Principles of Security

[Saltzer and Schroeder, The Protection of Information in Computer Systems, 1975]

• Accountability
• Complete Mediation
• Least Privilege
• Failsafe Defaults
• Separation of Privilege
• Defense in Depth
• Economy of Mechanism
• Open Design
• Psychological Acceptability

(Update to A1: made the list in P5 match this list)
EXERCISE: BANK BINGO
Accountability

Hold principals responsible for their actions
Accountability

Hold principals responsible for their actions

• **Authorization**: mechanisms that govern whether actions are permitted

• **Authentication**: mechanisms that bind principals to actions

• **Audit**: mechanisms that record and review actions
Accountability

Hold principals responsible for their actions

- **Authorization**: mechanisms that govern whether actions are permitted
- **Authentication**: mechanisms that bind principals to actions
- **Audit**: mechanisms that record and review actions

... Gold Standard [Lampson 2004]
Complete Mediation

Every operation requested by a principal must be intercepted and determined to be acceptable according to the security policy
Complete Mediation

Every operation requested by a principal must be intercepted and determined to be acceptable according to the security policy.

- Component that does the interception and determination is the reference monitor.
- Related to Accountability.
- Restricts caching of information, including previous decisions.
Least Privilege

Principals should be given the minimum privileges necessary to accomplish their task

• Limits the damage that can result from accident or malice
• Cf. "need to know"
Failsafe Defaults

Base decisions on the presence of privilege, not the absence of prohibition

• The default answer is "no"
• Say "yes" only when there is an explicit reason to do so
• Principals who discover they don't have access will complain
• Attackers who discover they do have access won't complain!
Failsafe Defaults

Java stack inspection circa 1998:

```java
checkPermission(T) {
    // loop newest to oldest stack frame
    foreach stackFrame {
        if (local policy forbids access to T by class executing in
            stack frame) throw ForbiddenException;

        if (stackFrame has enabled privilege for T)
            return;  // allow access

        if (stackFrame has disabled privilege for T)
            throw ForbiddenException;
    }

    // end of stack
    if (Netscape || ...) throw ForbiddenException;
    if (MS IE4.0 || JDK 1.2 || ...) return;
}
```
Separation of Privilege

• Different operations should require different privileges

• Supports Least Privilege

• In tension with usability: too many operations and objects and principals
Separation of Privilege

• Different operations should require different privileges
• Disseminate privileges for an operation amongst multiple principals (Separation of Duty)
Defense in Depth

Prefer a set of complementary mechanisms over a single mechanism

Complementary:

• **Independent**: attack that compromises one mechanism is unlikely to compromise others

• **Overlapping**: attacks must compromise multiple mechanisms to succeed
Economy of Mechanism

Prefer mechanisms that are simpler and smaller

• Easier to understand, construct, analyze
• Hence less likely to have unknown vulnerabilities
• Applies to any aspect of system, not just security

Trusted computing base (TCB): mechanisms that implement the core security functionality

...keep the TCB small
Open Design

Security shouldn't depend upon the secrecy of design or implementation

```c
/* efdtt.c Author: Charles M. Hannum <root@ihack.net> */

#define m(i)(x[i]^s[i+84])<<

unsigned char x[5],y,s[2048];main(n){for(read(0,x,5);read(0,s,n=2048);write(1,s ,n))if(s[y=s[13]%8+20]/16%4==1){int i=m(1)17^256+m(0)8,k=m(2)0,j=m(4)17^m(3)9^k *2-k%8^8,a=0,c=26;for(s[y]=16;-c;j*=2)a=a*2^i&1,i=i/2^j&1<<24;for(j=127;++j<n ;c=c>y)c+=y=i^i/8^i>>4^i>>12,i=i>>8^y<<17,a^=a>>14,y=a^a*8^a<<6,a=a>>8^y<<9,k=s [j],k="7Wo~'G_\216"[k&7]+2^"cr3sfw6v;*k+/n."[k>4]*2^k*257/8,s[j]=k^x(k&k*2&34) *6^c~y;}}
```
Open Design

Security shouldn't depend upon the secrecy of design or implementation

Arguments for open design:
• Secrets eventually come out: reverse engineering is possible, employees move around
• Making details public increases chance of identifying and repairing vulnerabilities
Open Design

Security shouldn't depend upon the secrecy of design or implementation

Arguments against open design:

• Secrecy supports Defense in Depth by making it harder to find vulnerabilities

• Lack of hard evidence that Linus' Law really holds ("given enough all eyeballs, all bugs are shallow")

• After identification, some vulnerabilities cannot quickly or easily be repaired
Psychological Acceptability

Minimize the burden of security mechanisms on humans

- Don't make operations (much) more difficult to complete than if security mechanisms were absent
- Don't make configuration difficult
- Produce comprehensible error messages

...always a tradeoff between security and usability
Principles of Security

• Accountability
• Complete Mediation
• Least Privilege
• Failsafe Defaults
• Separation of Privilege
• Defense in Depth
• Economy of Mechanism
• Open Design
• Psychological Acceptability
EXERCISE: BANK BINGO
EXERCISE: CHROMIUM BROWSER

(see CMS for files; do at home; we'll discuss it at beginning of next class)
Upcoming events

• [Mon] A1 due

"Important principles may, and must, be inflexible."
– Abraham Lincoln