Lecture 5: Principles of Security

CS 181S

September 19, 2018

Last Time

1. Functional requirements
2. Threat analysis
3. Harm analysis
4. Security goals
5. Feasibility analysis
6. Security requirements

Example: ELEANOR'S PLACE

- New restaurant in Claremont
- You are contracted to build a system for online reservations
- What are the functionality requirements for this reservation system?
- What is the threat model?
- What confidentiality, integrity, and availability harms does Eleanor's Place face?
- What security goals should it have?
- Are they feasible? How could these be refined to security requirements?
- What countermeasures could be employed to implement security requirements?

Countermeasures

Week	Day	Date	Topic	Reading	Due
			Introduction to Security		
1	**Wed**	**Sep 5**	Security Policies [slides] [pdf]	[Bishop, Ch. 1.1-1.3]	
2	**Mon**	**Sep 10**	Vulnerabilities [slides] [pdf]	[Notes] [Review Exercises]	
	Wed	**Sep 12**	Threat Models [slides] [pdf]		**T1 DUE**
3	**Mon**	**Sep 17**	Beyond Threats		
	Wed	**Sep 19**	Security Principles		**A1 DUE**
4	**Mon**	**Sep 24**	Security Principles	[Schneider, Ch. 1], [Saltzer-Schroeder]	
			Cryptography		
	Wed	**Sep 26**	Symmetric Crypto		**M0 DUE**
5	**Mon**	**Oct 1**	Public-Key Cryptography		
	Wed	**Oct 3**	Secure Channels		**M1 DUE**
6	**Mon**	**Oct 8**	Protocol Design		
	Wed	**Oct 10**	Key Management		**A2 DUE**
7	**Mon**	**Oct 15**	Applied Cryptography		
			Authentication		
	Wed	**Oct 17**	Human Authentication	[Schneider, Ch. 5]	**T2 DUE**
8	Mon	Oct 22	*Fall Recess, No Class*		
	Wed	**Oct 24**	Passwords		**M2 DUE**
9	**Mon**	**Oct 29**	Tokens		
	Wed	**Oct 31**	Certificates		**A3 DUE**
			Authorization		
10	**Mon**	**Nov 5**	DAC	[Schneider, Ch. 7.3]	
	Wed	**Nov 7**	Capabilities	[Schneider, Ch. 7.3] [Bishop, Ch. 14.2]	**T3 DUE**
11	**Mon**	**Nov 12**	MLS	[Bishop, Ch. 5-6.2]	
	Wed	**Nov 14**	Information Flow	[notes] [Sabelfeld-Sands]	**M3 DUE**
12	**Mon**	**Nov 19**	IFC (cont'd)	[notes]	
	Wed	Nov 21	*Thanksgiving, No Class*		
			Audit		
13	**Mon**	**Nov 26**	Logs		
	Wed	**Nov 28**	Blockchains		**T4 DUE**
			Applied Security		
14	**Mon**	**Dec 3**	Network Security	[notes]	
	Wed	**Dec 5**	Web Security	[notes]	**M4 DUE**
15	**Mon**	**Dec 10**	Privacy	[notes]	
	Wed	**Dec 12**	Trusted Hardware	[notes]	

Countermeasures

A defense that protects against attacks by neutralizing either the threat or vulnerability involved

Strategy:

- Prevent: block attack or close vulnerability — Prevention
- Deter: make attack harder
- Deflect: make other targets more attractive
- Mitigate: make harm less severe

} Risk Management

- Detect: as it happens or after the fact
- Recover: undo harm

} Deterrence through Accountability

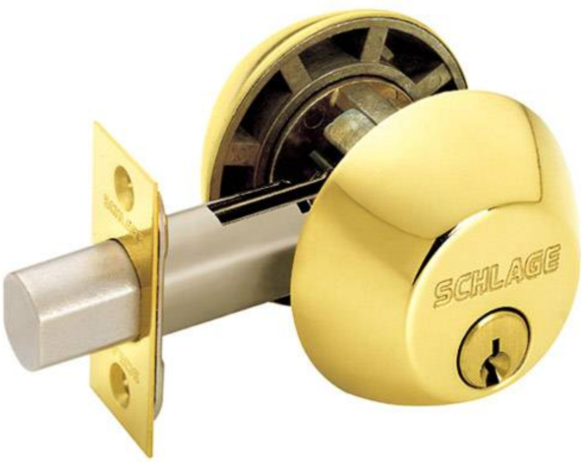
Principles of Prevention

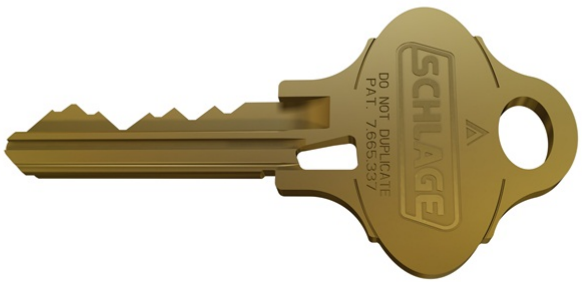
[Saltzer and Schroeder, *The Protection of Information in Computer Systems*, 1975]

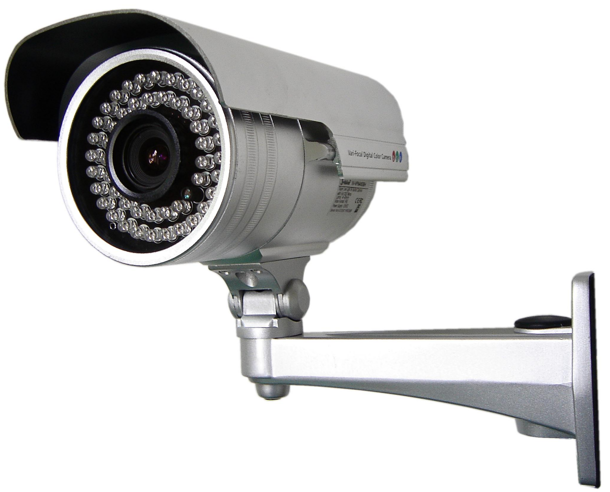
- Accountability
- Complete Mediation
- Least Privilege
- Failsafe Defaults
- Separation of Privilege
- Defense in Depth
- Economy of Mechanism
- Open Design
- Psychological Acceptability

Accountability

Hold principals responsible for their actions



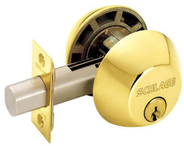


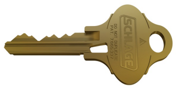


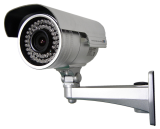
Accountability

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





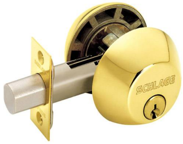


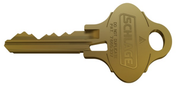
Accountability

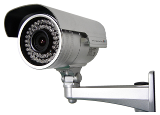
Hold principals responsible for their actions

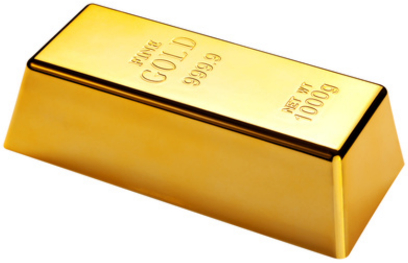
- **Au****thorization:** mechanisms that govern whether actions are permitted
- **Au****thentication:** mechanisms that bind principals to actions
- **Au****dit:** mechanisms that record and review actions

  ... Gold Standard [Lampson 2000]









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

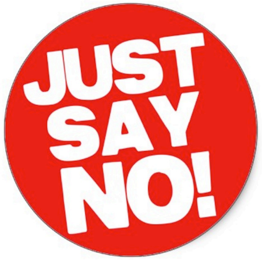
Least Privilege

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!

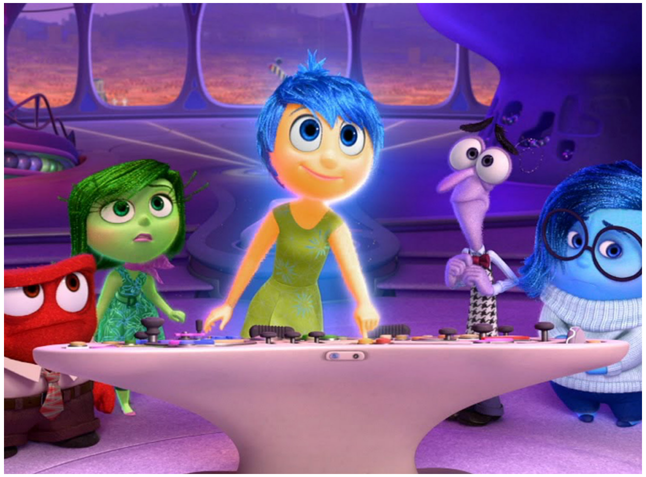
Failsafe Defaults

Java stack inspection circa 1998:

```
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
- Disseminate privileges for an operation amongst multiple principals (Separation of Duty)

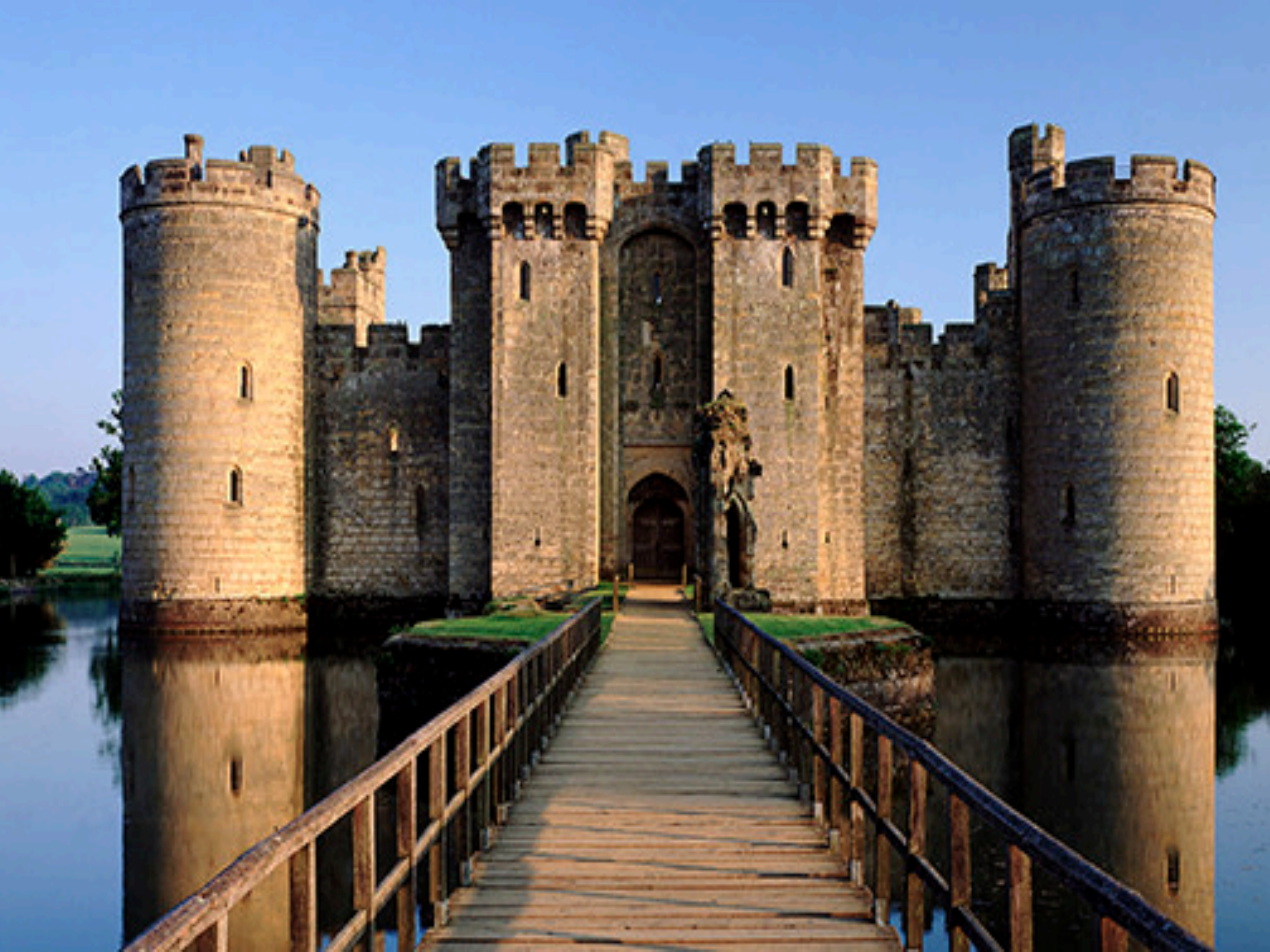


Defense in Depth

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



Exercise

- Consider the security mechanisms deployed in your dorm and/or in academic buildings on campus. These systems are designed to prevent access by unauthorized people.
  - To what extent do those security features enforce Complete Mediation?
  - To what extent do those security features enforce Least Privilege?
  - To what extent do those security features satisfy the independence requirement of Defense in Depth?
  - To what extent do those security features satisfy the overlapping requirement of Defense in Depth?

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

```
/*     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^(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

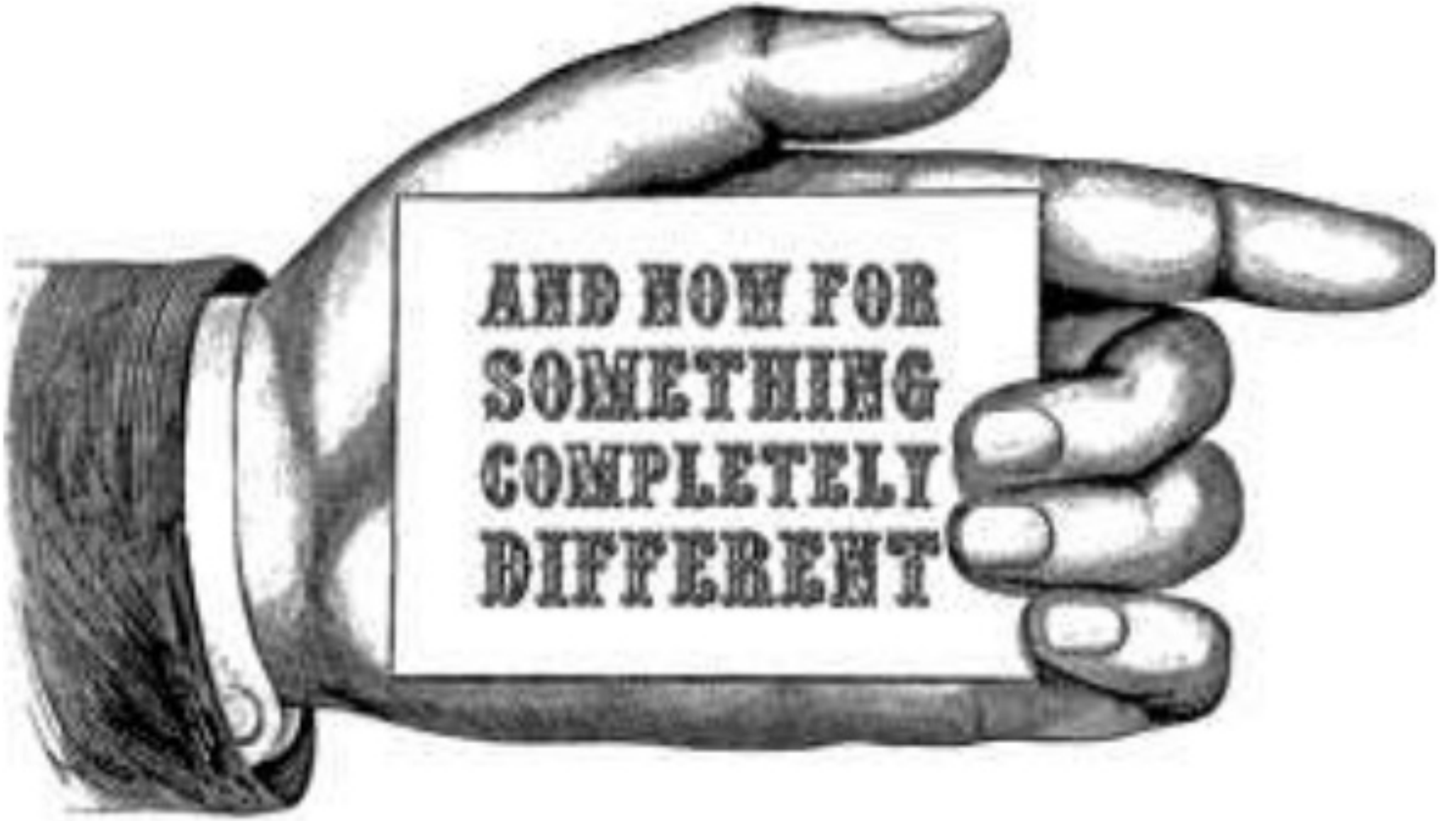
...always a tradeoff between security and usability

Alternative `su`

- A student who didn't take 181S decides to build a new version of `su` that works as follows:
  - If the open operation succeeds, then the password is checked. If it is indeed the correct password for `u2`, then `u1` is granted access to the account of `u2`.
  - If the open operation fails, then `u1` immediately is granted access to the account of the superuser ("`root`"). The student's intention is that `u1` would then be able to fix the misconfiguration.

Principles of Security

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


AND NOW FOR
SOMETHING
COMPLETELY
DIFFERENT

What skills will your project need?

Forming a group…

- What skills/experience will you bring to a group?
- What system(s) are you exciting about building?
- How challenging do you want your project to be?
- How often/when are you available to meet?