April 8, 2005 -- Lecture 30

22C:169
Computer Security

Douglas W. Jones

Department of Computer Science

Administrative Issues

Warning

Security is a system property

An emergent property of the whole

Insecurity can result from

insecure components or
insecure construction of system or
insecure administration of system

Security rarely emerges by accident

Adding a security module to a system rarely achieves much!

Security Policies

Unstated security policies invite accident

Security policy should be explicit

Security policy evaluation is critical

Oversights in policies are dangerous

Cannot implement contradictory policies

Cannot enforce impossibilities

Security Policy Content:

The obvious:

The access matrix.

Easy for static resources and users

Policy for adding resources and users.

Much more difficult

Goals, responsibility and commitment.

What are our priorities

Who is responsible

What resources are available

Security requirements

Derived from policy

Multiple sets of requirements could work

For any particular system, pick one set.

Functional requirements

This function must exist here.

eg: What data to encrypt

Performance requirements

This function must operate this well.

eg: How strongly encrypted

TCSEC Requirements

1) There must be a policy
2) Subjects and objects must be identified
3) All objects must be marked with level
4) Log of all actions that can affect security
5) Assurance by enforcement mechanisms
6) Mechanisms must be protected

Security evaluation

Given a statement of security policy

evaluation is possible

Security evaluation tests the hypothesis:

This system is secure.

Security evaluation is like natural science:

Proof of insecurity is possible.

Proof of security is not possible.

"This system is secure" means
"We have yet to find an insecurity."

Warning

Security evaluation is inadequate if it only

checks implementation of requirements
or checks that requirements meet policy

Security failure is

Failure of system to meet policy goals!

Implementors can say

But we implemented the requirements

This is passing the buck

Policy must pin down responsibility

An example failure

County election officer is

responsible for election system integrity
generally not a technical person

Election systems contain many options

setting these wrong can destroy integrity
setting these is highly technical

In the news April 1
Miami Election Head Forced Out

Security is a Fragile Property

Innocent changes may destroy security

Adding functionality is dangerous!

Microsoft Visual Basic example:
Policy directly created virus medium!

Therefore, security evaluation

must not be a one-time effort

must be continuous or periodic