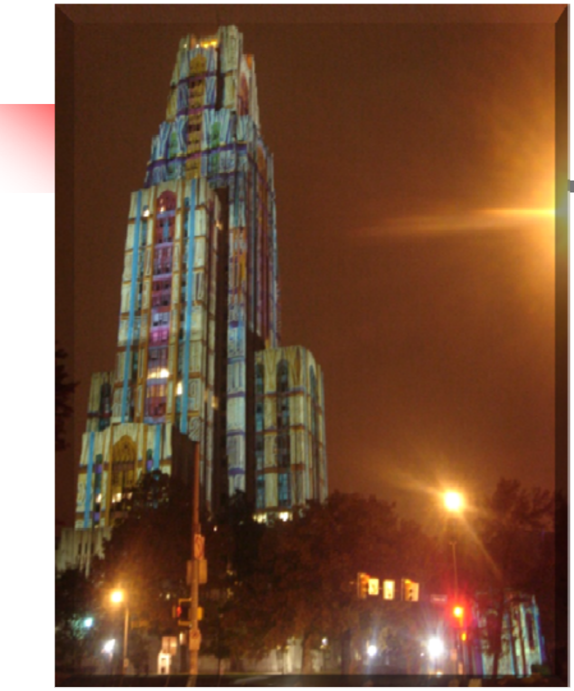
IS 2150 / TEL 2810
Information Security & Privacy



James Joshi
Professor, SIS

Lecture 9.1
March 23, 2016

Vulnerability Analysis
Risk Management

Objectives

- Understand/explain the issues related to, and utilize the techniques
  - Vulnerability analysis/classification
    - Techniques
    - Taxonomy
  - Security risks management

Vulnerability Analysis

- Vulnerability or security flaw: specific failures of security controls (procedures, technology or management)
  - Errors in code
  - Human violators
  - Mismatch between assumptions
- Exploit: Use of vulnerability to violate policy
- Attacker: Attempts to exploit the vulnerability

Techniques for Detecting Vulnerabilities

- System Verification
  - Determine preconditions, post-conditions
  - Validate that system ensures post-conditions given preconditions

  Can prove the absence of vulnerabilities
- Penetration testing
  - Start with system/environment characteristics
  - Try to find vulnerabilities

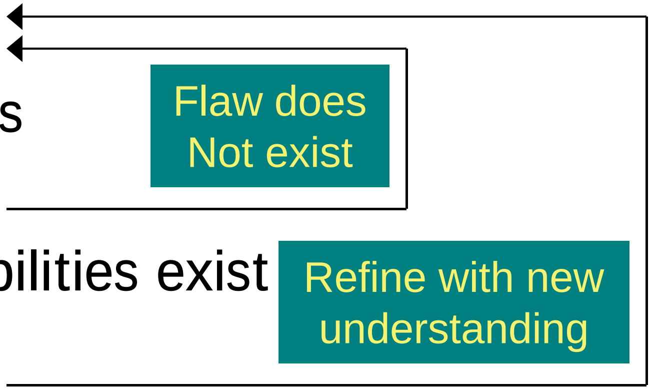
  Can not prove the absence of vulnerabilities

Types/layers of Penetration Testing

- Black Box (External Attacker)
  - External attacker has no knowledge of target system
  - Attacks built on human element – Social Engineering
- System access provided (External Attacker)
  - Red team provided with limited access to system
  - Goal is to gain normal or elevated access
- Internal attacker
  - Red team provided with authorized user access
  - Goal is to elevate privilege / violate policy

Red Team Approach
Flaw Hypothesis Methodology:

- Information gathering
  - Examine design, environment, system functionality
- Flaw hypothesis
  - Predict likely vulnerabilities
- Flaw testing
  - Determine where vulnerabilities exist
- Flaw generalization
  - Attempt to broaden discovered flaws
- Flaw elimination (often not included)
  - Suggest means to eliminate flaw


Flaw does Not exist

Refine with new understanding

Problems with Penetration Testing

- Nonrigorous
  - Dependent on insight (and whim) of testers
  - No good way of evaluating when “complete”
- How do we make it systematic?
  - Try all classes of likely flaws
  - *But what are these?*
- Vulnerability Classification!

Vulnerability Classification

- Goal: describe spectrum of possible flaws
  - Enables design to avoid flaws
  - Improves coverage of penetration testing
  - Helps design/develop intrusion detection
- How do we classify?
  - By how they are exploited?
  - By where they are found?
  - By the nature of the vulnerability?

Example flaw: xterm log

- *xterm* runs as root
    - Generates a log file
    - Appends to log file if file exists
- Problem: ln /etc/passwd log_file
- Solution

```
if (access(“log_file”, W_OK) == 0)
    If ((fd = open(“log_file”, O_WRONLY|O_APPEND)) < 0) {
    -  error handling
    }
```

What can go wrong?

Example: Finger Daemon *(exploited by Morris worm)*

- *finger* sends name to *fingerd*
  - *fingerd* allocates 512 byte buffer on stack
  - Places name in buffer
  - Retrieves information (local finger) and returns
- Problem: If name > 512 bytes, overwrites return address
- Exploit: Put code in “name”, pointer to code in bytes 513+
  - Overwrites return address

RISOS:Research Into Secure Operating Systems (7 Classes)

1. Incomplete parameter validation
   - E.g., buffer overflow –
2. Inconsistent parameter validation
   - Different routines with different formats for same data
3. Implicit sharing of privileged / confidential data
   - OS fails to isolate processes and users
4. Asynchronous validation / inadequate serialization
   - Race conditions and TOCTTOU flaws
5. Inadequate identification /authentication / authorization
   - Trojan horse; accounts without passwords
6. Violable prohibition / limit
   - Improper handling of bounds conditions (e.g., in memory allocation)
7. Exploitable logic error
   - Incorrect error handling, incorrect resource allocations etc.

Protection Analysis Model Classes

- Pattern-directed protection evaluation
  - Methodology for finding vulnerabilities
- Applied to several operating systems
  - Discovered previously unknown vulnerabilities
- Resulted in two-level hierarchy of vulnerability classes
  - Ten classes in all

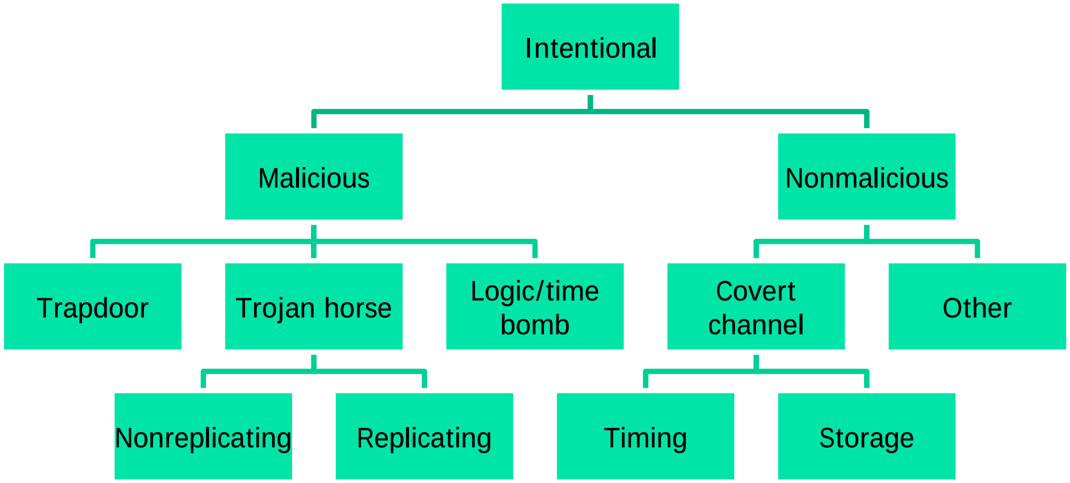
PA flaw classes

1. Improper protection domain initialization and enforcement
   a. *domain*: Improper choice of initial protection domain
   b. *exposed representations*: Improper isolation of implementation detail (Covert channels)
   c. *consistency of data over time*: Improper change
   d. *naming*: Improper naming (two objects with same name)
   e. *residuals*: Improper deallocation or deletion
2. Improper validation *validation of operands, queue management dependencies*:
3. Improper synchronization
   a. *interrupted atomic operations*: Improper indivisibility
   b. *serialization*: Improper sequencing
4. Improper choice of operand or operation *critical operator selection errors*

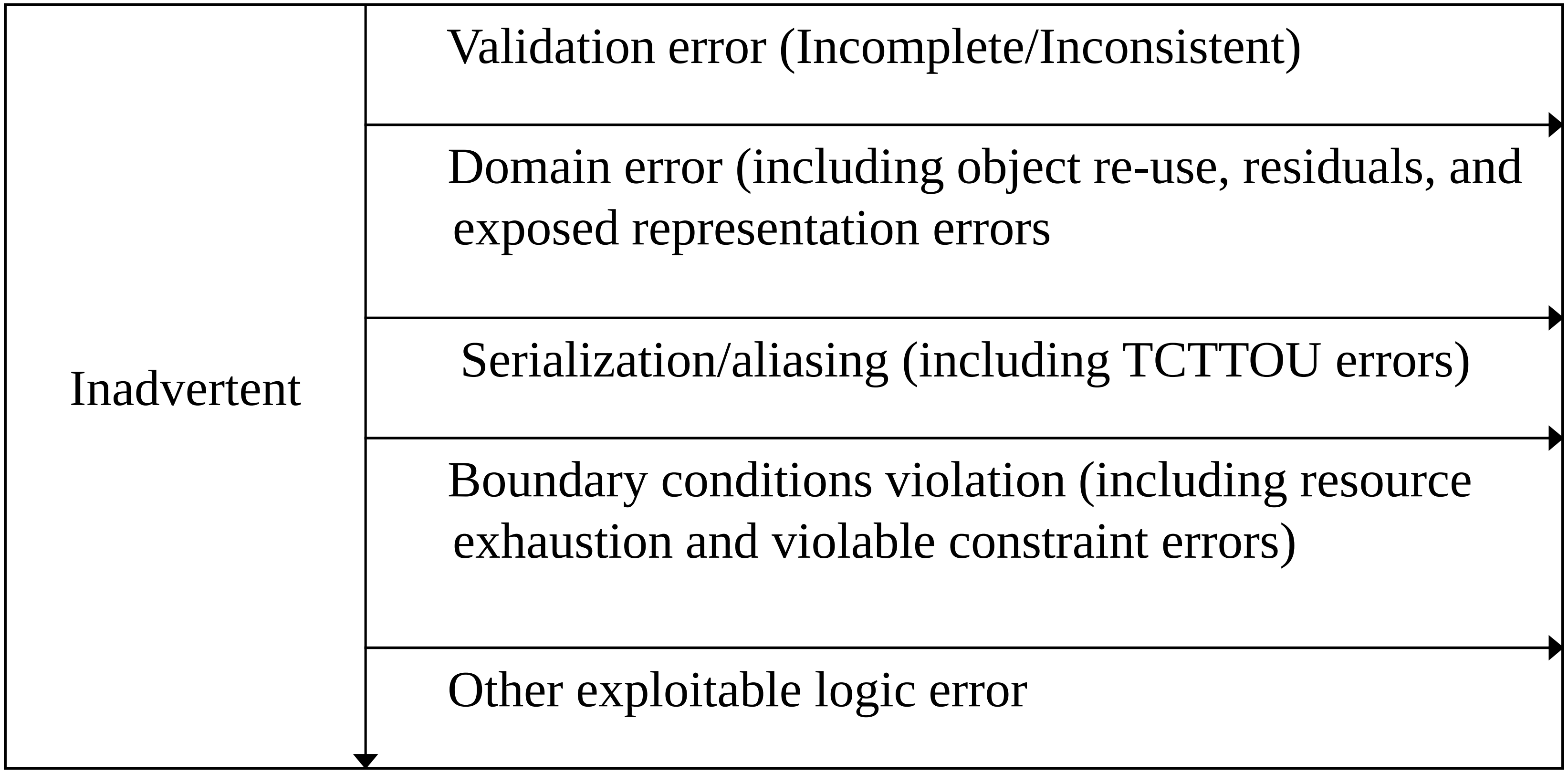
NRL Taxonomy

- Three classification schemes
  - How did it enter
  - When was it "created"
  - Where is it

Genesis



NRL Taxonomy (Genesis)



Inadvertent

Validation error (Incomplete/Inconsistent)

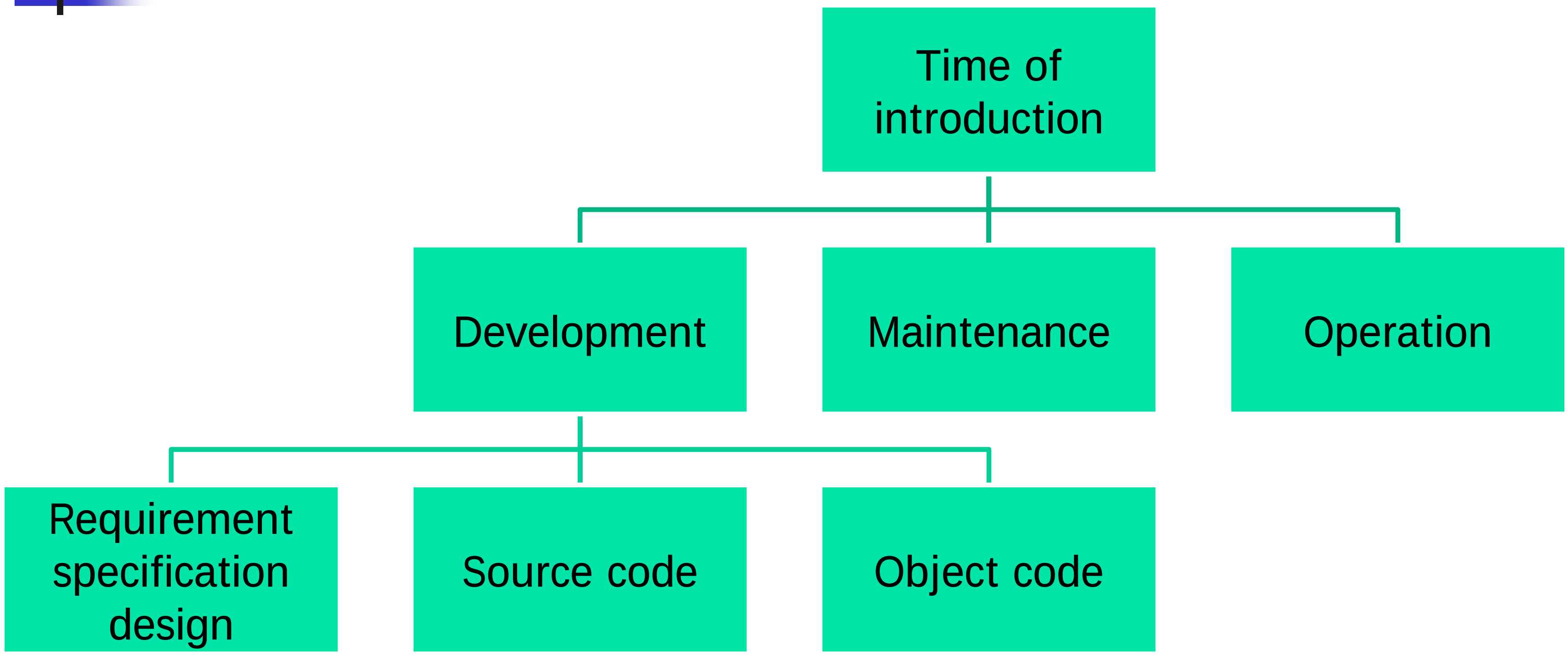
Domain error (including object re-use, residuals, and exposed representation errors

Serialization/aliasing (including TCTTOU errors)

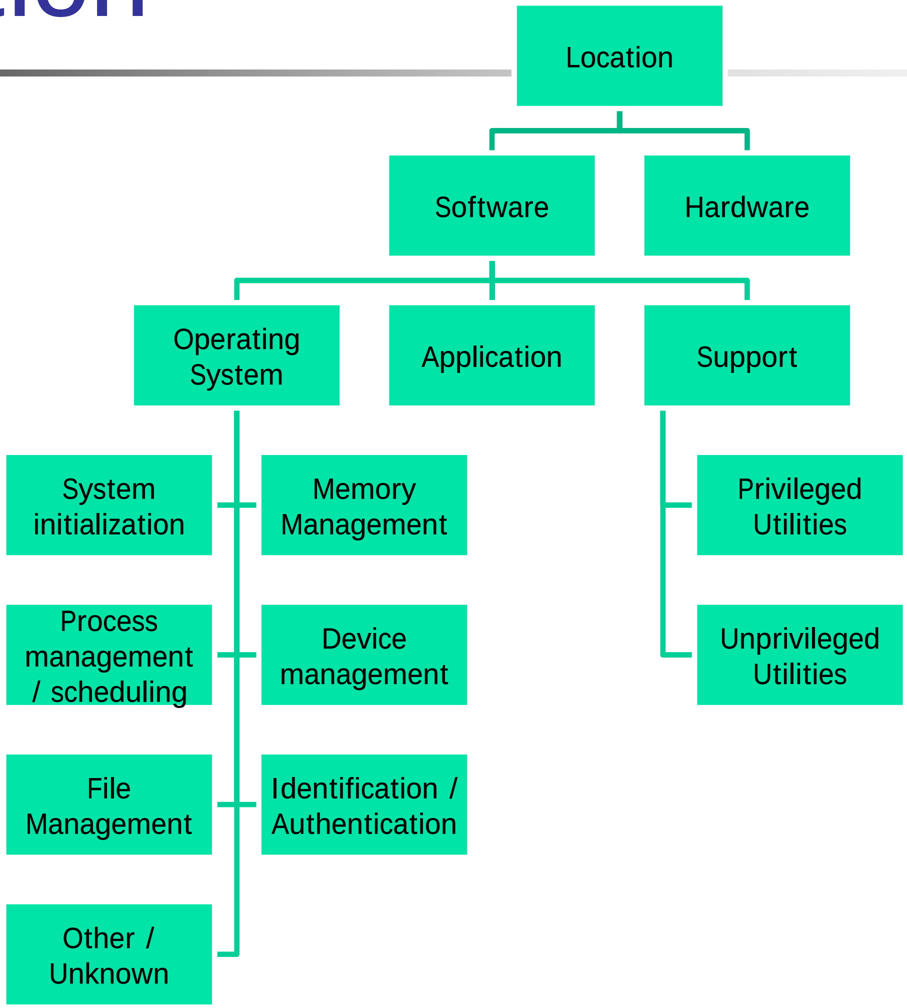
Boundary conditions violation (including resource exhaustion and violable constraint errors)

Other exploitable logic error

NRL Taxonomy: Time



NRL Taxonomy: Location



Aslam’s Model

- Attempts to classify faults unambiguously
  - Decision procedure to classify faults
- Coding Faults
  - Synchronization errors
    - Timing window
    - Improper serialization
  - Condition validation errors
    - Bounds not checked
    - Access rights ignored
    - Input not validated
    - Authentication / Identification failure
- Emergent Faults
  - Configuration errors
    - Wrong install location
    - Wrong configuration information
    - Wrong permissions
  - Environment Faults

Common Vulnerabilities and Exposures (cve.mitre.org)

- Captures *specific* vulnerabilities
  - Standard name
  - Cross-reference to CERT, etc.
- Entry has three parts
  - Unique ID
  - Description
  - References

Name	CVE-1999-0965
Description	Race condition in xterm allows local users to modify arbitrary files via the logging option.

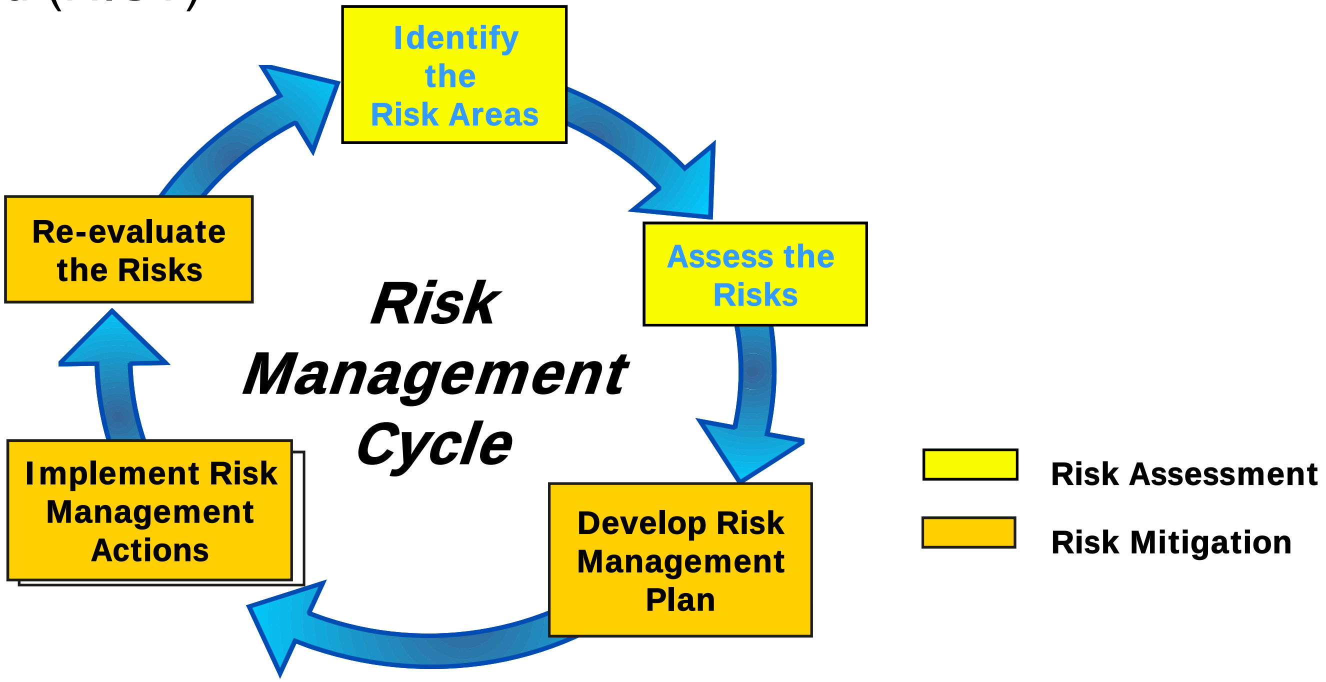
References
•CERT:CA-93.17
•XF:xterm

Risk Analysis

Overview of
Risk, Cost-benefit analysis

Risk Management

- The process concerned with identification, measurement, control and minimization of security risks in information systems to a level commensurate with the value of the assets protected (NIST)



Identify the Risk Areas

Assess the Risks

Develop Risk Management Plan

Implement Risk Management Actions

Re-evaluate the Risks

Risk Management Cycle

🟨 Risk Assessment

🟧 Risk Mitigation

Risk

- The *likelihood* that a particular *threat* using a specific *attack*, will exploit a particular *vulnerability* of a system that results in an undesirable *consequence* (NIST)
  - *Likelihood* of the threat occurring is the estimation of the probability that a threat will succeed in achieving an undesirable event

Risk Assessment/Analysis

- A process of analyzing *threats* to and *vulnerabilities* of an information system and the *potential impact* the loss of information or capabilities of a system would have
  - List the threats and vulnerabilities
  - List possible control and their cost
  - Do cost-benefit analysis
    - Is cost of control more than the expected cost of loss?
- The resulting analysis is used as a basis for identifying appropriate and cost-effective counter-measures
  - Leads to proper security plan

Risk Assessment steps

- Identify assets
  - Hardware, software, data, people, supplies
- Determine vulnerabilities
  - Intentional errors, malicious attacks, natural disasters
- Estimate likelihood of exploitation
  - Considerations include
    - Presence of threats
    - Tenacity/strength of threats
    - Effectiveness of safeguards
  - Delphi approach
    - Raters provide estimates that are distributed and re-estimated

Risk Assessment steps (2)

- Compute expected annual loss
  - Physical assets can be estimated
  - Data protection for legal reasons
- Survey applicable (new) controls
  - If the risks of unauthorized access is too high, access control hardware, software and procedures need to be re-evaluated
- Project annual savings of control

Example 1

- Risks:
  - disclosure of company confidential information,
  - computation based on incorrect data
- Cost to correct data: $1,000,000
  - @10% liklihood per year: $100,000
  - Effectiveness of access control sw:60%: -$60,000
  - Cost of access control software: +$25,000
  - Expected annual costs due to loss and controls:
    - $100,000 - $60,000 + $25,000 = $65,000
  - Savings:
    - $100,000 - $65,000 = $35,000

Example 2

- Risk:
  - Access to unauthorized data and programs
    - 100,000 @ 2% likelihood per year: $2,000
  - Unauthorized use of computing facility
    - 100,000 @ 40% likelihood per year: $4,000
- Expected annual loss:

  $6,000
- Effectiveness of network control: 100%

  -$6,000

Example 2 (2)

Control cost	
Hardware	+$10,000
Software	+$4,000
Support personnel	+$40,000
Annual cost:	+$54,000
Expected annual cost	
(6000-6000+54000)	+$54,000
Savings	
(6000 – 54,000)	-$48,000

Some Arguments against Risk Analysis

- Not precise
  - Likelihood of occurrence
  - Cost per occurrence
- False sense of precision
  - Quantification of cost provides false sense of security
- Immutability
  - Filed and forgotten!
  - Needs annual updates
- No scientific foundation (not true)
  - Probability and statistics

Summary

- Vulnerability Analysis – taxonomy
- Risk Management – cost benefit analysis