Threat Modelling

Rune Zakariassen
Developer and Platform Strategy Group

BRINGING CIVILIZATION TO ITS KNEES...
Goths
HACK
HACK
HACK
Vandals
HACK
HACK
HACK
Huns
HACK
HACK
Geeks
HACK
HACK
HACK
HACK
KEVIN SIERS ©2000 THE CHARLOTTE OBSERVER

Sometime in the mid 1990's

- A developer working on DCOM wrote some simple code to copy the server name from an object activation request
- The process listening for object activation requests on a well known port (135) is RPCSS…and it runs as LOCALSYSTEM
- The buffer allocated was the max size for a computer name, but nobody ever checked for a buffer overrun…and the rest is history

Buffer Overrun Examples

DCOM Remote Activation (MS03-026)

```
error_status_t _RemoteActivation(..., WCHAR *pwszObjectName, ... )  {
    *phr = GetServerPath( pwszObjectName, &pwszObjectName);
    ...
}

HRESULT GetServerPath(WCHAR *pwszPath, WCHAR **pwszServerPath ){
    WCHAR * pwszFinalPath = pwszPath;
    WCHAR wszMachineName[MAX_COMPUTERNAME_LENGTH_FQDN + 1];
    hr = GetMachineName(pwszPath, wszMachineName);
    *pwszServerPath = pwszFinalPath;
}

HRESULT GetMachineName(
    WCHAR * pwszPath,
    WCHAR   wszMachineName[MAX_COMPUTERNAME_LENGTH_FQDN + 1]) {

    pwszServerName = wszMachineName;
    LPWSTR pwszTemp = pwszPath + 2;
    while ( *pwszTemp != L'\\' )
        *pwszServerName++ = *pwszTemp++;
    ...
}
```

Sitting on port 135

Copies buffer from the network until \‘ char found

Blaster Virus

- July 16th 2003 – The Last Stage of Delirium Research Group informs Microsoft of a flaw in DCOM object activation but does not publish technical details
- July 25th 2003 – XFocus releases technical details of the flaw
- August 11th 2003 – nearly 4000 computers an hour are infected in the first week as un-patched systems are attacked
- Ultimately, over 1.5 million computers are infected
- 3, 370, 000 PSS calls in Sept ‘03 (normal virus volume is 350,000)
- “After two decades'worth of Swiss cheese software security, the world's biggest supplier of operating system software has run out of excuses. –Charles Cooper – CNET
- Estimated economic impact: $2 billion

Need a better way…

- Until you know your threats, you cannot secure your system
- Old way – based on experience and opinion (unstructured)
  - Security features are applied in a haphazard manner without knowing precisely what threats each feature is supposed to address.
  - How do you know when your application is –secure enough‖?
  - How do you know the areas where your application is still vulnerable?
- Threat modeling is a structured process by which you:
  - Systematically identify and rate the threats that are most likely to affect your system
  - Address threats with appropriate countermeasures in a logical order, starting with the threats that present the greatest risk.

What is Threat Modeling?

- Structured approach
  - Identify Threats
    - Based upon the application architecture
    - Rating and Prioritization
  - Define Countermeasures
    - Start with threats that present greatest risk
  - Measure Results
    - Impact, Probability, Cost, Benefit

Benefits of Threat Modeling

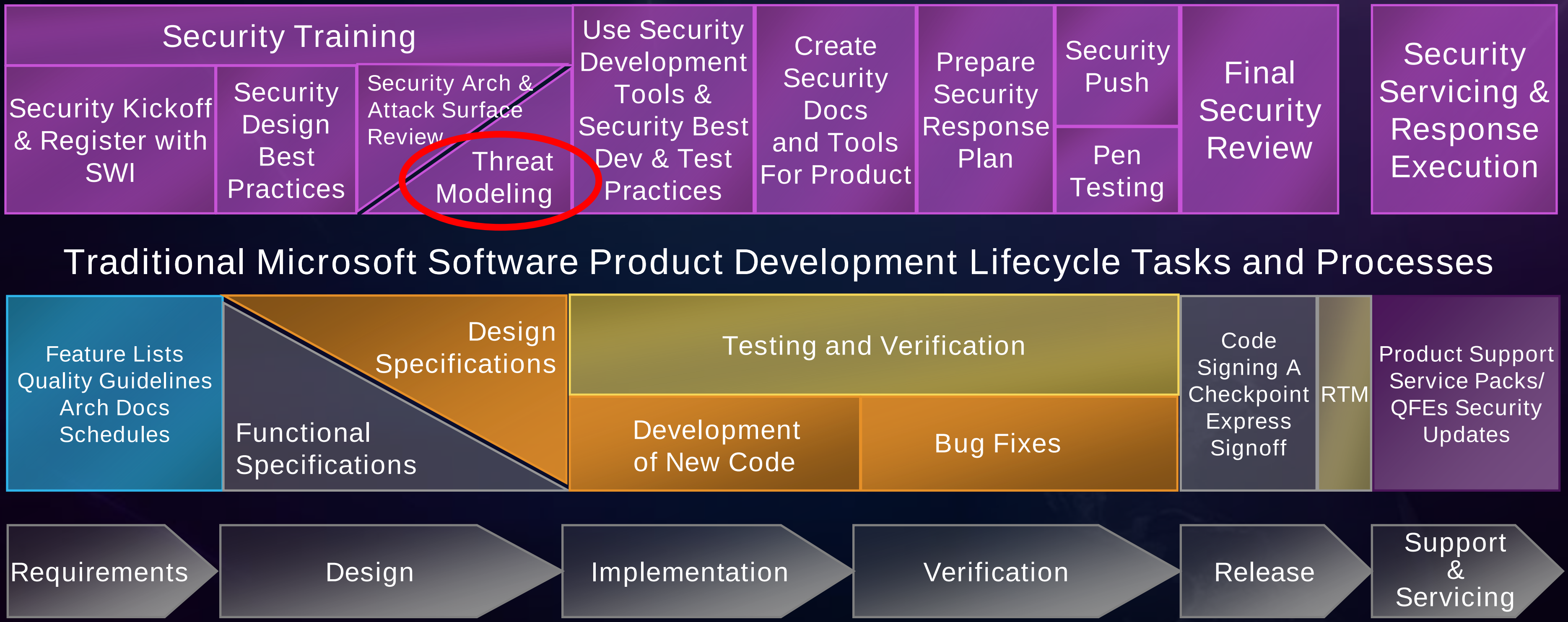
- Benefit
  - Allows business to define ―Secure Enough‖
  - Cost efficient and effective
  - Not a haphazard or random ―shotgun‖ approach
  - Document remaining vulnerabilities

When is Threat Modeling done?

- Initial effort: Immediately after functional specifications are completed
- Reviewed and/or Updated:
  - Validated at Design Review
  - Reviewed at Code Complete by Application Team
  - Reviewed at System Test by Application Team
  - Validated at Security Assessment (UAT)
  - Reviewed on Implementation by Application Team

Microsoft SDL

Security Development Lifecycle



Security Training

Security Kickoff & Register with SWI

Security Design Best Practices

Security Arch & Attack Surface Review

Threat Modeling

Use Security Development Tools & Security Best Dev & Test Practices

Create Security Docs and Tools For Product

Prepare Security Response Plan

Security Push

Pen Testing

Final Security Review

Security Servicing & Response Execution

Traditional Microsoft Software Product Development Lifecycle Tasks and Processes

Feature Lists
Quality Guidelines
Arch Docs
Schedules

Design Specifications

Functional Specifications

Testing and Verification

Development of New Code

Bug Fixes

Code Signing A Checkpoint Express Signoff

RTM

Product Support
Service Packs/
QFEs Security
Updates

Requirements

Design

Implementation

Verification

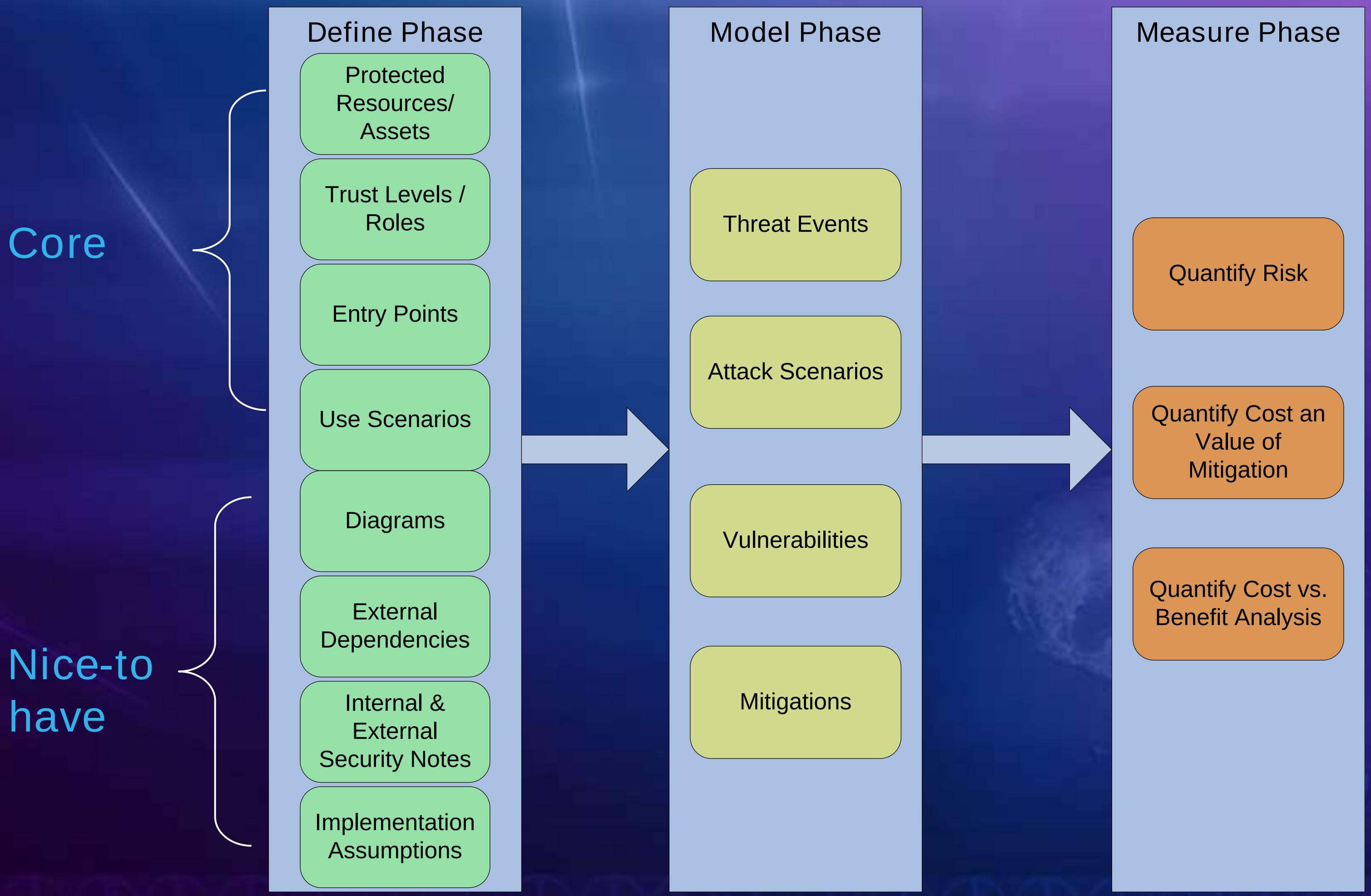
Release

Support & Servicing

Threat Modeling Process

- Asset Identification - Architecture Overview - Decompose Application	- Define - Assets - Entry Points - Roles
- Identify Threats - Document Threats	- Model - Threat Events - Attack Scenarios
- Rate Threats	- Measure - Risk - Mitigation Cost - Benefit

Threat Modeling Process: Define Phase



Enumerate Threats

- The critical point in creating a usable threat model
- Also the most difficult step in the process.
- Brainstorm attack hypotheses
  - Every idea should be considered no matter how remote
- For a given entry point where a specific external entity interfaces with the system…
  - What security-critical processing occurs?
  - What might an attacker try to do to thwart that processing?
  - How would an attacker use an asset outside of its expected use?

Enumerate Threats

- It is important not to confuse threats with vulnerabilities
  - A threat is simply what an adversary might *try* to do to a protected resource in the system
  - A vulnerability is a specific way that a threat is exploitable based on an unmitigated attack path
- Threats become more specific as the process model becomes more specific

Enumerate Threats

- Threats can apply a verb to an asset (adversary does something *to* an asset)
  - Adversary *captures* [password data] using a sniffer
- Or, they can result *in* an asset
  - Adversary *supplies a path name that exceeds MAX_PATH*, causing a buffer overflow that may result in the [ability to execute native code]
- In either case, threats are verbs

Identify Threats Using STRIDE

Types of threats	Examples
Spoofing	- Forging e-mail messages - Replaying authentication packets
Tampering	- Altering data during transmission - Changing data in files
Repudiation	- Deleting a critical file and denying it - Purchasing a product and denying it
Information disclosure	- Exposing information in error messages - Exposing code on Web sites
Denial of service	- Flooding a network with SYN packets - Flooding a network with forged ICMP packets
Elevation of privilege	- Exploiting buffer overruns to gain system privileges - Obtaining administrator privileges illegitimately

Tool: Threats Table

Threats	
Threat	
ID	1
Name	Adversary gains access to the remote administration interface resulting in access to the phone configuration.
Description	The Phone 1.0 has a remote administration interface that allows an authorized user to configure it via the PSTN. The interface is disabled by default, but can be enabled using the local keypad.
STRIDE Classification	Tampering Information Disclosure Denial of Service Elevation of Privilege
Mitigated?	No
Known Mitigation	If the remote administration interface is enabled, the end user should change the default password.
Investigation Notes	(none)
Entry Points	(6) Remote Administration (3) Telephone Line (2) Keypad
Assets	(5) Phone configuration

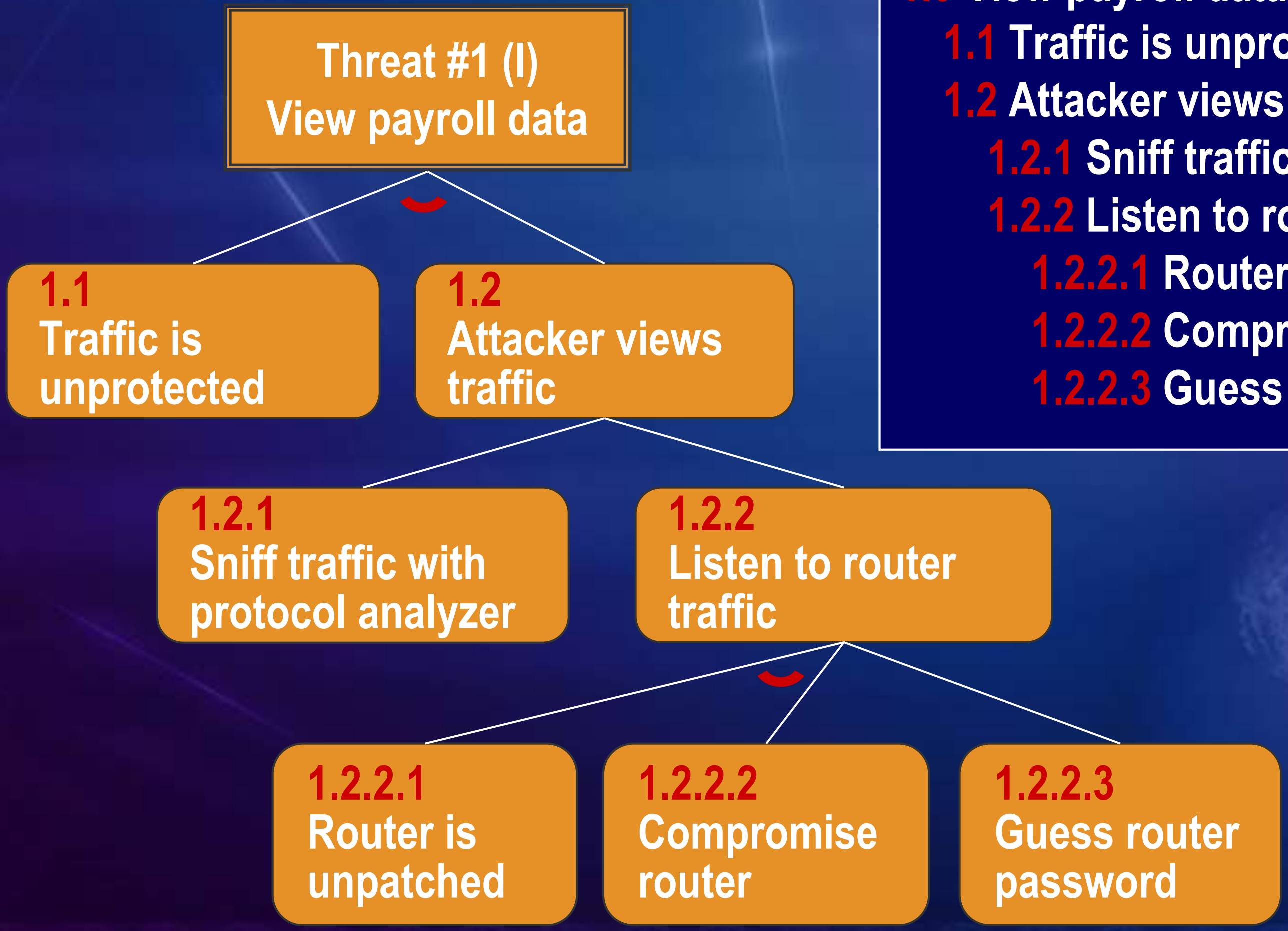
Determine if Vulnerabilities Exist

- A Threat that has no (or insufficient) mitigating factors results in a Vulnerability—that is, something an attacker can exploit
- For each Threat, determine if there are sufficient protections. Enumerate those that are Vulnerabilities
- Use Threat Trees to determine if a threat translates into a vulnerability

Tool: Threat Trees

- Used in Threat Modeling to analyze how a threat might be accomplished
- A threat tree is a hierarchical representation of conditions, with the root node being the threat.
- An *attack path* is a route from a leaf condition to the root threat, inclusive of any *and* condition.
- Threat Trees are used to determine valid attack paths for a threat. That is, any attack path that does not have a mitigating node is classified as a vulnerability.
- In its most basic form, a Threat Tree consists of a single Threat, and multiple Mitigated Conditions and Unmitigated Conditions.

Threat/Attack Trees



Threat #1 (I)
View payroll data

1.1
Traffic is unprotected

1.2
Attacker views traffic

1.2.1
Sniff traffic with protocol analyzer

1.2.2
Listen to router traffic

1.2.2.1
Router is unpatched

1.2.2.2
Compromise router

1.2.2.3
Guess router password

1.0 View payroll data (I)
- 1.1 Traffic is unprotected (AND)
- 1.2 Attacker views traffic
  - 1.2.1 Sniff traffic with protocol analyzer
  - 1.2.2 Listen to router traffic
    - 1.2.2.1 Router is unpatched (AND)
    - 1.2.2.2 Compromise router
    - 1.2.2.3 Guess router password

Use DREAD

- Damage (Impact)
  - How great can the damage be?
- Reproducibility (Probability)
  - How often does the attempt work?
- Exploitability (Probability)
  - How much expertise is required to affect it? What are the pre-conditions?
- Affected Users (Impact)
  - How many users are affected? What config options are used?
- Discoverability (Probability)
  - Likelihood if it goes unpatched, it would be discovered by security researchers, hackers

Calculate Risk Using DREAD

- Risk = Impact * Probability
- Impact = (DREAD)
  - Damage
    - Note: Damage is assessed in terms of Confidentiality, Integrity and Availability
  - Affected Users
    - How large is the user base affected?
- Probability = (DREAD)
  - Reproducibility
    - How difficult to reproduce? Is it scriptable?
  - Exploitability
    - How difficult to use the vulnerability to effect the attack?
  - Discoverability
    - How difficult to find?

DREAD Ratings

	Rating	High	Medium	Low
D	Damage Potential	Subvert the security system Get full trust authorization Run as administrator Upload content	Leaking sensitive information	Leaking trivial information
R	Reproducibility	The attack can be reproduced every time and does not require a timing window.	The attack can be reproduced, but only with a timing window and a particular race situation	The attack is very difficult to reproduce, even with knowledge of the security hole
E	Exploitability	A novice programmer could make the attack in a short time	A skilled programmer could make the attack, then repeat the steps	The attack requires an extremely skilled person and in-depth knowledge every time to exploit

DREAD Ratings

	Rating	High	Medium	Low
A	Affected users	All users, default configuration, key Customers	Some users, non-default configuration	Very small percentage of users, obscure feature; affects anonymous users
D	Discoverability	Published information explains the attack. The vulnerability is found in the most commonly used feature and is very noticeable	The vulnerability is in a seldom-used part of the product, and only a few users should come across it. It would take some thinking to see malicious use	The bug is obscure, and it is unlikely that users will work out damage potential

Tool: Vulnerabilities Table

Vulnerabilities	
Vulnerability	
ID	1
Name	A user gains access to the administration interface.
Description	If the default password is left unchanged, and the remote administration interface is enabled, then remote anonymous users can easily obtain access to the interface.
STRIDE Classification	Tampering Information Disclosure Denial of Service Elevation of Privilege
DREAD Rating	7.6 (D: 10, R: 10, E: 8, A: 2, D: 8)
Corresponding Threat	1 (Adversary gains access to the remote administration interface resulting in access to the phone configuration.)

Tool: Vulnerabilities Table

Vulnerabilities	
Vulnerability	
ID	2
Name	A user takes advantage of the password ring buffer
Description	If a user takes advantage of the fact that the password for the admin interface is a ring buffer, the attack could take significantly less than 10^8 attempts
STRIDE Classification	Tampering Information Disclosure Denial of Service Elevation of Privilege
DREAD Rating	3.8 (D: 10, R: 5, E: 1, A: 2, D: 1)
Corresponding Threat	1 (Adversary gains access to the remote administration interface resulting in access to the phone configuration.)

Choose Mitigation Strategies

- Decide what to do about each vulnerability.
- Fix it? Provide a work-around? Notify the end user? Do nothing?
- What is the risk associated with a vulnerability?

Mitigation Strategies

Threat Type	Mitigation Techniques
Spoofing Identity	Appropriate authentication Protect secret data Don‘t store secrets
Tampering with data	Appropriate authorization Hashes Message authentication codes Digital signatures Tamper-resistant protocols
Repudiation	Digital signatures Timestamps Audit trails

Mitigation Strategies

Threat Type	Mitigation Techniques
Information disclosure	● Authorization ● Privacy-enhanced protocols ● Encryption ● Protect secrets ● Don‘t store secrets
Denial of service	● Appropriate authentication ● Appropriate authorization ● Filtering ● Throttling ● Quality of service
Elevation of privilege	● Run with least privilege

Calculate the Value of Mitigations

- Mitigation Value = $M/D
  - Formula: Mitigation Cost / DREAD(delta)
  - Result: $ per DREAD point reduced
  - Per Vulnerability per Attack Scenario

Example:

Mitigation Value

= $5,000

	Before Mitigation	Mitigation Cost	After Mitigation	DREAD Delta
Damage	5	$0	5	0
Reproducibility	3	$20,000	1	2
Exploitability	4	$5,000	1	3
Affected Users	5	$0	5	0
Discoverability	3	$10,000	1	2
Total		$35,000		7

Knowing when you are done

- A complete model is one that explores all entry points.
- The model should also consider external dependencies (i.e., are you dependent on filesystem normalization matching your internal normalization).
- Threat models should include participation and review by persons not familiar with the components.

Knowing when you are done

- If the component is not yet implemented, an update to the threat model should be done post-implementation.
- Finally, models are done when there are no more threats left that require further investigation. It does not depend on the number of vulnerabilities found.

Summary

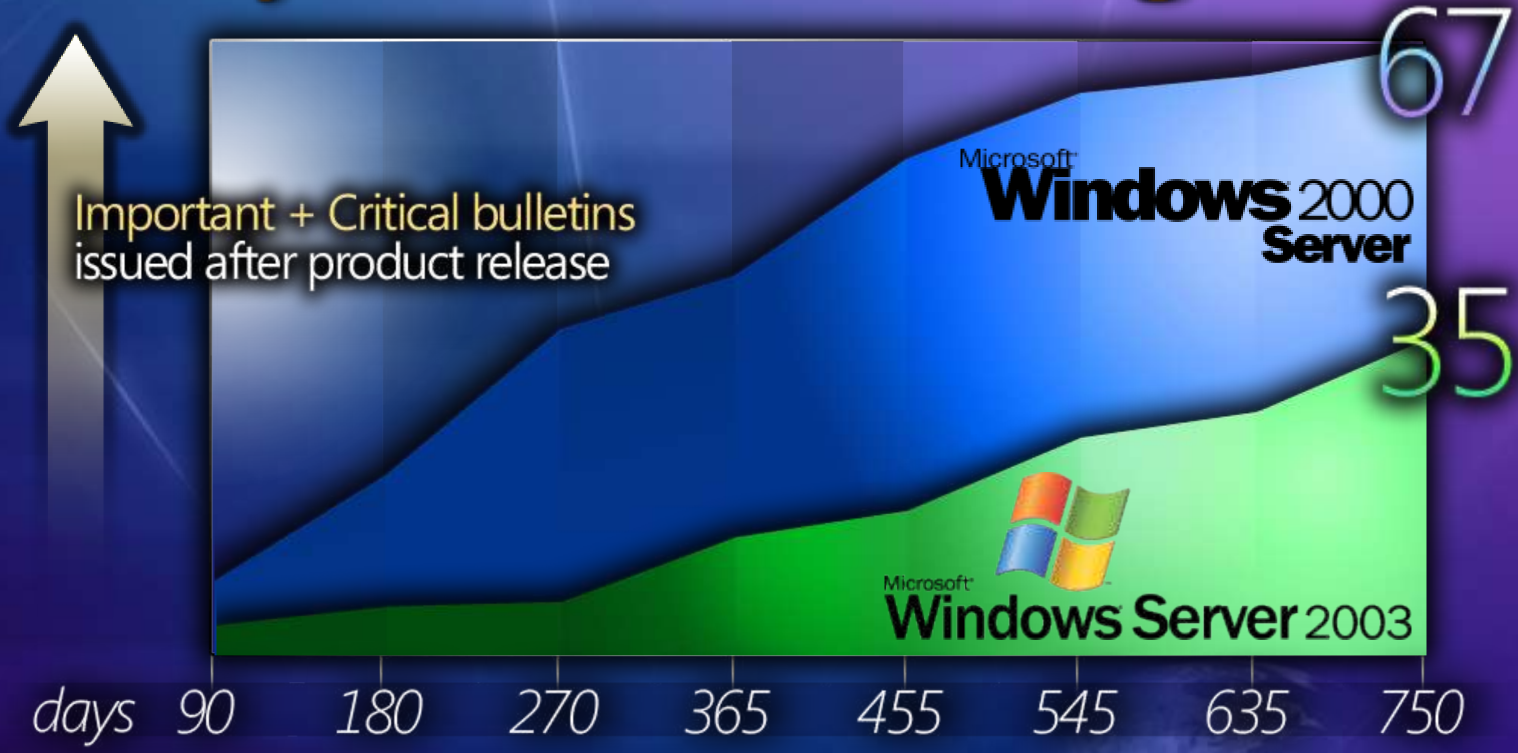
- Threat modeling is a structured approach to discovering the vulnerabilities of your system
- The threat model is a living document which must be updated each time the system is updated
- Threat modeling is the best way to identify and manage security risks of your system

Microsoft®

Your potential. Our passion.™

This presentation is for informational purposes only. Microsoft makes no warranties, express or implied, in this summary.

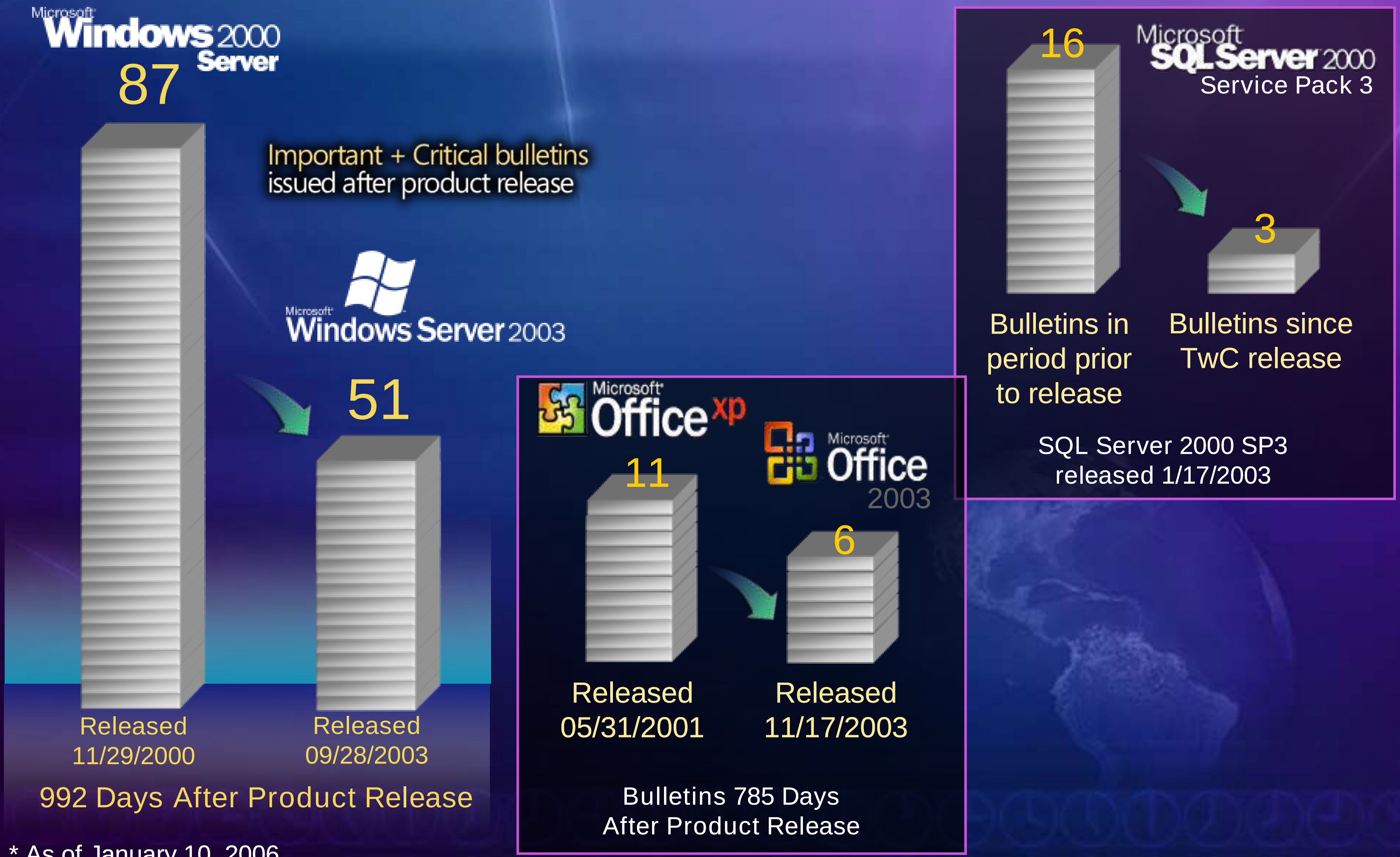
Security Focus Yielding Results



Source: Microsoft Security Bulletin Search

- Security Development Lifecycle working
- 200M Windows XP SP2 downloads
- Windows Server 2003 SP1 1.4M downloads
- Red Hat adopting our security response ratings

Focus Yielding Results



Important + Critical bulletins issued after product release

992 Days After Product Release

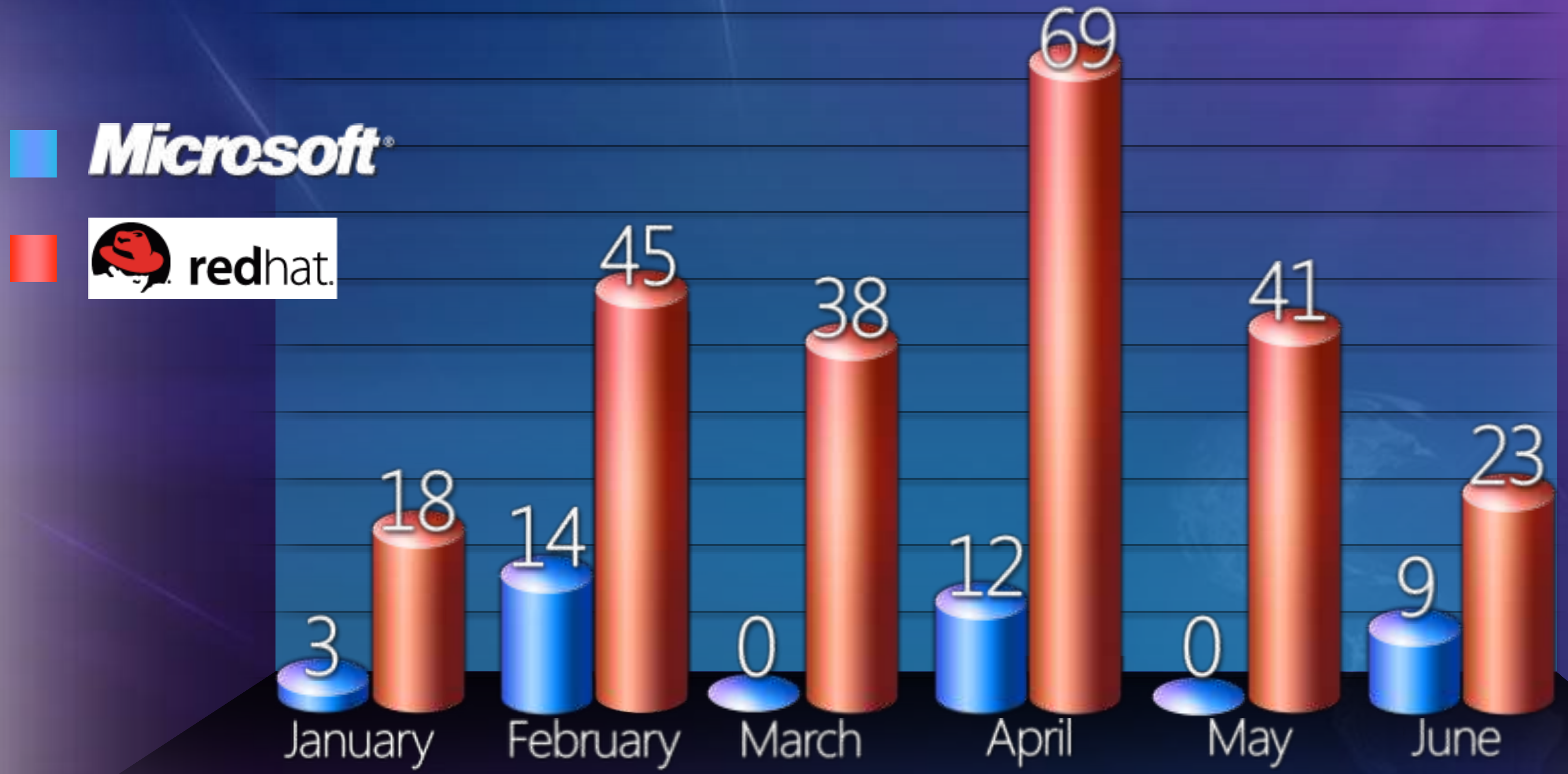
Bulletins 785 Days After Product Release

SQL Server 2000 SP3 released 1/17/2003

* As of January 10, 2006

Security Quality

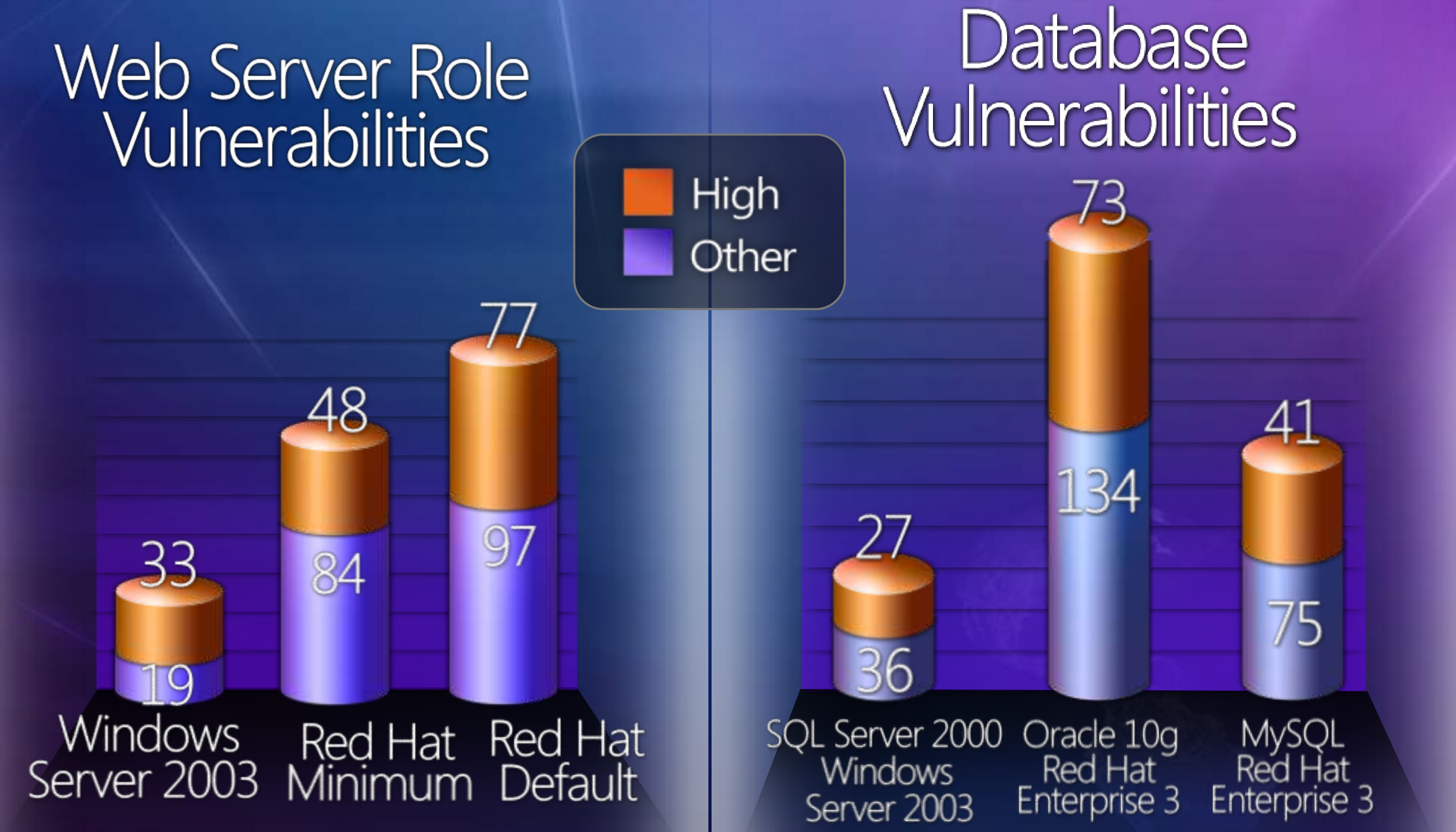
Vulnerabilities Corrected by Bulletins in 2005
Windows Server 2003 vs. Red Hat Enterprise Linux 3



Totals: Microsoft = 38 Red Hat = 234

Source: Vendor’s Public Security Bulletins as of July 2005

Security Quality



Source: "*Security Innovation* (March 2005):
"Role Comparison Report: Web Server Role"

ource: "*Security Innovation* (June 2005):
"Role Comparison Security Report: Database Server Role"

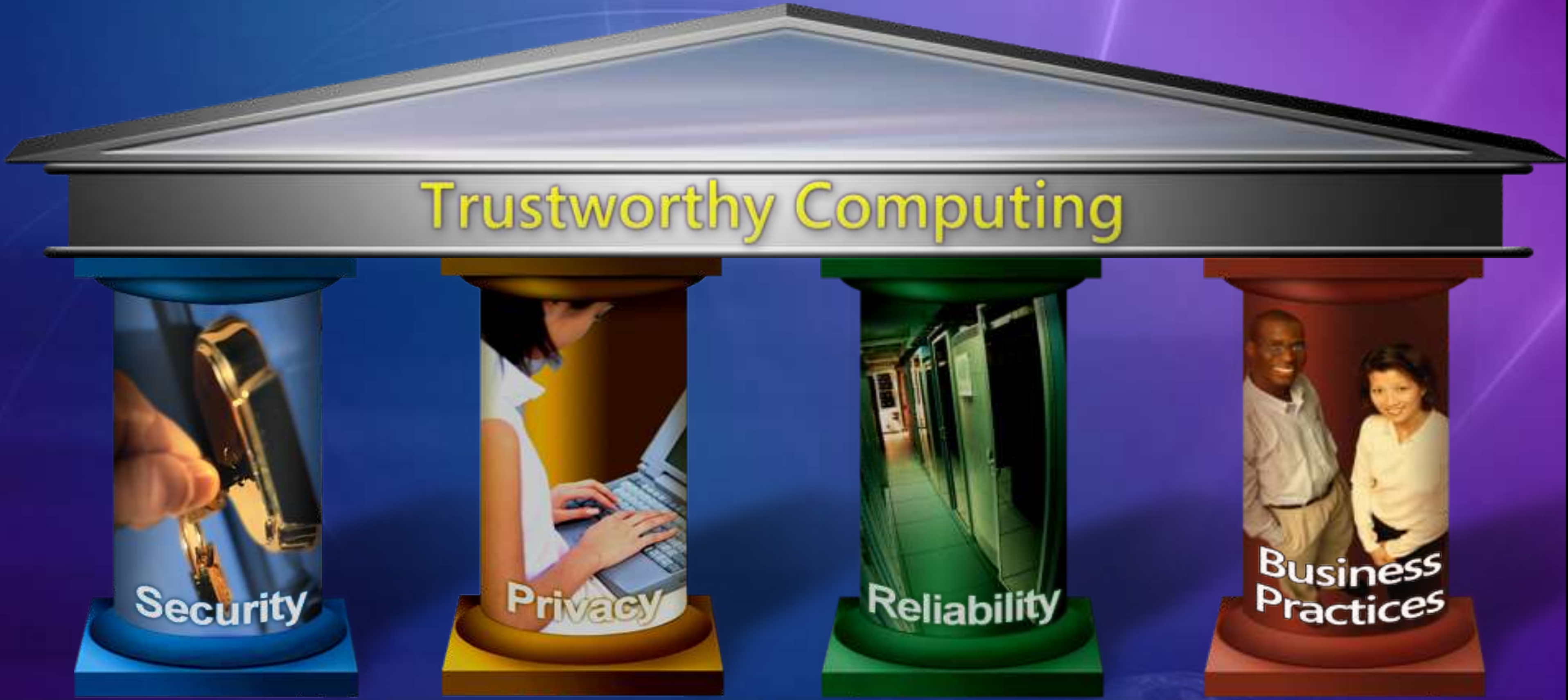
Security Quality

–Microsoft has significantly improved the security of its shipping products since the adoption of its security development life cycle. The first OS product to ship since Microsoft adopted its SDL was Windows Server 2003 (with IIS 6). Windows 2003 has had sufficient operational testing to be suitable for security-critical applicationsǁ

Neil McDonald
Group Vice President and Research Director
Gartner, Inc
(From Gartner Symposium May 2005)

Microsoft Security Progress & Strategy

Microsoft Corporation



Trustworthy Computing

Security

- Secure against attacks
- Protects confidentiality, integrity of data and systems
- Manageable

Privacy

- Protects from unwanted communication
- Controls for informational privacy
- Products, online services adhere to fair information principles

Reliability

- Predictable, consistent and available
- Easy to configure and manage
- Resilient
- Recoverable
- Proven

Business Practices

- Open, transparent interaction with customers
- Industry leadership
- Embracing of Open Standards

Top Security Challenges

- Reducing the frequency of security updates
- Rolling out security updates efficiently
- Implementing defense-in-depth measures
- Managing access in an extended enterprise
- Better guidance to secure systems

Microsoft's Security Progress

Microsoft Windows xp
Service Pack 2

- More than 260 million copies distributed
- 15 times less likely to be infected by malware
- Significantly fewer important & critical vulnerabilities

Microsoft Windows Server 2003
Service Pack 1

- Security configuration wizard
- More secure by design; more secure by default
- More than 4.5 million downloads

Microsoft Windows
AntiSpyware Beta

- Most popular download in Microsoft history
- Helps protect more than 25 million customers
- Great feedback from SpyNet participants

Microsoft Windows
Malicious Software Removal Tool

- 2B total executions; 200M per month
- Focus on most prevalent malware
- Dramatically reduced the # of Bot infections

As of January 2006

Microsoft's Security Vision Is Much More…

Establishing **trust** in computing to realize the full potential of an interconnected world

Microsoft's Security Focus

Strategy

A secure platform strengthened by security products, services and guidance to help keep customers safe

Technology Investments

- Excellence in fundamentals
- Security innovations

Prescriptive Guidance

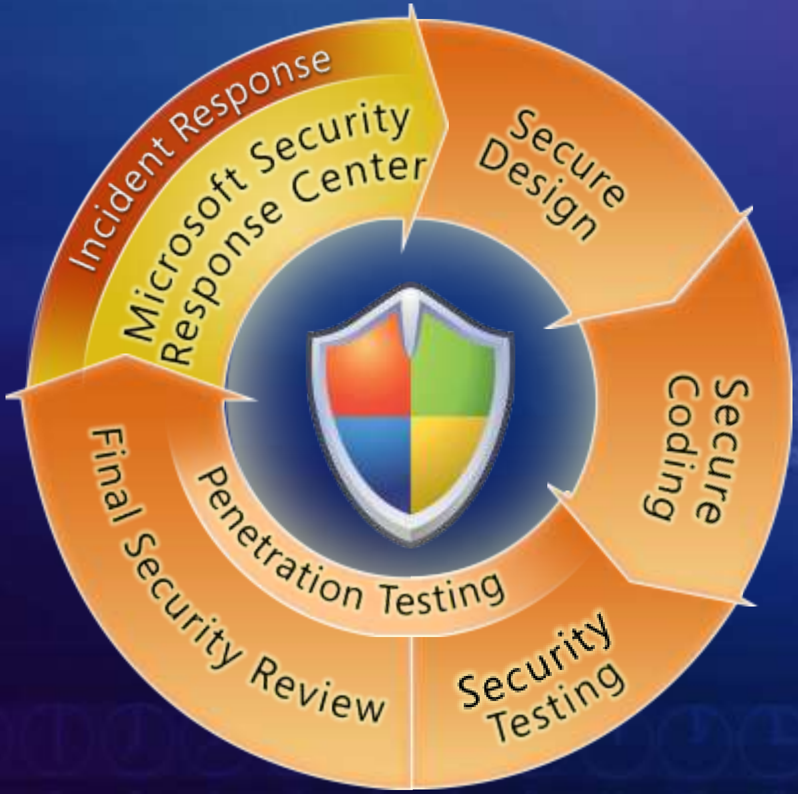
- Scenario-based content and tools
- Authoritative incident response

Industry Partnership

- Awareness and education
- Collaboration and partnership

Fundamentals

- Security Development Lifecycle
- Security Response Center
- Better Updates And Tools



Find Measures for Improvements

- #breaches
- #patches
- Time to Patch
- Time to discover vulnerability
- …

Terms

- CIA (Defensive Goals)
  - Confidentiality *(who can read what)*
  - Integrity *(who can write what)*
  - Availability *(who can access what)*
- STRIDE (Method of attack)
  - Spoofing (Confidentiality)
  - Tampering (Integrity)
  - Repudiation (Integrity)
  - Information Disclosure (Confidentiality)
  - Denial of Service (Availability)
  - Elevation of Privilege (Confidentiality, Integrity & Availability)
- DREAD (Measure of Risk)
  - Damage (Impact)
  - Reproducibility (Probability)
  - Exploitability (Probability)
  - Affected Users (Impact)
  - Discoverability (Probability)

STRIDE - Spoofing

- Spoofing threats allow an attacker to pose as another user or allow a rogue server to pose as a valid server
- Examples
  - Intercepting an HTTP authentication header and replaying it to spoof an authorized user
  - DNS Spoofing
    - Intercept DNS lookups and return an invalid address
  - DNS Cache poisoning
    - Fools ISP name servers for middleman attacks
- Question: Can someone –spoofll this system by pretending to be someone they are not?

STRIDE - Tampering

- Malicious modification of data
- Examples
  - Unauthorized changes to address on a credit account in a database
  - Alteration of data as it flows between two computers over an open network
  - Changing the contents of a file with weak ACLs to deface a website
    - January 2003, Microsoft New Zealand website defaced because of weak ACLs
- Question: What data would an attacker want to tamper with? How would they gain access to tamper with it?

STRIDE - Repudiation

- Users who deny performing an action without other parties having any way to prove otherwise
  - A user performing an illegal operation in a system that lacks the ability to trace the prohibited operations
- Nonrepudiation is the ability of a system to counter repudiation threats
  - User purchases an item and has to sign for the item upon receipt
- Question: What actions must we be able to prove?

STRIDE – Information Disclosure

- Disclosure of information to individuals who are not supposed to have access to it
  - Attacker reads credit card numbers from database
  - Attacker obtains database connection string from configuration file
  - Attacker obtains credentials from data in transit by setting up a ―man-in-the-middle‖ attack
  - ―Phishing‖ attacks
- Question: What information would an attacker want? How would they access it?

STRIDE – Denial Of Service

- Deny service to valid users
- Victims include Microsoft, Yahoo, SCO and many others
- Attacks launched by ―Zombies‖ (infected PCs)
  - Up to 50K machines participating
  - Often use malformed TCP packets
  - Or simple large ―Ping‖ messages
- Question: How would we prevent a DDOS attack?

STRIDE – Elevation Of Privilege

- An unprivileged user gains privileged access and thereby has sufficient access to compromise or destroy the entire system
  - Blaster
  - –Zombiell programs
  - Keystroke loggers
  - Slammer
  - Code-Red
- Question: How would an attacker elevate privilege on our system?