Microsoft

Threat Modeling

Deepak Manohar

SECURITY MATTERS
MICROSOFT SECURITY SUMMITS 2006

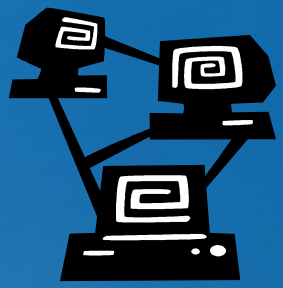
Outline

- Motivation
- Past Security Approaches
- Common problems with past security approaches
- Adversary's perspective Vs Defender's perspective
- Why defender's perspective?
- Threat Modeling Overview
- Knowledge gained from Previous versions
- ACE Threat Modeling V2.0
- Demo
- Conclusion

Motivation

- Why we should focus on Securing applications.
  - "75 percent of hacks happen at the application." - Lanowitz
  - Gartner predicts that if 50 percent of software vulnerabilities were removed prior to production use for purchased and internally developed software, enterprise configuration management costs and incident response costs each would be reduced by 75 percent.
- Why current security protection mechanisms are an afterthought or don't address the actual issue?
  - Firewalls, IDS are network/perimeter based attack detection systems
  - Security needs to be baked in
- Why Threat Modeling?
  - Cost of fixing bugs – Design time, Deployment time
  - Cost of identifying required security controls at design time vs post implementation

Age of Application Security



Desktop Security
- Anti-virus

Network Security
- Firewalls, IDS

Application Security
- Secure Development
- Code Scanning

Past Security Approaches

Penetration Testing

- Attempt to impersonate the adversary and “break-in”

Security Code Reviews

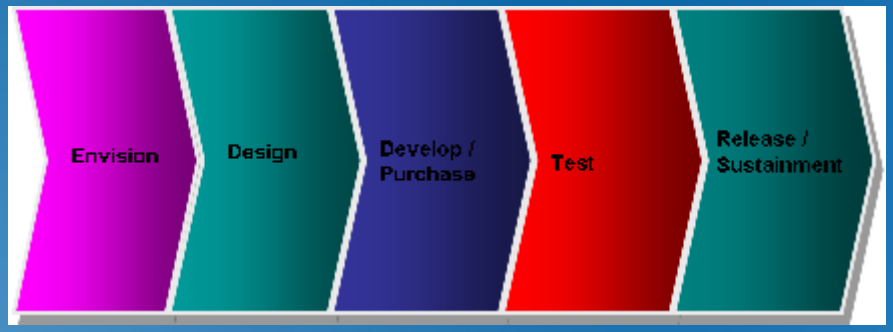
- Detect security flaws in code base

Security Design Reviews

- Detect security flaws in software architecture

Identify as many vulnerabilities as possible

Ideology similar to early testing ideology



Common Problems with Past Security Approaches

- No plan - almost random vulnerability searching
- No protection profile generated
- Similar treatment irrespective of Assurance level
- Looks at security from an ***adversarial*** perspective
- Cost Benefit analysis – not easy

Adversary's Perspective Vs Defender's Perspective

- Adversary's Perspective
  - Objective – Break the application
  - Looking for vulnerabilities that can used to carry out an attack
  - Vulnerabilities and attacks are simply a means to an end
  - Shortcoming – relationship between time and vulnerabilities discovered
- Defender's Perspective
  - Objective – Protect the application
  - Identify assets – rank assets
  - Protect assets
  - Build protection mechanisms (protection profile) around assets
  - Engineer security into solution

Why Defender's Perspective?

- What to protect?
- Prioritize effort based on asset value
- Build a protection profile
- Bake security in
- Better Cost Benefit Analysis

If you know the enemy and know yourself, you need not fear the result of a hundred battles. If you know yourself but not the enemy, for every victory gained you will also suffer a defeat. If you know neither the enemy nor yourself, you will succumb in every battle.

- Sun Tzu, *The Art of War*

Who we are – ACE

ACE – Application Consulting and Engineering

ACE Security

ACE Performance

ACE Engineering

ACE Services

Threat Modeling Overview

- Solid Understanding Application Architecture
- Systematically Identify and Rate threats
- Address rated threats with appropriate countermeasures

Threat Modeling Definitions

Threat → Possibility of something *bad* happening

- Realized through…

Attacks → How it happens (the exploit)

- Materialize through…

Vulnerabilities → Why it happens (the cause)

- Mitigated with…

Countermeasures → How to prevent it (the fix)

If a negative business impact cannot be illustrated, it's not a threat!

Knowledge gained from previous versions

- Application developers and security
- Threats need to be generated systematically
  - Avoid brainstorming threats
- Conversion from BRD/FSD/TSD
- Complexity of threat modeling process
  - Entry point/Exit point
  - Data flow diagrams
  - Control flow diagrams
- Application developers
  - Mentality
  - Time
- Difficult to quantify threat agent's skill

Objectives for ACE Threat Modeling?

Threat modeling methodology focused on typical enterprise IT (LOB) applications

Objective:

- Provide a consistent methodology for objectively identifying and evaluating threats to applications
- Translates technical risk to business impact
- Empower the business to manage risk
- Creates awareness of the security dependencies and assumptions

All without requiring security subject matter expertise

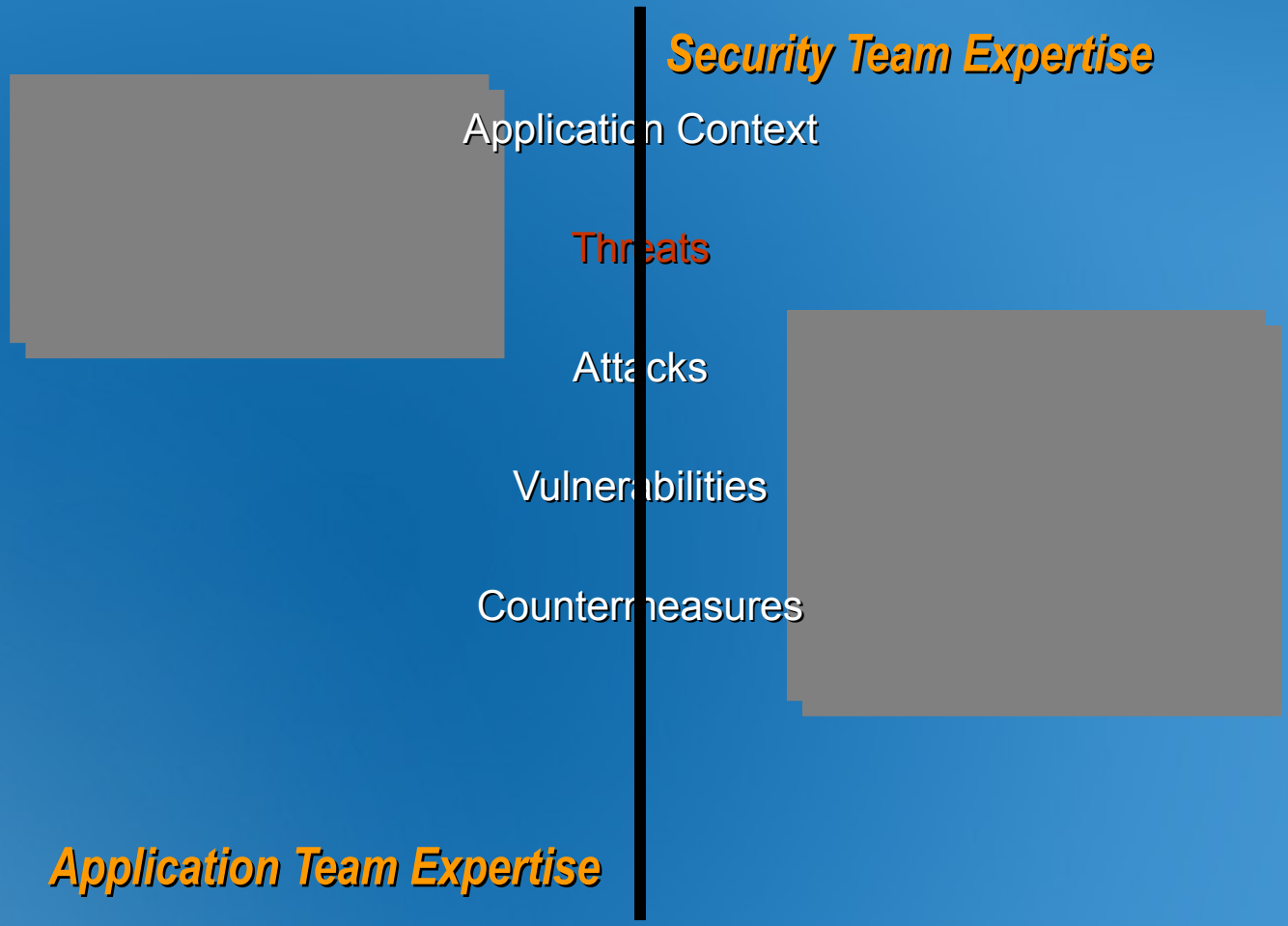
ACE Threat Modeling Benefits

- Benefits for Application Teams
  - Translates technical risk to business impact
  - Provides a security strategy
  - Prioritize security features
  - Understand value of countermeasures
- Benefits for Security Team
  - More focused Security Assessments
  - Translates vulnerabilities to business impact
  - Improved ‘Security Awareness’
- Bridges the gap between security teams and application teams

What is ACE Threat Modeling?

- Identify all possible threats to assets in applications
- Identify the misuse cases for an application
- Prioritize threats
- Provide controls to eliminate/mitigate threats
- Provide application specific countermeasures (security features)

Anatomy of a Threat

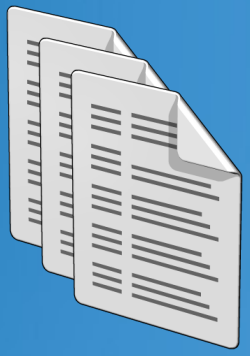


Decomposing the Application Context

Components

Roles

Data



Application Context Rules

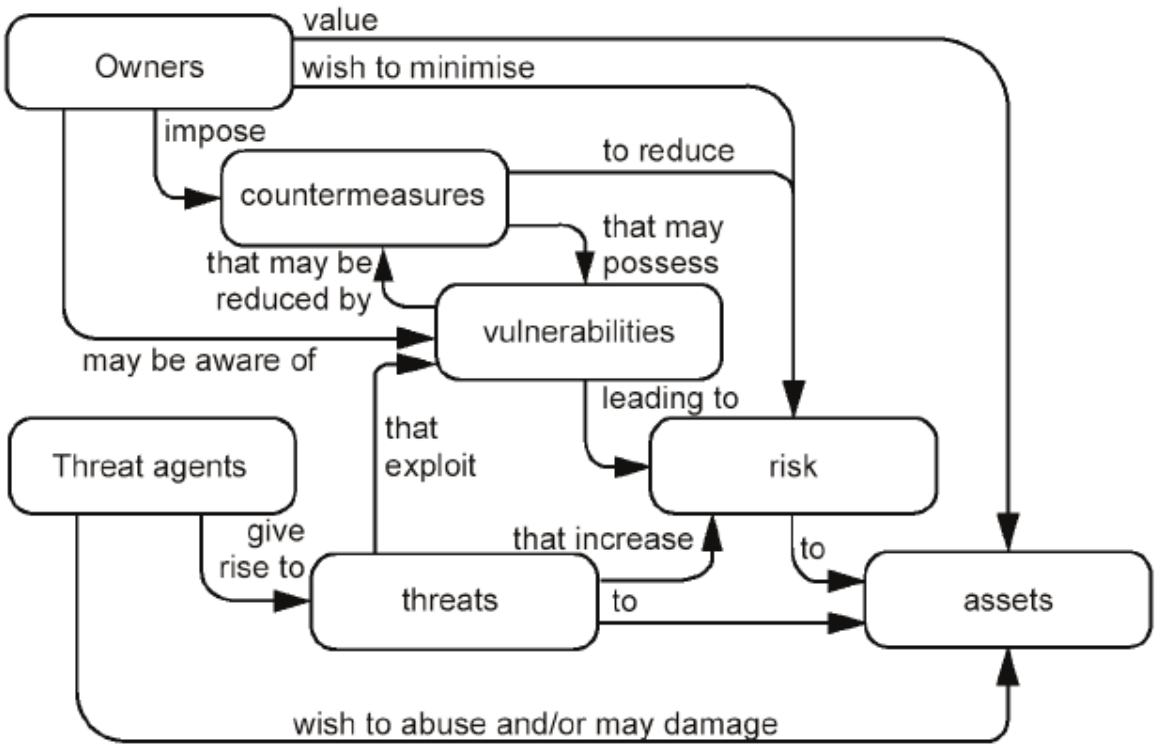
- Roles can interact with Components through defined Actions
- Components can interact with Components through defined Actions
- Data is stored inside Components
- Components can Create, Read, Update or Delete Data
- Data can flow between 2 interacting Components
- Data can flow between interacting Role and Component

Generating Threats

- Application Context defines allowable actions
  - Built by following our application context rules
- Systematic corruption of these actions are threats
  - Automatic Threat Generation

ISO 15408 - Security relationship

ISO/IEC 15408-1:2005(E)

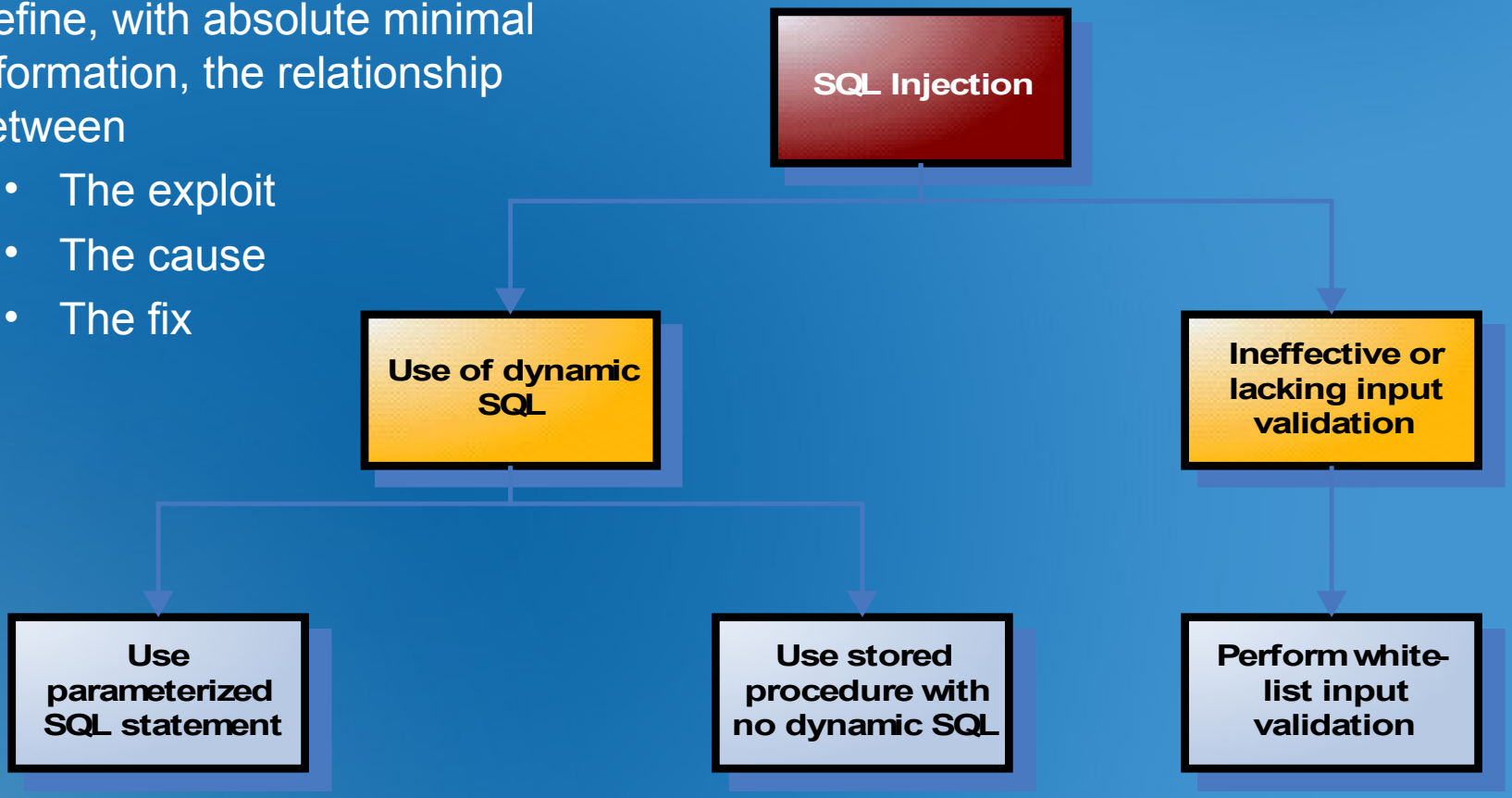


Attacks

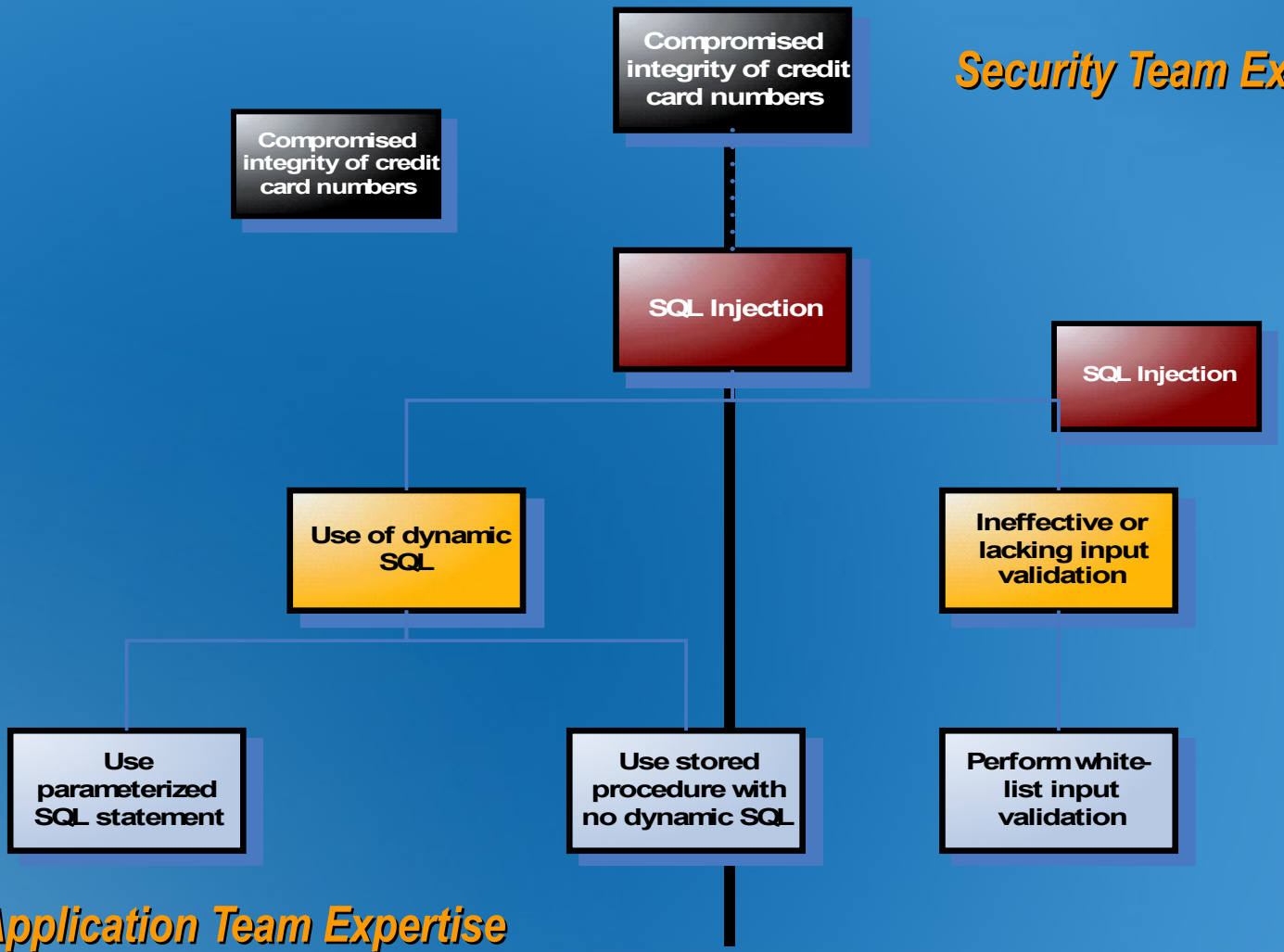
- Password Brute Force
- Buffer Overflow
- Canonicalization
- Cross-Site Scripting
- Cryptanalysis Attack
- Denial of Service
- Forceful Browsing
- Format-String Attacks
- HTTP Replay Attacks
- Integer Overflows
- LDAP Injection
- Man-in-the-Middle
- Network Eavesdropping
- One-Click/Session Riding/CSRF
- Repudiation Attack
- Response Splitting
- Server-Side Code Injection
- Session Hijacking
- SQL Injection
- XML Injection

Attack Library

- Collection of known Attacks
- Define, with absolute minimal information, the relationship between
  - The exploit
  - The cause
  - The fix



Threat-Attack Loose Coupling



Transparency with Attack Library

Application Context

Threats

Attacks

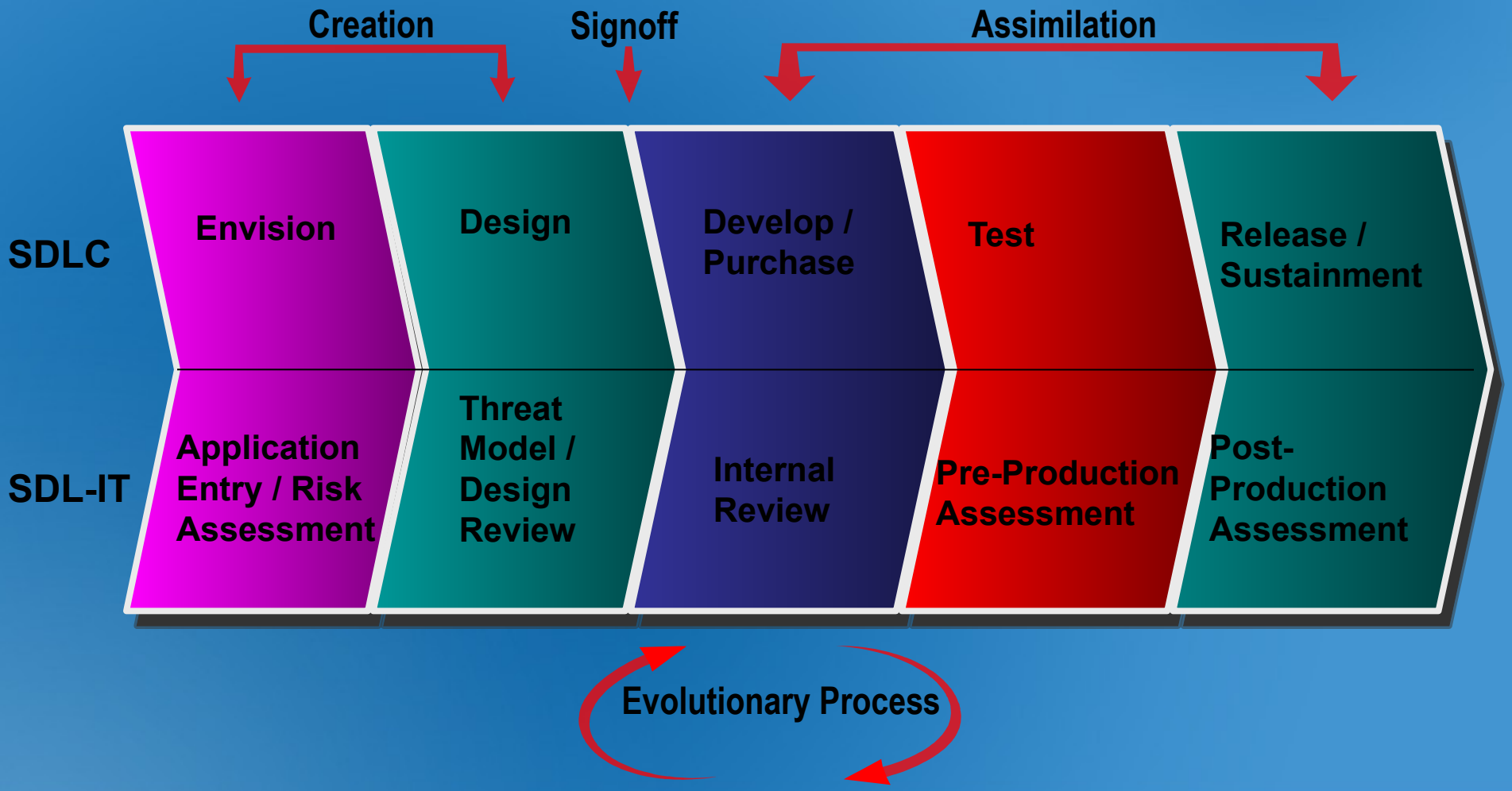
Vulnerabilities

Countermeasures

Threat Modeling & Security SMEs

- Attack Library created by security SMEs
  - Verifiable and repeatable
- Security SME provides TM completeness
  - Verifies that the threat model meets the application specifications
  - Plugs knowledge gaps in the threat model
    - New 0-day attack not part of the Attack Library
  - Performs potential optimization

ACE Threat Modeling during SDLC



Threat Analysis & Modeling v2.0

- Tool created to aid in the process of creating and assimilating threat models
- Automatic Threat Generation
- Automatic Attack coupling
  - Provides a security strategy
- Maintain repository of Threat Models for analysis*
  - Security landscape is evolving (new attacks, vulnerabilities, mitigations being introduced)

Threat Analysis & Modeling v2.0 (cont.)

- Analytics
  - Data Access Control Matrix
  - Component Access Control Matrix
  - Subject-Object Matrix
  - Component Profile
- Visualizations
  - Call/Data/Trust Flow
  - Attack Surface
  - Threat Tree
- Reports
  - Risk Owners Report
  - Design/Development/Test/Operations Team Report
  - Comprehensive Report

Tool Demo

Sample application

User Roles
- Unregistered users
- Registered users

Service Roles
- Webservice Role
- Database Role

Data
- Customer Accounts
- Customer CCs
- Product Information

Components
- Webservice
- Database

Use Cases
- Unregistered, Registered users read product information
- Unregistered Users can create customer accounts

Summary

- Methodology evolved from years of experience
- Methodology streamlined to minimize the impact to existing development process
  - Does not require security subject matter expertise
  - Collecting already known data points
- Consistent & objective methodology
- Methodology optimized for SDL-IT integration

Microsoft

Question and Answer

http://msdn.microsoft.com/security/securecode/threatmodeling/acetm/

SECURITY MATTERS
MICROSOFT SECURITY SUMMITS 2006