

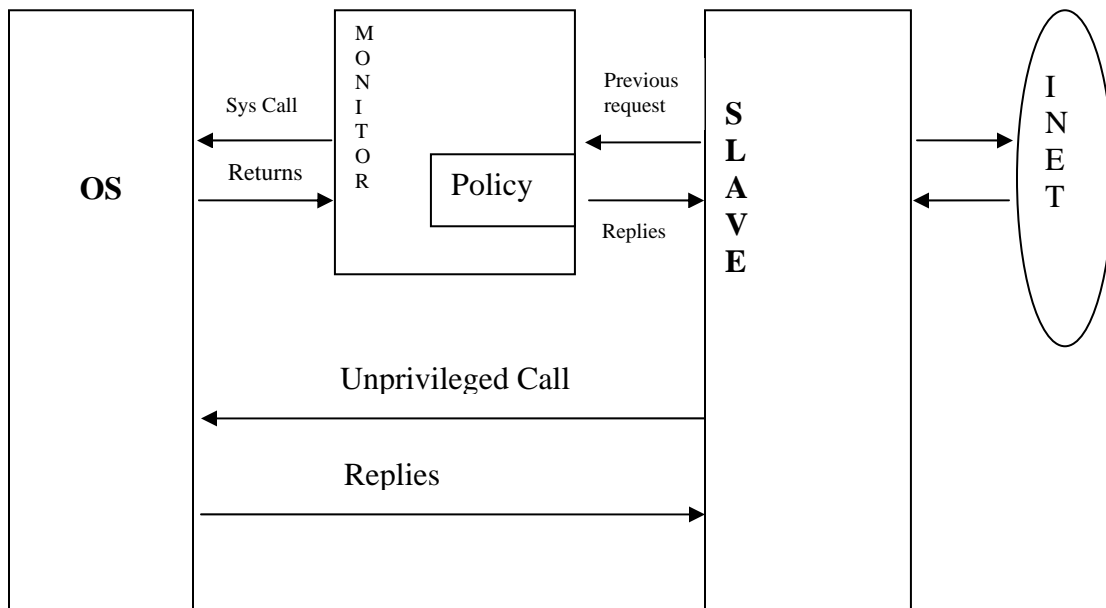
Session on 1st March on PrivTrans

Privilege separation

- Minimize damage from system compromise
- Easier to reason about over all system

Open SSH (daemon)

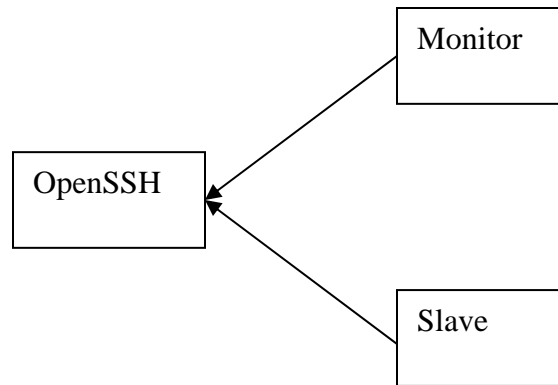
- Open hosts private key
 - Perform digital signature or private key
- OS level privileged operations
 - Opening a pseudo tty (or for instance any special operation)
- Switch its user-id
 - After user logs in



- Monitor spawns slave process

Advantages

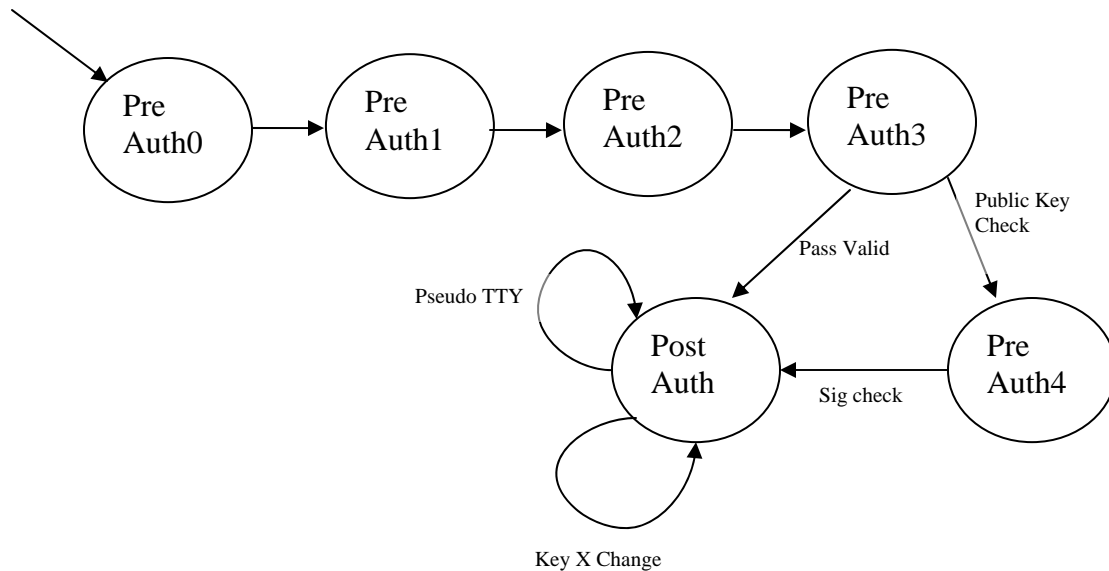
- If slave is compromised, limit damage??



Because of the above architecture, bugs can be detected at Slave level only.

- If bug goes to monitor, then
 - Probably a problem, no concrete solution
- If monitor is small, can verify its correctness?
 - Also less likely to contain bugs.
- If slave is buggy and user is in control of slave, then user can issue a command to monitor "Change my id to root"
 - Monitor must enforce a policy on its privileged interface.
- Interface => Must bind privilege & policy.

Policy



PrivTrans

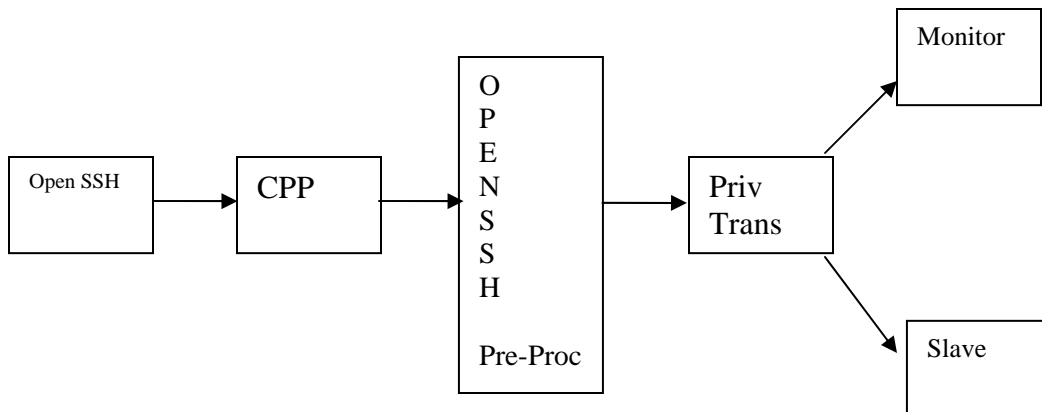
- Automatically perform privilege separation
 - Author labels “privileged”
 - Function
 - Data
- PrivTrans generates monitor of slave.

Privilege separation requirements

- Monitor should be small
- Policy should go in monitor
- Should not create new bugs
- IT'S EASY!!!
- Should facilitate code audits

Details

- “IT'S EASY”
 - For author
 - Easier than manual
- Facilitating audits
 - Note: PrivTrans works on CIL which runs during pre-processing.



- Shouldn't create new bugs
 - PrivTrans is not a sound transformation
- Small Monitor
- Policy
 - Doing this automatically is very hard.

End Note:

PrivTrans doesn't distinguish between Secrecy and Integrity