Enforcement

#enforcement/tcb

Trusted Computing Base (TCB)

- operating systems usually treat applications as blackboxes
- operating systems controls flows among applications
  - allowed flows defines the *security requirements*
- contains all the software and hardware deemed to be *trusted*

Trusted vs Trustworthy

- **trusted** - system that has been shown to meet well-defined criteria
  - *explicit*
- **trustworthy** - sufficient evidence that one can conclude the system will meet the criteria
  - *implicit*

Layers

- **application layer**
  - do not trust applications
  - need to depend on some application enforcement
    - lots of root processes
    - more semantics
    - can break systems
  - cannot treat applications as black boxes anymore
- **network layer**
  - *firewall* is the network access control
  - need to protect a network from external threats
  - the internal network (hosts) need to be ready for the approved but untrusted messages
- **virtual machine layer**
  - *isolation* - each vm is a protection domain
  - *problem* - vms are not homogeneous
    - some are security-critical applications

        - others are untrusted inputs and less critical applications
    - need a way to use vm isolation and flows among the vms to achieve security goals
- **architecture layer
    - hardware of the system
    - we want to trust it, but we can't
        - spectre, meltdown, etc
    - there have been lots of efforts looking at the interplay between *architecture* and *systems*

Security Enforcement

- access control is included in several applications
    - e.g. databases, web servers, browsers
- also included in some programming languages
    - e.g. java, python, ruby
- some systems do recognize that programs may contribute to access control
    - SELinux has a user-level policy server
- **requirement** - ensure that all layers are using their authority in a manner consistent with *system security goals*
- those **responsible**
    - programmers
    - tool-chain providers (e.g. compilers, runtimes)
    - os distributors
    - administrators
    - users
    - service providers
    - content providers

Questions to Consider

- how to define what is necessary?
    - also what is necessary for success?
- how to define enforcement for individual layers comprehensively?
- how to compose enforcement of all layers into a coherent security architecture?
- how to prove success?
- how to succeed without much or any user intervention?
- *is this enough?*