CLOEXEC) = 3
55    execve("/usr/local/sbin/ls", ["ls", "-la", "labdata"], 0x55e7f60e7400 /* 11 vars */) = -1 ENOENT (No such file or directory)
55    execve("/usr/local/bin/ls", ["ls", "-la", "labdata"], 0x55e7f60e7400 /* 11 vars */) = -1 ENOENT (No such file or directory)
55    execve("/usr/sbin/ls", ["ls", "-la", "labdata"], 0x55e7f60e7400 /* 11 vars */) = -1 ENOENT (No such file or directory)
55    execve("/usr/bin/ls", ["ls", "-la", "labdata"], 0x55e7f60e7400 /* 11 vars */) = 0
55    openat(AT_FDCWD, "./libhide.so", O_RDONLY|O_CLOEXEC) = 3
55    openat(AT_FDCWD, "/etc/ld.so.cache", O_RDONLY|O_CLOEXEC) = 3
55    openat(AT_FDCWD, "/lib/x86_64-linux-gnu/libselinux.so.1", O_RDONLY|O_CLOEXEC) = 3
55    openat(AT_FDCWD, "/lib/x86_64-linux-gnu/libc.so.6", O_RDONLY|O_CLOEXEC) = 3
55    openat(AT_FDCWD, "/lib/x86_64-linux-gnu/libpcre2-8.so.0", O_RDONLY|O_CLOEXEC) = 3
55    openat(AT_FDCWD, "/proc/filesystems", O_RDONLY|O_CLOEXEC) = 3
55    openat(AT_FDCWD, "/etc/nsswitch.conf", O_RDONLY|O_CLOEXEC) = 3
55    openat(AT_FDCWD, "/etc/passwd", O_RDONLY|O_CLOEXEC) = 3
55    openat(AT_FDCWD, "/etc/group", O_RDONLY|O_CLOEXEC) = 3
55    openat(AT_FDCWD, "labdata", O_RDONLY|O_NONBLOCK|O_CLOEXEC|O_DIRECTORY) = 3
55    openat(AT_FDCWD, "/etc/localtime", O_RDONLY|O_CLOEXEC) = -1 ENOENT (No such file or directory)
55    +++ exited with 0 +++
root@47f4e6cb2d11:/workspace/ldpreload_lab#
```

In the modified ls command we can see in the first line when trying to execute the command it uses the environment variable LD_PRELOAD which points to ./libhide.so which is our modified ls command. Then the next 2 lines are for the dynamic linker. After that we can see the env tries to run ls by searching if its in the $PATH and it searches 3 locations /usr/local/sbin/ls, /usr/local/bin/ls, /usr/sbin/ls these files don’t exist upon the 4 search in /usr/bin/ls the file is found and the next lines are the same as the same as the regular ls command.

Method 2 ldd

ls -al



When I run the ldd which is the dynamic link loader tool to check the regular version of ls it shows all the things inside the labdata directory and we can see nothing malicious is being linked to when we run ldd where the ls command is located.

Modified ls -al

When running the modified version of ls we can see not everything in the labdata directory is being shown. When we run the ldd on the modified ls we can see that one of the lines is linking to the malicious LD_PRELOAD of ./libhide.so (0x00007f1ad42f4000). Comparing both our ls and modified ls we can see that this line is not there in the regular ls which means it is linked to the malicious ls command.

Correlation of Evidence and Confidence

The ldd command shows that the libhide.so is being preloaded and dynamically linked to while strace shows the modified runtime behaviour. Together both these commands show that our ls command is compromised and is not working the way it's intended. My confidence level is high because I used two commands to determine if our ls command was modified.

Conclusion

LD_PRELOAD can reliably hide items from a process's userland view, but it's process-scoped and leaves detectable traces. Correlating kernel-level views and offline artifacts with userland outputs exposes the deception. Use layered defenses—kernel-side checks, integrity monitoring, and process inspection—to detect and mitigate linker interposition.