1. Implementation System Call

First what is System call

A system call is a way for a program to request a service from the operating system's kernel.

When we write a regular program (like in C, Python, etc.), it usually runs in user mode, which has limited privileges for safety. But certain tasks like reading a file, creating a process, accessing hardware, or allocating memory require more permissions and direct access to system resources. That's where system calls come in.

To make a system call that uses setuid() in a Qubes OS AppVM, you're essentially writing user-space code (in C, for example) that invokes the existing setuid() syscall provided by the Linux kernel within the AppVM's operating system (like Fedora or Debian). You’re not modifying Qubes OS itself—just interacting with the OS inside the VM.

As The installation process is not completed well I only try to explain the theoretical phase and include some system call codes from different documentation.

Let’s go step-by-step and write a basic C program that uses the setuid() system call inside a Qubes AppVM.

Step 1: C Program Using setuid()

#include <stdio.h>

#include <unistd.h>

#include <sys/types.h>

int main() {

uid\_t new\_uid = 1000; // Replace with the target UID

if (setuid(new\_uid) == 0) {

printf("Successfully changed UID to %d\n", new\_uid);

} else {

perror("setuid failed");

return 1;

}

// Run something with the new UID

system("whoami");

return 0;

}

Step 2: Compile the Program

Inside the AppVM terminal:

gcc -o setuid\_example setuid\_example.c

Step 3: Run the Program as Root

Because setuid() lowers privileges (and only root can change to another UID), you need to run the binary as root:

sudo ./setuid\_example

Note on Qubes OS Context:

Qubes OS uses virtualization to isolate environments.

This program runs within a single AppVM. It cannot affect or change the UID of processes in other VMs.

If your intention is to use setuid() to manage privilege separation within an AppVM, this is valid.

If you're trying to control or affect another qube, that must be done via Qubes RPC policies or inter-qube messaging.