Accountability and Freedom

Butler Lampson
Microsoft
October 27, 2005
Real-World Security

• It’s about risk, locks, and deterrence.
  – Risk management: cost of security < expected loss
    – Perfect security costs way too much
  – Locks good enough that bad guys break in rarely
  – Bad guys get caught and punished enough to be deterred, so police / courts must be good enough.
  – Can recover from damage at an acceptable cost.

• Internet security similar, but little accountability
  – Can’t identify the bad guys, so can’t deter them
Causes of Security Problems

• Exploitable bugs
• Bad configuration
  – TCB: Everything that security depends on
    Hardware, software, and **configuration**
  – Does formal policy say what I mean?
    • Can I understand it? Can I manage it?

• Why least privilege doesn’t work
  – Too complicated, can’t manage it

_The unavoidable price of reliability is simplicity_
—Hoare
The Access Control Model

1. **Isolation Boundary**: I am isolated if anything that goes wrong is my (program’s) fault

2. **Access Control** for channel traffic

3. **Policy** management

---

**Diagram:***
- **Principal**
- **Object**
- **Reference monitor**
- **Guard**
- **Policy**
- **Audit log**

**Connections:**
- Authentication
- Authorization

**Annotations:**
- 1. Isolation boundary
- 2. Access control
- 3. Policy
Access Control Mechanisms: The Gold Standard

- **Authenticate** principals: Who made a request
  - Mainly people, but also channels, servers, programs (encryption implements channels, so key is a principal)

- **Authorize** access: Who is trusted with a resource
  - *Group* principals or resources, to simplify management
    - Can define by a property, e.g. “type-safe” or “safe for scripting”

- **Audit**: Who did what when?

  - **Lock** = **Authenticate** + **Authorize**
  - **Deter** = **Authenticate** + **Audit**
Making Isolation Work

• Isolation is imperfect: Can’t get rid of bugs
  – TCB = 10-50 M lines of code
  – Customers want features more than correctness
• Instead, don’t tickle them.
• How? Reject bad inputs
  – Code: don’t run or restrict severely
  – Communication: reject or restrict severely
    • Especially web sites
  – Data: don’t send; don’t accept if complex
Bad = Unaccountable

- Can’t identify bad guys, so can’t deter them
- Fix? End nodes enforce accountability
  - Refuse inputs that aren’t accountable enough
    - or strongly isolate those inputs
  - Senders are accountable if you can punish them
  - *All trust is local*
- Need an ecosystem for
  - Senders becoming accountable
  - Receivers demanding accountability
  - Third party intermediaries
- To stop DDOS attacks, ISPs must play
For Accountability To Work

• Senders must be able to make themselves accountable
  – This means pledging something of value
    • Friendship
    • Reputation
    • Money
    • ...

• Receivers must be able to check accountability
  – Specify what is accountable enough
  – Verify sender’s evidence of accountability
Accountability vs. Access Control

• “In principle” there is no difference but
• Accountability is about punishment, not locks – Hence audit is critical
• Accountability is very coarse-grained
The Accountability Ecosystem

- Identity, reputation, and indirection services
- Mechanisms to establish trust relationships
  - Person to person and person to organization
- A flexible, simple user model for identity
- Stronger user authentication
  - Smart card, cell phone, biometrics
- Application identity: signing, reputation
Accountable Internet Access

• Just enough to block DDoS attacks
• Need ISPs to play. Why should they?
  – Servers demand it; clients don’t get locked out
  – Regulation?

• A server asks its ISP to block some IP addresses
• ISPs propagate such requests to peers or clients
  – Probably must be based on IP address
  – Perhaps some signing scheme to traverse unreliable intermediaries?
• High priority packets can get through
Accountability vs. Freedom

• Partition world into two parts:
  – Green Safer/accountable
  – Red Less safe/unaccountable

• Two aspects, mostly orthogonal
  – User Experience
  – Isolation mechanism
    • Separate hardware with air gap
    • VM
    • Process isolation
Without R|G: Today

- Less trustworthy
  - Less accountable entities

- More trustworthy
  - More accountable entities

N attacks/yr

(N >> m)

m attacks/yr

My Computer

- Less valuable assets
- More valuable assets

Total: N+m attacks/yr on all assets

Entities
- Programs
- Network hosts
- Administrators
With R|G

N attacks/yr on less valuable assets

(My Red Computer)

Less trustworthy
Less accountable entities

Less valuable assets

M attacks/yr on more valuable assets

(My Green Computer)

More trustworthy
More accountable entities

More valuable assets

(N >> m)

Entities
- Programs
- Network hosts
- Administrators
Must Get Configuration Right

- Keep valuable stuff out of red
- Keep hostile agents out of green

Less trustworthy
Less accountable entities

More trustworthy
More accountable entities

My Red Computer
- Valuable Asset
- Less valuable assets

My Green Computer
- More valuable assets
- Hostile agent
Why R|G?

• Problems:
  – Any OS will always be exploitable
    • The richer the OS, the more bugs
  – Need internet access to get work done, have fun
    • The internet is full of bad guys

• Solution: Isolated work environments:
  – Green: important assets, only talk to good guys
    • Don’t tickle the bugs, by restricting inputs
  – Red: less important assets, talk to anybody
    • Blow away broken systems

• Good guys: more trustworthy / accountable
  – Bad guys: less trustworthy or less accountable
Configuring Green

• Green = locked down = only whitelist inputs
• Requires professional management
  – Few users can make these decisions
  – Avoid “click OK to proceed”
• To escape, use Red
  – Today almost all machines are Red
R|G User Model Dilemma

• People don’t want complete isolation
  – They want to:
    • Cut/paste, drag/drop
    • Share parts of the file system
    • Share the screen
    • Administer one machine, not multiple
    • …

• But more integration can weaken isolation
  – Add bugs
  – Compromise security
Data Transfer

• Mediates data transfer between machines
  – Drag / drop, Cut / paste, Shared folders

• Problems
  – Red → Green : Malware entering
  – Green → Red : Information leaking

• Possible policy
  – Allowed transfers (configurable). Examples:
    • No transfer of “.exe” from R to G
    • Only transfer ASCII text from R to G
  – Non-spoofable user intent; warning dialogs
  – Auditing
    • Synchronous virus checker; third party hooks, ...
Where Should Email/IM Run?

• As productivity applications, they must be well integrated in the work environment (green)

• Threats—A tunnel from the bad guys
  – Executable attachments
  – Exploits of complicated data formats

• Choices
  – Run two copies, one in Green and one in Red
  – Run in Green and mitigate threats
    • Green platform does not execute arbitrary programs
    • Green apps are conservative in the file formats they accept
  – Route messages to appropriate machine
R|G and Enterprise Networks

- Red and green networks are defined as today:
  - IPSEC
  - Guest firewall
  - Proxy settings
  - ...

- The VMM can act as a router
  - E.g. red only talks to the proxy
Summary

• Security is about risk management
  – Cost of security < expected loss

• Security relies on deterrence more than locks
  – Deterrence requires the threat of punishment
  – This requires accountability

• Accountability needs an ecosystem
  – Senders becoming accountable
  – Receivers verifying accountability

• Accountability limits freedom
  – Beat this by partitioning: red | green
  – Don’t tickle bugs in green, dispose of red