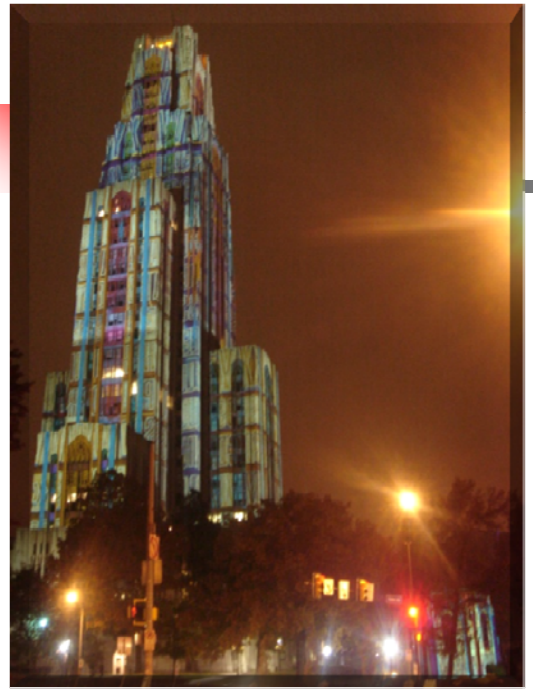
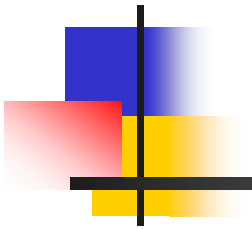
IS 2150 / TEL 2810
Introduction to Security



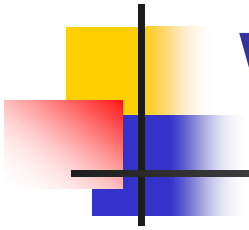
James Joshi
Associate Professor, SIS

Lecture 11 continue
Nov 29, 2010

Malicious Code,
Risk Analysis

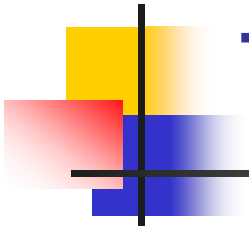


Malicious Code



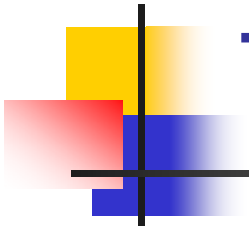
What is Malicious Code?

- Set of instructions that causes a security policy to be violated
  - unintentional mistake
  - Tricked into doing that?
  - “unwanted” code
- Generally relies on “legal” operations
  - Authorized user *could* perform operations without violating policy
  - Malicious code “mimics” authorized user



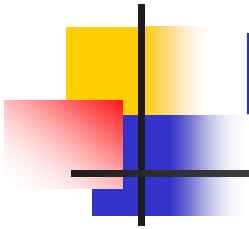
Types of Malicious Code

- Trojan Horse
  - What is it?
- Virus
  - What is it?
- Worm
  - What is it?



Trojan Horse

- Program with an overt (expected) and covert (unexpected) effect
  - Appears normal/expected
  - Covert effect violates security policy
- User tricked into executing Trojan horse
  - Expects (and sees) overt behavior
  - Covert effect performed with user's authorization
- Trojan horse may replicate
  - Create copy on execution
  - Spread to other users/systems



Example

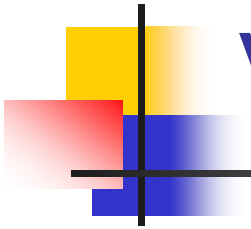
- *Perpetrator*

```
cat >/homes/victim/ls <<eof
cp /bin/sh /tmp/.xxsh
chmod u+s,o+x /tmp/.xxsh
rm ./ls
ls $*
eof
```

- *Victim*

```
ls
```

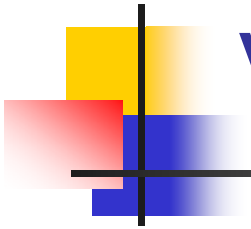
- What happens?
- How to replicate this?



Virus

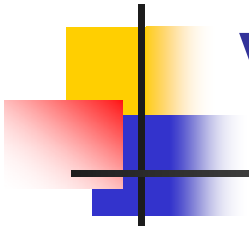
- Self-replicating code
  - A freely propagating Trojan horse
    - some disagree that it is a Trojan horse
  - Inserts itself into another file
    - Alters normal code with “infected” version
- Operates when infected code executed

```
If spread condition then
    For target files
        if not infected then alter to include virus
Perform malicious action
Execute normal program
```



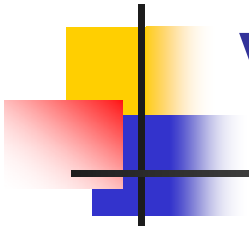
Virus Types

- Boot Sector Infectors (The Brain Virus)
  - Problem: How to ensure virus “carrier” executed?
  - Solution: Place in boot sector of disk
    - Run on any boot
  - Propagate by altering boot disk creation
- Executable infector
  - The Jerusalem Virus, Friday 13th, not 1987
- Multipartite virus : boot sector + executable infector



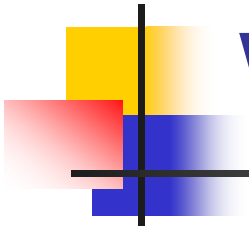
Virus Types/Properties

- Terminate and Stay Resident
  - Stays active in memory after application complete
  - Allows infection of previously unknown files
- Stealth (an executable infector)
  - Conceal Infection
- Encrypted virus
    - Prevents “signature” to detect virus
    - [Deciphering routine, Enciphered virus code, Deciphering Key]
- Polymorphism
    - Change virus code to something equivalent each time it propagates



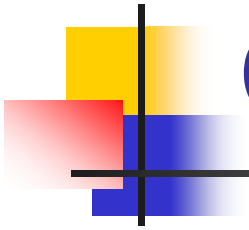
Virus Types/Properties

- Macro Virus
  - Composed of a sequence of instructions that is interpreted rather than executed directly
  - Infected “executable” isn’t machine code
    - Relies on something “executed” inside application
    - Example: Melissa virus infected Word 97/98 docs
- Otherwise similar properties to other viruses
  - Architecture-independent
  - Application-dependent



Worms

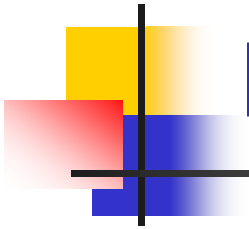
- Replicates from one computer to another
  - Self-replicating: No user action required
  - Virus: User performs “normal” action
  - Trojan horse: User tricked into performing action
- Communicates/spreads using standard protocols



Other forms of malicious logic

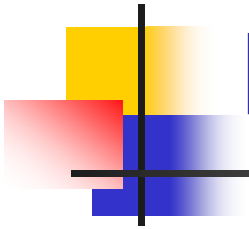
- We've discussed how they propagate
  - But what do they do?
- Rabbits/Bacteria
  - Exhaust system resources of some class
  - Denial of service; e.g., While (1) {mkdir x; chdir x}
- Logic Bomb
  - Triggers on external event
    - Date, action
  - Performs system-damaging action
    - Often related to event
- Others?

We can't detect it: Now what? Detection



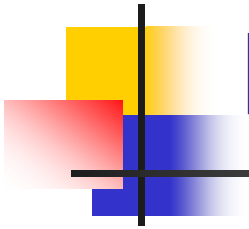
- Signature-based antivirus
  - Look for known patterns in malicious code
  - *Great business model!*
- Checksum (file integrity, e.g. Tripwire)
  - Maintain record of "good" version of file

- Validate action against specification
  - Including intermediate results/actions
  - *N*-version programming: independent programs
    - A fault-tolerance approach (diversity)



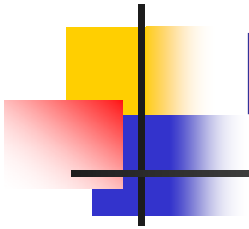
Detection

- Proof-carrying code
  - Code includes proof of correctness
  - At execution, verify proof against code
    - *If code modified, proof will fail*
- Statistical Methods
  - High/low number of files read/written
  - Unusual amount of data transferred
  - Abnormal usage of CPU time



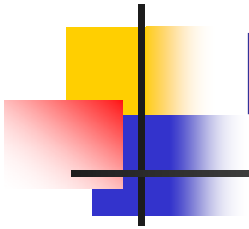
Defense

- Clear distinction between data and executable
  - Virus must write to program
    - Write only allowed to data
  - Must execute to spread/act
    - Data not allowed to execute
  - Auditable action required to change data to executable



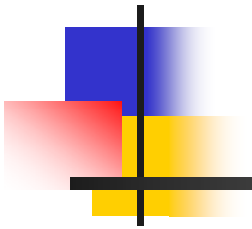
Defense

- Information Flow Control
  - Limits spread of virus
  - Problem: Tracking information flow
- Least Privilege
  - Programs run with minimal needed privilege



Defense

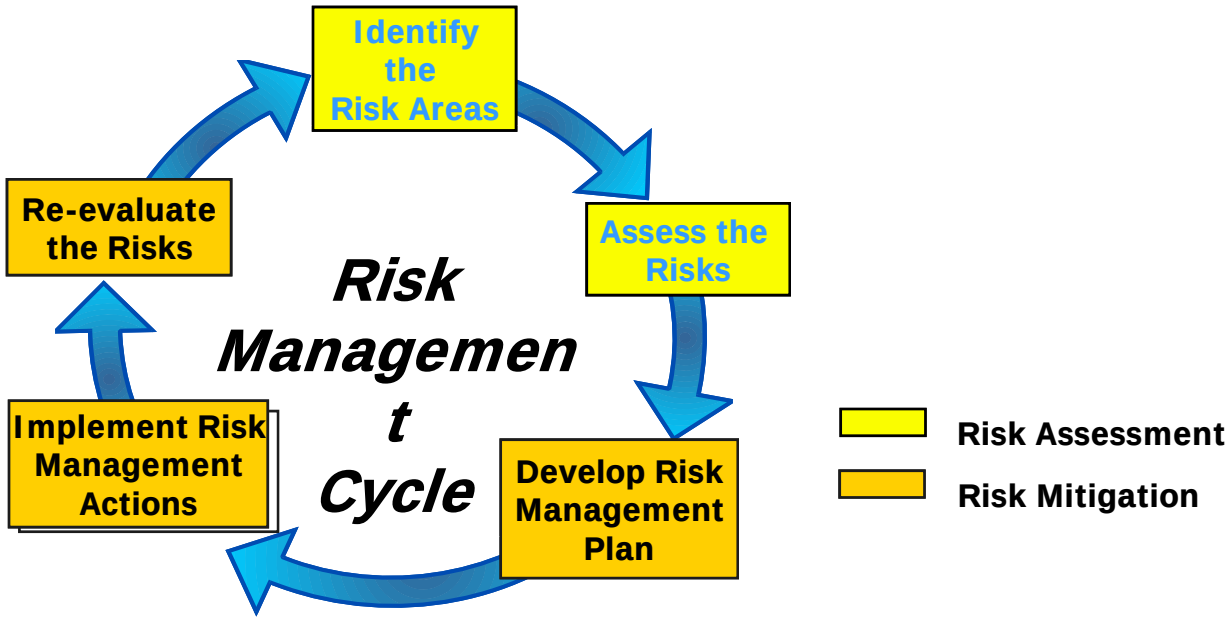
- Sandbox / Virtual Machine
  - Run in protected area
  - Libraries / system calls replaced with limited privilege set
- Use Multi-Level Security Mechanisms
  - Place programs at lowest level
  - Don’t allow users to operate at that level
  - *Prevents writes by malicious code*

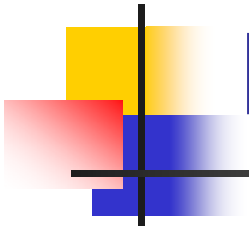


Risk Analysis

Risk Management

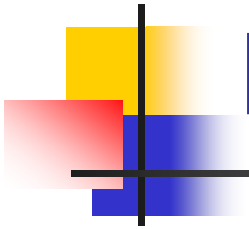
- The process concerned with identification, measurement, control and minimization of security risks in information systems to a level commensurate with the value of the assets protected (NIST)





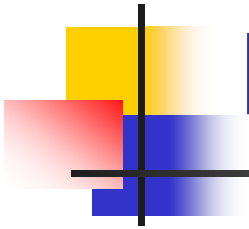
Risk

- 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



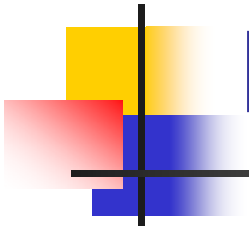
Risk Assessment/Analysis

- 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



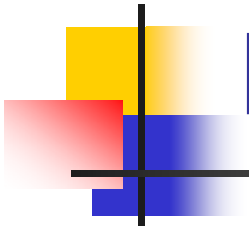
Risk Assessment steps

- 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

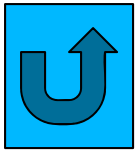
Example 2 (2)

- 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