

**NIS/CpE 691CE**

**Information System Security**

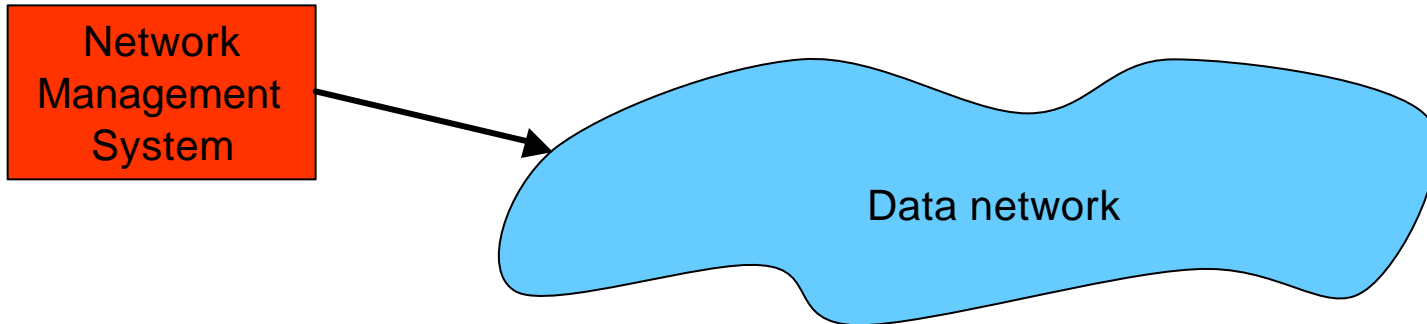
**Class 3 – 9/23/02**

# Some Important Topics in Information System Security

- Minimum privilege/minimum functionality
- Compartmentalization/Containment
  - Separation of Responsibility
  - Dual Controls
- Security Perimeters
- Trustworthiness/Design Correctness
- Single-points-of-failure/Choke-points
- Covert Channels
- Inference
- Implicit vs. Apparent Security

# Minimum privilege/Minimum functionality

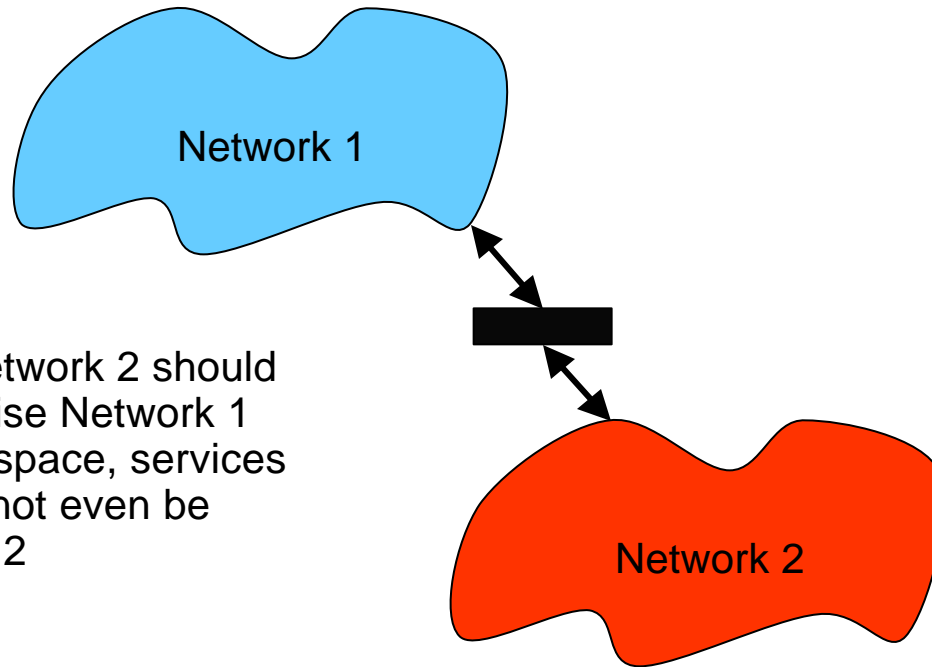
- Network Management System



- Applications running on NMS have ultimate control over operation of data network
  1. What capabilities do users really need to have to perform their job?  
Do users need to be able to monitor traffic on the network?  
Including (potentially) sensitive user traffic?
  2. What features does system really need to enable it to operate?  
Does NMS application code get compiled on NMS or is it downloaded?

# Compartmentalization/Containment

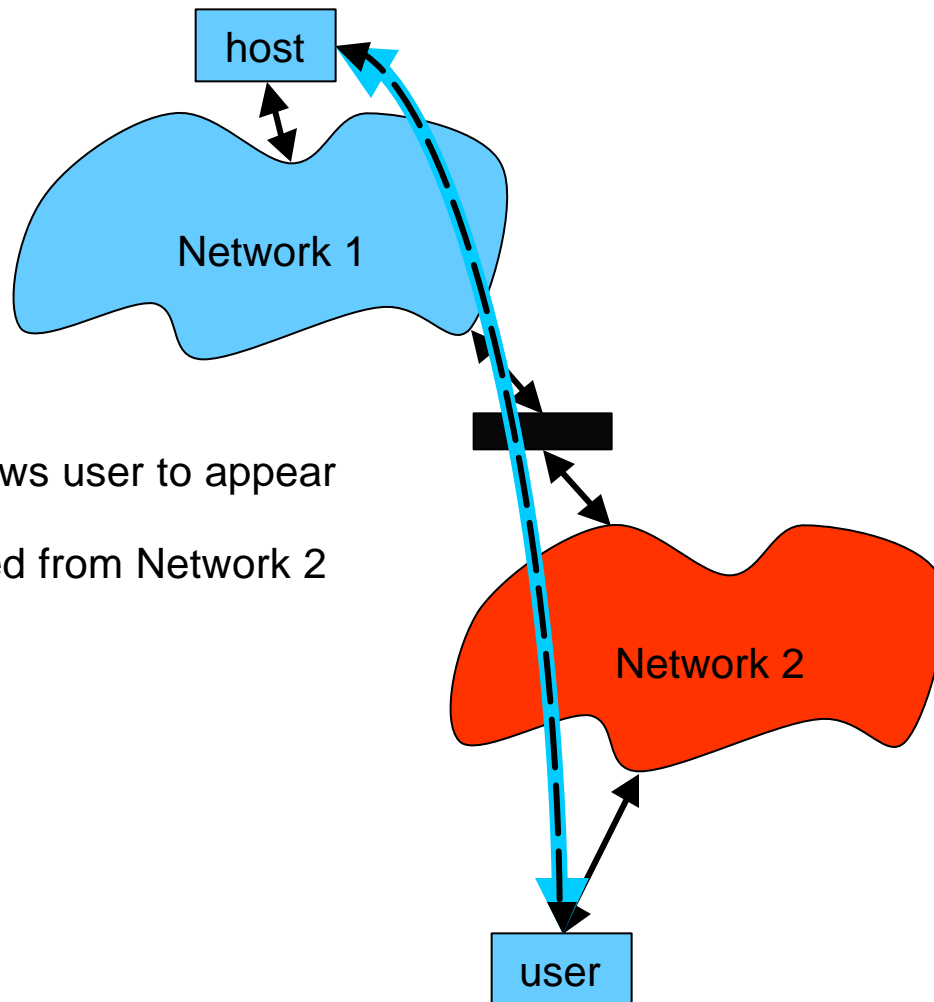
- Firewall



- Potential compromise of Network 2 should not be allowed to compromise Network 1
- Partitioning of traffic, namespace, services
- Entities on Network 1 may not even be visible to users on Network 2

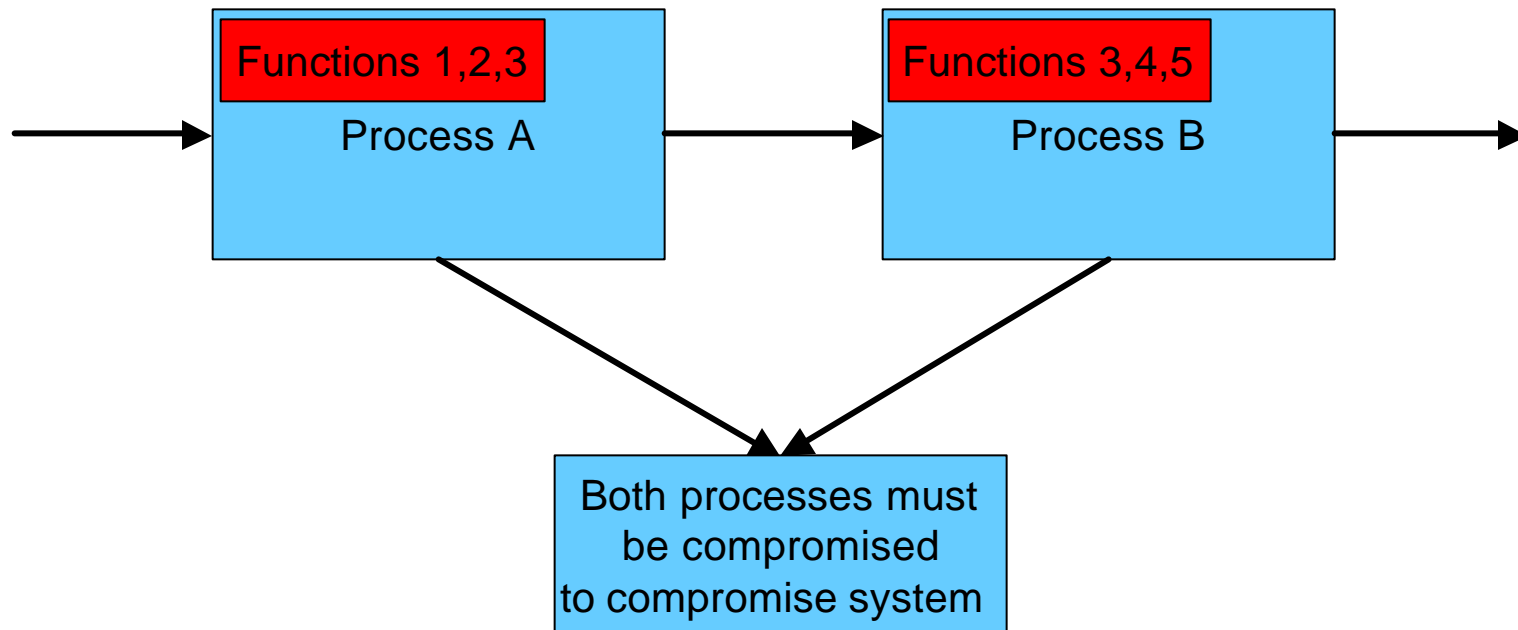
# Compartmentalization/Containment

- Virtual Private Network

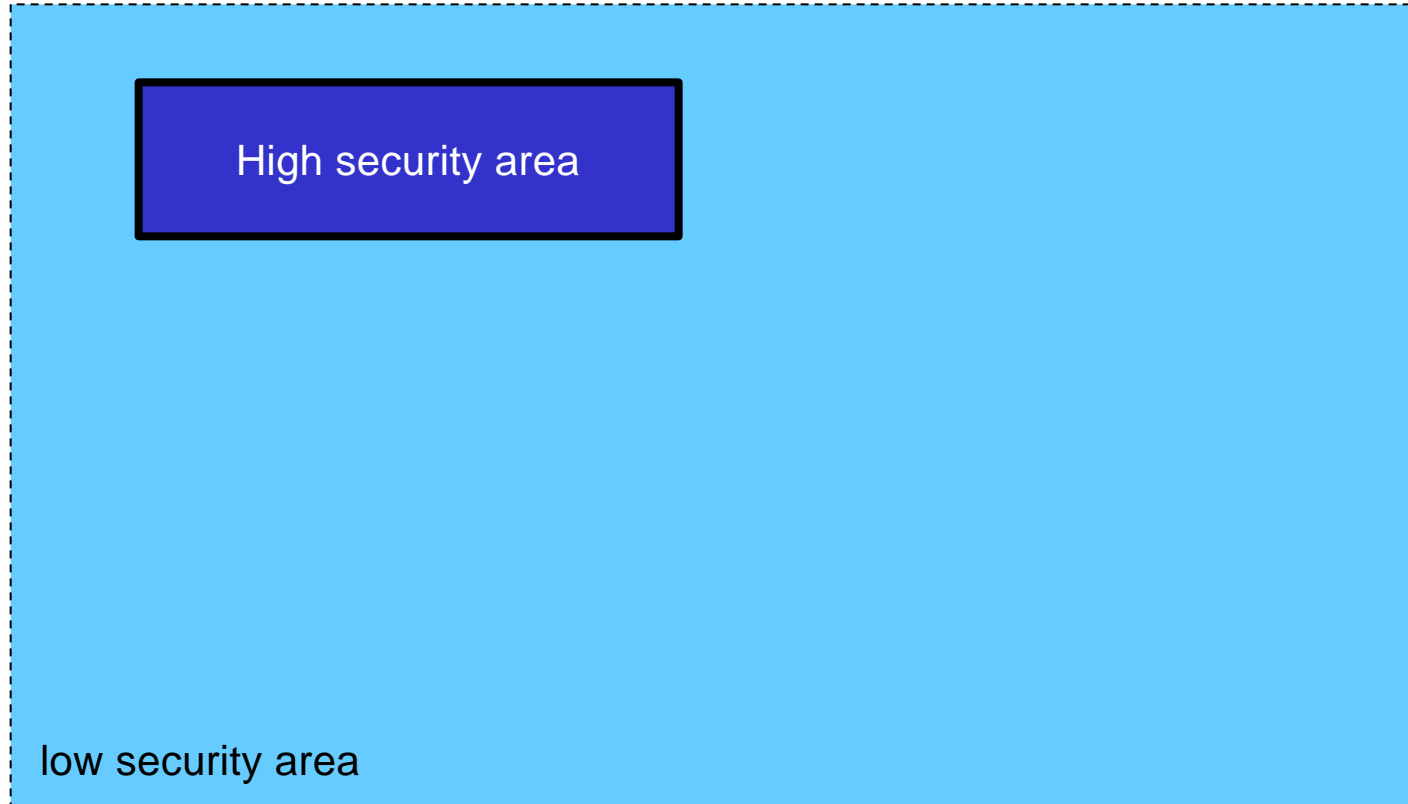


- Encrypted VPN 'tunnel' allows user to appear to be virtually on Network 1
- Tunnel is compartmentalized from Network 2

# Separation of Responsibility

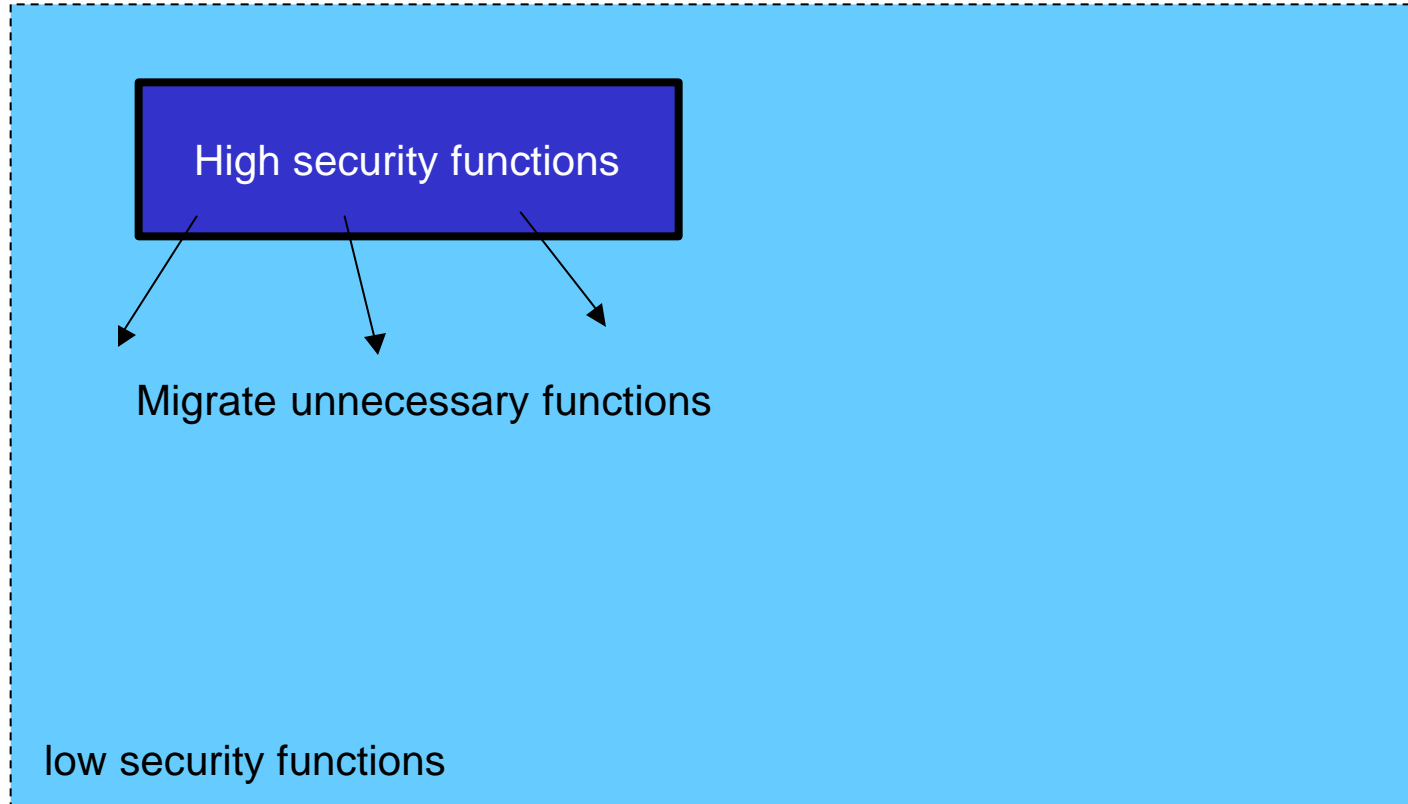


# Security Perimeters



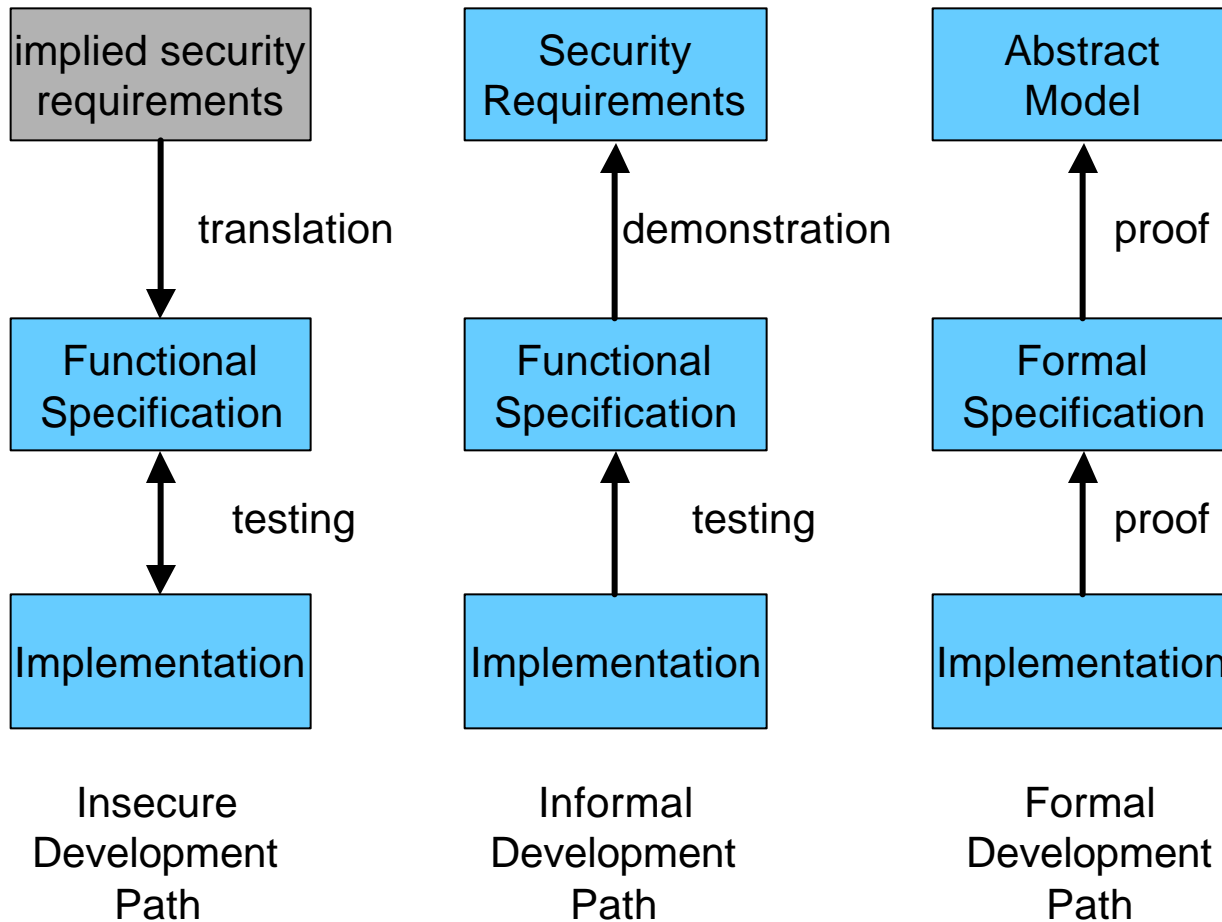
- Cost of protection scales with size of secure area
- Defining a small security perimeter containing critical assets allows focus on security priorities

# Security Perimeters

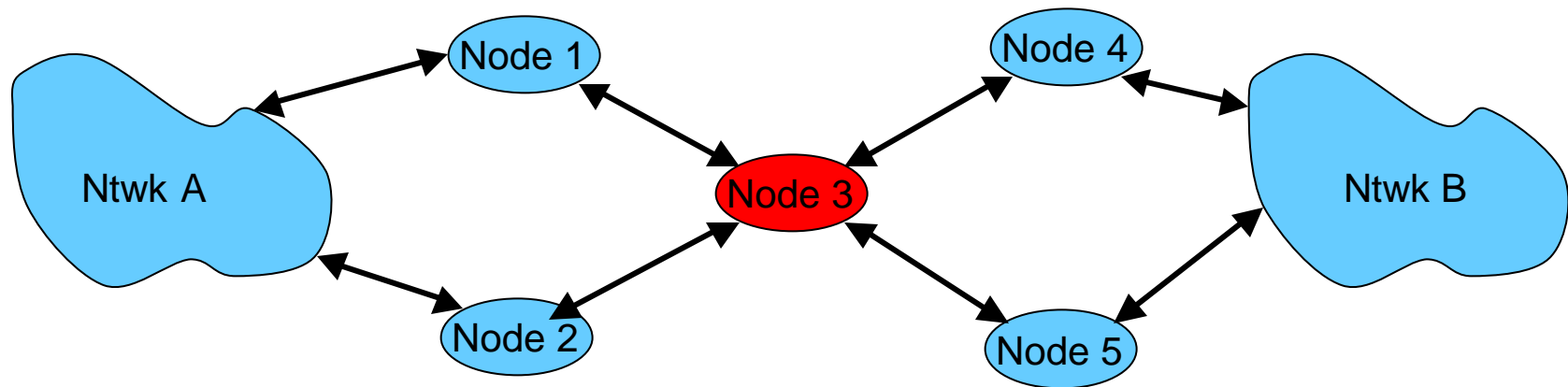


- Migrating unnecessary functions out of secure perimeter reduces need for inspection/assurance
- Reduces risk of compromise

# Trustworthiness/Design Correctness

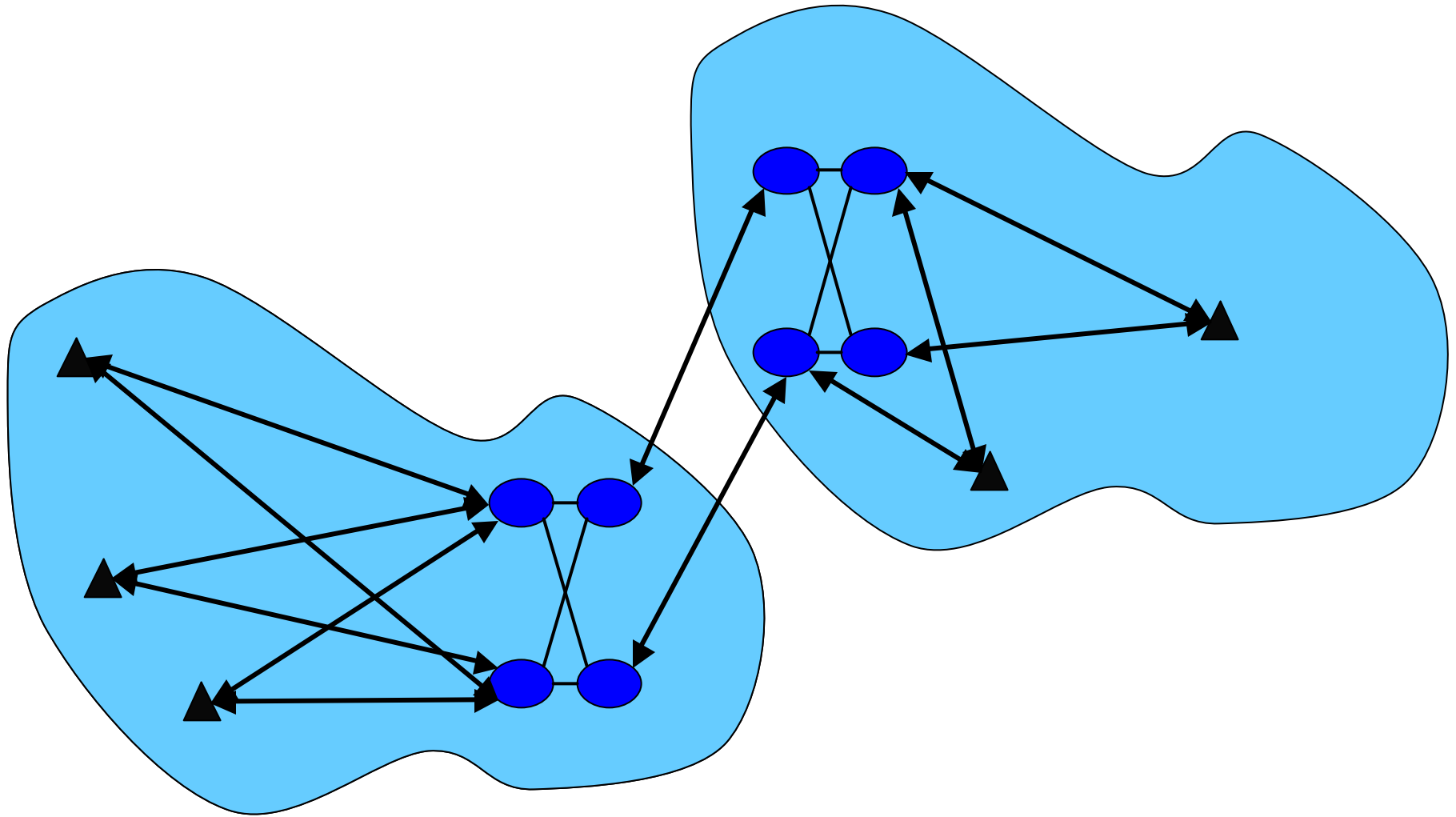


# Single-points-of-failure/Choke-points

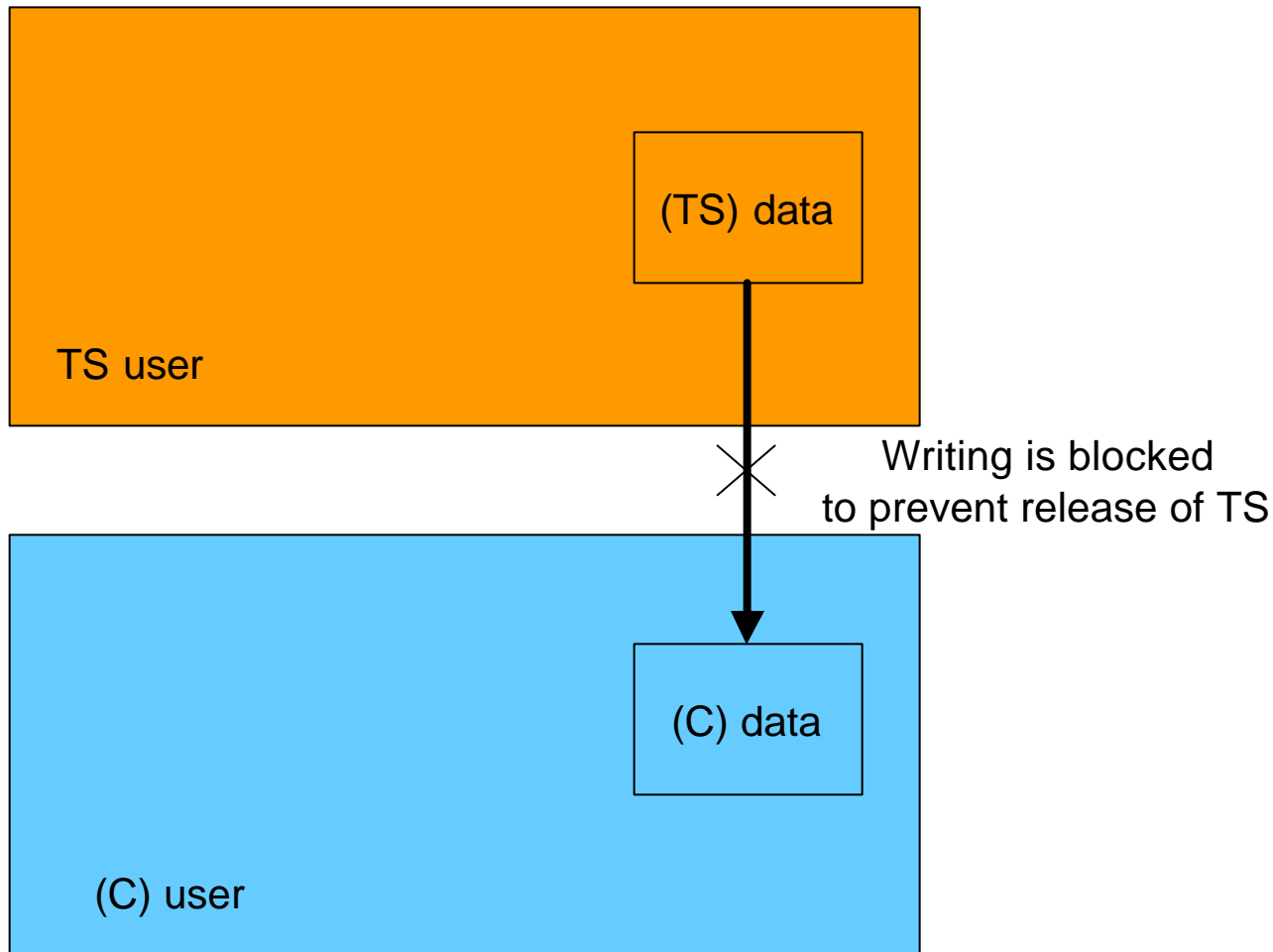


Node 3 is a single-point-of-failure (or attack) and a choke-point

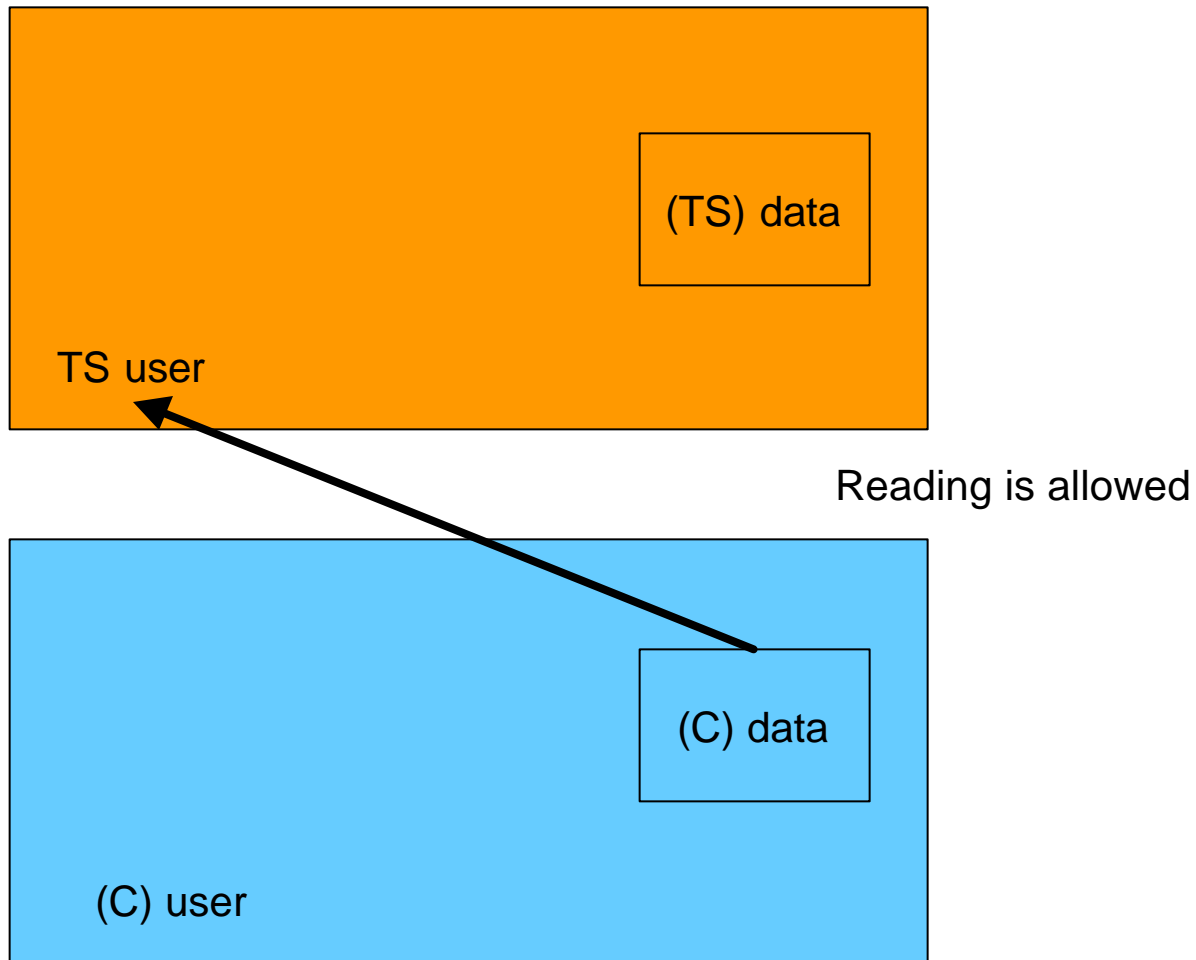
# Survivable Signalling Network (SS7)



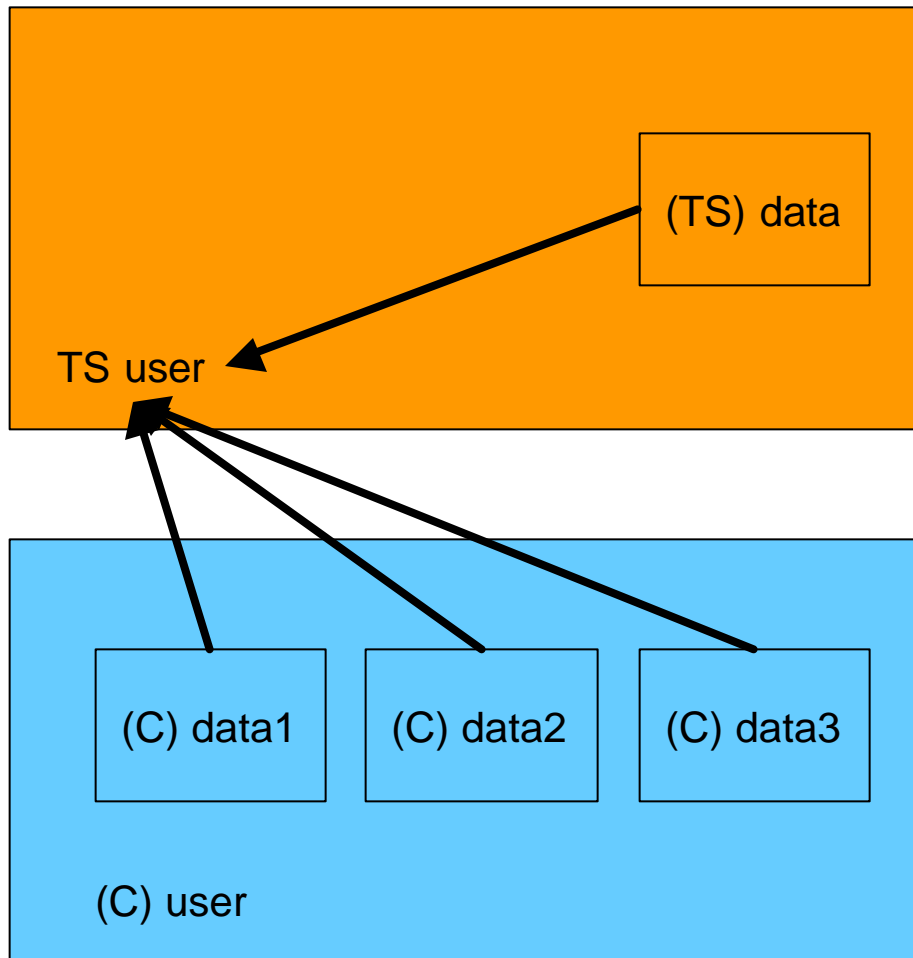
# Covert Channels - Storage Channel



# Covert Channels - Storage Channel



# Covert Channels - Storage Channel



Consider a DB with record locking:

TS: Open1, Open2

C: Open1(blocked), Open2(blocked),  
Open3(succeed)  
Until(Open1) {}  
Close3, Close1

TS: While(!Open3){  
Close1, Close2, Close3

// TS just sent a "0"

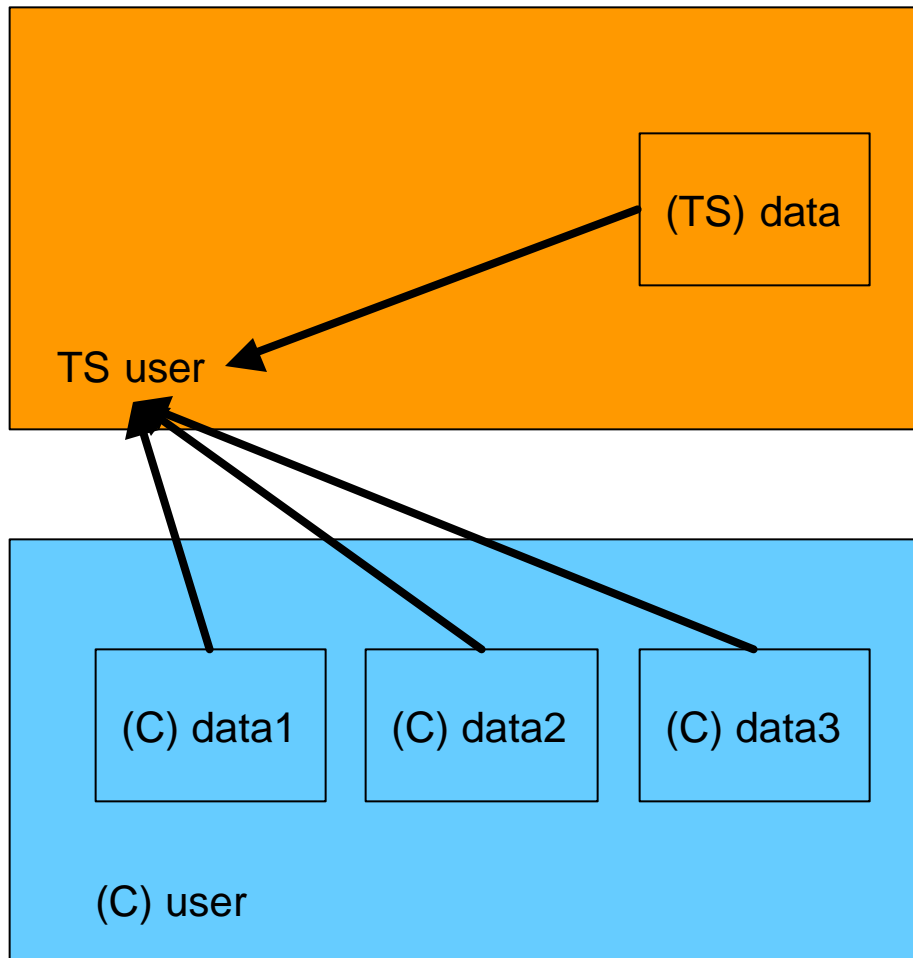
TS: Open2, Open3

C: Open1(succeed)  
Until(Open2){}  
Close1, Close2

TS: While(!Open1){  
Close1, Close2, Close3

//TS just sent a "1"

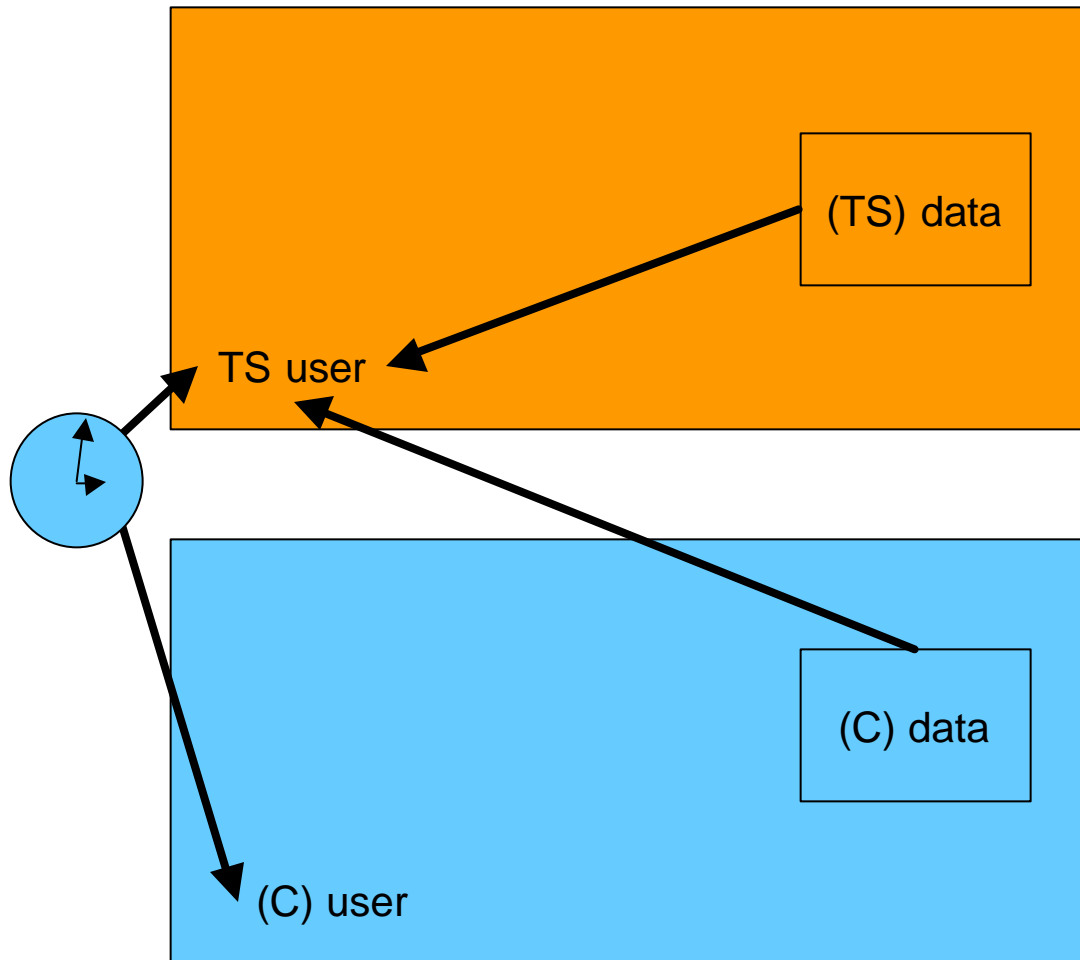
# Covert Channels - Storage Channel



This is an obvious covert channel, with wide bandwidth (on the order of the open/close speed of a data record)

Arbitrary covert channels can be exploited with  $P(\text{detection})$  related to utilized bandwidth.

# Covert Channels - Timing Channel



Synchronized access to lower level data is used by TS user to convey TS data to lower level user

Note: "TS user" might be Trojan Horse operating on behalf of TS user

# Inference

- Example 1:
  - Stevens has used Social Security Numbers as Student IDs for many years. Grades were (are?) posted by SSN. Name/SSN are never displayed together publicly
  - AT&T Bell Labs (That name carbon-dates how far the age of the issue) switched from Payroll Account Numbers (PANs) to SSNs as employee identifiers
  - The POST employee directory was searchable by PAN or SSN, but did not display them
  - Individual privacy can be compromised by SSN fairly easily
    - How can two relatively secure systems be played against each other?

# Inference

- Example 1:

- Stevens has used Social Security Numbers as Student IDs for many years. Grades were (are?) posted by SSN. Name/SSN are never displayed together publicly
- AT&T Bell Labs (That name carbon-dates how far the age of the issue) switched from Payroll Account Numbers (PANs) to SSNs as employee identifiers
- The POST employee directory was searchable by PAN or SSN, but did not display them
- Individual privacy can be compromised by SSN fairly easily

- How can two relatively secure systems be played against each other?

- A large percentage of part-time Stevens EE/CpE & CS graduate students have historically come from AT&T/Bell Labs
- Obtain the SSNs of Stevens EE/CpE/CS graduate students from posted grades
- Search the POST data base by SSN to identify individuals.
  - Individual privacy is compromised by the joint weakness of two systems that are relatively secure separately

# Inference

- Example 2
  - ref: Dorothy Denning, “The tracker - inference issues in database security”
  - Database contains User names, department, ages, salary, etc.
  - Individual records are protected against search by low level users: only trusted users may read separate records
  - Aggregate database statistics may be viewed by lower level users, e.g.,
    - “Show average salary of male employees”
    - “Show number of users earning more than \$100k”
  - Database security system prevents lower level user from retrieving data sets or statistics based on small number of records

# Inference

- Example 2

- ref: Dorothy Denning, “The tracker - inference issues in database security”

- Database contains User names, department, ages, salary, etc.

- Individual records are protected against search by low level users: only trusted users may read separate records

- Aggregate database statistics may be viewed by lower level users, e.g.,

- “Show average salary of male employees”

- “Show number of users earning more than \$100k”

- Database security system prevents lower level user from retrieving data sets or statistics based on small number of records

- The DB Inference problem:

- Attacker creates a series of queries that have a small sample size in their intersection

- Unless DB security system can assess sample sizes for all possible combinations of queries user has ever made, it is subject to an inference attack.

- Even if it does this, innocent queries can be denied because they MIGHT create inference vulnerability

# Implicit vs. Apparent Security

- User chosen passwords are notoriously insecure, often subject to dictionary attacks. Machine generated passwords are suggested as an alternative. Which is more secure?

- Password scheme1:

- character(k) = {a-z, 0-9, !@#\$%^&\*() } (46 symbols)

- PW = kkkkkk

- sample passwords: a5&98!, tfhe5&, 3thp1,

- Password scheme2

- vowel(v) = {aeiou}

- consonant(c) = {bcdfghjklmnpqrstvwxyz}

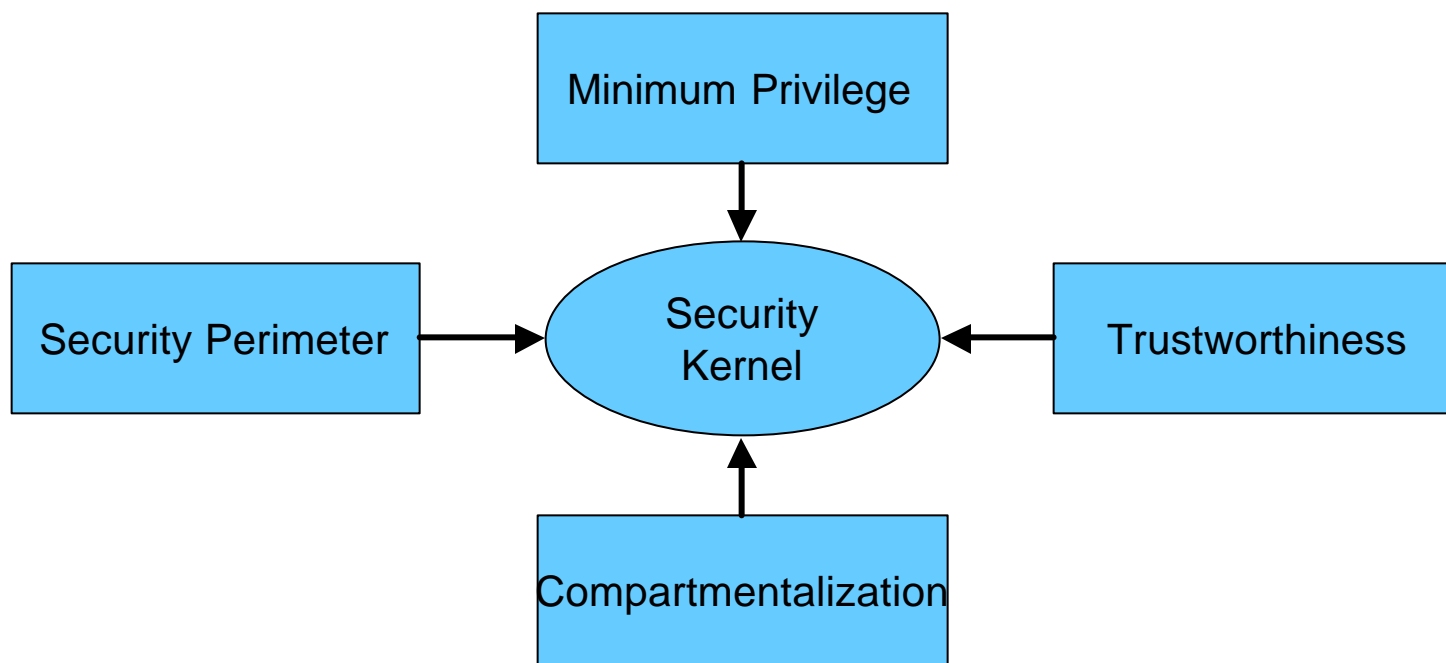
- PW = cvcvcvcvcv

- sample passwords: ponihavoka, risehipeta, tojifatese

# Implicit vs. Apparent Security

- User chosen passwords are notoriously insecure, often subject to dictionary attacks. Machine generated passwords are suggested as an alternative. Which is more secure?
  - Password scheme1:
    - character(k) = {a-z, 0-9, !@#\$%^&\*() } (46 symbols)
    - PW = kkkkkk
    - sample passwords: a5&98!, tfhe5&, 3thp1,
    - Total password space: 9,474,296,896**
  - Password scheme2
    - vowel(v) = {aeiou}
    - consonant(c) = {bcdfghjklmnpqrstvwxyz}
    - PW = cvcvcvcvcv
    - sample passwords: ponihavoka, risehipeta, tojifatese
    - Total password space: 10,000,000,000**
- **Apparent complexity of first scheme suggests higher security, but ease of memorization of second makes passwords more secure**

# Combining Concepts



# Homework 3

- We have discussed ~8 distinct security topics tonight. Consider any 5 and describe how they might apply to either
  - an Internet commerce site (e.g., Amazon, Ebay) or
  - airline physical security.
- Extra Credit: Apply all. Consider both issues