

Spezielle Kapitel aus Betriebssysteme: Secure Code - LVA 353.013 Part 2

secure: [si-'kyur]

1: free from danger

2: free from risk of loss

3: affording safety

Microsoft

The Attacker's Advantage ...

- #1 The defender must defend all points; the attacker can choose the weakest point.
- #2 The defender can defend only against known attacks; the attacker can probe for unknown vulnerabilities.
- #3 The defender must be constantly vigilant; the attacker can strike at will.
- #4 The defender must play by the rules; the attacker can play dirty.

Microsoft

Ridiculous Excuses We've Heard

- Excuse: No one will do that!
- Excuse: Why would anyone do that?
- Excuse: We've never been attacked
- Excuse: We're secure - we use crypto
- Excuse: We're secure - we use ACL
- Excuse: We're secure - we use a firewall
- Excuse: We've reviewed the code

Microsoft

Ridiculous Excuses We've Heard

- Excuse: We know it's the default...
- Excuse: If we don't run as admin
- Excuse: But we'll slip the schedule
- Excuse: It's not exploitable!
- Excuse: But that's the way we've always done it
- Excuse: If only we had better tool ...

Microsoft

<http://www.sans.org/top20/>



The Twenty Most Critical Internet Security Vulnerabilities (Updated) ~ The Experts Consensus

Version 6.0 November 22, 2005 Copyright (C) 2005, SANS Institute

Questions / comments may be directed to top20@sans.org.

To link to the Top 20 List, use the SANS Top 20 List logo at www.sans.org/top20/top20logo03.gif



Dimension Data CxO Security Assessment.
Small investment. Huge Dividends.
Benchmark against your peers.



-----Jump To Index of Top 20 Threats -----

Introduction

The SANS Top 20 Internet Security Vulnerabilities

Four years ago, the SANS Institute and the National Infrastructure Protection Center (NIPC) at the FBI released a document summarizing the Ten Most Critical Internet Security Vulnerabilities. Thousands of organizations used that list, and the expanded Top-20 lists that followed one, two, and three years later, to prioritize their efforts so they could close the most dangerous holes first. The vulnerable services that led to worms like Blaster, Slammer, and Code Red have been on these lists.

This SANS Top-20 2005 is a marked deviation from the previous Top-20 lists. In addition to Windows and UNIX categories, we have also included Cross-Platform Applications and Networking Products. The change reflects the dynamic nature of the evolving threat landscape. Unlike the previous Top-20 lists, this list is not "cumulative" in nature. We have only listed critical vulnerabilities from the past year and a half or so. If you have not patched your systems for a length of time, it is highly recommended that you first patch the vulnerabilities listed in the Top-20 2004 list.

We have made a best effort to make this list meaningful for most organizations. Hence, the Top-20 2005 is a consensus list of vulnerabilities that require immediate remediation. It is the result of a

[PDF](#) | [Printer Friendly Version](#) >>

Related Resources

[Tools and Services That Find & Fix the Top 20 Vulnerabilities\(v6\) on Your Systems & Networks Press Release \(2005-11-22\)](#)

Top 20 In The News

[Hackers pose new threat to desktop software - Financial Times](#)
[Hackers Targeting Security Programs - Washington Post](#)
[Viruses Get Smarter -- and Greedy - Business Week](#)

Top 20 Archive

[November, 2005 - Version 6 \(Current\)](#)
[October, 2004 - Version 5](#)
[October, 2003 - Version 4](#)
[October, 2002 - Version 3](#)
[May, 2001 - Version 2](#)
[June, 2000 - Version 1 \(Original Top 10\)](#)

Upcoming Conferences

[San Diego, CA - Dec. 4, 05](#)

Assignment – Solution

```
void f(){
    char buf[8];
    int *ret;

    // insert code here
    ret = (int*) (buf + 8 + 4 + 8 );
    *ret += 7;
}

int main(){
    int x = 0;
    f();
    x = 1;
    std::cout << "value= " << x << std::endl;
}
```

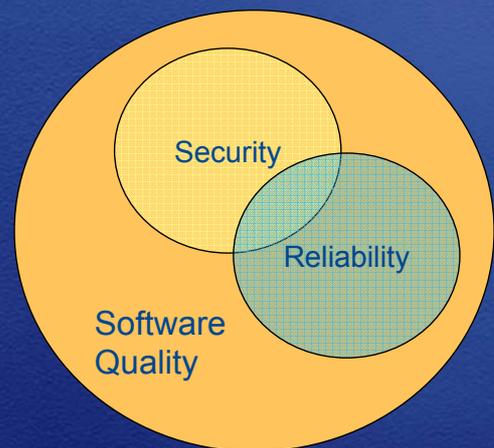
Microsoft

Summary

Microsoft

The role of Security

- Security \leftrightarrow Quality
 - Reliability & availability
- Security is **a** feature
 - Integral part
 - Not many features
- Poor security
 - Business downtime
 - Stole intellectual property
 - Reputation, financial, productivity – losses



Microsoft

Security Today

- Technology alone will not solve your problem
- Nobody believes anything bad can happen to them, until it does
- Security works only if the secure way also happens to be the easiest way
- In you do not keep up with security fixes, your network will not be yours for long
- There really is someone out there trying compromise your systems
- Your data and systems are of value to someone
- Security is not about risk elimination; it is about risk management

Microsoft

Note...

Security is only as good
as its weakest link

Microsoft

Security Features != Secure Features

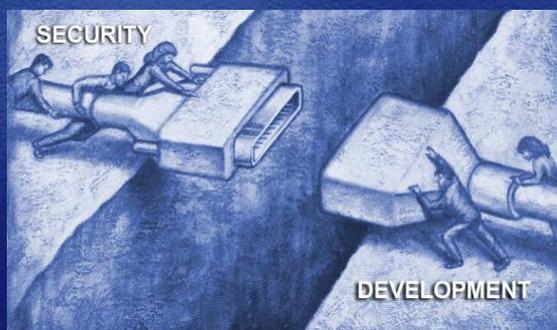
Microsoft

Why Security Vulnerabilities Occure

Security Professionals Don't Know the Applications

"As a Network Security Professional, I don't know how my company's applications are supposed to work so I deploy a protective solution...but don't know if it's protecting what it's supposed to."

The Application Security Gap



Application Developers and QA Professionals Don't Know Security

"As an Application Developer, I can build great features and functions while meeting deadlines, but I don't know how to build security into my applications."

Microsoft

Why Writing Secure Code is a Challenge

- Reasons developers give for not building secure applications
 - Security is boring
 - Security is often seen as a functionality disablement – gets in the way
 - Security is difficult to measure
 - Don't know how

Why Writing Secure Code is a Challenge

- Security vulnerabilities are expensive to fix
 - Coordination, finding bug, fixing the code, testers, and PR
 - Cost of lost productivity
 - Cost of lost trust of consumers

Why Writing Secure Code is a Challenge

- Attackers have the upper hand
 - Defender must defend all points; attacker only needs to find the weakest point
 - Defender can only defend against known attacks (attacks of today)

The Truth

- All software has security defects!
 - Yes, EVERYONE!
- Present development models cannot deliver secure software

A Security Framework: SD³ + C

Secure by Design

- Threat modeling
- Code inspection
- Process Improvement

Secure by Default

- Unused features off by default
- Reduce attack surface area
- Least Privilege

Secure by Deployment

- Prescriptive Guidance
- Security Tools
- Training and Education

Communications

- Community Engagement
- Transparency
- Clear policy

Microsoft

Defense in Depth (MS03-007)

Windows Server 2003 Unaffected

The underlying DLL
(NTDLL.DLL) not vulnerable

Code made more conservative during 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

Maximum URL length in IIS 6.0 is 16kb by default
(>64kb needed)

Even if the buffer was
large enough

Process halts rather than executes malicious code,
due to buffer-overflow detection code (-GS)

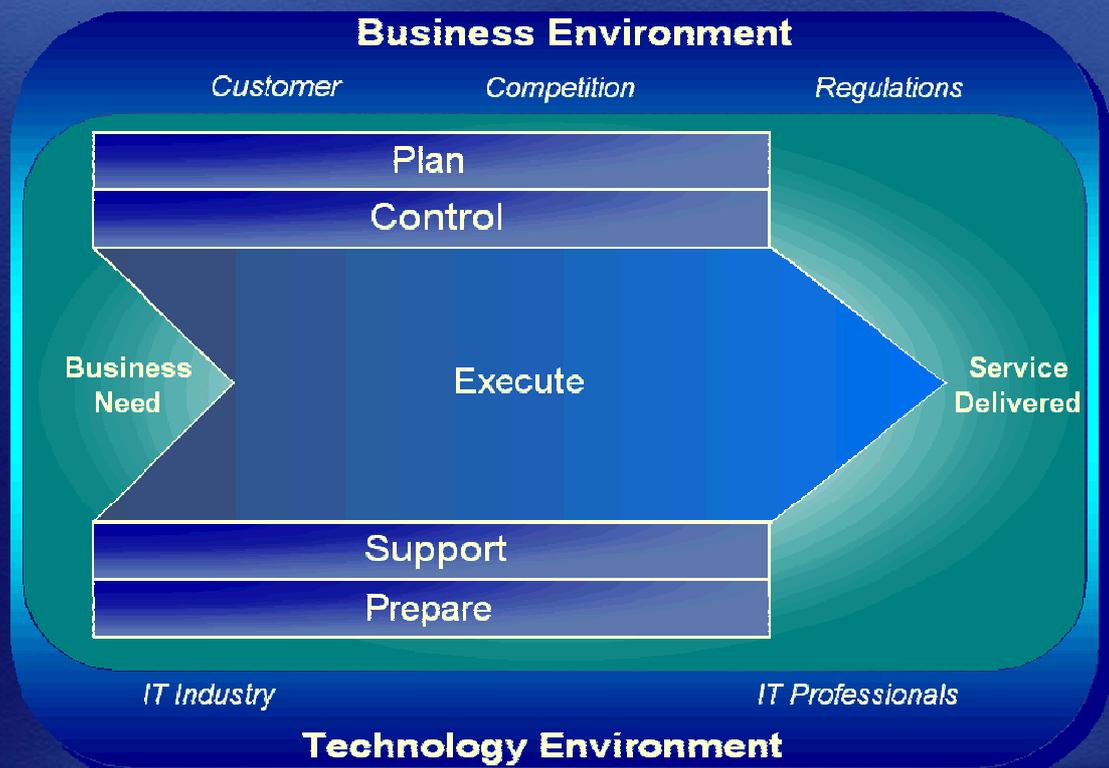
Even if there was an
exploitable buffer overrun

Would have occurred in `w3wp.exe` which is now
running as 'network service'

Introduction to SDL

Microsoft
Microsoft

Scope of Enterprise IT



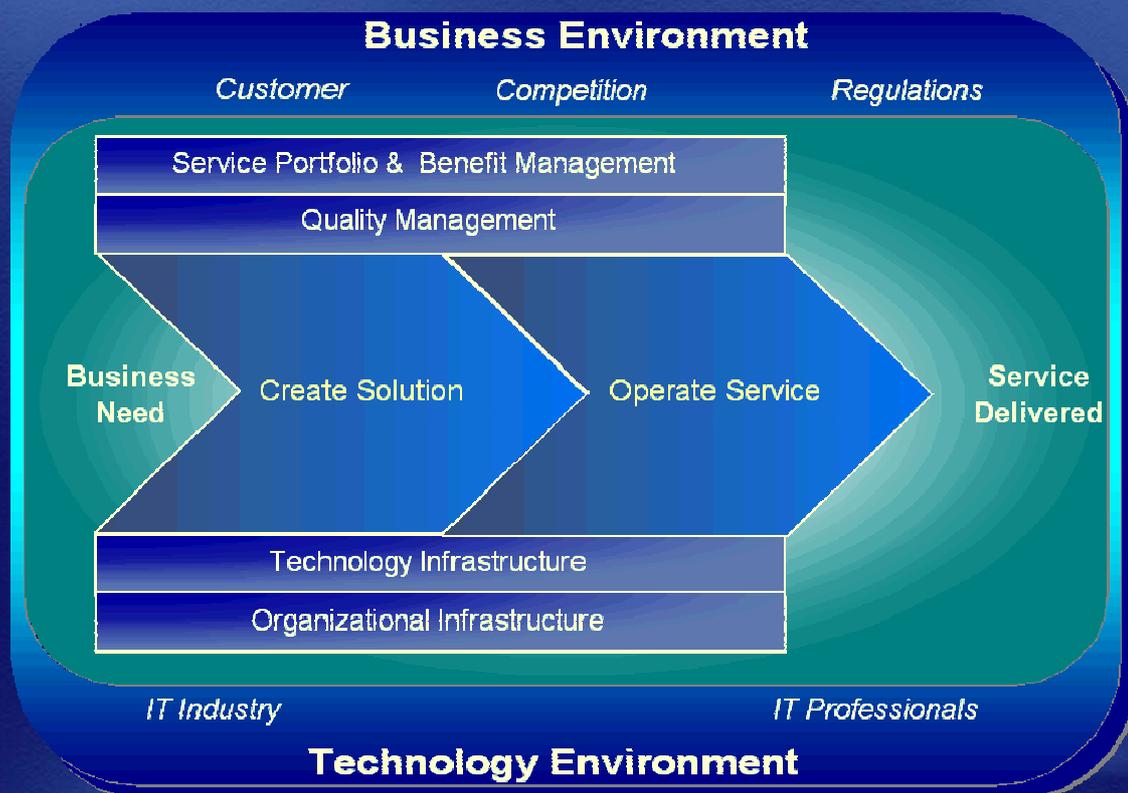
Microsoft

Why Application Security is Critical to Business

- Customer Expectations
- Internal Enterprise Applications
- Business Cost of Vulnerabilities

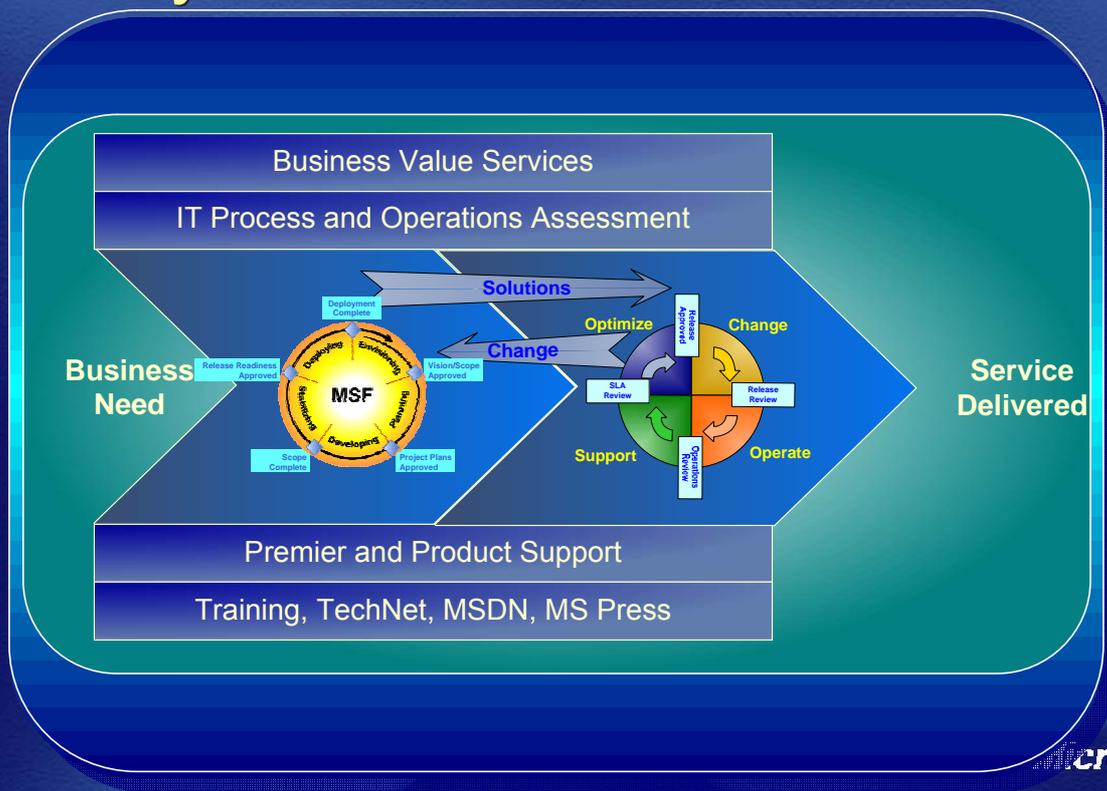
Microsoft

Scope of Enterprise IT



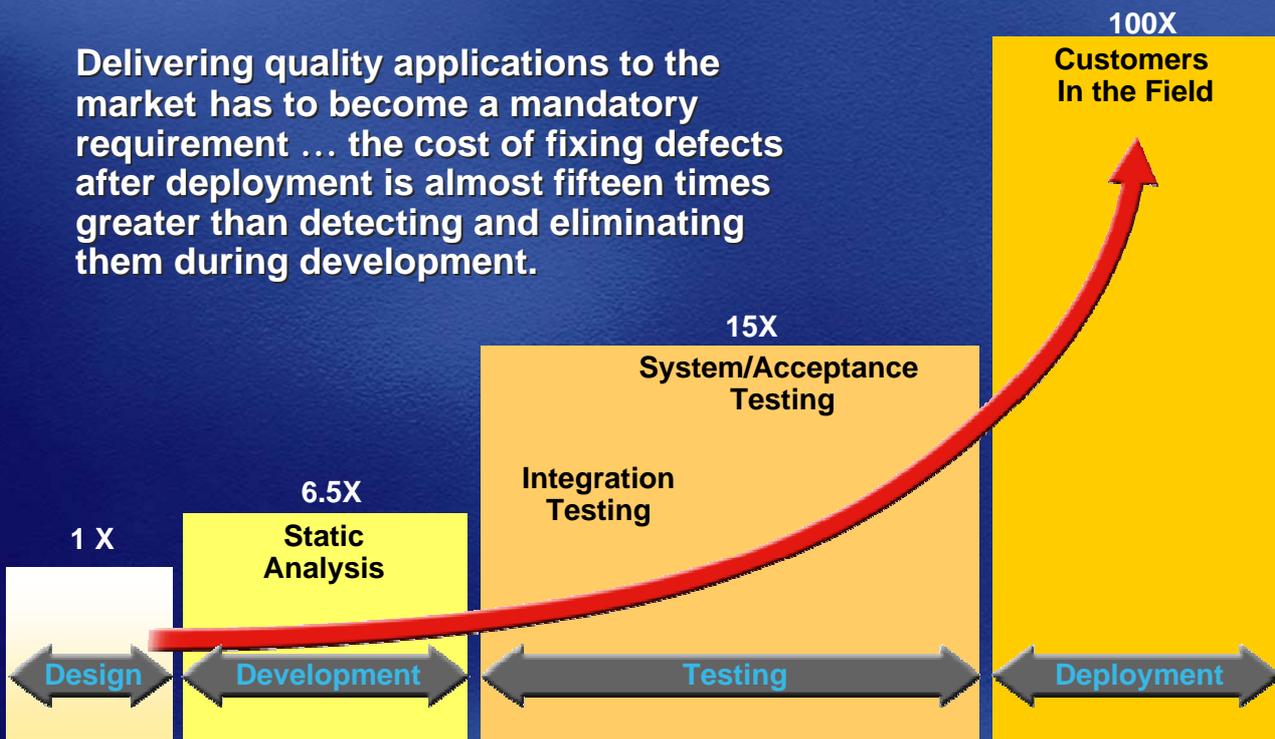
Microsoft

Frameworks: Consistent Lifecycle Processes



Why Software Development Must Change

Delivering quality applications to the market has to become a mandatory requirement ... the cost of fixing defects after deployment is almost fifteen times greater than detecting and eliminating them during development.



Source IDC and IBM Systems Sciences Institute

Elements that Drive Change



People: Providing guidance on secure application development



Process: Security cannot be an afterthought

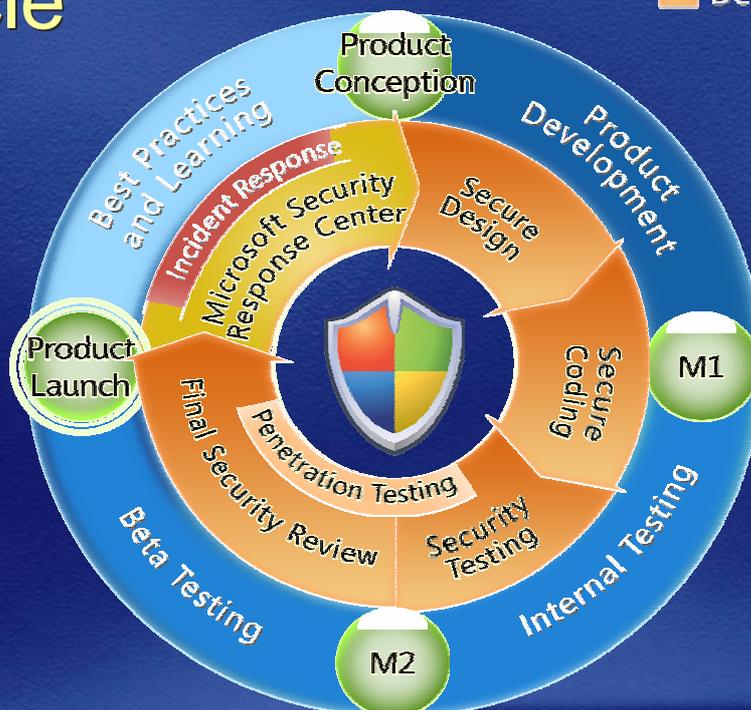


Tools: Providing the most innovative tools

Microsoft

The Security Development Lifecycle

- Microsoft Product Development Lifecycle
- Microsoft Security Development Lifecycle



Microsoft

The Security Development Lifecycle

A process by which Microsoft develops software, that defines security requirements and milestones

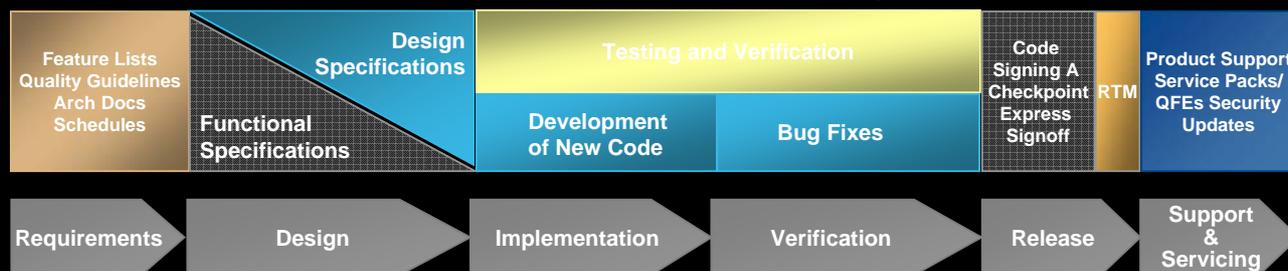
- Mandatory for almost all Microsoft products
- Evolving with new threats and technology
- Compatible with real-world development
- Effective at reducing vulnerabilities

Microsoft

Security Deployment Lifecycle Tasks and Processes



Traditional Microsoft Software Product Development Lifecycle Tasks and Processes



Microsoft

Education as a Driver



MSDN and TechNet
Sharing whitepapers and "how tos"



Patterns & Practices
Dedicated team focused on security guidance



Education
Train every Developer and IT Professional on security

Microsoft

Customer Experiences with SDL

Microsoft
Microsoft

SDL: The Customer View

- Security within the development lifecycle is a growing concern:
 - Increasing concern within customers of potential attacks at the application
 - CSO Councils over the last year Information/help with SDL was #1 request
 - How do I deal with the Asymmetrical problem?
 - *“How can you help me implement SDL in my organization in a pragmatic manner”*
 - *“What tools do you have to help me do more secure application development”*

Microsoft

Customer Experiences

- We are seeing increased activity around SDL:
 - Security certification for partners
 - Working with academia
 - Customer training for Archs/Snr Dev on SDL
 - Working with customers to review and implement SDL features into their own lifecycle
- There are a variety of experiences implementing SDL

Microsoft

Organizational Change

Stages of Adoption

Training / Awareness Dedicated Security Resources Build SDL into Development Process Release Controls ~~Planned~~ ~~Response~~



Increasing costs and organizational change

Increasing benefits

Microsoft

Key Decisions

- There are a number of factors impacting SDL implementation:
 - Executive buy-in
 - Separation of responsibility
 - Separate roles – Development, Test, Security
 - Mandate of SDL processes and tools within formal development methodology
 - Enforcement of exit criteria
- There are benefits at all levels of adoption

Microsoft

Drivers

- Regulatory Drivers
 - Sarbanes Oxley Act Section 404
 - The Basel II Framework - capital adequacy
- Business Drivers
 - Internet facing applications
 - Overall risk reduction
 - Increased attacks
 - Phishing
 - Software Virus

Microsoft

Benefits

- An enabler for effective measurement
 - Education – Individual and Team
 - Process implementation
 - In-process metrics provide early warning
 - Post-release metrics assess final payoff
- Increased awareness
 - “Before training... I thought a buffer overflow was what happened when I add too much Seltzer to my glass of water.” ☺
- Executives sleep easier at night

Microsoft

Lessons

- What we have learnt so far
 - Executive support is critical
 - Foster security champions
 - Dedicated security/risk teams help a lot
 - Engage external assistance for SDL training
 - Understand the risk/reward balance
 - Don't bite off more than you can chew
- What we are still learning
 - Expectations - It won't happen overnight...
 - Room for continuous improvement

Microsoft

Summary

- The Security Development Lifecycle
 - An integral part of Microsoft's development process
 - An effective process for removing software vulnerabilities
 - Evolving
 - Applicable to customer and ISV software

Microsoft

Threat Modeling



Some Important Definitions



87

- Threat Agent
 - Someone who could do harm to a system (also adversary)
- Threat
 - An adversary's goal
- Vulnerability
 - A flaw in the system that could help a threat agent realize a threat
- Asset
 - Something of value to valid users and adversaries alike
- Attack
 - When a motivated and sufficiently skilled threat agent takes advantage of a vulnerability

Why do Threat Modelling?

- To identify the threats your component faces and to challenge any assumptions that have been made
- To prioritise other security-related efforts
 - Code reviews
 - Fuzz testing
 - Penetration testing
- To look at the product with a different set of eyes
 - Highly technical and motivated criminal
 - Not your typical happy, paying customer
- To document everything for future generations

Microsoft

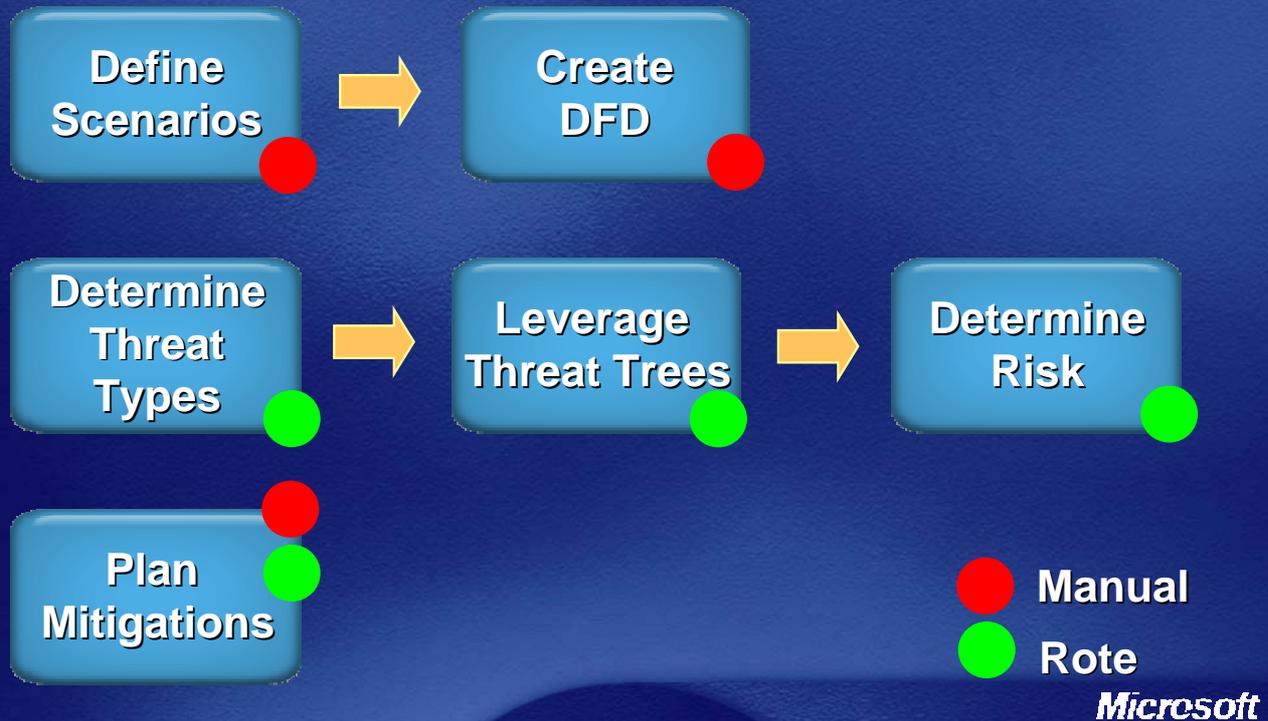
Threat Modeling Foundation

Use and Evolve Threat Models in

- Design
 - Mitigation and security designs
 - Drives attack surface design
- Development
 - Determines the most “insecure” portions of your application
 - General mitigations included in development guidelines
 - Drives security in code reviews and exit criteria
- Testing and Production
 - Drives security testing strategy (threats and mitigations)
 - All threats and mitigations must be tested
 - The job of a good security tester is to find other conditions in the threat tree
 - Attack points derived from threat model

Microsoft

The Updated Threat Modeling Process



Define Scenarios



- Identify what the application does
- Define the most common and realistic use scenarios for the application
 - Example from Windows Server 2003 and Internet Explorer
 - “Think about an admin browsing the Internet from a Domain Controller”
- Bounds the scope of what you need to model

Data Flow Diagrams

(DFDs)



90

- A DFD is a graphical representation of how data enters, leaves, and traverses your component
 - It is not a Class Diagram or Flow Chart!
 - Shows all data sources and destinations
 - Shows all relevant processes that data goes through
- Good DFDs are critical to the process
 - This point can't be emphasised enough!
 - Building DFDs == understanding the system
 - Analysing DFDs == understanding the threats

Microsoft

Create the DFD's



87



75

- Most “whiteboard architectures” are DFD-like



Microsoft

Create the DFD's Implementation Examples

External Entity



- Real People
- News feeds
- Data feeds
- Events
- Notifications
- Etc.

Process



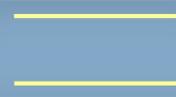
- Services
- Web Services
- Assemblies
- DLLs
- EXEs
- COM object
- Etc.

Dataflow



- Function call
- Network traffic
- Shared memory
- Etc.

Data Store



- Database
- File
- Registry
- Shared Memory
- Queue/Stack
- Etc.

Microsoft

Privilege Boundaries



- Specific DFD addition to TMs
- Boundary between DFD elements with different privilege levels
 - Machine boundary (data from the other machine could be anonymous)
 - Process boundary (e.g.; User process \leftrightarrow SYSTEM process)
 - Kernel \leftrightarrow User mode

Microsoft

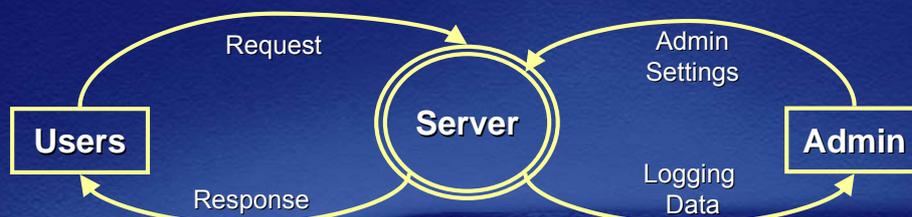
Types of DFDs

- Context Diagram
 - Very high-level; entire component / product / system
- Level 0 Diagram
 - High level; single feature / scenario
- Level 1 Diagram
 - Low level; detailed sub-components of features
- Level n Diagram
 - Even more detailed; unlikely to go beyond Level 2

Microsoft

Create the DFD's Context Diagram

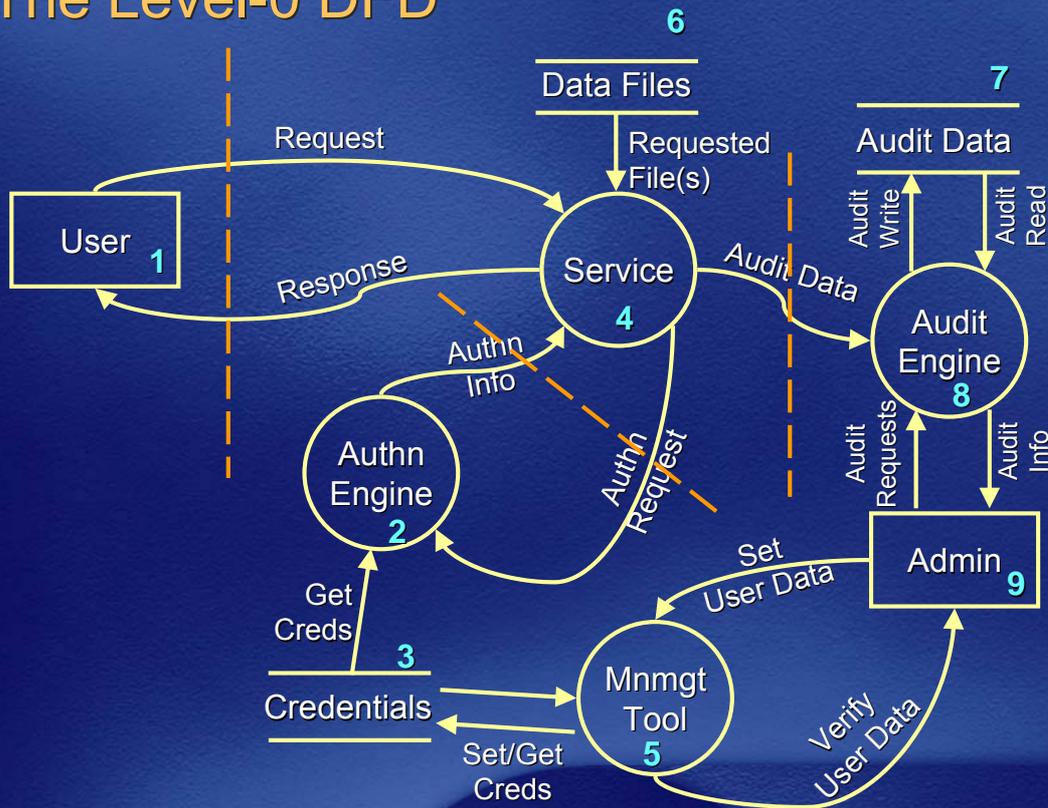
- A Context Diagram is a DFD that ***maps out the landscape*** of your component
 - Complexity Check:
 - How many external things does it interact with?
 - How many “moving parts” is it likely to have?
 - Scariness Check:
 - Does it accept traffic from the network?
 - Does it interact with lower-privilege components?



Microsoft

Create the DFD's

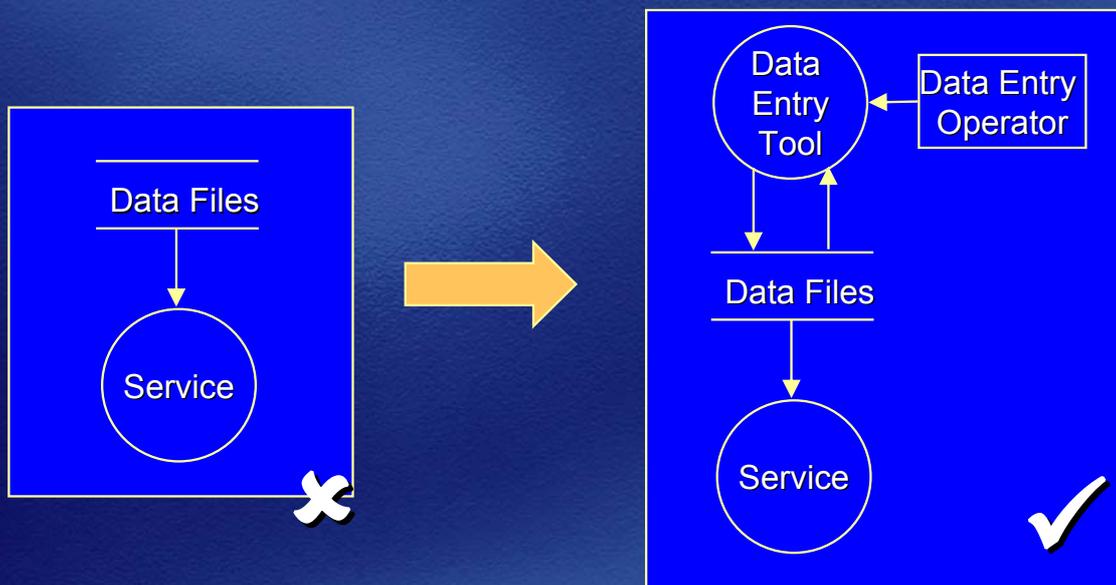
The Level-0 DFD



Microsoft

Common DFD "bugs"

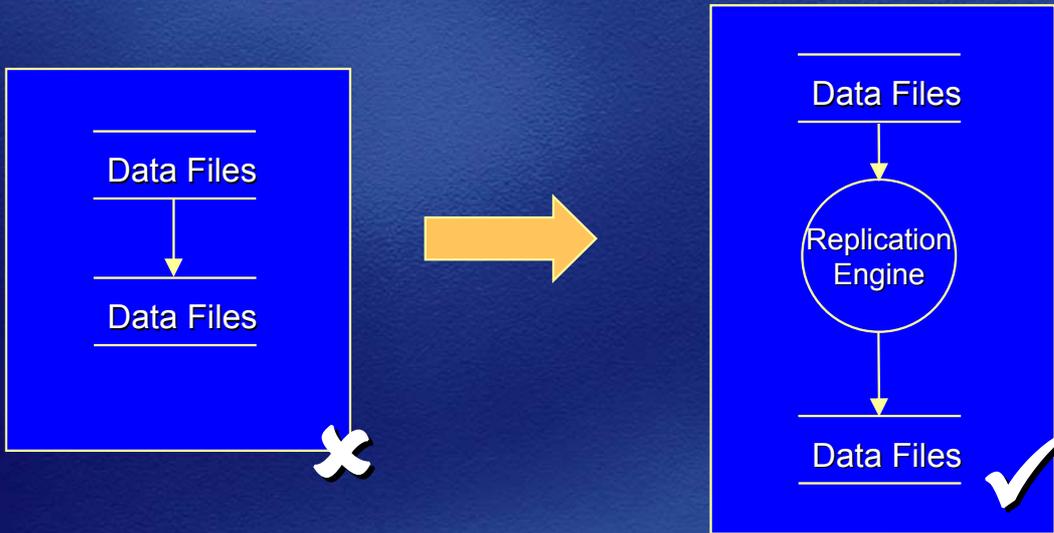
(1) How does the data get into the data store?



Microsoft

Common DFD “bugs”

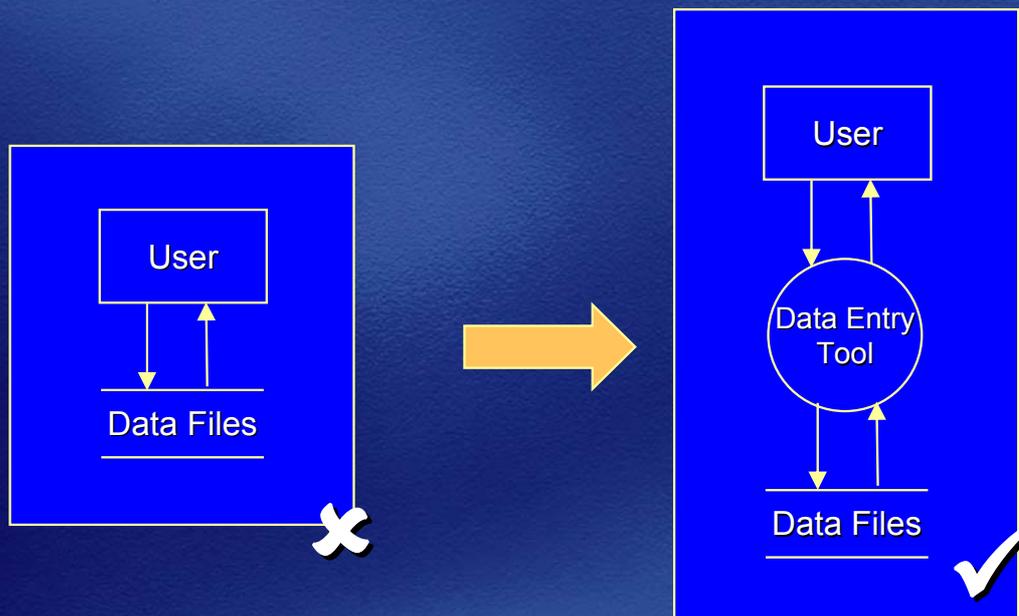
(2) How does data move from one data store to another?



Microsoft

Common DFD “bugs”

(3) How does data move from a user to a data store?



Microsoft

DFD Element Threat Types



- Each DFD element (Asset) is susceptible to certain kinds of threats
 - Spoofing
 - Tampering
 - Repudiation
 - Information Disclosure
 - Denial of Service
 - Elevation of Privilege

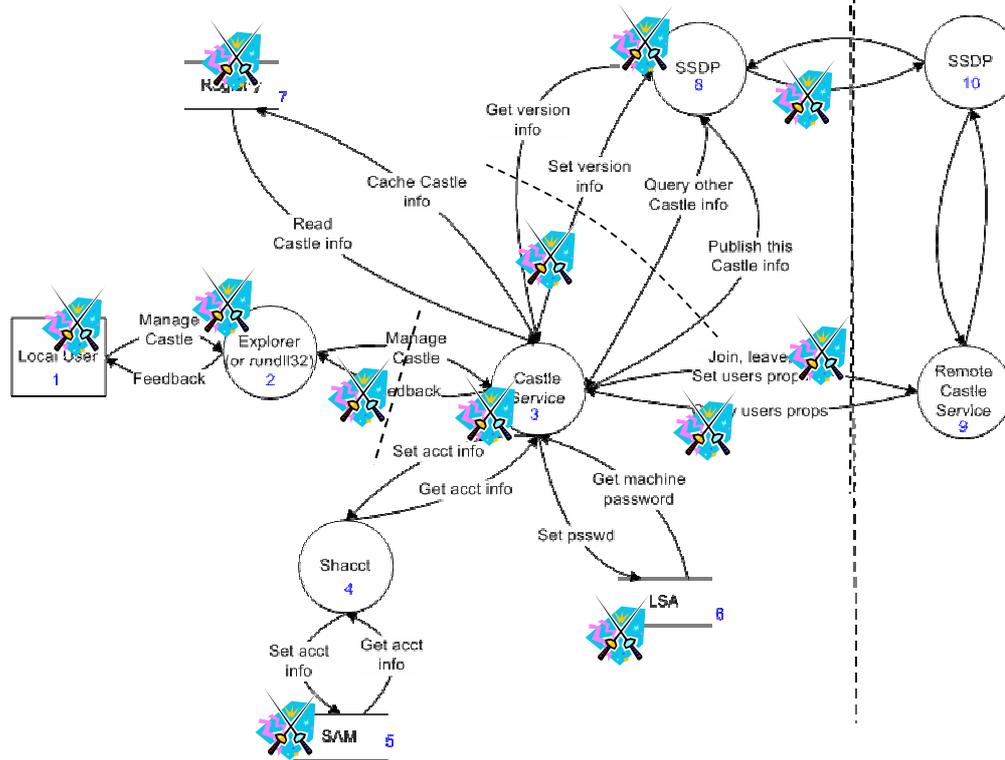
Microsoft

What is Repudiation?

- Something you probably won't need to worry too much about!
 - Usually involves policies (read: you'll need a lawyer)
- Mitigate with Non-repudiation techniques
- Non-repudiation services generate evidence which will help a disinterested party that a specific subject performed a specific action
- Evidence of Origination, Submission & Receipt

Microsoft

Every Asset is Subject to Attack



soft

Determining Threats

- Prime Threat
 - Based on DFD asset type
- Secondary Threat
 - Based on threat trees
 - Related issues

Microsoft

Prime Threats by Asset Type

Asset	S	T	R	I	D	E
 External Entity	✓		✓			
 Process	✓	✓	✓	✓	✓	✓
 Data Store		✓	✓	✓	✓	
 Dataflow		✓		✓	✓	

Microsoft

List all Assets from the DFD

Every asset is subject to prime threats

- External Entities

- 1

- Processes

- 2, 3, 4, 8

- Data Stores

- 5, 6 & 7

- Data Flows

- 1→2, 2→1, 2→3, 3→2, 2→7, 7→2, 3→4, 4→3, 4→5, 5→4, 3→6, 6→3, 3→8, 8→3, ...

Asset	S	T	R	I	D	E
 External Entity	✓		✓			
 Process	✓	✓	✓	✓	✓	✓
 Data Store		✓		✓	✓	
 Dataflow		✓		✓	✓	

Microsoft

A Complete List of Prime Threats

- Spoofing
 - E: 1
 - P: 2, 3, 5, 8
- Tampering
 - P: 2, 3, 5, 8
 - DS: 5, 6, 7
 - DF: 1→2 etc
- Repudiation
 - E: 1
 - P: 2, 3, 5, 8
- Information Disclosure
 - P: 2, 3, 5, 8
 - DS: 5, 6, 7
 - DF: 1→2 etc
- Denial of Service
 - P: 2, 3, 5, 8
 - DS: 5, 6, 7
 - DF: 1→2 etc
- Elevation of Privilege
 - P: 2, 3, 5, 8

Microsoft

Threat Trees

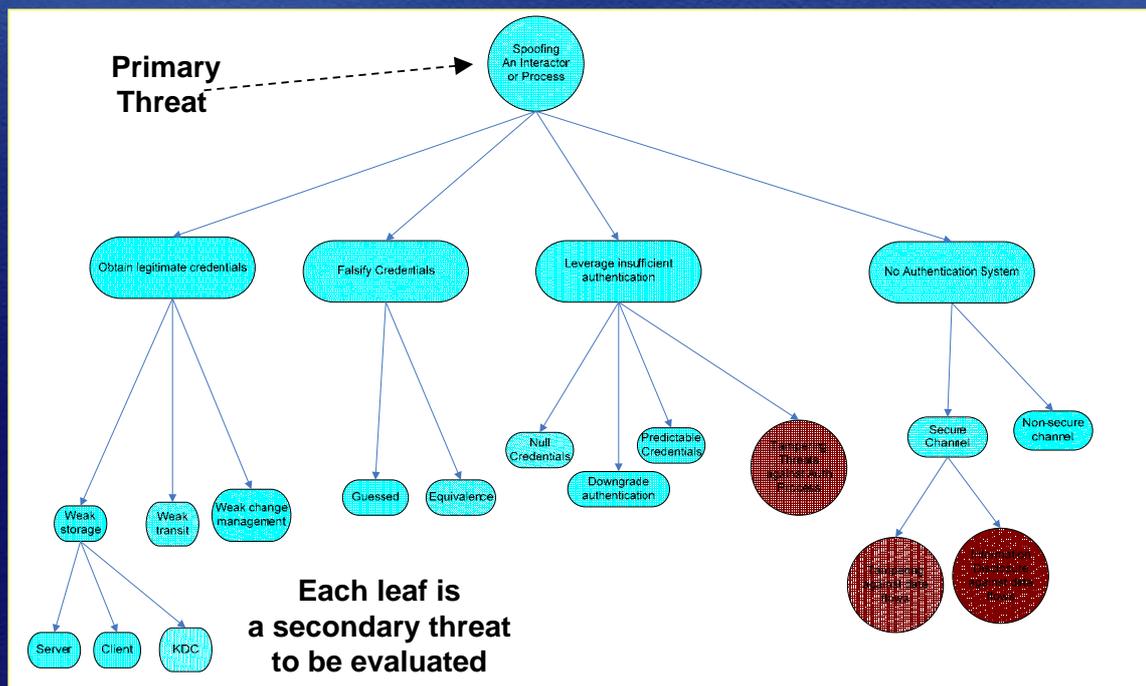
- A graphical representation of security-relevant pre-conditions in a system
- First outlined in Amoroso's "Fundamentals of Computer Security Technology"
- Based on hardware fault trees
- There are many "threat tree patterns"



Microsoft

Threat Tree Pattern Example

Spoofing



Microsoft

Threat Reduction

- Assets...
 - ...within the same trust boundary
 - ...using like technology
- Can be treated as one unit
 - Saves time!

Microsoft

A Special Note about Information Disclosure threats

All information disclosure threats are potential privacy issues.

Raising the Risk.

Is the data sensitive or PII?

Microsoft

Determine Risk

Calculating Risk with Heuristics

- Simple rules of thumb
- Needs to be something that is relevant to your business
- Microsoft's are derived from the MSRC bulletin rankings, E.g.:
 - Critical: Run malicious code, Most 'E' vulns
 - Important: Denial of service against a server
 - Moderate: Server DoS that stops once attack stops
 - Low: DoS against a client

Microsoft

Determine Risk



93

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”?

Microsoft

Plan Mitigations



106

Mitigating Threats

- Options:
 - Leave as-is
 - Remove from product
 - Remedy with technology countermeasure
 - Warn user
- What is the risk associated with the vulnerability?

Microsoft

Plan Mitigations

Mitigation Techniques



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

Microsoft

Testing Mitigations

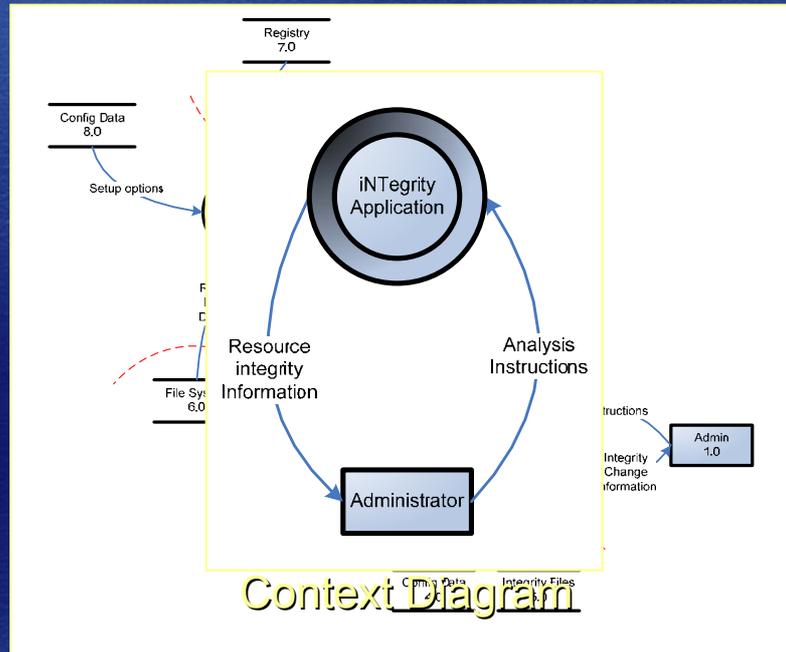
- All threats and mitigations **must** be tested
- The job of a good security tester is to find other conditions in the threat tree
- Threats have mitigations
- Mitigations can be attacked
- Spoofing
 - Authentication
 - Password guessing, brute force, Authn downgrade

Attend “Penetration Testing Principles”

Microsoft

Worked Example

iNTegrity Application



Context Diagram

The Level-0 DFD

Microsoft

Worked Example:

Identify all the DFD assets

- External Entities

- Admin (1.0)

- Processes

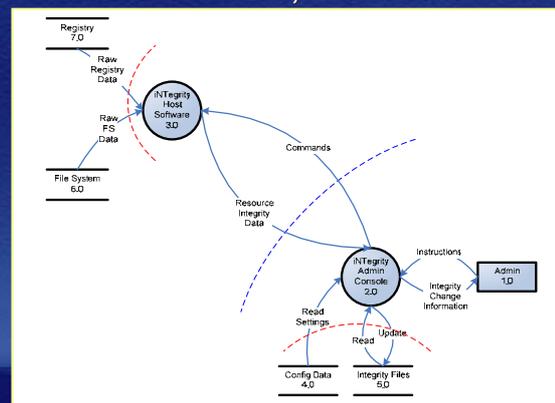
- iNTegrity Host (3.0)
- iNTegrity Admin Console (2.0)

- Data Stores

- Registry (7.0)
- File System (6.0)
- Config Data (8.0)
- Config Data (4.0)
- Integrity Files (5.0)

- Data Flows

- 8.0 -> 3.0, 7.0 -> 3.0, 6.0 -> 3.0
- 3.0 -> 2.0, 2.0 -> 3.0
- 1.0 -> 2.0, 2.0 -> 1.0
- 4.0 -> 2.0
- 5.0 -> 2.0, 2.0 -> 5.0



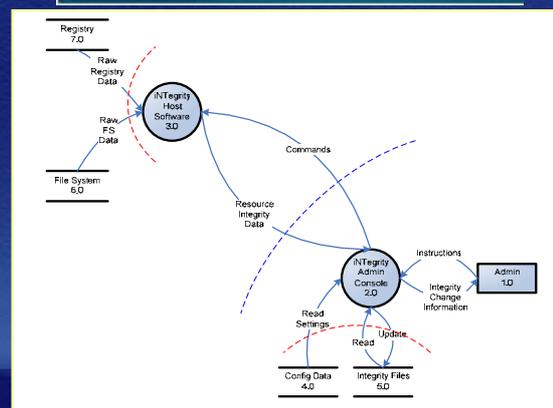
ft

Worked Example:

Identify all threat types per asset

- External Entities (SR):
 - 1
- Processes (STRIDE):
 - 3.0 and 2.0
- Data Stores (TID):
 - 7.0, 8.0, 6.0, 4.0, 5.0
- Data Flows (TID):
 - 8.0->3.0,
 - 7.0->3.0, 6.0->3.0,
 - 3.0<->2.0, 1.0<->2.0,
 - 5.0<->2.0,
 - 4.0->2.0

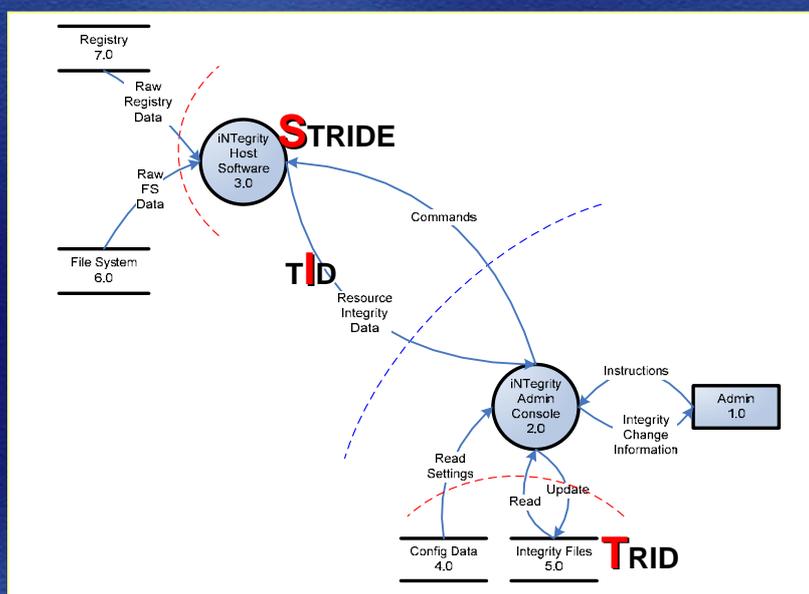
Asset	S	T	R	I	D	E
External Entity	✓		✓			
Process	✓	✓	✓	✓	✓	✓
Data Store		✓		✓	✓	
Data Flow				✓	✓	✓



Worked Example:

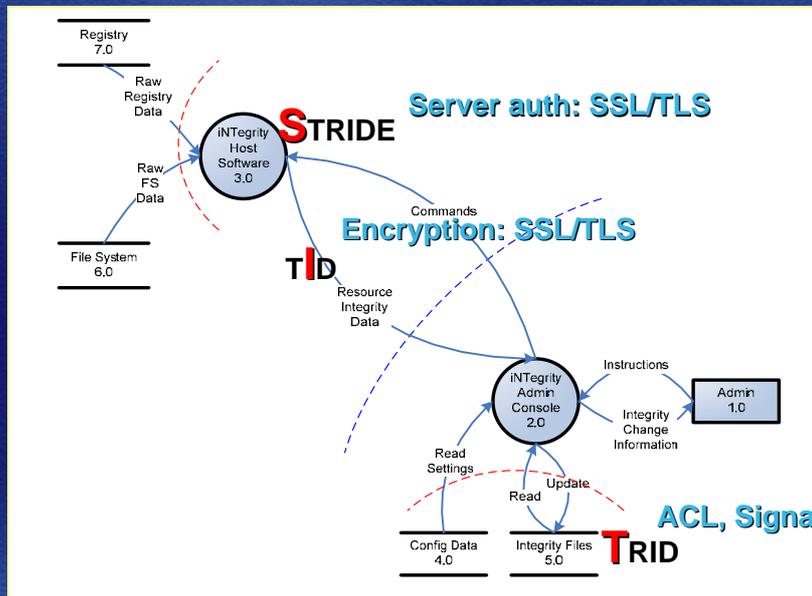
Threat Modeling and Mitigation

- Identify three threats, one for a data flow, one for a data store and one for a process



Worked Example: Threat Modeling and Mitigation

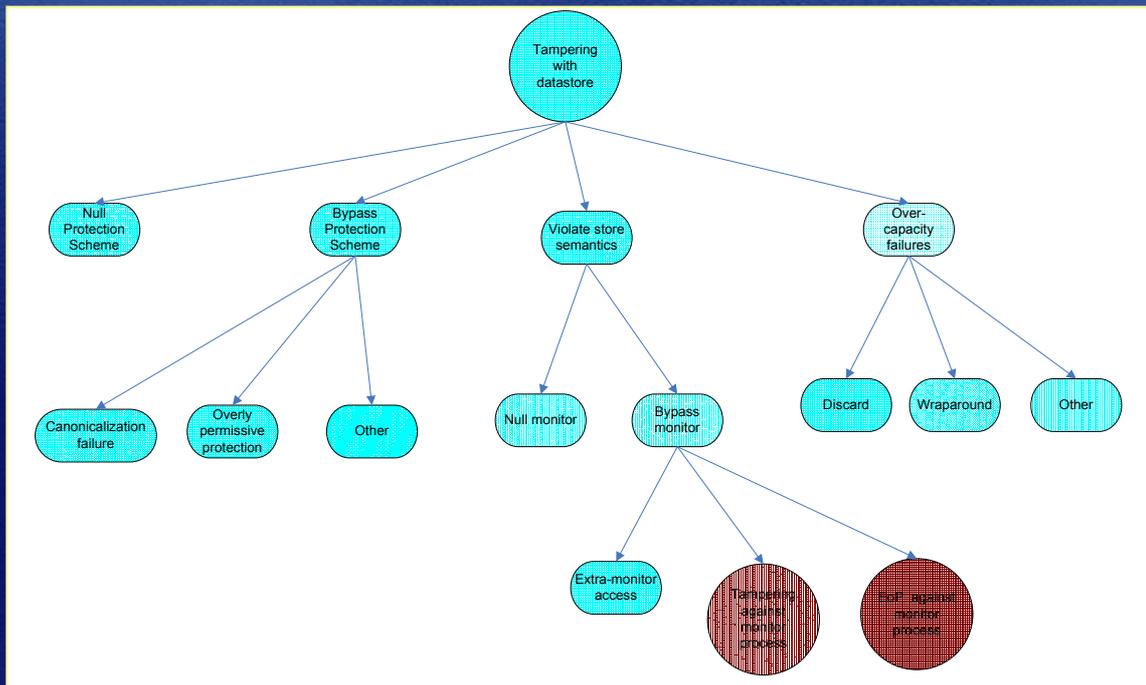
- Identify first order mitigations for each threat



Threat Tree Patterns

Threat Tree Pattern Examples

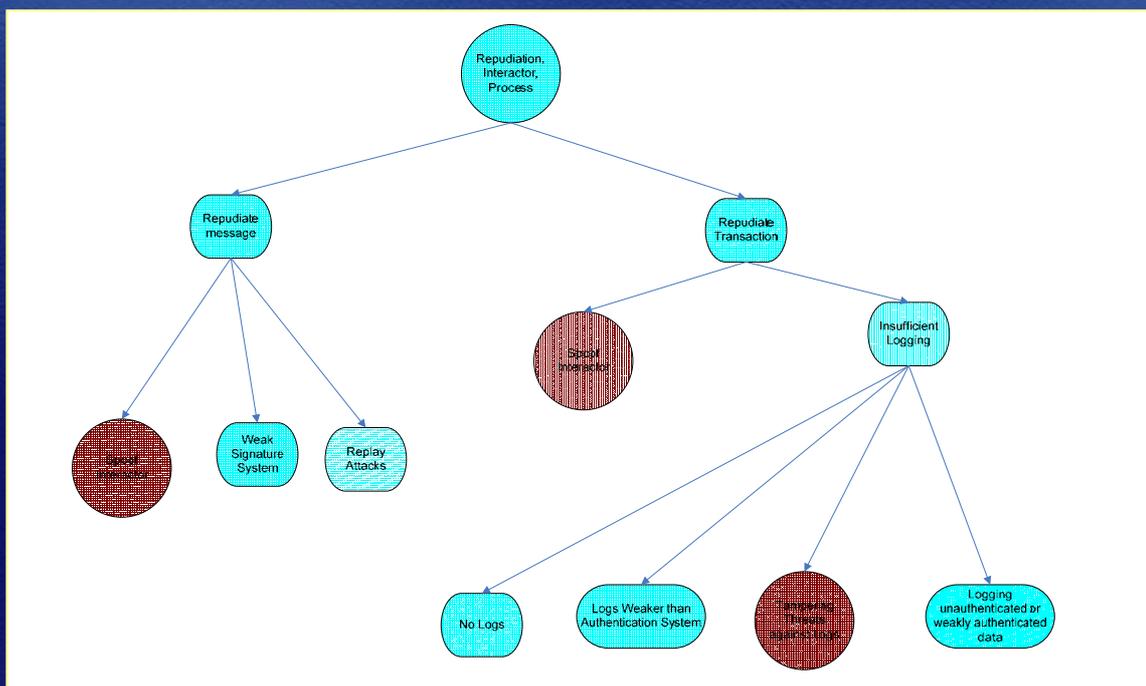
Tampering against Data Store



Microsoft

Threat Tree Pattern Examples

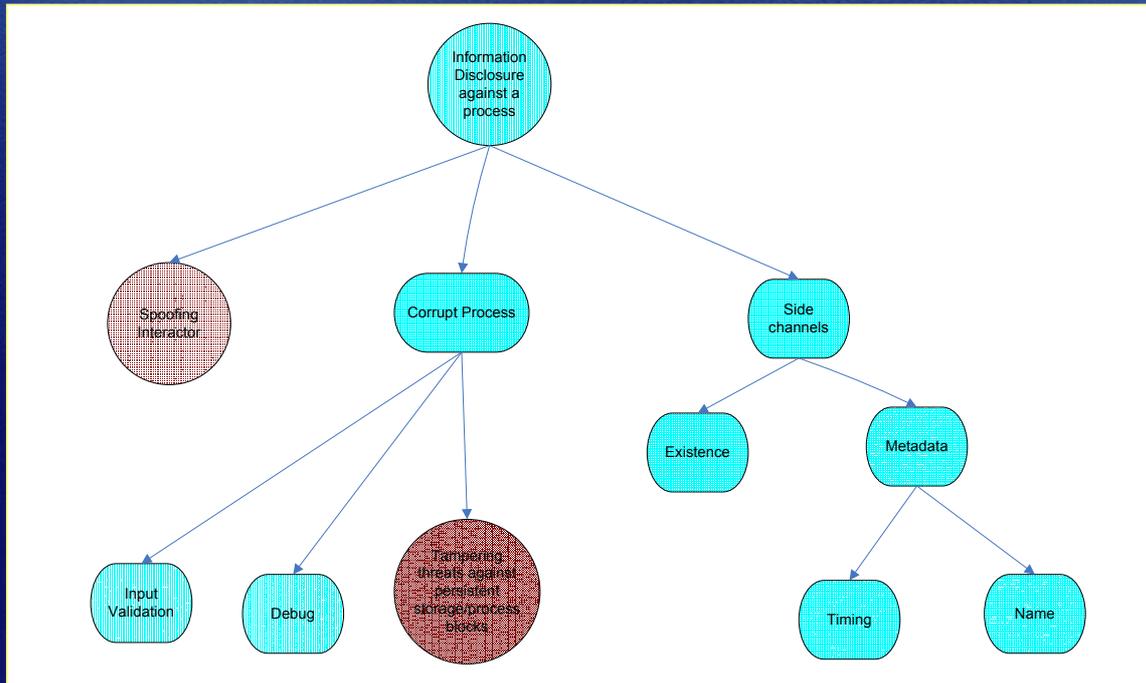
Repudiation



Microsoft

Threat Tree Pattern Examples

Information Disclosure (Process)



Microsoft

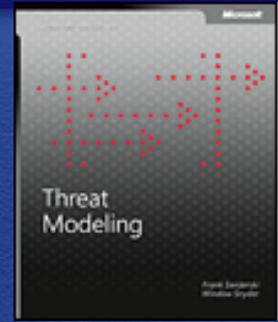
“Best Practice” Mitigations

- Mitigate condition as high in the tree as possible
- Mitigate across one side of an AND-clause
- Application defenses are better than infrastructure defenses

Microsoft

Summary

- The Threat Modeling Process
 - Define key scenarios
 - Model the application using DFDs
 - Determine threat types for each DFD element
 - Determine risk
 - Plan mitigations



Microsoft

Summary

- The Threat Modeling Resources and Tool
 - <http://msdn.microsoft.com/security/securecode/threatmodeling>
- Threat Modeling Book
 - <http://www.microsoft.com/MSPress/books/6892.asp>
- Article: Guerrilla Threat Modelling
 - <http://blogs.msdn.com/ptorr/archive/2005/02/22/GuerillaThreatModelling.aspx>



Microsoft

Requirement Phase

requirement: [re·quire·ment]

1: something wanted or needed

2: something essential to the existence or occurrence of something else



Sample: Process Activity

- Requirements
- Recommendations
- Resources
- Metrics
- Exit Criteria

Process Activities

- Education&Awareness
- Project Inception

Microsoft

Sample Security Plan

- 1 Overview
- 2 Security Hardening Activities
 - 2.1 Early Planning Phase
 - 2.2 Design Phase
 - 2.3 Development Phase
 - 2.4 Verification & Stabilization Phase
 - 2.5 Shipping Phase
 - 2.6 Miscellaneous Requirements
 - 2.7 Documentation & Samples
- Appendix
 - Appendix A
 - Appendix B
 - Appendix C

Microsoft

Sample Activity - Early Planning Phase

Activity: Employee Training

Owners: Manager

Deliverables: Completion of mandated training as soon as possible.
Completion of optional training as appropriate.

Reviewers: Security Consultant

Description:

The following training session is mandatory for all disciplines. Managers should ensure that all employees in their organization take this training session at the next available opportunity.

<http://SecurityTeam/Mandatory.html>

Reference:

Refer to the following links for additional information about other training sessions (esp for new hires): <http://SecurityTeam/Optional.html>

Microsoft

2.2 Design Phase

- Functional Spec Security Sections
- Design Spec Security Sections
- Threat Modeling

Microsoft

2.3. Development Phase

- Security Code Reviews
- Baseline Build Tools & Prefix, PreFast, FxCop Code Quality

Microsoft

2.4 Verification & Stabilization Phase

- Component Testing (includes Threat Model testing)
- Penetration Testing
- Minimal Privilege & Multi-box Testing

Microsoft

2.5 Shipping Phase

- Security Reviews of Threat Models

Microsoft

2.6 Miscellaneous Requirements

- Requirements for Lockdown & Scanning Tools
- Requirements (US Export, France, EU)

Microsoft

2.7 Documentation & Samples

- Samples Security Code Reviews
- Security Scrub of Product Documentation
- Prescriptive Guidance for Secure Deployment

Microsoft

Gathering Requirements

- Deployment Environment
- Customer Expectations
- Type of Application
- Who Develops the Security Requirements?

Microsoft

Customer Expectations

- Is this a security product?
- Is this application responsible for critical business infrastructure?
- Is This application widely deployed?
- Does it use file types that are considered safe?
- Is there external information that positions this product as more secure than its competitors?

Microsoft

Gathering Requirements

- Deployment Environment
- Customer Expectations
- Type of Application
- Who Develops the Security Requirements?

Microsoft

Deriving Requirements

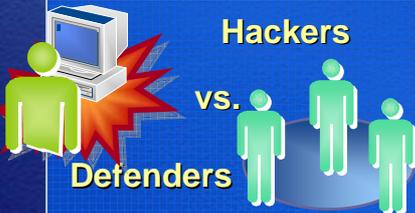
- What/Whom/How long?
- craft requirements well
- Manage Security Risks
- Standardized security analysis guidelines
- Solid System Specification

Microsoft

Security Design and Development Principles

Microsoft
Microsoft

Security Challenges

Challenges	Reasons
 <p>Hackers vs. Defenders</p>	<ul style="list-style-type: none"> ▪ Hacker needs to understand one vulnerability; defender needs to secure all entry points ▪ Hackers outnumber defenders ▪ Hackers have unlimited time
 <p>Security vs. Usability</p>	<ul style="list-style-type: none"> ▪ Secure systems become harder to use ▪ Complex and strong passwords are difficult to remember ▪ Users prefer simple passwords ▪ Developers and management think that security does not add any business value
 <p>Do I need security...</p> <p>Security as an Afterthought</p>	<ul style="list-style-type: none"> ▪ Managers do not build time for security implementation into the schedule

Microsoft

Security and Design

Why Security in the design phase?

- Saves you Money later
- If the design is not secure, the application cannot be secure
- A secure design is the starting point for all security in the application



Microsoft

Security and Design

Focus Areas in the Design Phase

- Training, education and resource assignment
- Define product security goals
 - Often driven by policy (for better or worse)
- Threat Modelling and Threat Mitigation design
- Security Design Specifications
 - Identity Management
 - Input Validation
 - Configuration and Session Management
 - Sensitive Data and Cryptography
 - Auditing and logging
 - Error and Exception Handling

Microsoft

Security and Development

Why Security during Development?

- Even if the design is solid, if the implementation is weak then the system fails
- Developers introduce the majority of security bugs
 - Buffer Overruns, Integer Overflows, SQL injection, etc.



Microsoft

Security and Development

Focus Areas in the Development Phase

- Secure coding guidelines, standards and principles
- Code and Peer Reviews
 - Incorporate security in your reviews
 - Use threat models to direct and prioritize
 - Use external assistance where appropriate
- Security Tools
 - The best tool is between your ears!
 - Trained developers are good at finding specific vulnerabilities
 - They are not good at finding all instances
 - Use and build tools to scale the problem

Microsoft

Key Security Design and Development Principles

- Living in an un-trusted world
- Living without admin
- Reducing your exposure
- Living with code failure
- Protecting your secret stuff
- Why it's good to be managed

Microsoft

Key Security Principles

Living in an un-trusted world

- Security Features ≠ Secure Features
- Don't Trust Input, Assume it's All Evil
 - Always validate data as it crosses trust boundaries
 - Don't rely on client side validation
 - Filter and Sandbox all input
- Assume external systems are insecure

Microsoft

Key Security Principles

Do you really need to be admin?

- Use Least Privilege (to build, test and run)
 - All applications should execute with the least privilege to get the job done and no more
 - You will make mistakes
 - Malicious code executing in a highly-privileged process runs with extra privileges
 - Many viruses spread because the recipient has administrator privileges
- Design for Separation of Privilege

Microsoft

Key Security Principles

Practical Least Privilege

- Elevate as necessary
 - RunAs
 - MakeMeAdmin (http://blogs.msdn.com/aaron_margosis)
 - Fast User Switching
 - Terminal Services / Remote Desktop

- Add Granular Permissions



New in Visual Studio 2005

- Permission Calculator
- Code Access Security - IntelliSense in Zone, Debugging in Zone

Microsoft

Key Security Principles

Reducing your exposure

- Reduce Your Attack Surface (early)
 - The interfaces exposed to an attacker
 - Surfaces enabled by default are most valuable to the attacker
 - Minimizing attack surface minimizes complexity
 - Use only the services that your application requires
- Employ Secure Defaults
 - Install application in a secure state
 - Users should have to enable features that reduce security
 - Users should NOT have to disable features to achieve security

Microsoft

Attack Points unter windows

- Open sockets
- Open RPC endpoints
- Open named pipes
- Services
- Services running by default
- Services running as SYSTEM
- Active Web handlers (ASP files, HTR files, and so on)
- Active ISAPI Filters
- Dynamic Web pages (ASP and such)

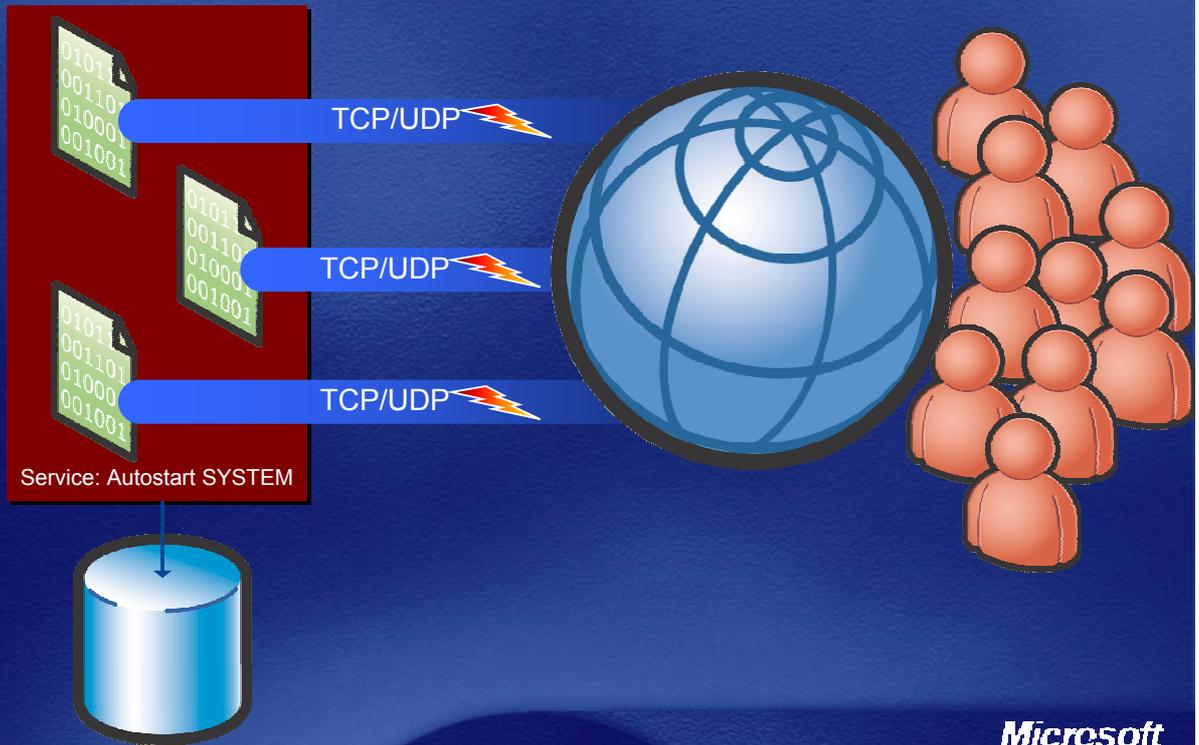
Microsoft

Attack Points unter windows

- Dynamic Web pages (ASP and such)
- Executable virtual directories
- Enabled Accounts
- Enabled Accounts in admin group
- Null Sessions to pipes and shares
- Guest account enabled
- Weak ACLs in the file system
- Weak ACLs in Registry
- Weak ACLs on shares

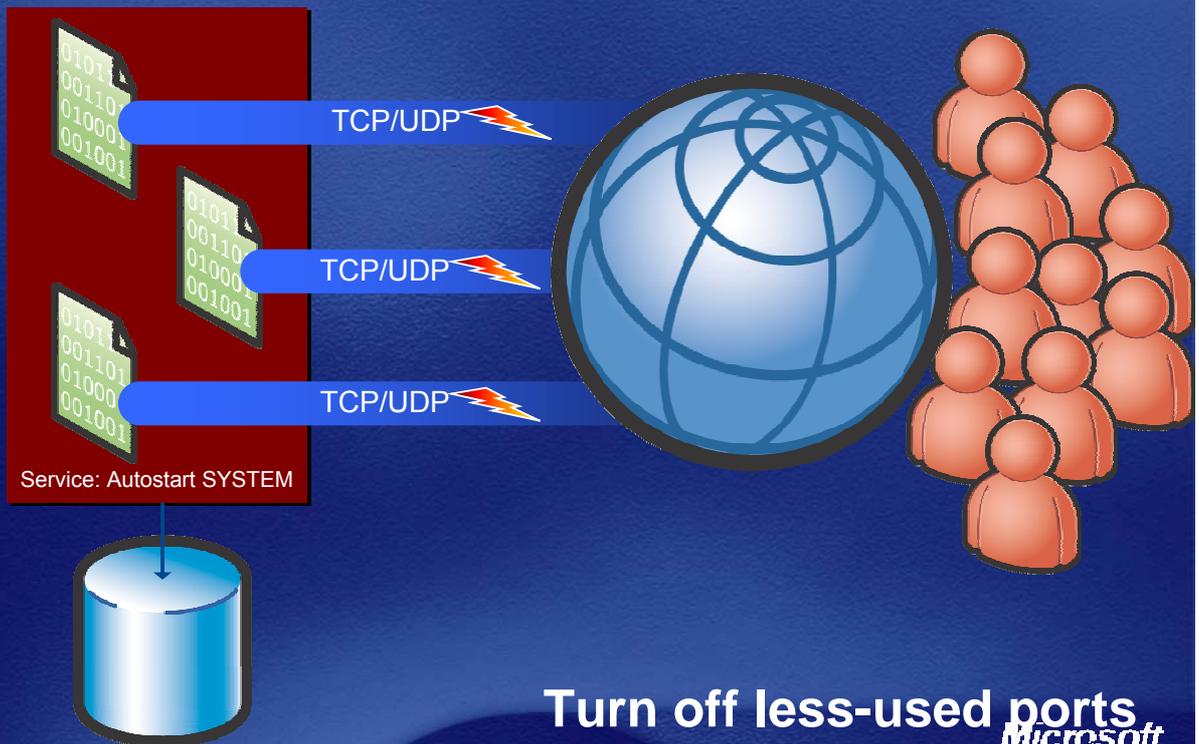
Microsoft

Attack Surface Reduction Ideas



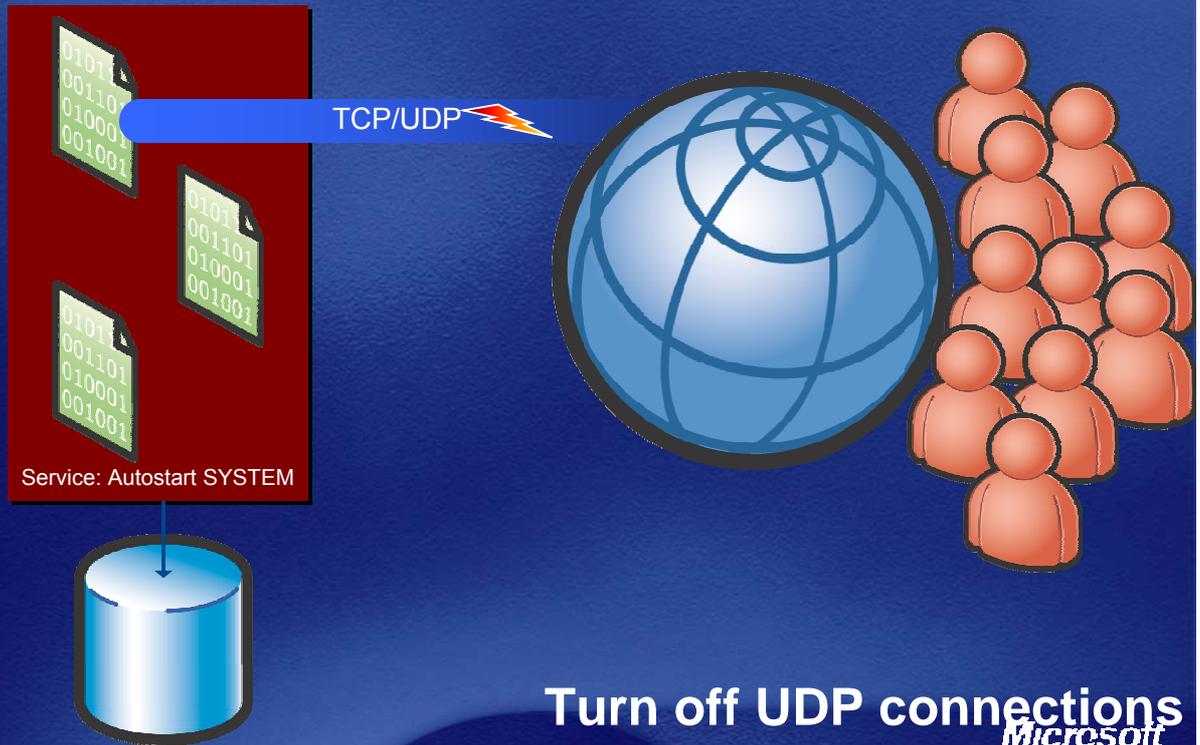
Microsoft

Attack Surface Reduction Ideas



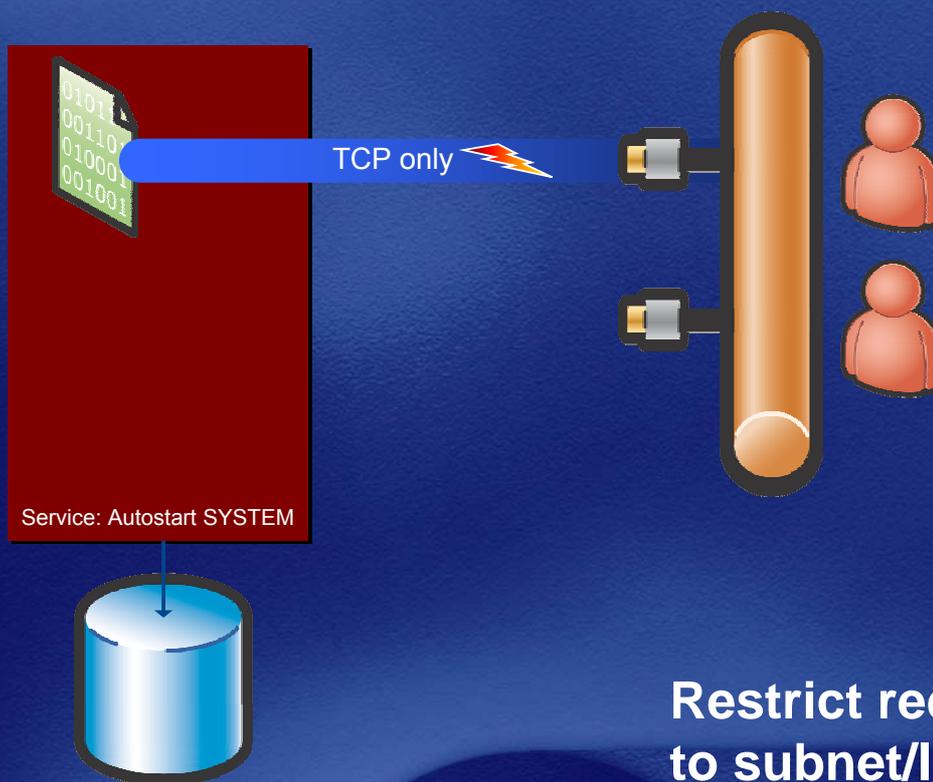
Turn off less-used ports
Microsoft

Attack Surface Reduction Ideas



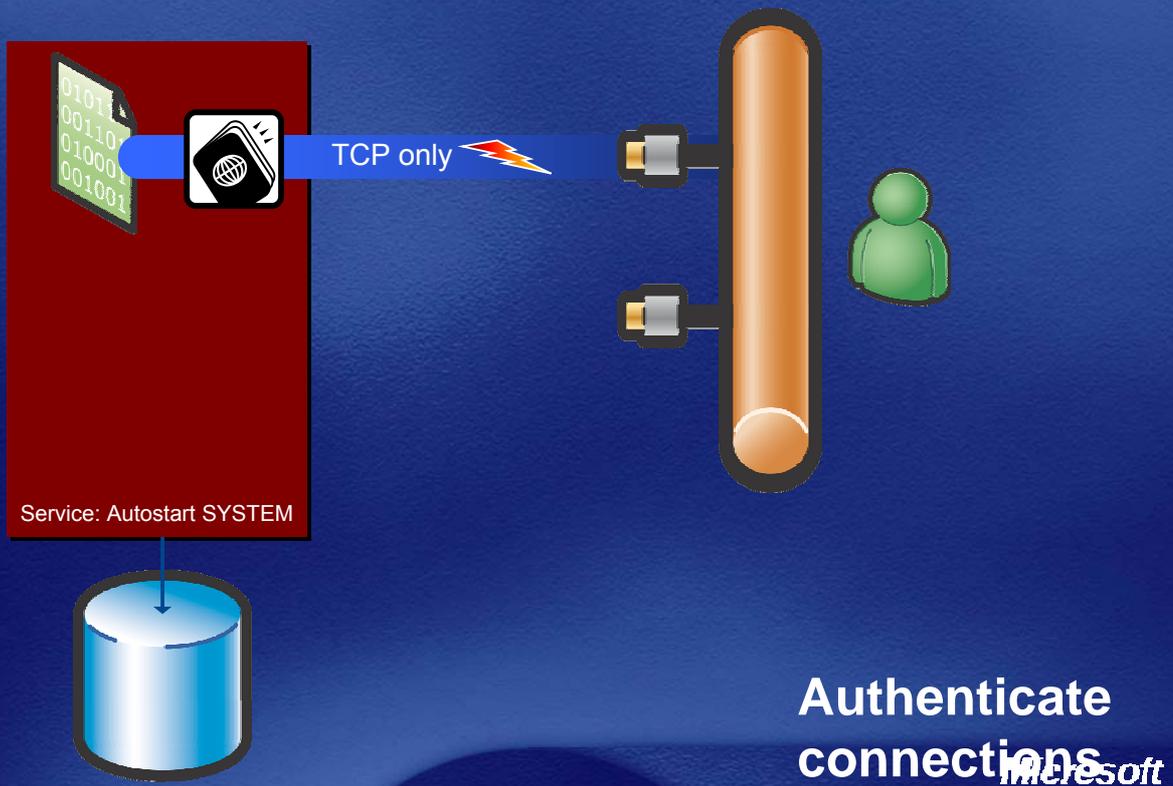
Turn off UDP connections
Microsoft

Attack Surface Reduction Ideas

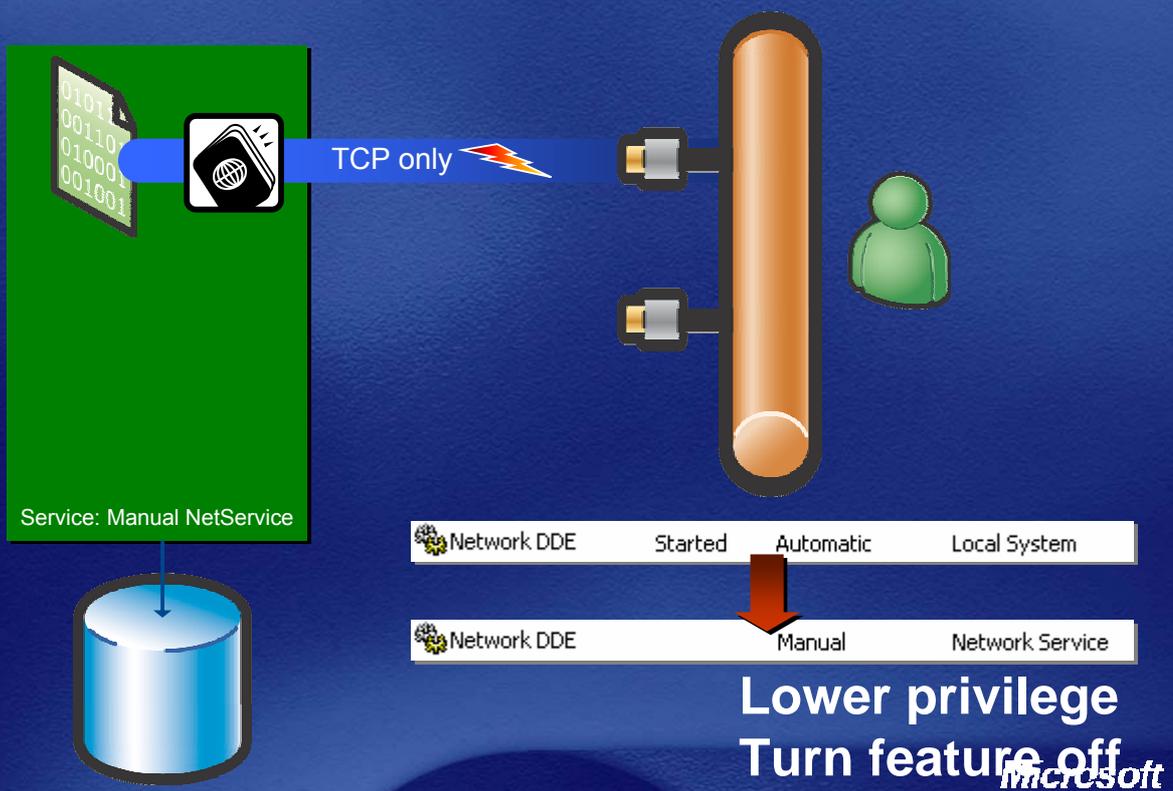


Restrict requests
to subnet/IP range
Microsoft

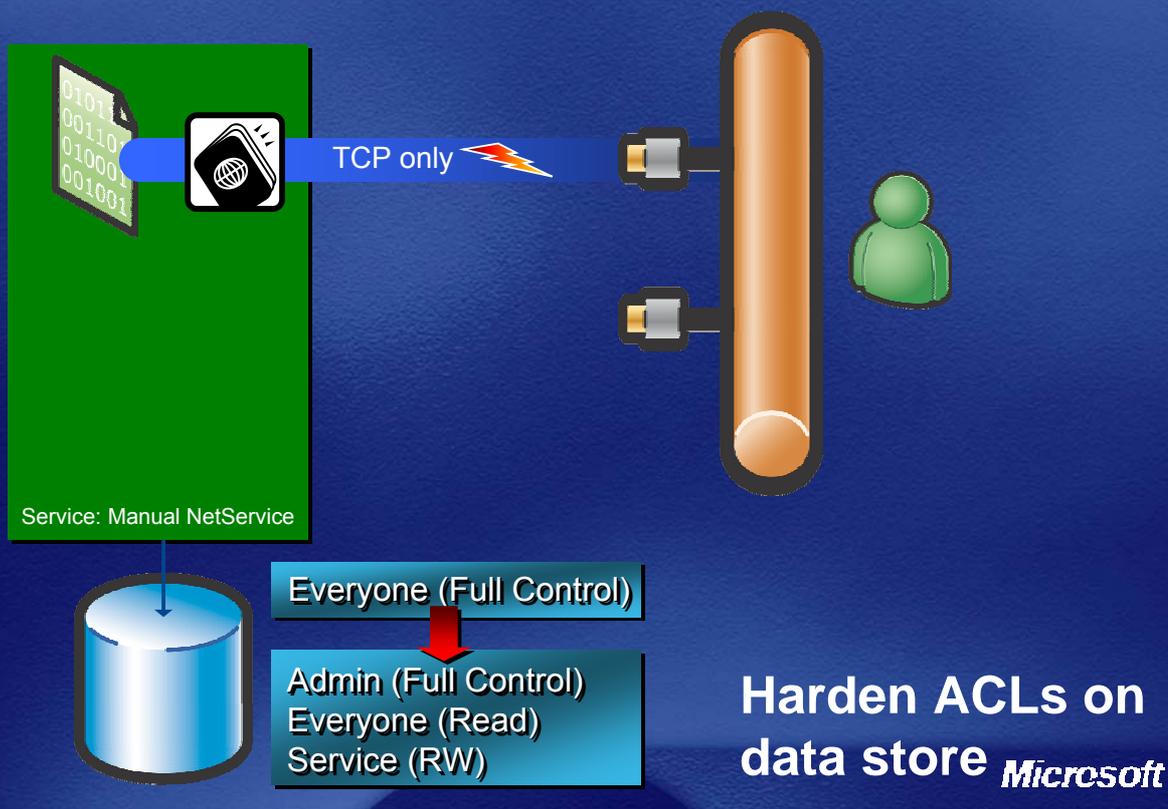
Attack Surface Reduction Ideas



Attack Surface Reduction Ideas



Attack Surface Reduction Ideas



Key Security Principles

Code fails... really, it does!

- Plan on Failure, Fail in a Secure Mode
 - Failure code path should be most secure
 - Don't log detailed error to the client
- Learn From Mistakes (yours and theirs)
 - Understand them; and fix them correctly
 - Build security into your response plans
- Defence in Depth
 - Threat risk goes down as threat difficulty goes up
 - Driven by policy

Key Security Principles

Protecting your secret stuff

- Treat the storage medium as if it were at risk
 - Confidentiality and Integrity
- Avoid Storing Secrets
 - If required, store hashes of secrets
 - Take appropriate security measures
- Never Depend on “Security by Obscurity”
 - Obscurity cannot provide real security
 - Eg: roll your own crypto, hiding security keys in files, relying on undocumented registry keys



New in Visual Studio 2005

- Data Protection API built into .NET Framework 2.0

Microsoft

Key Security Principles

Why it's good to be managed

- .NET:
 - Mitigates the most prevalent security issue – the buffer overrun
 - Has a rich code access security model
 - Simplifies the programming model



New in Visual Studio 2005

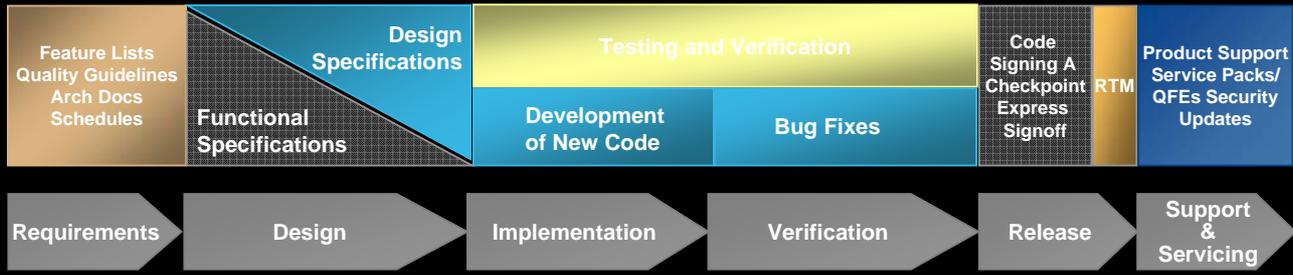
- ASP.NET v2 security helpers
- Static Analysis Tools: PREfast and FxCop
- VB.NET My Classes
- AppVerifier
- C++ SafeCRT Libraries, /GS Switch

Microsoft

Security Deployment Lifecycle Tasks and Processes



Traditional Microsoft Software Product Development Lifecycle Tasks and Processes



Microsoft

Microsoft®

Your potential. Our passion.™

Microsoft