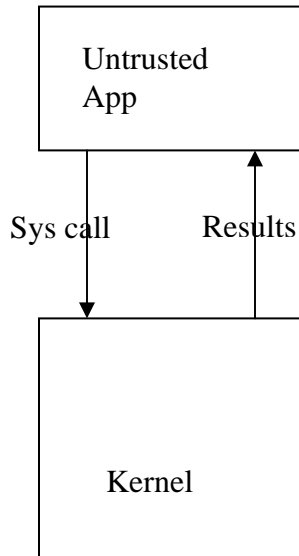


## OSTIA(SANDBOXING)

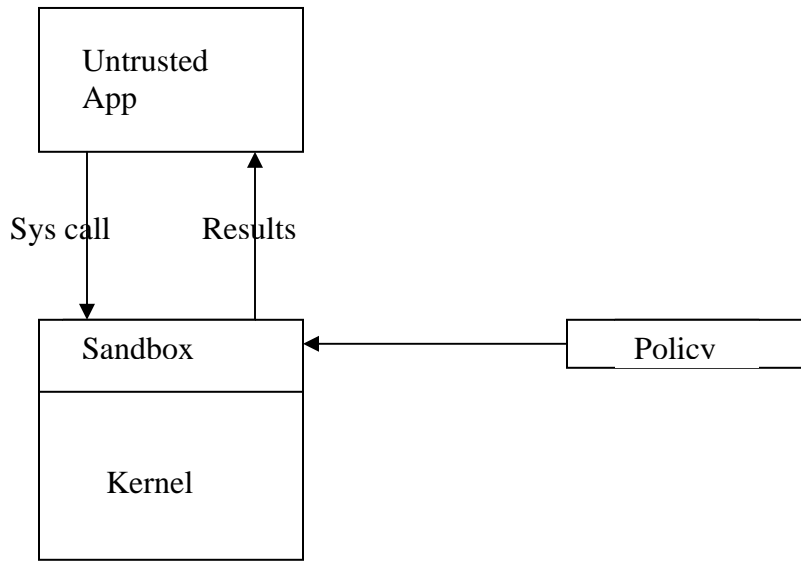
- Don't trust the code
- Want to run it safely
- =>System Call Monitoring

### Basic Structure



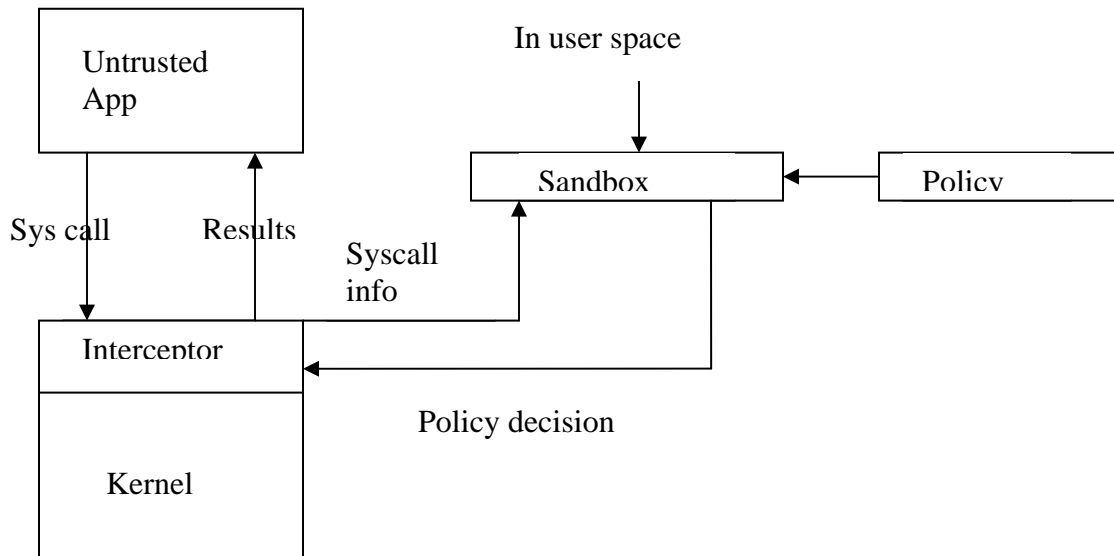
- How do we implement this ?

Idea 1: Add in the kernel



Disadvantage : Sandbox may introduce bugs in the kernel

Idea 2: Move sandbox out of the kernel (Hybrid)

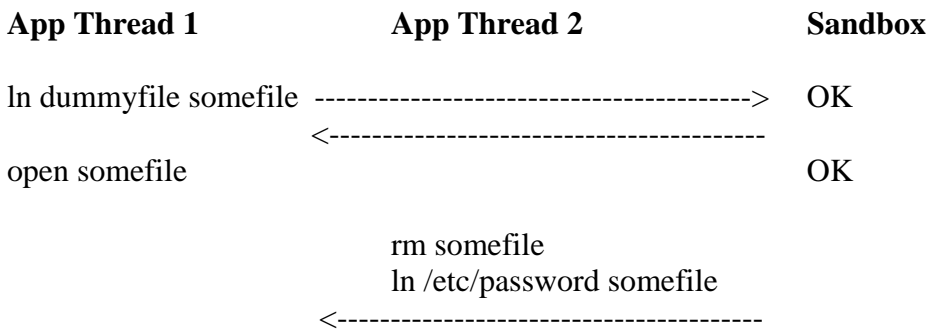


Disadvantage :

- Races
  - memory race
  - OS race

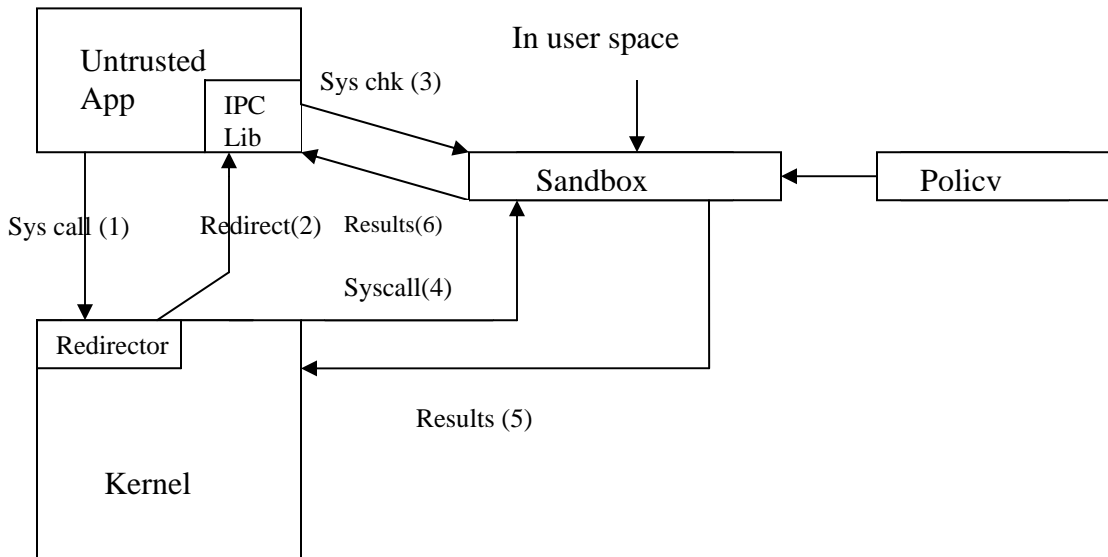
Memory race : The untrusted app requests to open a file. Sandbox does the access control check and gives OK signal if it is indeed OK. Then the app changes the name of the file for some malicious purpose which the kernel opens on the premise that sandbox has given the OK signal for that file. This is basically a time of check to time of use race condition.

OS race:



Note: ---> indicates context switch.

These are prevented by the following scheme:



-Delegating Architecture

- makes memory races go away
- makes it possible to do policy checks in race free manner

-Performance?

- many system calls are unmediated
- patch the code to use IPC lib

-IPC lib is not trusted.

**RACE-FREE POLICY ENFORCEMENT**

-Suppose the policy is:

Can only open files in "/var/tmp"

Then what about /var/tmp/../../ha

- Canonicalize?
- Walk filename by hand
- Races?